Sentiment Analysis: Product Reviews

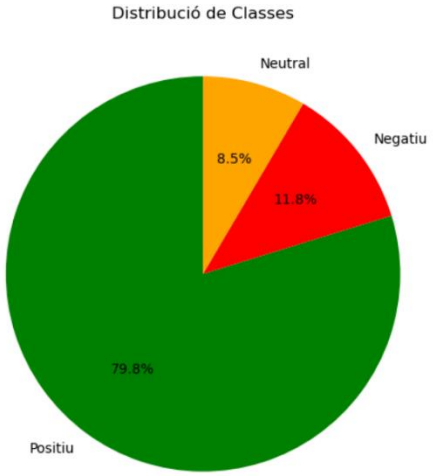
Introducció i Dataset

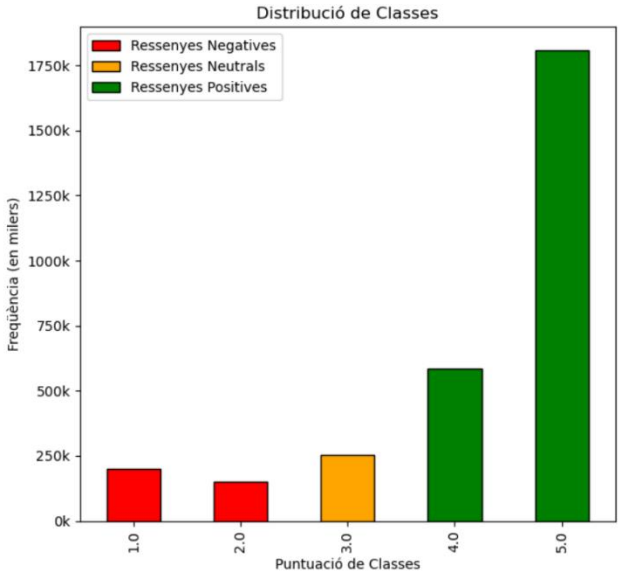
En aquest informe documentarem el treball dut a terme en el marc del nostre projecte de Sentiment Analysis. Aquest té com a objectiu analitzar diverses ressenyes d'un producte, en aquest cas llibres, per determinar-ne la puntuació i dictaminar el tipus de ressenya, fent una classificació multiclasse.

Mitjançant diversos models de Machine Learning, es pretén comparar-ne l'eficàcia amb datasets balancejats i desequilibrats, centrant-se en la identificació de patrons en les ressenyes textuals per predir sentiments associats a les puntuacions.

Inicialment tenim 3.000.000 d'instàncies al nostre dataset, de les quals n'agafarem les 3 columnes/característiques més importants que són l'ID de la ressenya, la puntuació donada i el comentari de text.

D'aquesta manera, agrupem el conjunt de puntuacions, les quals oscil·len entre 1 i 5, en 3 categories definides com Negativa (puntuacions 1-2), Neutre (3) i Positiva (4-5).





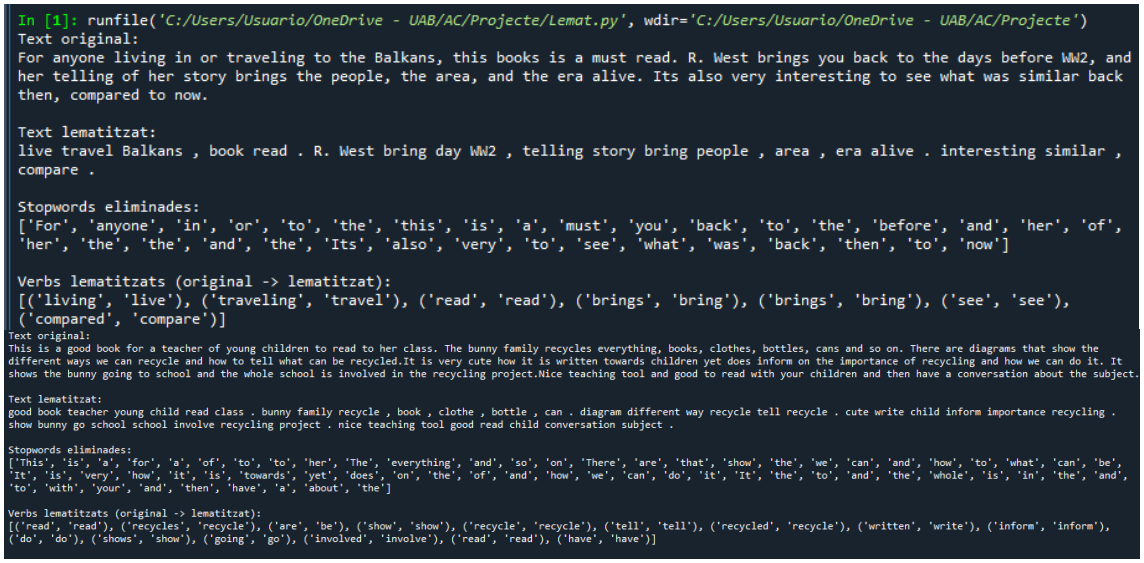
A partir d'aquí agafarem un conjunt no balancejat de 100.000 ressenyes aleatoritzades, i un altre de 100.000 però balancejat, amb la mateixa quantitat de ressenyes negatives, neutres i positives (20.000 de cada puntuació).

Preprocessament de Dades

Una bona implementació del projecte depèn en gran mesura d'un preprocessament adequat de les dades textuals amb les que treballem. En aquest sentit, hem seguit els següents passos:

Hem aplicat la **Lematització**, un procés que redueix les paraules a la seva forma base o "lemma" per unificar termes amb significats semblants (per exemple, "living", "lived" i "lives" es redueixen a "live"). Així, ajudem a reduir la variabilitat del llenguatge i millorar la qualitat de l'anàlisi.

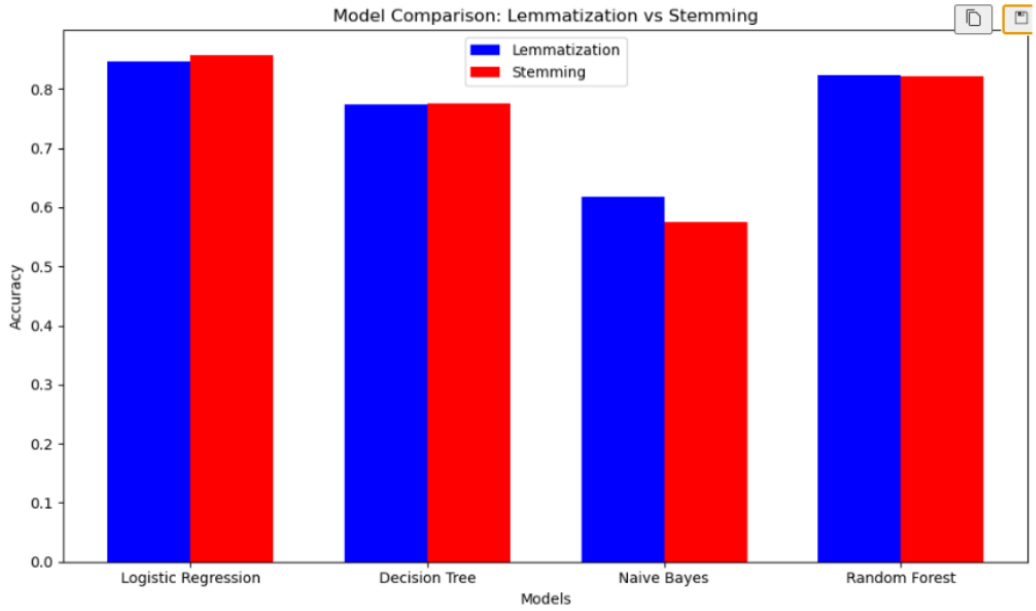
Algunes de les comprovacions que fem per veure com es lematitzen els texts:



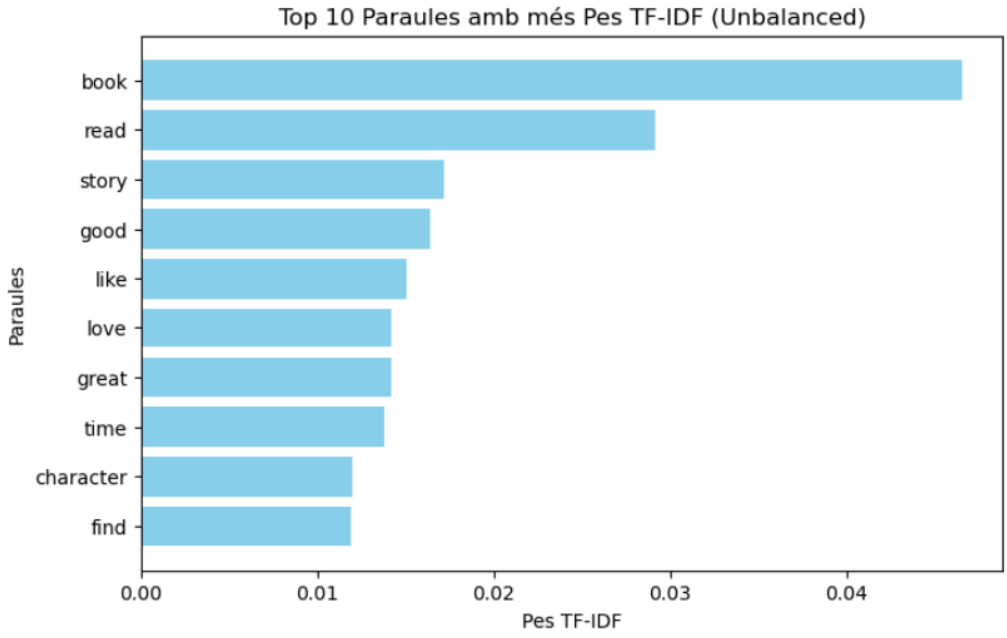
Suprimim també les **Stop Words**, les paraules comunes que no tinguin un valor semàntic important o que no siguin rellevants (com articles o connectors: "the", "in", "to", etc.).

Posteriorment també eliminarem **valors NaN**, on les dades són buides; els **caràcters extra** innecessaris com exclamacions, arroves, etc. també es suprimiran, igual que els **duplicats**.

Cal destacar que també vàrem provar d'utilitzar Stemming enlloc de lematització en els models que explicarem posteriorment, el qual redueix les paraules al seu lexema base (tot i que pot presentar errades com passar de "better" a "bet"), però no hi va haver diferències significatives:

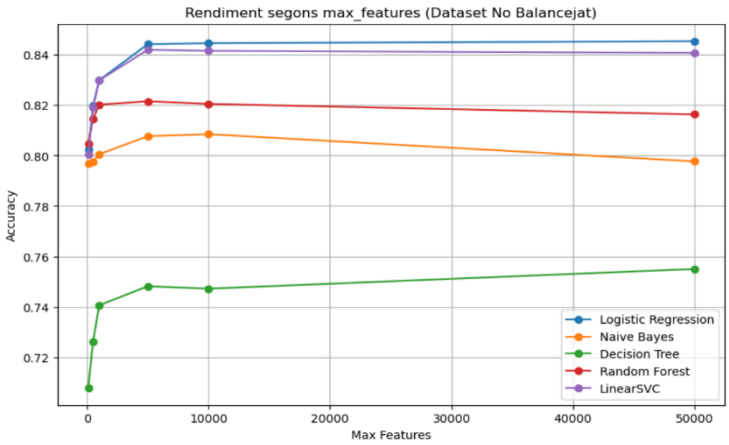


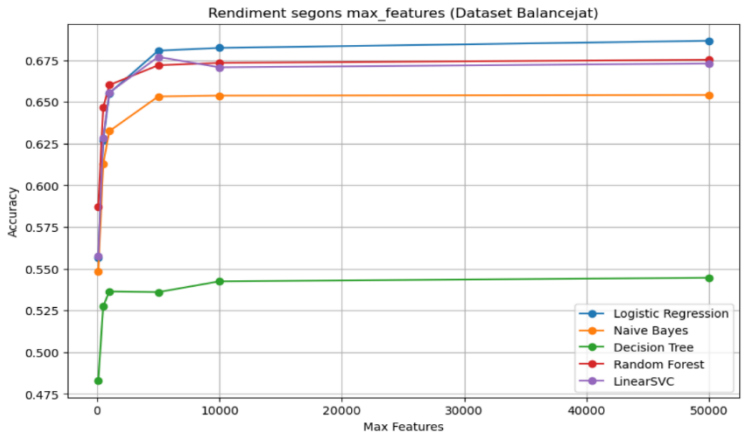
Per poder passar el text a un model, farem servir la **vectorització TF-IDF**, transformant el text en vectors numèrics que capturen la importància relativa de cada paraula. Amb això podem veure algunes de les paraules amb més freqüència relativa:



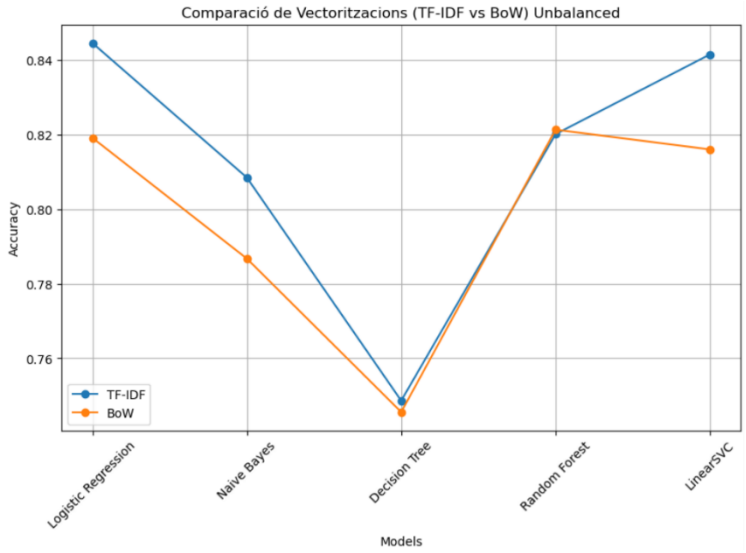
Per determinar la mida òptima del vector TF-IDF, hem provat diferents valors de max_features i hem comparat el rendiment dels models de classificació utilitzats. Després d'analitzar els resultats, vam seleccionar una mida de **10.000 paraules**, ja que oferia un millor rendiment global en la classificació.

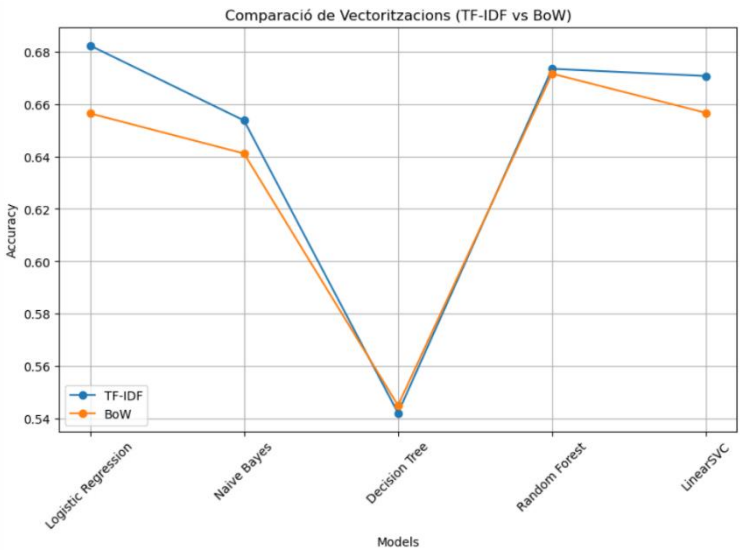
Els gràfics adjunts mostren com el rendiment dels models canvia segons el valor de max_features, tant per al conjunt de dades balancejat com per al no balancejat. Més enllà de 10.000, l'augment del vocabulari no va suposar una millora substancial en els resultats.



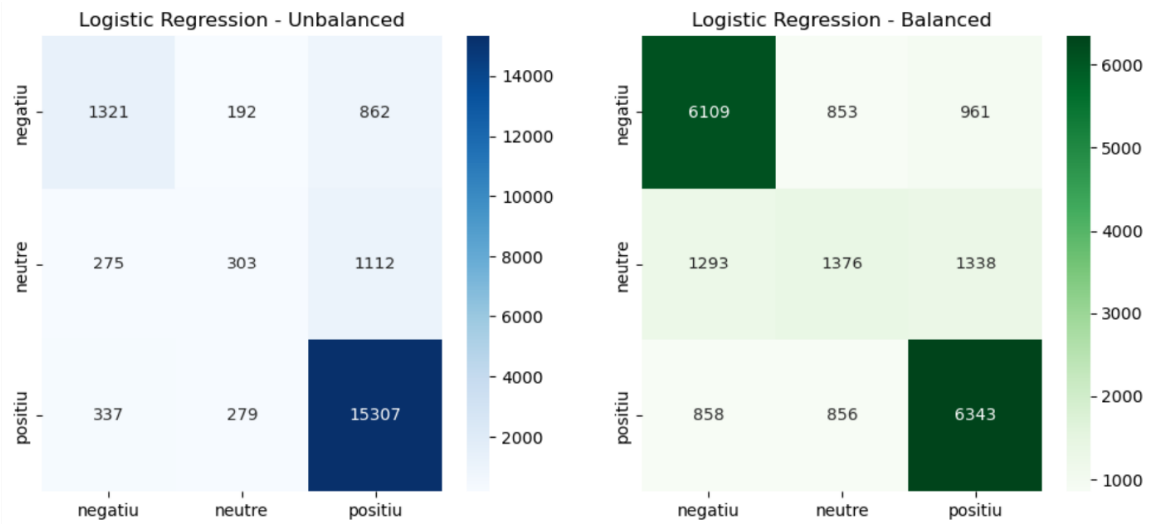


També vam comparar la vectorització TF-IDF amb la de Bag of Words, per comprovar quina funcionava millor de les dues, quedant-nos amb la primera ja que dona resultats lleugerament superiors:





Model: Logistic Regression



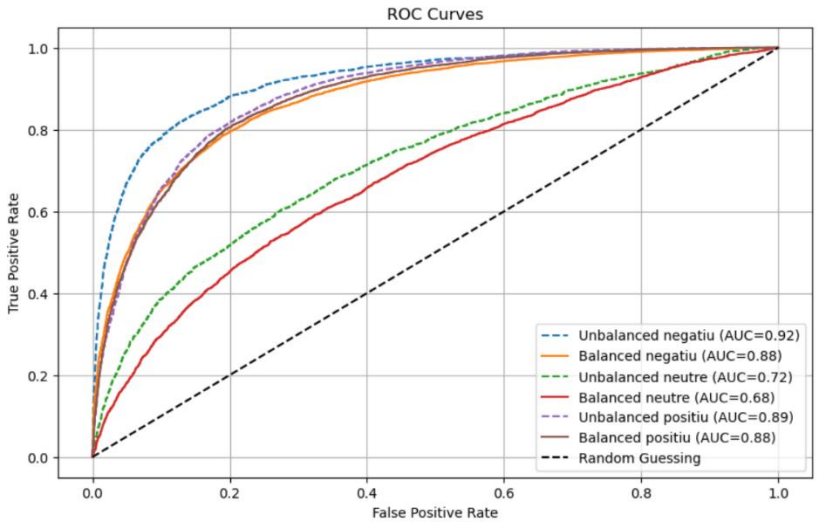
	precision	recall	f1-score	support
negatiu	0.75	0.48	0.59	2375
neutre	0.51	0.08	0.14	1690
positiu	0.86	0.98	0.92	15923
accuracy			0.85	19988
macro avg	0.71	0.52	0.55	19988
weighted avg	0.82	0.85	0.81	19988

	precision	recall	f1-score	support
negatiu	0.73	0.81	0.77	7923
neutre	0.51	0.24	0.33	4007
positiu	0.72	0.83	0.77	8057
accuracy			0.71	19987
macro avg	0.65	0.63	0.62	19987
weighted avg	0.68	0.71	0.68	19987

Comparem el nombre de prediccions correctes i incorrectes. En aquest primer model veiem que, amb el dataset unbalanced, té molts encerts en la classe “positiu” (la dominant), i presenta importants dificultats en les altres dues classes menys representades de “neutre” i “negatiu”, mostrant un biaix cap a la classe dominant.

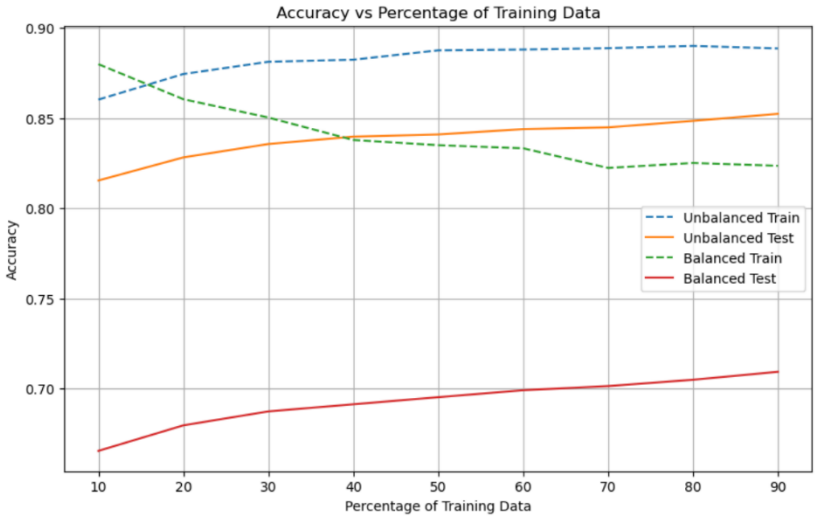
En canvi, amb el datset balanced el model distribueix millor els encerts entre les tres classes, sacrificant alguns positius en favor de les altres dues classes neutres i negatius (tot i seguir tenint dificultats especialment per la classe neutre, però no errant tant com anteriorment), amb resultats més equilibrats.

Això reflecteix que balancejar les dades ajuda a tractar el biaix i millora la precisió en les classes menys representades.

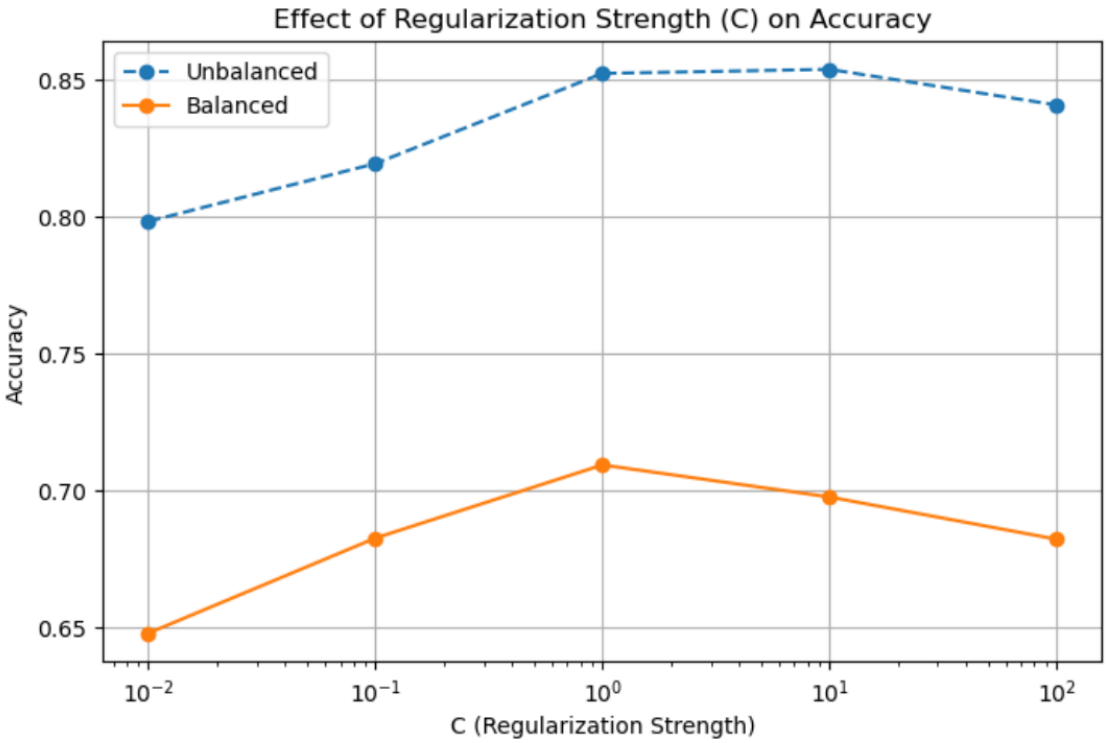


Seguim amb les corves ROC per mostrar la relació entre el True Positive Rate i el False Positive Rate per cada classe; on l'àrea sota la corba (AUC) indica com de bé el model pot distingir les classes, com més s'apropi a 1 millor és la classificació, i com més s'apropi a 0.5 més s'apropa a una classificació aleatòria.

Podem observar que en el dataset unbalanced, el model aconsegueix una AUC elevada per a la classe dominant i també per a la classe "negatiu", mentre que per a la classe "neutre" és considerablement més baixa. Això reflecteix que el model està més adaptat a predir correctament la classe majoritària i les classes amb patrons més clars, però té dificultats amb la classe "neutre". En canvi, en el dataset balanced, les AUC per a totes les classes disminueixen lleugerament. Això es pot atribuir al fet que el balanç redistribueix el focus del model, sacrificant lleugerament el rendiment de les classes més fàcils de predir per intentar capturar millor les classes minoritàries, encara que en aquest cas no ha millorat el rendiment per a la classe "neutre".



Mirem les divergències entre línies de train i test que poden indicar overfitting o underfitting. Agafant l'exemple de les dades balancejades, veiem que l'accuracy en train és alt però en test és força baix, indicant un cert overfitting (on el model aprèn massa dels detalls del train i no generalitza bé). Com més gran es va fent el percentatge de dades, més es redueix aquesta "disparitat" i veiem una lleugera convergència. Així amb datasets balancejats, l'augment de dades té un impacte més consistent i positiu en el rendiment del model.

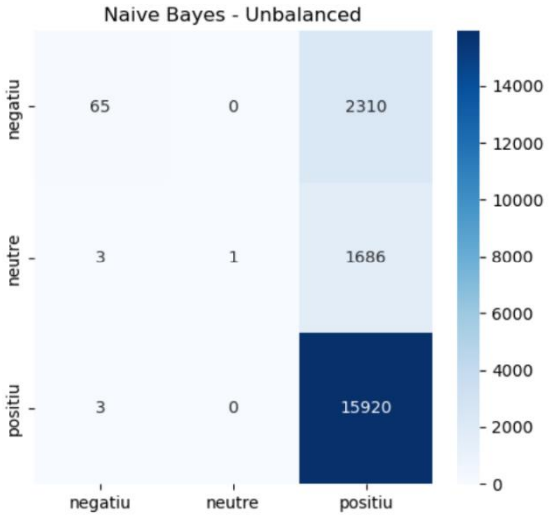


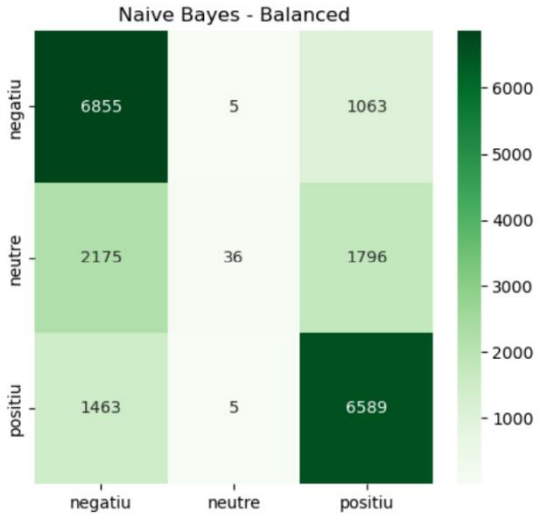
Per acabar amb Logistic Regression, ens fixarem en l'impacte del hiperparàmetre C que controla el balanç entre el biaix (error per subentrenament) i la variància (error per sobreentrenament). Un C baix evita l'overfitting però pot subentrenar el model, mentre que un C alt permet el model ajustar-se millor a les dades però pot causar overfitting.

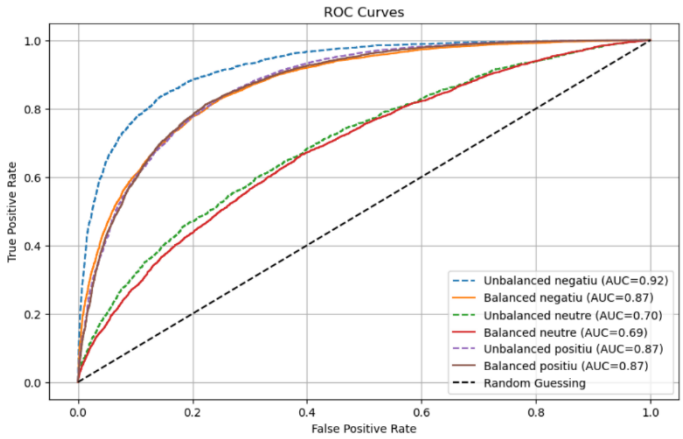
Veiem que els valors intermedis de C (C=1, C=10) maximitzen l'accuracy en els dos tipus de dades, evitant tant l'underfitting com l'overfitting.

Les diverses probes d'altres models son similars, per tant hi haurà execucions semblants, sense tants resultats a destacar.

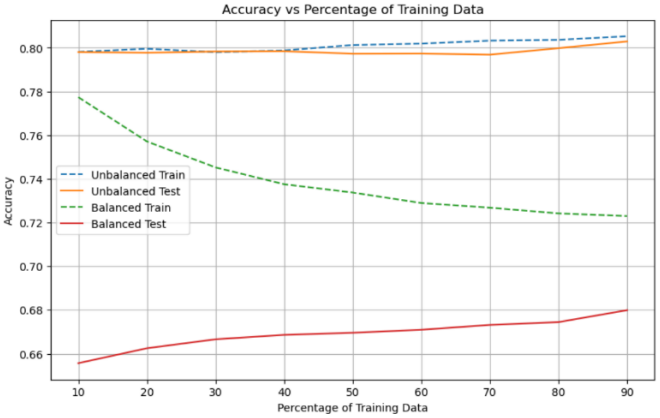
Model: Naive Bayes

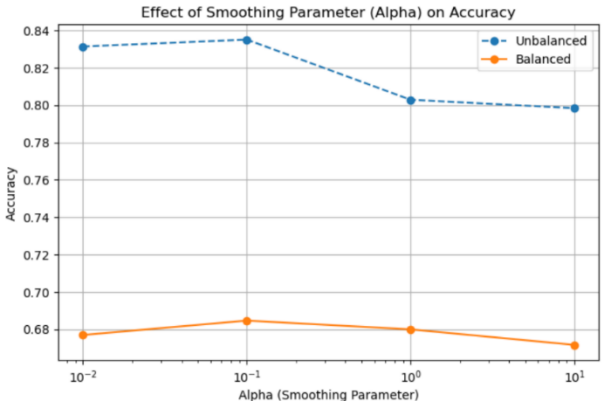






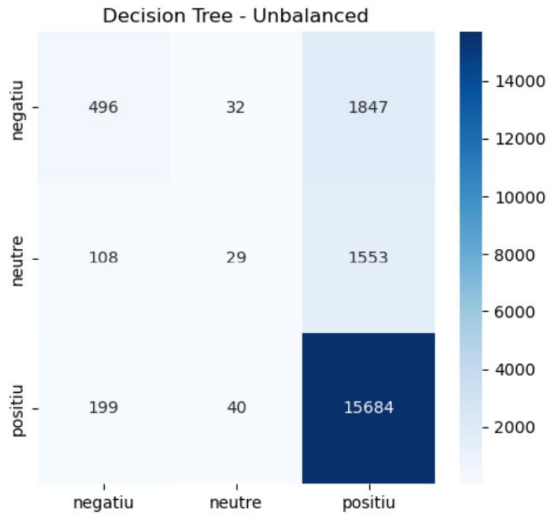

```
Naive Bayes - Unbalanced Dataset

              precision    recall  f1-score   support

     negatiu       0.92      0.03      0.05      2375
      neutre       1.00      0.00      0.00      1690
     positiu       0.80      1.00      0.89     15923

    accuracy                           0.80     19988
   macro avg       0.90      0.34      0.31     19988
weighted avg       0.83      0.80      0.71     19988

Naive Bayes - Balanced Dataset

              precision    recall  f1-score   support

     negatiu       0.65      0.87      0.74      7923
      neutre       0.78      0.01      0.02      4007
     positiu       0.70      0.82      0.75      8057

    accuracy                           0.67     19987
   macro avg       0.71      0.56      0.51     19987
weighted avg       0.70      0.67      0.60     19987
```

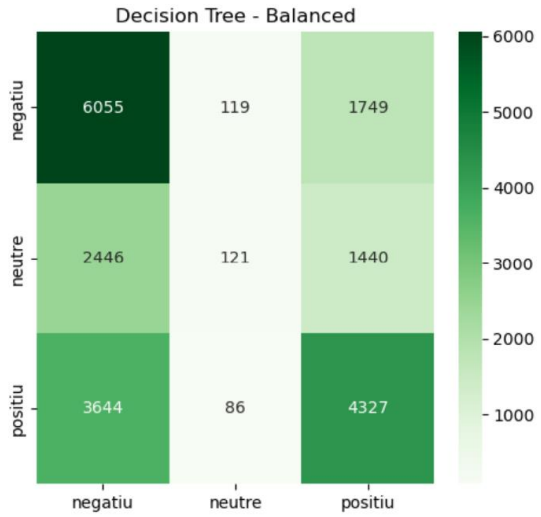




Com podem veure en les classes neutres les classificacions son molt pobres, demostrant clarament que és la classe que més falla. En aquest cas el millor valor de l'hiperparàmetre alfa és 0,1.

Model: Decision Tree

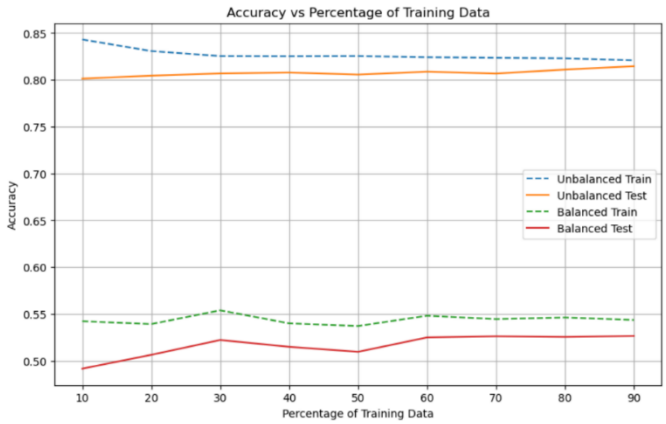


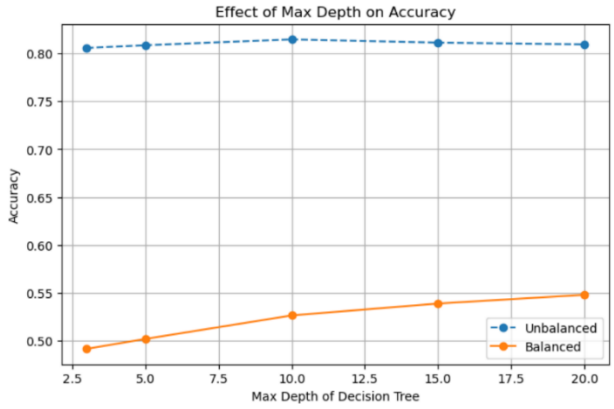


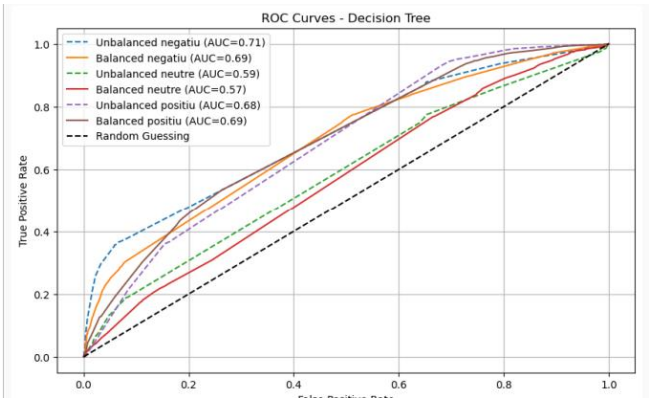
	precision	recall	f1-score	support
negatiu	0.62	0.21	0.31	2375
neutre	0.29	0.02	0.03	1690
positiu	0.82	0.98	0.90	15923
accuracy			0.81	19988
macro avg	0.58	0.40	0.41	19988
weighted avg	0.75	0.81	0.75	19988

Decision Tree - Balanced Dataset

	precision	recall	f1-score	support
negatiu	0.50	0.76	0.60	7923
neutre	0.37	0.03	0.06	4007
positiu	0.58	0.54	0.56	8057
accuracy			0.53	19987
macro avg	0.48	0.44	0.41	19987
weighted avg	0.50	0.53	0.47	19987

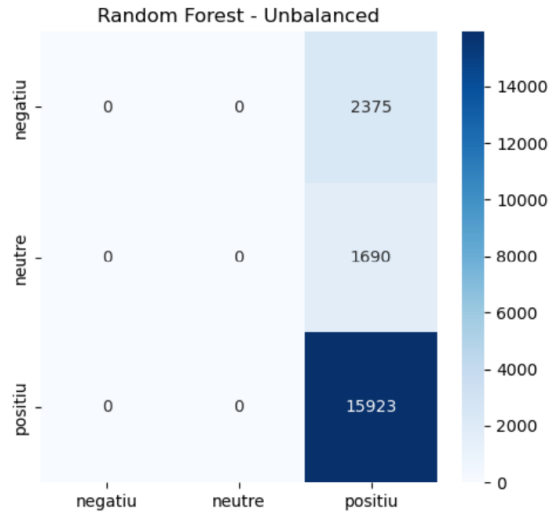


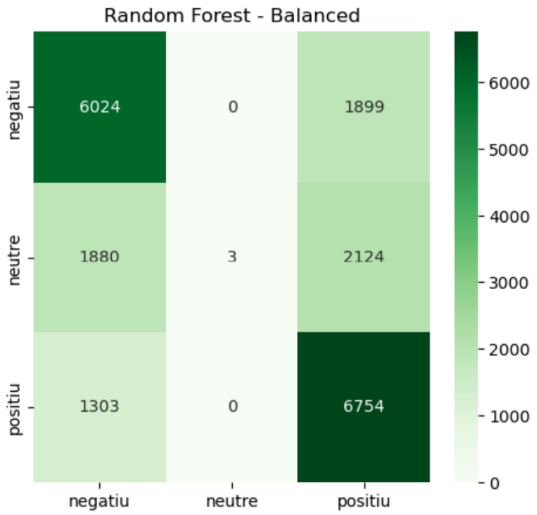




Com es pot veure, aquest és el model que pitjor ens funciona, el rendiment no és gens bo. Podem veure com l'hiperparàmetre max depth a mesura que va augmentant, millora la accuracy. Tot i així en la corba ROC és la que més s'apropa a la aleatorietat, demostrant que el model no ens ha servit.

Model: Random Forest

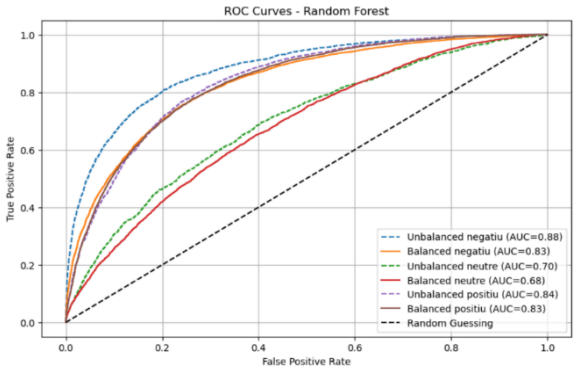


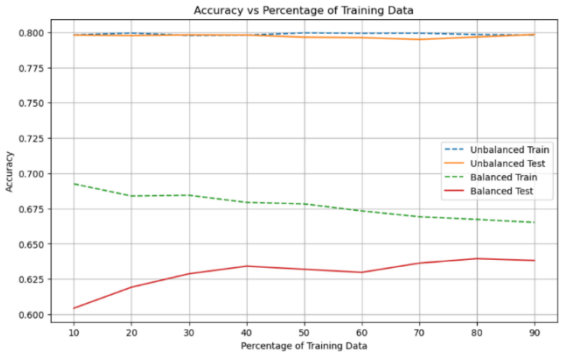


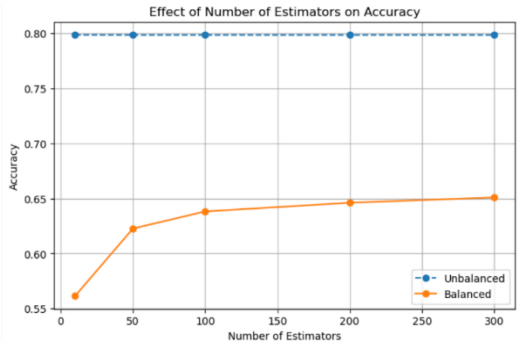
	precision	recall	f1-score	support
negatiu	0.00	0.00	0.00	2375
neutre	0.00	0.00	0.00	1690
positiu	0.80	1.00	0.89	15923
accuracy			0.80	19988
macro avg	0.27	0.33	0.30	19988
weighted avg	0.63	0.80	0.71	19988

Random Forest - Balanced Dataset

	precision	recall	f1-score	support
negatiu	0.65	0.76	0.70	7923
neutre	1.00	0.00	0.00	4007
positiu	0.63	0.84	0.72	8057
accuracy			0.64	19987
macro avg	0.76	0.53	0.47	19987
weighted avg	0.71	0.64	0.57	19987

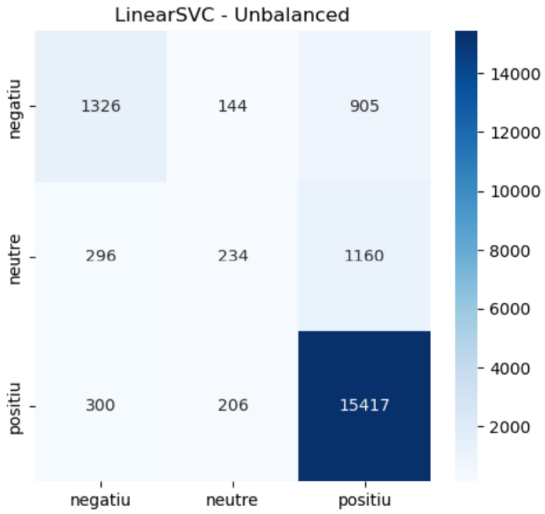


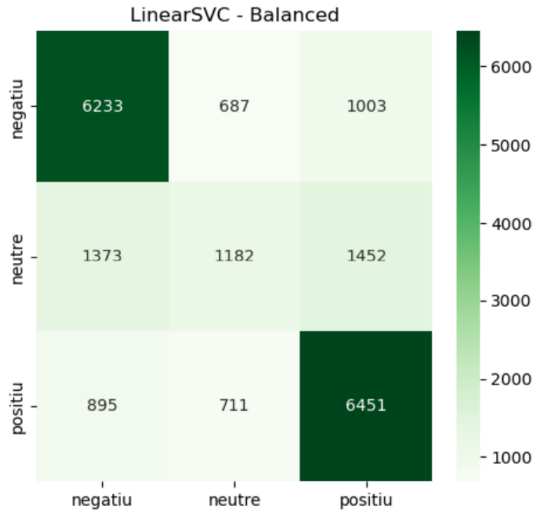




En aquest cas ens torna a passar el mateix amb la classe neutral, mentre que en positiu i negatiu funciona força bé, la classe neutre no funciona com ens agradaria. L'hiperparàmetre del nombre d'estimadors suposa lleugera millora fins a 200 estimadors i a més ja no surt a compte pel temps que requereix l'execució i la millora que suposa.

Model: Linear SVC

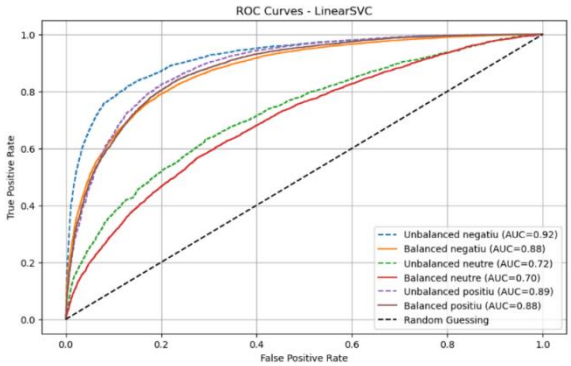


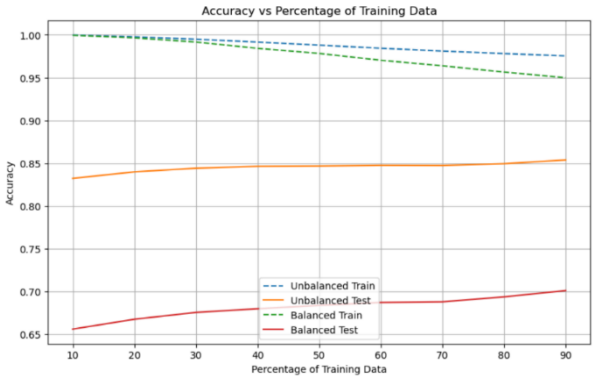


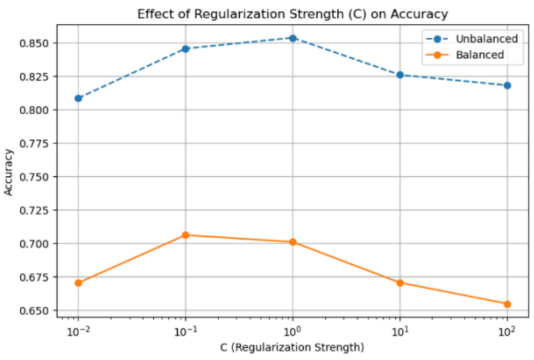
	precision	recall	f1-score	support
negatiu	0.69	0.56	0.62	2375
neutre	0.40	0.14	0.21	1690
positiu	0.88	0.97	0.92	15923
accuracy			0.85	19988
macro avg	0.66	0.55	0.58	19988
weighted avg	0.82	0.85	0.83	19988

LinearSVC - Balanced Dataset

	precision	recall	f1-score	support
negatiu	0.73	0.79	0.76	7923
neutre	0.46	0.29	0.36	4007
positiu	0.72	0.80	0.76	8057
accuracy			0.69	19987
macro avg	0.64	0.63	0.63	19987
weighted avg	0.67	0.69	0.68	19987







Aquest model dona resultats similars al primer model, els quals són prou acceptables. En aquest cas l'hiperaparàmetre C 0.1 per el data set balancejat i 1 pel data set no balancejat.

Canvis respecte a la Presentació

En primer lloc, vam afegir mètriques de rendiment segons la mida del vector (max_features) i una comparació entre la vectorització tf-idf i BoW.

Per altra banda, a l'hora de fer l'informe vam canviar la quantitat de dades o instàncies que agafàvem, passant a agafar ara 100.000 entrades.

Hem afegit per cada model diferents proves (curves roc, percentatge de train, etc) així com canvis en els hiperparàmetres per comparar el rendiment.

A més hem inclòs el model SVC a l'informe, és a dir hem provat tot el que vam fer amb els altres models també amb aquest.

Conclusions

Com a part principal de les nostres conclusions començarem dient que el major problema que hem tingut ha estat el nostre data set. Per una banda al ser tan gran el temps d'execució pujava molt. Per altra banda teníem un data set molt desbalancejat, ja que la gran majoria de ressenyes eren positives i això feia que els models tinguessin un biaix cap a la classe més representada.

Després de compara els diversos models podem arribar a la conclusió que el que millors resultats ens ha donat ha estat el Logistic Regression, un model simple però eficaç. Hem vist que aquest model té un equilibri entre precisió i la seva capacitat de generalització. Mostra una bona classificació entre les classes positives i negatives, mentre que amb la neutre és la que més pateix, tenint molts errors.

Hem vist la importància d'un data set balancejat perquè el model no presenti un biaix cap a cap classe en específic. Quan totes les categories estan representades de manera proporcional, el model pot aprendre patrons sense donar preferència a les classes amb més mostres, donant un rendiment més fiable, ja que en proporció cada classe està més representada que en un data set no balancejat.

També des d'un principi havíem hagut d'aplicar un millor preprocessament amb la lematització (que al inici no ho vam tenir en compte), per així reduir la complexitat del text i eliminar variacions de paraules innecessàries. A més vam estar fent proves per veure quina era la millor vectorització per nosaltres, canviant la mida del vector per comprovar el rendiment amb tf-idf en comparació a BoW.

Com a conclusió general podem dir que amb aquest projecte hem vist que una combinació de dades balancejades, un model adequat i bon preprocessament de les dades permeten fer un model de sentimental analisys (en el nostre cas classificació) raonable, sense obviar les limitacions que hi ha presents.