UNIVERSITAT DE BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

MACHINE LEARNING PER A L'ANÀLISI I CLASSIFICACIÓ DE SENTIMENTS

Autor: Carlos Pons Gomila

Director:	Guillem Pascual Guinovart
Realitzat a:	Departament de Matemàtiques i Informàtica
Barcelona,	22 de juny de 2017

Abstract

Continuously people debate and express their opinions in their day to day life. Internet has encouraged these opinions to be made public through social networks or review platforms regarding a brand, product or service. To analyze this intake of information, mining of opinions is gaining strength.

A scraping system has been implemented to obtain a real data *corpus* and algorithms based on natural language processing (NLP) and Machine Learning are used to implement a sentiment analysis framework. By means of supervised learning, the reviews written at TripAdvisor regarding tourist accommodation are classified. A brief introduction to the world of neuronal networks is made and the *Keras* library is used in a regression problem to predict scores.

Finally, the result of analyzing opinions applied to hotels in the city of Barcelona is presented on a web map, obtaining an indicator that represents the degree of positive or negative comments for each hotel.

Resum

Contínuament les persones debaten i expressen la seva opinió en el seu dia a dia. Internet ha potenciat que aquestes opinions es facin públiques mitjançant les xarxes socials o plataformes de ressenyes respecte una marca, producte o servei. Per analitzar aquesta ingesta d'informació, la mineria d'opinions està cobrant gran importància.

S'ha implementat un sistema d'Scraping per a obtenir un corpus de dades real i s'utilitzen algorismes basats en el processament del llenguatge natural (NLP) i Machine Learning per posar en pràctica un anàlisis de sentiment d'opinions. Mitjançant l'aprenentatge supervisat es classifiquen les ressenyes escrites en el portal de Tripadvisor en referència a allotjaments turístics. Es realitza una breu introducció al món de les xarxes neuronals i s'utilitza la llibreria *Keras* en un problema de regressió per predir puntuacions.

Finalment, es presenta en un mapa web el resultat d'anàlisi d'opinions aplicat als hotels de la ciutat de Barcelona, obtenint un indicador que representi quin és el grau de comentaris positius o negatius per a cada hotel.

Agraïments

En especial, agrair a Guillem Pascual, tutor d'aquest treball, la seva constant dedicació i paciència amb mi al llarg del projecte, donar-li les gràcies per tots els consells que m'ha donat i per guiar-me en tot el procés.

A la meva família, pels ànims i suport que m'han donat al llarg d'aquests anys, a la meva germana i als meus pares, per la seva ajuda i per donar-me sempre suport en els moments difícils, i a la meva parella, pels ànims i per aguantar-me dia a dia.

Finalment, donar les gràcies a la Universitat de Barcelona, professors i companys amb qui he compartit aquesta experiència.

Taula de continguts

Taula de Figures

1. Introducció

1.1 Marc del treball

Amb la ràpida expansió de les noves tecnologies i internet com a font d'informació, s'ha vist incrementat l'ús dels portals web relacionats amb el turisme. Cada cop més usuaris consulten informació a internet per planificar un viatge.

Tripadvisor.com i Booking.com són un clar exemple, aquí es poden cercar els hotels i restaurants d'una ciutat o saber quines activitats d'oci es poden realitzar, d'entre altres opcions de recerca. Aquestes pàgines web es caracteritzen per permetre a l'usuari expressar la seva opinió, mitjançant una ressenya escrita i una valoració numèrica respecte el lloc que han visitat. A partir d'aquestes dades cada lloc obté una puntuació segons el criteri de cada portal, que serveix per confeccionar una classificació dels emplaçaments segons la destinació.

La puntuació s'estableix segons diversos factors, com donar més importància a la valoració numèrica de les critiques més recents o el número total de comentaris rebuts, d'entre altres. En aquest càlcul no es té en compte realment l'anàlisi del text de cada opinió [1]. Això pot donar casos dispars on la valoració numèrica d'una opinió sigui molt alta però el contingut del text descriu una valoració negativa o molt negativa, i a la inversa. Si un lloc ha rebut molts comentaris d'aquest tipus, la seva puntuació estarà desvirtuada.

Principalment per a les empreses del sector turístic, plataformes com Tripadvisor.com s'han convertit en una eina a incloure en la gestió del dia a dia, permet monitoritzar la retroalimentació dels clients el qual pot ajudar a que millorin els punts febles del negoci. Per els usuaris, saber què n'opinen els demes i veure aquesta opinió d'una manera clara, ràpida i resumida, cada cop es més freqüent en aquest tipus de recerca donada la gran quantitat d'informació que es pot trobar.

Aquest treball es centrarà en donar importància a l'anàlisi del text de cada opinió. Mitjançant tècniques de *Natural Language Processing* i l'ús *de Machine Learning*, incloent els models clàssics i *Deep Learning*, s'entrenaran diferents models de

classificació de sentiments, que aprendran si una ressenya expressa una opinió positiva o negativa. Els resultats obtinguts, segons aquest aprenentatge servirà per, a més a més de la valoració numèrica, per a cada opinió, tenir en compte aquesta polaritat de sentiment en el càlcul de la puntuació total del lloc.

1.2 Motivacions

La motivació principal d'aquest projecte es aprofundir els coneixements adquirits en assignatures cursades durant el Grau d'Enginyeria Informàtica que més van cridar la meva atenció, Taller de Nous usos de la informàtica: on s'introdueix com tractar grans volums de dades per extreure'n informació i els algorismes o tècniques per fer-ho i Intel·ligència artificial on s'han treballat els arbres de decisió.

La classificació de sentiments en les ressenyes d'un producte o servei és un repte, fent que sigui un camp d'investigació que està en alça [2]. Donat que no hi ha actualment cap plataforma de ressenyes hoteleres que ho estigui aplicant, i en concret sobre la que s'ha realitzat el projecte, em va portar a investigar com fer possible mostrar aquesta informació a l'usuari, i que pugui ser implementat en futures millores de les plataformes turístiques.

1.3 Objectius

Objectiu general:

L'objectiu general és crear una web interactiva on es mostrin visualment els resultats d'anàlisi de sentiments d'opinions i d'extracció de característiques, aplicat en aquest cas a comentaris d'hotels, fent servir GoogleMaps. L'usuari ha de poder diferenciar de manera clara aquells hotels que han rebut segons aquest anàlisi, pitjors o millors crítiques per part dels clients en conjunt, oferint d'aquesta manera una capa més d'informació als sistemes actuals que proveeixen els portals turístics.

Objectius específics:

Per dur a terme l'objectiu general, s'ha de complir els següents aspectes.

- Dissenyar un sistema per recavar dades fent servir tècniques Scraping. S'ha de permetre processar un gran volum de dades d'un conegut portal web, tripadvisor.com, per crear una base de dades pròpia amb dades reals.
  Per a cada hotel s'han d'obtenir les crítiques escrites en llengua anglesa, les seves valoracions i la data que es va fer pública la crítica. El sistema ha de ser el màxim de consistent i robust per ser més eficient.

  Aquest sistema de recaptació ha de ser fàcilment modificable i adaptable per a altres webs similars en un mateix propòsit.

- Crear diferents representacions de les dades anteriors, embeddings, fent servir tècniques de NLP que capturin les característiques més representatives, com bag-of-words o word2vec.
- Implementar diferents models de classificació de sentiments d'opinions fent servir la llibreria sklearn en Python, i analitzar-ne els resultats d'encert.
- Visualitzar la distribució que segueixen els resultats de predicció segons els embeddings creats, utilitzant tècniques com PCA i T-SNE que permetin veure la dispersió entre les possibles agrupacions.
- Crear una xarxa neuronal i que la sortida d'aquesta sigui la predicció de la puntuació de cada opinió. D'aquesta manera, en casos on no es disposi prèviament d'aquesta informació, la xarxa sàpiga predir i obtenir-ne resultats similars.
- Calcular per a cada hotel que s'hagi analitzat, la seva puntuació mitja, tenint en compte ara sí, els resultats d'anàlisi anteriors de cada opinió, tant de sentiment i puntuació com d'altres característiques com la data de publicació.
- Crear la corresponent visualització dels resultats de tots els hotels de la ciutat de Barcelona fent servir la API de GoogleMaps.

1.4 Estructura de la memòria

En el primer capítol s'ha introduït el marc del problema i els objectius que s'esperen aconseguir en aquest projecte, en endavant la memòria es distribueix donant un enfocament més teòric en la primera part per posteriorment introduir-se a la part pràctica i resultats obtinguts.

- **Metodologia i tecnologies:** Introducció a les tecnologies que s'han fet servir per l'anàlisi d'unt text i com s'obtenen característiques importants.
  S'explica també el conjunt de dades utilitzat, els problemes que presenten i com minimitzar-los sempre que es pugui.

- **Desenvolupament pràctic i resultats:** On s'explicarà els diferents mètodes que composen l'arquitectura presentada a nivell d'implementació i els passos seguits de manera documentada per a dur a terme la classificació d'opinions. S'exposaran i compararan diferents models d'entrenament i com s'interpreten les característiques extretes. Es veuen els resultats obtinguts en cada una de les etapes fent servir els mètodes clàssics i utilitzant xarxes neuronals, es realitzen comparatives dels models d'entrenament segons el temps d'execució i el percentatge d'encert en la classificació.

- **Conclusions:** Es donarà una **v**aloració personal sobre el projecte, s'exposaran les conclusions a les que s'ha arribat segons els resultats obtinguts, es parlarà dels objectius assolits i la línia que pot seguir en un futur aquest projecte.

2. Estat de l'art

En els darrers anys, l'anàlisi de sentiments (Sentiment analysis), o mineria d'opinions (Opinion mining), és un tema de gran interès. La tasca de classificar de manera automàtica un text escrit en llenguatge natural en un sentiment positiu o negatiu, és un repte. En casos concrets es fa fins i tot complicat per a les persones humanes poder dur a terme aquesta categorització rotunda[3]. Entendre què expressa un text pot ser interpretat per cada persona d'una manera diferent. Diverses tecnologies han permès avanços en aquest aspecte obtenint bons resultats de classificació. Tècniques d'aprenentatge automàtic [4], les aproximacions semàntiques [5] i les xarxes neuronals[6], són un exemple.

Les aproximacions semàntiques processen el text i el divideixen en paraules per a passar eliminar les paraules més comuns d'aquella llengua abans d'aplicar altres tècniques com l'agrupació de les paraules segons l'arrel o lema. Posteriorment es comprova l'aparició dels termes que resten per assignar el valor de polaritat a aquell text segons la suma dels valors individuals de cada terme. Aquestes tècniques fan ús de diccionaris lèxics que denoten orientació semàntica de polaritat, d'entre altres normalitzacions lingüístiques.

L'aprenentatge automàtic consisteix en entrenar un model de classificació fent servir algorismes d'aprenentatge supervisat a partir de dades prèviament etiquetades, que permeten obtenir un model que servirà per realitzar les prediccions en dades no etiquetades.

Els dos mètodes tenen les seves pròpies limitacions. L'aproximació semàntica requereix de terminologies i lèxics que persones humanes han establert, en aquest, cas els errors comesos pel sistema poden ser mitigats ampliant els diccionaris terminològics afegint-hi nous termes, tot i que caldrà de nou invertir més temps per l'ampliació d'aquests. En els sistemes basats en l'aprenentatge supervisat o les xarxes neuronals, els errors comesos pel sistema no són tant senzills de corregir, disposar de nova informació o informació amb més riquesa lingüística pot ésser un problema, i s'haurà de tornar a entrenar el model perquè aprengui de nou.

Les darreres investigacions al respecte s'han portat a terme fent servir xarxes neuronals i tècniques de Deep Learning mitjançant aprenentatge no supervisat [7]. Aquests analitzen tot el text i classifiquen les diferents parts que el conformen, a cada paràgraf tracten les frases en diferents nivells de positivitat o negativitat i no de manera global per a tot el text o document.

Altres investigacions aprofundeixen en l'aprenentatge d'aspectes com la part irònica o sarcàstica d'una opinió, un tema molt mal de resoldre a nivell semàntic. També s'analitzen els textos tenint en compte aquelles representacions amb signes de puntuació que puguin expressar una sensació, més coneguts com emoticones ja que molts cops es tendeix a expressar-se d'aquesta forma. On més s'han aplicat aquestes tècniques de processament del llenguatge natural fins al moment és en l'anàlisi de crítiques de cinema i comentaris escrits en xarxes socials com twitter, que han permès analitzar la tendència que segueixen els comentaris actuals donat milers de tòpics.

3. Metodologia i tecnologies. Un enfoc teòric

3.1 Obtenció de dades.

3.1.1 Scraping

Per obtenir dades d'una web, si aquesta ho permet, es fa servir el que s'anomena un API web, un mòdul addicional del lloc web on es permeten realitzar consultes al portal i que aquest retorni tant sols aquella informació sol·licitada. En canvi, en altres pàgines aquesta informació tant sols s'hi pot accedir visualment dins la pròpia pàgina web, el que dificulta la ràpida obtenció d'aquesta si es tracta de recavar gran quantitat d'informació.

En casos on no hi ha una API disponible per poder extreure dades, per copiar-les s'ha de recórrer a sistemes que simulin la feina que una persona humana realitzaria davant l'ordinador per obtenir-les dins la pròpia web, d'una manera automàtica i guardant de manera estructurada i eficient les dades obtingudes. Aquests sistemes s'anomenen Web Scraping [8].

En pàgines web estàtiques, aquesta informació es pot trobar en el mateix document html que conforma la pàgina, en canvi, en pàgines web dinàmiques, el funcionament és diferent.

Les pàgines web dinàmiques són aquelles que la informació que conté es presenta a partir de que l'usuari fa una petició a la pàgina. La informació és visible i accessible un cop es fa la crida al servidor i aquest respon mostrant la informació per pantalla. Diferents llenguatges de programació permeten generar de manera dinàmica els resultats a mostrar mitjançant bases de dades que emmagatzemen aquesta informació en la banda del servidor.

Aquest segon, és el sistema que utilitza Tripadvisor i per això cal utilitzar mecanismes per fer aquestes peticions de manera automàtica simulant ser una persona humana la que està realitzant aquestes peticions. S'ha fet servir el framework Selenium [9], de codi obert, que permet mecanismes automàtics per extreure dades estructurades de pàgines web dinàmiques fent servir els selectors html [10] mitjançant un script que contindrà les ordres a ser executades i reproduïdes. Aquesta informació es guardarà en una base de dades pròpia que es farà servir durant el projecte.

3.1.2 PhantomJS

Selenium necessita d'un navegador que pugui carregar el contingut web per posteriorment atacar a aquesta informació. D'entre els navegadors disponibles, s'ha fet servir PhantomJS [11], sense interfície gràfica s'executa per consola i es pot controlar fent servir el llenguatge de programació Python, el qual permet executar-hi scripts per a carregar les pàgines i poder interactuar amb elles. No disposar d'interfície gràfica dificulta l'exploració visual però fa més àgil la carrega de les webs i/o monitorització.

3.2 Representacions de les dades

3.2.1 Consideracions

Per a començar a modelar el sistema, un cop recavada la informació, aquesta ha d'estar conformada per dades coherents. En un anàlisis de text d'un conjunt de dades que es parla de cinema no es pot fer servir per comparar opinions que parlen de política, per exemple. La base de dades obtinguda correspon a un domini de dades associat a ressenyes hoteleres. Com s'ha comentat a l'estat de l'art, fer servir sistemes basats en l'aprenentatge automàtic requereix de dades prèviament etiquetades. Les dades recavades presenten el problema de no formar part d'un *corpus* de dades etiquetat, per tant s'experimentaran amb aquestes fent un anàlisi previ i es faran servir opinions escrites en llengua anglesa, donat que hi ha mes dades disponibles en terminologia i camps com el processament del llenguatge natural estan mes avançats en aquesta llengua.

Per a que una màquina sàpiga interpretar les dades que conformen un text, aquestes han de ser representades en un format numèric, no es pot treballar amb text directament, cal trobar una representació numèrica per a cada paraula i/o frase. Això s'anomena embedding de dades.

3.2.2 TF-IDF

La semàntica lingüística d'un text es representa segons un conjunt de paraules clau, és lògic pensar que no totes les paraules d'un text són rellevants, per tant, no tots els termes d'aquell text tenen la mateixa importància per representar la semàntica [12]. Aquest sistema assigna un pes a cada paraula, un valor numèric. Quant més gran sigui aquest valor, més importància tindrà aquella paraula a l'hora d'extreure'n les principals característiques del text.

"Term frequency - Inverse document frequency", defineix una mètrica per calcular el pes d'una paraula per un text en un conjunt de documents, tenint en compte que:

- Si en un document apareix repetidament un terme, es considera que es fortament representatiu en la semàntica, per el que tindrà un major pes.
- En quants més documents es faci referència un terme, menys útil serà aquell terme, per el que tindrà un menor pes.

Term frequency: El factor **tf** cobreix el primer punt, es la suma de totes les ocurrències o el número de vegades que una paraula apareix en un text o document.

Inverse document frequency: El factor **idf** fa referència al segon apartat, mesura si el terme es troba present en gran part dels documents o no, dividint el número total de documents per el número de documents que contenen aquell terme realitzant el logaritme d'aquest quocient.

$$w_{i,j} = tf_{i,j} \cdot log\left(\frac{N}{df_i}\right)$$

On $tf_{(i,j)}$ es el número de cops que *"i"* apareix en *"j"* i df_i el número de documents que contenen *"i"*. *N* es el número total de documents.

3.2.3 Bag-of-Words

Una de les representacions més simples i utilitzades per descriure un text és la representació bag-of-words. Aquesta representació converteix un text en un vector de *N* paraules. Consisteix en seleccionar un conjunt d'*N* paraules que descriuran segons les seves característiques aquell text, el que permet trobar similituds per a la classificació [13]. Prèviament es filtren aquelles paraules que no interessen perquè seran en un principi poc rellevants, i segons les modificacions que es faci al text original, s'obtindrà un model més o menys redundant, format per les N paraules mes representatives que descriuen el text (Figura 1).

Les possibles modificacions prèvies que poden aplicar-se són:

- Eliminar paraules més comuns donat un idioma, "Stop Words". S'inclouen els determinants, les preposicions, conjuncions o paraules més comuns donada una temàtica en concret. En general són pobres semànticament.
- Transformar totes les paraules en majúscules o minúscules, el que facilitarà processar el conjunt del text.
- Representar cada paraula segons la seva arrel, sintetitzant les dades, donat que no es tindran en compte les formes verbals, pluralitat i singularitat, o gènere.
- Similar al procediment anterior, representar cada paraula segons el seu lema, per permetre'n extreure sinònims e identificar contextos.

Addicionalment es pot combinar aquest mètode juntament amb TF, ajustant més els resultats de sortida. L'elecció d'aquest conjunt de dades per representar el vector de característiques és un pas crític. En funció de com bones siguin aquestes, millor funcionarà el sistema.

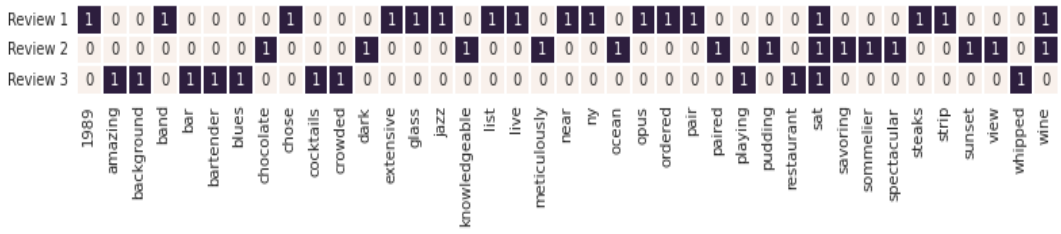


Figura 1: Matriu binaria de representació de paraules. Per cada paraula s'indica si aquesta apareix o no en cadascuna de les opinions. També es pot representar de manera multinomial, segons el número de cops que cada paraula apareix dins cada opinió.

Font: Data-Science OpenTable

3.2.4 Word2Vec

Aquesta model s'utilitza per l'aprenentatge de representacions vectorials de paraules. Internament fa ús de l'aprenentatge semàntic mitjançant una xarxa neuronal poc profunda per comprendre el significat d'una paraula en una oració. Permet crear diferents agrupacions de paraules que tinguin el mateix significat, o siguin similars, mitjançant la seva proximitat dins el conjunt[14], és dons un espai multidimensional on cada paraula es representa com un vector (Figura 2).

Aquest aprenentatge és un mètode no supervisat, donat que les paraules no estan prèviament etiquetades. El fonament principal és que les paraules que apareixen en contextos semblants, són semblants.

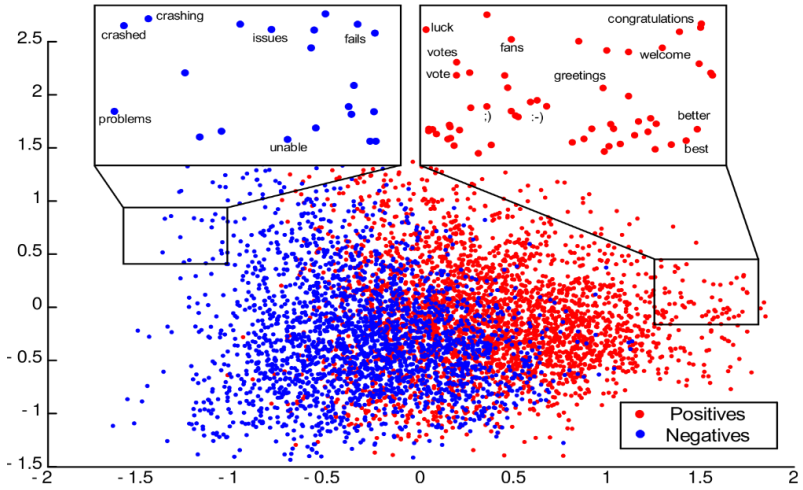


Figura 2: Representació de paraules segons la predicció del seu context amb exemples positius i negatius. **Font:** *Sentiment Analysis of Tweets from Twitter – MICC– Univ. of Firenze. 2016*

3.3 Visualització

Per visualitzar gràficament les característiques anteriors, es necessiten mecanismes que permetin d'alguna forma veure com estan distribuïdes aquestes dades. Com les representacions anteriors són vectors, dades multidimensionals de molt alta dimensionalitat, en conjunt es fa difícil la seva interpretació.

Hi ha sistemes que permeten visualitzar aquestes representacions de les dades per poder analitzar-les i entendre la seva distribució, reduint la dimensió de la matriu característica. Per exemple ajudarà a veure si hi ha prou diferenciació entre cadascun dels possibles clústers obtinguts, per veure la riquesa lingüística del conjunt de dades o detectar possibles agrupacions de paraules similars. Aquest tipus de visualitzacions són també molt comuns per el processament d'imatges similars, d'entre altres aplicacions.

3.3.1 PCA

L'anàlisi de Components Principals tracta de reduir un conjunt d'informació format per moltes característiques diferents en un altre més petit amb l'objectiu de perdre la menor quantitat d'informació possible identificant patrons en les dades (Figura 3). Es basa en una tècnica estadística de síntesis d'informació on els resultats obtinguts, els nous components principals, són combinacions lineals dels vectors originals. Aquesta representació interna es projecta sobre les direccions de més variància en un nou espai de dades que ha de complir ser de dimensió igual o més petit que el conjunt inicial [15] conservant la major part d'informació.

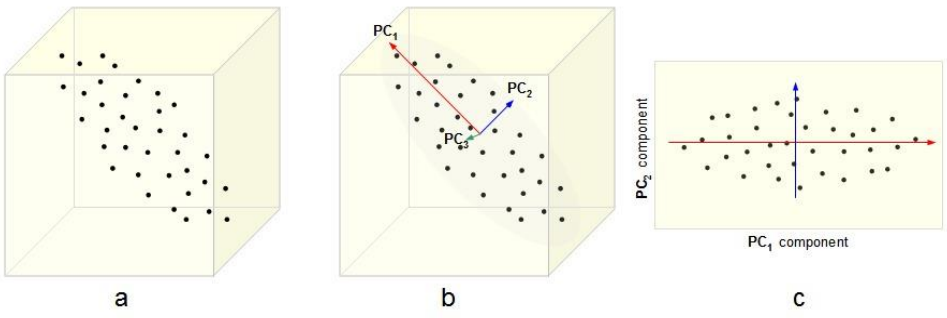


Figura 3: Transformació PCA - **a)** Representació 3-D del conjunt d'entrada. **b)** En aquest cas els 3 principals vectors ortogonals que representen les dades, ordenats segons la seva variància. **c)** Es projecten les dos components principals en base la seva importància, descartant les que en tenen menys, d'aquesta manera es redueix la dimensionalitat.

Font: *Dimensionality Reduction Methods, arxive-cnx: m11461*

3.3.2 T-SNE

Aquest sistema permet reduir l'espai multidimensional de les dades a dues o tres dimensions[16]. L'algorisme consisteix en una reestructuració dels punts en l'espai multidimensional on s'hi representen les dades per reduir la dimensió. Els punts entre si més propers tendiran a ajuntar-se i els més llunyans a separar-se.

Mitjançant uns paràmetres, es pot canviar la seva execució per defecte, per veure les diferents convergències dels resultats d'un mateix model i decidir quins paràmetres s'ajusten més a una correcta visualització [17].

3.4 Models

3.4.1 Conceptes bàsics

Una de les tasques bàsiques de la mineria d'opinions és fer front al problema de classificar dades mitjançant l'aprenentatge supervisat. El problema de classificació és aquell que té com a objectiu donar una estimació o predicció sobre quina categoria pertany una entrada [18]. El procés de classificació consta de dues parts: el procés d'aprenentatge i el procés de predicció (Figura 4).

En aquests sistemes les dades es separen en diferents conjunts abans de començar el primer procés, dades d'entrenament, que s'utilitza per construir el model, i dades de test, per a la validació d'aquest. Es formen aleatòriament i acostuma a ser un 70% de les dades per entrenar el sistema, i l'altre 30% per validar-ho.

Inicialment es du a terme una sèrie d'observacions de les dades d'entrenament que estan associades a una certa categoria o etiqueta a la que pertany, juntament amb les característiques extretes amb els mètodes vists anteriorment de representació de les dades o d'extracció de característiques amb l'objectiu d'aprendre les diferències entre aquestes.

Durant l'entrenament, el sistema captura aquesta informació que posteriorment el classificador farà servir per realitzar les prediccions i de les que també s'han obtingut les seves característiques per veure a quina categoria pertany.

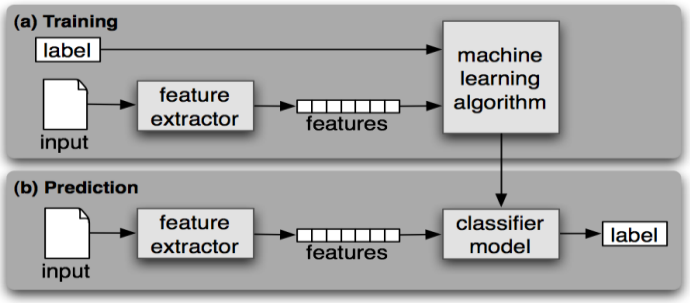


Figura 4: Procés de classificació supervisada.

Font: Learning to Classify Text, llibre NLTK, cap.6 – Supervised Classification

El sistema en aquest punt té una tassa d'encert i inversament una tassa d'error. Per saber que el sistema prediu correctament noves dades amb el model que s'ha obtingut, es fan servir sistemes de validació, per compensar tasses d'error optimistes en models massa adaptats donades les dades d'entrenament. Consisteix en dividir les dades d'entrada en diferents conjunts, cada cop que s'executa l'algorisme ho fa amb particions de dades diferents a l'anterior iteració per modelar el sistema. El conjunt de tècniques "Cross-Validation" permet realitzar aquest procés de manera automàtica. Finalment s'obté una mesura de l'eficiència general del model predictiu, segons la mitja.

D'entre el diferents tipus de validació creuada existents s'ha utilitzat "k-Fold Cross Validation" (Figura 5).

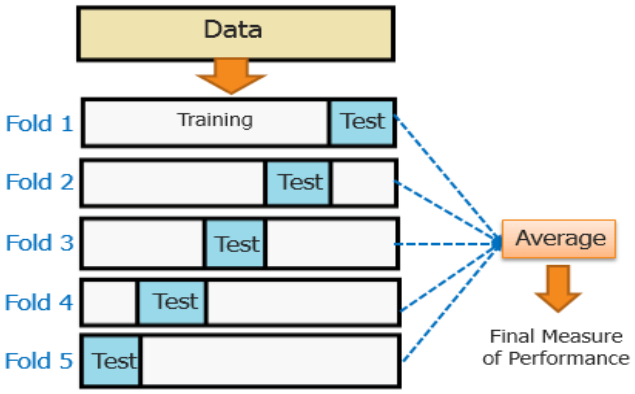


Figura 5: Procés "k-Fold Cross Validation" amb *k=5*

Font: edureka - Big Data Analytics Blog.

3.4.2 Sistemes clàssics

3.4.2.1 Naive Bayes

Un cop representades les paraules que formen el vector de característiques, es necessita un procés d'aprenentatge que permeti passar de la descripció de les dades d'una opinió a una categoria. Naive Bayes (NB) és el classificador bayesià més senzill i s'utilitza quan es vol classificar un exemple X descrit per un conjunt d'atributs a_j dins un conjunt C que té un valor assignat de classe c_i.

Els classificadors probabilístics Bayesians es basen en el teorema de Bayes per realitzar els càlculs per trobar la probabilitat condicionada que un conjunt de dades pertany a una classe particular o a una altra.

$$P(c_i\,, a_j) = \frac{P(a_j \mid c_i) \cdot\ P(c_i)}{P(a_j)}$$

En molts casos, $P(c_i)$ es considera equiprobable i al ser $P(a_j)$ un denominador comú en tots els casos, es pot simplificar a:

$$P(c_i \mid a_j) = c \cdot P(a_j \mid c_i)$$

Fins aquest punt, la majoria de classificadors bayesians funcionen de la mateixa manera. Naive Bayes es caracteritza per pressuposar que els atributs, o les paraules de cada exemple són independents entre sí, el qual relaxa el problema, al no tenir en compte la probabilitat de cada document ja que no aportarà informació per a la classificació, això fa que sigui diferent dels demés mètodes [19]. La probabilitat que un document pertany a una classe s'assumeix com la probabilitat conjunta de tots els termes que apareixen en els documents de la classe.

Es classifiquen els nous exemples d'acord amb el valor més probable atenent els valors de les seues característiques. La decisió final correspon a la categoria amb més probabilitat condicional.

$$P(a_{jk} = 1 \mid\ c_i = C) = \frac{A}{B}$$

On *A* és el número d'opinions de la categoria *C* on hi apareix la paraula "k*"* dins la frase "j*",* i *B* és el número total d'opinions de la categoria *C*.

Per solucionar problemes on la probabilitat sigui 0, per exemple, en casos on en una opinió no aparegui una paraula i que faci que aquella opinió no pugui ser classificada, s'utilitza la correcció de Laplace, per donar una baixa probabilitat en comptes de 0.

Segons la fórmula anterior, la correcció de Laplace es pot aplicar com:

$$P(a_{jk} = 1 |\ c_i = C') = \frac{A+1}{B+M}$$

On *M* és el número de categories existents.

3.4.2.2 Random Forest

Aquest algorisme és un dels mètodes d'aprenentatge supervisats més utilitzats en la bibliografia clàssica i està compost per arbres de decisió, que es poden veure com un conjunt de decisions que es duen a terme a partir de les dades disponibles que conformen el vector de característiques. S'estructuren jeràrquicament en forma d'arbre fent que sigui possible aplicar-se a tasques de classificació fent servir més d'un arbre per un conjunt d'entrenament. Per la seva implementació s'ha fet servir la llibreria sklearn [20] amb Python utilitzant diferents números d'arbres on a la part de desenvolupament pràctic s'exposaran els resultats.

Així, doncs, l'objectiu dels arbres de decisió és obtenir les regles o relacions que permeten determinar la classe d'un exemple mitjançant el seu recorregut dins l'arbre, segons els valors dels seus atributs (Figura 6).

Un arbre de decisió està format per un node arrel, nodes internes, nodes terminals i branques.

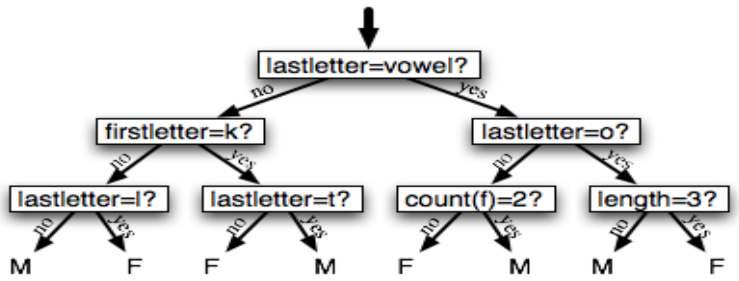


Figura 6 : Arbre de decisió. Decisió que pot prendre un arbre al que es dona com a entrada la darrera lletra del nom d'una persona per determinar si aquesta és home o dona.

Font: Learning to Classify Text, llibre NLTK, chap.6 – Decision Trees.

Cada node representa un atribut de les dades d'entrenament i les branques corresponen als possibles valors que pot prendre aquell atribut. Finalment els nodes terminals contenen la categoria o etiqueta assignada. El punt fort de l'algorisme Random Forest és que permet crear diferents arbres de predicció per un mateix conjunt de dades d'entrenament. Cada arbre correspondrà a un subconjunt diferent de característiques. D'entre tots ells es prendrà la decisió final de predir a quina classe pertany l'entrada.

Quants més arbres es creen més possibilitats diferents tindrà l'algorisme per d'entre tots ells, trobar la millor predicció, segons el que ha decidit per exemple la majoria, requerirà també més temps de càlcul, per tant s'ha de trobar un terme mig entre aquest número d'arbres, la grandària de les dades a entrenar i el temps d'execució.

Random Forest és robust contra dades no normalitzades i presenta bons resultats contra el problema d'overfitting. Aquest problema es dona quant el model s'ajusta molt bé a les dades inicials però baixa el rendiment en el moment que es prediuen resultats per noves dades.

3.4.2.3 Logistic Regression

La regressió logística està dins els conjunt de tècniques de classificació i s'utilitza en situacions les quals s'ha de classificar les dades segons dues categories o classes. Tracta de correlacionar la probabilitat d'una variable binària *Y*, que pot prendre valors "0" i "1", amb una variable escalar *x*. L'idea és que la regressió aproximi la probabilitat de que la variable *Y* prengui el valor "0" o "1" (Figura 7) segons les característiques de la variable *x* amb una funció logística.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Aquest mètode s'utilitza constantment en prediccions sota un escenari del que es coneixen prèviament les dades i les característiques poden ésser tant numèriques com categòriques. Per exemple, la probabilitat de que una persona tingui un atac de cor dins un període de temps determinat podria predir-se a partir de la seva edat, sexe i l'índex de massa corporal, fent servir una regressió.

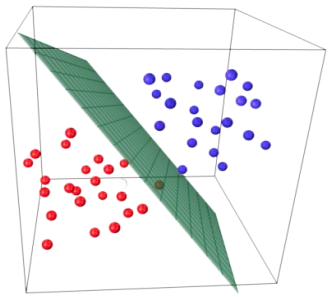


Figura 7: Frontera de separació entre dues classes d'una regressió logística. En verd, el pla que separa aquestes dades linealment entre els valors o punts de cada classe.

Font: Github

3.4.3 Xarxes neuronals

3.4.3.1 TensorFlow/Keras

Per treballar amb xarxes neuronals sense entrar en un nivell molt baix d'implementació, s'ha utilitzat *Keras* [21], una llibreria a més alt nivell que no pas Tensorflow o Theano i que fa més senzilla la creació i manipulació de xarxes. Una xarxa neuronal és un conjunt de neurones artificials que simulen el comportament de les neurones biològiques utilitzant models matemàtics de processament de dades (Figura 8).

Per crear la xarxa amb *Keras* s'ha fet servir la classe seqüencial, on cada capa està connectada amb la capa anterior i la següent. El propòsit és que la xarxa aprengui a fer prediccions segons classificació binària o mitjançant regressió.

Les dades d'entrada en el model neuronal són els vectors característics obtinguts amb word2vec.

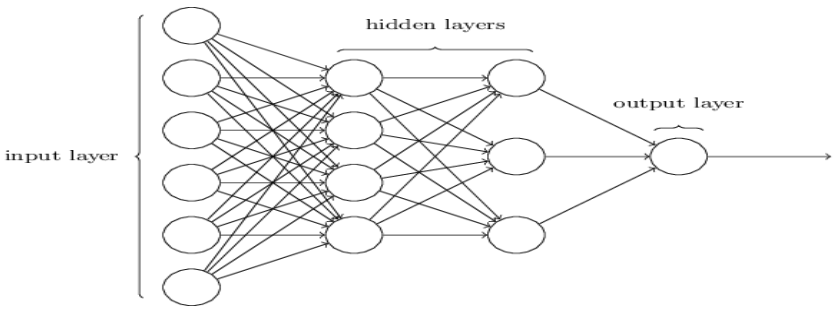


Figura 8: Estructura d'una xarxa neuronal senzilla. Les diferents etapes del procés s'anomenen "capes" i després de cada capa s'aplica una operació no lineal.

Font: Michel A.Nielsen, "Neural Networks and Deep Learning" – chap 1.

Les capes necessàries per donar estructura a la xarxa són:

- **Dense:** Representa una capa de neurones on totes les sortides de la capa anterior estan connectades amb les entrades d'aquesta. Per a cada neurona es realitza un producte matricial segons els pesos corresponents.
- **Convolution:** S'apliquen diferents filtres convolucionals a la capa anterior per obtenir un resultat. (Figura 9)
- **Activation:** S'aplica la funció de rectificació *ReLU* (Figura 11 c).
- **Max-Pooling:** S'encarrega de reduir els paràmetres a analitzar rebaixant la seva dimensionalitat, de manera que es queda amb les activacions més altes per cada subconjunt d'entrada. (Figura 10)
- **Dropout:** Desactiva alguna de les sortides de la capa anterior, excloent un percentatge de les neurones per reduir l'overfitting**.**
- **Flatten:** Converteix representacions matricials de dues o més dimensions en un vector, fent que la següent capa tingui com a entrada aquesta representació per poder ser processada.
- **Batch-Normalization:** Normalitza la sortida a mitja 0 i desviació estàndard 1.

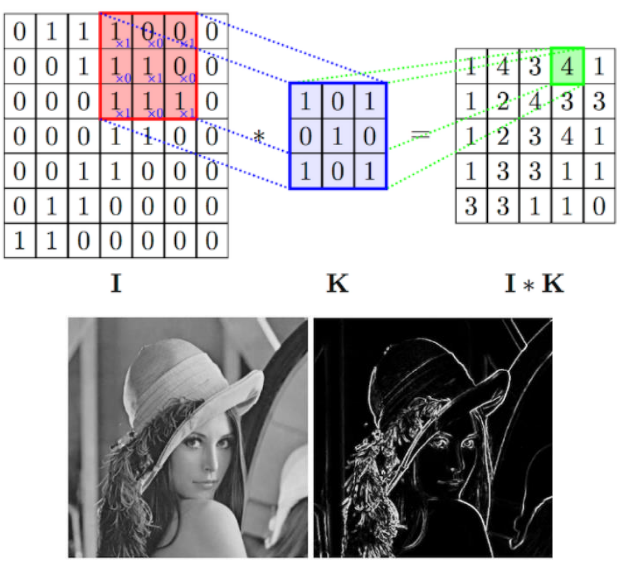


Figura 9: Exemple d'aplicació d'una convolució per obtenir els contorns d'una imatge. Amunt el procés de convolució per a una imatge, on a la matriu original se li aplica una màscara per obtenir una nova imatge que representi els contorns, com es pot veure a sota.

Font: Deep Learning – Convolutional NN with *Keras*, Cambridge Spark.

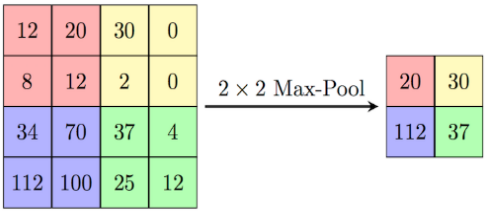
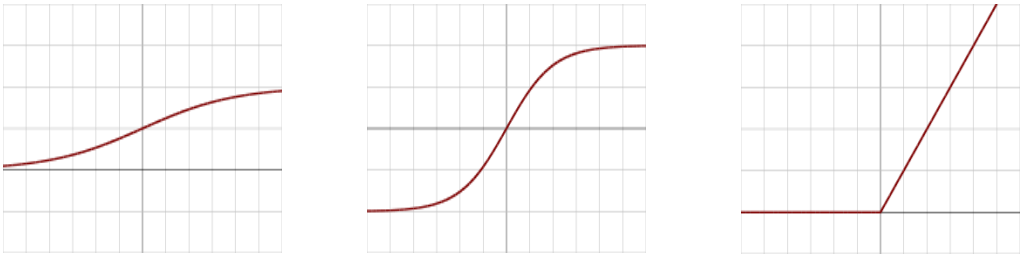


Figura 10: Diagrama Max-Pooling aplicat a una imatge, es divideix la imatge d'entrada en diferents parts i sobre cada una d'aquestes s'agafa el màxim.

Font: Deep Learning – Convolutional NN with *Keras*, Cambridge Spark.

Les neurones biològiques poden estar en dos estats, de manera activa o inactiva, es a dir tenen un "estat d'activació" segons si es troben excitades o no. Les neurones artificials poden simular aquest comportament amb funcions threshold. Si el resultat de la funció threshold és major que un valor llindar, la neurona s'activa i emet una senyal a les neurones de la capa següent, en cas contrari la neurona no envia cap senyal i es manté inactiva. En general és més comú que es facin servir funcions d'activació no lineals (Figura 11) on el resultat pot prendre un valor dins un cert rang.



a) Sigmoid b) TanH c) ReLU

Figura 11: Funcions d'activació no lineals que s'han fet servir en la implementació de la xarxa.

En la funció d'activació Sigmoidea, (Figura 11 a) s'observa com la sortida de la està definida en un rang [0, 1] i s'aplica la mateixa fórmula que s'ha vist en la regressió logística.

Similar a la funció anterior, la tangent hiperbòlica (Figura 11 b) defineix la sortida en un rang de [-1, 1]:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

La funció de rectificació lineal, comparat amb les dues anteriors, permet accelerar el procés d'entrenament i millora el procés d'aprenentatge ajustant el gradient de la funció i es calcula com:

$$f(x) = \{\, x, \quad si\ x > 0 \,\}$$
$$\{\, 0, \quad altrament \,\}$$

3.5 Mapa Interactiu

3.5.1 API Google

Google Maps és un servidor de mapes en plataformes web que proporciona de manera pública diferents API que s'ha fet servir per a desenvolupar aquest projecte. D'entre les possibilitats que ofereix aquesta plataforma, es permet enllaçar un mapa a qualsevol lloc web i poder personalitzar la manera de representar-hi dades addicionals als típics marcadors de posició, d'entre altres més opcions.

En concret, s'ha utilitzat el servei de GEO codificació [22], i una capa de visualització de dades mitjançant Javascript. El primer d'aquest consisteix en convertir adreces en coordenades geogràfiques (latitud i longitud) per poder posicionar cada hotel en el mapa, mentre que la capa de visualització permet emmagatzemar diferents paràmetres d'informació en format GeoJSON per la posterior representació en el mapa.

3.5.2 Càlcul per hotel

En el mapa, es vol diferenciar aquells hotels que han estat millor o pitjor valorats segons els usuaris, així doncs, per representar-ho, a cada hotel se li ha assignat una puntuació (Equació 1), tenint en compte els següents aspectes.

- El número de dies que han passat des de la publicació de cadascuna de les crítiques analitzades. Les crítiques més recents, tindran major rellevància que no pas les més antigues.
- La puntuació de cada opinió predita per la xarxa en un rang de 0 a 50.
- La categoria a la que pertany cada opinió, positiva o negativa, segons la predicció de classificació amb regressió logística.
- Mitja global de la puntuació anterior segons el número de comentaris.
- Un factor de penalització, per a controlar la disparitat del número d'opinions i la representació de la seva mitja.

El número d'opinions en total que ha rebut l'hotel en qüestió és un aspecte crític. Hi haurà hotels amb molts comentaris i d'altres que en tindran molt pocs, per això s'aplicarà "Damped Mean", per harmonitzar aquests casos en general dins la pròpia fórmula per el càlcul de la puntuació de cada hotel.

(Equació 1) El ***rating*** és la predicció de la xarxa i el ***sentiment*** la categoria de classificació segons Random Forest, positiva (+1) o negativa (-1). ***N*** són el número de comentaris, i el factor de penalització és representa com ***k***. La mitja global de comentaris ($\boldsymbol{\mu}$) es calcula com el número d'opinions de l'hotel dividit per el número total d'hotels.

$$\frac{\sum_{i}^{N}\left(\frac{rating \cdot sentiment}{dies}\right) + k \cdot \mu}{N + k} \qquad (1)$$

4. Desenvolupament i resultats

Per aplicar les tècniques més comuns en el processament del llenguatge natural s'ha fet servir la llibreria NLTK (Natural Language Tool Kit) [23], que proveeix les eines necessàries i pot fer-se servir en llenguatge Python.

El tractament vectorial i anàlisi matricial de les dades s'ha realitzat amb les llibreries de Pandas i Numpy, que proporcionen una estructura de dades flexible i permeten treballar amb elles de forma eficient, de manera similar a com ho fan les bases de dades relacionals. El sistema presentat està estructurat en diferents arxius de Notebook.

4.1 Requisits previs

4.1.1 Scraping

Prèviament, per recavar les dades de Tripadvisor s'ha fet una inspecció de com funciona el portal a nivell intern, per saber com està dissenyada la seva estructura, on està emmagatzemada la seva informació i quines són les crides a la base de dades de Tripadvisor que fa que es retorni i mostri per pantalla la informació fent servir els mecanismes automàtics amb PhantomJS i Selenium.

En concret, s'establirà una connexió amb la web de Tripadvisor que mostrarà els hotels que pertanyen a una ciutat. Els resultats es mostren de 30 en 30, per tant, caldrà anar avançant en la paginació. Per a cada hotel, s'agafarà la seva URL d'entrada i s'accedirà a les seves opinions. Ara les opinions es presenten de 5 en 5, novament caldrà avançar fins que hi hagi opinions, si es el cas.

El problema es presenta en el moment que una opinió és molt extensa i a la pàgina no és visible tota aquesta informació. Per això caldrà sol·licitar al servidor que mostri tota aquesta informació i esperar a que aquesta sigui visible en pantalla. Lògicament quant s'apliquen aquests sistemes automàtics pot haver-hi problemes de connexió amb internet o altres que fan que la pàgina no carregui de cop o s'hagi d'esperar a que acabi de carregar-se el seu contingut, en aquest cas es faran servir sistemes d'espera per assegurar que es captura informació i el sistema sigui el màxim d'eficient possible. De cada opinió es capturarà la seva corresponent valoració numèrica, text d'opinió i addicionalment per les opinions de Barcelona, la data en que es va fer pública la critica.

4.1.2 Bases de dades

Per entrenar el sistema de classificació, s'ha utilitzat una base de dades que està formada per les opinions dels hotels de les principals ciutats espanyoles, formant en conjunt un total de 550.000 crítiques, en llengua anglesa. Per a cadascuna s'inclou la valoració numèrica d'aquesta i el text d'opinió.

La part predictiva s'ha realitzat sobre el conjunt d'opinions que formen en total els quasi 500 hotels de Barcelona, formant en total unes 210.000 opinions amb la data de publicació i també en angles.

4.1.3 Etiquetatge i Exemples

Tripadvisor classifica les opinions dels usuaris en 5 categories, segons una valoració numèrica de 10 a 50 que prèviament l'usuari ha indicat amb un número d'estrelles (Figura 12).

Brilliant! ★★★★★

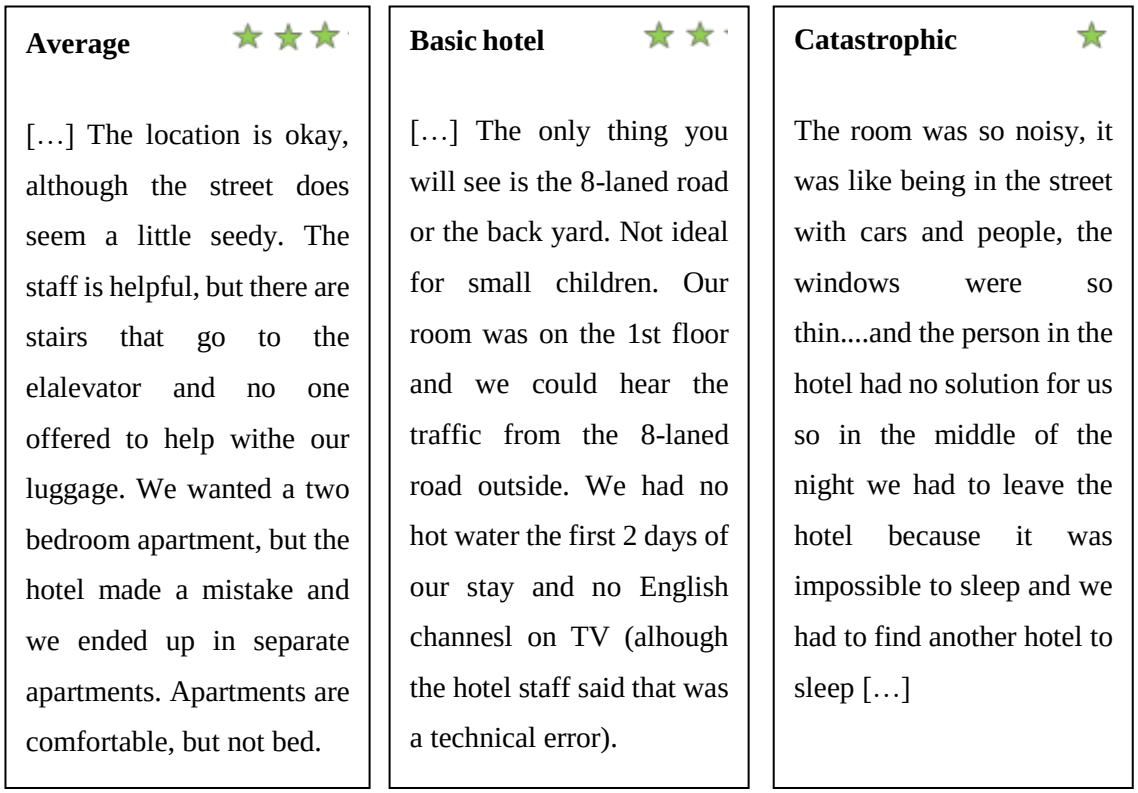
Myself and my girlfriend stayed here for 3 nights over a weekend. The location was excellent and the view of the palace was Brilliant. All of the staff were very friendly and the room was always kept clean and tidy. Would highly recommend and would defianley visit again!

a) Opinió "***Excel·lent***"

Recommended ★★★★

Lovely hotel, room with a view of the palace, all staff so friendly and helpful, great breakfast, little things like happy to print stuff for you, store your bags all no problem no cost to them, small things but mean a lot, and central too with nice restaurants near by […].

b) Opinió "***Molt bona***"



c) Opinió "***Normal***" d) Opinió "***Pobre***" e) Opinió "***Terrible***"

Figura 12: Exemple d'opinions de Tripadvisor segons la seva valoració.
Font: Opinions extretes de Tripadvisor

A nivell general els comentaris dels usuaris són més positius que negatius. Durant el procés d' Scraping més d'un 75% de les opinions són positives (categories "*Excel·lent*" i "*Molt bona*"). En aquest projecte es considera que l'usuari tendeix a posicionar-se en aquestes 2 primeres categories si la seva valoració realment és bona (Figures 12a i 12b). La resta de categories s'hi denoten problemes o inconvenients i són comentaris amb una crítica més negativa que positiva, que mostren que l'usuari no s'ha quedat del tot satisfet (Figures 12c, 12d i 12e).

Per modelar el sistema classificador, de les 5 categories s'ha passat a 2. Essent les valoracions 50 i 40 com a positives i la resta com a negatives. El principal motiu d'aquesta modificació ha estat la poca diferenciació a nivell semàntic i lingüístic de les opinions de cada categoria com per dividir-ho en tantes, donat que simplement es tracta d'analitzar a nivell general si aquella opinió denota un sentiment bo o dolent.

Així llavors, es considera que la resta de categories pertanyen a comentaris negatius (valoracions 30,20 i 10). En un primer moment, les opinions corresponents a la categoria "*Normal*" (valoració 30) es van etiquetar amb una etiqueta *neutre.*

Un cop entrenat el sistema amb 3 possibles categories, *positives, negatives i neutres*, la tassa d'encert dels diferents models entrenats era molt baixa i finalment es va decidir per deixar-ho en dues categories, *positives i negatives*. A l'apartat de resultats es justificarà aquesta decisió.

4.2 Models

4.2.1 Anàlisi de Sentiment

Models utilitzats per dur a terme la classificació de les opinions:

- Random Forest (*ensemble Decision Trees*)
- Naive Bayes (*probabilístic model*)
- Regressió logística (*linear model*)

Aquests s'utilitzaran per entrenar fent servir diferents tipus de vectors característics i comprovar amb quina representació de les dades s'adapten millor els classificadors.

4.2.1.1 BoW-Tf-idf

Després de netejar el *corpus* de dades seguint les passes vistes anteriorment en l'apartat (3.2), aquest s'ha normalitzat, combinant la tècnica bag-of-words i tf-idf, obtenint un conjunt de les paraules més representatives. El mòdul *Scikit-Learn* de Python, inclou les funcions per dur a terme aquest procés amb els mètodes *CountVectorizer i TfidfTransformer*.

En un primer anàlisi s'han fet servir dades prèviament etiquetades en 3 categories (positiu, neutre i negatiu), els resultats dels classificadors mostren clarament que hi ha una baixa tassa d'encert, en els 3 models en general (Figura 13). Això es deu a la falta de paraules suficientment representatives dins el vector d'entrenament característic per l'etiqueta "neutre".

Els classificadors s'han entrenat amb dades balancejades fent servir 550.000 exemples. El temps d'entrenament de Random Forest amb 100 arbres és considerablement superior que la resta de classificadors amb els paràmetres per defecte. Els resultats d'encert dels 3 sistemes són:

embedding	RandomForest Tassa d'encert	MultinomialNB Tassa d'encert	LogisticRegresion Tassa d'encert
Bag-of-Words +TF-IDF	**57%**	51%	55%

Figura 13: Resultat dels diferents classificador amb 3 categories etiquetades.

En canvi, etiquetant les opinions en positiu i negatiu el sistema ha millorat notablement (Figura 14), es confirma que cadascun dels models ha trobat una millor classificació si tant sols aprèn a classificar-ho en positiu i negatiu, ja que en el moment de cercar patrons durant l'entrenament, les correspondències amb les dades de test s´ajusten més. Altre cop, es fan servir dades balancejades i en aquest cas els tres classificadors es situen en el 80% d'encert en la predicció.

embedding	RandomForest Tassa d'encert	MultinomialNB Tassa d'encert	LogisticRegresion Tassa d'encert
Bag-of-Words +TF-IDF	80**%**	**81%**	79%

Figura 14: Resultat dels diferents classificador amb 2 categories etiquetades.

Segons les N característiques més representatives del conjunt de dades d'entrenament es pot observar que hi ha més paraules amb una semàntica més positiva que negativa. (Figura 15).

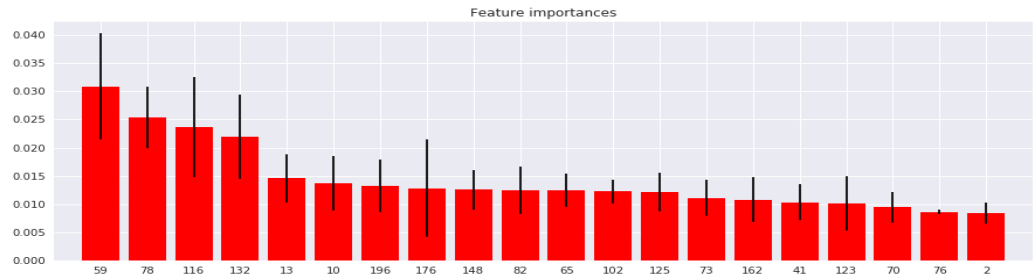


Figura 15: Histograma de les 20 paraules més característiques del conjunt d'entrenament fent servir el model Random Forest. Ordenadament, "***excellent***", "***great***", "***ok***" i "***poor***" són les que més pes tenen.

4.2.1.2 Word2vec

Per aplicar aquesta tècnica d'extracció de característiques s'ha fet servir la llibreria gensim [24], que inclou els mètodes per crear una representació vectorial de les dades segons diferents paràmetres que decidiran la grandària i qualitat del vector característic. Les dades d'entrada són processades prèviament separant les paraules mitjançant "tokens", tal i com s'ha fet per el mètode anterior. En aquest cas l'eliminació de paraules buides o "Stop Words" no s'aplica, donat que aquest mètode d'embedding de dades té en compte el context del text durant l'anàlisi. Freqüentment pot haver-hi casos on precisament siguin els articles o qualque d'aquestes paraules més comuns que facin que el sistema pugui aprendre millor.

Els paràmetres crítics per crear el model són:

- **Word vector dimensionality***:* Mida per representar cada paraula com un vector, com més gran sigui, més temps d'execució requerirà l'algorisme, en canvi, no sempre s'obtindrà un millor model de representació.

- **Minimum word count:** Número de cops que ha d'aparèixer la paraula d'entre totes les opinions per afegir-la dins el vocabulari, qualsevol paraula per sota d'aquest número d'aparicions no es tindrà en compte. Aquest valor ajuda a reduir el número total de paraules del vocabulari.

- **Context/Window Size***:* Es el número de paraules que s'agafarà per representar el context.

De les dades d'entrada es creen subconjunts de paraules diferents per representar i aprendre el context. L'algorisme crea diferents mostres d'entrenament combinant les paraules que es troben dins el marge (Figura 16), aquest analitza segons la repetició de les mostres per trobar característiques importants que defineixin un patró.

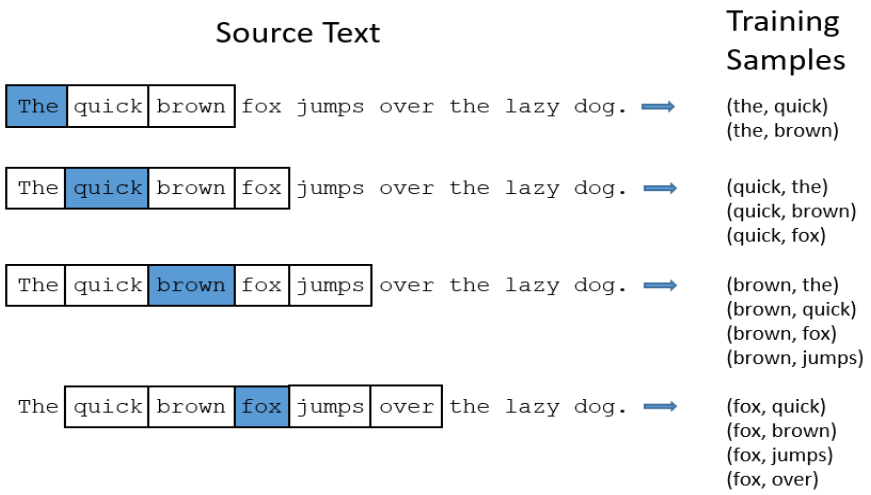


Figura 16: Creació de les mostres d'entrenament amb Window igual 2.

Font: Tutorial - word2vec from Chris McCormick

Els valors que s'han fet servir respectivament són 300x40x10. La dimensió del model final influirà en el temps de càlcul que requereix la creació dels corresponents vectors. Les dades que s'han fet servir per entrenar el sistema són les mateixes que s'han fet servir anteriorment, etiquetades amb 2 categories, positiu i negatiu. Amb aquesta representació, els models classificadors prediuen millors resultats d'encert (Figura 17).

En gran part sorprèn com el model de regressió logística és el que obté una millor tassa d'encert, ja que es tracta d'un model molt simple en contrapartida de Ranfom Forest, que és més robust. S'intueix en aquest cas que les dades que s'han fet servir per entrenar el model són suficientment representatives com per a que la regressió sigui linealment separable en dues categories.

embedding	RandomForest Tassa d'encert	GaussianNB Tassa d'encert	LogisticRegresion Tassa d'encert
Word2vec	84%	80%	**88%**

Figura 17: Resultat d'encert dels classificadors fent servir Word2vec

El conjunt de dades que s'ha fet servir per testejar el sistema es correspon en tots els casos al conjunt de dades de Barcelona, que està composta per 200.000 opinions i el vocabulari obtingut pel model més o menys està compost per 17.000 paraules.

Fent servir el classificador de regressió logística per classificar el sentiment de cada opinió en positiva o negativa, es mostren alguns exemples de les prediccions del sistema en diferents casos d'encert i error (Figura 18).

"Beautiful Boutique Hotel in the Heart" label: 🟢 predicted: 🟢

This is a beautiful hotel in the centre of Barcelona. The room was elegant and spacious. Rooftop terrace restaurant and bar offer panoramic views and excellent food. The breakfast offered in the second floor restaurant was first class too […]

"Disgusting" label: 🔴 predicted: 🔴

This apartment is part of another apartment, therefore all smells and noises are only separated by an internal door, there is no outside window so lights have to be on constantly and are switched off by the other. If there was a fire in the hallway, there would be no escape due to lack of outside window. It's cheap but not worth the risk.

a) Classificació **correcte**. A dalt una opinió classificada com a positiva i a sota una negativa, en els dos casos, tant l'usuari com el sistema coincideixen en el sentiment que expressa l'opinió.

"Excellent hotel but for safety issues" label: 🔴 predicted: 🟢

Rooms decent-sized, clean, relatively new. Excellent breakfast. Bathrooms are room, HOWEVER, something should be done about the slippery ceramic tile floors in the bathroom - it's extremely easy for water to get on the floor because the bath/shower only has a half glass wall - no shower door or curtain covering the length of the tub - and the hand towels are hung under the sink. VERY unsafe. […]

"Calm in the city that never sleeps!" label: 🟢 predicted: 🔴

Fantastic Situation! Room well appointed with large bathroom. You wouldn't know you were in the middle of a city that never sleeps. We got home after a busy day, and slept well in the air conditioned quiet room. The buffet breakfast selection was large, and as a gluten- intolerant I asked for G-F toast WHICH THEY HAD! So can't thank them enough - as not all places have even heard of it.

b) Classificació **errònia**. A dalt, l'usuari valora la seva opinió com a negativa, en canvi el sistema ho classifica com a positiu, per tant és un fals positiu. A sota el cas invers com a fals negatiu. En els dos casos està clar que s'ha classificat malament, segons el seu contingut.

“good location” label: 🟢 predicted: 🔴
Rooms facing the street are noisy because of traffic. Lighting system very design but too difficult to use. Air conditioning system inefficient. Excellent location near main attractions. Staff very helpful and friendly.
“Good value” label: 🔴 predicted: 🟢
Prefer to be closer to Plaza Catalunya, a 30 minute walk or less by metro. Close to metro and bus stop. Hop on hop off bus stops nearby. Close to plenty of shopping. No ice machine but ice available from cafeteria during open hours. No wash cloths so we purchased shower loofah from euro store. No in room coffee machine but Costa coffee shop next door. Staff was courteous and very helpful. Overall an enjoyable experience.

c) Classificació **errònia**. En aquest cas, en els dos exemples pot haver-hi ambigüitats en el text que hagin provocat la mala classificació.

Figura 18: Exemples classificats amb word2vec com a embedding de dades i regressió.

4.2.2 Representacions

La diferència amb la tassa d’encert en la classificació que hi ha fent servir una tècnica d’embedding o altre, es pot veure justificat en com estan distribuïdes les dades en un espai de representació 2D o 3D amb la tècnica T-SNE.

Els sistemes de classificació que han estat entrenats amb el vector característic *bag-of-words + Tf-idf* mostren una baixa tassa d’encert en la classificació, això es deu a que les paraules que s’han fet servir per crear el vector no són atribuïbles a una categoria positiva o negativa en concret, solament té en compte la freqüència d’aparició de les paraules que formen el conjunt de paraules representatives (Figura 19). En canvi, amb el mètode *word2vec* aquesta visualització té més sentit i s’observa com els punts de cada color estan més agrupats (Figura 20 a). El motiu és que les paraules que conformen el vocabulari s’han obtingut segons el context del text i poden ser associades amb més o menys encert cap a una polaritat positiva o negativa per predir amb millor encert les opinions.

En tots els casos els resultats que es mostren amb T-SNE els punts es corresponen a les mateixes opinions de test que han estat predites pel sistema, en concret es mostren 1000 resultats de classificació positiva i 1000 resultats de classificació negativa.

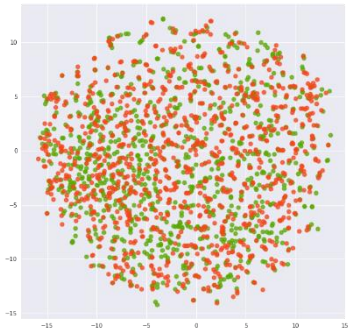
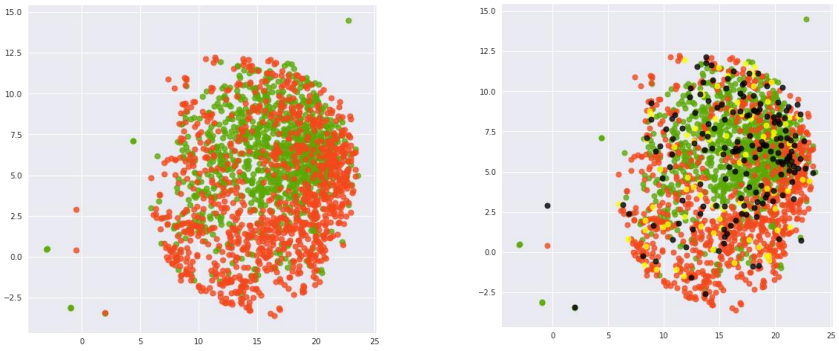


Figura 19: Visualització amb T-SNE dels resultats de classificació i BoW+TF-IDF com a embedding de dades i prediccions del sistema amb Random Forest.

Els comentaris dels usuaris que s'han analitzat tendeixen a ser més positius que negatius, també en el cas de les opinions negatives aquestes són més pobres semànticament. Com es pot apreciar (Figura 20 b), els punts negres es corresponen a les valoracions que han estat mal classificades, majoritàriament coincideixen amb les valoracions negatives per part dels usuari i on la classificació del sistema és positiva (fals positiu). Els punts grocs es corresponen a les valoracions positives del usuaris que el sistema ha classificat de manera negativa (fals negatiu). En general hi ha més errors en la classificació quant l'usuari ha puntuat la opinió de manera baixa i el sistema ho a predit positivament.



a) Classificació b) Classificació errònia

Figura 20: Visualització amb T-SNE dels resultats de classificació i word2vec com a embedding de dades fent servir regressió logística com a sistema predictiu.

Els resultats de classificació de sentiments amb la xarxa neuronal mostren com les dades estan més agrupades (Figura 21), el que permet visualitzar-ho amb 3 dimensions més clarament, al contrari que els casos anteriors fent servir els mètodes clàssics de classificació, on els resultat no estaven suficientment diferenciats a l'espai de representació.

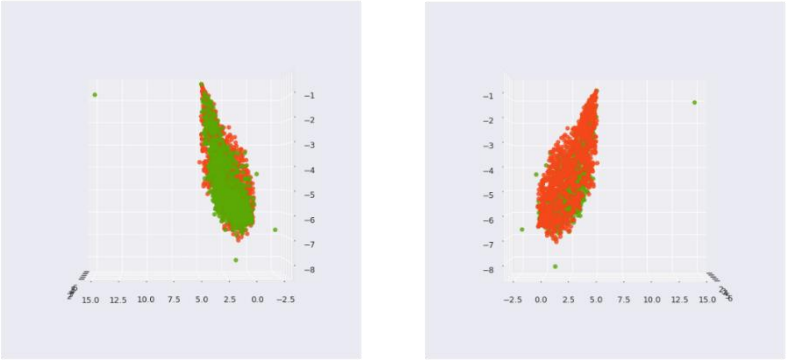


Figura 21: Visualització 3D amb T-SNE dels resultats de classificació i un embedding de dades resultat de la sortida de la darrera capa densa. A la dreta es pot visualitzar segons la orientació en l'eix XY (no hi ha rotació), i a l'esquerra amb una rotació de 180º, eix ZY. Per tant hi ha un pla de projecció on les dades de les dues classes són suficientment diferenciables a l'espai.

4.2.3 Regressió i puntuació

En aquest cas no es tracta d'una classificació lineal, sinó d'un problema de regressió fent servir una xarxa neuronal. Per incloure més informació i complexitat en el càlcul de la puntuació d'un hotel, es farà servir una predicció numèrica en un rang de 10 a 50 mitjançant regressió, fent servir *Keras*. La puntuació numèrica que l'usuari va donar a la seva opinió servirà com a etiqueta per entrenar el model de la xarxa i que aquest trobi la millor predicció.

El percentatge d'encert de la xarxa ve determinat pel marge estimat entre la regressió predita i el valor real de cada opinió, per tant, es pot veure que a mesura que s'incrementa el marge la tassa d'encert creix (Figura 23). El marge ha de ser coherent per no sobreestimar el sistema, per tant en aquest cas la tassa d'encert variarà en funció de com estrictes es sigui amb el marge.

Cal recordar que les puntuacions etiquetades prenen valors de 10-50 i que aquestes s'han normalitzat entre 0 i 1 per la xarxa, per tant els marges del gràfic anterior es corresponen a un valor de marge 0.1, sent un marge 5 sense normalitzar.

La xarxa està formada per 22 capes on s'hi realitzen diferents operacions entre l'entrada i sortida de cada capa, inicialment aquestes inclouen convolucions per tal de trobar una estructura que representin les dades, uns patrons, tal com per exemple les paraules que sempre surten juntes o les que tant sols apareixen en unes certes dades. Per a que l'estructura anterior sigui tractable per el sistema i s'executi amb un temps raonable s'ha de reduir la seva dimensionalitat, la capa "Polling" especialment realitza aquesta operació. Les capes denses incrementen el número de paràmetres possibles per donar expressivitat al model i generalitzar el problema d'aprenentatge, segons el número de neurones de cada capa "Dense" es trobaran altres característiques representatives, que no s'havien trobat fins el moment i que permeten una variabilitat i millora en el model predictiu. El temps d'execució per a cada iteració de la xarxa, "epoch", és d'uns 10 minuts fent servir la GPU.

La xarxa aplica una regressió per estimar la puntuació predita, per tant aquesta va associat a un cert "error", per tal de calcular-lo s'ha fet servir "MSE", que mesura aquest error en el conjunt del sistema. Com a optimitzador de la xarxa s'utilitzarà l'algorisme de gradient descendent Adam per tal de propagar l'error, el qual té *Keras* per defecte i més s'utilitza a l'estat de l'art.

La tendència inicial de la xarxa mostra com aquesta s'adapta bé a les dades del conjunt de test i baixa el percentatge d'encert o *accuracy*, a mesura que s'itera sobre el sistema veient novament les dades en general, la corba del conjunt de dades d'entrenament en canvi puja, el qual és un problema d'overfitting. Aquest s'ha solucionat fent servir el model en el punt el qual la xarxa generalitza millor. Com es pot veure (Figura 22) a la dècima iteració, després del que ja ha aprés, és té el model que millors resultats prediu, desestimant els demés models. Aquesta tècnica es coneix com a early stopping.

Una altre solució per rebaixar el problema d'overfitting és reduir la dimensió del problema, amb menys paràmetres baixant per exemple el número de neurones de les capes "Dense".

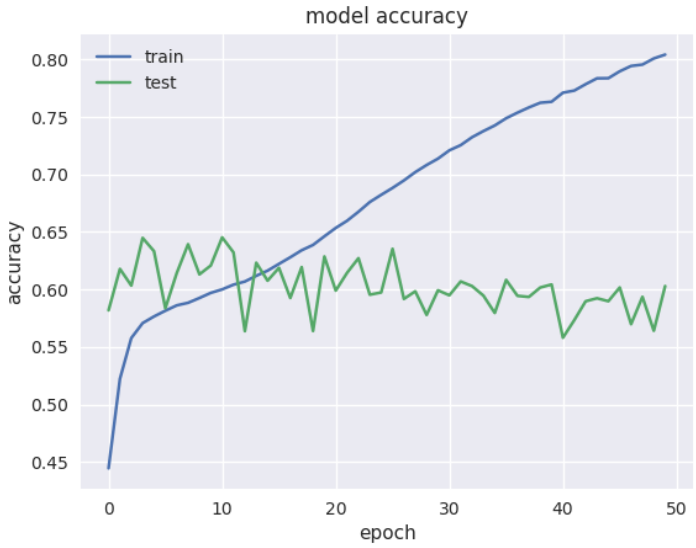


Figura 22: Evolució del percentatge d'encert per 50 "epochs", que són els cops que s'entrena la xarxa fent servir les mateixes dades i amb un número de paràmetres diferents. L'eix X representa el número d'iteracions o "epochs" i l'eix Y el percentatge d'encert.

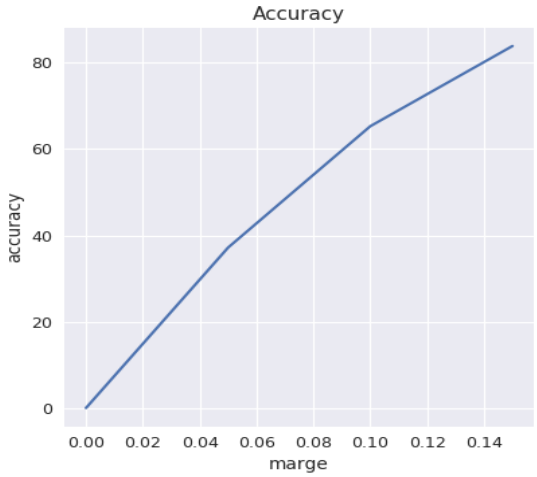


Figura 23: Increment d'encert de la xarxa segons el marge.

En aquest cas la xarxa arriba a un 65% d'encert, amb un marge de confiança de 5. En la (Figura 23) es pot veure que aquest percentatge d'encert creix conforme es fa gran aquest marge.

Les alteracions en la corba de les gràfiques venen donades pel fet que s'està aplicant una regressió on les dades d'entrada originals prenen els valors, 10, 20, 30, 40 i 50. En comparar les prediccions amb aquests valors fa provocar aquest soroll a la corba d'encert i error. L'error comès a partir de la iteració 10 mostra com tendeix a pujar (Figura 24) amb les mateixes dades de test de manera considerable quasi a la meitat, mentre que per les dades d'entrenament, a la darrera iteració l'error comès tendeix a 0.

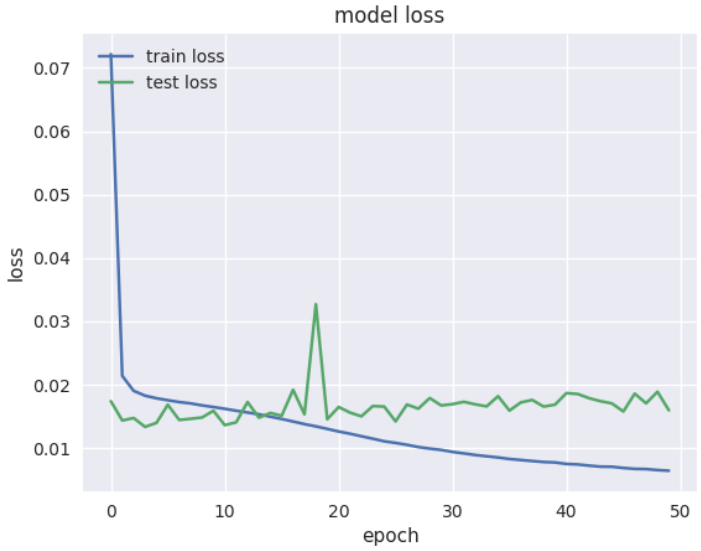


Figura 24: Iteració de l'error comès per la xarxa. Es mostra l'overfitting en la iteració 10 on les dades de test tenen millors resultats, menys error que amb les dades d'entrenament.

5. Mapa

El posicionament en el mapa de cadascun dels hotels analitzats es realitza mitjançant un "marcador o etiqueta" circular associat a un color (Figura 25), segons la puntuació calculada amb la fórmula vista anteriorment, en un rang de 0 a 100.

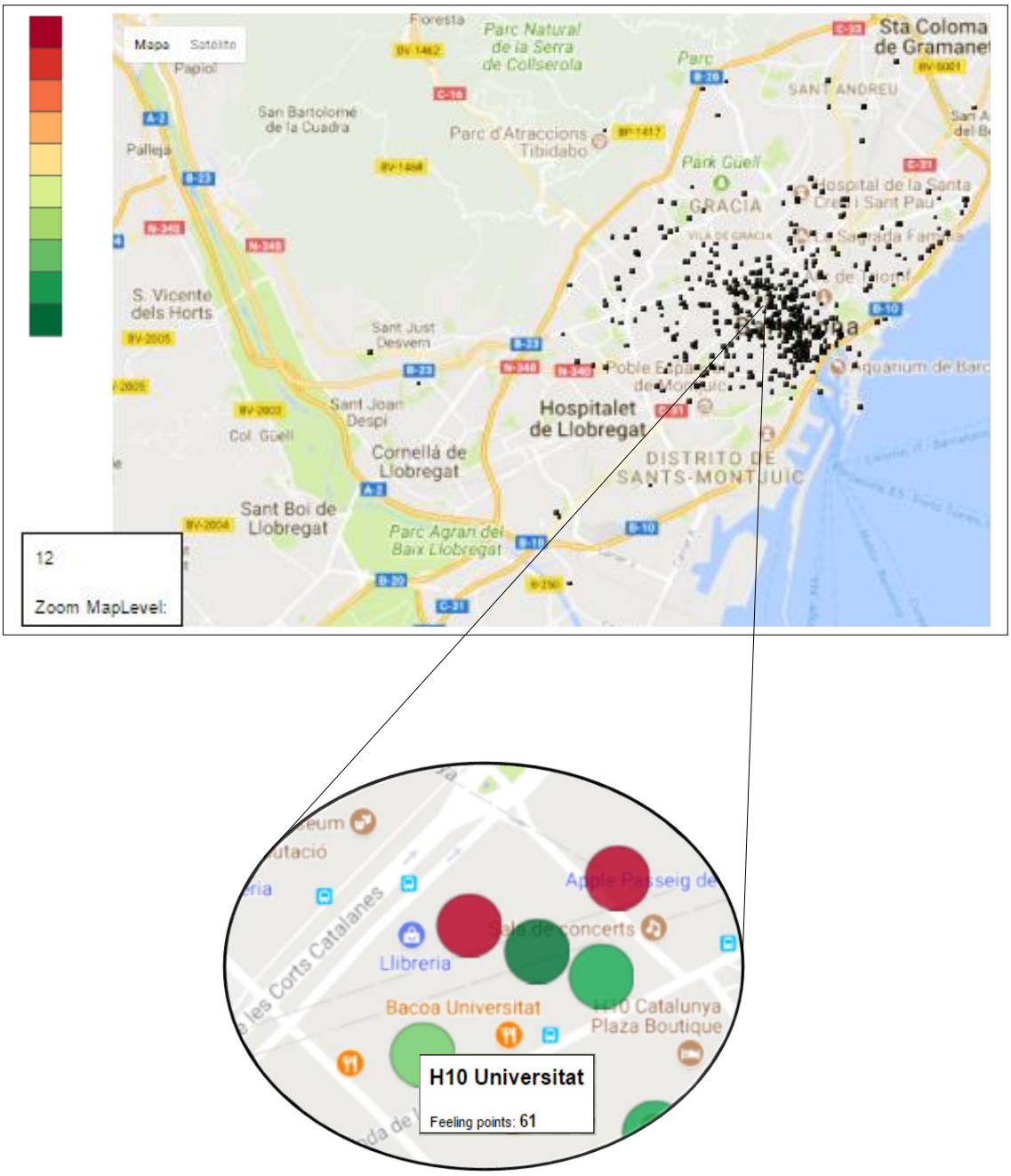


Figura 25: Prototipus del mapa per la implementació via web.

Els colors conformen una paleta que està composta per 10 tonalitats, intuïtivament els colors vermells es corresponen a puntuacions baixes (0-50) i els tons verds a puntuacions més bones o molt bones (51-100).

Per nivells baixos de zoom (vista del mapa més amunt), els marcadors poden estar sobreposats, això ocorre quan hi ha molta proximitat entre els hotels (Figura 26). En aquest cas s'ha escollit que el que està per sobre sigui el millor valorat de l'entorn.

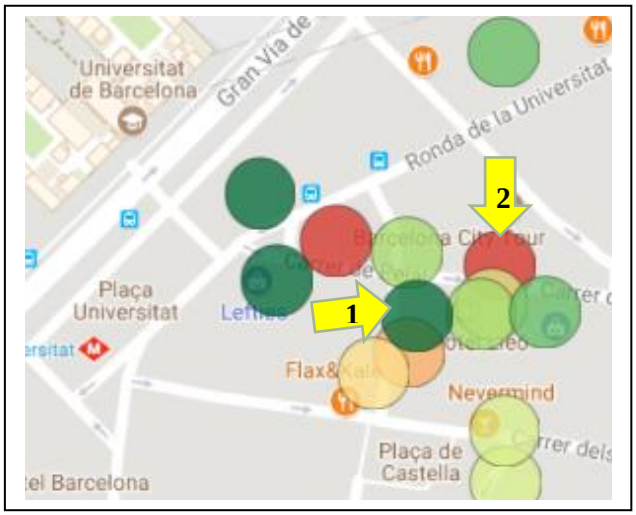


Figura 26: Solapament de marcadors en el mapa. 2 exemples. Es veu com l'hotel més ben puntuat (1) es sobreposa a la resta si aquests es troben en un radi proper. El cas contrari (2) s'observa com l'hotel menys valorat està per sota.

El mapa està disponible a la web: ***http://www.pushedx.net/map.html***

6. Conclusió i discussió dels resultats

Hi ha diverses maneres de resoldre problemes de classificació de sentiments en textos i ressenyes de productes o serveis, en aquest treball s'han utilitzat els sistemes de classificació supervisada clàssics i les xarxes neuronals supervisades per classificar opinions segons positives o negatives.

Aquests sistemes han permès un canvi de paradigma, on es passa d'analitzar no només taules de valoracions numèriques com a indicador de com de bo o dolent és una marca, producte o servei, a analitzar-se el sentiment d'un text per a conèixer millor la positivitat o negativitat que expressa aquella valoració.

S'ha aconseguit de manera satisfactòria recavar les dades necessàries de Tripadvisor, que han permès crear un sistema que aconsegueix un 88% d'encert en la classificació dels sentiments. La comparativa de resultats amb els diferents mètodes d'embedding utilitzats, mostra com la representació word2vec és millor que no pas els mètodes més tradicionals, com BoW. El fet que Tripadvisor permeti a l'usuari donar una valoració numèrica a la seva ressenya i utilitzar aquesta com a etiqueta prèvia per entrenar els models de classificació, ha demostrat en aquest experiment que els usuaris tendeixen més a expressar una connotació positiva o negativa en els seus comentaris que no pas un estat de neutralitat. S'entén que un usuari li ha agradat o no la seva estància i així ho mostren els resultats d'anàlisi de sentiments fent servir sistemes de visualitzacions de característiques amb T-SNE.

El resultat de predicció de sentiment en positiu i negatiu amb els mètodes clàssics de classificació i el resultat del model predictiu de puntuació amb *Keras* fent servir la regressió, s'han unit a la fórmula de "Damped Mean" per donar-li uniformitat als resultats tenint en compte aquells hotels que tenen moltes més crítiques i els que en tenen menys. La puntuació que s'ha assignat a cadascun dels hotels de Barcelona pot veure's influenciat segons les seves ressenyes. Tripadvisor disposa de filtres que permeten esborrar comentaris d'un hotel quan es creu que s'han escrit de manera fraudulenta, cau la possibilitat doncs, que durant el procés de predicció s'hagin trobat aquests casos.

Amb les API's de Google s'ha implementat la interfície mapa, que mostra els resultats d'anàlisi que s'han fet per els hotels de Barcelona on es pot observar en grans trets la concentració dels hotels més ben valorats en el centre de la ciutat i no pas a la perifèria. En comú quasi totes les opinions fan referència a la ubicació de l'emplaçament en qüestió, així, s'entén que els hotels que es situen en el centre en aquest aspecte són positius sense comptar-hi altres factors.

S'ha establert l'objectiu que es va proposar a l'inici del projecte i en general estic content d'haver realitzat aquest treball. La recerca d'informació ha estat un aspecte molt important durant tot el projecte i ha servit per conèixer més a fons sobre quins són els nous desenvolupaments que s'estan realitzant en aquest camp d'anàlisi de sentiments, en textos escrits i pujats a la web per usuaris corrents. El fet de poder fer servir un servidor de la universitat per poder realitzar-hi les proves, sobre tot les corresponents a la xarxa ha estat fonamental per tal de poder disposar d'un equip amb prestacions GPU, fent que el temps d'execució hagi estat increïblement menor per obtenir els resultats i poder testejar amb diferents paràmetres i capes de la xarxa per veure com s'ajusta millor el sistema.

6.1 Treball futur

L'arquitectura del sistema presentat pot ser utilitzat en altres àmbits sense necessitat de molts canvis, així doncs es pot fer servir per altres propòsits similars. Per plataformes turístiques pot ser interessant la seva implementació, no solament de cara a mostrar als usuaris que pensen els demés sinó per millorar a nivell intern les seves competències professionals. En concret, amb el mateix sistema es pot analitzar els comentaris del mateix portal de Tripadvisor sobre referències de Restauració.

Les crítiques obtingudes estan escrites per usuaris de la web, aquestes dificulten la tasca d'aprenentatge, donat que poden tenir errors ortogràfics o expressions informals, una alternativa és utilitzar un *corpus* de dades escrit prèviament per professionals o ja etiquetat per alguna institució per entrenar el sistema. Per altre banda seria interessant recavar pels mateixos hotels comentaris d'altres webs com Booking.com i comparar-ne el resultat, o utilitzar la mitja amb els dos llocs.

Com a possibles millores del sistema es pot incloure informació semàntica en l'anàlisi de text, fent servir recursos com SentiWordNet [25] i adaptar el sistema per a que permeti la classificació de la polaritat d'un text subjectiu en altres idiomes.

Pel que fa el servei mapa web pot afegir-se la possibilitat que l'usuari canviï el paràmetre de penalització per el càlcul de la puntuació de cada hotel, el que permet per exemple descartar aquells que tenen poques opinions si aquest número és molt alt, o determinar altres paràmetres com mostrar resultats en base a paraules clau que l'usuari introdueixi. La interacció amb el mapa es pot millorar afegint un control per la grandària dels punts representats en funció del nivell de Zoom i podent clicar a sobre de cada hotel per mostrar informació complementària com les darreres opinions o fotografies del lloc.

Bibliografia

[1] Tripadvisor. (2016, May). *Tripadvisor*. [online] https://www.tripadvisor.co.uk/TripAdvisorInsights/n2701/changes-tripadvisor-popularity-ranking-algorithm (2017, February)

[2] Bo Pang and Lillian Lee (2008), "Opinion Mining and Sentiment Analysis", Foundations and Trends® in Information Retrieval: Vol. 2: No. 1–2, pp 1-135. http://dx.doi.org/10.1561/1500000011

[3] Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. Language resources and evaluation, 39(2), 165-210.

[4] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[5] Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 417-424). Association for Computational Linguistics.

[6] Ren, Y., Zhang, Y., Zhang, M., & Ji, D. (2016, February). Context-Sensitive Twitter Sentiment Classification Using Neural Network. In AAAI (pp. 215-221).

[7] Radford, A., Jozefowicz, R., & Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. arXiv preprint arXiv:1704.01444.

[8] Herrouz, A., Khentout, C., & Djoudi, M. (2013). Overview of web content mining tools. arXiv preprint arXiv:1307.1024.

[9] Seleniumhq.org. (2017). Selenium Documentation- Selenium Documentation. [online] http://www.seleniumhq.org/docs/ (2017, Febrauary)

[10] Sorens, M. (2017). CSS, DOM and Selenium: The Rosetta Stone - Simple Talk. [online] https://www.simple-talk.com/wp-content/uploads/imported/1269-Locators_table_1_0_2.pdf (2017, Febrauary)

[11] Phantomjs.org. (2017). Quick Start | PhantomJS. [online] http://phantomjs.org/quick-start.html, (2017, Febrauary)

[12] Wu, H. C., Luk, R. W. P., Wong, K. F., and Kwok, K. L. 2008. Interpreting TF-IDF term weights as making relevance decisions. ACM Trans. Inform. Syst. 26, 3, Article 13 (June 2008), 37 pages. http://doi.acm.org/ 10.1145/1361684.1361686

[13] Kaggle 2014. Bag of Words Meets Bags of Popcorn, Use Google's Word2Vec for movie reviews. [online] https://www.kaggle.com/c/ word2vec-nlp-tutorial

[14] TensorFlow. (2017). Vector Representations of Words | TensorFlow. [online] https://www.tensorflow.org/tutorials/word2vec (2017, April).

[15] Sebastian Raschka's Website. (2017). Principal Component Analysis. http://sebastianraschka.com/Articles/2015_pca_in_3_steps.html (2017, April).

[16] Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

[17] Wattenberg, M., Viégas, F. and Johnson, I. (2017). How to Use t-SNE Effectively. http://distill.pub/2016/misread-tsne/

[18] Nltk.org. (2017). Chapter 6. Learning to Classify Text. [online] http://www.nltk.org/book/ch06.html (2017, March)

[19] Nlp.stanford.edu. (2017). Naive Bayes text classification. [online] https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html (2017, March)

[20] Scikit-learn.org. (2017). scikit-learn: machine learning in Python — scikit-learn [online] documentation. http://scikit-learn.org/stable/

[21] Keras.io. (2017). Keras Documentation. https://keras.io/ [online]

[22] Google Developers. (2017). Guía del desarrollador | Google Maps Geocoding API | https://developers.google.com/maps/documentation/geocoding/intro?hl=es-419 [online]

[23] Nltk.org. (2017). nltk.classify.scikitlearn — NLTK 3.2.4 documentation. [online] http://www.nltk.org/_modules/nltk/classify/scikitlearn.html

[24] Radimrehurek.com. (2017). gensim: topic modelling for humans. [online] http://radimrehurek.com/gensim/

[25] Ohana, B., & Tierney, B. (2009). Sentiment Classification of Reviews Using SentiWordNet. 9th. IT&T Conference, Dublin Institute of Technology, Dublin, Ireland. 22nd.-23rd. October.