UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRAU D'ENGINYERIA INFORMÀTICA

TREBALL FINAL DE GRAU

APRENENTATGE AUTOMÀTIC APLICAT A LA VALORACIÓ D'ALLOTJAMENTS D'AIRBNB

Autor: Joan Orteu Saiz

Director: **Dr. Santi Seguí**
Realitzat a: **Departament de Matemàtiques i Informàtica**

Barcelona, **17 de gener de 2024**

Índex

Abstract

The evaluation of accommodations on the Airbnb platform is a critical factor for both travelers and hosts. In this study, we delve into how machine learning models can predict and contribute to improving the customer experience. Initially, we propose a solution to address the question "what improvements can be made to an accommodation to increase its rating and what will be the resulting gain?". The goal is to provide useful information to hosts, enabling them to make informed decisions to enhance the quality of their services and, consequently, elevate customer satisfaction and ratings.

This study tackles a fundamental part of this solution: obtaining data, preprocessing, and training predictive models to anticipate accommodation ratings.

Through this research, we explore the possibilities at the intersection of artificial intelligence and the shared accommodation industry. We contribute to the growing field of practical application of machine learning, laying the groundwork for future improvements and innovations in this domain.

Resum

La valoració dels allotjaments a la plataforma Airbnb és un factor crític per als viatgers i amfitrions. En aquest treball, aprofundim en com els models d'aprenentatge automàtic poden predir i contribuir a millorar l'experiència dels clients. Inicialment, proposem una solució per respondre a la pregunta "quines millores es poden realitzar a un allotjament per augmentar la seva valoració i quin serà el guany resultant?". L'objectiu és proporcionar informació útil als amfitrions, permetent-los prendre decisions informades per millorar la qualitat dels seus serveis i, per consegüent, elevar la satisfacció dels clients i les seves valoracions.

Aquest estudi aborda una part fonamental d'aquesta solució: l'obtenció de dades, el preprocessament de les dades i l'entrenament de models predictius per anticipar les valoracions dels allotjaments.

Mitjançant aquesta recerca, explorem les possibilitats que ofereix la intersecció entre la intel·ligència artificial i la indústria de l'allotjament compartit. Contribuïm al creixent camp de l'aplicació pràctica de l'aprenentatge automàtic, posant les bases per a futures millores i innovacions en aquest àmbit.

Capítol 0

Estructura de la Memòria

1. **Introducció:** S'introdueix la temàtica i la problemàtica que es tracta-da en aquest treball. Es descriu la motivació i es planteja què es vol aconseguir amb el treball, els objectius i les dades que s'utilitzaran.

2. **Planificació:** S'expliquen els detalls de la planificació inicial d'aquest projecte, incloent-hi les temporitzacions de les tasques, i es compara amb el desenvolupament real que ha tingut.

3. **Base de Dades:** Es detalla el procés d'extracció de les dades i s'explora el seu contingut mitjançant una anàlisi descriptiu.

4. **Metodologia:** S'aborden les tècniques específiques de processament de dades per a cada tipus d'atribut. Es presenten les transformacions previstes i s'analitza el procés d'entrenament de models.

5. **Results:** Es detalla el mètode utilitzat per validar els mètodes entrenats i es presenten les conclusions obtingudes a partir de les proves executades.

6. **Conclusions i futur treball:** Es realitza una síntesi dels resultats obtinguts i es presenten les conclusions derivades de la recerca i les proves. S'analitzen les implicacions dels resultats i es destaquen possibles direccions futures per a investigacions relacionades.

7. **Annex:** Informació extra sobre el desenvolupament del projecte, com taules i imatges, entre d'altres.

Capítol 1

Introducció

1.1 Airbnb

Airbnb és una empresa que ofereix una plataforma digital dedicada a l'oferta d'allotjaments a particulars i turístics. Per mitjà d'aquesta plataforma, els amfitrions poden publicitar i llogar les seves propietats als hostes. Amfitrions i hostes poden valorar-se mútuament, actuant com a referència per a futurs usuaris. Fundada el 2008, la plataforma s'ha convertit en una de les opcions més populars per a viatgers a nivell mundial.

A través d'Airbnb, els amfitrions poden posar a disposició habitacions, apartaments sencers o altres tipus d'allotjament, oferint als hostes una varietat de opcions úniques i autèntiques. El servei no només permet estades més personalitzades en comparació amb les opcions d'allotjament tradicionals, sinó que també ofereix una àmplia gamma de preus que s'adapten a diferents pressupostos.

Els usuaris poden explorar l'oferta d'allotjament a través del lloc web d'Airbnb o de l'aplicació mòbil, veient fotografies, llegint ressenyes d'altres hostes i comunicant-se directament amb els amfitrions abans de prendre una decisió de reserva.

Amb una presència global, Airbnb ha redefinit la forma en què les persones experimenten els seus viatges, connectant cultures i oferint experiències úniques d'allotjament arreu del món. La plataforma ha obert noves oportunitats tant per als viatgers com per als amfitrions, fomentant la interconnexió i la diversitat en el món del viatge.

1.2 Problema

En el context actual de l'ús generalitzat d'Airbnb com a plataforma d'allotjament, els amfitrions es troben amb el desafiament constant d'oferir una experiència de qualitat als seus hostes per obtenir bones valoracions. Aquestes valoracions no només influeixen en la reputació individual dels amfitrions, sinó que també tenen un impacte directe a la visibilitat del seu allotjament i, per tant, en la capacitat d'atraure nous hostes.

En aquest entorn competitiu, és evident que les valoracions dels hostes són crítiques per als amfitrions. Un aspecte clau que els amfitrions han de tenir en compte és la comunicació efectiva. En molts casos, les experiències negatives es poden evitar o mitigar a través d'una comunicació clara i anticipada amb els hostes. Això pot incloure proporcionar informació detallada sobre l'allotjament, les expectatives de la zona i resoldre qualsevol pregunta o inquietud abans que els hostes arribin. Una falta de comunicació pot conduir a malentesos que, malauradament, es reflecteixen en valoracions negatives i poden afectar la percepció general dels hostes respecte a l'experiència. Per tant, una millora substancial en la comunicació pot ser un factor crucial per elevar la satisfacció dels hostes i, com a resultat, millorar les valoracions i la reputació de l'amfitrió.

És crucial reconèixer que algunes accions poden tenir un impacte més important i immediat en les valoracions dels hostes. La pregunta central que es planteja és: quines millores es poden realitzar a un allotjament per a augmentar la valoració i quin serà el guany resultant?

1.2.1 Solució Actual

En el mercat actual, existeixen diverses solucions com AirDNA [5] que ofereixen eines per a la gestió i optimització d'allotjaments a través de l'anàlisi de dades i la comparació amb la competència. Aquestes solucions tenen com a objectiu principal l'augment del benefici monetari, proporcionant als amfitrions informació detallada sobre els preus òptims per a la seva propietat i altres estratègies financeres per maximitzar els ingressos.

No obstant, una limitació crucial d'aquestes solucions radica en la seva manca d'abordatge holístic cap a la millora de l'experiència de l'hoste. Aquestes eines es centren en les variables financeres i de rendiment econòmic, i deixen de banda altres aspectes essencials que afecten directament les valoracions dels hostes.

En aquesta perspectiva, la manca de solucions que es centrin específi-

cament en la millora de les valoracions i l'experiència global dels hostes crea una oportunitat per a un enfocament més integral. És vital reconèixer que la satisfacció dels hostes va més enllà de la mera eficiència financera.

Una forma de solucionar el problema és la versió manual: l'amfitrió busca a la pàgina web el que ell considera els seus competidors amb millor valoració i estudia què estan fent que ell no. Un també pot mirar les seves valoracions i solucionar els problemes plantejats, però aquí citaré una frase molt famosa de Henry Ford: "Si els hagués preguntat a la gent què volien, m'haurien dit que volien cavalls més ràpids".

Per tant hi ha una clara oportunitat per fer una eina que doni prioritat a la qualitat del servei i a la satisfacció dels hostes com a objectiu principal; també es podrien entendre aquestes variables com a motor principal del creixement econòmic i sotenible. Aquesta eina donaria informació útil i complementaria a les solucions actuals.

1.2.2 Solució Plantejada

Es proposa una solució innovadora per abordar la millora de les valoracions dels allotjaments a través d'una eina en línia especialitzada. Aquesta eina processaria les dades específiques de cada allotjament i generaria una predicció de la valoració esperada (a través d'un model preentrenat). La solució inclouria un estudi detallat que destaqués la rellevància dels diversos atributs en la predicció final de la valoració (utilitzant tècniques de Inteligència Artificial Explicable com SHAP o LIME).

A més a més, es proporcionaria una anàlisi exhaustiu de les cinc millores més importants que podrien ser implementades per elevar la percepció dels hostes i, per tant, millorar les valoracions. Aquesta aproximació no només proveiria als amfitrions d'eines valuoses per comprendre millor les expectatives dels hostes, sinó que també oferiria orientacions pràctiques per optimitzar l'experiència global d'allotjament i millorar significativament les valoracions rebudes.

1.2.3 Motivació

La motivació darrere d'aquest treball, és aprendre a fer un projecte d'aprenentatge automàtic i es va creure que la millor forma d'aprendreho, seria solucionar un problema real aplicant l'aprenentatge automàtic. S'ha escollit solucionar aquest problema degut a que els meus pares es dediquen al turisme i m'han transmès la seva passió pel sector.

1.3 Objectius

Fins a aquest moment, hem concebut una solució integral per abordar un problema establert (quines millores es poden realitzar a un allotjament per a augmentar la valoració a la pàgina web d'Airbnb i quin serà el guany resultant?). No obstant això, en aquest treball ens limitarem a abordar una part, que consta de tres objectius essencials: adquirir i estructurar una base de dades rellevant, realitzar un preprocessament meticulós de les dades per assegurar-ne la seva idoneïtat per a l'anàlisi i, finalment, desenvolupar un model predictiu de les valoracions.

Capítol 2

Planificació

En aquesta secció, es detalla l'organització durant el semestre per a l'execució del projecte. S'indiquen les diferents parts del projecte, juntament amb la planificació de temps destinada i la dedicació real en cadascuna. Les fases establertes inclouen: Documentació, Obtenció de Dades, Desenvolupament del Projecte, Redacció de la Memòria i Entrega Final del Treball. Cada fase ha estat organitzada amb una sèrie de tasques específiques que es descriuen a continuació.

La primera part del projecte, es centra en la documentació i l'obtenció de dades:

1. **Orientació:** Lectura de les Diapositives Informatives, revisió de les diapositives disponibles al campus virtual per obtenir una visió inicial del treball i orientar-ne l'enfocament i l'elaboració.

2. **Estudi sobre l'Aprenentatge Automàtic:** Realització d'investigació per adquirir una comprensió àmplia i crítica sobre Machine Learning. Aquesta inclou la lectura del llibre "Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow" de Aurélien Géron.

3. **Obtenció de Base de Dades:** Identificació i adquisició de la base de dades necessària per a la implementació del projecte.

Per a la segona fase del projecte s'han establert les següents tasques rellevants:

4. **Analitzar la Base de Dades i Fer Transformacions Prometedores:** Realització d'una anàlisi exhaustiu de la base de dades adquirida

durant la primera fase. Aquesta tasca inclourà identificar oportunitats de millora i aplicar transformacions significatives per optimitzar les dades per a l'etapa següent.

5. **Entrenar Models:** Desenvolupament i entrenament de models predictius basats en les dades preprocessades. Aquesta fase implica l'aplicació pràctica dels conceptes de Machine Learning adquirits durant la documentació inicial.

6. **Analitzar Resultats Obtinguts:** Examinar els resultats obtinguts dels models entrenats durant la fase anterior, destacant les tendències i les observacions rellevants, identificar patrons i tendències i formular conclusions.

7. **Redacció de la memòria:** La penúltima fase del Treball de Final de Grau (TFG) és la dedicada a la redacció de la memòria. Fins a aquest punt, el desenvolupament del TFG ha adoptat una dinàmica semblant a un "treball de camp". Aquesta etapa té com a objectiu plasmar de manera sistemàtica en el document escrit tot el procés i les conclusions obtingudes durant la realització del treball.

8. **Revisió:** Finalment, es reserva una última fase per a la realització d'una revisió general en col·laboració amb el tutor. Aquesta revisió implica un intercanvi d'opinions i recomanacions que permetran assegurar la coherència i la qualitat final del treball, així com afegir perspectiva i refinament abans de la seva entrega definitiva.

2.1 Diagrama de Gantt

Un cop establertes aquestes tasques principals, s'ha adoptat un diagrama de Gantt per a organitzar-se de manera eficient durant el semestre. Aquest diagrama detalla totes les tasques a realitzar, juntament amb la distribució temporal planificada per a cadascuna d'elles.

Els diagrames de Gantt representats a les Figures 2.1 i 2.2 il·lustren la planificació inicial i final, respectivament.

Durant la realització de les tasques planificades, es va observar que la lectura del llibre proposat i la realització dels exercicis associats demanava més temps del previst inicialment. Això es deu al fet que el redactor tenia coneixements previs molt bàsics, i aquesta etapa es va revelar com una inversió valuosa per establir una base sòlida per a la continuació del treball.

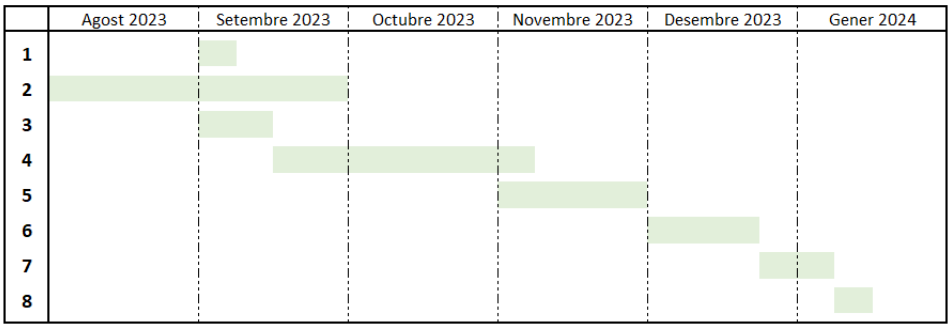


Figura 2.1: Diagrama de Gantt planificat inicial

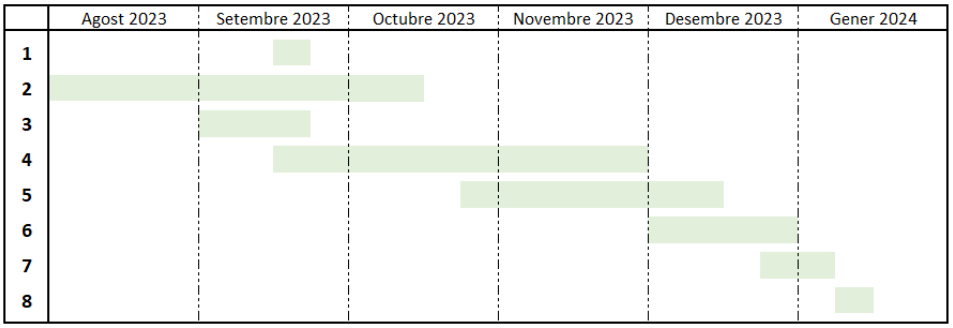


Figura 2.2: Diagrama de Gantt final

En relació amb l'obtenció de la base de dades, es va experimentar inicialment amb la idea de fer *web scraping* i crear la base de dades des de zero. Però a mesura que es va avançar en el procés, es va concloure que aquest enfocament era massa extens pel temps reduït que es tenia. Així doncs, es va identificar la pàgina web *Inside Airbnb* com una alternativa més eficient i es va procedir amb aquesta opció.

Cal de destacar l'anàlisi de la base de dades i a la implementació de transformacions significatives. Aquesta prioritat es va establir considerant la importància d'aquesta fase en la preparació de les dades per a l'entrenament dels models predictius, ja que es va reconèixer que aquesta etapa constitueix una part crítica per garantir resultats robustos i significatius en les etapes posteriors del projecte.

Capítol 3

Base de Dades

3.1 Extracció de Dades

Per a aquest estudi, la base de dades utilitzada és crucial per al desenvolupament i entrenament del model predictiu, així com per a l'anàlisi de les valoracions i la millora de l'experiència dels hostes. Degut a les restriccions de temps associades amb el desenvolupament del projecte, es va considerar la necessitat d'utilitzar una base de dades ja completa en lloc de crear una eina que utilitzi *web scraping* per l'obtenció de dades en temps real.

La base de dades seleccionada és la proporcionada per la pàgina web **Insideairbnb** [4], d'on s'ha extret una base de dades específica per a la ciutat de Barcelona actualizada per última vegada el 06/10/2023. Aquesta empresa fa tècniques de *web scrapping* periòdicament per a obtenir aquestes dades actualitzades. La base de dades consta de tres taules clau:

1. **Taula *"calendar"*:** Aquesta taula conté informació relativa al calendari futur de reserves i preus. Proporciona una visió temporal de l'ocupació dels allotjaments i els preus associats en un període determinat. No serà utilitzada en aquest estudi. A l'annex (3) s'hi pot trobar la taula de tots els atributs.

2. **Taula *"listings"*:** Aquesta taula conté tota la informació disponible sobre els allotjaments de Barcelona. Això inclou detalls com la descripció, la localització, la informació de l'amfitrió, les valoracions dels hostes i altres atributs rellevants. Aquesta taula serveix com a font integral d'informació per al desenvolupament del model predictiu. A l'annex (1) s'hi pot trobar la taula de tots els atributs.

3. **Taula *"reviews"*:** Aquesta taula conté tots els comentaris associats amb els allotjaments de Barcelona, així com la informació sobre qui va fer aquests comentaris. Aquesta font de dades permet una anàlisi detallada de les valoracions dels hostes i pot ser utilitzada per comprendre les preferències i les experiències dels usuaris. A l'annex (2) s'hi pot trobar la taula de tots els atributs.

D'aquestes taules, cal remarcar els atributs objectiu de la taula *listings*, que són els següents:

- *"review_scores_rating"*: valoració global de l'allotjament
- *"review_scores_accuracy"*: valoració de la precisió de les descripcions a la realitat de l'allotjament.
- *"review_scores_cleanliness"*: valoració de la netedat de l'allotjament.
- *"review_scores_checkin"*: valoració del check-in a l'allotjament.
- *"review_scores_communication"*: valoració de la comuniació amb el host de l'allotjament.
- *"review_scores_location"*: valoració de la localització de l'allotjament.
- *"review_scores_value"*: valoració del valor de l'allotjament.

L'ús d'aquesta base de dades específica de Barcelona proporciona un conjunt de dades detallat i extens que permetrà una anàlisi profunda de les dinàmiques d'allotjament a la ciutat i facilitarà el desenvolupament del model predictiu i les recomanacions d'ampliació.

Se suposa que cada ciutat tindrà unes característiques particulars que atrauran un cert tipus de clients. En aquest treball s'ha triat la ciutat de Barcleona. Tot i això, es podria ampliar l'estudi a qualsevol altre ciutat/-regió.

3.2 Anàlisi Descriptiu

L'anàlisi descriptiu de la base de dades és una fase crítica en qualsevol projecte d'investigació o anàlisi de dades. Aquesta etapa té com a objectiu principal explorar, resumir i presentar les característiques fonamentals del conjunt de dades. Mitjançant tècniques estadístiques i gràfiques, l'anàlisi

descriptiu proporciona una visió comprensiva de les distribucions, tendències i patrons que defineixen les variables d'estudi. A través d'aquesta etapa, s'estableixen les bases per a conclusions significatives i orientacions informades per al desenvolupament del projecte.

En aquesta base de dades, hi ha variables de tipus text, numèriques, categòriques, dates, llistats, etc. Cada tipus de variable haurà de ser tractada d'una manera específica, aquestes tècniques de processament de dades seran discutides a la secció de "Tractament de Dades" d'aquest treball. El que sí que farem en aquesta secció és destacar algunes variables que mereixen més atenció:

- *"amenities".* Aquesta variable conté un llistat amb tots els serveis que té l'allotjament. Cal remarcar que no és una variable categòrica simple, ja que hi ha molts serveis que es poden agrupar, com per exemple hi ha 167 categories que contenen la paraula wifi ('Fast wifi – 100 Mbps', 'Fast wifi – 102 Mbps', 'Fast wifi – 103 Mbps', 'Fast wifi – 105 Mbps'...).

- *"description".* Aquesta variable conté una descripció de l'allotjament, la qual conté informació rellevant per a l'allotjament, que pot o no estar en altres variables de la base de dades, com per exemple el públic objectiu, la superfície o la capacitat.

- *Variables relacionades amb la localitació.* hi ha diverses variables que tracten sobre la zona en la que està situat l'allotjament, però pot ser que no estigin ben delimitades. Per exemple, les variables de longitud i latitud presenten una granularitat excessiva per a l'entrenament del model, ja que la seva naturalesa precisa pot conduir a una sensibilitat excessiva a petites variacions en la ubicació geogràfica. Un altre exemple és la de *"neighbourhood_ group_cleansed"*, que agrupa els allotjaments en 10 zones de Barcelona: pot tenir mal delimitat els límits o està sent massa poc granular.

Per obtenir més informació sobre la base de dades, s'han dut a terme un seguit de proves:

Prova 1: Número de comentaris per Allotjament

En aquesta prova s'ha comprovat que el número de comentaris que apareix associat a cada allotjament de la taula *listings* SÍ concorda amb el número de comentaris que trobem a la taula *reviews*.

En la base de dades hi ha 17230 allotjaments amb un total de 729005 comentaris, però no totes les valoracions dels allotjaments són ajustades a la realitat. Per validar la valoració d'un allotjament, inicialment es posa un mínim de 100 comentaris per allotjament per tenir-lo en compte per a entrenar els nostres models (posteriorment, ja buscarem un número mínim amb més sentit). hi ha 2320 allotjaments amb un mínim de 100 comentaris (amb una mitjana de 214 comentaris/allotjament); en concret, hi ha 497897 comentaris asociats a aquests allotjaments, el que sembla una base de dades de tamany prou raonable per entrenar un model. A l'Annex, podem trobar un gràfic de la disminució d'allotjaments en funció del número mímin de comentaris (1).

Prova 2: Rellevància dels comentaris

En aquesta prova s'ha comprovat (de forma rudimentària) que tots els allotjaments tenen comentaris rellevants i que el 66% de tots els comentaris també ho són. S'ha definit un comentari rellevant aquell que conté alguna paraula relacionada als diferents atributs objectiu, les valoracions (aquest llistat conté paraules en castellà i en anglès, va ser creat per ChatGPT i es pot trobar als codis del Github). A l'Annex, es pot trobar un gràfic (2) amb la classificació dels comentaris segons la categoria en la que són rellevants. També s'ha estudiat la longitud dels comentaris i resulta que la mitjana de paraules per comentari és de 42 paraules.

Per tant, sembla que hi ha prous comentaris rellevants de longitud raonable.

Prova 3: Correlació entre atributs objectiu

En aquesta prova, s'han comprovat diverses coses. Primer s'ha observat que per allotjaments amb al menys 10 comentaris, no hi ha valors absents. Després s'ha fet una visualització per veure la distribució i correlació de les valoracions. D'aquí s'hi poden extreure les següents conclusions:

- Totes les valoracions tenen distribucions molt inclinades cap a la dreta, o sigui, que la majoria dels valors estan entre 4 i 5 (en la taula de l'Annex (9) hi ha el resum estadístic de les valoracions dels allotjaments), el que pot generar dificultats a l'hora de fer la regressió.

- Totes les valoracions estan molt correlacionades, en particular, a l'Annex hi ha una taula amb els valors de correlació de la valoració global

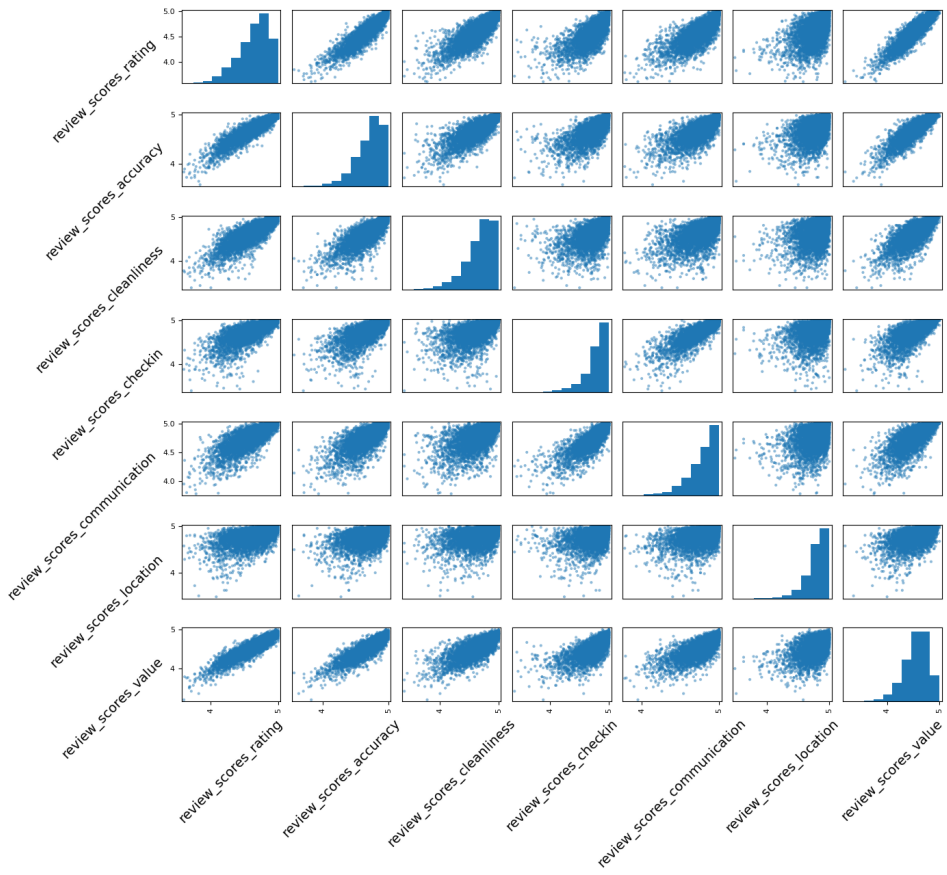


Figura 3.1: Aquesta matriu de dispersió representa les correlacions de cada tipus de valoració contra cada tipus de valoració. A la diagonal hi ha histogrames per entendre la distribució de cada atribut.

de l'allotjament amb els altres tipus de valoració (4) i s'hi pot observar que la valoració de localització és la menys correlacionada i per tant segurament costi més assolir una precisió alta a l'hora de predir la localització.

- Com la localització és la valoració menys correlacionada amb la valoració global, el propietari pot fer moltes coses per millorar l'experiència dels clients.

- El propietari pot millorar la valoració global del seu allotjament dient la veritat de l'allotjament, proporcionant una comunicació fluida i donant un bon servei de check-in. Totes amb cost zero i un alt retorn en satisfacció del client.

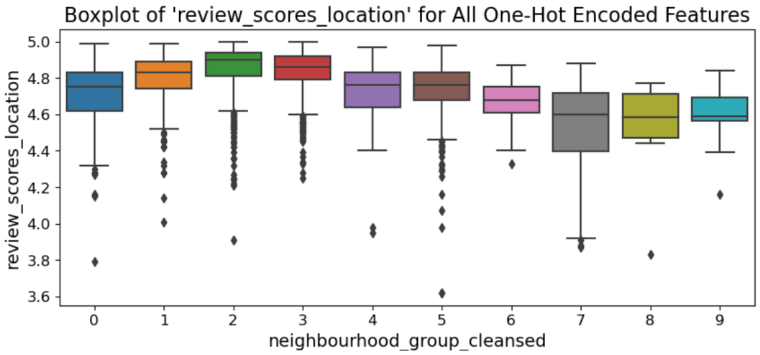


Figura 3.2: BoxPlot de la valoració de localització per barri.

També s'han estudiat els atributs més correlacionats amb les valoracions, els podem trobar a l'Annex (les taules 4, 5, 6, 7, 8). D'elles, s'hi pot extreure algunes conclusions interessants:

- Una millora fàcil i que portarà a un efecte molt positiu a la valoració de l'allotjament, és proporcionar coixins i llençols extres, sabó, un kit de primers auxilis, cortines fosques, llibres i que el host li doni la venvinguda al client. Això és degut a que està correlacionat positiva i significativament amb la majoria de les valoracions.

- Es pot veure una clara correlació negativa entre la quantitat d'allotjaments i la impersonalitat del servei amb la valoració.

Prova 4: Valoració de localització

En aquesta prova, es vol mostrar la relació entre les diferents variables que es tenen sobre la localització i la valoració de localització de l'allotjament. També es busca proposar transformacions i creació de variables prometedores per treballar al llarg del projecte.

El primer gràfic 3.2 (la seva taula d'atributs 3.1), és el BoxPlot de les valoracions de localització dels allotjaments de barcelona classificat per les 10 categories de la variable *"neighbourhood_group_clensed"*. En ell, podem observar com les mitjanes de la valoració de localització de les zones són molt altes. Tot i així, totes les zones exceptuant Les Corts, tenen bastants valors atípics. Per tant, segurament hi hagi formes millors de classificar les observacions o crear variables amb més bon rendiment.

A continuació es mostra una visualització de les valoracions dels apartaments (veure Figura 3.3). La posició dels punts s'extreu de les coordenades reals de l'allotjament, el color indica la valoració de localització,

Attributes	Values
0	Eixample
1	Sant Martí
2	Gràcia
3	Ciutat Vella
4	Sarrià-Sant Gervasi
5	Sants-Montjuïc
6	Les Corts
7	Horta-Guinardó
8	Sant Andreu
9	Nou Barris

Taula 3.1: Atributs i els seus valors corresponents.

i el tamany del cercle indica el preu. En aquest mapa, s'han afegit les localitzacions amb els seus noms per millorar la interpretabilitat del mapa. Aquests punts s'han seleccionat de manera que descriuen millor les valoracions de localització.

D'aquesta representació, es dedueix que pot ser interessant crear una nova variable per a cada lloc marcat al mapa, la qual contingui la distància euclidiana entre l'allotjament i el lloc corresponent. Aquesta nova variable pot proporcionar més informació sobre la relació espacial entre l'allotjament i els diversos punts d'interès, contribuint a una millor comprensió de les dades.

3.2.1 Gestió dels Valors que Falten

L'exploració inicial de la base de dades revela la presència de valors que falten en diversos atributs (veure taula 3.2). A continuació, es presenta una anàlisi dels valors que falten i les estratègies per tractar-los.

Valors que Falten a les Files

S'observa que aproximadament el 18.85% dels allotjaments tenen almenys 5 valors que falten a la seva informació. Això descarta la opció de simplement eliminar les observacions amb un nombre significatiu de valors que falten, i es planteja una solució aproximada per a cada atribut.

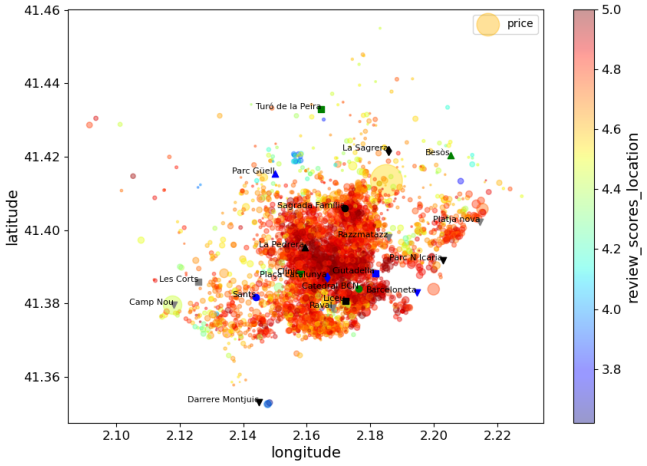


Figura 3.3: Mapa de tots els allotjaments de Barcelona a la plataforma. El color indica la valoració de la localització, el tamany del cercle indica el preu i estan col·locats a escala utilitzant longitud i latitud real.

Valors que Falten als Atributs Objectiu (Valoracions)

Una anàlisi de valors "missing" revela que no hi ha valors que falten per als allotjaments que tenen més de 10 ressenyes. Això proporciona una base sòlida per a la fiabilitat de les valoracions de les ressenyes en aquests casos.

Altres Valors que Falten

Els valors que falten varien segons les diferents característiques:

- *"bathrooms"* i *"calendar_updated"* estan buides en el 100% de les observacions i, per tant, seran eliminades.
- Les variables que tenen relació amb la localització, "neighborhood_overview" i "neighbourhood", són de tipus categòric. Per tant, un cop es transformin en variables numèriques, els valors *Missing* seran transformats a la mediana.
- Les variables *"host_neighbourhood"* i *"host_about"* s'eliminen ja que són irrellevants per a l'objectiu del projecte.
- La variable *"license"* serà transformada a una variable booleana indicant si falta o no.

Característica	Percentatge de Valors que Falten
"bathrooms"	100.0%
"calendar_updated"	100.0%
"neighborhood_overview"	25.44%
"neighbourhood"	25.44%
"host_neighbourhood"	23.13%
"bedrooms"	22.07%
"license"	19.98%
"host_about"	19.95%
"host_is_superhost"	19.78%
"host_location"	11.83%
"host_response_time"	8.03%
"host_response_rate"	8.03%
"host_acceptance_rate"	6.24%
"beds"	0.63%
"bathrooms_text"	0.1%
"description"	0.03%

Taula 3.2: Percentatge de valors que falten per característica

- La variable *"host_is_superhost"*, en cas de faltar, es substituirà per 0 donant a entendre que no és "super host", ja que és una marca de valor segur per part del propietari de l'allotjament.

- Per a aquelles característiques amb menys d'un 15% de valors que falten i *"bedrooms"*, es procedirà a la substitució mitjançant la mediana (un cop siguin variables de tipus numèric).

En resum, la gestió dels valors que falten implica abordar cada situació de manera específica, adaptant diferents mètodes de tractament segons la naturalesa i el context de les dades mancants.

3.2.2 Gestió d'Observavions Anòmales

L'efectiva gestió d'observacions anòmales és crucial en l'anàlisi de dades, ja que aquests valors atípics poden tenir un impacte significatiu en la interpretació i precisió dels resultats. Aquesta secció es centra en les estratègies i mètodes utilitzats per identificar, avaluar i, si escau, tractar els valors atípics presents a la nostra base de dades. Abordar aquesta qüestió és essencial per garantir que les conclusions i els models derivats de l'anàlisi reflecteixin amb precisió la naturalesa de les dades i proporcionin resultats robustos i confiables.

En aquesta primera anàlisi de la base de dades no s'han observat valors anòmals notables.

3.2.3 Eliminar Atributs

L'últim pas és eliminar totes les variables que no ens serviran, ja sigui per impossibilitat de fer una transformació útil, per poca consistència o per no ser rellevants. Les variables eliminades en primera instància són: *"id"*, *"listing_url"*, *"scrape_id"*, *"last_scraped"*, *"source"*, *"name"*, *"neighborhood _overview"*, *"picture_url"*, *"host_id"*, *"host_url"*, *"host_ name"*, *"host _since"*, *"host_location"*, *"host_about"*, *"host_thumbnail_url"*, *"host _picture _ url"*, *"host _ neighbourhood"*, *"host_verifications"*, *"name"*, *"has_availability"*, *"calendar_last _ scraped"*.

Capítol 4

Metodologia

En aquest capítol, s'estudiaran les diferents tècniques que s'aplicaran per extreure prediccions de la base de dades origen. Primer s'exploraran les diferents tècniques de tractament de dades, aplicat a la tipologia de dades trobades a la base de dades que ja tenim i les automatitzacions dels tractaments de dades aplicats per a facilitar l'estudi. Després investigarem els algorismes d'aprenentatge automàtic utilitzats en aquest treball.

4.1 Tractament de Dades

En aquesta secció estudiarem els tractaments de dades específics per a cada tipus de dades que hi ha a la base de dades. També estudiarem la posible creació de nous atributs per a millorar la capacitat predictiva dels nostres models i finalment veurem les automatitzacions que es faran per a estandaritzar i optimitzar el processament de les dades.

4.1.1 Categòriques i Text

One Hot Encoding

One Hot Encoding és una tècnica per tractar les variables tipus text categòriques. La majoria d'algoritmes d'aprenentatge automàtic prefereixen treballar amb nombres i aquí és on entra aquesta tècnica, converteix les categories en números. Podem generar nous atributs, un per cada categoria o crear un atribut que mapegi cada classe amb un número enter. L'opció de crear un atribut que mapegi cada classe amb un número enter pot tenir sentit per dues raons: la primera, perquè ocupa molt menys espai, ja que és genera una matriu densa; i la segona, és útil en els casos que les categories tinguin moltes similituds. Nosaltres, en canvi, crearem un nou atribut

binari per cada categoria perquè, tot i ser una matriu dispersa i ocupar més espai, ens serà més útil, ja que es creu que donarà millor rendiment d'aprenentatge i millorarà la interpretabilitat del model.

En aquesta base de dades, hi ha diversos atributs tipus text que són categòrics: *"property_type"*, *"room_type"*, *"amenities"*, *"neighbourghood"*, *"neighbourhood _vleansed"* i *"neighbourhood_group_clensed"*. Per a aplicar aquest mètode, es podria fer manualment, però s'utilitzarà la llibreria *sklearn* per la integració amb els *Transformers* i *Pipelines* de *sklearn*, que més endevant en farem ús. En el codi (4.1) hi ha un exemple sobre com transformar una variable categòrica en diversos atributs de tipus binari.

Però a vegades hi ha massa categories i s'ha de tractar diferent: per exemple, l'atribut *"amenities"*, conté 1044 categories diferents i descriu massa categories amb massa granularitat. Per a solucionar aquest problema es pot aplicar One Hot Encoding directament i després aplicar algun mètode de reducció dimensional com PCA. La tècnica que s'ha decidit utilitzar en aquest cas: primer agrupar paraules similars, després aplicar One Hot Encoding i després aplicar PCA per reduir més la dimensió i entrenar amb més facilitat el model. Aquestes agrupacions de categories s'han fet amb criteri expert. Per exemple, es va observar que hi havia 167 categories que contenien la paraula "wifi" d'alguna forma o altre, totes aquestes categories s'han comprimit a "wifi". Fent aquesta agrupació, hem perdut granularitat i potencialment precisió per al model, però hem guanyat interpretabilitat del model i velocitat d'entrenament i predicció.

Codi 4.1: Codi Python per aplicar One Hot Encoding a la variable neighbourhood_group_cleansed.

```
from sklearn.preprocessing import OneHotEncoder

neighbourhood_encoder = OneHotEncoder()
nei_groups_1hot =
   neighbourhood_encoder.fit_transform(A[
   ['neighbourhood_group_cleansed']])
```

Embedding

Text embedding fa referència a la representació numèrica d'un text en un espai vectorial continu. És un mètode utilitzat en Processament del Llenguatge Natural (PNL) per transformar paraules o frases en vectors de números.

Els models de *text embedding* aprenen a assignar vectors numèrics a les paraules o frases de manera que paraules similars estiguin a prop de vectors similars en aquest espai vectorial. Aquesta representació vectorial captura les relacions semàntiques entre les paraules.

hi ha diversos mètodes per aconseguir *text embedding*, com ara *Word Embeddings* (incloent Word2Vec, GloVe) o *Embeddings* basats en *transformers* (com BERT, GPT), entre d'altres. Aquestes tècniques són àmpliament utilitzades en tasques de processament del llenguatge natural, com ara classificació de text, traducció automàtica i recuperació d'informació textual.

En aquest treball, hem utilitzat un model preentrenat d'embeddings basat en *transformers* que es diu *bert-base-uncased*. Aquest processament es pot aplicar a qualsevol variable tipus text, per exemple els atributs *"description"*, *"reviews"*, *"neighbourhood_overview"*. Concretament, la variable *"description"* conté molta informació, des del tamany de l'allotjament a especificacions que no estan escrites en cap altre variable.

El codi per aplicar el model *bert-base-uncased* és el següent:

Codi 4.2: Codi Python per aplicar Embedding a la variable description.

```
from transformers import BertTokenizer, BertModel
import torch

# Carregar el model pre-entrenat BERT i el
   tokenitzador (es necessita tenir instalat
   transformers)
model_name = "bert-base-uncased"
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertModel.from_pretrained(model_name)

text_column = filtered_insight['description']
embeddings = []

for text in text_column:
    # Tokenitzar i convertir les frases
    tokens = tokenizer(text, padding=True,
       truncation=True, return_tensors="pt")

    # Calcular els embeddings
    with torch.no_grad():
```

```
        outputs = model(**tokens)
        sentence_embedding =
            outputs.last_hidden_state.mean(
            dim=1).squeeze().numpy()

    embeddings.append(sentence_embedding)

embeddings_array = np.array(embeddings)
```

PCA

PCA (*Principal Component Analysis*) és un algoritme de reducció dimensional. Primer identifica l'hiperplà que hi ha més a prop de les dades i, després, fa una projecció de les dades a ell. D'aquesta manera es projecten les dades a una dimensió inferior i per tant passem a tenir una base de dades amb menys atributs.

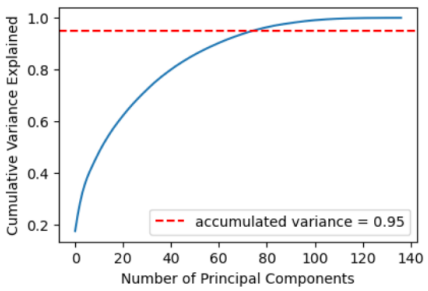
El problema aquí és agafar el nombre adequat de dimensions, el que s'ha fet en aquest treball és quedar-nos amb la mínima dimensió possible per conservar el 0.95 de la variança total de les dades. Això es fa amb la llibreria *sklearn*, en el Codi (4.3) podeu trobar un exemple de com s'han aplicat als atributs creats amb l'algoritme de OHE sobre l'atribut "*amenities*".

S'ha aplicat aquesta tècnica al resultat d'aplicar OHE a *"amenities"* i als Embeddings de *"description"*. Els resultats han estat: en el cas d'*"amenities"*, s'han reduit de 137 atributs a 76; i en el cas de la *"description"*, s'han reduit de 786 a 133 noves variables. En els gràfics (4.1a i 4.1b), s'hi pot observar la suma acumulada de variable en funció del número de components, o sigui, de la dimensió de l'hiperplà sobre el que projectarem les dades; la línea horitzontal vermella, indica el tall de 0.95 de variància acumulada.

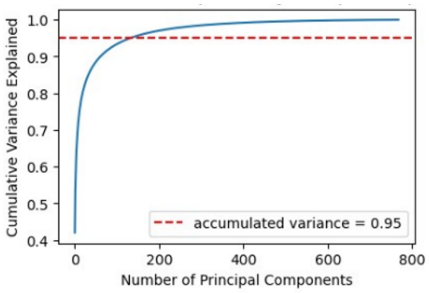
Codi 4.3: Codi Python per aplicar One Hot Encoding a la variable neighbourhood_group_cleansed.

```
from sklearn.decomposition import PCA

variance = 0.95
amenities_pca = A[filtered_amenities_list].copy()
pca = PCA(n_components = variance)
compressed = pca.fit_transform(amenities_pca)
```



(a) Aquest gràfic conté la suma acumulada de variança en funció del tamany de l'hiperplà per aplicar PCA a la variable amenities després d'haber aplicat OHE.



(b) Aquest gràfic conté la suma acumulada de variança en funció del tamany de l'hiperplà per aplicar PCA a la variable descripció després d'haver aplicat Embedding.

Altres

Hi ha variables tipus text que es poden transformar a variables numèriques amb una mica de tractament personalitzat. En aquesta base de dades variables binàries que, en lloc de ser representades amb 0 i 1, utilitzen "f" i "t", o que números amb el símbol de les unitats al final. Anem a tractar una a una les variables d'aquest tipus:

1. Transformar les variables *"host_is_superhost"*, *"host_has_profile _pic"*, *"host_identity_verified"* de "f" i "t" a 0 i 1.

2. Transformar la variable *"price"* a valor de punt flotant, això ho farem eliminant el símbol $.

3. Transformar totes les variables que són dates *"first_review"*, *"last_review"* a marca de temps.

4. Transformar les variables *"host_acceptance_rate" i "host_response_rate"* treure el símbol % i convertir-la en float.

4.1.2 Numèriques

En aquesta secció, abordarem el tractament de les variables numèriques per optimitzar la seva contribució als models d'aprenentatge automàtic. Aquest procés inclou diverses tècniques que s'ajusten a les diferents característiques i distribucions de les dades. Les següents estratègies són considerades per millorar la consistència i el rendiment dels models:

1. **Estandardització (Standardization):** Utilitzar `StandardScaler` per estandarditzar les variables numèriques assegurant que tinguin una

mitjana zero i una desviació estàndard d'1. Això és especialment útil quan s'usen algoritmes que depenen de distàncies o gradients, com ara SVMs o mètodes basats en gradient.

2. **Normalització (Normalization):** Si les variables numèriques tenen una distribució no gaussiana, es pot considerar la normalització, que escala les dades a un rang específic, com ara (0, 1) o (-1, 1). Això és útil en algunes situacions, com ara amb algoritmes basats en arbres o xarxes neuronals (es pot utilitzar el MinMaxScaler de sklearn).

3. **Transformació Logarítmica:** Si les variables tenen una distribució molt esbiaixada (*skewed*), una transformació logarítmica pot ajudar a normalitzar-les. Això és especialment útil en dades com ingressos o preus.

4. **Binning (Segmentació):** Convertir variables numèriques en categories discretes mitjançant el binning o segmentació. Això pot ser útil en algunes situacions, com per exemple quan es treballa amb variables d'edat o ingressos.

5. **Scaling Robust (Escala Robusta):** Si les dades contenen *outliers*, es podria considerar l'ús de l'escalat robust utilitzant, per exemple, `RobustScaler`. Aquest mètode és menys sensible als *outliers*.

En aquesta base de dades, es tractaran les dades numèriques aplicant totes les tècniques esmentades i al final s'usarà la que millor funcioni.

4.1.3 Creació de nous atributs

L'apartat de creació de nous atributs, permet millorar la representació de les dades i, en conseqüència, potenciar el rendiment dels models d'aprenentatge automàtic. En aquesta secció, explorarem la importància i els beneficis de la creació de nous atributs, així com algunes de les estratègies i tècniques que es poden utilitzar per desenvolupar variables addicionals que capturin millor la informació rellevant de les dades originals. Mitjançant la innovació i la derivació de noves característiques, buscarem enriquir la complexitat i la qualitat del conjunt de dades, proporcionant així als models un conjunt d'informació més ric i útil per a la tasca específica que es pretén abordar. Aquí hi ha el llistat de noves variables (sense tenir en compte les creades per OneHotEncoding) a estudiar:

1. **Distància a llocs remarcats:** Com s'ha vist a la Prova 4 de l'apartat d'Anàlisi Descriptiu, es crearà diveres variables amb la distància Euclídea de cada allotjament a cada lloc remarcat.

2. **Sentiment dels comentaris:** Utilitzar algun model de llenguatge preentrenat per a extreure una mesura de polaritat de cada comentari. Per a fer-ho, s'utilitzarà la classe *SentimentIntensityAnalyzer* de la llibreria *vaderSentiment.vaderSentiment*.

3. **Extreure variables de la descripció:** Crear noves varaibles de la informació que hi ha a la variable *"description"*. Es crearan les següents variables: públic objectiu, la superfície o la capacitat. Es preten utilitzar Chat GPT, fent us de *Prompt Injection* per analitzar les descripcions i extreure aquestes noves variables. Per a fer-ho, s'utilitzarà la llibreria gratuita de Python *g4f*.

4.1.4 Automatitzacions

Per a automatitzar el procés de tractament de dades, s'ha fet una llibreria amb tres classes de Python. Aquestes tenen la funció de facilitar, estandaritzar i eficientar el procés de tractament de dades, entrenament i testeig de models d'aprenentatge automàtic:

- "AmenitiesTransformer": És un *Custom Transformer* (després estudiarem aquest concepte) per a poder fer el tractament de *One Hot Encoding* i combinació de categories dintre d'un flux de treball més eficient.

- "DataPreProcess": És la classe que conté totes les funcions de processament de dades (a part del tractament a la classe "amenities". Entre altres, té de guardar i carregar bases de dades ja tractades, aplicar embeddings, descriure bases de dades i aplicar el tractament a la base de dades.

- "ModelMaking": És la classe que conté totes les funcions relacionades amb l'entrenament i el testeig de models. Conté funcions per guardar i carregar models (i les seves bases de dades), visualitzar els resultats, estudiar la importància dels atributs als models i entrenar models.

L'us de *Estimators*, *Predictors*, *Transformers* i *Pipelines*. L'ús d'aquests conceptes ajuda la optimització dels models, millor interpretació, eficiència del codi, col·laboració efectiva. Ara veurem què són, per a què serveixen i com els podem aplicar.

Estimators, Predictors, Transformers i Pipelines

1. **Estimators:**

   - **Definició**: En la biblioteca Scikit-learn i altres frameworks, un *estimator* és un objecte que modela dades mitjançant l'aprenentatge supervisat o no supervisat.
   - **Característiques:** Un *estimator* té la capacitat de ser entrenat sobre dades i, un cop entrenat, pot fer prediccions sobre noves dades.
   - **Exemple:** Els models com ara regressors lineals o arbres de decisió són exemples d'estimators. S'inicialitzen, s'entrenen amb dades d'entrada i, finalment, es poden utilitzar per fer prediccions.

2. **Predictors:**

   - **Definició:** Un *predictor* és una instància ja entrenada d'un *estimator* que és capaç de fer prediccions sobre noves dades.
   - **Funció:** Després d'entrenar un *estimator*, es converteix en un *predictor* que pot prendre noves dades d'entrada i generar prediccions o classificacions.
   - **Exemple:** Si tenim un model de regressió logística, un *predictor* seria una instància d'aquest model ja ajustat amb dades i preparat per fer prediccions.

3. **Transformers:**

   - **Definició:** Un *transformer* és un component que processa les dades d'entrada i les transforma d'acord amb certes regles o operacions.
   - **Funció:** Es pot utilitzar per netejar, normalitzar o crear noves característiques a partir de les dades originals.
   - **Exemple:** Un *transformer* podria ser una funció que converteix variables categòriques en codificació one-hot o que estandarditza les dades.

4. **Pipelines:**

   - **Definició:** Un *pipeline* és una seqüència ordenada de *transformers* i *estimators* que s'aplica seqüencialment al conjunt de dades.

- **Funció:** Simplifica el flux de treball de Machine Learning, permetent la creació d'un procés ordenat que abasta des de la càrrega de dades fins a la predicció.
- **Exemple:** Un *pipeline* podria consistir en etapes com la càrrega de dades, la normalització i l'entrenament del model, tot organitzat de manera estructurada.

Codi 4.4: Codi Python que mostra una implementació de Pipelines i Transformers per fer processament de dades.

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler,
   OneHotEncoder
from sklearn.impute import SimpleImputer

num_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy="median")),
    ('std_scaler',StandardScaler())
])

def identity(df):
    return df
iden = FunctionTransformer(identity)

pipe = ColumnTransformer([
    ('1HOT',OneHotEncoder(), hot_features),
    ('numbers',num_pipeline, num_features),
    ('identity',iden, other_features)
],
    sparse_threshold=0 )

pipe.fit(A)
```

En aquest fragment de codi, es realitza el pre-processament de les dades, abordant diferentment les variables categòriques, numèriques i altres. Per a les dades categòriques, s'aplica la tècnica One-Hot Encoding per transformar-les en representacions numèriques. Pel que fa a les dades numèriques, els valors buits es completen utilitzant la mediana de cada columna. Finalment, els altres atributs es processen de manera idèntica, sense cap transformació especial. Aquesta etapa de pre-processament és

crucial per garantir que les dades siguin aptes per als algoritmes d'aprenentatge automàtic, assegurant alhora que les característiques es presentin de la manera més informativa possible.

En aquest exemple només es fa processament de dades, però també s'hi pot afegir l'entrenament dintre dels *Pipelies* i això és especialemnt interessant per l'etapa de millora dels hiperparàmetres del model, com per exemple la tècnica de *Grid Search* (tot i que aquest pas està fora de l'enfoc de l'estudi).

4.2 Entrenament

Un model d'aprenentatge automàtic és un algorisme o conjunt d'algorismes que s'entrenen utilitzant dades per fer prediccions o prendre decisions sense ser programats explícitament per a aquestes tasques. Aquest tipus de models formen part del camp de l'aprenentatge automàtic, que és una branca de la intel·ligència artificial.

En l'aprenentatge automàtic, els models aprenen a reconèixer patrons o a fer generalitzacions a partir de les dades d'entrada sense ser programats amb instruccions específiques per a cada tasca. L'entrenament d'un model implica exposar-lo a un conjunt de dades d'entrada, que ja conté les sortides desitjades o etiquetes. El model ajusta els seus paràmetres internament per minimitzar l'error entre les seves prediccions i les sortides reals.

Hi ha diversos tipus de models d'aprenentatge automàtic, com ara els de regressió per a la predicció de valors numèrics, de classificació per a la categorització d'observacions en classes, i altres models més avançats com els models de xarxes neuronals.

En el context d'aquest treball, els models d'aprenentatge automàtic seran les eines principals utilitzades per predir les valoracions dels allotjaments basant-se en les característiques proporcionades.

Fins a aquest punt s'han explorat dades i tècniques de preprocessament de dades, aquestes dades s'utilitzaran per entrenar els nostres models d'aprenentatge automàtic. Per a fer-ho, es divideix la base de dades en dos conjunts: un conjunt d'entrenament i un conjunt de prova. Aquesta pràctica ens permet utilitzar dades conegudes per ensenyar el model i avaluem la seva habilitat per fer prediccions sobres dades noves que no ha vist en l'entrenament. Per a fer-ho, utilitzarem la funció train_test_split() de sklearn per dividir la base de dades original en 80% per entrenament i

20% per testeig i és el que s'utilitzarà en aquest treball.

Una excel·lent alternativa és utilitzar la funcionalitat de validació creuada amb K-fold de Scikit-Learn. que divideix aleatoriament el conjut d'entrenament en n subconjunts distintius anomenats folds. Després entrena i avalua el model n vegades, selseccionant un fold diferent per a l'avaluació cada vegada i entrenant amb els altres n-1 folds. El resultat és una matriu que conté les n valoracions d'avaluació. Aquest mètode és molt més fort, però no l'utilitzarem, ja que no ens interessa tant millorar el model, en aquest treball vusquem depurar la base de dades.

En aquest treball, només entrenarem models de Regressió Lineal i *Random Forest*. S'espera que els models entrenats amb l'algoritme de *Random Forest* siguin significativament millors, ja que és molt potent. Ara veurem com funcionen aquests algoritmes.

4.2.1 Regressió Lineal

La regressió lineal és un tipus de model d'aprenentatge automàtic utilitzat per predir valors numèrics basant-se en una relació lineal entre les variables d'entrada i la variable objectiu. En la seva forma més simple, per una sola variable d'entrada (x) i una variable de sortida (y), la regressió lineal es pot expressar com:

$$y = mx + b \tag{4.1}$$

On:

- y és la variable de sortida (o resposta)
- x és la variable d'entrada
- m és la pendent de la recta (pendent de regressió)
- b és la intersecció amb l'eix y (interceptació)

En el cas de més d'una variable d'entrada, la formulació general es converteix en:

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n \tag{4.2}$$

On:

- b_0 és la intersecció amb l'eix y
- $b_1, b_2, \ldots, b_n$ són els coeficients associats a les variables d'entrada $x_1, x_2, \ldots, x_n$.

En el context del nostre treball, la regressió lineal es pot utilitzar per predir les valoracions dels allotjaments basant-se en les característiques proporcionades.

A l'aprenentatge automàtic, els vectors sovint es representen com vectors columna, que són matrius 2D amb una sola columna. Si θ i x són vectors columna, llavors la predicció és $y = \theta^T x$, on θ^T és la transposada de θ (un vector fila en lloc d'un vector columna) i $\theta^T x$ és la multiplicació de matrius de θ^T per x. És, evidentment, la mateixa predicció, però ara es representa com una matriu d'una sola cel·la en lloc d'un valor escalar. Utilitzarem aquesta notació a partir d'ara.

Però com s'entrena? Entrenar un model significa ajustar els seus paràmetres perquè el model s'ajusti millor al conjunt d'entrenament. Per a això, necessitem primer una mesura de com de bé (o malament) el model s'ajusta a les dades d'entrenament. La mesura de rendiment més comuna d'un model de regressió és l'Error Quadràtic Mitjà (RMSE) 4.3.

$$RMSE(X, h_\theta) = \sqrt{MSE(X, h_\theta)} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\theta^T x_i - y_i)^2} \tag{4.3}$$

On:

- m és el nombre d'observacions d'entrenament
- θ és el vector de paràmetres del model
- x_i és el vector d'atributs de la mostra i
- y_i és la sortida real de la mostra i

Per tant, per entrenar un model de regressió lineal, s'ha de trobar el valor de θ que minimitza l'RMSE. En la pràctica, és més senzill minimitzar l'error mitjà quadràtic (MSE) que l'RMSE, i porta al mateix resultat (ja que el valor que minimitza una funció també minimitza la seva arrel quadrada). La fórmula MSE per l'entrenament de Regressió Lineal és la següent:

$$MSE(X, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta^T x_i - y_i)^2 \tag{4.4}$$

4.2.2 Arbres de Decisió

Els arbres de decisió són models predictius que utilitzen una estructura jeràrquica d'"if-else" per prendre decisions. Cada node representa

una pregunta o predicció, mentre que les branques són les diferents respostes possibles. Són algoritmes versàtils d'aprenentatge automàtic que poden realitzar tasques de classificació i regressió, i fins i tot tasques de multi-sortida. Són algoritmes potents capaços d'ajustar conjunts de dades complexes. Aquí es detallen els aspectes clau relacionats amb els arbres de decisió:

Entrenament d'un Arbre de Decisió

Per entrenar un arbre de decisió, es busca dividir el conjunt de dades de manera que les subdivisions siguin cada vegada més pures. La puresa es mesura utilitzant la Gini impurity o l'entropia. El criteri Gini per a un node t amb N_t mostres és:

$$Gini(t) = 1 - \sum_{i=1}^{J} p(i|t)^2 \tag{4.5}$$

On $p(i|t)$ és la proporció de mostres de la classe i al node t. L'entropia es defineix com:

$$H(t) = -\sum_{i=1}^{J} p(i|t) \log_2(p(i|t)) \tag{4.6}$$

El procés d'entrenament implica seleccionar els atributs i els punts de tall que maximitzin la reducció de la Gini impurity o l'entropia.

Balanceig i Regularització

Els arbres de decisió poden tenir una tendència a sobreajustar-se al conjunt d'entrenament. Per mitigar això, es poden utilitzar tècniques com la tala (pruning) de l'arbre i la limitació de la seva profunditat. Aquesta regularització controla la complexitat de l'arbre i evita models massa específics per al conjunt d'entrenament. Un altre forma de mitigar-ho és utilitzant tècniques de reducció de components com és PCA, que resulta en una millor orientació de les dades d'aprenentatge.

Arbres de Decisió per a Regressió

Mentre que els arbres de decisió són sovint utilitzats per problemes de classificació, també es poden aplicar a tasques de regressió. Per fer-ho, el criteri de divisió es canvia per minimitzar la suma dels quadrats dels errors (SSE) a cada node.

Suposem que tenim un node t amb N_t mostres, i els valors de la variable objectiu per a cada mostra estan donats per y_i (on $1 \leq i \leq N_t$). La suma dels quadrats dels errors (SSE) per al node t es defineix com:

$$SSE(t) = \sum_{i=1}^{N_t} (y_i - \bar{y}_t)^2 \tag{4.7}$$

A l'hora de fer divisions en el procés d'entrenament, es busca minimitzar la reducció en el SSE. La fórmula per calcular la reducció en SSE (R_{SSE}) quan es divideix un node en dos nodes fills (t_1 i t_2) és:

$$R_{SSE} = SSE(t) - (SSE(t_1) + SSE(t_2)) \tag{4.8}$$

La divisió es realitza seleccionant l'atribut i el punt de tall que maximitzin R_{SSE}. Això implica buscar la combinació d'atribut i punt de tall que minimitzi la suma dels SSE dels nodes resultants després de la divisió.

Aquesta metodologia de SSE permet als arbres de decisió adaptar-se a problemes de regressió, ja que la divisió es realitza amb l'objectiu de reduir la variància dels valors de la variable objectiu dins cada node de l'arbre.

Els arbres de decisió tenen tendència a sobreajustar-se quan es tracta de tasques de regressió. Més en general, el principal problema amb els arbres de decisió és que són molt sensibles a petites variacions en les dades d'entrenament. De fet, com que l'algorisme d'entrenament utilitzat per Scikit-Learn és estocàstic, és possible obtenir models molt diferents fins i tot amb les mateixes dades d'entrenament (a menys que es configuri el paràmetre random_state).

Els *Random Forest* poden limitar aquesta inestabilitat mitjançant la mitjana de les prediccions de molts arbres, com veurem en la següent apartat.

4.2.3 Aprenentatge Conjunt i Random Forest

Classificador de Votació

Un classificador de votació, o *Voting Classifier*, és un tipus de model d'aprenentatge supervisat que combina diverses prediccions realitzades per diferents models per millorar el rendiment global. Per exemple si entrenes 1000 classificadors que individualment són correctes el 51% de les vegades, es pot esperar una precisió del 75% en el mètode conjunt de votació.

Aquesta tècnica, funcionen millor com més independents siguin els predictors entre ells. Una forma d'obtenir classificadors diversos, és entrenar diferents algoritmes. Això millora les possibilitats de que fassin errors diferents, millorant la precisió global.

En sklearn, la biblioteca de Python per a l'aprenentatge automàtic, es pot utilitzar la classe `VotingClassifier` per construir aquest tipus de models. Aquesta classe permet combinar múltiples estimadors amb diferents algoritmes o paràmetres.

Les tècniques de *bagging* (abreviatura de *Bootstrap Aggregating*) i *pasting* són mètodes de votació d'aprenentatge automàtic.

Bagging

Amb el *Bagging* (abreviatura de *Bootstrap Aggregating*), es crea un conjunt de models utilitzant mètodes d'aprenentatge semblants, com ara arbres de decisió, però amb petites diferències en les dades d'entrenament. Aquestes petites diferències es generen mitjançant mostreig amb reemplaçament (*bootstrap*). Cadascun dels models resultants és entrenat amb una mostra diferent, i les seves prediccions es combinen mitjançant un vot majoritari per a tasques de classificació o una mitjana per a tasques de regressió.

Exemple de *Bagging* amb sklearn:

Codi 4.5: Codi Python que mostra l'entrenament d'un model utilitzant l'algoritme de Bagging amb la llibreria sklearn.

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier

# Crear un classificador d'arbres de decisio
base_classifier = DecisionTreeClassifier()

# Crear el BaggingClassifier
bagging_clf = BaggingClassifier(base_classifier,
    n_estimators=10, random_state=42)

# Entrenar el model
bagging_clf.fit(X_train, y_train)
```

```
12
13 # Realitzar prediccions
14 y_pred = bagging_clf.predict(X_test)
```

Pasting

El *pasting* és una variant del bagging en què les mostres es prenen sense reemplaçament. Això significa que cada mostra només es pot seleccionar una vegada per construir cada model del conjunt. Aquesta tècnica es pot utilitzar quan el conjunt de dades és prou gran i permet crear models amb diverses perspectives sense la necessitat de repetir mostres.

L'ús de *pasting* en sklearn seria similar al de *bagging*, simplement especificant `bootstrap=False` a la funció BaggingClassifier.

Random Forest

Un Bosc Aleatori és un conjunt d'Arbres de Decisió, generalment entrenat mitjançant el mètode de la Bagging (o Pating), típicament amb max_samples establert a la mida del conjunt d'entrenament. En lloc de construir un BaggingClassifier i passar-li un DecisionTreeClassifier, es pot utilitzar la classe RandomForestClassifier, que és més convenient i està optimitzada per a Arbres de Decisió (Similarment, hi ha una classe RandomForestRegressor per a tasques de regressió, que és la que s'utilitza en aquest treball).

Amb algunes excepcions, un RandomForestClassifier té tots els hiperparàmetres d'un DecisionTreeClassifier (per controlar com es creixen els arbres), més tots els hiperparàmetres d'un BaggingClassifier per controlar el conjunt mateix.

L'algorisme de Bosc Aleatori introdueix una altra dosi d'atzar en el creixement dels arbres; en lloc de buscar la millor característica quan es divideix un node, cerca la millor característica entre un subconjunt aleatori de característiques. L'algorisme resulta en una major diversitat d'arbres, la qual cosa (una vegada més) intercanvia un major biaix per una menor variància, generalment produint un model global millor.

A continuació hi ha un codi que s'ha utilitzat per entrenar i validar la precisió dels models entrenats amb l'algoritme de Boscos de Decisió per les diferents bases de dades.

Codi 4.6: Codi Python per dividir la base de dades en entrenament i testeig, s'entrena el model Baseline i s'extreuen els resultats.

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

#Partir la base de dades
X_train, X_test, y_train, y_test =
   train_test_split(A, B, test_size=0.2,
   random_state=42)

#Entrenar el model
regressor = RandomForestRegressor(n_estimators=100,
   random_state=0)
regressor.fit(X_train, y_train)

#Extreure resultats
y_pred = regressor.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

Capítol 5

Resultats

5.1 Mesura de Validació

S'han utilitzat dues mesures de rendiment per avaluar els models: el coeficient de determinació R^2 (que és el que utilitzarem principalment per comparar resultats) i l'Error Quadràtic Mitjà o MSE. El coeficient de determinació proporciona una indicació de la precisió dels models en predir les dades observades, un valor proper a 1 indicaria una alta correlació entre les prediccions i les valoracions reals, mentre que un valor proper a 0 indicaria una baixa correlació, mentre que el MSE mesura la mitjana dels quadrats dels errors residuals.

Per comparar els models, s'ha analitzat la precisió dels models calculant R^2 i també es calcula els valors de MSE per tenir una millor visió. A més, s'han generat gràfics d'importància d'atributs per a cada model, proporcionant una visualització detallada del pes relatiu de cada atribut en la predicció. Primer es va fer un model de referència o *baseline* per a poder comparar i poder saber si els tractaments de dades aplicats són bons o no cal utilitzar-los.

La combinació d'aquestes dues mesures i els gràfics d'importància d'atributs ha estat crucial per a una avaluació completa dels models i ha guiat les decisions per a ajustos i millores en la metodologia.

5.2 Proves

En aquesta secció es mostraran tots els resultats de les proves realitzades al llarg d'aquest estudi.

5.2.1 Prova: Model de Referència

Primer cal fer un model de referència per a poder comparar els tractaments de dades. Aquest model s'ha entrenat utilitzant la base de dades *"listings"* amb el filtre de 100 comentaris per allotjament. Els camps utilitzats són els originals de la base de dades *"listings"* amb alguns dels tractaments:

- S'ha aplicat *One Hot Encoding*, Embedding i PCA a les variables esmentades a l'apartat anterior.
- No s'ha aplicat el tractament a les variables de tipus text classificades com a altres.
- No s'han creat nous atributs.

Score type	**mse_RL**	**mse_RF**	**r2_RL**	**r2_RF**
rating	0.0227	0.0202	0.3746	0.4442
accuracy	0.0168	0.0148	0.3326	0.4163
cleanliness	0.0287	0.0262	0.2398	0.3073
checkin	0.0135	0.0099	0.4178	0.5706
communication	0.0133	0.0118	0.3751	0.4446
location	0.0123	0.0117	0.2136	0.2513
value	0.0194	0.0179	0.3754	0.4264

Taula 5.1: Resultats dels models *baseline* de Regressió Lineal i *Random Forest*.

Aquesta taula conté els resultats dels dos models inicials. Com podem observar, els resultats utilitzant l'algoritme de *Random Forest* són significativament millors i, per tant, serà considerat com el primer model de referència. Aquest algorisme serà utilitzat per entrenar tots els models a partir d'ara.

(Els codis relacionats a aquest test estan en els documents *1-Estudi BDD.ipynb, 2-One-hot Encoding.ipynb i 3-First Model.ipynb*).

5.2.2 Prova: Número Mínim de Valoracions per Allotjament

En aquesta prova, es vol estudiar el compromís entre tenir més dades i tenir dades més sorolloses. Es busca la base de dades que doni millor resultat amb l'algoritme RF. Per a fer-ho, s'entrena el model amb l'algoritme RF amb diverses bases de dades, aquestes tindran un número mínim

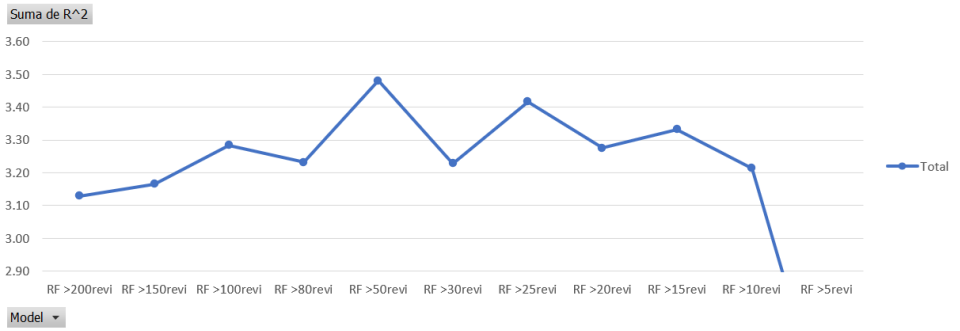


Figura 5.1: Precisió mitjana de models entrenats amb diferents restriccions mínimes de comentaris per allotjament.

de comentaris per allotjament: 200, 150, 100, 80, 50, 30, 25, 20, 15, 10 i 5. Per tant, el model que s'ha entrenat amb un mínim de 200 comentaris per allotjament tindrà poques dades i poc soroll respecte el model que s'entrena amb un mínim de 5 comentaris per allotjament, que tindrà moltes més dades, però seran més sorolloses.

Els atributs utilitzats són els originals de la base de dades *"listings"* amb alguns tractaments:

- S'ha aplicat *One Hot Encoding* a les variables *"neighbourhood_group_cleansed"*, *"neighbourhood_cleansed"*, *"property_type"*, *"room_type"*, *"host_response_time"* i el tractament descrit anterior ment a *"amenities"* sense PCA (per guanyar interpretabilitat).
- No s'han utilitzat les variables generades a través de l'*Embedding* ja que donaven resultats amb menys precisió.
- S'ha aplicat el tractament a les variables de tipus text classificades com a "altres".
- No s'han creat nous atributs.

En aquest gràfic (5.1) es pot observar com el número mínim de comentaris per allotjament que dona millors resultats i, per tant, amb un bon compromís entre més dades i més soroll, és utilitzar els allotjaments amb mínim 50 comentaris (a l'Annex es pot trobar la taula amb totes les precisions: 10). A partir d'ara, només s'utilitzarà la bases de dades amb mínim 50 comentaris per allotjament i per tant el nou model de referència és el RF entrenat amb aquesta base de dades, que li direm "RF_base_50".

(Els codis relacionats a aquest test estan en els documents *4-Num_min_rev.ipynb i model_1.py*).

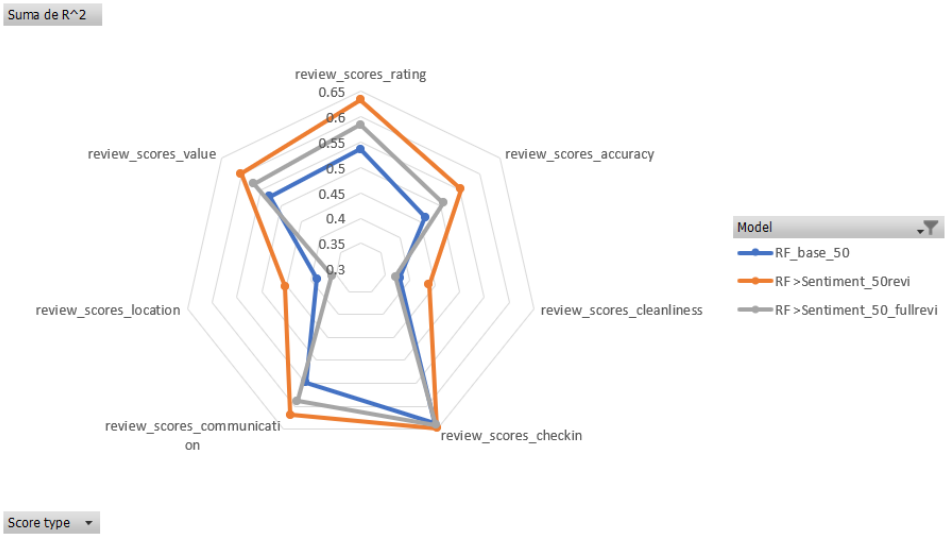


Figura 5.2: Comparativa de precisió R^2 dels models entrenats amb sentiment, contrastant amb el model base entrenat a la prova anterior.

5.2.3 Prova: Anàlisi de Sentiment

En aquesta prova, es creen 11 nous atributs a partir d'extreure el sentiment de cada comentari de la base de dades *"reviews"*. Aquestes noves variables són:

- "average_value_per_listing_id:" Que contindrà la mitjana del sentiment de les comentaris de cada allotjament.
- "senti:" Amb i entre 1 i 10, seran els 5 millors i pitjors sentiments dels comentaris de cada allotjament

El model "RF>Sentiment_50revi" està entrenat amb la mateixa base de dades que el model "RF_base_50", afegint només la variable que indica la mitjana del sentiment dels comentaris de cada allotjament. El model "RF>Sentiment_50_fullrevi" està entrenat amb la mateixa base de dades que el model "RF_base_50", afegint les 11 variables esmentades.

Els resultats (5.2) indiquen una clara millora de totes les valoracions dels allotjaments. Cal notar, que el model entrenat afegint només la variable "RF>Sentiment_50revi" ha funcionat millor que el que s'ha ensenyat amb la base de dades i els 11 nous atributs, el que ens indica que no sempre afegir variables millora la precisió, segurament aquestes "senti" 10 variables són molt sorolloses.

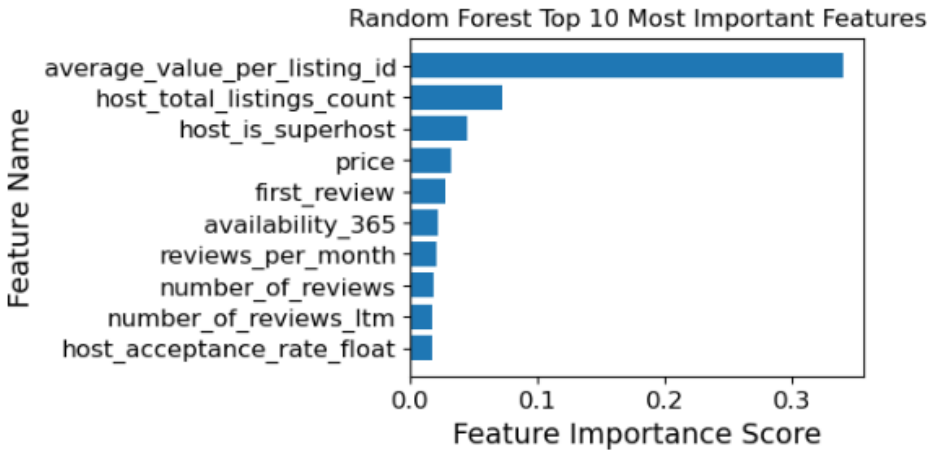


Figura 5.3: Importància d'atributs del model "RF>Sentiment_50revi" (la variable "average_value_per_listing_id" indica el sentiment mitjà de les valoracions de cada allotjament).

El gràfic 5.3 mostra la importància dels 10 atributs més importants del model "RF>Sentiment_50revi", ens indica que la variable amb més importància és la que s'ha afegit en aquesta prova: la mitjana del sentiment dels comentaris de cada allotjament. El gràfic d'importància del model "RF>Sentiment_50_fullrevi" (no adjuntat en aquesta memòria) conté 8 atributs de sentiment (dels 11 que s'han creat).

Aquestes variables milloren molt la precisió de totes les valoracions, però potser no són indicades per a l'objectiu global d'aquest projecte.

(Els codis relacionats a aquest test i a tots els pròxims, estan en els documents *5-Feature Engeniering.ipynb i model_1.py*).

5.2.4 Prova: Anàlisi amb ChatGPT

En aquesta prova es vol extreure informació de la variable *"description"*, que conté la descripció dels allotjaments. S'han creat 4 noves variables:

- "size": Aquesta variable indica els metres quadrats de l'allotjament i retorna 0 en cas de que no s'especifiqui.
- "capacity": Aquesta variable indica la capacitat màxima de persones permeses i retorna 0 en cas de que no s'especifiqui.
- "alowed_under_25": Aquesta variable conté un 1 si permet joves menors de 25 anys (explicitament) i 0 si no s'especifica res i -1 en el cas de que prohibeixi els menors de 25 anys.

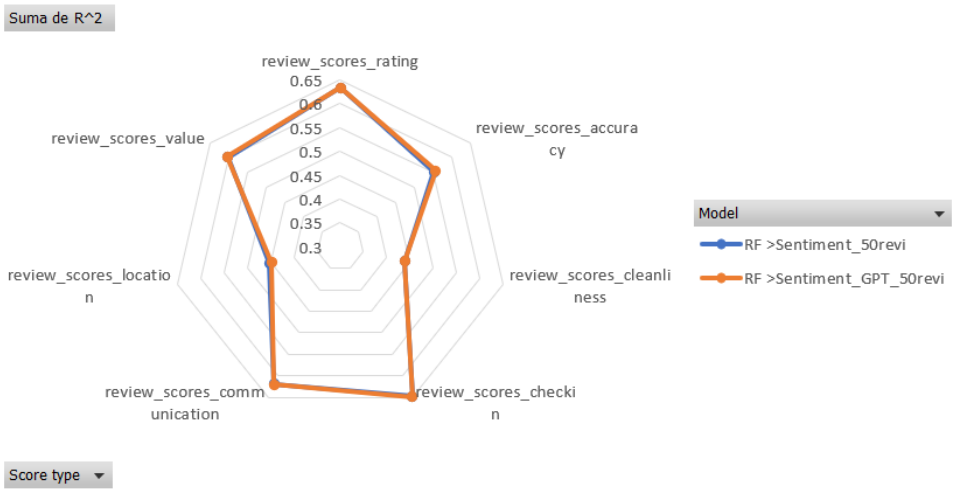


Figura 5.4: Comparativa de la precisió R^2 del model de rèferència i el model entrenat amb les variables creades amb l'anàlisi de la descripció utilitzant ChatGPT. (No hi ha millora)

- "family_friendly": Aquesta variable conté un 1 si explicita que l'allotjament està enfocat a families, 0 en el cas de que no s'especifiqui i -1 en el cas que no es recomani l'allotjament a famílies.

Per a extreure aquestes noves variables de la descripció, s'ha utilitzat el paquet *g4f*, que fa crides a ChatGPT de forma gratuïta. No s'han aconseguit extreure les dades de forma consistent, ja que molts cops no classificava correctament cap de les variables. Degut a això, els resultats d'entrenar un model amb o sense aquestes noves variables, són idèntics com es pot oservar en el gràfic 5.4.

5.2.5 Prova: Localització

En aquesta prova es vol millorar la predicció de valoració de la localització dels models. Per a fer-ho, s'ha creat una variable nova per a cada punt de referència del mapa 3.3 i aquestes variables contentn la distància euclídea a cada allotjament. Els resultats han siguit molt positius, ja que s'ha millorat la precisió de totes les valoracions. També s'ha entrenat un model que solament prediu la valoració de localització, la qual ha donat millor resultats que entrenar el mateix algoritme RF, però que en lloc de només predir la valoració de localització, també ha de predir els altres 6 tipus de valoracions. Anem a veure els resultats dels diferents models.

La taula 5.6 mostra la precisió de predir la valoració de localitza-

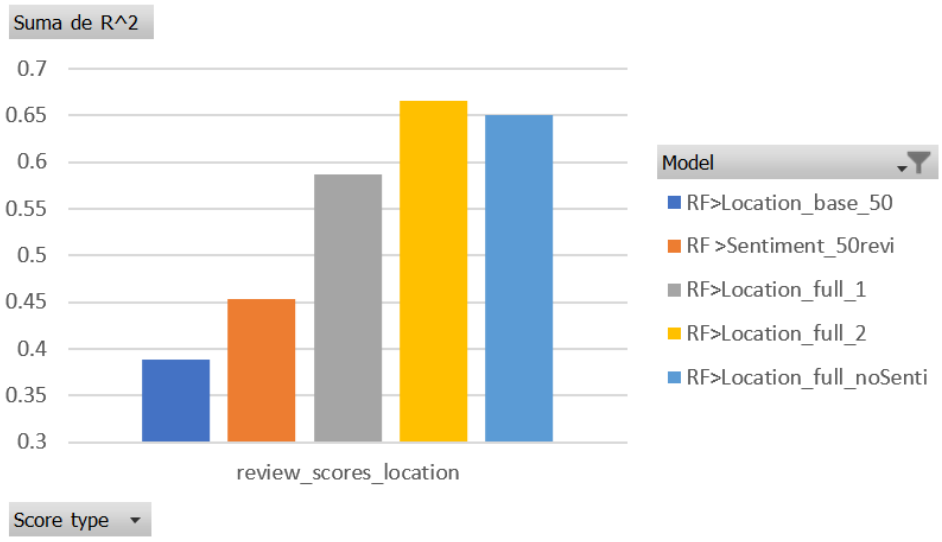


Figura 5.5: Commparativa de precisió R^2 de les prediccions de localització de diferents models.

ció dels diferents models entrenats. Els models "RF>Location_base_50" i "RF>Sentiment_50revi" són entrenats en proves anteriors, que prediuen totes les valoracions. En canvi els altres tres models, només prediuen la valoració de localització; la base de dades utilitzada per als tres models és la mateixa que s'ha utilitzat per entrenar "RF>Location_base_50", però afegint algunes variables més:

- "RF>Location_full_noSenti": Se li ha afegit les variables de distància euclidiana i les de GPT.
- "RF>Location_full_1": Se li ha afegit les variables de distància euclidiana, les 11 variables de sentiment i les variables creades per GPT.
- "RF>Location_full_2": Se li ha afegit les variables de distància euclidiana, la mitjana del sentiment dels comentaris de cada allotjament i les variables creades per GPT.

En el gràfic 5.6, s'hi pot observa la comparativa entre els diferents models predictors de totes les valoracions. Els models "RF>Location_base_50" i "RF> Sentiment _50revi" són entrenats en proves anteriors. El "RF> Location_full_3" està entrenat amb la mateixa base de dades que "RF>Location _full_1", eliminant les columnes "sent{i}".

Per tant, es pot observar com el millor model predictor de totes les valoracions ha estat entrenat amb els següents tractaments:

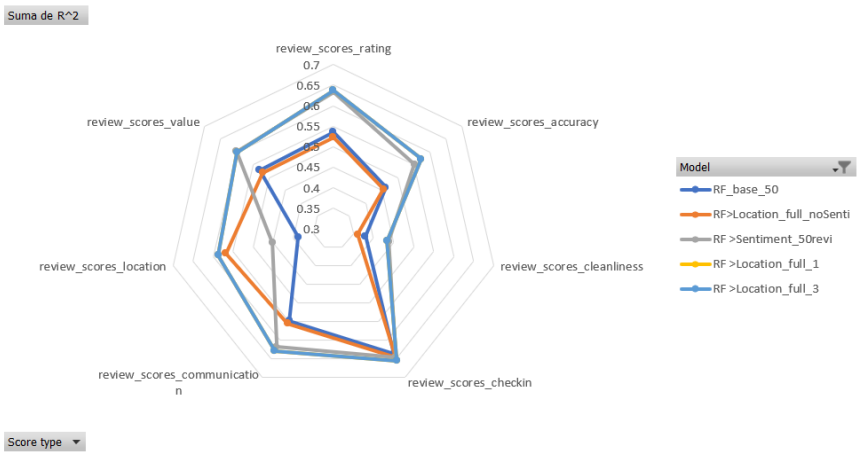


Figura 5.6: Commparativa de precisió R^2 de models entrenats al llarg del projecte.

- S'han tractat els valors extranys i els valors que faltaven de forma adequada i s'han escalat les variables de tipus numèric.
- S'ha aplicat *One Hot Encoding* a les variables *"neighbourhood_group_cleansed"*, *"neighbourhood_ cleansed"*, *"property_type"*, *"room_type"*, *"host _response_time"* i el tractament especial a *"amenities"* sense PCA.
- No s'han utilitzat les variables generades a través de l'*Embedding* ja que donaven resultats amb menys precisió.
- S'ha aplicat el tractament a les variables de tipus text classificades com a "altres".
- S'ha creat una variable de mitjana de sentiment dels comentaris.
- S'ha creat diverses variables de distància a punts concrets de Barcelona per millorar la representació de la localització.

És important remarcar, que s'ha millorat el poder predictiu en totes les valoracions al llarg de les iteracions. En particular, la valoració de localització és la que ha patit un guany més significatiu, millorant de 0.21 fins a 0.66 (10 i 11). És interessant notar que, entrenar un RF per predir totes les valoracions dona pitjors resultats que entrenar un RF general, això s'haurà de tenir en compte a l'hora de millorar la precisió a través de la millora del model.

Capítol 6

Conclusions i Futur Treball

Els objectius establerts per a aquest estudi s'han aconseguit amb èxit. Això ha implicat la recopilació i anàlisi d'una base de dades rellevant per predir les avaluacions dels allotjaments d'Airbnb, així com el preprocés de les dades i, finalment, l'entrenament i estudi de models d'aprenentatge automàtic per predir les avaluacions dels allotjaments.

S'ha millorat significativament la precisió de les prediccions de totes les valoracions a través de les millores en el processament de les dades. És destacable que algunes valoracions són més predictibles que d'altres, com és el cas de la localització, on hem aconseguit duplicar la precisió respecte al cas base.

Pel que fa al futur treball, es podria continuar explorant maneres de millorar la precisió de les prediccions de les valoracions mitjançant l'ajust d'algoritmes d'aprenentatge i la variació dels hiperparàmetres. A més, es podria abordar la interpretació dels models, amb l'objectiu de proporcionar un resum executiu que sigui fàcil d'interpretar per als usuaris, facilitant la presa de decisions. Finalment, per a una implementació pràctica, es podria desplegar el projecte en una pàgina web, actualitzant els models amb noves dades i adaptades per a cada regió i presentant informes de manera altament interpretable.

Bibliography

[1] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow", 2019.

[2] A. Géron, repositori Github del llibre "Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow", 2019.
`https://github.com/ageron/handson-ml2`

[3] C. Molnar, Interpretable machine learning, 2022.

[4] Pàgina web Inside Airbnb
`http://insideairbnb.com/`

[5] Pàgina web AirDNA
`https://www.airdna.co`

[6] J. Orteu, repositori Github del projecte "Interpreting-Credit-Score-Models", 2024.
`https://github.com/Joanorteu99/Interpreting-Credit-Score-Models`

Annex A: Taules

id	host_response_time	latitude	minimum_maximum_nights	last_review
listing_url	host_response_rate	longitude	maximum_maximum_nights	review_scores_rating
scrape_id	host_acceptance_rate	property_type	minimum_nights_avg_ntm	review_scores_accuracy
last_scraped	host_is_superhost	room_type	maximum_nights_avg_ntm	review_scores_cleanliness
source	host_thumbnail_url	accommodates	calendar_updated	review_scores_checkin
name	host_picture_url	bathrooms	has_availability	review_scores_communication
description	host_neighbourhood	bathrooms_text	availability_30	review_scores_location
neighborhood_overview	host_listings_count	bedrooms	availability_60	review_scores_value
picture_url	host_total_listings_count	beds	availability_90	license
host_id	host_verifications	amenities	availability_365	instant_bookable
host_url	host_has_profile_pic	price	calendar_last_scraped	calculated_host_listings_count
host_name	host_identity_verified	minimum_nights	number_of_reviews	calculated_host_listings_count_entire_homes
host_since	neighbourhood	maximum_nights	number_of_reviews_ltm	calculated_host_listings_count_private_rooms
host_location	neighbourhood_cleansed	minimum_minimum_nights	number_of_reviews_l30d	calculated_host_listings_count_shared_rooms
host_about	neighbourhood_group_cleansed	maximum_minimum_nights	first_review	reviews_per_month

Table 1: Aquesta taula conté tots els atributs de la base de dades “listings”.

listing_id	date	reviewer_name
id	reviewer_id	comments

Table 2: Aquesta taula conté els atributs de la base dades “reviews”

listing_id	date	available
price	adjusted_price	minimum_nights
maximum_nights		

Table 3: Aquesta taula conté els atributs de la base dades “calendar”

Variable	Correlació
review_scores_rating	1.000000
review_scores_accuracy	0.917143
review_scores_value	0.913943
review_scores_cleanliness	0.838268
review_scores_communication	0.797940
review_scores_checkin	0.757434
review_scores_location	0.514725

Table 4: Aquesta taula conté la correlació entre la valoració global i les altres valoracions.

Variable	Rating	Checkin	Communi	Location	Value
host_is_superhost	0.447795	0.355040	0.398738	0.212574	0.379004
Extra pillows and blankets	0.270498	0.267787	0.247198		0.280403
soap	0.227851	0.196948	0.199217		0.202229
parking	0.217668	0.247025	0.227889		0.214308
First aid kit	0.214844	0.235148	0.202435		0.243617
Room-darkening shades	0.214027	0.188156			
Carbon monoxide alarm	0.208656	0.187388	0.197678		
number_of_reviews	0.204902	0.221893	0.227014	0.112241	0.255112
books	0.202167	0.184789	0.191174	0.103755	0.193249
Host greets you	0.194115	0.246449	0.256893	0.115131	0.212924

Table 5: Els 10 atributs més coorelacionats amb les valoracions.

Variable	Location
review_scores_location	1.000000
neighbourhood_cleansed_la Dreta de l'Eixample	0.273154
neighbourhood_group_cleansed_Eixample	0.260270
host_is_superhost	0.212574
neighbourhood_cleansed_Sant Pere, Santa Caterina i la Ribera	0.157851
neighbourhood_cleansed_el Barri Gòtic	0.122624
neighbourhood_group_cleansed_Ciutat Vella	0.115193
Host greets you	0.115131
number_of_reviews	0.112241
books	0.103755
Fire extinguisher	0.102584

Table 6: Els 10 atributs més positivament correlacionats amb la valoració de la Ubicació.

Ubicació	Correlació
Eucld_Catedral BCN	-0.432894
Eucld_Plaça catalunya	-0.431912
Eucld_Ciutadella	-0.410291
Eucld_Liceu	-0.397391
Eucld_La Pedrera	-0.350497
Eucld_Raval	-0.350484
Eucld_Clínic	-0.328364
Eucld_Barceloneta	-0.316939
Eucld_Razzmatazz	-0.292228
Eucld_Sagrada Família	-0.266382
neighbourhood_cleansed_el Carmel	-0.227089

Table 7: Els deu atributs amb correlació inversa més important per a la valoració de la Localització.

Atribut	Rating
calculated_host_listings_count	-0.359
calculated_host_listings_count_entire_homes	-0.353
host_total_listings_count	-0.337
host_listings_count	-0.327
Smart lock	-0.212
availability_90	-0.187
availability_30	-0.165
availability_60	-0.162
property_type_Entire rental unit	-0.154
availability_365	-0.151
first_review	-0.145

Table 8: Els deu atributs amb correlació inversa més important per a la valoració de la valoració global.

	rating	accuracy	cleanliness	checkin	communication	location	value
count	2320	2320	2320	2320	2320	2320	2320
mean	4.69	4.767	4.728	4.823	4.831	4.813	4.625
std	0.186	0.155	0.192	0.143	0.138	0.131	0.172
min	3.84	3.57	3.72	3.7	4.04	3.95	3.89
0.25	4.58	4.68	4.63	4.76	4.77	4.75	4.52
0.5	4.72	4.8	4.77	4.86	4.86	4.84	4.65
0.75	4.83	4.88	4.87	4.92	4.93	4.91	4.75
max	5	5	5	5	5	5	4.96

Table 9: Taula de resum de les diferents valoracions de la taula *listings*.

Etiquetes de Fila	**Rating**	**Accuracy**	**Cleanliness**	**Checkin**	**Communication**	**Location**	**Value**	**Total General**
RF >200 revi	0.535	0.494	0.407	0.442	0.451	0.300	0.499	3.129
RF >150 revi	0.514	0.476	0.394	0.544	0.498	0.242	0.497	3.165
RF >100 revi	0.501	0.459	0.373	0.615	0.520	0.319	0.495	3.284
RF >80 revi	0.530	0.495	0.402	0.566	0.489	0.246	0.506	3.232
RF >50 revi	0.536	0.462	0.379	0.638	0.548	0.388	0.529	3.480
RF >30 revi	0.511	0.442	0.411	0.599	0.505	0.269	0.490	3.228
RF >25 revi	0.526	0.477	0.398	0.620	0.517	0.351	0.527	3.417
RF >20 revi	0.525	0.497	0.382	0.572	0.487	0.285	0.528	3.276
RF >15 revi	0.542	0.485	0.431	0.570	0.501	0.281	0.521	3.332
RF >10 revi	0.521	0.467	0.419	0.521	0.468	0.303	0.515	3.214
RF >5 revi	0.358	0.357	0.314	0.428	0.394	0.222	0.366	2.439
Total General	5.600	5.113	4.310	6.113	5.380	3.207	5.474	35.197

Table 10: Precisió de R^2 de models entrenats amb diferents quantitats mínimes de valoracions i el mateix tractament inicial de variables.

Annex B: Imatges

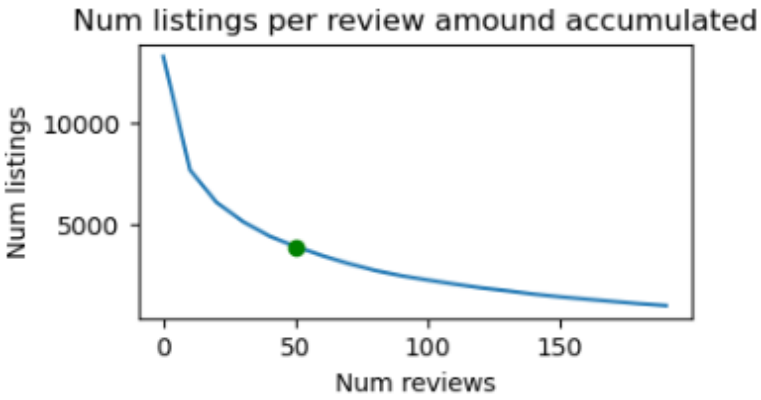


Figure 1: En aquest gràfic es pot observar la disminució d'allotjaments a mesura que augmentem la restricció de número mínim de comentaris. Aquest gràfic ens ajuda a determinar el punt de tall.

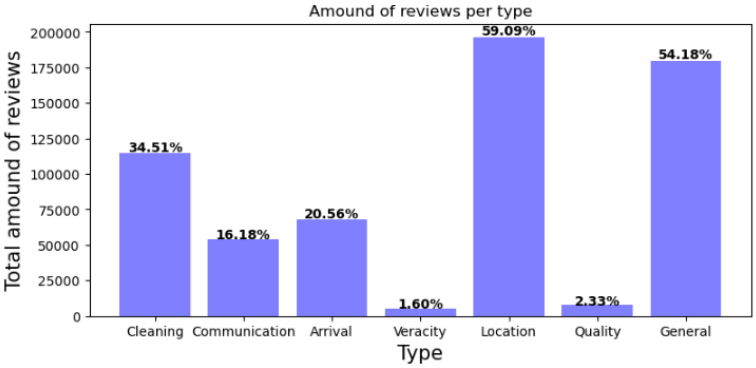


Figure 2: Classificació de comentaris segons el tema del que tracten

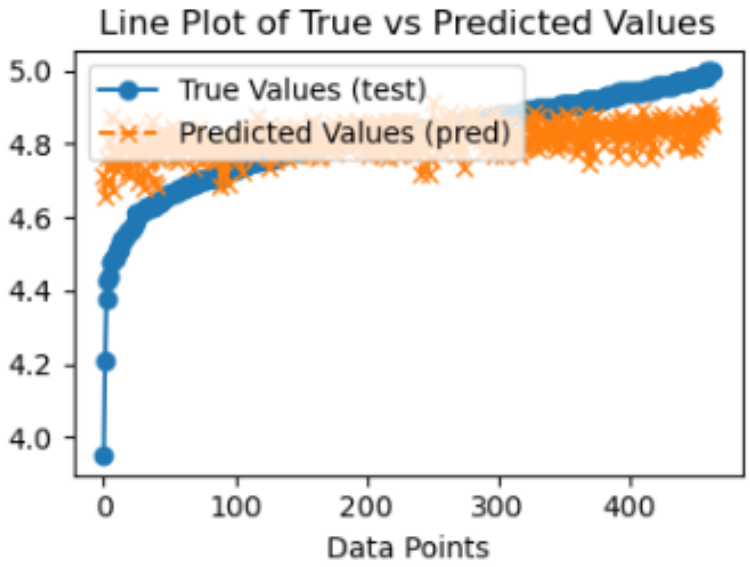


Figure 3: En aquest gràfic s'hi mostra en blau els resultats esperats i en taronja els resultats predits de valoració de Localització dels allotjaments pel model *Baseline*.

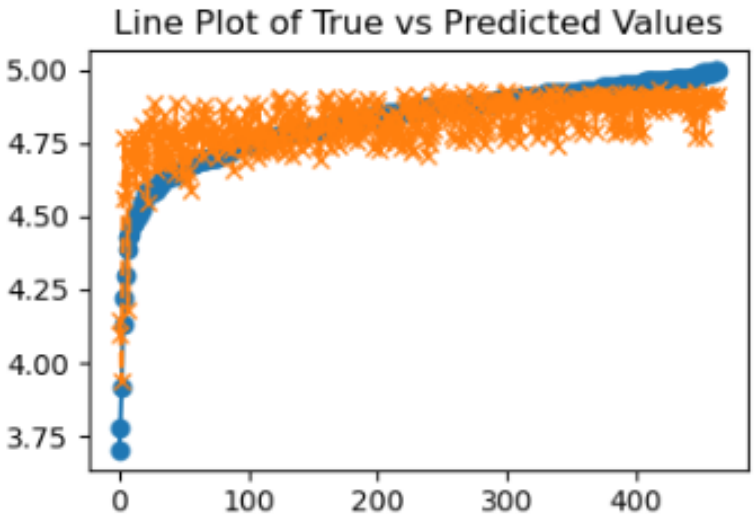


Figure 4: En aquest gràfic s'hi mostra en blau els resultats esperats i en taronja els resultats predits de valoració del Check-In dels allotjaments pel model *Baseline*.

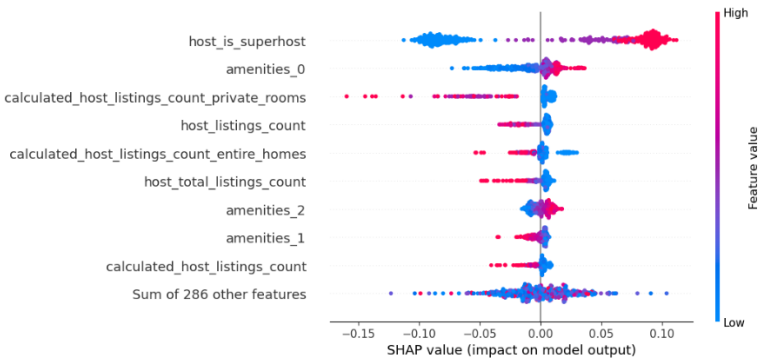


Figure 5: En aquest gràfic s'hi mostra una visió global de les contribucions marginals SHAP dels atributs més importants per al model *baseline*.

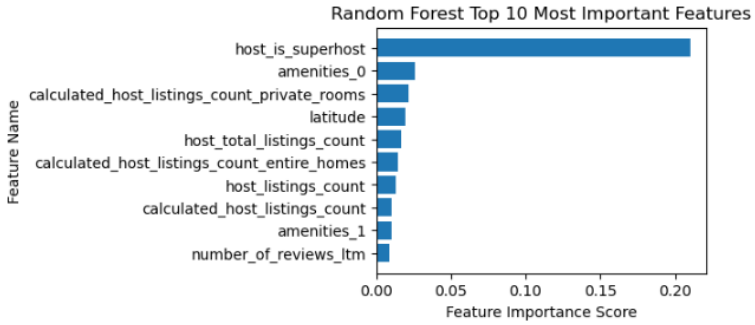


Figure 6: En aquest gràfic s'hi mostren els 10 atributs més importants i la seva importància del model *baseline*.

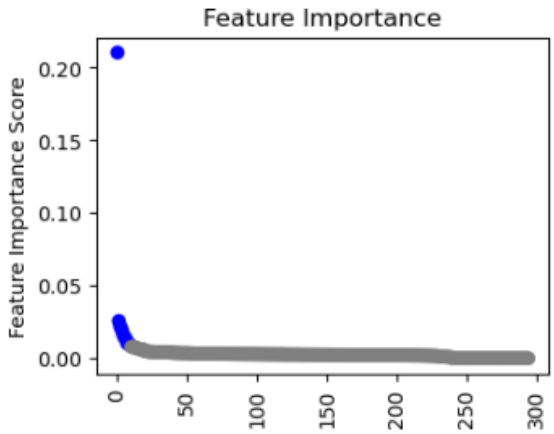


Figure 7: En aquest gràfic s'hi mostren tots els atributs i la seva importància del model *baseline*.

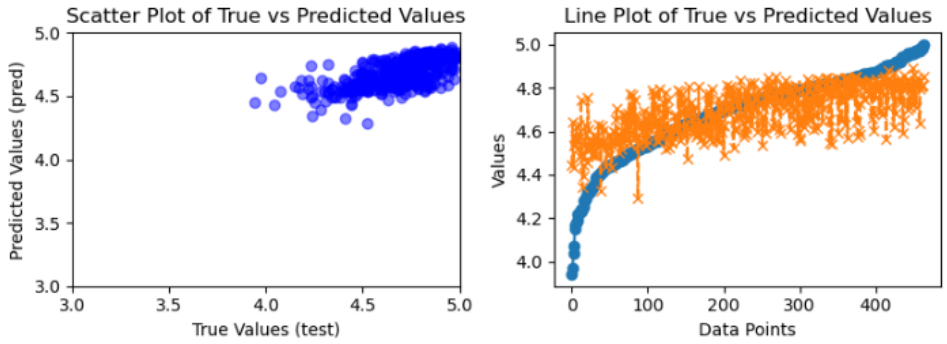


Figure 8: En aquest gràfic s'hi mostra la precisió de valoració global del model de referència.

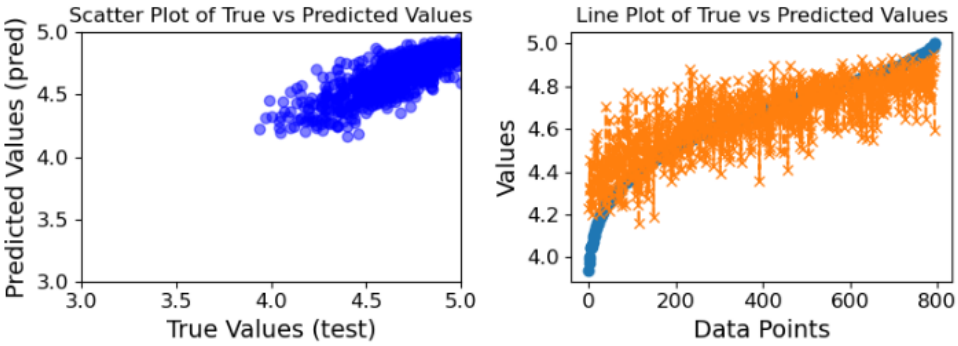


Figure 9: En aquest gràfic s'hi mostra la precisió de valoració global del millor model entrenat.

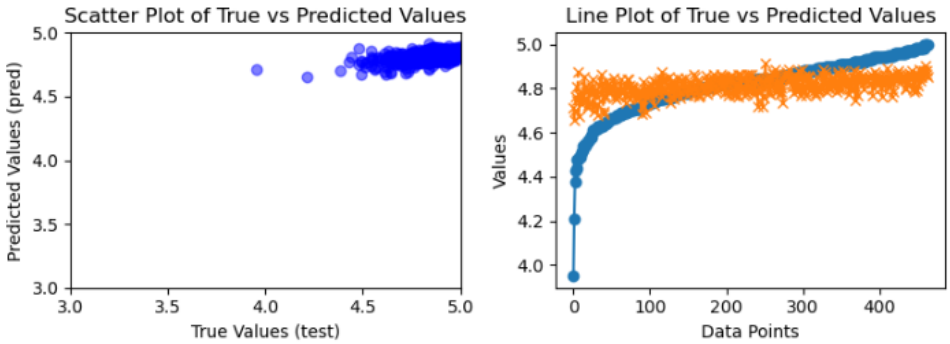


Figure 10: En aquest gràfic s'hi mostra la precisió de localització del model de referència.

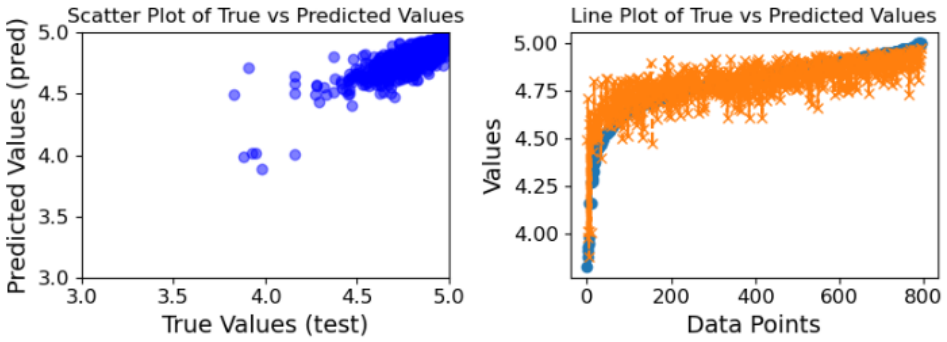


Figure 11: En aquest gràfic s'hi mostra la precisió de localització del millor model entrenat.