TREBALL DE FI DE GRAU

Grau en Estadística

Títol: Índex d'Incertesa basat en els discursos inaugurals dels presidents dels Estats Units (1929-2021).

Autor: Pep Murtra i Sabaté

Director: Salvador Torra Porras

Departament: Econometria, Estadística i Economia Aplicada

Convocatòria: Juny 2024

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Resum

La incertesa és present en el nostre dia a dia, però no disposem d'una mesura clara per avaluar-la. Sabem mesurar el risc, però la incertesa no compta amb regles establertes per determinar-ne el grau, cosa que impedeix associar-hi un valor numèric.

En aquest context, entren en joc l'anàlisi de textos, el Processament del Llenguatge Natural -NLP- i l'aprenentatge automàtic. Els textos ens proporcionen una gran quantitat d'informació, i la incertesa pot manifestar-se en aquests. En particular, els discursos polítics, pel seu paper fonamental, poden revelar elevats graus d'incertesa, per això és d'especial interès analitzar-los.

En aquest treball, es començarà analitzant diferents textos dels discursos inaugurals dels presidents dels Estats Units i s'acabarà desenvolupant un índex que mesurarà el grau d'incertesa. No obstant això, en l'estudi present, el procés de desenvolupament és molt important. Per tant, el "com" serà de gran importància. Això inclou el processament del llenguatge natural, la transformació de paraules a números amb l'aplicació del TF-IDF, la preparació del model escollit i, finalment, la creació de l'índex, que reflectirà, en certa manera, tots els passos anteriors i, consegüentment, pretendrà mesurarà el grau d'incertesa.

Paraules claus: incertesa, discursos, processament de llenguatge natural -NLP-, Freqüència dels terme -TF-, Freqüència Inversa dels Documents -IDF-, aprenentatge automàtic, mètriques d'avaluació, AdaBoost classifier, S&P 500, probabilitat a posteriori.

Abstract

Uncertainty is present in our daily lives, yet we lack a clear measure to evaluate it. We know how to measure risk, but uncertainty does not have established rules to determine its degree, which prevents us from assigning it a numerical value.

In this context, text analysis, Natural Language Processing -NLP-, and machine learning come into play. Texts provide us with a vast amount of information, and uncertainty can manifest within them. In particular, political speeches, due to their fundamental role, can reveal high degrees of uncertainty, making their analysis of special interest.

In this work, we will begin by analysing various texts from United States presidents' inaugural speeches and conclude by developing an index to measure the degree of uncertainty. However, in the present study, the development process is particularly significant. Therefore, the "how" will be of great importance. This includes natural language processing, the transformation of words into numbers, the preparation of the chosen model, AdaBoost, and finally, the creation of the index, which will reflect, to some extent, all the preceding steps and, consequently, the degree of uncertainty.

Key words: uncertainty, speeches, Natural Language Processing -NLP-, Term Frequency -TF-,Inverse Document Frequency -IDF_, Machine Learning, evaluation metrics, AdaBoost classifier, S&P 500 index, posterior probability.

Classificació AMS

62-XX STATISTICS

62-07 Data analysis

62Q06 Statistical tables

68-xx COMPUTER SCIENCE

68Txx Artificial intelligence

68T50 Natural Language processing

68Wxx Alogorithms

68W01 General

Índex

1. Introducció

El llenguatge, en política, exerceix un paper fonamental en la percepció del discurs; cada paraula compta per al ciutadà a l'hora d'interpretar allò que escolta. Els polítics seleccionen curosament cada expressió per comunicar el que volen transmetre, per això, usant el llenguatge com a l'eina principal a través de la qual, els líders polítics comuniquen les seves idees. La capacitat d'influència d'un polític sobre la població és considerable, arribant fins i tot a afectar les empreses i els seus inversors. Cal tenir present, que un discurs pot generar una connexió emocional, ja sigui perquè els ciutadans se senten escoltats pel polític o perquè comprenen les preocupacions que aquest manifesta.

Com va dir Michel Foucault, el discurs és una superfície que engloba la lluita política, essent una arma de poder, de control i de subjecció (Nosetto, 2017:51). Per tant, el que un president diu o deixa de dir influeix tant en les persones com en l'economia. A més a més, aquestes paraules poden crear realitats, generar expectatives, prometre futurs fets i, en conseqüència, influir en la percepció de la realitat. D'altra banda, si el discurs adopta un to econòmic, pot afectar en diversos aspectes d'aquesta, com la inversió i el desenvolupament. També es poden percebre les incerteses o les certeses que un polític transmet a través de les seves paraules. El que s'extreu d'aquesta influència dels discursos, és la possibilitat de dur a terme una investigació precisa i obtenir resultats empírics per tal d'establir relacions, com és el cas que ens ocupa en aquesta anàlisi. Al llarg de la història, han existit diferents discursos icònics en l'àmbit polític. Un exemple és el discurs d'acomiadament de Ronald Reagan, el 40è president dels Estats Units, amb la famosa frase "*We the People tell the government what to do; it doesn't tell us*." Un altre és el discurs inaugural de John F. Kennedy el 1961, en què va proclamar: "*My fellow Americans: ask not what your country can do for you—ask what you can do for your country*." També és destacable el discurs de Winston Churchill el 1940, amb la famosa frase "*We shall fight on the beaches*." La història és plena de discursos polítics amb frases cèlebres. El llenguatge és una eina poderosa; a través de les seves paraules, els líders poden inspirar, motivar i, de vegades, canviar el curs de la història. Aquest treball se centra en l'anàlisi d'un conjunt de textos dels últims 24 presidents dels Estats Units per la posterior creació d'un índex que mesuri el grau d'incertesa.

En l'era la informació tenim dades a tot arreu. Aquestes dades si s'analitzen correctament poden revelar patrons i tendències molt avantatjosos per la societat i les empreses. Però no tota la informació s'emmagatzema en forma de dades numèriques, sinó que hi ha informació que ho fa, a través de textos, llibres i articles, entre d'altres. L'any 2016 Google estima que hi havia 143.064.880 llibres (Lara, 2021). Segons les estadístiques de Internet Live Stats, fins al 2016 hi havia 1.880 milions de llocs web (Roa, 2021). Aquesta xifra demostra la gran quantitat de textos que tenim a l'abast, i no només textos, sinó la gran quantitat d'informació que podem recollir.

El treball neix de la voluntat i gairebé obsessió de poder extreure la màxima informació possible de la manera com la societat s'expressa, mostrant la possibilitat que la informació no estigui perduda, independentment de com estigui expressada, alhora que s'intenta mantenir l'objectivitat en l'explicació dels resultats. Diverses motivacions impulsen a continuar amb l'anàlisi. En primer lloc, la curiositat generada per la magnitud, de l'habilitat de poder extreure resultats numèrics d'un contingut que originalment era textual, sigui en una xarxa social o en un discurs. D'altra banda, en el procés de transformació del text a dades numèriques, sorgeix el repte de treure informació d'un discurs possiblement subjectiu i assegurar que aquesta informació sigui el màxim de verídica possible, evitant que la subjectivitat inicial provoqui una

desconfiança en els resultats finals. A més a més, tenir la possibilitat d'assignar un valor numèric a un concepte com la incertesa, genera un repte molt atractiu. En el present treball, es presenta el repte de mesurar el grau d'incertesa. Un terme relacionat amb incertesa, és el risc, com diu Hillson, i per tant la incertesa ens explica o deixa d'explicar informació incompleta i no mesurable (Dönmez i Grote, 2018:96). En conseqüència, tenim el risc sense quantificar. En el present treball, es buscarà poder assignat un número a aquesta incertesa, així obtenint un grau d'aquesta que ens pugui ajudar a prendre decisions. Deixant quantificades tots els vessants del risc. En el present anàlisis, és realitzat, a partir de la convicció que la incertesa pot ser mesurada i que aquesta mesura i juguen un gran paper els discursos polítics i el llenguatge en general, ja que com diu Medvid si l'oratòria és junta amb la política, s'obtenen un seguit d'influències a la vida de les persones (Medvid et al., 2022:145), És per això, que la hipòtesi principal de l'anàlisi, és que la millor manera de mesura la incertesa és a partir dels textos, i que en cas que els resultats esdevinguin els no desitjats, aquest es veuran afectats en gran manera, per com s'han treballat els textos, i no tant pel "que". L'anàlisi té com a objectiu principal, la creació d'un indicador, que ens pugui donar referències sobre el grau d'incertesa que està present en l'economia d'un país. És a dir, l'objectiu és trobar un canal, per mesurar la incertesa, i comprovar, a través del S&P500, la utilitat d'aquest.

És necessari considerar diversos desafiaments presents en l'anàlisi de textos en l'àmbit computacional i, per això, i en cerca de la creació d'un índex d'incertesa i de la sensibilitat de la base de dades, cal tenir molta cura, del "com" és creat l'índex, dels passos previs a aquest, per això, en primer lloc, i juga un paper clau l'aplicació de tècniques de Processament del Llenguatge Natural -NLP-. Aquest tractament dels textos, esdevé menys intuïtiu per diversos motius. Un d'ells és que per a cada individu o text, no hi ha un únic valor, sinó diversos, fet que fa impossible tractar el valor o el caràcter com una unitat única. Un altre motiu és que paraules que són idèntiques per a l'ull humà, com ara *build* o *building*, són considerades diferents pel programa. És crucial reduir la dimensionalitat al màxim possible, i aquí els NLP juguen un paper clau. Un bon processament facilita la construcció del nostre futur model. No obstant això, l'aplicació dels NLP també comporta el repte d'haver de passar per alt certs detalls del text. Per això, aquesta fase ha de tractar el text de manera sistemàtica, identificant i processant els elements més rellevants per a l'anàlisi. L'ús del NLP és fonamental per treballar amb bases de dades textuals extenses, assegurant que es puguin extreure *insights* valuosos sense perdre informació crucial. A més, el tractament computacional dels textos implica altres reptes com la desambiguació semàntica, l'extracció de sentiments i la identificació de patrons lingüístics. Cada un d'aquests aspectes requereix algoritmes sofisticats i models de dades robustos capaços de gestionar la complexitat i la variabilitat inherent en els textos humans.

Altres objectius, secundaris, són l'aplicació de tècniques i mètodes, per tal de passar d'uns texts a números i treure informació d'us rellevant. A més a més, aquest estudi buscarà proporcionar una base per futures investigacions en aquesta àrea, contribuint a una comprensió més profunda de com la comunicació política pot influir en les decisions dels inversors i, per extensió, en els mercats financers. Tot aquest procés ha de passar, per un model d'aprenentatge automàtic, i en el cas de l'estudi ho farà a través del model classificatori AdaBoost. L'índex creat serà, l'eix central del treball, ja que proporcionarà una mesura quantitativa de la incertesa de l'economia. El repte d'aquest treball radica en aquest índex, pel fet que, convertir un llenguatge subjectiu en una mètrica numèrica sobre la qual es pot treballar, extreure conclusions i comparar de manera objectiva, presenta certes dificultats per tal d'incloure l'objectivitat més gran possible en l'anàlisi. Per avaluar l'eficàcia de l'índex i comprendre la seva utilitat, es compararà amb l'índex S&P 500. Tot i que es podrien haver triat altres índexs com el *Dow Jones Industrial Average* -DJIA-, que inclou les 30 empreses més grans dels Estats Units, el

Nasdaq Composite, que inclou totes les empreses que cotitzen en la borsa de valors NASDAQ, o el S&P 100.

El treball es divideix en 11 apartats, dels quals el primer, que és el present, introdueix el tema de l'anàlisi, així com els objectius i les hipòtesis formulades. A continuació, es presenta la metodologia de treball utilitzada per obtenir els resultats. En aquest apartat es detallen les llibreries emprades per a la creació del codi, així com altres aspectes rellevants per a l'elaboració de les diferents parts del treball. La tercera secció mostra la teoria en la qual es basa el treball, indicant d'on s'ha extret cadascun dels mètodes utilitzats en la part pràctica, així com el rerefons matemàtic dels diferents mètodes emprats. Les seccions 4, 5, 6, 7, 8, 9 presenten les diferents fases de la part pràctica de l'estudi, seguint un ordre cronològic: la secció 4 correspon a la primera fase realitzada, i així successivament. En aquests apartats si poden apreciar fins a 8 fases que corresponen a les fases seguides a la part pràctica del treball. A l'apartat 10 es presenten les conclusions, on es fa una síntesi dels resultats obtinguts i es desenvolupen possibles línies futures d'investigació. A l'apartat 11 es detallen els diferents recursos utilitzats per a l'elaboració teòrica, que posteriorment s'han aplicat en la part pràctica. Finalment, es troben els annexos, on es presenten diferents gràfics utilitzats per millorar l'enteniment de certes parts de l'anàlisi, així com fragments del codi desenvolupat. Finalment, en l'apartat 12, si poden trobar els annexos, on es veuen alguns dels gràfics usats, així com, les *stopwords* eliminades, el diccionari de paraules utilitzat i el codi fet servir en la part pràctica.

Agreixo al professor Salvador Torra Porras per guiar-me i il·luminar-me en el camí de l'anàlisi de textos. També agreixo a la meva parella i a la meva família el suport i els ànims que m'han donat en tot moment.

2. Metodologia

En el present apartat, es busca explicar la manera en la qual s'ha portat a terme la part pràctica d'aquest treball. Per fer-ho, caldrà seguir cronològicament les diferents parts de l'anàlisi, ja que, en cas contrari, no seria entenedor.

En primer lloc, és imprescindible considerar l'estat actual de la qüestió en la temàtica d'estudi. En aquest treball, diversos anàlisis previs marquen el context de l'anàlisi. Com s'ha mencionat anteriorment, s'esmenta l'estudi de Avela i Lehmus (2020), el qual desenvolupa un índex d'incertesa basat en titulars de notícies finlandeses. Aquest article detalla els passos necessaris per treballar amb textos, presenta conceptes bàsics de NLP i proposa diversos models, entre ells una variant del model *Naive Bayes* coneguda com *Transformed Weight-normalized Complement Naive Bayes* -TWVNB-. A més, aquest estudi discuteix altres mètodes de NLP que són utilitzats en l'anàlisi actual. En relació amb els NLP, un altre influent article és el de Tabassum i Patil: "*A survey on text pre-processing & feature extraction techniques in natural language processing*", que proporciona informació clau per seleccionar els passos de processament. Tot i això, el llibre de Bird et al., 2009, ha portat les eines per trobar les llibreries adequades per diferents aspectes del treball, des de la importació de textos fins a l'aplicació dels NLP.

Respecte als models d'aprenentatge automàtic, aquesta anàlisi es basa en diversos articles, com el de Robert Schapire, pioner de la metodologia de *boosting* que es farà servir en aquest treball. Aquesta metodologia va ser posteriorment millorada per Freund el 1996, i posteriorment per Bloehdorn i Hotho el 2004. Per tant, aquests quatre autors són crucials en la selecció del model. El mateix Schapire, juntament amb Freund, són referència en aquesta anàlisi per entendre el model *AdaBoost*, ja que van introduir aquest algoritme de *boosting* amb la publicació de l'article "*Experiments with a new boosting algorithm*". L'estudi de Farzi i Bolandi té força pes, a causa de la informació explicada en aquest, que ha permès tenir una comparativa dels diferents models d'assemblatge existents, i amb el seu estudi *Estimation of organic facies using ensemble methods in comparison with conventional intelligent approaches: A case study of the South Pars Gas Field, Persian Gulf, Iran, Modeling Earth Systems and Environment*, ha permès al present treball, entendre no sols l'algoritme usat, sinó d'altres.

La part pràctica d'aquest estudi ha sigut realitzada a través del llenguatge de programació *Python*, a causa de com diu el llibre de Bird et al., 2009,

> "Python is a simple yet powerful programming language with excellent functionality for processing linguistic data".

La metodologia inicial emprada en aquest estudi es basa en els primers passos en el llibre Bird, El qual és un llibre, com el seu nom indica, sobre Processament del Llenguatge Natural. En aquest, hi trobem la principal llibreria usada en aquesta anàlisi, que és la *Natural Language Toolkit* -NLTK-. Aquesta inclou extens programari, dades i documentació totalment gratuïta (Bird et al., 2009:xiv). Aquesta llibreria va ser creada el 2001 pel *Department of Computer and Information Science* de la Universitat de Pennsylvania (Bird et al., 2009:xiv). És a partir d'aquesta llibreria de la qual podem extreure diversos textos per a l'anàlisi. En el cas d'aquest estudi, els discursos escollits són del paquet *book*. Aquest paquet conté diversos textos, entre els quals es troba: el text *Inaugural Address Corpus*. Aquest conté tots els discursos dels presidents dels Estats Units, des de George Washington el 1789 fins a Joe Biden el 2021, per tant, es tenen 59 textos. Un cop s'han carregat els textos, cal fer una sèrie de

modificacions, com, introduir els textos en un *data.frame*[1], que permeti accedir a la base de dades més fàcilment, per fer-ho ha calgut l'ús de la llibreria *panda*, creada el 2008 per l'empresa AQR Capital Management (*Pandas - Python Data Analysis Library*, n.d.). Aquesta llibreria ofereix la possibilitat de treballar amb *data.frames* ràpid i eficientment, amb un seguit de funcions, que permeten la manipulació de l'objecte sense gaire complexitat -aquesta és importada a *Paython*, a través de l'àlies *pd*-. Seguidament, és necessari modificar certs aspectes de la base de dades originals, com per exemple separar, en columnes diferents, el nom del president i l'any del discurs. En els passos següents, s'ha fet ús de la totalitat dels discursos inaugurals dels Estats Units, és a dir, des de George Washington el 1789 fins a Joe Biden el 2021. Això és degut, a què s'ha cregut que el model, aportarà millors resultats amb la totalitat de la base de dades, tot i això, pels resultats finals, s'han usat els últims 24 registres de la base de dades.

A continuació, cal conèixer la base de dades, saber la llargada dels discursos, les paraules úniques i el nombre de discursos, entre d'altres. Per obtenir el nombre de caràcters del text, utilitzem la funció *len(text)*, la qual ens retorna la longitud total del text en termes de paraules i símbols de puntuació que apareixen en el text (Bird et al., 2009:7). Per mostrar els resultats d'aquest càlcul, és necessari instal·lar la llibreria *matplotlib* per *Python*. Aquesta funció permet crear gràfics, i visualitzar-los (*Pyplot Tutorial — Matplotlib 3.9.0 Documentation*, n.d.) en l'estil MATLAB[2], l'ús d'aquesta llibreria ha estat present en tot el codi de *Python*, i tots els gràfics de font pròpia han estat fets a través d'aquesta eina. En aquest procés de conèixer la bases de dades, cal calcular diferents mesures que ens ajuden a entendre el nombre de caràcters, com la mitjana, la mediana, el rang i la variància, entre d'altres. Per poder realitzar aquest càlcul, s'ha utilitzat la llibreria *pandas*, com s'ha comentat anteriorment. Tot seguit, s'ha volgut determinar el nombre de paraules de cada text. Per fer-ho, ha calgut separar el text en paraules i comptar la llargada total de les paraules. A partir d'aquest resultat, s'han calculat les paraules úniques. Per fer-ho, s'ha separat el text en paraules i, mitjançant la funció *set()*, s'han mantingut només les paraules úniques, pel fet que aquesta funció no permet duplicats. Per tant, en tenir el nombre de paraules i les paraules úniques, ja podem obtenir la freqüència de paraules úniques de cada text. A més a més, és essencial identificar les paraules més freqüents, ja que això pot donar-nos una idea de l'estil del discurs de cada president. Per a això, s'utilitzen diverses llibreries com *pandas* o *matplotlib.pyplot*. La més important en aquest cas és la llibreria *wordcloud*, la qual permet la creació d'un núvol de paraules. Per generar aquest núvol de paraules, és necessari convertir tot el text a minúscules, separar-lo en paraules i, posteriorment, comptar quants cops apareix cada paraula en el text.

Després de completar la fase de descriptiva abans del processament, és necessari procedir amb el processament dels textos. La metodologia seguida consisteix en diverses etapes, aquestes etapes, han sigut fetes seguint un raonament. En primer lloc, es realitza l'eliminació de signes de puntuació i caràcters especials. Per a això, s'utilitza la funció *str.replace* de la llibreria *string*. Aquest procés inicial implica la supressió de signes de

[1] Un data.frame is una estructura de dades organitzada en una taula de 2 dimensions, on trobem columnes i files. ("What are DataFrames?", n.d.))

[2] Llenguatge basat en matrius que permet l'expressió més natural de les matemàtiques computacionals (Componentes Electrónicas LTDA., 2022)

puntuació, ja que aquests elements no són interpretats pel codi i només introdueixen soroll (Tabassum i Patil, 2020:4865). Un cop s'ha confirmat l'extracció dels signes de puntuació dels textos, es procedeix a revisar altres caràcters potencialment sorollosos en l'anàlisi. En el context d'aquest estudi, s'han identificat caràcters com `\x84|\x94` o `¡`, que també s'eliminen per assegurar una depuració exhaustiva. En segon lloc, és crucial la *tokenització* dels textos, és a dir, la seva separació en *tokens* o paraules individuals. Aquest procés s'efectua mitjançant la funció word_*tokenize* per obtenir una llista de paraules (Bird et al., 2009:80).

En tercer lloc, s'ha de filtrar les paraules que no aporten significat en el context dels NLP conegudes com a *StopWords*. La identificació i eliminació d'aquestes *StopWords* contribueixen significativament a millorar els resultats dels algoritmes aplicats (Tabassum i Patil, 2020:4865). En aquesta fase s'ha inclòs un paquet dins de la llibreria NLTK, anomenat *stopwords*, on s'inclouen totes les paraules considerades irrellevants (Bird et al., 2009:61) o sense informació. En l'annex, es recullen aquestes paraules. Cal precisar que les paraules són en anglès, igual que els textos de l'estudi. Un cop s'han carregat les llibreries, cal eliminar les paraules que pertanyen a la llista de *stopwords*. A més a més, durant aquest mateix procés, cal convertir totes les paraules en minúscules, per evitar que el codi identifiqui com a diferents, paraules que són iguals, com per exemple, *The* i *the* (Bird et al., 2009:61). En aquest procés, és necessari revisar els textos i les paraules per assegurar-nos que els diversos passos s'han dut a terme adequadament. En quart lloc, cal reduir la dimensionalitat de les paraules, és a dir, fer que aquestes s'assemblin el màxim possible per tal de reduir el nombre de disconformitats en la base de dades. En aquest procés, apareix el *stemming*. La llibreria *NLTK* presenta diferents tipus de *stemmers,* com ara el *PorterStemmer*` o el *LancasterStemmer* (Bird et al., 2009:107) en el nostre cas, usarem el *SnowBall stemmer* (GeeksforGeeks, 2022), usarem la llibreria *nltk.stem* i el paquet *SnowballStemmer*.

Per tal de prosseguir amb l'anàlisi, el següent pas consisteix a aplicar la mateixa metodologia descriptiva utilitzada anteriorment, però aquest cop emprant el text processat. L'objectiu és obtenir un text amb la mínima dimensionalitat possible sense perdre informació rellevant. Malgrat això, és necessari tornar a unir les paraules per tal de facilitar el càlcul dels valors anteriors, per a la qual cosa utilitzarem la funció *join().* Un cop completada aquesta operació, cal recalcular el nombre de caràcters, així com els diversos estadístics associats a aquests. El mateix procés s'ha de continuar amb el recompte de paraules úniques. A més a més, es tornarà a generar el gràfic de núvol de paraules realitzat prèviament. Aquesta part ens permetrà avaluar la necessitat d'aplicar tècniques d'NLP, així com entendre millors la base de dades. En aplicar aquestes tècniques, els textos es modifiquen, i aquests canvis ens ajuden a identificar les primeres tendències de cada text. Per exemple, es poden detectar paraules que es repeteixen amb freqüència o possibles errors de processament que hagin pogut passar desapercebuts. D'aquesta manera, no només es busca optimitzar l'anàlisi textual, sinó també millorar la qualitat del processament i assegurar-se que els textos tractats proporcionin informació precisa i útil per a l'estudi.

En aquest punt, disposem d'un conjunt de textos que ja han estat processats, tot i que encara falten diversos passos per preparar la base de dades. Els models d'aprenentatge automàtic sempre requereixen dades que hagin estat etiquetades prèviament (Avela i Lehmus, 2020:7). Per tant, en aquesta fase és necessari seleccionar els textos i etiquetar-los de manera adequada. S'ha optat per un sistema de classificació binària. La raó d'aquesta decisió radica en la gran subjectivitat que envolta la incertesa, la qual complicaria el model si s'introduïssin diverses categories. La frase de Hillson exemplifica aquesta subjectivitat quan afirma que "la incertesa és un esdeveniment desconegut d'un conjunt desconegut de resultats possibles"

(Hillson, 2003:27). Per tant, s'han definit dues classes: *Uncertainty* i *no uncertainty*. És important destacar que l'objectiu de l'estudi és construir un indicador del grau d'incertesa, per tant, les classes han de ser orientades en aquesta direcció. La manera de definir les classes és a través d'un diccionari de paraules incertes, que consisteix en 52 paraules sinònimes d'incertesa, extret, en gran part de la pàgina web Thesaurus.

El següent pas consisteix a aplicar el *stemming* a les paraules del diccionari d'incertesa, amb l'objectiu de facilitar els passos d'etiquetatge posteriors. Seguidament, és necessari definir com es vol que el codi compti les paraules. Per això, s'han cercat totes les paraules del diccionari en cada text i s'ha realitzat un recompte de totes les paraules incertes en cada text. Cal recordar que en aquest punt el text està en format *token*, el que facilita la cerca i el recompte de paraules totals. A continuació, s'ha realitzat el següent càlcul per determinar la freqüència de les paraules incertes en cada text:

$$\text{Uncertainty words} = \frac{\text{Nº Uncertainty words}}{\text{Total words}} \quad (1)$$

És necessari estandarditzar els resultats per evitar que les diferències entre textos provinguin, no del procés d'etiquetatge, sinó de les mateixes distincions entre textos (Muralidharan, 2010:1). Un cop tenim els valors per cada text, cal determinar com separem els textos, per tal de tenir una distribució de les classes balancejada, per fer-ho usem la mitjana dels valors recentment calculats i dividim la base de dades en *Uncertainty* i *no uncertainty*.

Un cop tenim la base de dades netejada i processada, ja tenim les eines suficients per començar a crear el model. Però, per treballar el model, cal separar la base de dades entre les dades d'entrenament i les de prova. Per fer-ho, s'introdueix per primer cop, la llibreria Sckit-learn. Aquesta va néixer el 2007, i va ser creada per David Cournapeu en un projecte de Google (Maddukuri, 2022), aquesta implementa diferents algoritmes d'aprenentatge automàtic, validació creuada i visualització (GeeksforGeeks, 2024b), aquesta llibreria serà clau en els pròxims passos i serà clau a l'hora de crear el model i testejar aquest. Entre les opcions que presenta la llibreria *Sklearn* trobem el mètode *traint_test_split(),* aquesta ens permetrà separar la base de dades en 4 subconjunts, dos corresponents a les característiques, és a dir els textos, un per entrenament i l'altre per prova, i els 2 subconjunts restants hi tindrem les etiquetes dels textos, tant per l'entrenament com per les proves (*Train_Test_Split*, n.d.). Per altra banda, s'ha de definir com fem la separació, per fer-ho s'ha testejat el model, i s'han provat diverses separacions, però el que ha donat millors resultats, és la separació 60-40, 60% de dades per l'entrenament i 40% per les proves. Seguidament, s'ha determinat un valor de llavor, per assegurar que cada cop que s'executa el codi, la separació és la mateixa. Per altra banda, en dividir la base de dades, hi ha la possibilitat que la divisió no mantingui la distribució de les classes prèvia, per això, cal usar el paràmetre *stratify* per continuar tenint la mateixa distribució.

Però encara tenim la base de dades en format de llenguatge natural, i el nostre classificador no té coneixements de les relacions entre paraules. Aquest pas el realitzem mitjançant l'aplicació de la freqüència de terme -*Term frequency* en anglès, TF- i la freqüència inversa del document -*Inverse Document Frequency* en anglès, IDF-. Així, busquem assignar un pes a cada paraula en cada discurs segons la seva singularitat, capturant la rellevància entre les paraules i els textos (Yun-tao et al., 2005:49). Utilitzant la funció *TfidfVectorizer*, iniciem aquest procés de transformació del llenguatge natural a dades numèriques. En aquest punt, definim paràmetres crucials per a l'aplicació del model. Aquest mètode ens permet filtrar les paraules que apareixen amb molta freqüència i que no són *StopWords*. En el context del nostre

estudi, aquestes podrien ser paraules com *people* o *america*, entre d'altres. Per això, establim que en la transformació de la base de dades, s'ignorin els termes que apareixen en més del 80% dels documents, i, de manera contrària, filtrar les paraules que apareixen en menys del 5% dels documents, o sigui, aquelles que només apareixen en 3 documents o menys. Aquest procés es du a terme per dos motius: primer, per reduir encara més la dimensionalitat del conjunt de dades; segon, per evitar la inclusió de soroll causat per paraules que apareixen tan sovint que no aporten valor, o per paraules que apareixen tan poc que perden rellevància per al model. Finalment, només s'utilitzaran unigrames[3].

Un cop realitzat el TF-IDF, es pot procedir a la creació del model. Cal mencionar primerament que aquest model es crearà utilitzant la llibreria *Scikit-learn*. Inicialment, s'ha de definir l'estimador dèbil, que es combinarà a través de les diferents etapes del model per obtenir un model més potent (Bloehdorn i Hotho, 2004:154). Com mostren diversos estudis, com els de Freund i Schapire, o Bloehdorn i Hotho, l'estimador base normalment és un arbre amb un sol node, conegut com a *weak learner*. Una vegada creat l'estimador dèbil, cal establir els paràmetres del model *AdaBoost*. Primer, cal determinar el nombre d'estimadors necessaris per al model; un nombre diferent implica un nombre diferent de classificadors dèbils (Gao i Liu, 2020:3). A més, és necessari establir el pes que es donarà a cada classificador en cada iteració, com més gran sigui el valor més pes tindrà cada classificador, els valors poden anar de 0 a 1.0. Per fer-ho, es fa servir un *grid search* per trobar els paràmetres ideals del model. Aquest procés cerca entre una combinació de possibles valors dels paràmetres, per determinar quina combinació aporta una millor exactitud al model. En el cas del model d'aquest estudi, els paràmetres triats han estat 300 estimadors i una taxa d'aprenentatge de 0.9.

La mateixa llibreria de *sklearn* aporta diferents paquets per avaluar el model, el paquet *classifcation report*, que ens mostra les mètriques de p*recision, recall, F1 Score, Accuracy, Support i Macro Average* (Prasanna, 2024). A més del paquet *confusion_matrix*, que, aporta una matriu amb els valors verdaderament positius i negatius, i els falsos positius i negatius (*Confusion_Matrix*, n.d.). A continuació, és necessari aplicar una validació creuada per determinar si el model està patint *overfitting* o *underfitting*. Per validar el nostre model, utilitzarem una validació creuada de 3 divisions, de manera que tindrem tres subconjunts de dades. En cada iteració dels plegaments, entrenarem el model amb dos subconjunts i el provarem amb l'altre. La validació es considerarà completada quan tots els subconjunts hagin passat per totes les fases. Posteriorment, és precís la creació de la corba ROC i el valor AUC. Aquest s'han calculat de forma que es pugui tenir un resultat de la corba, com del valor de l'AUC, en forma d'interval, a causa dela possible varietat que pot presentar el model.

Tot just el model ja ha sigut avaluat, cal començar a calcular l'indicador d'incertesa, aquesta ha sigut calculat, usant tres mesures, la primera, ja l'hem comentada, que és el valor del TF-IDF, de cada paraula, la segona és, el pes de la paraula calculada pel model. Aquest valor es calcula com la reducció total del criteri que aporta aquesta característica. També es coneix com la importància de Gini (*AdaBoostClassifier*, n.d.). Per altra banda, també s'haurà de tenir en compte el nombre de cops que la paraula apareix en el text. L'índex és calculat tal que:

[3] Paraules úniques (Nguyen, 2024).

$$UNC_i = \frac{\mathrm{TF-IDF_i}}{\max(\mathrm{TF-IDF})} + \frac{\mathrm{Ada\_importance_i}}{\max(\mathrm{Ada\_importance})} + \frac{\sum_{j=1}^{\mathrm{len}(\mathrm{text}_i)} \mathrm{Uncertainty\ word_i}}{\max(\mathrm{Nº Uncertainty\ word})} \quad (2)$$

Per tant, el càlcul, serà el valor de TF-IDF del text i, que surt de fer la mitjana del TF-IDF de totes les paraules incertes, entre el valor màxim de TF-IDF de la mitjana de valors de TF-IDF per cada text. Més, la mitjana del pes atorgat pel model a les paraules, enter el pes màxim. Més, la mitjana del nombre de paraules incertes per cada text, entre valor màxim de paraules incertes de tots els textos. A partir d'aquest càlcul, s'obtindrà el valor de l'índex que posteriorment serà usat en comparació amb el S&P500.

Una altra mesura per avaluar la informació proporcionada pel model és la probabilitat a posteriori. Aquesta mesura indica la probabilitat que el model assigna a un text determinat. Cada categoria té una probabilitat associada, per la qual cosa cada text té una probabilitat a posteriori de no ser incert o de ser-ho. En el model *AdaBoost*, segons la llibreria *Scikit-Learn*, aquesta probabilitat es calcula mitjançant la mitjana ponderada de les probabilitats de classe predites pels classificadors en el conjunt (*Loading. . .*, n.d.-b).

En aquest punt, és on entra en joc, la comparació amb el S&P500, les dades d'aquest han sigut extretes de la pàgina web *Slick Charts*, que conté dades del mercat financer així com informació d'aquests, ofereix índex de marcat, com el S&P500 o el NASDAQ entre d'altres (*S&P 500 Total Returns*, 2024). La font de les dades és de 98 registres, de 1926 a 2024, però, a causa de les necessitats de l'estudi només usarem les dades de 1929 a 2021. Aquestes dades representen el retorn del S&P500, inclouen els retorns generats pels dividends i els retorns generats pels canvis de preu, aquests es troben en forma de percentatge. En els registres del 1929 i del 1933, no s'hi té en compte els primers quatre mesos de l'any, pel fet que la vintena esmena de la constitució es va canviar després del discurs del 1933 i va passar de fer-se el 4 de març al 20 de gener (Klein & Klein, 2023). Per tant, del 1937 al 2021, s'ha extret el primer mes de l'any, ja que, s'entén que l'impacte del discurs es recull en els mesos posteriors. Els valors seran usats per comprovar, si l'indicador creat, mostra el grau d'incertesa, i si aquest té algun tipus d'afecta en l'economia.

3. Marc Teòric

3.1. Discursos Inaugurals

L'oratòria és l'art d'oferir discursos (Alattar 2014:2) i si aquest és junta amb la política, obtenim un seguit d'influències a la vida social de les persones (Medvid et al., 2022:145). Per tant, l'estudi dels discursos dels presidents, en tots els tipus d'estudi que es puguin realitzar sobre aquests, tenen certa naturalesa sociològica (Medvid et al., 2022:145). Per això, la importància d'un discurs d'un president i de com es forma i de les paraules usades per aquest, és clau pel seu impacte, i és com es presenta el discurs és el que el fa poderós (Altgeld's opinion, 1901:8):

> "He must furnish the ideas, he must cloth them in words, he must give these a rhythmic arrangement, and he must deliver them with all the care with which a singer sings a song. Each of these elements is of supreme importance. The ideas must be bright and seem alive. The language must be chaste and expressive. The arrangement must be logical, natural and effective. There must be a natural unfolding of the subject-matter."

Aquest poder és usat per diferents objectius i per diferents motius, per tant, pel seu ús, s'ha de ser coneixedor cap a quina audiència i quin afecte vols que tingui el teu discurs, ja que aquest és una fletxa que té com a objectiu obtenir un afecte específic (Lasswell, H. D. (2006: s.p.). En conseqüència, ens movem en un tipus de comunicació amb un objectiu totalment clar, que és incidir en la vida tant de les persones com de les empreses. S'ha de tenir en compte el contingut que conté un discurs polític, ja que aquest pot ser clau per tal d'analitzar el tipus d'afecte que posteriorment tindrà. Segons Matsko, (Matsko, 2003:s.p.), un discurs polític és: "un discurs preparat sobre qüestions polítiques agudes, que conté avaluacions de diferents modalitats, justificacions, fets, plans i perspectives de transformacions polítiques futures". Per oferir aquests discursos cal una estratègia comunicativa, en aquesta hi hem de trobar mesures concretes per aconseguir l'objectiu del nostre discurs (Medvid et al., 2022:147).

En el cas que ens pertoca, l'anàlisi va enfocat als discursos inaugurals del mandat dels presidents dels Estats Units, com s'ha comentat anteriorment aquests tenen diferents enfocaments, com menciona la pàgina oficial de la casa blanca (White House Historical Association, n.d.-b), alguns presidents dediquen els seus discursos als problemes de la nació com Franklin D. Roosevelt o com John F. Kennedy el 1961, amb un discurs que va tenir com a objectiu "alleugerir el pànic d'una població agafada per la Gran Depressió"(White House Historical Association, n.d.-b). El procés per escollir president als Estat Units, és un procés llarg i amb diferents etapes. Segons el segon article de la Constitució americana el poder executiu recau en el president que és escollit cada 4 anys (*Article II*, n.d.). Segons la vintena esmena de la constitució dels Estats Units, des del 1933, el dia inaugural del mandat, va passar del 4 de març al 20 de gener (Klein & Klein, 2023).

Què és la incertesa? Segons el *Cambridge Dictionary* la incertesa és una situació en la qual alguna cosa no es coneix, o alguna cosa que no es coneix del tot o la sensació de no estar segur del que passarà en el futur. De fet, segons l'economista Frank Knight "*Uncertainty is one of the fundamental facts of life. It is as ineradicable from business decisions as from those in any other field*" (Knight, F. H, 1921:347). Per tant, estem davant un concepte molt rellevant en el nostre dia a dia, de fet és fonamental per entendre molts aspectes de la vida, a més ens permet entendre moltes reaccions tant de polítics, com de la gent, així com dels mercats.

Sovint associem certs aspectes a un possible esdeveniment negatiu, però no sempre té perquè ser així, de fet, en el cas de la incertesa, sabem, i ho veiem cada dia que aquesta no té ni connotació negativa ni positiva, no és ni bona ni dolenta (Dönmez i Grote, 2018:95), el que

la fa bona o dolenta és la reacció posterior que es té. En conseqüència, cal afrontar aquesta des d'un punt de vista neutre. De fet, l'estudi de la incertesa és força comú, ja que aquesta té molts vessants, té el vessant sociològic, el vessant polític, la social i l'econòmica entre d'altres. És normal que es generin diferents teories sobre aquesta, una d'elles és la teoria de la reducció de la incertesa de Charles Berger, en què s'analitzen les maneres en què els individus responen a la incertesa i els resultats associats amb aquesta (Goldsmith, 2001:514). De fet, aquesta teoria ha estat creada per preveure i explicar la freqüència en què diversos comportaments comunicatius poden ocórrer, i la relació entre la freqüència de comportament i els nivells d'incertesa (Goldsmith, 2001:514). Per tant, aquesta engloba molts aspectes de la nostra vida i moltes variants que cal tenir en compte a l'hora de comunicar, per exemple, la naturalesa variada de la incertesa cobreix aspectes com el risc, l'ambigüitat i l'equivocitat (Dönmez i Grote, 2018:95). Ara, ja es té un marc sobre el qual actua la incertesa, però podem mesurar aquests trets de la incertesa, fins a quin punt, podem mesurar aquesta.

Com es deia, es pot mesurar la incertesa, sabem que el risc, per exemple, està associat amb probabilitat, però, en canvi, la incertesa és "un esdeveniment desconegut d'un conjunt desconegut de resultats possibles" (Hillson, 2003:27). El mateix Hillson mencionar que el "Risk et permet mesurar la incertesa". És a dir tenim, el risc mesurable seria el mateix risc, que per exemple pugui tenir una acció de borsa, però, en canvi, quantificar la incertesa genera més complicacions, és per això, que per valorar numèricament la incertesa es necessita el risc. És a dir, arribem a un punt on tenim, el risc mesurable a través de la probabilitat i la incertesa que ens explica o deixar d'explicar informació incompleta i no mesurable (Dönmez i Grote, 2018:96), però que si ens ajudem del risc, la podem quantificar. Knight, per altra banda, remarca que "la incertesa és la raó subjacent per al benefici empresarial" (Knight, F. H, 1921:347).

En trobem diversos exemples dels efectes de la incertesa en la política, segons la firm a Standard & Poor, senyala i culpa, entre altres raons principals, la incertesa política, de la davallada del deute del Tresor dels Estats Units l'agost del 2011, amb Obama de president[4]. De fet, com diu Mark Zandi:

> "I'm increasingly of the view that the reason why our economy can't get into a higher gear is because of the uncertainty created by Washington, that this brinkmanship which happens every 3, to 6, to 12 months is corrosive on our collective psyche and it's weighing on our willingness and ability to take risk.[5]"

Per tant, tenim un impacte coherent de la incertesa en l'economia, afectant els resultats de l'economia, com s'ha comentat anteriorment. Però no tota mena d'incertesa és dolenta, el que la fa dolenta és com es reacciona, per exemple com diuen Pástori i Veronesi:

> "We generally do not insist on knowing in advance how exactly a doctor will perform a complex surgery; should unforeseeable circumstances arise, it is useful for a qualified surgeon to have the freedom to depart from the initial plan (Pástor i Veronesi, 2013:521)."

Per tant, la gestió de la incertesa ens determina les conseqüències futures i els resultats ens

[4] Com diu David Beers, ex-director de l'empresa financera Standard & Poor's, en una conferencia amb peridoistes el 6 d'agost de 2011, afageix. "debate this year has highlighted a degree of uncertainty over the political policymaking process which we think is incompatible with the AAA rating.

[5] Chief Economist de Moody's Analytics en una trobada amb periodistes a Washington. Political Uncertainty Will Continue To Stunt Economic Growth, Janet Novack, October 16, 2013

evidenciaran si aquesta ha sigut manipulada de la forma idònia o no. Per això és necessari cert grau de consciència a l'hora d'afrontar tal problemàtica, i ser conscients de la importància de la reacció i no tant de la certesa que la mateixa incertesa generarà en un futur, valgui la redundància.

Tot seguit, cal tenir present l'efecte de la incertesa en els mercats, hem vist que la incertesa pot ser bona o dolenta, que el que varia és la manera d'afrontar aquesta. En coneixem els afectes i les seves causes, de fet, aquests ja estan demostrats en l'economia, però se'ns genera el dubte de com els mercats, l'economia, reaccionen a la incertesa política. Estudis com el de Pastor i Veronesi, 2013, clarifiquen aquest camí i ens ensenyen com aquesta inseguretat provoca una pujada de la volatilitat en la borsa (Pastor i Veronesi, 2013:522). De fet, s'escau la possibilitat que la incertesa modifiqui la percepció d'inversió així com la forma en què s'afronta tal predicció en un entorn incert. A més, en moments d'incertesa ens endinsem en un pla d'ineficiència dels tipus d'inversions, és per això que l'elevada incertesa provoca en l'inversor la propensió de realitzar inversions divisibles irreversibles, com d'altra forma ho fa amb les indivisibles (Bernanke, 1983:12).

3.2. Natural Language Processing (NLP)

Segons el diccionari d'oxford, el processament són una sèrie d'accions que un ordinador realitza en les dades per produir una sortida. El processament és aplicat en tots els tipus d'anàlisis relacionades amb aprenentatge automàtic. Concretament, en el cas del nostre estudi la tècnica que s'usarà per al processament és el NLP, aquest segons Jin i Mihalcea, 2023, és l'estudi de mètodes computacionals per automàticament analitzar text i extreure informació rellevant per la posterior anàlisi. Quan diem *Natural Language*, ens referim a la comunicació diària que usem els humans, la comunicació amb anglès, català o portuguès, entre d'altres. Per tant, llenguatge totalment diferenciat del llenguatge matemàtic o de programació (Bird et al., 2009:9).

Els models NLP estan basats en la sintaxi del text i sovint aquests, com a conjunt, no tenen la capacitat de detectar la semàntica, per això els textos són processats com paraules individual, *tokens* (Avela i Lehmus, 2020:4). Així doncs, veiem que actualment ja tenim les eines per analitzar text i que a causa de la subjectivitat, que aquests, molt sovint poden aportar, se'ns presenta, un repte a l'hora de processar textos, ja que les decisions que prenguem en aquest camí, podem ser claus a l'hora, de posteriorment obtenir resultats.

Els NLP estan patint un creixement molt gran per això és important tenir coneixement d'aquests (Bird et al., 2009:10). De fet, els NLP, tenen la capacitat de comprendre grans quantitats de text, i per exemple, poden descobrir quina és la temàtica principal del text (Calvo-González et at., 2022:2). Per tant, s'han d'entendre els NLP com qualsevol manipulació computacional de llenguatge natural (Bird et al., 2009:9). De fet, es remarca el progrés dels NLP, en tasques com la comprensió lectora, respondre preguntes i implicació textual, entre d'altres (Brown et al. 2020:1). Podem trobar diferents casos d'èxit i d'utilitat pel que fa a l'ús d'aquesta eina, com és el cas de l'estudi Brown et al el 2009. Altres estudis relacionats amb la nostra anàlisi com (Avela i Lehmus, 2010)[6]. Estudi d'ús fonamental per l'estudi que ens ocupa,

[6] Estudi on es desenvolupa un índex d'incertesa a través dels titulars de les notícies finlandeses.

o (Calvo-González et at., 2022:2)[7] , o (Jin i Mihalcea, 2023)[8]. Els passos a realitzar pel processament són força similars en la majoria d'estudis, aquests inclouen signes de puntuació, paraules en minúscula, *tokenization*, exclusió de paraules irrellevants[9] i *lemmatization/stemming*.

Pel que fa als signes de puntuació, com mostra l'article de Tabassum i Patil, el 2020, es diu que: treure els signes de puntuació és clau, a causa que el codi no entén aquests símbols i només aporten soroll no desitjat (Tabassum i Patil, 2020:4865). En el mateix sentit l'estudi, es menciona el fet que una variació en la majúscula d'una paraula, com "Índia" i "índia" portar a resultats diferents (Tabassum i Patil, 2020:4865). En el cas de la *tokenization*[10] l'utilitzem per a la separació de la puntuació de les paraules (Mihalcea et al., 2006:289). Per altra banda, s'ha d'extreure tota aquella informació que no guarda cap informació semàntica (Avela i Lehmus, 2020:9), és a dir, les *stopwords*.[11]

En el cas de la *Lemmatization* segons el Cambridge Dictionary és, el procés de reduir les diferents formes d'una paraula a una sola forma[12]. Per contra, s*temming* segons el Cambridge Dictionary, és una part central d'alguna cosa de la qual altres parts poden desenvolupar-se o créixer, o que forma un suport.[13] La transformació del vocabulari presenta un repte en si mateix, en la cerca d'unificar criteris, i de trobar la màxima semblança entre paraules similars, perquè, per exemple, *building* i *built*, per l'ull humà és una paraula molt similar, però per l'ull computacional, no ho és, per això el desafiament es troba en reduir el màxim possible les paraules, per aconseguir la màxima semblança entre paraules similars, però sense perdre la informació del significat d'aquesta. Típicament, l'ús del *stemming* està més normalitzat, a causa de dos motius. El primer és que trobar l'arrel de la paraula -el *stem*- requereix menys esforç computacional. El segon motiu, és que el *stem*, reté més informació de la paraula (Avela i Lehmus, 2020:9). Per altra banda, aquesta tècnica ens permet reduir la dimensió del diccionari del document (Balakrishnan i Lloyd-Yemoh, 2014:175). Cal, però, saber que la *lemmatization* et permet quedar-te amb una paraula identificable en un diccionari (Bird et al., 2009:10).

Avui dia coneixem diferents tècniques per realitzar el *stemming*, entre elles hi trobem el *P aice/Husk stemmer*, *Porter's stemmer* i *Lovin's stemmer* (Balakrishnan i Lloyd- Yemoh, 2014: 175). Porter stemmer és una tècnica altament usada en el món dels NLP i de la recuperació d'informació (Balakrishnan i Lloyd- Yemoh, 2014:175), aquest té força avantatges entre elles,

---

[7] Estudi a través de NLP per veure les dinàmiques dels presidents de Sud Amèrica pel que fa a les polítiques prioritàries.

[8] Ús d'NLP per fer polítiques.

[9] Stop-words, paraules d'alta freqüència com the, to i also (Bird et al., 2009:60).

[10] Cambridge dictionary, 2024 (Cambridge English Dictionary: Meanings & Definitions, 2024) s.f. *the process of dividing a series of characters (= letters, numbers, or other marks or signs used in writing or printing) into separate tokens (= units) that have a meaning*.

[11] S'entén per *stopwords* en l'àmbit lingüístic article, conjuncions i pronoms sense significant semàntic

[12] Per exemple: Reduint "build", "building", or "built" a la forma "build".

[13] L'arrel de la paraula "hiring" és "hir".

la seva simplicitat i eficiència (Wiese et al., 2011:496). La tècnica de *stemmer* primerament cr eada el 1980, va ser millorada, pel que es coneix com *second Porte*r o *snowball stemmer* (Wi ese et al., 2011:496). Hi ha diverses diferències entre ambdós algoritmes, la diferència principal és que el *Snowball stemmer* és més agressiu que el *Porter stemmer* (GeeksforGeeks, 2022b).

3.2.1. Term frequency/inverse document frequency (TF-IDF)

Fins ara se sap com s'ha realitzat el processament de text, i fins ara només hem treballat amb lletres i paraules, però necessitem d'alguna manera extreure informació a nivell numèric. Per poder aplicar els algoritmes d'aprenentatge automàtic, és necessari aplicar mesures que ens permetin transformar les dades textuals a numèrica i realitzar el comentat anteriorment, aquestes tècniques són *Document Frequency*, *Information Gain*, *mutual information*, *chi-square statist*ic i *term stregnth* (Yang & Pedersen, 1997:s.p.), entre d'altres. Tot i això, l'ús de *Term frequency/inverse document frequency* -TF-IDF- és el mètode més utilitzat. (Yun-tao et al., 2005:49).

Per entendre el TF-IDF, assumim que tenim un vector amb totes les paraules de cada text que representa un document d, per tant, d_ij és el token i del document j (Avela i Lehmus, 2020:10):

$$\mathrm{d}_{(\mathrm{i,j})} = (\mathrm{t}_1, \mathrm{t}_2, \dots, \mathrm{t}_\mathrm{n}) \tag{3}$$

Per *term-frequency* fa referència al nombre de cops el *token* i apareix en el document d (Yun-tao et al., 2005:50). Per tant, tindrem per cada document un vector amb la freqüència de cada paraula. Però ara ens falta la segona part, la referent a *Inverse document frequency* (IDF), que com el seu nom indica, és el valor del *token* inversament proporcional a la freqüència que apareix en el document (Yun-tao et al., 2005:50). La seva fórmula és.

$$\mathrm{IDF}\left(\mathrm{d}_{\mathrm{i,j}}\right) = \log * \frac{\mathrm{N_d}}{1 + |\{\mathrm{d}: \mathrm{t} \in \mathrm{d}\}|} \tag{4}$$

On N_d és el nombre de documents (Avela i Lehmus, 2020:11) i $|\{d: t \in d\}|$ és el nombre de documents on el terme t apareix, quan se satisfà que $tf(t,d) \neq 0$ (Chaudhary, 2021). Per aconseguir aquesta transformació, de lletres a números, només queda el següent:

$$\mathrm{TF} * \mathrm{IDF} \tag{5}$$

Al final s'obté un valor que ens permet identificar els termes importants del nostre text i poder diferenciar-los dels termes més comuns del text[14] (Parmar i Tiwari, 2024:167).

[14] Sense contar les stop-words que ja han sigut extretes abans de fer el TF-IDF.

En resum, s'obté un seguit de valors que ens permet continuar treballant amb els nostres textos, però a escala numèrica, és precís l'ús d'aquesta eina, a causa del fet que permet identificar les característiques més rellevants per a la posterior classificació, així com permet identificar aquelles característiques menys rellevants per la nostra anàlisi.

3.3. Aprenentatge automàtic

Hi ha diferents objectius a l'hora de realitzar la classificació de text, aquest inclou la classificació basada en temes i classificació basada en el gènere (Ikonomakis et al, 2005:s.p.). Cada una de les possibles classificacions conte els seus avantatges i inconvenients. Per tal de poder escollir, el millor mètode de classificació cal entendre les capacitats de tots els algoritmes i entendre l'efectivitat d'aquests (Kowsari et al., 2019:2).

L'aparició de les tècniques de *machine learning* va ser considerada, pel món de l'anàlisi de text, com una revolució (Kheiri i Karimi, 2023:s.p.), en conseqüència som davant d'un procés, clau i d'un avanç important en aquest camp. Per entendre els tipus d'aprenentatge automàtic cal classificar els algoritmes de *machine learning* entre supervisats i no supervisats. La diferència més clara entre aquest dos tipus d'aprenentatge és que els supervisats requereixen unes dades d'entrenament etiquetades i l'aprenentatge no supervisat requereix d'unes dades sense etiquetar, per trobar patrons dins del conjunt de dades (GeeksforGeeks, 2024c).

Algoritmes d'aprenentatge automàtic serien *K- means*, Naïve Bayes, *Suport Vector Machine, Random Forest* (Kheiri i Karimi, 2023:s.p.) o *Decision Trees*. De fet, per la metodologia de classificació de text és comunament aplicat Naïve Bayes, degut a dos principis que poden ser claus a l'hora de triar quin model utilitzar, aquests són que la seva implementació és ràpida i fàcil d'utilitzar (Rennie, et al, 2003:s.p.). Aquest model neix a partir, i com el seu nom indica de la teoria de Bayes (Kowsari et al., 2019:25). És necessari comentar que aquest model, tot i la seva popularitat, presenta certs errors, com la mala gestió de les dades esbiaixades, o els errors amb les magnituds dels pesos. (Rennie et al, 2003:s.p.). Tot i això, altres variants d'aquest han sorgit per intentar mitigar els erros i realitzar un model un punt més complex. Per exemple trobem el model *Multinomial Naive Bayes*, que incorpora una característica important amb relació a la freqüència de la informació (Rennie et al, 2003:s.p.). És l'estudi de Rennie, que incorpora i modifica el model *Multinomial Naive Bayes* -MNB-, per arribar a un model anomenat *Transformed Weight-normalized Complement Naive Bayes* -TWCNB- (Rennie, et al, 2003:s.p.). Aquest model corregeix els errors comentats anteriorment, els biaixos i les debilitats del model causades per dades no balancejades i les dependències entre paraules, que el model *Naive Bayes* original no té en consideració (Avela i Lehmus, 2020:10). Un altre model, comparable al *Naive Bayes*, seria el *Suport Vector Machin*e (SVM), tot i que aquest és més lent i més difícil d'implementar (Rennie, et al, 2003:s.p.). Primerament, va ser dissenyat per classificacions de classes binàries (Kowsari et al., 2019:28). Algunes de les millores que aporta el SVM, és el de la funció *kernel*, que és capaç de realitzar un mapa d'un espai de punts de dades que no és separable linealment en un altre on sí que ho es (Mullen i Collier, 2004:s.p.). Aquest presenta millors resultats que el MNB, i una petita millora que el TWCNB com mostra l'estudi de Rennie, 2003.

3.3.1. Arbres de decisió

En el cas que ens pertoca trobem l'ús dels arbres de decisió o *decision trees* en anglès, aquest segueixen la mateixa estructura que podríem veure en un arbre qualsevol (Patel i Prajapati: 2018:74). Aquest classificador forma part d'un aproximament conegut com a *multistage decision making* (Safavian i Landgrebe, 1991:660). De fet, com diu l'estudi sobre la metodologia dels classificadors *Decision Trees* la idea bàsica d'aquest i del *multistage approach* és:

> "break up a complex decision into a union of several simpler decisions, hoping the final solution obtained this way would resemble the intended desired solution" (Safavian i Landgrebe, 1991:660)"

Per construir un arbre de decisió cal saber primer quina és la idea principal d'aquest, aquesta és crear un arbre a partir dels atributs per categoritzar les nostres dades (Kowsari et al., 2018:32). Com qualsevol model d'aprenentatge automàtic, els objectius dels arbres de decisió són 4, classificar correctament el conjunt de dades d'entrenament, ser capaç de generalitzar perquè les dades de prova aconsegueixin els millors resultats possibles, que sigui fàcil d'actualitzar i que tingui una estructura senzilla (Safavian i Landgrebe, 1991:661). A més a més, és necessari saber, l'estructura de l'arbre i com ha de ser construït, en relació amb aquest tema, l'estudi de Kurzyński, el 1983, aquest ens parla de tres conceptes que ha de tenir l'arbre, i que es veuran reflectits en terme d'eficiència i qualitat del model. Els tres components són (Kurzyński, 1983:819).

1. "the choice of a tree structure or hierarchical ordering of the pattern classes",
2. "the choice of features to be used at each nonterminal node[15]",
3. "the decision rules for performing the classification at each nonterminal node".

En el nostre cas ens interessa l'aproximament d'arbres de decisió realitzada per l'algoritme *classification and regression trees* (CART), que va ser introduït el 1984 per Breiman, aquest menciona en un article posterior que el mètode CART troba la seqüència de sub-arbres podats de T[16] de cost mínim. El millor sub-arbre podat, en aquesta seqüència se selecciona o bé realitzant *cross-validation*[17] o a través d'un conjunt de proves (Breiman, 1996:131). Aquest mètode usa l'índex de Gini com a mesura de divisió de l'arbre (Priyam et al., 2013:335), en coneixem altres mètodes usats per l'arbre de decisió com els algoritmes: *Iterative Dichotomaiser 3 -ID3-, scalable parallelizable induction of decision tree algorithm -SPRINT-* i *supervised learning in ques* -SLIQ- (Priyam et al., 2013:335),. En aquest estudi ens centrarem en el mètode CART. Per explicar el mètode usarem l'article de Rutkowski et al., 2014, on es proposa un nou algoritme a partir del mètode CART.

[15] Segon Oxford Rearch, un node no és terminal, i per tant, hi ha més branques sortint d'aquest node (Oxford Reference, n.d.)

[16] Sigui T un arbre de decisió.

[17] Tècnica usada en aprenentatge automàtic per avaluar l'actuació del model en dades no vistes.

Procés inicial:

- Anomenem un node inicial L_0,
  - Aquest conté un subconjunt de les dades d'entrenament S
- Per cada node L_q creat, es processa un subconjunt particular S_q del conjunt de dades d'entrenament S.
- Si tots els elements del conjunt S_q pertany a la mateixa classe, el node s'etiqueta com a terminal.
- En cas contrari es divideix el node:
  - Per cada atribut disponible a^i[18] , s'avalua segons una mesura de divisió -índex de gini-.
- Per cada atribut disponible del conjunt de valors de l'atribut A^i[19], es divideixen en dos subconjunts disjunts, $A_L^i \; i \; A_R^i$.
  - El conjunt de totes les particions possibles del conjunt A^i es denota com V_i.
- Els subconjunts $A_L^i \; i \; A_R^i$, divideixen els conjunts de dades S_q, en dos subconjunts disjunts.
  - $L_q(A_L^i)$ subconjunt de dades que correspon a A_L^i:

$$\mathrm{L}_q(A_L^i) = \{\mathrm{s_j} \in \mathrm{S_q} \mid \mathrm{v_{ij}} \in A_L^i\} \tag{6}$$

  - $R_q(A_L^i)$ subconjunt de dades que correspon a A_R^i.

$$\mathrm{R_q}(A_L^i) = \{\mathrm{s_j} \in \mathrm{S_q} \mid v_{\mathrm{ij}} \in A_R^i\} \tag{7}$$

- Ara definim el node esquerre,

$$\mathrm{P_{L,q}}(A_L^i) = \frac{|\mathrm{L_q}(A_L^i)|}{|\mathrm{S_q}|} \tag{8}$$

  És a dir nombre d'elements en $L_q(A_L^i)$ entre nombre total d'elements S_q. El mateix pel node dret R.

- Com les dues fraccions són dependents l'una de l'altre: $\mathrm{p_{R,q}}(\boldsymbol{A_L^i}) = 1 - \mathrm{p_{L,1}}(\boldsymbol{A_L^i})$. Per tant, només necessitem el valor d'un dels paràmetres.

[18] Per exemple si parlem de cotxes, seria el color, el model...

[19]Per exemple, si parlem de model, seria "Audi", "Toyota"...

$$\mathrm{p_{R,q}}(A_L^i) = 1 - \mathrm{p_{L,1}}(A_L^i) \tag{9}$$

- Si ara ho fem per K classes, obtenim, per cada $p_{kL,q}\boldsymbol{A_L^i}$, el nombre d'elements de la classe K en $L_q(\boldsymbol{A_L^i})$ entre nombre total d'elements S_q.

- El càlcul de la fracció S_q, és el nombre d'elements de la casse K en S_q entre nombre total d'elements en S_q.

- La mesura d'impuritat és l'índex de Gini i per qualsevol S_q calculem:

$$\mathrm{Gini(S_q)} = 1 - \sum_{k=1}^{K} (p_{k,q})^2 \tag{10}$$

- L'índex de Gini és igual a 0 si totes les categories cauen en una sola, i el màxim sobte quan tots els casos són distribuïts de manera igualitària.

- Ara hem de calcular la impuresa ponderada de la següent manera:

$$\mathrm{wGini(S_q, A_{iL})} = p_{L,q}(A_{iL})\mathrm{Gini}\left(\mathrm{L_q(A_{iL})}\right) + (1 - p_{L,q}(A_{iL}))\mathrm{Gini}\left(\mathrm{R_q(A_{iL})}\right) \tag{11}$$

- Ara calculem l'índex de Gini pels dos nodes:

$$\mathrm{Gini}\left(\mathrm{L_q}\left(\mathrm{A_i^L}\right)\right) = 1 - \sum_{\mathrm{k=1}}^{\mathrm{K}} (\mathrm{p_{kL,q}}\left(\mathrm{A_i^L}\right))^2 \tag{12}$$

$$\mathrm{Gini}\left(\mathrm{R_q}\left(\mathrm{A_i^R}\right)\right) = 1 - \sum_{\mathrm{k=1}}^{\mathrm{K}} (\mathrm{p_{kR,q}}\left(\mathrm{A_i^R}\right))^2 \tag{13}$$

- Arribats aquest punt, la millor divisió de l'arbre s'escull maximitzant el Gini *Gain* o minimitzant el valor de la impuresa de Gini (Jairi, 2023).

- Per calcular el Gini *Gain*, s'efectua la següent operació:

$$\mathrm{g(S_q, A_L^i)} = \mathrm{Gini(S_q)} - \mathrm{wGini(S_q, A_L^i)} \tag{14}$$

  - D'entre totes les particions A_L^i del conjunt A^i, obtenim el valor màxim del guany Gini, que es fa de la següent manera:

$$\widehat{\mathrm{A}}_{\mathrm{L,q}}^{\mathrm{i}} = \arg\max_{\mathrm{A_i^L} \in \mathrm{V_i}} \mathrm{g(S_q, A_i^L)} \tag{15}$$

- La partició $\tilde{A}_{L,q}^i$ la partició òptima del conjunt A^i pel subconjunt S_q de les dades d'entrenament.

- Ara ens quedem amb els subconjunts

$$L_q^i \equiv L_q^i(\hat{A}_{L,q}^i) \quad (16)$$

$$R_q^i \equiv R_q^i(\hat{A}_{R,q}^i) \quad (17)$$

3.3.2. Adaboost classifier

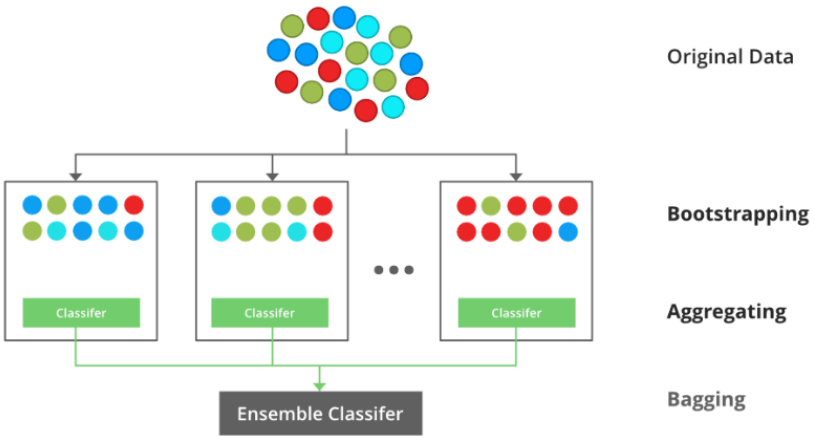
Respecte a l'elecció del model, cal anar un punt més enllà que els algoritmes d'aprenentatge automàtic, i per això ens hem de plantejar: i si poguéssim, multiplicar l'execució d'algun dels models mencionats anteriorment, i si poguéssim fer els suficients models perquè en el conjunt final d'aquest poguéssim obtenir un model més precís. Doncs això és el que realitzant els *ensamble methods*. E*nsemble* segons el *Cambridge Dictionary* és un grup de coses o persones que actuen o es reuneixen en conjunt. Per tant, els *ensemble methods* o *ensemble learnings*, combinen o reuneixen en un conjunt de múltiples algoritmes per millorar el resultat final del model. Aquests models es divideixen en dos tipus el *boosting* i *bagging*, els també coneguts *Voting Classification Techniques*. Els dos han estat provats com a casos d'èxits, com mostra el cas de Farzi i Bolandi el 2016. Però tots dos tenen diferències.

Pel que fa a la tipologia *bagging*, va ser proposat el 1994 per Breiman, on a través de rèpliques *bootstraps* del conjunt d'entrenament, i usant un conjunt de dades de proves en mètodes de classificació com arbres de regressió, va descobrir que el *bagging* pot donar millores substancials en la precisió -*accuracy*- (Breiman, 1996:123). Segons Breiman,

> "It is a relatively easy way to improve an existing method, since all that needs adding is a loop in front that selects the bootstrap sample and sends it to the procedure and a back end that does the aggregation".

La raó per l'ús de *bootstrap* és per millorar l'estabilitat del model i la precisió, com comentava Breiman, dels models de *Machine learing*, utilitzats per classificació i tècniques de regressió (Farzi i Bolandi, 2016:5).

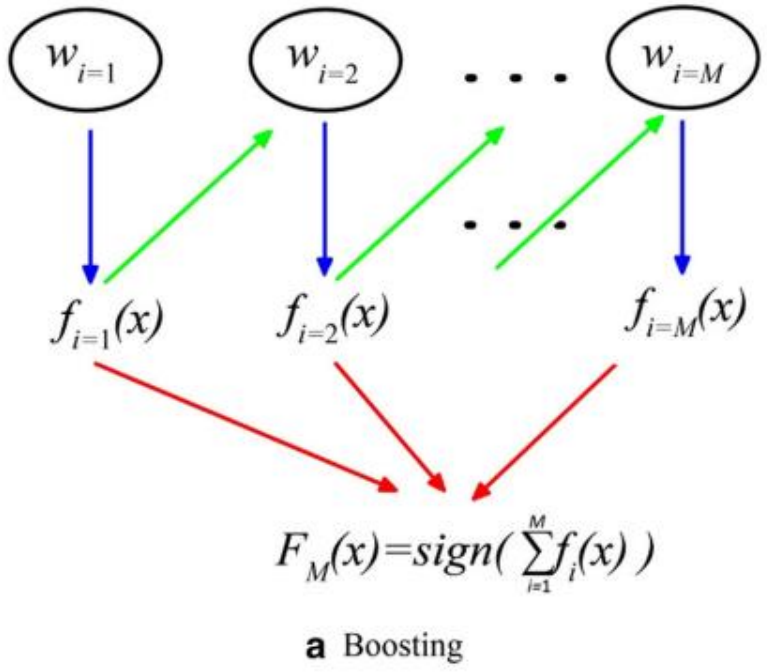
Figura 3.1. Bagging esqueme (Farzi i Bolandi, 2016:4)



En el cas que ens interessa, és el *boosting*, aquest va ser introduït amb l'objectiu de convertir un *weak lerner*[20] en un algoritme que aconsegueix una precisió -*accuracy*- alta (Schapire, 1990:197). Aquesta idea va ser primerament introduïda el 1990 per Schapire. La idea inicial d e Schapire va ser millorada per Freund el 1992 i el 2004 per Bloehdorn i Hotho. Aquesta apli cació va comptar amb l'ús de TF- IDF per posteriorment entrenar un model AdaBoost, del qu al es detallarà, pròximament, el seu ús en a el treball.

El funcionament d'aquest mètode és diferent del vist anteriorment, de manera esquemàtica, aquest funciona de la següent manera:

Figura 3.2. Boosting esqueme (Farzi i Bolandi, 2016:4)



a Boosting

Com es mostra en la figura 3.2, en cada etapa del procés *boosting* es generen els resultats del classificador (Freund, 1992:393), en la primera iteració w_1, el model s'entrena "sense cap informació prèvia", i la iteració segueix el procés habitual de qualsevol classificador. És partir d'aquí, on cada $f_i(x)$ es entrenat amb la a informació dels pesos w_i -fletxa blava- que depenen de l'actuació o de la realització dels classificadors previs $f_{i-1}(x)$ -fletxa verda- (Farzi i Bolandi, 2016:3), aquests pesos determinen la importància de cada model d'aprenentatge dèbil. Un cop tenim entrenats, els *weak learners* -en el cas de l'estudi, els arbres de decisió-, cal posar-los en comú per obtenir un únic resultat -com mostra la figura 3.2 amb les fletxes vermelles-. Cal per tant, tenir un conjunt de dades d'entrenament, que denotarem com $\{x_i, y_i\}_{i=1}^{N}$, sent x_i, cada característica del vector, per exemple el valor resultant de TF-IDF, i y_i la classificació de les múltiples classes segons l'estudi (Farzi i Bolandi, 2016:3).

[20] El model pot produir una hipòtesi que només funciona lleugerament millor que l'endevinar a l'atzar (Schapire, 1990:197)

$$F_M(x) = \text{sign}(\sum_{i=1}^{M} f_i(x)) \quad (18)$$

El que fa aquesta funció, que també apareix en al figura 3.2, és la suma de tots els classificadors $f_i(x)$, i posteriorment aplica la funció signa, que retornarà -1 si la suma és negativa i +1 si és positiva (*Sign Function (Signum): Definition, Examples*, n.d.). Aquesta funció és funció que ens permetrà realitzar prediccions en nous conjunts de dades.

Coneixem diversos models que apliquen l'algoritme de boosting, en coneixem, per exemple, el *Gradient Boosting*, *XG- Boost*, el *CatBoost*, (Atiq,2022:12), LogitBoost, LS Boos t i AdaBo*ost* (Farzi i Bolandi, 2016:4), entre d'altres. Hi ha models de *boosting* que són especialitzats per classificació binària, d'altres per classificació de múltiples classes, i d'altres per ambdós casos. Per exemple la variant del *boosting LogitBoost*, usa els passos de quasi Newton[21] per ajustar un model logístic additiu simètric a través de la màxima versemblança (Friedman et al., 2000:356). Un altre exemple d'algoritme *boosting* és, el *XGBoost*, que consisteix a combinar diferents *weak learners*, que en aquest cas són arbres de decisió amb baixa precisió, per acabar tenint un model amb més precisió (Chen i Guestrin, 2016:786). Però en l'estudi que ens pertoca, cal conèixer l'ús de l'algoritme de *boosting* AdaBoost, de l'anglès *Adaptative Boosting*, aquest és considerat un dels algoritmes de classificació més precisos (Sharma i Singh, 2022:7). Va ser introduït per Freund i Schapire el 1996. És en el 1996 on es presenten dos tipus de AdaBoost, anomenats com M1 i M2, que només difereixen per la seva gestió (Freund i Schapire, 1996:2). La diferència entre amb dos models és que el M1, actualitza els pesos basats en l'error de classificació, per tant, cada classificador ha de ser millor que endevina aleatòriament, com a resultat, no és capaç de manejar hipòtesis dèbils quan l'error és major a ½ (Freund i Schapire, 1996:4). Per contra, el M2, utilitza la funció de "pseudopèrdua" per ajustar els pesos en la confiança del classificador, per tant, aquesta funció només ha de ser una mica millor que endevinar aleatòriament (Freund i Schapire, 1996:5). Com s'indica en l'estudi Freud i Schapire el mètode M2 presenta millors resultats que el M1.

Per entendre, una mica millor la funció de *pseudo-loss* i la seva utilitat, cal saber, que el gran avantatge d'aquesta funció és que penalitza les assumpcions dèbils sota les següents situacions: L'etiqueta predita racionalment no té cap etiqueta correcta, això vol dir, que si les etiquetes predites no inclouen l'etiqueta correcta la funció de "pseudopèrdua", penalitzarà el classificador. (Chengsheng et al., 2018:1793). El segon punt parla del contrari, de si cada conjunt d'etiquetes predites és incorrecta la funció de "pseudopèrdua", penalitzarà el classificador (Chengsheng et al., 2018:1793).

La forma de funcionar dels models AdaBoost és la següent:

- Abans de començar hem de tenir una seqüència de m exemples, on cada un conte la variable explicativa $x_1, \ldots, x_m$ i la variable explicada $y_1, \ldots, y_m$. Cada una forma una parella x_i i y_i.

[21] Aquest mètode s'utilitza quan el mètode de Newton és difícil d'utilitzar o quan calcular l'Hessià és massa car per iteració. Aquest mètode aproxima l'Hessià utilitzant només la informació del gradient (Shi et al., 2022).

- Es defineix $B = \{(i, y): i \in \{1, \dots, m\}, y \neq y_i\}$ com el conjunt de totes les parelles (i, y) on i fa referència a l'índex del conjunt de dades d'entrenament i y és una etiqueta o classes associada amb l'exemple i.

- Ara definim una distribució de no classificació, a partir de tots els conjunts de B que no s'han classificat correctament. Aquesta distribució es defineix de la següent manera:

$$\mathrm{D}_1(\mathrm{i}, \mathrm{y}) = 1/|\mathrm{B}| \text{ for } (\mathrm{i}, \mathrm{y}) \in \mathrm{B} \tag{19}$$

- Ara iterem per tot $t = 1, 2, \dots, T$, sent T un nombre específic d'iteracions

    - A cada iteració obtenim una classificació dèbil -*weak learner*- i, per tant, una distribució D_t.

    - En cada ronda, es planteja la mateixa hipòtesi. $h_t: X\ x\ Y \rightarrow [0,1]$. On ens preguntem sobre quina és l'etiqueta Y segons X. O més clarament, ens preguntem si x_i és l'etiqueta correcta y_i, o el conjunt d'etiquetes incorrectes y. Per tant, tenim un pes on queda representat la penalització per no classificar correctament les etiquetes, aquest pes es representa com $D_t(i, y)$. En conseqüència, els valors de h_t són probabilitats de ser correctament identificats si $h_t(x_i, y_i) = 1$ i $h_t(x_i, y) = 0$, incorrectament si $h_t(x_i, y_i) = 0$ i $h_t(x_i, y) = 1$ i si són iguals s'interpreta com un error aleatori.

    - Seguidament, construïm la funció de "pseudopèrdua" respecte a la hipòtesi h_t. Usant la distribució D_t.

$$\epsilon = \frac{1}{2}\sum_{(\mathrm{i},\mathrm{y})\in \mathrm{B}} \mathrm{D}_\mathrm{t}(\mathrm{i}, \mathrm{y})(1 - \mathrm{h}_\mathrm{t}(\mathrm{x}_\mathrm{i}, \mathrm{y}_\mathrm{i}) + \mathrm{h}_\mathrm{t}(\mathrm{x}_\mathrm{i}, \mathrm{y})) \tag{20}$$

    - A continuació calculem, que ens ajuda a ajustar els pesos per la següent iteració.

$$\beta_\mathrm{t} = \frac{\epsilon_\mathrm{t}}{(1 - \epsilon_\mathrm{t})} \tag{21}$$

    - Ara cal actualitzar,

$$\mathrm{D}_\mathrm{t}: \mathrm{D}_{\mathrm{t}+1}(\mathrm{i}, \mathrm{y}) = \frac{\mathrm{D}_\mathrm{t}(\mathrm{i}, \mathrm{y})}{\mathrm{Z}_\mathrm{t}} * \beta^{(\frac{1}{2})(1+\mathrm{h}_\mathrm{t}(\mathrm{x}_\mathrm{i}, \mathrm{y}_\mathrm{i}) - \mathrm{h}_\mathrm{t}(\mathrm{x}_\mathrm{i}, \mathrm{y}))} \tag{22}$$

      on Z_t és una constant de normalització.

- Finalment, obtenim la hipòtesis final, per x, l'etiqueta y.

$$\mathrm{h}_{\mathrm{fin}}(\mathrm{x}) = \mathrm{argmax}_{\mathrm{y}\in\mathrm{Y}} \sum_{\mathrm{t}=1}^{\mathrm{T}} \left(\log \frac{1}{\beta_\mathrm{t}}\right) \mathrm{h}_\mathrm{t}(\mathrm{x}_\mathrm{i}, \mathrm{y}) \tag{23}$$

Aquesta explicació és la que donen Freund i Schapire, 1996:5, en el seu primer plantejament del model *AdaBoost*. Cal comentar que el classificador dèbil, té com a objectiu trobar una hipòtesi dèbil h_t amb una petita funció de "pseudopèrdua".

3.4. Mètriques i mètodes d'avaluació

Un cop s'ha pogut aplicar el model de la manera corresponen és necessari l'aplicació de mètriques i tècniques d'avaluació. Aquestes mètriques ens permeten determinar la qualitat de l'algoritme usat, calculant diferents mesures de rendiment (Handelman et al., 2019:40).

Les mètriques d'avaluació són totes aquelles eines numèriques que usem per a la valoració del model aplicat: aquests instruments són la *accuracy*, *sensitivty*, *specificty*, *precision*, *F1-score* i AUC (Rainio et al., 2024:7). Cadascuna d'aquestes mètriques ens dona un valor amb un significat diferent, però totes avaluen el rendiment dels models d'aprenentatge automàtic (Santafe et al., 2015: 468). Cal tenir en compte, les diferències en les mètriques que s'usen per classificació binària i per a classificació de multi classe (Rainio et al., 2024:2,3). En aquest estudi, usem una classificació binària. Però per començar a entendre aquestes tècniques cal comprendre 4 conceptes claus en l'avaluació de les mètriques del model, aquests aspectes són els valors verdaderament positius o *true-positive*s -TP-, verdaderament negatius o *true-negatives* -TF-, falç positius o *false-positive* -FP- i falç negatius o *false-negatives* -FN- (Handelman et al., 2019:41). Positiu i negatiu, és només una forma d'expressar els resultats, la classe positiva és la classe objectiu de l'estudi, per exemple si es vol determinar si el cel és blau, la classe positiva és blau i la negativa és no és blau. Però tornant a cada designació de les categories: els valors TP són els valors positius correctament predits, els TF, són els valors negatius correctament predits (Handelman et al., 2019:41). Els FP és un resultat realment negatiu, però ha sigut predita com a positiu i FN és una instància realment positiva que ha sigut predita com a negativa (Zhu et al., 2010:2). En conseqüència, es genera una matriu amb les etiquetes comentades anteriorment tal que:

Quadre 3.1. Mètriques d'avaluació (Handelman et al., 2019:40)

Predicció \Realitat	**Realitat**	
	Positius	**Negatius**
Positius	TP	FP
Negatius	FN	TN

A partir d'aquest punt, es generen diferents mètriques d'avaluació, comentades anteriorment, primerament es comentaran les més comunes. Aquestes són:

- Exactitud o *Accuracy*: percentatge de nombre de casos predits correctament respecte al nombre total de casos (Zhu et al., 2010:2):

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \in [0,1] \tag{24}$$

- Sensibilitat o *sensitivity* o *recall*: percentatge de nombre de casos verdaderament positius entre el nombre de casos realment positius (Rainio et al., 2024:2).

$$\text{Rec.} = \frac{\text{TP}}{\text{TP} + \text{FN}} \in [0,1] \tag{25}$$

- Especificitat o *specificity*: percentatge de nombre de casos verdaderament

negatius entre el nombre de casos realment negatius (Rainio et al., 2024:2).

$$\text{Spe.} = \frac{\text{TN}}{\text{TN} + \text{FP}} \in [0,1] \tag{26}$$

- Precisió o *precision*: percentatge de nombre de casos verdaderament positius entre el nombre de casos predits com a positius (Rainio et al., 2024:2).

$$\text{Pre.} = \frac{\text{TP}}{\text{TP} + \text{FP}} \in [0,1] \tag{27}$$

- *F1 score*: és una funció que mesura els valors realment positius i els valors predits com a positius (*precison* i *recall*):

$$\text{F1. score} = \frac{2 \cdot Prec. \cdot Rec}{Prec. + Rec} \in [0,1] \tag{28}$$

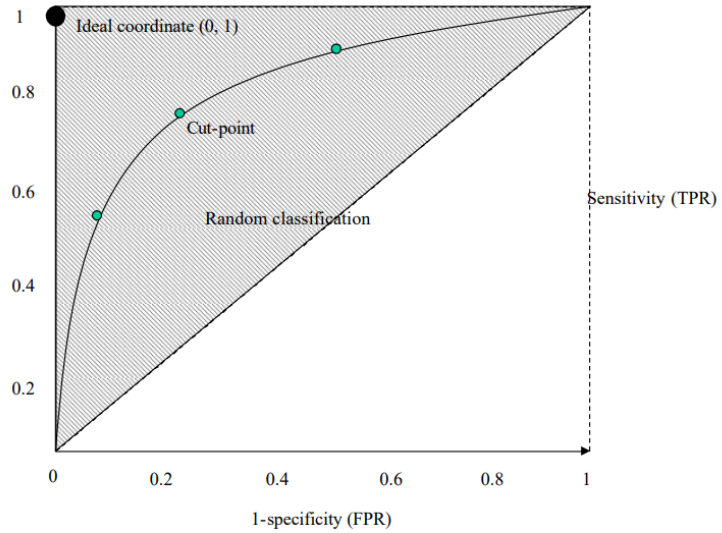
Cal però, saber de l'existència d'altres mètriques, que involucren els termes usats anteriorment, per això cal tenir en compte les usades en cas de classificació binària (Rainio et al., 2024:7). A través dels càlculs anteriors, és precís, comentar dos mètodes per l'avaluació d'algoritmes de classificació de text: aquest són el *receiver operating characteristics* -ROC- (Hanley i McNeil, 1982:29) i l'àrea sota la corba ROC, AUC (Rainio et al., 2024:2).

Les corbes ROC, són creades a partir de calcular la sensitivitat contra el percentatge de falsos negatius (Handelman et al., 2019:41), és a dir, sensitivitat -TPR- i, 1-specificitat (Zhu et al., 2010:3).

$$1 - \text{specificitat} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{29}$$

Per tant la corba ROC, quedaria tal que:

Figura 3.3. Representació corba ROC (Zhu et al., 2010:3).



Observant la figura 3.3., es pot veure com la corba es situa entre els punts (0,0) i (1,1). El millor classificador obtindrà un punt a la corba ROC de 1-specificity=0 i sensitivitat=1 (Pepe et al.,2008:1), ja que ens estaria dient que el 0% dels casos han estat etiquetats com a positius un cas realment negatiu i que un 100% dels casos són identificats correctament com a positius. Per tant, com més a prop estiguin els punts de la corba a la diagonal, menys exacte és l'algoritme i com més a prop de la coordenada (0,1) més precís (Zhu et al., 2010:2). Però cal tenir dos conceptes, que maca l'estudi de Zhu et al., 2010.

1. "The faster the curve approach the ideal point, the more useful the test results are."

2. "The slope of the tangent line to a cut-point tells us the ratio of the probability of identifying true positive over true negative."

Referent al segon punt, el mateix estudi es menciona que si la ràtio és 1 el punt de tall no incorpora informació addicional, si és més gran que 1, ajuda a trobar els resultats realment positius, i si es més petit que 1, la probabilitat de la categoria positiva decreix. Això és degut a la següent fórmula (Zhu et al., 2010:4):

$$\mathrm{LR} = \frac{\mathrm{Rec.}}{1 - \mathrm{Spe.}} \tag{30}$$

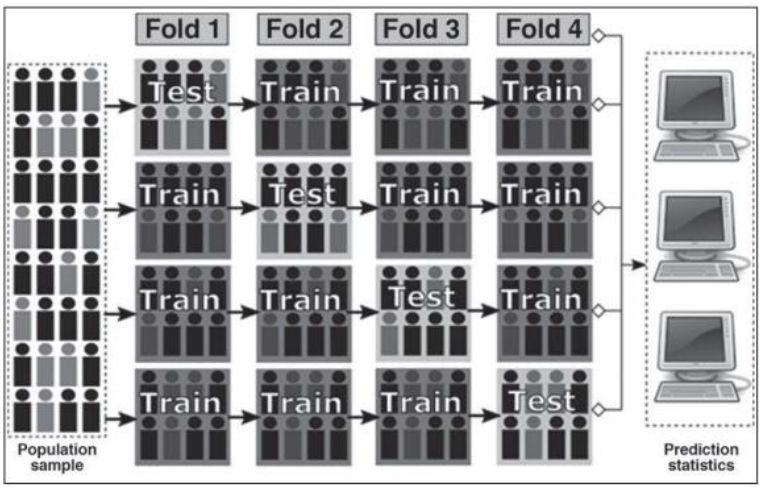
Per tenir, una altra mesura de la precisió del model, usant la corba ROC, es pot calcular l'àrea sota la corba ROC. Seguint amb l'explicació anterior, i usant la figura 3.3., es determina que el valor de l'AUC és tota l'àrea que s'ocupa per sota de la corba ROC (Zhu et al., 2010:4).

$$\mathrm{AUC} = \int_0^1 \mathrm{ROC(t)dt} \tag{31}$$

En conseqüència, com més gran sigui el valor de l'AUC més precís serà el nostre model (Chen i Guestrin, 2016:788).

A continuació, cal entendre que és la validació creuada o *cross-validation,* aquest és el mètode en què alguns algoritmes comencen a generar mesures de rendiment (Handelman et al., 2019:41). Es coneixen diferents tipus de validació creuada com, *Holdout Validation, Leave One Out Cross Validation* -LOOCV-, Validació creuada estratificada i k-fold cross-validation (GeeksforGeeks, 2023). En el cas que ens pertoca, l'interès se centra en la *k-fold cross-validation* -CV-. Aquest consisteix en un conjunt de dades dividit en k plegaments, el nombre de repeticions, ens donarà una idea de quantes vegades el model executarà el conjunt de dades, separat en k parts, on k-1 formen part de les dades d'entrenament i la resta les de prova (Wong i Yey, 2019:1586), per tant, el valor de k, pot variar, com més gran sigui, més execucions es tindran. L'esquema d'aquest mètode és mostra en la figura 3.4.:

Figura 3.4. Representació de la tècnica de *cross-validation* (Handelman et al., 2019:40).



Un dels aspectes positius d'aquesta metodologia és que els resultats de cada prova són extrets de cada *fold* i, per tant, s'obté la mitjana de cada prova (Schaffer, 1993:137), el que ens dona una visió global dels resultats del model, per altra banda, t'assegures que s'usen totes les dades tant per l'entrenament com per la prova (Sreekrishnan, 2023). A més a més, limita el risc de sobre ajustament, ja que el model s'aplica en diferents combinacions de dades i en conseqüència, el model no aprèn les variacions del conjunt, sent més fiable a l'hora de generalitzar (Handelman et al., 2019:41).

3.5. S&P 500

Per acabar de complementar l'estudi cal parlar sobre el S&P 500. La creació d'aquest índex borsari, data del 1923, quan va ser creat com a índex cobrint 233 empreses (Spglobal, n.d.), però no va ser fins al 1957, que l'indicador, tal com el coneixem avui, va aparèixer. El canvi va ser causar per una manera més eficient de realitzar els càlculs de l'índex. Aquest indicador inclou 500 empreses de gran capitalització de diferents sectors, i per tant capturar les tendències de l'economia empresarial americana, a més, és actualitzat cada quadri mestre, així com és usat com a indicador de referència per la majoria d'inversors (Beers, 2023).

Per altra banda, a l'hora de comparar sèries temporals, pot ser necessari estandarditzar o n ormalitzar. L'objectiu d'aquestes dues accions és transformar les dades en una escala adequada per a l'anàlisi, eliminant, en la mesura del possible, els efectes de les fonts sistemàtiques de variació (Muralidharan, 2010:1). L'estandardització és el mètode que aproxima una variable a una variable normal (Muralidharan, 2010:7). Per altra banda, els mètodes de normalització són: normalització per la mitjana, per quantils, entre d'altres (Muralidharan, 2010:6).

4. Fase 1: Comprensió dels textos

4.1. Enteniment de les variables

En aquest apartat es busca realitzar un aproximament a les bases del problema per tal d'entendre les magnituds d'aquest i per entendre la base de dades usades per aquest projecte. Per aquesta secció, cal entendre clarament la base de dades, així com las seves variables i característiques. Per altra banda, cal analitzar aquestes variables i poder entendre les característiques que ens aporten. En la següent taula es mostra una part de la base de dades usada:

Figura 4.1. Vista base de dades discursos inaugurals dels Presidents dels EE.UU (font pròpia).

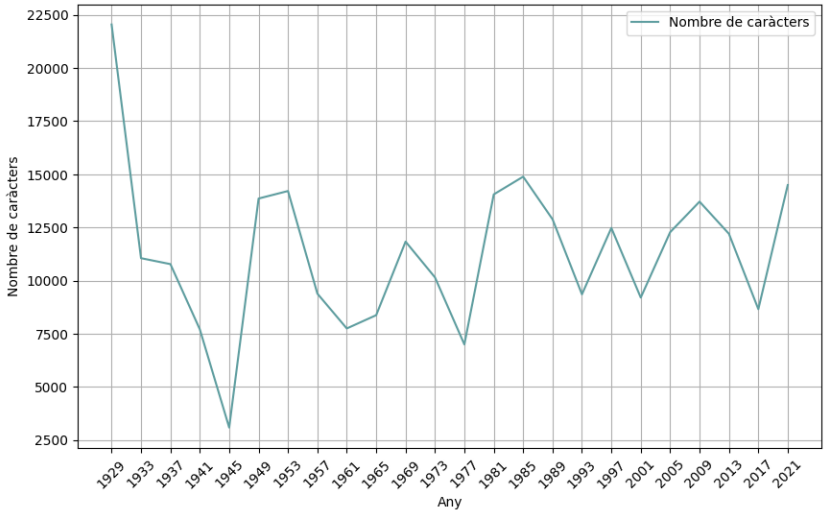
	president	year	speeches
36	Hoover	1929	My Countrymen : This occasion is not alone the...
37	Roosevel	1933	I am certain that my fellow Americans expect t...
38	Roosevel	1937	When four years ago we met to inaugurate a Pre...
39	Roosevel	1941	On each national day of inauguration since 178...
40	Roosevel	1945	Chief Justice , Mr . Vice President , my frien...
41	Truman	1949	Mr . Vice President , Mr . Chief Justice , and...
42	Eisenhower	1953	My friends , before I begin the expression of ...

Com es pot observar en la figura 4.1., la base de dades està composta per tres columnes. La primera columna, anomenada "president", fa referència al cognom del president que va pronunciar el discurs. Per exemple, en la fila 36, el discurs va ser pronunciat per Herbert Hoover. La segona columna, anomenada "year", indica l'any en què es va pronunciar el discurs. Aquesta data correspon a l'any d'inici de la proclamació del discurs, ja que el discurs inaugural del president es pronuncia a l'inici d'aquest. Finalment, la tercera columna, anomenada "speeches", conté el text del discurs pronunciat pel president corresponent. Per tant, si prenem com a exemple la fila 37, aquesta correspon al discurs pronunciat per Franklin D. Roosevelt l'any 1933. Aquesta estructura de la base de dades ens permet tenir una visió clara i organitzada de la informació, facilitant així l'anàlisi i interpretació dels discursos presidencials dels EE.UU. al llarg del temps.

4.2. Anàlisis descriptiva abans del processament

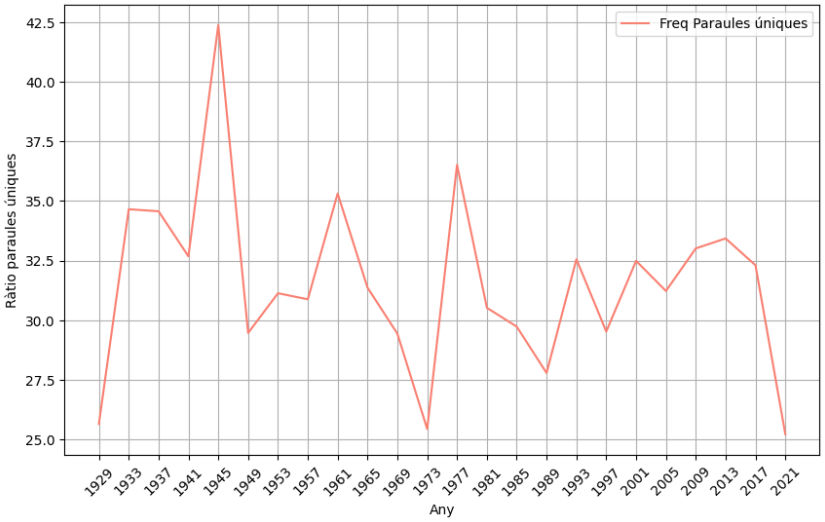
En aquest, apartat es busca evidenciar les característiques més rellevants que ens aporten el processament de textos a través dels NLP, els canvis que es generen i com aquests afecten la base de dades usada per l'anàlisi. Per fer-ho cal evidenciar i mostrar diferents tipus de descriptius que puguin ser útils per la visualització. L'anàlisi es farà sobre els 24 discursos analitzats, que després s'usaran per a la seva correlació amb l'índex borsari S&P 500.

Figura 4.2. Sèrie temporal del nombre de caràcters (font pròpia).



En la figura 4.2. es mostra una sèrie temporal del nombre de caràcters i del nombre de paraules dels 24 últims discursos inaugurals dels presidents dels Estats Units, sense comptar els espais ni els signes de puntuació. La sèrie comença amb Herbert Hoover el 1929 i arriba fins a Joe Biden el 2021. Com s'observa en el gràfic, les dades varien des de 3.086 caràcters fins a 22.043, donant un rang de 18.957 caràcters, cosa que indica una gran dispersió -desviació estàndard de la sèrie de 3.691,44-. El màxim de caràcters es va assolir en el primer any de la sèrie, amb el discurs del president Herbert Clark Hoover, un republicà, amb quasi 22.500 caràcters. És en l'últim discurs de Roosevelt que s'arriba al text més curt, amb només 3.049 caràcters. A mesura que avança la sèrie, es poden veure canvis notables en la longitud dels textos. Seguidament, és necessari presentar una mostra de la diversitat de cada discurs. Entenem per diversitat la varietat de paraules diferents que conté un text. La mesura de la diversitat es defineix com la ràtio de paraules úniques respecte al nombre total de paraules del text. Així, com més elevat sigui el percentatge, més variabilitat de paraules hi trobarem. Aquesta ràtio ens permet avaluar la riquesa lèxica del text, indicant-nos fins a quin punt s'utilitzen paraules diferents en la construcció del discurs.

Figura 4.3. Sèrie temporal de la ràtio de paraules úniques (font pròpia).



Com es mostra en la figura 4.3., la ràtio varia considerablement al llarg de la sèrie. El discurs amb més variabilitat és el de Franklin D. Roosevelt el 1945[22], amb una ràtio del 42,38%. Per contra, el text amb menys diversitat és el de Joe Biden el 2021, amb un 25,22%, seguit molt de prop pel discurs de Richard Nixon el 1971, amb un 25,44%. La mitjana de la sèrie se situa en el 31,557%.

4.2.1. Freqüències de les paraules

A continuació es mostren dos gràfics, de la tipologia anomenada núvol de paraules o *wordcloud*, en anglès, pels dos primers presidents dels Estats Units utilitzats per a l'anàlisi, Herbert Hoover -1929- i Joe Biden -2021-. L'objectiu d'aquest gràfic és visualitzar el nombre de vegades que es repeteixen les paraules, o identificar els caràcters més utilitzats per cada president. A més, el context d'aquest estudi, ens permetrà observar la necessitat d'aplicar tècniques de NLP per a dur a terme una anàlisi més precisa. És necessari mencionar que com més gran apareix la paraula en el gràfic, més elevada serà la seva freqüència.

Figura 4.4. *Wordcloud* discurs de Herbert Hoover, 1929 (font pròpia).

[22] 1945 és l'any del desembarcament de Normandia, que marca l'inici de la alliberació dels aliats. (*Cronología De Los Acontecimientos Internacionales*, n.d.)

Figura 4.5. *Wordcloud* de Joe Biden, 2021 (font pròpia).

S’observa en les figures 4.4. i 4.5. que les paraules més repetides en ambdós discursos són *the, of, and, is, that* i *in*[23]. Com es pot intuir, aquestes paraules no aporten cap informació rellevant al text. Per exemple, la preposició *of* no transmet cap informació significativa. És per això que és necessari tractar i processar el text abans de dur a terme una anàlisi més profunda. D'altra banda, observem la presència de diversos signes de puntuació com " ; ", " , ", i " . ". És important tenir en compte que, en aquest context, els estem considerant com a caràcters. Aquest fet pot causar problemes en l'anàlisi, ja que els signes de puntuació no aporten informació rellevant al text en si mateix. La seva inclusió en el recompte de caràcters pot distorsionar els resultats i conduir a conclusions inadequades. En aquest sentit, cal destacar la importància de fer una anàlisi més exhaustiva i sistemàtica per a identificar i eliminar tota la informació redundant i les paraules que no aporten valor rellevant a l'anàlisi.

5. Fase 2: Processament de les dades

En aquesta etapa del procés, és imprescindible normalitzar les dades per garantir que l’estructura dels textos sigui tan homogènia com sigui possible. Per normalitzar el text, disposem de diverses eines, tal com s’ha mencionat anteriorment. Algunes d’aquestes eines inclouen l’extracció de signes de puntuació, la *tokenització* del text, l’eliminació de paraules que no aporten informació rellevant i la reducció de les paraules a la seva forma d’arrel. És vital prestar atenció a qualsevol anomalia que pugui aparèixer en els textos durant aquesta fase. Qualsevol desviació de la normalitat pot afectar la precisió dels resultats finals, per la qual cosa és important assegurar-se que les dades estan adequadament normalitzades abans de procedir a les etapes d’anàlisi posteriors.

[23] Són conjuncions, articles, preposicions i adverbis (Agencia de Marketing Online BlackBeast, 2019)

5.1. Signes de Puntuació i Minúscules

A continuació cal extreure dels textos tots aquells signes de puntuació, o signes en general, ja que el codi no entén les puntuacions i, per tant, la sexa existència només ens aporta soroll (Tabassum i Patil, 2020:4865). Cal recordar que el codi identifica com diferent una paraula amb majúscula i una en minúscula i, com a resultat, les transformacions consegüents també les consideraran diferents (Tabassum i Patil, 2020:4865).

Seguidament, es presenta un fragment del discurs pronunciat per John F. Kennedy l'any 1961, en el quadre 5.1. El text que es visualitza a la columna esquerra de la taula correspon al format original del discurs. Aquest format inclou tots els elements característics d'un text escrit, com ara els signes de puntuació, les majúscules i altres elements tipogràfics. En contraposició, el text que es mostra a la columna dreta de la taula ha estat sotmès a un procés de normalització. Aquest procés ha eliminat tots els signes de puntuació, inclòs comes i punts, i ha convertit totes les lletres a minúscules. Aquesta versió normalitzada del text facilita l'anàlisi lingüística i computacional, ja que elimina les variacions superficials que poden interferir amb aquests processos.

Quadre 5.1. Exemple d'exclusió dels signes de puntuació i de posada en minúscula de les paraules (font pròpia).

Abans dels Signes de Puntuació	Després dels Signes de Puntuació
Vice President Johnson , Mr . Speaker , Mr . Chief Justice , President Eisenhower , Vice President Nixon , President Truman , reverend clergy , fellow citizens , we observe today not a victory of party , but a celebration of freedom -- symbolizing an end , as well as a beginning -- signifying renewal , as well as change . For I have sworn I before you and Almighty God the same solemn oath our forebears l prescribed nearly a century and three quarters ago .	vice president johnson mr speaker mr chief justice president eisenhower vice president nixon president truman reverend clergy fellow citizens we observe today not a victory of party but a celebration of freedom symbolizing an end as well as a beginning signifying renewal as well as change for i have sworn i before you and almighty god the same solemn oath our forebears l prescribed nearly a century and three quarters ago

Les diferències entre els textos són notablement evidents quan es realitza una comparació directa. Observem una distinció clara, per exemple, en la majúscula inicial de la frase o en la referència a l'anterior president, el president Eisenhower. En el text original, aquestes paraules apareixen en majúscules, mentre que en el text normalitzat no. Pel que fa als signes de puntuació, les diferències també són significatives. Les comes han estat eliminades, igual que el punt final i altres guions. Cal destacar la reducció en el nombre de caràcters del text d'exemple, en el tros de discurs d'abans d'extreure els signes de puntuació hi trobem 357 caràcters, inclòs espais i signes de puntuació. Per contra, el text de després d'extreure aquests signes de puntuació, ha vist reduït el nombre de caràcters a 229. És a dir, el nombre de caràcters s'ha reduït en 128, cosa que és equivalent a un 36%.

5.2. Tokenització de les discursos

Ara, cal separar les frases i els discursos en paraules, és a dir cal processar el text, per dividir el text en unitat mínima d'informació, per tenir el conjunt de dades en un grup de llenguatge de dades (Bird et al., 2009:109), aquest llenguatge d'informació mínima és conegut com a *token*. En aquest sentit, ens cal tenir les paraules separades per poder treballar millor, en el nostre cas se separen les paraules del discurs. Aquest procés ens ajudarà a filtrar pròximament paraules no necessàries per a l'anàlisi (Tabassum i Patil, 2020:4865).

Quadre 5.2. Exemple d'aplicació de la *tokenització* (font pròpia).

Abans de la Tokenització	Després de la Tokenització
vice president johnson mr speaker mr chief justice president eisenhower vice president nixon president truman reverend clergy fellow citizens we observe today not a victory of party but a celebration of freedom symbolizing an end as well as a beginning signifying renewal as well as change for i have sworn i before you and almighty god the same solemn oath our forebears l prescribed nearly a century and three quarters ago	['vice', 'president', 'johnson', 'mr', 'speaker', 'mr', 'chief', 'justice', 'president', 'eisenhower', 'vice', 'president', 'nixon', 'president', 'truman', 'reverend', 'clergy', 'fellow', 'citizens', 'we', 'observe', 'today', 'not', 'a', 'victory', 'of', 'party', 'but', 'a', 'celebration', 'of', 'freedom', 'symbolizing', 'an', 'end', 'as', 'well', 'as', 'a', 'beginning', 'signifying', 'renewal', 'as', 'well', 'as', 'change', 'for', 'i', 'have', 'sworn', 'i', 'before', 'you', 'and', 'almighty', 'god', 'the', 'same', 'solemn', 'oath', 'our', 'forebears', 'l', 'prescribed', 'nearly', 'a', 'century', 'and', 'three', 'quarters', 'ago']

Com es pot observar en el quadre 5.2., un cop s'han extret els signes de puntuació i s'ha convertit el text a minúscules, és necessari procedir a l'aplicació de la *tokenització* de les paraules. Aquest procés es detalla en el Quadre 5.2. S'observa com cada paraula es converteix en una unitat independent.

5.3. StopWords

Seguidament, és necessari procedir a l'eliminació de certes paraules que es consideren d'alta freqüència. Aquestes inclouen conjuncions, articles, preposicions i adverbis, és a dir, paraules que no aporten informació semàntica significativa (Avela i Lehmus, 2020:9). Aquesta eliminació és un pas crucial abans de continuar amb el processament del text (Bird et al, 2009:60). No obstant això, és important tenir en compte quines paraules es consideren *stopwords* o paraules d'aturada. Aquesta etapa permet reduir el soroll i la complexitat dels textos, facilitant així l'anàlisi posterior i millorant la precisió dels resultats.

Quadre 5.3. Exemple d'eliminació de paraules no informatives -StopWords- (font pròpia).

Abans de l'extracció de les StopWords	Després de l'extracció de les StopWords
['vice', 'president', 'johnson', 'mr', 'speaker', 'mr', 'chief', 'justice', 'president', 'eisenhower', 'vice', 'president', 'nixon', 'president', 'truman', 'reverend', 'clergy', 'fellow', 'citizens', 'we', 'observe', 'today', 'not', 'a', 'victory', 'of', 'party', 'but', 'a', 'celebration', 'of', 'freedom', 'symbolizing', 'an', 'end', 'as', 'well', 'as', 'a', 'beginning', 'signifying', 'renewal', 'as', 'well', 'as', 'change', 'for', 'i', 'have', 'sworn', 'i', 'before', 'you', 'and', 'almighty', 'god', 'the', 'same', 'solemn', 'oath', 'our', 'forebears', 'l', 'prescribed', 'nearly', 'a', 'century', 'and', 'three', 'quarters', 'ago']	['vice', 'president', 'johnson', 'mr', 'speaker', 'mr', 'chief', 'justice', 'president', 'eisenhower', 'vice', 'president', 'nixon', 'president', 'truman', 'reverend', 'clergy', 'fellow', 'citizens', 'observe', 'today', 'victory', 'party', 'celebration', 'freedom', 'symbolizing', 'end', 'well', 'beginning', 'signifying', 'renewal', 'well', 'change', 'sworn', 'almighty', 'god', 'solemn', 'oath', 'forebears', 'l', 'prescribed', 'nearly', 'century', 'three', 'quarters', 'ago']

El quadre 5.3 presenta un exemple d'un fragment del discurs de John F. Kennedy de l'any 1961. La reducció de paraules és notable en aquest exemple. Cal destacar que tots els mots "i" han estat eliminats, així com altres paraules com "as" o el substantiu "end". És important realçar que en aquest petit exemple, el nombre total de paraules s'ha reduït de 75 a 46 després de l'eliminació de paraules redundants. Això significa que només en aquest fragment de text, s'ha aconseguit una reducció de 29 paraules, que representa una disminució del 38,7%.

5.4. Stemming

Finalment, per completar la normalització del text per a la seva anàlisi, és necessari portar cada paraula a la seva forma d'arrel. Aquest procés implica l'eliminació de tots els sufixos i prefixos de les paraules, un pas essencial per a la neteja dels textos. Aquesta normalització permet que el model reconegui les paraules en les seves formes bàsiques, independentment de les seves variacions morfològiques. Cal Recordar que l'ús de la tècnica de *stemming* ens permet preservar una gran quantitat d'informació (Avela i Lehmus, 2020:9).

Quadre 5.4. Exemple de l'aplicació de la tècnica *Stemming* (font pròpia).

Abans de l'aplicació del Stemming	Després de l'aplicació del Stemming
['vice', 'president', 'johnson', 'mr', 'speaker', 'mr', 'chief', 'justice', 'president', 'eisenhower', 'vice', 'president', 'nixon', 'president', 'truman', 'reverend', 'clergy', 'fellow', 'citizens', 'observe', 'today', 'victory', 'party', 'celebration', 'freedom', 'symbolizing', 'end', 'well', 'beginning', 'signifying', 'renewal', 'well', 'change', 'sworn', 'almighty', 'god', 'solemn', 'oath', 'forebears', 'l', 'prescribed', 'nearly', 'century', 'three', 'quarters', 'ago']	['vice', 'presid', 'johnson', 'mr', 'speaker', 'mr', 'chief', 'justic', 'presid', 'eisenhow', 'vice', 'presid', 'nixon', 'presid', 'truman', 'reverend', 'clergi', 'fellow', 'citizen', 'observ', 'today', 'victori', 'parti', 'celebr', 'freedom', 'symbol', 'end', 'well', 'begin', 'signifi', 'renew', 'well', 'chang', 'sworn', 'almighti', 'god', 'solemn', 'oath', 'forebear', 'l', 'prescrib', 'near', 'centuri', 'three', 'quarter', 'ago']

Com es pot observar en el quadre 5.4., es detalla el procés de normalització final del text. Aquesta fase és d'una importància crucial, ja que és en aquest moment quan obtenim el text en el format requerit per a la seva posterior implementació en el model d'anàlisi. En la taula es pot apreciar que diverses paraules han estat simplificades a les seves formes més bàsiques. Per exemple, la paraula 'president' s'ha convertit en "*presid*", "*justice*" en "*justic*", "*observe*" en "*observ*", i "*prescribed*" en "*prescrib*", entre d'altres. Aquesta simplificació de paraules és un pas fonamental en el procés de normalització del text.

6. Fase 3: Etiquetatge previ dels textos

Com s'ha pogut verificar en les observacions prèvies, la base de dades que s'ha utilitzat per a aquest estudi no disposa de cap columna que serveixi com a objectiu o classificació. Aquesta situació implica que, per a aquest conjunt de dades en particular, és necessari establir un sistema d'etiquetatge basat en les dades obtingudes, específicament, a partir dels textos disponibles. L'etiquetatge dels discursos que es presenten en aquest treball és d'una importància crucial, ja que és a partir d'aquesta base que el model d'aprenentatge automàtic serà entrenat. Per aquesta raó, és absolutament imprescindible ser extremadament meticulós en l'elecció de les mètriques que es faran servir per a la creació d'aquest sistema d'etiquetatge. En aquest context, és important tenir en compte dos factors clau. En primer lloc, l'objectiu principal d'aquest estudi és la creació d'un índex d'incertesa. Per tant, aquesta classificació estarà orientada cap a la identificació i quantificació de la incertesa. En segon lloc, és crucial seleccionar un diccionari de paraules que siguin sinònimes d'incertesa. És important assegurar-se que s'han recollit totes les paraules que poden ser considerades sinònimes d'incertesa.

Exemple d'etiquetatge de Textos:

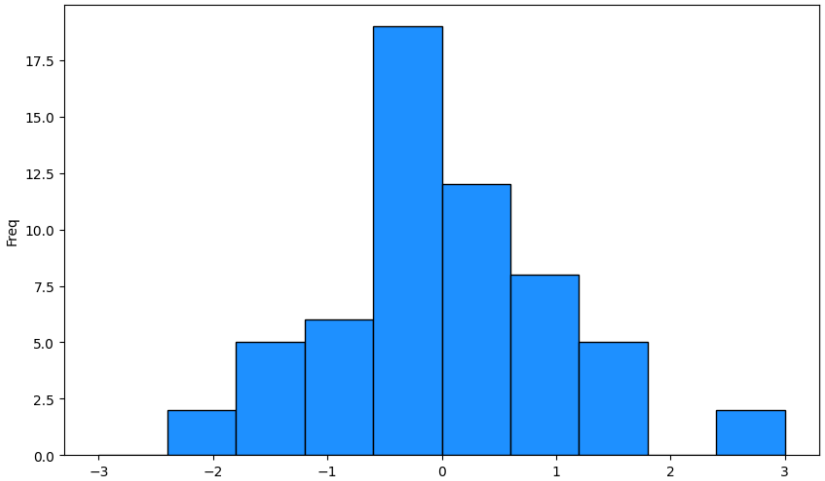
En el context de l'anàlisi lingüística, un dels aspectes que es consideren és la presència de paraules que denoten incertesa. Aquestes paraules, que són sinònimes d'incertesa, poden ser identificades a partir de les seves arrels. Alguns exemples d'aquestes paraules són *possible, could, like, perhaps, worry, wonder, might i distrust.* Per il·lustrar aquest punt, podem referir-nos a algunes frases pronunciades per Joe Biden, l'actual president dels Estats Units. En el discurs inaugural, el 2021 (House, 2021), Biden va dir: "Distrusting those who don't look *like*

you do". En una altra ocasió, Biden va afirmar que "is *perhaps* our nation's greatest strength". Finalment, Biden també va expressar "I understand they *worry* about their jobs". En total, en el discurs de Biden es van identificar 14 paraules d'incertesa. Tenint en compte que el discurs constava de 1.191 paraules, això representa un 1,175% de paraules d'incertesa. No obstant això, aquesta anàlisi no acaba aquí. Per obtenir una mesura més precisa de la incertesa en el discurs, es pot restar la mitjana i dividir aquest resultat per la desviació estàndard, és a dir, estandaritzar. Això ens permet obtenir una sèrie de valors amb una mitjana de 0 i una desviació estàndard d'1. En l'exemple que hem exposat, aquest càlcul ens dona un resultat d'1.32. És important tenir en compte que com més gran sigui aquest valor, més incert serà el text.

Després de la creació de l'índex, és imprescindible desenvolupar un histograma. Aquest histograma tindrà la funció crucial de determinar el valor específic que establirà si un text es classificarà com a "Incertesa" o "No Incertesa". L'ús d'un histograma en aquest context és vital perquè ens proporciona una representació gràfica de la distribució de les dades. Aquest valor llindar serà el punt de decisió que determinarà com s'etiquetarà cada text en el nostre estudi. Si un text té un valor per sobre del llindar, es considerarà *Uncertainty*. En canvi, si el valor és inferior al llindar, el text es classificarà com a *Not Uncertainty*. En aquest cas, l'etiquetatge es farà per a tots els textos de la base de dades, des de 1789 fins a 2021, degut a que tots els discursos seran usats per l'aplicació del model.

La figura 6.1, mostra com quasi la meitat dels valors es troben entre el valor de -0.5, i 0.5, en concret un 45,76%, és a dir 27 textos de 59 es troben en un rang de valors que es consideraran de baixa incertesa o de no incertesa dèbil. Per altra banda, cal comentar els valors atípics del gràfic, aquests es troben, per la part d'incertesa – és a dir, la de valors positius-, en

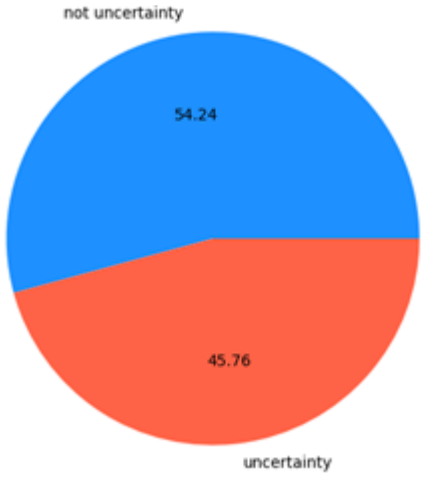
Figura 6.1. Histograma del percentatge estandarditzat de paraules incertes en els textos (font pròpia).



dos textos, primer en el de George Washington, el 1789-, on s'han trobat 10 paraules, el seu percentatge de paraules incertes és d'1,54% sobra el total de paraules. El valor més elevat trobat en el procés d'etiquetatge dels textos és el del president Lyndon B. Johnson, el 1965, amb 11 paraules d'incertesa en el seu text. Per contra, si ens fixem en els valors de no incertesa -és a dir, els negatius-, hi apareixen el segon discurs del president Washington, el 1793, on no apareix com paraula relacionada amb incertesa, el mateix passa amb el president Nixon el 1973, és per aquest motiu, que ambdós presidents obtenen el mateix valor a l'hora d'etiquetar els textos, aquest és de -2.28.

Un cop vista la distribució del càlcul per etiquetar els textos, cal etiquetar-lo. S'ha usat el valor de la mitjana del percentatge estandarditzat calculat com a valor llindar per anomenar els textos, assumint els erros que pugui causar, alhora mirant la seva precisió en el futur model. Cal tenir, en compte, en realitzar l'etiquetatge, els problemes de balanceig que es poden ocasionar si no es divideix adequadament la base de dades. La distribució de les classes de la base de dades és força equilibrada, i tot i haver-hi més nombre de discursos *not uncertainty* la diferència és d'un 8,48% o de 5 textos. Per tant, la distribució de 54,24% per la classe *not uncertainty* i 45,76% per la classe *uncertainty*, permet a aquest estudi treballar amb dades balancejades.

Figura 6.2. Gràfic de sectors, de la distribució de classes de la base de dades de les etiquetes de la base de dades (font pròpia).



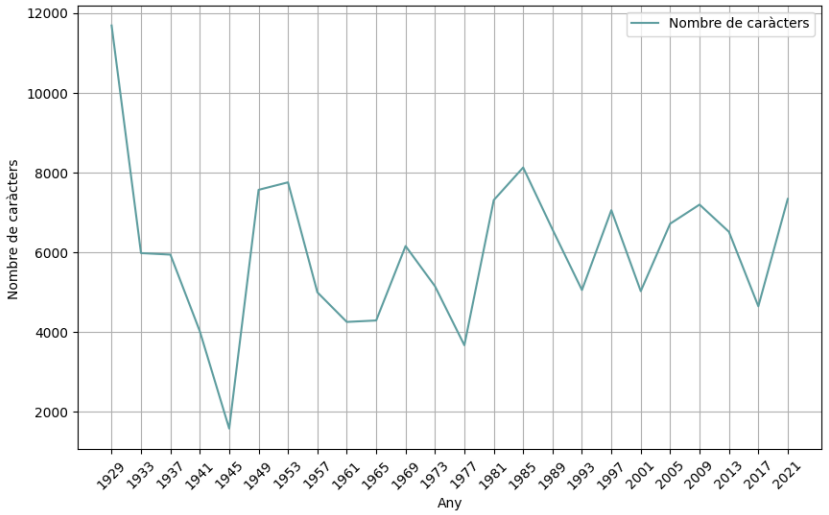
Com mostra la figura 6.2., la distribució de les classes de la base de dades és força equilibrada, i tot i haver-hi més nombre de discursos not uncertainty la diferència és d'un 8,48% o de 5 textos. Per tant, la distribució de 54,24% per la classe not uncertainty i 45,76% per la classe *uncertainty*, permet a aquest estudi treballar amb dades balancejades.

7. Fase 4: Comprensió dels textos després del processament

7.1. Anàlisis descriptiu després del processament

En aquest apartat, s'exposarà els mateixes anàlisis descriptives, que en fase 1 i 2. En la pròxima anàlisi, es pretén veure la variació de les mètriques un cop netejats els textos de l'anàlisi a través de les tècniques de NLP. Les conseqüències de l'aplicació els podem veure en els següents resultats.

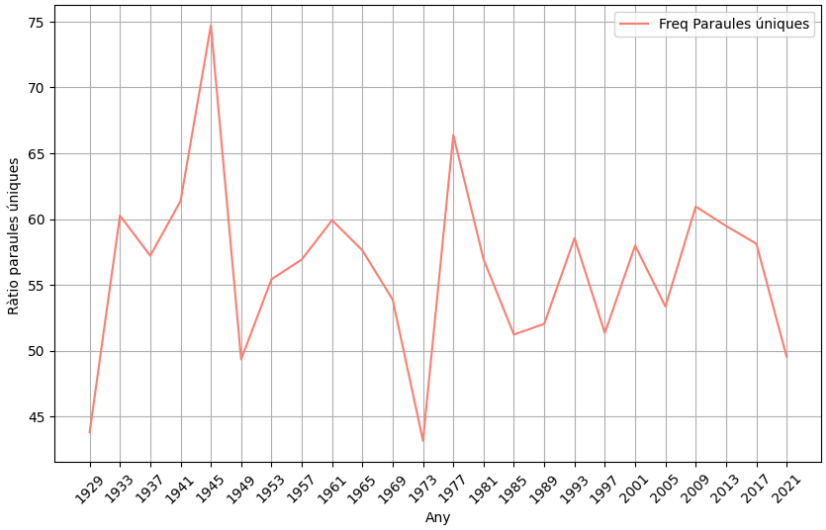
Figura 7.1. Sèrie temporal del nombre de caràcters (font pròpia).



En primer lloc, cal observar que la silueta de la sèrie temporal en el gràfic anterior no ha variat, tot i haver aplicat les tècniques de processament. El pic i el punt més baix continuen sent els mateixos. No obstant això, el que ha canviat és la grandària dels textos. De fet, en haver reduït la dimensionalitat de les dades, també es redueix la dispersió del text. El discurs amb el nombre de caràcters més elevat correspon al primer de l'anàlisi, el de Herbert Hoover el 1929. Abans del processament, aquest discurs comptava amb 22.043 caràcters, mentre que després del processament en té 11.686. Això significa que la dimensionalitat del text ha disminuït en 10.175 caràcters, equivalent a un decreixement d'un 46,15%. Aquesta reducció ens serà d'utilitat per al tractament posterior del text. Pel que fa al punt més baix, també es nota una clara disminució en el nombre de caràcters, passant de 3.086 a 1.571. Això representa una diferència de 1.515 caràcters, equivalent a una disminució del 49,09%.

A continuació, cal representar la nova dispersió dels textos. Com s'ha fet anteriorment, per mesurar la dissonància, s'ha calculat la ràtio de paraules úniques respecte al total de paraules.

Figura 7.2. Sèrie temporal de la ràtio de paraules úniques (font pròpia).



En la figura 7.2. s'observen certes modificacions en la silueta dels gràfics en comparació amb la situació anterior al processament. Aquestes variacions es localitzen en el segon i tercer discurs -1937 i 1941- de Franklin D. Roosevelt, en el segon discurs de Dwight D. Eisenhower -1957-, en el discurs de Lyndon B. Johnson -1965-, en el discurs de Ronald Reagan -1985- i en el discurs de Barack Obama -2013-. Aquests canvis en les tendències es deuen principalment a l'homogeneïtzació de les paraules i al consegüent augment de la ràtio.

L'increment del percentatge de paraules úniques pot ser atribuït a dos factors. Un d'ells és la reducció de la dimensionalitat de les dades, és a dir, el denominador del càlcul ha disminuït. Hem eliminat paraules sense informació i signes de puntuació, fet que ha reduït gairebé a la meitat el nombre total de paraules del text. Un altre cas, seria que el numerador, és a dir, el nombre de paraules úniques, hagués augmentat a causa del processament. No obstant això, aquesta situació no té gaire sentit, ja que durant aquest procés s'han normalitzat diferents paraules, i finalment la paraula s'ha quedat en la seva arrel. Per tant, una paraula com b*uilding* i *build*, que anteriorment es comptaven com a dues paraules diferents, ara es compten com a una sola, reduint el nombre del numerador. En conclusió, l'augment de la ràtio de paraules úniques es deu a la disminució, gairebé a la meitat, del nombre total de paraules. Per tant, observant els nous resultats, es pot afirmar que el text amb més diversitat és el de Franklin D. Roosevelt -1945-, amb un 74,70% de paraules úniques en el text. El president amb una diversitat de text menys elevada i, com a resultat, amb un discurs no gaire ric lèxicament, és Richard Nixon, en el seu segon discurs el 1974, amb una ràtio del 43,17%, seguit molt de prop per Herbert Hoover, amb un 43,81%.

7.1.1. Freqüències de les paraules

Un aspecte important, recau en la representació gràfica dels nous núvols de paraules, seleccionats amb l'objectiu de destacar les paraules més freqüents. És imperatiu tenir present que en els gràfics anteriors, abans del processament, es mostrava informació no rellevant, com ara conjuncions, articles, preposicions i adverbis. Això evidencia la necessitat de depurar els discursos. Per consegüent, els gràfics posteriors ja incorporen les paraules informatives de cada text, les quals han estat sotmeses al procés de *stemming*. Això pot comportar que en alguns casos es trobin paraules que presenten la seva arrel.

Figura 7.3. *Wordcloud* discurs de Herbert Hoover, 1929 (font pròpia).

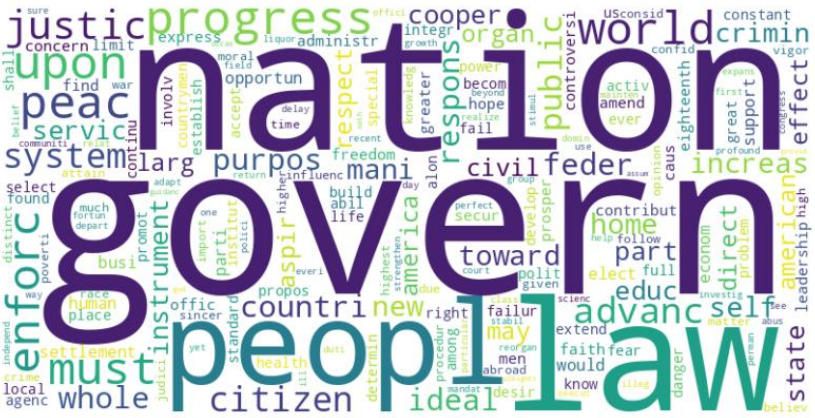


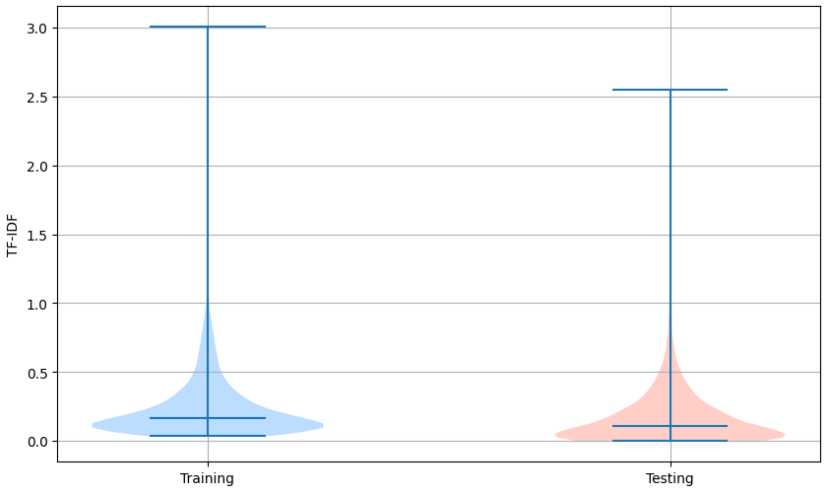
Figura 7.4. *Wordcloud* discurs de Joe Biden, 2021 (font pròpia).

El núvol de punts de les figures 7.3. i 7.4., tal com abans del processament, mostra les paraules més freqüents dels discursos de Herbert Hoover i de Joe Biden. Com es pot observar en aquest exemple, les paraules més recurrents són *nation*, *america*, *american*, *people*, i altres paraules que principalment busquen fomentar la unitat del poble nord-americà, i molt relacionades en treballar junts cap a un objectiu comú. Cal destacar la disparitat en les paraules més freqüents de cada discurs. El discurs de Hoover, en gran part, promou la noció d'un govern com a salvaguarda de la nació, mantenint un ordre establert, amb paraules que apareixen un elevat nombre de cops com *nation*, *govern*, *peopl* i *law*. En contraposició, el discurs de Joe Biden sembla posar més èmfasi en la idea d'unitat i identitat del poble nord-americà amb l'ajuda de vocabulari com *america*, *american*, *us*, *nation* o *one*.

8. Fase 5 i 6: Resultats AdaBoost Classifier

En aquesta exposició detallada, es presenten els resultats derivats de l'estudi realitzat. La primera fase a l'hora d'analitzar els resultats correspon a mostrar la eficàcia del classificador utilitzat en l'estudi. Aquesta part de l'anàlisi, que anomenarem, validació del model, es centra en determinar com de precís ha estat el classificador en la seva tasca Les dades han estat dividides en dos conjunts: les dades d'entrenament i les dades de prova. Aquesta àmplia base de dades pot ser una eina útil per identificar canvis de tendència al llarg del temps. No obstant això, és important recordar que els presidents que s'han analitzat de manera més detallada són els que van ocupar el càrrec des del 1929 fins al 2021. Els presidents anteriors a aquesta data s'han utilitzat principalment com a dades d'entrenament i prova per al model. Aquesta distinció és crucial per a la interpretació correcta dels resultats obtinguts en l'estudi. Per la tal d'entrenar el model, aquest ha sigut separat entre *traint* i *test*, s'ha usat una separació 60-40, és a dir, el model ha sigut entrenar amb el 60% de la base de dades, i posteriorment ha sigut provat amb el 40% de la base de dades, amb aquesta partició s'ha obtingut millors resultats, que amb altres possibles separacions com 70-30. És necessari comentar, que els dos subconjunts mantenen la proporció de les classes, és a dir, el percentatge de textos de les classes *uncertainty* i *no uncertainty*, s'ha mantingut en la separació. En el següent gràfic es mostra la distribució dels dos subconjunts.

Figura 8.1. Diagrama de freqüències pels subconjunts *training* i *test* (font pròpia).



La figura 8.1. representa la distribució de freqüències del conjunt de dades després d'aplicar TF-IDF. Com és d'esperar, el conjunt *training* mostra una distribució més àmplia de la freqüència dels termes. El conjunt *test* té una distribució més estreta, el que suggereix que els termes tenen una distribució més concentrada. En conseqüència, es podria dir que el model estarà exposat a una gran varietat de termes durant l'entrenament, fet que és positiu per la seva capacitat de generalització.

8.1. Avaluació dels Resultats

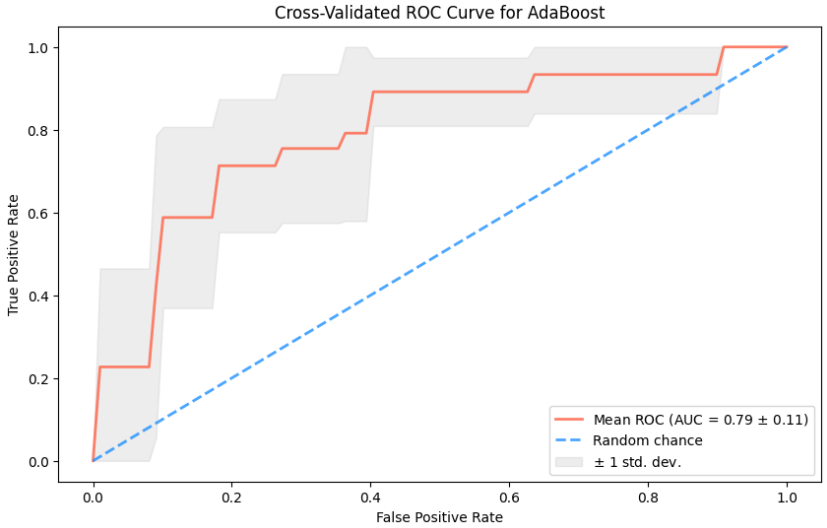
Els resultats del classificador AdaBoost, contenen diferents mètriques d'avaluació, com *accuracy, precisió, recall i f1-score.* També cal tenir en compte, l'aplicació de la tècnica de 3-*fold cross-validation* per acabar de verificar els resultats. En la taula següent es mostren els resultats d'aplicar el model al conjunt de prova, conformat per 18 textos.

Quadre 8.1. Mètriques d'avaluació de l'aplicació del model AdaBoost classifier (font pròpia).

	Accuracy	AUC	Precision	Recall	F1-score
Uncertainty	0.71	0.73	0.71	0.64	0.67
Not Uncertainty			0.70	0.78	0.74

Els resultats del quadre 8.1. fan referència als resultats de l'aplicació del model al 40% de les dades; per tant, estem parlant de resultats obtinguts una vegada el model ha sigut entrenat amb el 60% de les dades restants. Cal posar en context cada valor vist a la taula. *L'accuracy* del model és del 71%, el que indica que és un model prou bo, donades les condicions del problema i el seu context. Això significa que el model classifica correctament 71 de cada 100 textos. Aquest 71%, no obstant això, ha d'acompanyar-se d'altres mesures com la precisió per a cada classe. En aquest cas, quan el model classifica un text com *not uncertainty*, és correcte el 70% de les vegades. En canvi, pel grup *uncertainty*, el 71% s'han identificat correctament - *precision*. Pel que fa al *recall*, indica que per tots els textos que realment pertanyen al grup *not uncertainty*, el model ha identificat correctament el 78% dels casos. En conclusió, mirant l'*F1-score*, que dona la mitjana harmònica entre la precisió i el *recall*, observem que la classe *uncertainty* presenta resultats més baixos en la sensibilitat. Això vol dir que el model només ha estat capaç d'identificar com a incerts el 64% de tots els textos incerts. No obstant això, quan els textos han estat predits com a incerts, la predicció ha estat correcta en un 71% dels casos en el cas de la classe *uncertainty*.

Figura 8.2. Corba ROC model AdaBoost (font pròpia).



A més a més, cal comentar els resultats de la corba ROC i de l'AUC, que es mostren a la figura 8.2. En primer lloc, cal destacar que a la figura 8.2, mostra els resultats de calcular el valor de l'AUC, a través de validació encreuada de 3 plegaments. La corba es troba per sobre

de la línia que creua el gràfic -línia blava-, la qual indica que el model no és millor que triar aleatòriament. Cal recordar que com més allunyada estigui la corba de la línia discontinua, millor serà el model. Això ens permet veure que per una taxa de falsos positius -FPR- d'aproximadament 0.2, la taxa de vertaders positius -TPR- ha d'estar propera a 0.6. D'altra banda, la taxa màxima de TPR que pot aconseguir-se, que és 1-0, és amb una taxa de FPR de 0.9. Per altra banda, el gràfic, mostra les variacions de la corba ROC aportades per la validació creuada. Aquest marge de confiança és trobat a partir de, a cada punt sumar o restar el valor de la corba ROC. Ara, si ens fixem en el valor de l'AUC, aquest és de 0.79, el que indica que el model té la capacitat de classificar correctament el 79% de les instàncies i sembla ser capaç de distingir entre instàncies negatives i positives. A més a més, la variació d'aquest valor és de més/menys 0.11 punts.

9. Fase 7 i 8: Resultats Índex d'incertesa

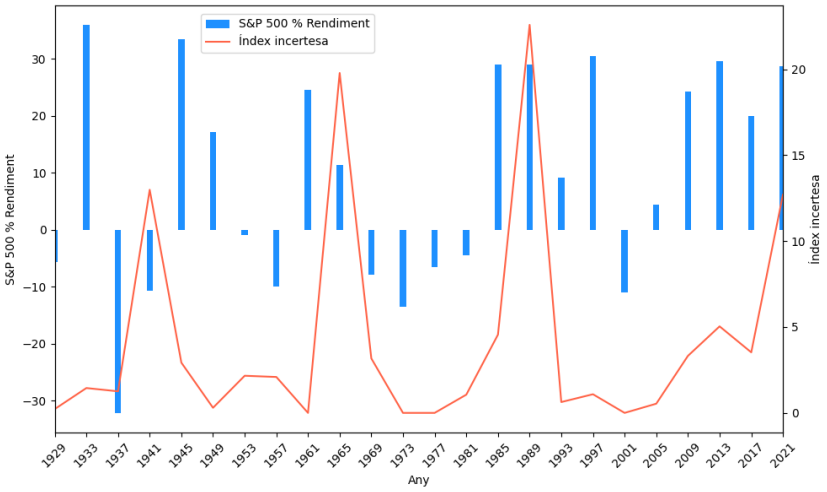
A continuació, és precís presentar els resultats de l'índex creat. Aquest índex ha estat calculat utilitzant els pesos atorgats pel TF-IDF, els pesos de cada paraula proporcionats pel model, així com el nombre de vegades que aquestes paraules han aparegut en els textos inicials. En la propera explicació es presenten els resultats de diverses formes: es presenta l'índex creat, la probabilitat a posteriori de pertànyer a la classe incerta, i la sèrie temporal de l'S&P 500 des de 1929 fins al 2021, així com les tres sèries normalitzades.

La sèrie temporal mostra el percentatge de rendiment anual de l'índex S&P 500. Cal tenir en compte que aquests valors poden ser tant positius com negatius. El valor mínim d'aquesta sèrie seria si el rendiment hagués tocat fons, situant-se a -100% de rendiment, mentre que el valor màxim no té un límit teòric, ja que una inversió pot créixer més d'un 100%. A més, és necessari mostrar els valors d'aquest índex, en la totalitat de l'any de l'índex. Aquesta anàlisi permet avaluar amb més precisió la incertesa mesurada per l'índex, proporcionant així una visió més completa i profunda de la seva influència. Per a la creació de l'índex s'han tingut en compte diversos factors, com s'ha comentat anteriorment. Així doncs, l'índex és calculat, fent la ràtio del valor de cada component, entre el valor màxim de tots els textos d'aquella component i multiplicant cada resultat de cada component. Finalment, s'ha multiplicat el resultat per 100 per obtenir una escala més manejable.

La figura 9.1. mostra el rendiment percentual de l'indicador S&P 500, del 1929, fins al 2021, destacar, que l'inici de la sèrie es veu afectada pel Crack del 29, ocorregut, com el seu nom indica, el 1929 (Sala, 2023). Continuant amb la gran crisi, el 1933, es pot observar com el S&P 500, ja s'ha recuperat i es situa en valor de 36% , en aquell mateix any, el president Franklin D. Roosevelt, va prometre, i va començar a implementar els coneguts com a *New Deal*, una seria de mesures, econòmiques i socials per començar amb la recuperació del país (Onion, 2023). Tot i això, i observant el gràfic, el 1937, es troba el valor més baix de tota la sèrie amb un % de retorns de -32.11, en aquell, i tot i l'intent, de Franklin D. Roosevelt de frenar la crisi, el 1937, els Estats Units van patir una forta recessió, a causa de l'estímul del govern per reduir la despesa (Onion, 2023). Aquests exemples serveixen, per veure que el valor del S&P 500, en algunes ocasions, es veu afectat pels esdeveniments que afecten el país.

La figura 9.1., també mostra l'índex d'incertesa creat en aquest estudi. Els valors de 0 o propers a 0 indiquen un situació, en l'economia, poc incerta o gens incerta, i a mesura que l'indicador augmenta, la situació esdevé més incerta, per tant aquest indicador presenta una escala diferent a la del S&P500. L'escala d'aquest índex va de 0, que representa el valor mínim, fins a un valor màxim teòric que s'obtindria si un text tingués el valor màxim de TF-IDF, el màxim pes del model i el màxim nombre de paraules, entre tots els textos. Com es pot observar a la figura 9.1, l'índex presenta quatre grans pics: el 1941, 1965, el valor més elevat el 1965 i el 1989. Aquests quatre pics corresponen a grans situacions d'incertesa, sent l'any 1989 el més incert de tots. Cal destacar que quatre anys es situen en valors de 0, indicant una situació amb un grau d'incertesa molt modest. Aquest índex està subjecte als errors de les prediccions generades pel model i a les diferències creades per l'etiquetatge dels textos.

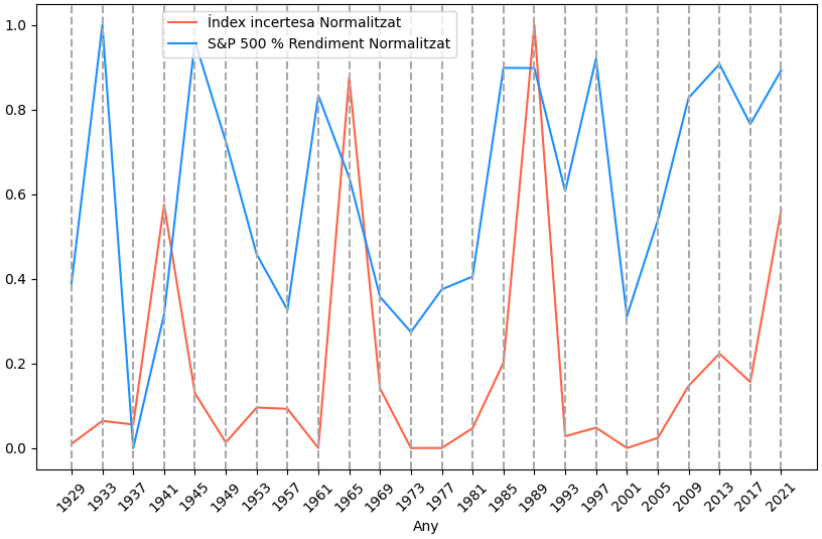
Figura 9.1. Distribució conjunta, Índex d'incertesa i S&P500, des del 1929 fins al 2021 (font pròpia).



Si analitzem les dues sèries conjuntament, es pot identificar diverses semblances i discordances. En primer lloc, és important precisar que les siluetes de les sèries no són coincidents, la qual cosa indica que el nostre indicador, inicialment, no segueix l'S&P500. No obstant això, és necessari adoptar una visió més àmplia, ja que poden existir certes tendències detectades per l'índex. Al llarg de la sèrie, tot i presentar diferents escales, es poden observar dinàmiques semblants. Per exemple, de 1929 a 1937, ambdós gràfics augmenten el seu valor de 1929 a 1933, per després disminuir de 1933 a 1937. Un patró similar es presenta de 1965 a 1985, on ambdós gràfics mostren característiques tendencials semblants. Finalment, de 1993 a 2021, es destaca que ambdues sèries mostren tendències similars, on trobem, des de 2001, tres augments seguits dels dos indicadors, una posterior disminució, per acabar augmentant, de 2017 a 2021. Malgrat aquestes semblances, es poden observar certes incongruències al llarg de la sèrie. Per exemple, de 1941 a 1945, quan l'índex mostra un valor força alt i decau el 1945, en canvi el retorn de l'S&P500 es situa en valors negatius el 1941 i augmenta fins a valors propers al 33%, el 1945

A causa de la diferència d'escales i de les similituds i incongruències entre l'índex i l'S&P500, és necessari normalitzar tant els valors de l'índex com els de l'S&P500 per tal de situar ambdues sèries temporals en una escala similar, facilitant així la seva comparació. Amb l'objectiu d'obtenir una comparació més significativa, s'han transformat les dues sèries. Com a resultat, s'ha obtingut un rang de valors on el valor mínim possible és 0 i el màxim possible és 1.

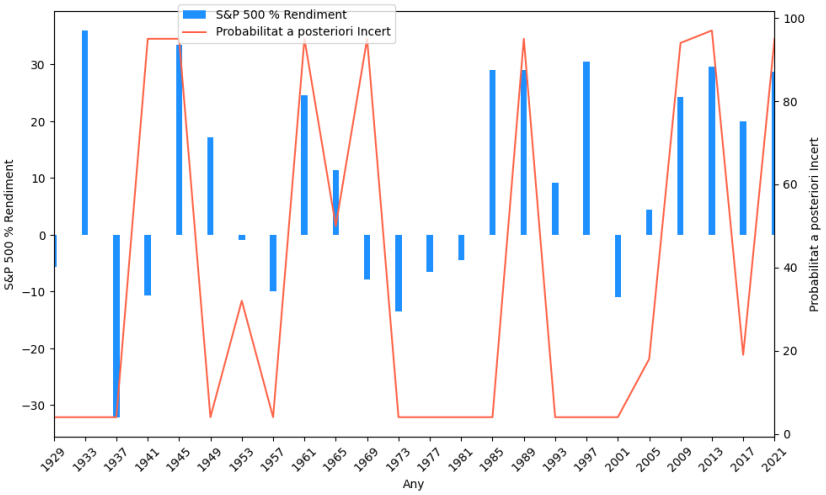
Figura 9.2. Distribució conjunta normalitzada, Índex d'incertesa i S&P500, des del 1929 fins al 2021 (font pròpia).



En la figura 9.2. es poden distingir dues sèries temporals del 1929 al 2021. La línia blava mostra els valors normalitzats del S&P500, mentre que la línia taronja representa els valors normalitzats de l'índex d'incertesa. Cal tenir en compte no la magnitud sinó les tendències alcistes o baixistes dels resultats. Com s'ha indicat en el comentari de la figura 9.1., les tendències de 1929 a 1937 es confirmen, afegint-hi una similitud de tendència alcista de 1937 a 1941. Altres semblances es troben en la caiguda d'ambdós valors de 1945 a 1949, així com la disminució de 1965 a 1973 i el posterior augment de 1977 a 1989. A més, cal destacar les semblances en la caiguda de les tendències de 1989 a 2021, tot i que les magnituds difereixen considerablement. Finalment, cal mencionar que el S&P500 presenta una desviació estàndard de 0.24, mentre que l'índex d'incertesa té una desviació estàndard de 0.27. Es pot intuir que la volatilitat de l'índex ve donada, entre d'altres, pel seu valor màxim el 1989.

Si s'observa la probabilitat a posteriori del model de ser incert, s'obtenen els següents resultats:

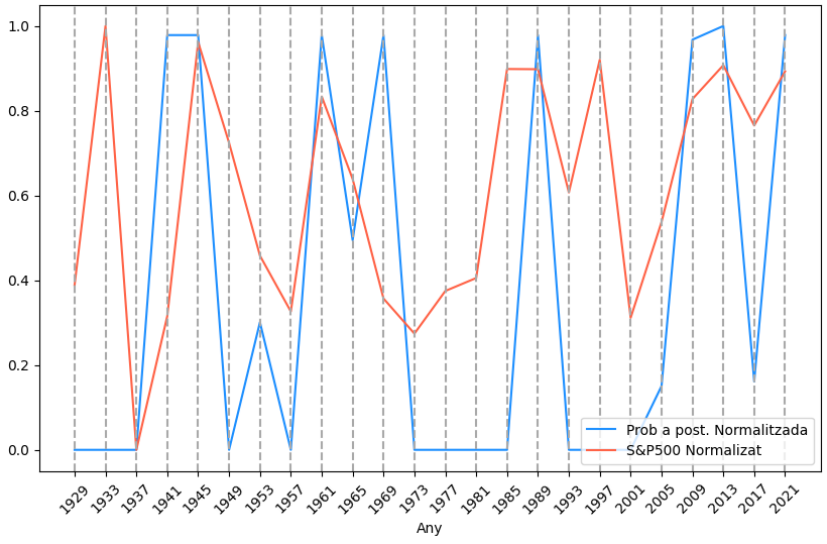
Figura 9.3. Distribució conjunta, Probabilitat a posteriori i S&P500, des del 1929 fins al 2021 (font pròpia).



En el cas de la figura 9.3., s'obté la probabilitat a posteriori de pertànyer a la classe incerta o no incerta, junt amb el percentatge de retorns del S&P500. Per tant, com més alta sigui la probabilitat, més gran serà la probabilitat de situació d'incertesa esperada en l'economia. Si ens fixem en el període de 1937 a 1941, el valor de la probabilitat és gairebé del 0% al principi i augmenta fins a prop del 90%. En el mateix període, l'S&P500 mostra una pujada en el seu percentatge de rendiment. Cal afegir que de 1941 a 1945, la probabilitat es manté elevada, mentre que l'S&P500 continua amb la seva tendència alcista. D'altra banda, el 1945, la probabilitat es troba en el seu màxim de la sèrie, però el 1949 decau fins al 0%. En aquest mateix període, l'S&P500 també decau, passant del 34,40% el 1945 al 17,22% el 1949. Aquesta tendència, no es manté constant al llarg de tota la sèrie, i hi ha certes èpoques en les quals una baixa probabilitat d'incertesa no implica un rendiment percentual més baix de l'S&P500, com per exemple l'any 1973 al 1977.

Com en el cas anterior, és necessari obtenir les distribucions conjuntes de totes dues sèries temporals per tal de tenir una perspectiva més clara de la capacitat del model, així com de la probabilitat com a alternativa de l'índex.

Figura 9.4. Distribució conjunta, Probabilitat a posteriori i S&P500 normalitzats, des del 1929 fins al 2021 (font pròpia).



En la figura 9.4. es mostra la distribució conjunta, destacant, com s'ha comentat anteriorment, certes tendències conjuntes, com les de 1937 a 1945, de 1957 a 1961 i de 2001 a 2021. No obstant això, les diferents probabilitats a posteriori mostren una considerable variabilitat en els resultats. De fet, la desviació estàndard de les sèries és de 0,44, la qual cosa és força elevada tenint en compte que l'escala de la sèrie va de 0 a 1. Tot i que presenta alguns bons resultats en alguns canvis de tendència esmentats anteriorment, no sembla representar adequadament les tendències de la sèrie de l'S&P500.

10. Conclusions

El present estudi ha permès explorar l'anàlisi de textos mitjançant tècniques de Processament del Llenguatge Natural -NLP-, a més de presentar una manera de transformar informació de llenguatge natural a dades numèriques. En primer lloc, s'ha evidenciat la necessitat d'aplicar certs processos, com el *stemming*, per tal de millorar la interpretabilitat de la base de dades. Durant les diferents etapes del NLP, s'ha observat la necessitat imperiosa de reduir la dimensionalitat dels textos per facilitar la classificació rellevant dels models. L'anàlisi ens ha conduït des d'un text convencional fins a una representació numèrica, gràcies a tècniques com el TF-IDF.

En l'aplicació del model, s'ha experimentat la realitat d'aquest tipus d'anàlisi, que exigeix una elecció acurada dels passos del procés. El model, per la seva essència, implica un tractament especial, però la seva potència el fa molt atractiu per l'anàlisi i per futurs estudis. Tal com s'ha demostrat en la fase 5 i 6, apartat 9, corresponent a l'avaluació del model, aquest ha obtingut una precisió del 71% en la validació creuada, un valor acceptable segons l'àmbit d'estudi. No obstant això, presenta errors considerables en relació amb la classe *Uncertainty*, amb un *recall* del 64%, fet que indica que ha errat en la identificació d'un 36% dels textos incerts. Per tant, tot i que el model aconsegueix uns resultats acceptables per a la classe de no incertesa, no ho fa, en el mateix grau per a la classe d'incertesa. Pel que fa al model en conjunt, s'observen bons resultats en una mètrica com el valor AUC. Tot i els resultats del model, cal destacar que aquests no presenten sobre ajustament, una característica que podria ser comuna en aquest tipus de classificacions. Per tant, la conclusió de l'estudi és que el model no ha aconseguit identificar un patró pels textos incerts, cosa que provoca alguns errors en la classificació, sembla que el model no ha acabat de generalitzar adequadament.

Si ens fixem en els resultats de l'objectiu principal d'aquest treball, que és la creació de l'índex d'incertesa, i referent a la hipòtesi principal, no la podem acceptar com a certa, donats els resultats de la fase 7, apartat 8. Tot i que s'han trobat certes tendències coincidents, no s'ha pogut demostrar una relació significativa entre ambdues sèries temporals. Es creu que aquestes coincidències generen un indici de relació, la qual cosa no implica que hi hagi una relació definida, sinó que incrementa les sospites que, amb metodologies més complexes, es podria arribar a demostrar. Aquest fet, però, no necessàriament contradiu la hipòtesi principal, ja que altres factors poden haver influït en aquests resultats. No es descarta que l'ús d'altres tècniques en certs punts del treball hagués produït uns resultats més alineats amb els esperats. Per tant, tot i que els resultats no permeten acceptar la hipòtesi principal de l'anàlisi, obren la porta a dubtes sobre les possibles millores del present treball, sempre tenint en compte que l'objectiu d'aquest, així com l'anàlisi de texts en el present treball, parteix d'un concepte no mesurable: la incertesa. A més a més, és necessari establir un punt de comparació, especialment amb l'article de referència per a la present anàlisi. Com s'ha comentat en diversos punts de l'estudi, aquest és l'estudi d'Avela i Lehmus. Estudis similars, com el mencionat, utilitzen textos més curts i senzills en la seva dimensionalitat, com ara titulars de notícies, entre d'altres. És per això que un text més llarg presenta certes dificultats afegides que cal tenir en compte. En conseqüència, l'ambició del treball genera més complicacions en els punts de la creació de l'índex.

Arribats a aquest punt i malgrat els resultats obtinguts, les línies de treball continuen obertes. Les conclusions obtingudes no indiquen la fi del projecte un cop s'escrigui l'última paraula, sinó que aquest pot créixer i ser modificat per tal d'obtenir més explicacions i resultats. És ben conegut, que el camp de l'anàlisi de textos, està força desenvolupat i que grans corporacions utilitzen aquestes tècniques per a objectius ambiciosos. Per exemple, es podrien

aplicar tècniques de *deep learning* com BERT, *Bidirectional Encoder Representations from Transformers*. Aquest està dissenyat per pre-entrenar representacions bidireccionals profundes a partir de text no etiquetat condicionat conjuntament en tots els contextos (Devlin et al., 2018:4171). D'altra banda, es podrien aplicar tècniques com *Generative Pre-trained Transformer* -GPT-, que aprèn patrons estadístics del llenguatge natural i genera llenguatge similar al de l'ésser humà (Saka et al., 2023:2). Aquestes són només algunes de les possibles aplicacions dels NLP que es podrien desenvolupar en futurs estudis. Altres línies d'investigació inclouen el *clustering* de textos, així com altres models d'aprenentatge automàtic com el TWCNB, mostrat en l'estudi d'Avela i Lehmus, o models de *bagging* i altres models de *boosting*.

En resum, aquesta anàlisi exposa una fracció de les possibilitats que ofereix l'anàlisi de textos i les investigacions que es podrien dur a terme, així com demostra l'ús potencial del grau d'incertesa com a indicador. Tot i això, com s'ha mencionat prèviament, a causa de la complexitat de l'estudi, es presenten futures línies d'investigació.

11. Bibliografia

Alattar, R. A. S. (2014). A speech act analysis of American presidential speeches. *Arts Journal*, *110*, 1-39.

Accelerating progress since 1860. (n.d.). Spglobal. https://www.spglobal.com/en/our-history#second

AdaBoostClassifier. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

Agencia de Marketing Online BlackBeast. (2019, July 8). ¿Qué son las Stop Words y para qué sirven? – BlackBeast. Blackbeast.pro. https://blackbeast.pro/diccionario/stop-words/

Altgeld, J. P. (1901). Oratory: Its Requirements and Its Rewards. CH Kerr.

Article II. (n.d.). LII / Legal Information Institute. https://www.law.cornell.edu/constitution/articleii

Atiq, A., Abeed, A. S., Efat, A. A., & Momin, A. (2022). Sentimental analysis on political speeches (Doctoral dissertation, Brac University).

Avela, A., & Lehmus, M. (2020). It's in the News: Developing a Real Time Index for Economic Uncertainty Based on Finnish News Titles (No. 84). ETLA Working Papers.

Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: A comparison of retrieval performances.

Beers, B. (2023). Why Do Investors Use the S&P 500 as a Benchmark? Investopedia. https://www.investopedia.com/ask/answers/041315/what-are-pros-and-cons-using-sp-500-benchmark.asp

Bernanke, B. S. (1983). Irreversibility, Uncertainty, and Cyclical Investment. The Quarterly Journal of Economics, 98(1), 85–106. https://doi.org/10.2307/1885568

Bloehdorn, S., & Hotho, A. (2004, August). Boosting for text classification with semantic features. In International workshop on knowledge discovery on the web (pp. 149-166). Berlin, Heidelberg: Springer Berlin Heidelberg.

Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python.

Calvo-González, O., Eizmendi, A., & Reyes, G. (2022). The Shifting Attention of Political Leaders: Evidence from Two Centuries of Presidential Speeches. arXiv preprint arXiv:2209.00540.

Cambridge English Dictionary: Meanings & Definitions. (2024). https://dictionary.cambridge.org/dictionary/english

Chaudhary, M. (2021). TF-IDF Vectorizer scikit-learn - Mukesh Chaudhary - Medium. *Medium*. https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

Chengsheng, T. U., Bing, X. U., & Huacheng, L. I. U. (2018). The application of the AdaBoost algorithm in the text classification. In 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (pp. 1792-1796). IEEE.

Cronología de los acontecimientos internacionales. (n.d.). https://embamex.sre.gob.mx/polonia/index.php/es/la-historia-de-los-ninos-de-santa-rosa/16-sin-categoria/54-cronologia-de-los-acontecimientos-internacionales

Componentes Electrónicas LTDA. (2022). Qué es MATLAB? - MATLAB y Simulink - Componentes Electrónicas. https://www.compelect.com.co/que-es-matlab/

Confusion_matrix. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

Data processing noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com. (n.d.). https://www.oxfordlearnersdictionaries.com/definition/english/data-processing

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dönmez, D., & Grote, G. (2018). Two sides of the same coin–how agile software development teams approach uncertainty as threats and opportunities. Information and Software Technology, 93, 94-111.

Farzi, R., & Bolandi, V. (2016). Estimation of organic facies using ensemble methods in comparison with conventional intelligent approaches: A case study of the South Pars Gas Field, Persian Gulf, Iran. Modeling Earth Systems and Environment, 2, 1-13.

Freund, Y. (1992, July). An improved boosting algorithm and its implications on learning complexity. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 391-398).

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In *icml* (Vol. 96, pp. 148-156).

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Special invited paper. additive logistic regression: A statistical view of boosting. Annals of statistics, 337-374.

Gao, R., & Liu, Z. (2020). An improved adaboost algorithm for hyperparameter optimization. In Journal of Physics: Conference Series (Vol. 1631, No. 1, p. 012048). IOP Publishing.

GeeksforGeeks. (2023). Cross validation in machine learning. GeeksforGeeks. https://www.geeksforgeeks.org/cross-validation-machine-learning/

GeeksforGeeks. (2024). Difference between Supervised and Unsupervised Learning. GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-supervised-and-unsupervised-learning/

GeeksforGeeks. (2024). Learning model building in SciKit-Learn. GeeksforGeeks. https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/#what-is-scikitlearn

GeeksforGeeks. (2022). Snowball stemmer NLP. GeeksforGeeks. https://www.geeksforgeeks.org/snowball-stemmer-nlp/ Goff, J. (2011, August 5).

S&P seen surrendering to Tea Party with credit downgrade - InvestmentNews. InvestmentNews. https://www.investmentnews.com/industry-news/features/sp-seen-surrendering-to-tea-party-with-credit-downgrade-38048

Goldsmith, D. J. (2001). A normative approach to the study of uncertainty and communication. Journal of communication, 51(3), 514-533.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, *143*(1), 29-36.

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., ... & Asadi, H. (2019). Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. American Journal of Roentgenology, *212*(1), 38-43.

Hillson, D. (2003). Effective opportunity management for projects: Exploiting positive risk. Crc Press.

House, W. (2021, January 21). *Inaugural Address by President Joseph R. Biden, Jr.* The White House. https://www.whitehouse.gov/briefing-room/speeches-remarks/2021/01/20/inaugural-address-by-president-joseph-r-biden-jr/

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. WSEAS transactions on computers, 4(8), 966-974.

Jairi, I. (2023, April 17). Gini Gain vs Gini Impurity | Decision Tree — A Simple Explanation. *Medium*. https://medium.com/@jairiidriss/gini-gain-vs-gini-impurity-decision-tree-a-simple-explanation-a24ebfeebee9

Kheiri, K., & Karimi, H. (2023). Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. arXiv preprint arXiv:2307.10234.

Klein, C., & Klein, C. (2023). *Why does Inauguration Day fall on January 20?* HISTORY. https://www.history.com/news/why-does-inauguration-day-fall-on-january-20

Knight, F. H. (1921). Risk, uncertainty and profit (Vol. 31). Houghton Mifflin.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

Kurzyński, M. W. (1983). The optimal strategy of a tree classifier. *Pattern Recognition*, *16*(1), 81-87.

Lara, V. (2021). ¿Cuántos libros existen en el mundo? Hipertextual. https://hipertextual.com/2017/04/cuantos-libros-existen-mundo

Lasswell, H. D. (2006). The language of power. Political linguistics. Yekaterinburg, (20).

Loading. . . (n.d.). https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#sklearn.ensemble.AdaBoostClassifier.predict_proba

Maddukuri, H. (2022). Introduction to SciKit Learn with a practical example in Python. *Medium*. https://medium.com/python-pandemonium/introduction-to-scikit-learn-with-a-practical-example-in-python-efb037993be0

Matsko, L. (2003). Rytoryka (Rhetoric, in Ukrainian). Kyiv: Vyshcha shkola.

Mihalcea, R., Liu, H., Lieberman, H. (2006). NLP (Natural Language Processing) for NLP (Natural Language Programming). In: Gelbukh, A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2006. Lecture Notes in Computer Science, vol 3878. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11671299_34

Muralidharan, K. (2010). A note on transformation, standardization and normalization. *IUP J. Oper. Manag*, *9*, 116-122.

Mullen, T., & Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 412-418).

Nguyen, K. (2024). N-gram language models - MTI Technology - Medium. *Medium*. https://medium.com/mti-technology/n-gram-language-model-b7c2fc322799

Nosetto, L. (2017). Las palabras y las cosas. Michel Foucault y la centralidad de la cuestión del origen en los discursos políticos. *Nómadas. Critical Journal of Social and Juridical Sciences*, *51*(2).

Novack, J. (2013). Political uncertainty will continue to stunt economic growth. Forbes. https://www.forbes.com/sites/janetnovack/2013/10/16/its-the-washington-uncertainty-stupid/?sh=74d74f832bf7

Roa, M. M. (2021b). ¿Cuántos sitios web hay en el mundo? *Statista Daily Data*. https://es.statista.com/grafico/19107/numero-de-sitios-web-existentes-en-internet/

Medvid, O., Vashyst, K., Sushkova, O., Sadivnychyi, V., Malovana, N., & Shumenko, O. (2022). Us presidents'political speeches as a means of manipulation in 21st century society. *Wisdom*, (2S (3)), 144-156.

Onion, A. (2023). New deal - programs, social security & FDR. *HISTORY*. https://www.history.com/topics/great-depression/new-deal

Oxford reference. (n.d.). Oxford. https://www.oxfordreference.com/display/10.1093/oi/authority.20110803100238563

Pandas - Python Data Analysis Library. (n.d.). https://pandas.pydata.org/about/

Parmar, M., & Tiwari, A. (2024). Enhancing Text Classification Performance using Stacking Ensemble Method with TF-IDF Feature Extraction. In 2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI) (pp. 166-174). IEEE.

Pástor, Ľ., & Veronesi, P. (2013). Political uncertainty and risk premia. Journal of financial Economics, 110(3), 520-545.

Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering, 6(10), 74-78.

Pepe, M., Longton, G. M., & Janes, H. Estimation and Comparison of Receiver Operating Characteristic Curves 2008.

Prasanna, C. (2024). Classification Report Explained — Precision, Recall, Accuracy, Macro average, and Weighted Average. *Medium*. https://medium.com/@chanakapinfo/classification-report-explained-precision-recall-accuracy-macro-average-and-weighted-average-8cd358ee2f8a

Priyam, A., Abhijeeta, G. R., Rathee, A., & Srivastava, S. (2013). Comparative analysis of decision tree classification algorithms. *International Journal of current engineering and technology*, *3*(2), 334-337.

Pyplot tutorial — Matplotlib 3.9.0 documentation. (n.d.). https://matplotlib.org/stable/tutorials/pyplot.html

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. Scientific Reports, 14(1), 6086.

Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In Proceedings of the 20th international conference on machine learning (ICML-03) (pp. 616-623)., 14

Rutkowski, L., Jaworski, M., Pietruczuk, L., & Duda, P. (2014). The CART decision tree for mining data streams. Information Sciences, 266, 1-15.

Rus, V., McCarthy, P. M., & Graesser, A. C. (2006, February). Analysis of a textual entailer. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 287-298). Berlin, Heidelberg: Springer Berlin Heidelberg.

Saka, A., Taiwo, R., Saka, N., Salami, B. A., Ajayi, S., Akande, K., & Kazemi, H. (2023). GPT models in construction industry: Opportunities, limitations, and a use case validation. Developments in the Built Environment, 100300.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.

Schapire, R. E. (1990). The strength of weak learnability. Machine learning, 5, 197-227.

Sala, À. (2023, October 27). Pánico en Wall Street. El Crack de 1929. historia.nationalgeographic.com.es. https://historia.nationalgeographic.com.es/a/crack-1929-imagenes_14851

Santafe, G., Inza, I., & Lozano, J. A. (2015). Dealing with the evaluation of supervised classification algorithms. Artificial Intelligence Review, 44, 467-508.

Schaffer, C. (1993). Selecting a classification method by cross-validation. Machine learning, *13*, 135-143.

Sharma, B., & Singh, P. (2022). An improved anti-phishing model utilizing TF-IDF and AdaBoost. Concurrency and Computation: Practice and Experience, 34(26), e7287.

Shi, Y., Yang, K., Yang, Z., & Zhou, Y. (2022). Convex optimization. In Elsevier eBooks (pp. 37–55). https://doi.org/10.1016/b978-0-12-823817-2.00012-7

Sign Function (Signum): definition, examples. (n.d.). Statistics How To. https://www.statisticshowto.com/sign-function/

Sreekrishnan, R. (2023, May 18). Demystifying K-Fold Cross Validation - Ramprabhu Sreekrishnan - Medium. Medium. https://medium.com/@prabhucs01/demystifying-k-fold-cross-validation-3c9d410adddd

S&P 500 Total returns. (2024, June). Slickcharts. https://www.slickcharts.com/sp500/returns

Synonyms and Antonyms of Words | Thesaurus.com. (2020b, September 15). Thesaurus.com. https://www.thesaurus.com/browse/uncertainty

Tabassum, A., & Patil, R. R. (2020). A survey on text pre-processing & feature extraction techniques in natural language processing. International Research Journal of Engineering and Technology (IRJET), 7(06), 4864-4867.

Train_test_split. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

What are DataFrames? (n.d.). Databricks. https://www.databricks.com/glossary/what-are-dataframes

White House Historical Association. (n.d). Presidential Inaugurations: The inaugural address. WHHA (en-US). https://www.whitehousehistory.org/presidential-inaugurations-the-inaugural-address

Wiese, A., Ho, V., & Hill, E. (2011,. A comparison of stemmers on source code identifiers for software search. In 2011 27th IEEE International Conference on Software Maintenance (ICSM) (pp. 496-499). IEEE.

Wong, T. T., & Yeh, P. Y. (2019). Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1586-1594.

Yang, Y., & Pedersen, J. O. (1997, July). A comparative study on feature selection in text categorization. In *Icml* (Vol. 97, No. 412-420, p. 35).

Yun-tao, Z., Ling, G., & Yong-cheng, W. (2005). An improved TF-IDF approach for text classification. *Journal of Zhejiang University-Science A*, *6*(1), 49-55.

Zhu, W., Zeng, N., & Wang, N. (2010). Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, *19*, 67.

12. Annexos

12.1. WordCloud abans del processament

Figura 12.1. *Wordcloud* discurs de Franklin D. Roosevelt, 1933 (font pròpia).

Figura 12.2. *Wordcloud* discurs de Franklin D. Roosevelt, 1937 (font pròpia).

Figura 12.3. *Wordcloud* discurs de Franklin D. Roosevelt, 1941 (font pròpia).

Figura 12.4. *Wordcloud* discurs de Franklin D. Roosevelt, 1945 (font pròpia).

Figura 12.5. *Wordcloud* discurs de Harry S. Truman, 1949 (font pròpia).

Figura 12.6. *Wordcloud* discurs de Dwight D. Eisenhower, 1953 (font pròpia).

Figura 12.7. *Wordcloud* discurs de Dwight D. Eisenhower, 1957 (font pròpia).

12.2. StopWords

"a", "about", "above", "after", "again", "against", "ain", "all", "am", "an", "and", "any", "are", "aren", "aren't", "as", "at", "be", "because", "been", "before", "being", "below", "between", "both", "but", "by", "can", "couldn", "couldn't", "d", "did", "didn", "didn't", "do", "does", "doesn", "doesn't", "doing", "don", "don't", "down", "during", "each", "few", "for", "from", "further", "had", "hadn", "hadn't", "has", "hasn", "hasn't", "have", "haven", "haven't", "having", "he", "her", "here", "hers", "herself", "him", "himself", "his", "how", "i", "if", "in", "into", "is", "isn", "isn't", "it", "it's", "its", "itself", "just", "ll", "m", "ma", "me", "mightn", "mightn't", "more", "most", "mustn", "mustn't", "my", "myself", "needn", "needn't", "no", "nor", "not", "now", "o", "of", "off", "on", "once", "only", "or", "other", "our", "ours", "ourselves", "out", "over", "own", "re", "s", "same", "shan", "shan't", "she", "she's", "should", "should've", "shouldn", "shouldn't", "so", "some", "such", "t", "than", "that", "that'll", "the", "their", "theirs", "them", "themselves", "then", "there", "these", "they", "this", "those", "through", "to", "too", "under", "until", "up", "ve", "very", "was", "wasn", "wasn't", "we", "were", "weren", "weren't", "what", "when", "where", "which", "while", "who", "whom", "why", "will", "with", "won", "won't", "wouldn", "wouldn't", "y", "you", "you'd", "you'll", "you're", "you've", "your", "yours", "yourself", "yourselves"

12.3. WordCloud després del processament

Figura 12.8. *Wordcloud* discurs de Franklin D. Roosevelt, 1933 (font pròpia).

Figura 12.9. *Wordcloud* discurs de Franklin D. Roosevelt, 1937 (font pròpia).

Figura 12.10. *Wordcloud* discurs de Franklin D. Roosevelt, 1941 (font pròpia).

Figura 12.11. *Wordcloud* discurs de Franklin D. Roosevelt, 1945 (font pròpia).

Figura 12.12. *Wordcloud* discurs de Harry S. Truman, 1949 (font pròpia).

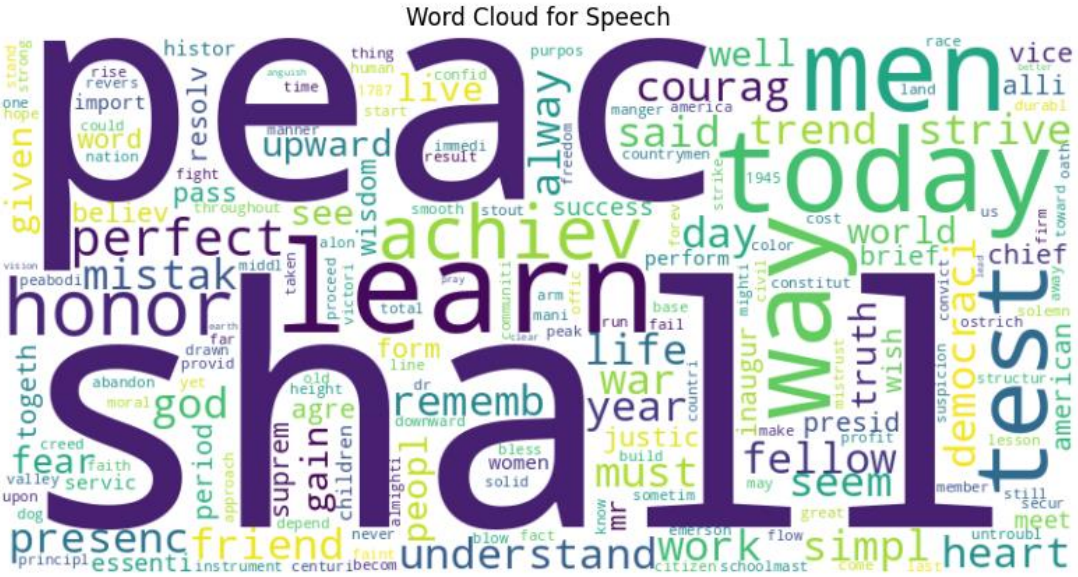


Figura 12.13. *Wordcloud* discurs de Dwight D. Eisenhower, 1953 (font pròpia).

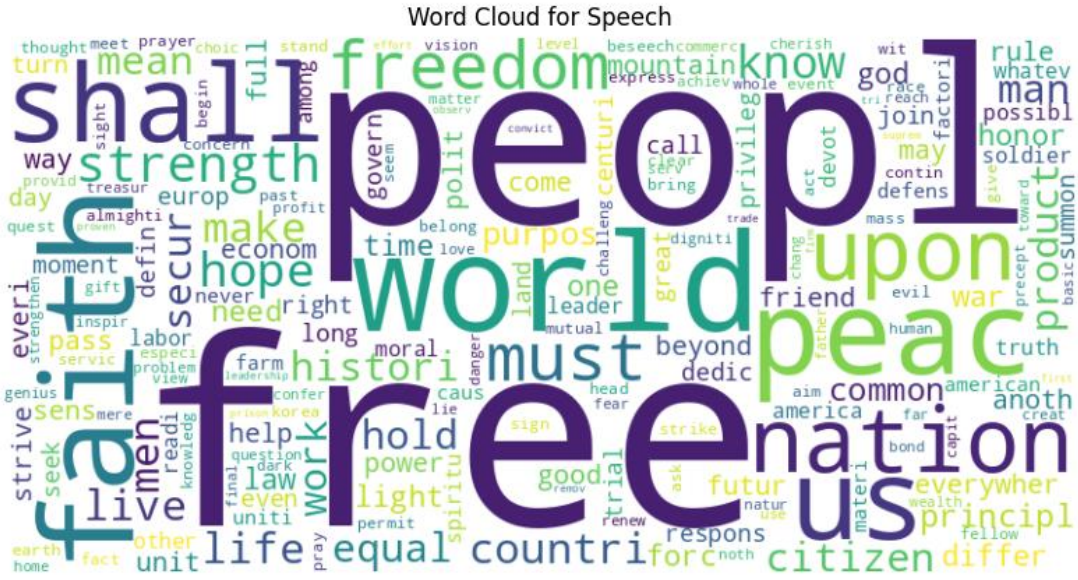


Figura 12.14. *Wordcloud* discurs de Dwight D. Eisenhower, 1957 (font pròpia).

12.4. Sinònims d'incertesa

"ambiguity", "ambivalence", "anxiety", "concern", "confusion", "doubt",

"distrust", "mistrust", "skepticism", "suspicion", "trouble","uncertainty", "uneasiness",

"unpredictability", "worry", "maybe", "possibly", "might", "could",

"unsure", "uncertain", "likely", "unlikely", "perhaps", "not sure",

"bewilderment", "conjecture", "contingency", "dilemma", "disquiet",

"doubtfulness", "dubiety", "guesswork", "hesitancy", "hesitation",

"incertitude", "inconclusiveness", "indecision", "irresolution",

"insecurity", "misgiving", "mystification", "oscillation", "perplexity",

"puzzle", "puzzlement", "qualm", "quandary", "query", "reserve",

"scruple", "vagueness", "wonder"

12.5. Codi

```
import nltk
nltk.download()
from nltk.book import *
import os

"""### Carraguem els 59 textos"""

import pandas as pd

# File IDs
file_ids = inaugural.fileids()

speeches = file_ids

transcripts = pd.DataFrame()

def get_transcripts(file_ids, transcripts):
  transcripts = []
  for file_id in speeches:
      # Treure els noms dels presidents i l'any
      t_year, t_president = file_id.split('-')
      t_president = t_president.rstrip('.txt')  # Treure txtx.

      t_content = ' '.join(inaugural.words(file_id))

      record = {
          'president': t_president,
```

```
            'year': t_year,
            'speeches': t_content
        }
        transcripts.append(record)
    return pd.DataFrame(transcripts)

data = get_transcripts(file_ids, transcripts)
data.to_csv("us_presidents_transcripts.csv", sep="|")
speeches = data
speeches.reset_index(drop=True, inplace=True)
speeches.index = speeches.index + 1
speeches[35:59]

speeches.at[58, "speeches"]

"""### Descriptiva Preprocesament"""

speeches['num_characters'] = speeches['speeches'].apply(lambda speech: len(speech))
def count_words(text):
    return len(text.split())
speeches['num_paraules'] = speeches['speeches'].apply(count_words)
speeches['blank spaces'] = speeches['num_characters']- speeches['num_paraules']
#speeches['unique_words'] = speeches['speeches'].apply(lambda speech: len(set(speech)))
#speeches['distinct_words'] = speeches['speeches'].apply(lambda speech:
round(len(speech)/len(set(speech)),2))

last = speeches.tail(24)

import matplotlib.pyplot as plt
```

```
import pandas as pd
plt.figure(figsize=(15, 6))

plt.plot(last["year"], last['num_characters'], label='Nombre de caràcters', color='cadetblue')
#plt.plot(last["year"], last['num_paraules'], label='Nombre de paraules', color='salmon')
#plt.plot(last["year"], last['unique_words'], label='Freq', color='orchid')
#plt.plot(last["year"], last['distinct_words'], label='Freq', color='mediumaquamarine')



#for year in speeches["year"]:
   # plt.axvline(x=year, color='red', linestyle='--')

plt.xticks(last["year"], rotation=45)
plt.legend()
plt.grid(True)

print("El nombre de caràcteres màxim es:", last['num_characters'].max(), "El nombre de caràcteres mínim es:", last['num_characters'].min())

mean_length = last['num_characters'].mean()
median_length = last['num_characters'].median()
mode_length = last['num_characters'].mode()[0]
range_length = last['num_characters'].max() - last['num_characters'].min()
variance_length = round(last['num_characters'].var(), 2)
std_dev_length = round(last['num_characters'].std(), 2)
Q1 = last['num_characters'].quantile(0.25)
Q3 = last['num_characters'].quantile(0.75)
IQR = Q3 - Q1
min_length = last['num_characters'].min()
```

```
max_length = last['num_characters'].max()


# Per crear un data.frame
results = {
    'Measure': [
        'Mean Length',
        'Median Length',
        'Mode Length',
        'Range',
        'Variance',
        'Standard Deviation',
        'Q1 (25th percentile)',
        'Q3 (75th percentile)',
        'Interquartile Range (IQR)',
        'Minimum Length',
        'Maximum Length'
    ],
    'Value': [
        mean_length,
        median_length,
        mode_length,
        range_length,
        variance_length,
        std_dev_length,
        Q1,
        Q3,
        IQR,
        min_length,
        max_length
```

```
    ]
}

results_df = pd.DataFrame(results)
results_df

import pandas as pd

def count_unique_words(text):
    words = text.lower().split()
    unique_words = set(words)
    return len(unique_words)

speeches['unique_words'] = speeches['speeches'].apply(count_unique_words)
speeches['unique_words'] = (speeches['unique_words']/speeches['num_paraules'])*100
#speeches['unique_words'] = speeches['speeches'].apply(unique_words_frequency)
#speeches['distinct_words'] = speeches['speeches'].apply(lambda speech:
round(len(speech)/len(set(speech)),2))

last = speeches.tail(24)

import matplotlib.pyplot as plt
import pandas as pd
plt.figure(figsize=(15, 6))

#plt.plot(last["year"], last['num_characters'], label='Nombre de caràcters', color='cadetblue')
#plt.plot(last["year"], last['num_paraules'], label='Nombre de paraules', color='salmon')
plt.plot(last["year"], last['unique_words'], label='Freq Paraules úniques', color='salmon')
#plt.plot(last["year"], last['distinct_words'], label='Freq', color='mediumaquamarine')
```

```
#for year in speeches["year"]:
   # plt.axvline(x=year, color='red', linestyle='--')

plt.xticks(last["year"], rotation=45)
plt.legend()
plt.grid(True)

from collections import Counter
import pandas as pd
from nltk.corpus import wordnet
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import random
import numpy as np

random.seed(10)
np.random.seed(10)

def count_unique_words(text):
    words = text.lower().split()
    return words

def generate_word_cloud(speech):
    # Word frequencies
    tokens = count_unique_words(speech)
    word_freq = Counter(tokens)

    wordcloud = WordCloud(width=800, height=400, background_color ='white',
```

```
random_state=10).generate_from_frequencies(word_freq)

    # Word cloud
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title('Word Cloud for Speech')
    plt.show()

#generate_word_cloud(last.iloc[7]["speeches"])
#for speech in last['speeches']:
    #generate_word_cloud(speech)

from textblob import TextBlob

def sentiment_analysis(speech):
    analysis = TextBlob(speech)
    return analysis.sentiment.polarity

initial_sentiments = [sentiment_analysis(speech) for speech in last["speeches"]]
initial_sentiments

"""### Treiem els signes de puntuació"""

import string

# Puntuació a treure
punct = string.punctuation
```

```
# Treiem la puntuació de tots els textos
speeches['speeches'] = speeches['speeches'].str.replace('['+punct+']', '', regex=True)

# Treiem el càracter que apareix en el speech de l'Obama
speeches['speeches'] = speeches['speeches'].str.replace('\x80|\x94', '', regex=True)

# El mateix per la â
speeches['speeches'] = speeches['speeches'].str.replace('â', '', regex=True)

chars_to_replace = ['¡', '¡¦', '¡§', '¡¨', '¡¨¡', '¦', '§', '¨', '\n', '\n']

# Canviem els caràcters per cada text
for char in chars_to_replace:
    speeches['speeches'] = speeches['speeches'].str.replace(char, '')



speeches['speeches'] = speeches['speeches'].str.lower()
speeches

"""### Word Segmentation"""

nltk.download('punkt')

speeches['speeches'] = speeches['speeches'].str.replace('-', '_')

# Tokenitzacio
speeches['speeches'] = speeches['speeches'].apply(nltk.word_tokenize)

# Canvis de guines _, -
```

```
speeches['speeches'] = speeches['speeches'].apply(lambda x: [word.replace('_', '-') for word in x])


speeches


"""### StopWords"""


from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize


stopwords_sp = set(stopwords.words('english')) #definim stop words

lower_speech=[]


speeches['speeches'] = speeches['speeches'].apply(lambda speech: [word.lower() for word in speech if word.lower() not in stopwords_sp])


speeches


from nltk.stem import SnowballStemmer

stemm = SnowballStemmer('english')

speeches['speeches'] = speeches['speeches'].apply(lambda speech: [stemm.stem(word) for word in speech])

speeches


"""### Descriptivo pos preprocessing"""


def join_words(word_list):

    return " ".join(word_list)


speeches['joined_words'] = speeches['speeches'].apply(join_words)
```

```
speeches['num_characters_dp'] = speeches['joined_words'].apply(lambda speech:
len(speech))


def count_words(text):
    return len(text.split())


speeches['num_paraules_dp'] = speeches['joined_words'].apply(count_words)
speeches['blank spaces_dp'] = speeches['num_characters_dp']- speeches['num_paraules_dp']
#speeches['unique_words'] = speeches['speeches'].apply(lambda speech: len(set(speech)))
#speeches['distinct_words'] = speeches['speeches'].apply(lambda speech:
round(len(speech)/len(set(speech)),2))


last = speeches.tail(24)


import matplotlib.pyplot as plt
import pandas as pd
plt.figure(figsize=(15, 6))


plt.plot(last["year"], last['num_characters_dp'], label='Nombre de caràcters',
color='cadetblue')
#plt.plot(last["year"], last['num_paraules'], label='Nombre de paraules', color='salmon')
#plt.plot(last["year"], last['unique_words'], label='Freq', color='orchid')
#plt.plot(last["year"], last['distinct_words'], label='Freq', color='mediumaquamarine')



#for year in speeches["year"]:
  # plt.axvline(x=year, color='red', linestyle='--')


plt.xticks(last["year"], rotation=45)
```

```
plt.legend()
plt.grid(True)

mean_length = last['num_characters_dp'].mean()
median_length = last['num_characters_dp'].median()
mode_length = last['num_characters_dp'].mode()[0]
range_length = last['num_characters_dp'].max() - last['num_characters_dp'].min()
variance_length = round(last['num_characters_dp'].var(), 2)
std_dev_length = round(last['num_characters_dp'].std(), 2)
Q1 = last['num_characters_dp'].quantile(0.25)
Q3 = last['num_characters_dp'].quantile(0.75)
IQR = Q3 - Q1
min_length = last['num_characters_dp'].min()
max_length = last['num_characters_dp'].max()

results = {
    'Measure': [
        'Mean Length',
        'Median Length',
        'Mode Length',
        'Range',
        'Variance',
        'Standard Deviation',
        'Q1 (25th percentile)',
        'Q3 (75th percentile)',
        'Interquartile Range (IQR)',
        'Minimum Length',
        'Maximum Length'
    ],
```

```
    'Value': [
        mean_length,
        median_length,
        mode_length,
        range_length,
        variance_length,
        std_dev_length,
        Q1,
        Q3,
        IQR,
        min_length,
        max_length
    ]
}

results_df = pd.DataFrame(results)
results_df

import pandas as pd

def count_unique_words(text):
    words = text.lower().split()
    unique_words = set(words)
    return len(unique_words)

speeches['unique_words_dp'] = speeches['joined_words'].apply(count_unique_words)
speeches['unique_words_dp'] =
(speeches['unique_words_dp']/speeches['num_paraules_dp'])*100
#speeches['unique_words'] = speeches['speeches'].apply(unique_words_frequency)
```

```
#speeches['distinct_words'] = speeches['speeches'].apply(lambda speech:
round(len(speech)/len(set(speech)),2))


last = speeches.tail(24)


import matplotlib.pyplot as plt

import pandas as pd

plt.figure(figsize=(15, 6))


#plt.plot(last["year"], last['num_characters'], label='Nombre de caràcters', color='cadetblue')

#plt.plot(last["year"], last['num_paraules'], label='Nombre de paraules', color='salmon')

plt.plot(last["year"], last['unique_words_dp'], label='Freq Paraules úniques', color='salmon')

#plt.plot(last["year"], last['distinct_words'], label='Freq', color='mediumaquamarine')


#for year in speeches["year"]:

   # plt.axvline(x=year, color='red', linestyle='--')


plt.xticks(last["year"], rotation=45)

plt.legend()

plt.grid(True)


from collections import Counter

import pandas as pd

from nltk.corpus import wordnet

from wordcloud import WordCloud

import matplotlib.pyplot as plt

import random

import numpy as np
```

```
random.seed(42)
np.random.seed(42)

def count_unique_words(text):
    if isinstance(text, list):
        text = ' '.join(text)
    words = text.lower().split()
    return words

def generate_word_cloud(speech):
    tokens = count_unique_words(speech)
    word_freq = Counter(tokens)

    wordcloud = WordCloud(width=800, height=400, background_color ='white',
random_state=42).generate_from_frequencies(word_freq)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title('Word Cloud for Speech')
    plt.show()

#generate_word_cloud(last.iloc[5]["speeches"])
#Generate word cloud for each speech
#for speech in last['joined_words']:
    #generate_word_cloud(speech)

from collections import Counter
```

```
all_tokens = [token for tokens in speeches['speeches'] for token in tokens]

word_freq = Counter(all_tokens)
import matplotlib.pyplot as plt

most_common_words = word_freq.most_common(10)

words, frequencies = zip(*most_common_words)

plt.figure(figsize=(10, 5))
plt.bar(words, frequencies, color='skyblue')
plt.xlabel('Words')
plt.ylabel('Frequencies')
plt.title('Most Frequent Words in Speeches')
plt.xticks(rotation=45)
plt.show()

import gensim
from gensim import corpora

dictionary = corpora.Dictionary(speeches["speeches"])
dictionary.filter_extremes(no_above=0.9)



# Converting list of documents (corpus) into Document Term Matrix using dictionary
prepared above.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in speeches["speeches"]]
Lda = gensim.models.ldamodel.LdaModel
```

```
# Running and Trainign LDA model on the document term matrix.
ldamodel = Lda(doc_term_matrix, num_topics=2, id2word = dictionary, passes=50)

print(ldamodel.print_topics(num_topics=3, num_words=3))

"""Ensemble learning"""

import pandas as pd
from nltk.corpus import wordnet
import matplotlib.pyplot as plt

uncertainty_words = ["ambiguity", "ambivalence", "anxiety", "concern", "confusion",
"doubt",
                     "distrust", "mistrust", "skepticism", "suspicion", "trouble", "uncertainty",
                     "uneasiness","unpredictability", "worry", "maybe", "possibly", "might", "could",
                     "unsure", "uncertain", "likely", "unlikely", "perhaps", "not sure",
                     "bewilderment", "conjecture", "contingency", "dilemma", "disquiet",
                     "doubtfulness", "dubiety", "guesswork", "hesitancy", "hesitation",
                     "incertitude", "inconclusiveness", "indecision", "irresolution",
                     "insecurity", "misgiving", "mystification", "oscillation", "perplexity",
                     "puzzle", "puzzlement", "qualm", "quandary", "query", "reserve",
                     "scruple", "vagueness", "wonder"]

stemmer = SnowballStemmer('english')
uncertainty_words = [stemmer.stem(word) for word in uncertainty_words]
# Stemming per les uncertainty words
#uncertainty_words = [stemmer.stem(word) for word in uncertainty_words]
uncertainty_keywords = uncertainty_words
```

```
def has_high_uncertainty(text):
    words=text

    # Paraules totals i paraules incertes totals
    total_words = len(words)
    uncertainty_words = sum(1 for word in words if word in uncertainty_keywords)
    #certainty_words = sum(1 for word in words if word in certainty_keywords)

    # Percentage of uncertainty words
    percentage = uncertainty_words / total_words
    return percentage
    #if percentage > threshold:
    #    return "uncertainty"
    #else:
     #   return "certainty"
def label_uncertainty(number, threshold):
    if number > threshold:
        return 'uncertainty'
    else:
        return 'not uncertainty'
import pandas as pd
import numpy as np


#speeches['uncertainty_words'], speeches['certainty_words'] =
zip(*speeches['speeches'].apply(has_high_uncertainty))


#speeches['uncertainty_percentatge'], speeches['number_uw']=
speeches['speeches'].apply(has_high_uncertainty).apply(pd.Series)


speeches['uncertainty_percentatge']= speeches['speeches'].apply(has_high_uncertainty)
```

```
speeches['uncertainty_words']=(speeches['uncertainty_percentatge']-
np.mean(speeches['uncertainty_percentatge'])) / np.std(speeches['uncertainty_percentatge'])

#speeches['uncertainty_words']=100/speeches['uncertainty_words']

#speeches['uncertainty_words'] = speeches['uncertainty_words'].replace(np.inf, 0)


# Creació histogram

n=plt.hist(speeches['uncertainty_words'], bins=10, range=[-3, 3],color='dodgerblue',
edgecolor='black')


plt.xlabel('Value')

plt.ylabel('Frequency')


speeches['uncertainty_label'] = speeches['uncertainty_words'].apply(label_uncertainty,
args=(np.mean(speeches["uncertainty_words"]),))


import matplotlib.pyplot as plt


# Colors

colors = ['dodgerblue','lightblue']

X=speeches.drop("uncertainty_label",axis=1)

y=speeches["uncertainty_label"]

y.value_counts().plot.pie(autopct='%.2f', colors=colors)


plt.show()


import matplotlib.pyplot as plt

speeches["year"] = speeches["year"].astype(int)


sp = speeches[speeches["year"]>=1929]
```

```
colors = ['dodgerblue','lightblue']
X=sp.drop("uncertainty_label",axis=1)
y=sp["uncertainty_label"]

y.value_counts().plot.pie(autopct='%.2f', colors=colors)

plt.show()

y.value_counts().plot.pie(autopct='%.2f', colors=colors)

a= speeches.at[45, "speeches"]

matching_words = [word for word in a if word in uncertainty_words]

# Treure duplicats
matching_words = list(set(matching_words))

print(f"Matching words: {matching_words}")

from sklearn.model_selection import train_test_split
from sklearn.ensemble import BaggingClassifier
from sklearn import tree
from sklearn.feature_extraction.text import TfidfVectorizer

speeches['speeches'] = speeches['speeches'].apply(lambda x: ' '.join(map(str, x)))

x = speeches['speeches']
y = speeches['uncertainty_label']
z = speeches['president']
```

```
d = speeches['year']


# Separem les dades entre train i test
x_train, x_test, y_train, y_test, Train_president, Test_president, Train_year, Test_year = train_test_split(x, y, z, d,  test_size=0.4, random_state=1, stratify=y)


print("Training set label distribution:", Counter(y_train))
print("Testing set label distribution:", Counter(y_test))


# Vectorització TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, min_df=0.05, ngram_range= (1, 1))
Train_X_Tfidf = tfidf_vectorizer.fit_transform(x_train)
Test_X_Tfidf = tfidf_vectorizer.transform(x_test)
X_Tfidf = tfidf_vectorizer.fit_transform(x)


y_train = y_train.map({'not uncertainty': 0, 'uncertainty': 1})
y_test = y_test.map({'not uncertainty': 0, 'uncertainty': 1})
y_ = y.map({'not uncertainty': 0, 'uncertainty': 1})


print(f"Train_X_Tfidf: {Train_X_Tfidf.shape}")
print(f"Test_X_Tfidf: {Test_X_Tfidf.shape}")


train_term_freq = np.sum(Train_X_Tfidf.toarray(), axis=0)
test_term_freq = np.sum(Test_X_Tfidf.toarray(), axis=0)


combined_freq = np.concatenate((train_term_freq, test_term_freq))
dataset_labels = ['Training'] * len(train_term_freq) + ['Testing'] * len(test_term_freq)


# Violin plot
```

```
plt.figure(figsize=(10, 6))
parts=plt.violinplot([train_term_freq, test_term_freq], showmeans=False, showmedians=True)
parts['bodies'][0].set_facecolor('dodgerblue')
parts['bodies'][1].set_facecolor('tomato')
plt.xticks([1, 2], ['Training', 'Testing'])
plt.ylabel('TF-IDF')
plt.grid(True)
plt.show()

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn import metrics

def evaluate(model, X_train, X_test, y_train, y_test):
    y_test_pred = model.predict(X_test)
    y_train_pred = model.predict(X_train)

    print("TESTING RESULTS: \n================================")
    clf_report = pd.DataFrame(classification_report(y_test, y_test_pred, output_dict=True))
    print(f"CONFUSION MATRIX:\n{confusion_matrix(y_test, y_test_pred)}")
    print(f"ACCURACY SCORE:\n{accuracy_score(y_test, y_test_pred):.4f}")
    print(f"CLASSIFICATION REPORT:\n{clf_report}")

from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier

random.seed(42)
np.random.seed(42)
```

```
weak_learner = DecisionTreeClassifier(max_depth=1)


ada_boost_clf = AdaBoostClassifier(estimator=weak_learner, n_estimators= 300,
learning_rate=0.9, random_state=42)

ada_boost_clf.fit(Train_X_Tfidf, y_train)

evaluate(ada_boost_clf, Train_X_Tfidf, Test_X_Tfidf, y_train, y_test)


from sklearn.model_selection import cross_val_score, KFold

num_folds = 3

kf = KFold(n_splits=num_folds, shuffle=True, random_state=42)

cross_val_results = cross_val_score(ada_boost_clf, X_Tfidf, y_, cv=kf)


print(f'Cross-Validation Results (Accuracy): {cross_val_results}')

print(f'Mean Accuracy: {cross_val_results.mean()}')

cv_auc_scores = cross_val_score(ada_boost_clf, X_Tfidf, y_, cv=kf, scoring='roc_auc')


print("Cross-Validation AUC scores:", cv_auc_scores)

print("Mean AUC:", np.mean(cv_auc_scores))


ada_boost_clf.fit(Train_X_Tfidf, y_train)


Train_X_Tfidf_dense = Train_X_Tfidf.toarray() if hasattr(Train_X_Tfidf, 'toarray') else
Train_X_Tfidf

Test_X_Tfidf_dense = Test_X_Tfidf.toarray() if hasattr(Test_X_Tfidf, 'toarray') else
Test_X_Tfidf

probabilities_train = ada_boost_clf.predict_proba(Train_X_Tfidf_dense)

probabilities_test = ada_boost_clf.predict_proba(Test_X_Tfidf_dense) #predicts class
probabilities for each sample


posterior_probabilities_test = (probabilities_test[:,1])  # Example probabilities
```

```
posterior_probabilities_train = (probabilities_train[:,1])

indice_test = (posterior_probabilities_test * 100).astype(int)
indice_train = (posterior_probabilities_train * 100).astype(int)

print(indice_test)

from sklearn.metrics import roc_curve, auc

mean_fpr = np.linspace(0, 1, 100)
tprs = []
aucs = []

# Iterem sobre cada plegament
for train_index, test_index in kf.split(X_Tfidf, y_):
    X_train, X_test = X_Tfidf[train_index], X_Tfidf[test_index]
    y_train, y_test = y_.iloc[train_index], y_.iloc[test_index]
    ada_boost_clf.fit(X_train, y_train)

    y_score = ada_boost_clf.predict_proba(X_test)[:, 1]

    # Corba ROC i AUC
    fpr, tpr, thresholds = roc_curve(y_test, y_score)
    tprs.append(np.interp(mean_fpr, fpr, tpr))
    tprs[-1][0] = 0.0

    roc_auc = auc(fpr, tpr)
    aucs.append(roc_auc)
```

```
# Plot corba ROC

plt.figure(figsize=(10, 6))


mean_tpr = np.mean(tprs, axis=0)

mean_tpr[-1] = 1.0

mean_auc = auc(mean_fpr, mean_tpr)

std_auc = np.std(aucs)


plt.plot(mean_fpr, mean_tpr, color='tomato', label=f'Mean ROC (AUC = {mean_auc:.2f} $\pm$ {std_auc:.2f})', lw=2, alpha=.8)


# Línia aleatòria

plt.plot([0, 1], [0, 1], linestyle='--', lw=2, color='dodgerblue', label='Random chance', alpha=.8)


std_tpr = np.std(tprs, axis=0)

tprs_upper = np.minimum(mean_tpr + std_tpr, 1)

tprs_lower = np.maximum(mean_tpr - std_tpr, 0)

plt.fill_between(mean_fpr, tprs_lower, tprs_upper, color='darkgrey', alpha=.2, label=r'$\pm$ 1 std. dev.')


plt.xlim([-0.05, 1.05])

plt.ylim([-0.05, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Cross-Validated ROC Curve for AdaBoost')

plt.legend(loc='lower right')

plt.show()


# Valor AUC
```

```
print(f"Mean AUC: {mean_auc:.2f}")


ada_boost_clf.fit(X_Tfidf, y_)
feature_importances = ada_boost_clf.feature_importances_
feature_names = tfidf_vectorizer.get_feature_names_out()


X = tfidf_vectorizer.transform(speeches["speeches"])


feature_importance_dict = {feature_names[i]: feature_importances[i] for i in
range(len(feature_names))}


index_values = np.zeros(X.shape)


for doc_idx in range(X.shape[0]):
    for word_idx in range(X.shape[1]):
        word = feature_names[word_idx]
        tfidf_value = X[doc_idx, word_idx]
        importance = feature_importance_dict[word]
        index_values[doc_idx, word_idx] = tfidf_value * importance


for doc_idx in range(X.shape[0]):
    print(f"Document {doc_idx}:")
    for word_idx in range(X.shape[1]):
        word = feature_names[word_idx]
        index_value = index_values[doc_idx, word_idx]
        if index_value > 0:  # Valors q no siguin 0
            print(f"{word}: {index_value}")


from collections import Counter
```

```
random.seed(42)

uncertainty_words = [

    'ambigu', 'ambival', 'anxieti', 'concern', 'confus', 'doubt', 'distrust',
    'mistrust', 'skeptic', 'suspicion', 'troubl', 'uneasi', 'unpredict',
    'worri', 'mayb', 'possibl', 'might', 'could', 'unsur', 'uncertain', 'like',
    'unlik', 'perhap', 'not sur', 'bewilder', 'conjectur', 'conting',
    'dilemma', 'disquiet', 'doubt', 'dubieti', 'guesswork', 'hesit', 'hesit',
    'incertitud', 'inconclus', 'indecis', 'irresolut', 'insecur', 'misgiv',
    'mystif', 'oscil', 'perplex', 'puzzl', 'puzzlement', 'qualm', 'quandari',
    'queri', 'reserv', 'scrupl', 'vagu', 'wonder'
]

df_speeches = pd.DataFrame(speeches["speeches"])

def word_count(speech):
    words = speech.lower().split()
    return dict(Counter(words))

df_speeches['Word_Counts'] = df_speeches['speeches'].apply(word_count)

rows = []

for idx, word_counts in enumerate(df_speeches['Word_Counts']):
    for word, count in word_counts.items():
        rows.append({'Index': idx, 'Word': word, 'Count': count})
```

```
df_word_counts = pd.DataFrame(rows)

results = df_word_counts[df_word_counts['Word'].str.contains('|'.join(uncertainty_words))]

print(results)

import math

random.seed(42)

df_speeches = pd.DataFrame(speeches["speeches"])

tfidf_vectorizer = TfidfVectorizer()
X_Tfidf = tfidf_vectorizer.fit_transform(df_speeches["speeches"])

ada_boost_clf.fit(X_Tfidf, y_)

feature_importances = ada_boost_clf.feature_importances_
feature_names = tfidf_vectorizer.get_feature_names_out()

feature_importance_dict = {feature_names[i]: feature_importances[i] for i in
range(len(feature_names))}

def word_count(speech):
    words = speech.lower().split()
    return dict(Counter(words))

df_speeches['Word_Counts'] = df_speeches['speeches'].apply(word_count)
```

```
def process_speech(speech_text, word_counts):
    speech_tfidf = tfidf_vectorizer.transform([speech_text])

    tfidf_values = speech_tfidf.toarray()[0]
    index_values = feature_importances

    words_with_values = [
        (feature_names[i], index_values[i], tfidf_values[i], word_counts.get(feature_names[i],
0))
        for i in range(len(feature_names))
        if feature_names[i] in uncertainty_words and index_values[i] > 0
    ]

    return words_with_values

all_speeches_results = []
for idx, speech_text in enumerate(df_speeches["speeches"]):
    word_counts = df_speeches.iloc[idx]['Word_Counts']
    words_with_values = process_speech(speech_text, word_counts)
    for word, value, tf_idf, count,  in words_with_values:
        all_speeches_results.append({
            "speech_index": idx,
            "word": word,
            "value": value,
            "tf_idf":tf_idf,
            "count": count
        })

results_df = pd.DataFrame(all_speeches_results)
```

```
max_tf_idf = results_df['tf_idf'].max()
max_weight = results_df['value'].max()
max_count = results_df['count'].max()

results_df = results_df[results_df['count']>0]

grouped_df = results_df.groupby('speech_index').agg({
    'tf_idf':'mean',
    'value':'mean',
    'count':'sum',
    'vc' : 'mean'
}).reset_index()

max_tf_idf = grouped_df['tf_idf'].max()
max_weight = grouped_df['value'].max()
max_count = grouped_df['count'].max()

grouped_df["vc"] =
(grouped_df["tf_idf"]/max_tf_idf)*(grouped_df["value"]/max_weight)*(grouped_df["count"]
/max_count)

print(grouped_df)

df = pd.DataFrame(grouped_df)

all_speech_indexes = range(59)

full_df = pd.DataFrame({'speech_index': all_speech_indexes})
```

```
results_df_2 = pd.merge(full_df, df, on='speech_index', how='left')

results_df_2['vc'].fillna(0, inplace=True)

results_df_2.sort_values(by='speech_index', inplace=True)

results_df_2["vc"] = results_df_2["vc"]
results_df_2.index = results_df_2.index + 1
results_df_2["speech_index"] = results_df_2["speech_index"] +1

results_df_2["uncertainty_pc"]=speeches["uncertainty_percentatge"]
print(results_df_2)

data = np.array(results_df_2["vc"])

results_df_2 = pd.DataFrame({"vc": data})

results_df_2.index = results_df_2.index + 1

print("Datos estandarizados (Z-score):")
print(results_df_2)

speech_uncertainty_test = pd.DataFrame({
    'President': Test_president,
    'Year': Test_year,
    'Proba': indice_test,
})
speech_uncertainty_train = pd.DataFrame({
    'President': Train_president,
```

```
    'Year': Train_year,
    'Proba': indice_train
})
speech_uncertainty = pd.concat([speech_uncertainty_test, speech_uncertainty_train])
speech_uncertainty

from google.colab import files
uploaded = files.upload()
df = pd.read_excel('SP500.xlsx')

df['% Return']=df["Total Return"]
df = df.groupby('Year').agg({
    '% Return': 'mean'
}).reset_index()
df["% Return"]=df["% Return"]
years_range = range(1929, 2024, 4)
filtros = [df['Year'] == year for year in years_range]
df = df.loc[pd.concat(filtros, axis=1).any(axis=1)]
df = df.sort_values(by="Year")

a = df.loc[3, '% Return']*1/3
df.loc[3, '% Return'] = df.loc[3, '% Return']-a
a = df.loc[7, '% Return']*1/3
df.loc[7, '% Return'] = df.loc[7, '% Return']-a

for i in range(11, 95, 4):
  a = df.loc[i, '% Return']*1/12
  df.loc[i, '% Return'] = df.loc[i, '% Return']-a
df
```

```
speech_uncertainty['Year'] = speech_uncertainty['Year'].astype(int)

speech_uncertainty = speech_uncertainty.sort_values(by="Year")

speech_uncertainty["Uncertainty Index"] = results_df_2["vc"]

speech_uncertainty["Uncertainty Index"]  = speech_uncertainty["Uncertainty Index"]
.fillna(0)

valor=1925

speech_uncertainty = speech_uncertainty[speech_uncertainty['Year'] > valor]

speech_uncertainty["Uncertainty Index"] = speech_uncertainty["Uncertainty Index"]*100


speech_uncertainty["Uncertainty Index"]


df['% Return'] = df['% Return'].astype(int)


fig, ax1 = plt.subplots(figsize=(10, 6))


ax1.bar(df["Year"], df['% Return'], color='dodgerblue', label='S&P 500 % Rendiment')

ax1.set_xlabel('Any')

ax1.set_ylabel('S&P 500 % Rendiment', color='black')


ax2 = ax1.twinx()

ax2.plot(speech_uncertainty['Year'], speech_uncertainty['Uncertainty Index'], 'tomato',
label='Índex incertesa')

ax2.set_ylabel('Índex incertesa', color='black')

#ax2.tick_params(axis='y', labelcolor='blaack')


ax1.set_xlim([1929, 2021])


tick_years = df["Year"]

ax1.set_xticks(tick_years)
```

```
ax1.set_xticklabels(tick_years, rotation=45)

fig.tight_layout()  # Ajustar automáticamente el espaciado para evitar superposición

# Añadir la leyenda
fig.legend(loc='upper center', bbox_to_anchor=(0.35, 0.97))
# Mostrar el gráfico
plt.show()

fig, ax1 = plt.subplots(figsize=(10, 6))

ax1.bar(df["Year"], df['% Return'], color='dodgerblue', label='S&P 500 % Rendiment')
ax1.set_xlabel('Any')
ax1.set_ylabel('S&P 500 % Rendiment', color='black')

ax2 = ax1.twinx()
ax2.plot(speech_uncertainty["Year"], speech_uncertainty["Proba"], 'tomato',
label='Probabilitat a posteriori Incert')
ax2.set_ylabel('Probabilitat a posteriori Incert', color='black')
#ax2.tick_params(axis='y', labelcolor='blaack')

ax1.set_xlim([1929, 2021])

tick_years = df["Year"]
ax1.set_xticks(tick_years)
ax1.set_xticklabels(tick_years, rotation=45)

fig.tight_layout()  # Ajustar automáticamente el espaciado para evitar superposición
```

```
# Añadir la leyenda
fig.legend(loc='upper center', bbox_to_anchor=(0.35, 1))
# Mostrar el gráfico
plt.show()

import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import zscore
from sklearn.preprocessing import MinMaxScaler

index_values_reshaped = np.array(speech_uncertainty['Uncertainty Index']).reshape(-1, 1)
sp500_returns_reshaped = np.array(df['% Return'] ).reshape(-1, 1)

scaler = MinMaxScaler()

index_values_standardized = scaler.fit_transform(index_values_reshaped)
sp500_returns_standardized = scaler.fit_transform(sp500_returns_reshaped)
plt.figure(figsize=(10, 6))

plt.plot(speech_uncertainty["Year"], index_values_standardized, c='tomato', label='Índex incertesa Normalitzat')
plt.plot(speech_uncertainty["Year"], sp500_returns_standardized, c='dodgerblue', label='S&P 500 % Rendiment Normalitzat')
for year in speech_uncertainty["Year"]:
    plt.axvline(x=year, color='darkgrey', linestyle='--')
plt.xlabel('Any')
#plt.ylabel('Índice de Incertesa Normalizat')
plt.xticks(speech_uncertainty["Year"], rotation=45)
plt.legend(loc='upper center', bbox_to_anchor=(0.35, 1))
plt.show()
```

```
volatility_sp500 = np.std(sp500_returns_standardized)
volatility_uncertainty = np.std(index_values_standardized)
mean_sp500 = np.mean(sp500_returns_standardized)
mean_uncertainty = np.mean(index_values_standardized)
print(f"Volatility (S&P500): {volatility_sp500}")
print(f"Volatility (Uncertainty Index): {volatility_uncertainty}")
print(f"Mean (S&P500): {mean_sp500}")
print(f"Mean (Uncertainty Index): {mean_uncertainty}")

import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import zscore
from sklearn.preprocessing import MinMaxScaler

plt.figure(figsize=(10, 6))
prob_post_values_reshaped = np.array(speech_uncertainty["Proba"]).reshape(-1, 1)
sp500_returns_reshaped = np.array(df['% Return'] ).reshape(-1, 1)

scaler = MinMaxScaler()

prob_post_values_standardized = scaler.fit_transform(prob_post_values_reshaped)
sp500_returns_standardized = scaler.fit_transform(sp500_returns_reshaped)

plt.plot(speech_uncertainty["Year"], prob_post_values_standardized, c='dodgerblue',
label='Prob a post. Normalitzada')
plt.plot(speech_uncertainty["Year"], sp500_returns_standardized, c='tomato', label='S&P500
Normalizat')
for year in speech_uncertainty["Year"]:
    plt.axvline(x=year, color='darkgrey', linestyle='--')
```

```
plt.xlabel('Any')
#plt.ylabel('Escaled')
plt.xticks(speech_uncertainty["Year"], rotation=45)
plt.legend()
plt.show()

volatility_sp500 = np.std(sp500_returns_standardized)
volatility_prob = np.std(prob_post_values_standardized)
print(f"Volatility (S&P500): {volatility_sp500}")
print(f"Volatility (Posterior Probability): {volatility_prob}")
```