UPCAlumni

Classificació de Veus Reals i Artificials amb Models Predictius

Daniel Álvarez Sarroca

IAA - GIA

ÍNDEX

1 INTRODUCCIÓ

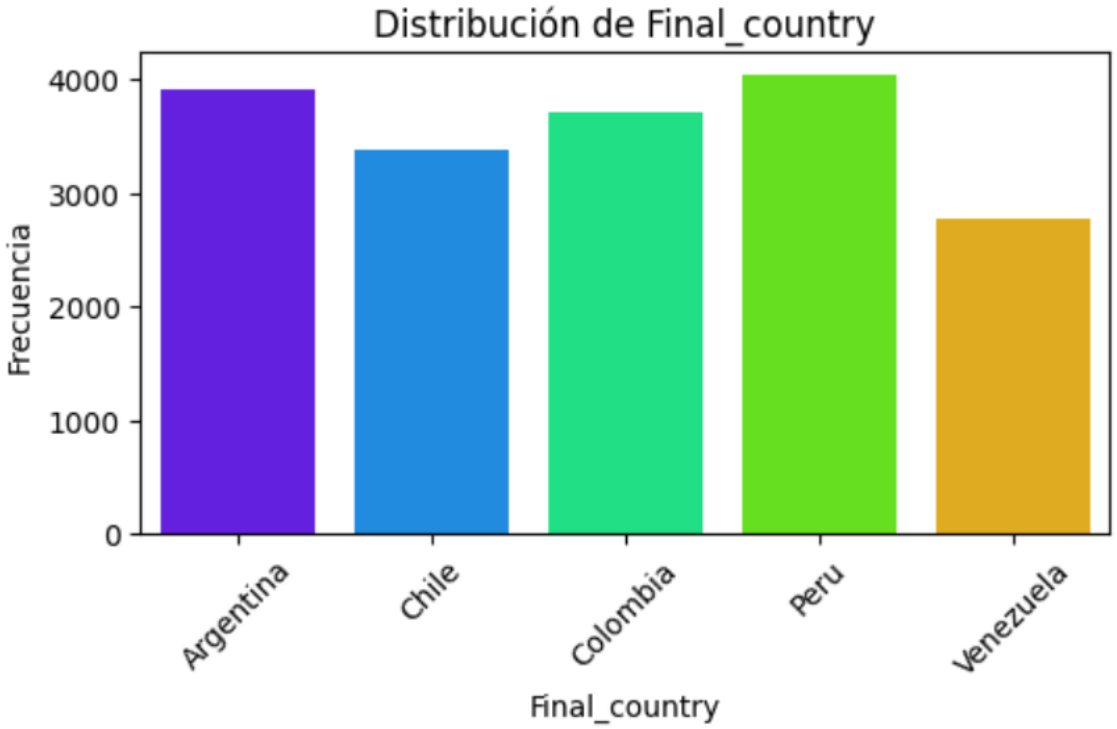
Aquest informe té com a missió documentar quin ha estat el procés que s'ha implementat durant aquesta pràctica, la qual té com a objectiu principal desenvolupar un model capaç de classificar si una veu és real o ha estat generada per intel·ligència artificial, utilitzant tècniques d'aprenentatge supervisat. Per a poder assolir-ho, s'han implementat tres diferents models, un KNN, un d'arbres de decisió, i un SVM, però abans d'entrenar-lo, s'ha de crear un conjunt de dades el qual haurà de passar per un procés de preprocessat i eliminació de variables.

2 ANÀLISI I PREPROCESSAT DE LES DADES:

2.1 Anàlisi estadístic de variables:

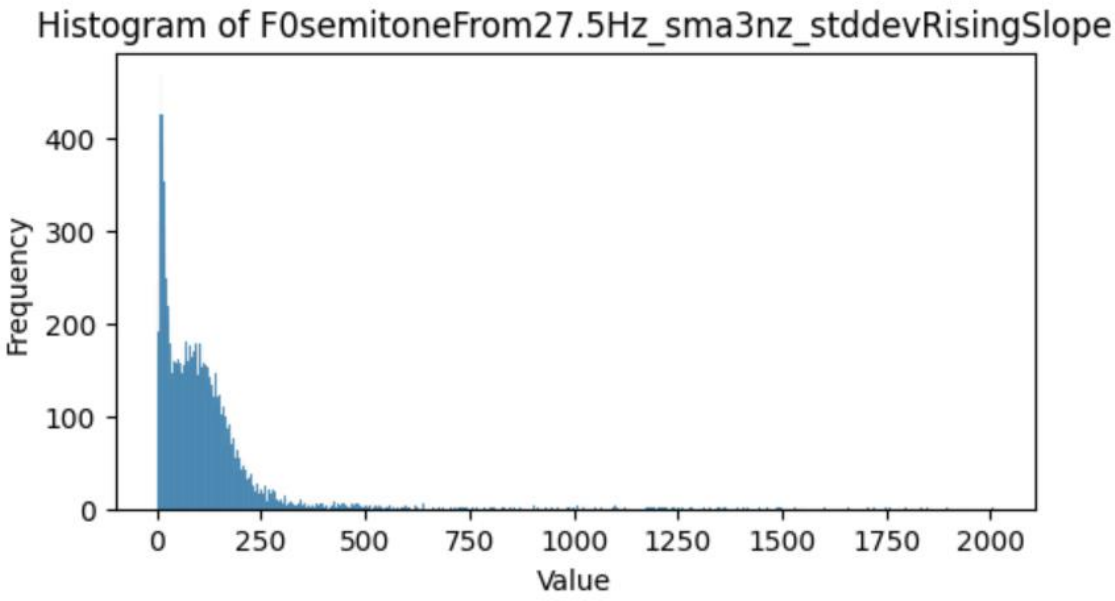
Per a començar amb la secció, es demana fer un anàlisi sobre les diferents variables donades en els arxius amb les dades. En concret, es calcularà estadístiques descriptives bàsiques com la mitjana, la variància, el mínim, el màxim i altres mesures d'interès per cada variable. A més, es comentarà la seva distribució generada per un gràfic.

El notebook 1 conté una taula resum amb aquestes estadístiques per a totes les variables i el codi utilitzat per generar-la. Tot i això, però, es posarà el focus en comentar detalladament un parell de variables (una categòrica i una numèrica) que es considerin especialment rellevants o interessants pel context de les dades.



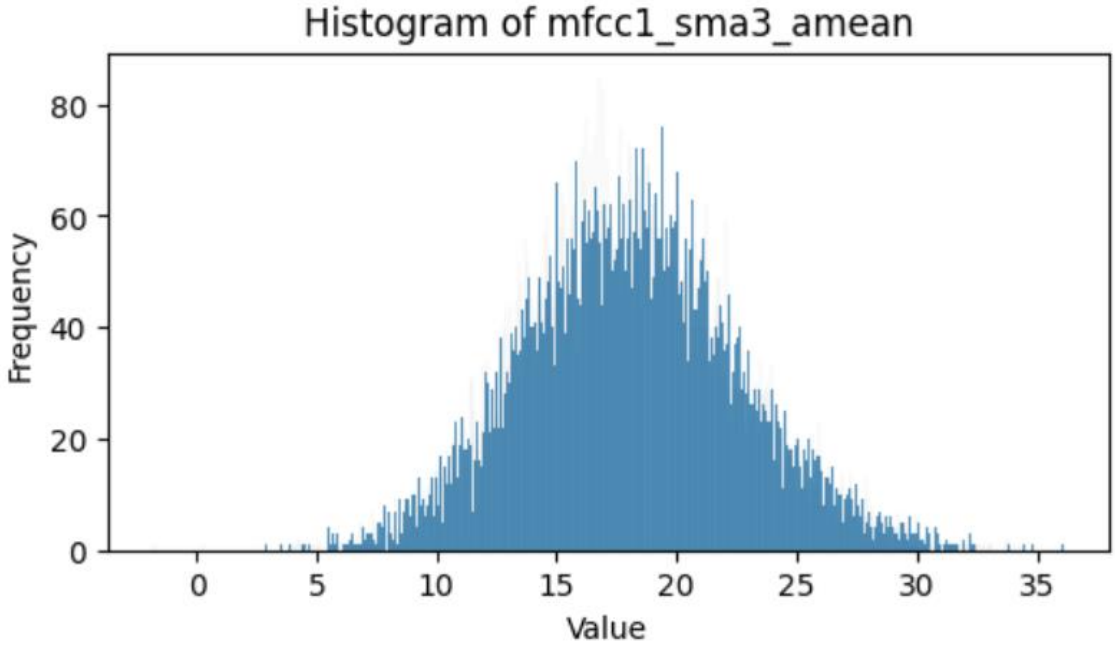
Aquesta variable és una fusió de les variables Target_country i Country, aquesta fusió s'explica detalladament en la següent secció, i ens mostra una representació diversa dels països de cada individu del conjunt donat, indicant freqüència de cada categoria. Argentina i Perú són els països amb un nombre més alt de persones en les dades, tocant quasi els 4.000 o sobrepassant-los, mentre que Xile i Colòmbia tenen una representació lleugerament inferior, però encara significativa. Veneçuela, en canvi, presenta la menor freqüència, amb menys de 3.000 registres.

Aquesta distribució indica una lleugera desigualtat en la representació dels països, que podria influir en els resultats dels models si no es considera adequadament. No obstant això, la presència de tots els països amb proporcions relativament properes permet mantenir la diversitat geogràfica en l'anàlisi.



La distribució de la variable *F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope* és fortament asimètrica, amb una alta concentració de valors baixos que decreix ràpidament a mesura que els valors augmenten. Això ens indica que la majoria de les observacions es troben en un rang reduït, mentre que hi ha una cua llarga cap a valors més alts, indicativa de possibles outliers o valors extrems.

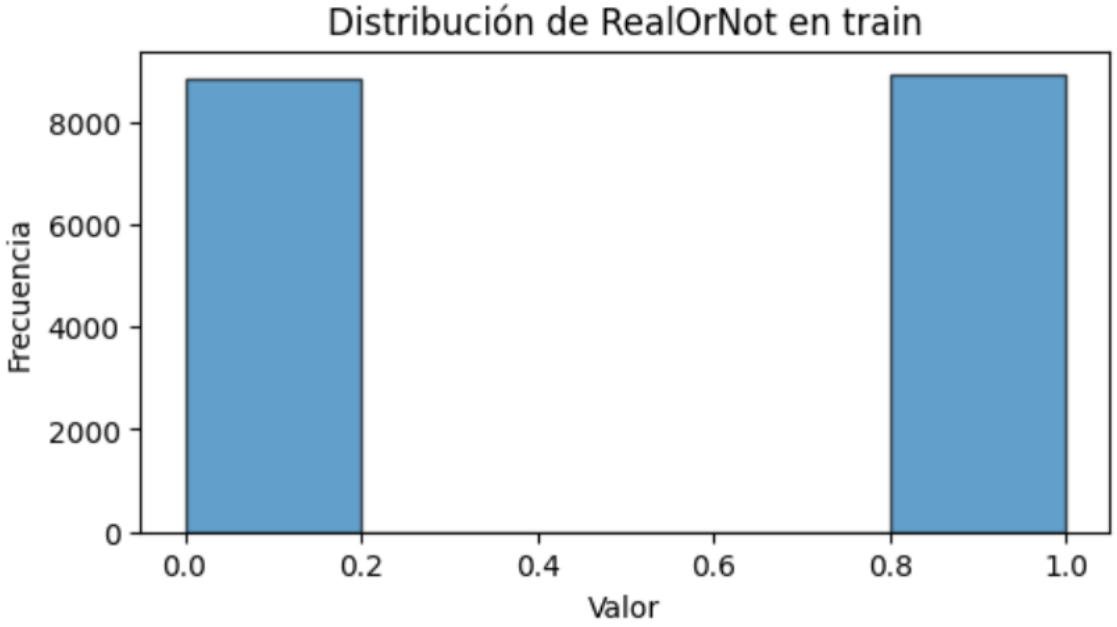
No obstant això, la interpretació exacta del significat d'aquesta variable és limitada, ja que no es disposa d'una metadada específica que expliqui el seu contingut o rellevància. Això complica determinar si aquesta distribució és esperada en el context de l'anàlisi o si requeriria un tractament addicional. Igualment, s'aplicarà alguna tècnica de normalització per a les variables que no segueixin una distribució gaussiana, ja sigui mitjançant transformacions o altres.

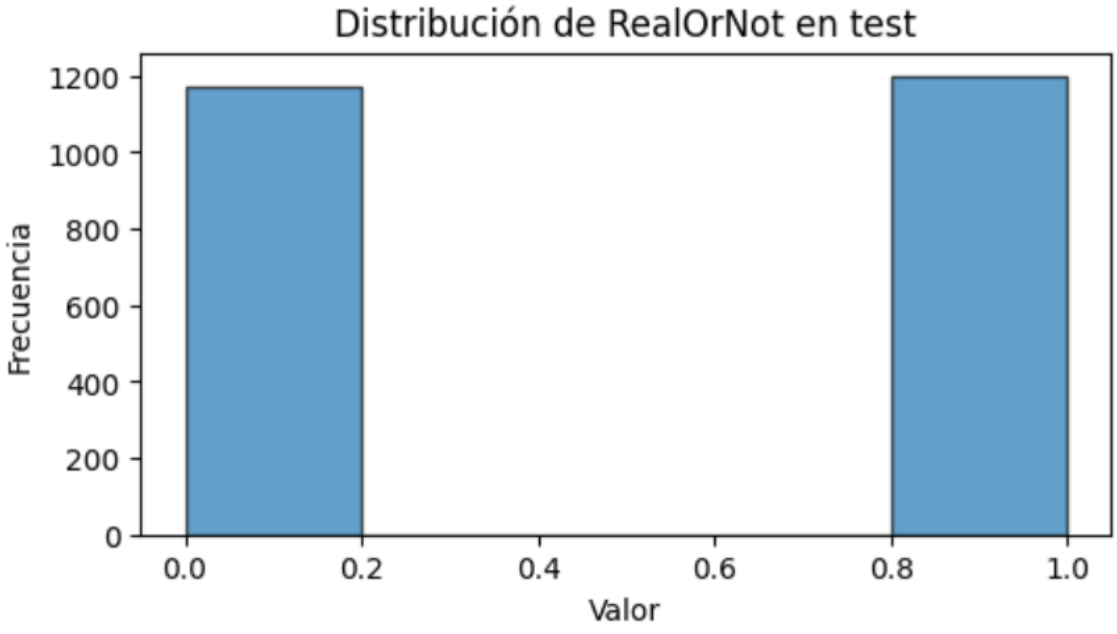


I aquí un exemple de l'altra cara de la moneda, la variable *mfcc1_sma3_amean* presenta clarament una distribució gaussiana, amb una major concentració de valors al voltant de la mitjana, situada aproximadament entre 15 i 20. Per a variables que segueixin la forma de la campana no caldrà normalitzar-les parlant de la seva distribució

2.2 Estudi de balanceig de la variable objectiu:

En aquest punt, l'objectiu és analitzar el balanceig de les classes en els conjunts de dades de train i test proporcionats. Per a això, es mostrarà la distribució que té aquesta variable en un histograma que permetrà visualitzar les freqüències de cada classe en ambdós conjunts i identificar possibles desequilibris que podrien afectar la capacitat predictiva del model.





Com es pot veure en els dos histogrames la variable RealOrNot és mostra equilibrada tant en el conjunt de train com en el de test. Aquest balanceig pràcticament perfecte elimina la necessitat d'aplicar tècniques balanceig de dades, i assegura que el model entrenat tindrà les mateixes oportunitats d'aprendre patrons per a ambdues classes. A més, per a veure quants valors exactes hi havia en cada conjunt de dades, es va implementar el mètode **value.counts()** que em retornava el següent:

```
Realornot Train
1    8939
0    8865

Realornot Test
1    1200
0    1172
```

Això bàsicament reforça el fet que no cal implementar cap mètode de balanceig de dades.

2.3 Partició Train/Val i preprocessament amb imputacions

Abans de procedir amb la partició en conjunts de test i validació, va ser necessari crear un conjunt de dades final que s'utilitzarà per a totes les etapes del projecte, incloent el preprocessament, la imputació de dades, l'entrenament de models i altres anàlisis. Per a construir aquest dataframe final, es va combinar els conjunts de dades *train, smile* i *full* mitjançant la variable *UniqueID*, que permet vincular cada individu amb les seves característiques i variables corresponents.

Després d'aquesta fusió, es va decidir conservar només les variables categòriques relacionades amb el sexe i el país, concretament: target_sex, target_country, sex i country, juntament amb totes les variables numèriques. La raó d'aquesta selecció és que, per a la predicció i la construcció de models, considero que aquestes variables categòriques són les més rellevants, mentre que la majoria de les altres eren informació poc útil, com ara rutes d'arxius i identificadors que no aporten valor analític.

D'altra banda, encara que no s'ha arribat a la part de recodificació de variables, es va combinar les columnes categòriques mencionades en dues úniques variables la funció *combine_first*. Aquesta funció va permetre completar els valors nuls d'una columna utilitzant els valors corresponents d'una altra columna, consolidant tota la informació disponible en dues noves columnes: Final_sex i Final_country. Així es podia reduir els valors nuls al mínim possible.

Així doncs, abans de començar a imputar, es va realitzar la partició del conjunt final en un 80% per a entrenament i un 20% per a la validació:

	Partition	Size	Real Samples (%)	Synthetic Samples (%)
0	Train	14243	50.207119	49.792881
1	Validation	3561	50.210615	49.789385

2.4 Tractament de missings:

Un cop realitzada la partició, es va procedir a preparar els conjunts d'entrenament i validació mitjançant les imputacions necessàries. Per començar amb la gestió dels valors nuls, es va utilitzar el mètode .isnull().sum(), que retorna la quantitat total de valors NaN presents al conjunt de dades. En aquest cas, no es van detectar valors nuls. Tot i així, es va analitzar possibles valors zero en les variables numèriques, ja que aquests també podrien ser indicatius de missings. Per fer-ho, es va implementar una funció específica per comptar la quantitat de zeros en aquestes variables, amb l'objectiu de detectar i gestionar qualsevol present en les dades. Aquesta va retornar la següent taula per a train i test, respectivament:

	Column	Zero_Count
0	F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope	1
1	F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope	370
2	F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope	119
3	jitterLocal_sma3nz_amean	0
4	loudness_sma3_amean	0
5	mfcc1_sma3_amean	0
6	mfcc1_sma3_stddevNorm	0
7	mfcc2_sma3_amean	0
8	mfcc2_sma3_stddevNorm	0
9	mfcc3_sma3_amean	0
10	mfcc3_sma3_stddevNorm	0
11	slopeUV500-1500_sma3nz_amean	0
12	spectralFlux_sma3_stddevNorm	0

	Column	Zero_Count
0	F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope	0
1	F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope	85
2	F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope	29
3	jitterLocal_sma3nz_amean	0
4	loudness_sma3_amean	0
5	mfcc1_sma3_amean	0
6	mfcc1_sma3_stddevNorm	0
7	mfcc2_sma3_amean	0
8	mfcc2_sma3_stddevNorm	0
9	mfcc3_sma3_amean	0
10	mfcc3_sma3_stddevNorm	0
11	slopeUV500-1500_sma3nz_amean	0
12	spectralFlux_sma3_stddevNorm	0

Els resultats mostren clarament que algunes variables numèriques presenten un nombre significatiu de valors zero, especialment les tres primeres variables en el conjunt d'entrenament i la segona i tercera en el conjunt de validació. Per gestionar aquests casos, es va optar per imputar els valors nuls utilitzant la mediana, ja que aquesta és una mesura més robusta davant la presència d'outliers i garanteix més fiabilitat. Un cop completada aquesta imputació, es va verificar que ja no quedaven valors zero, fet que va permetre continuar amb les següents etapes del preprocessament sense inconvenients.

2.5 Tractament d'outliers:

En aquesta secció s'analitza la presència de valors atípics (outliers) en els conjunts de dades d'entrenament i validació, ja que aquests poden distorsionar les anàlisis estadístiques i afectar el rendiment dels models. Per aquest motiu, s'ha procedit a identificar-los mitjançant tècniques adequades i, si escau, a gestionar-los de manera efectiva.

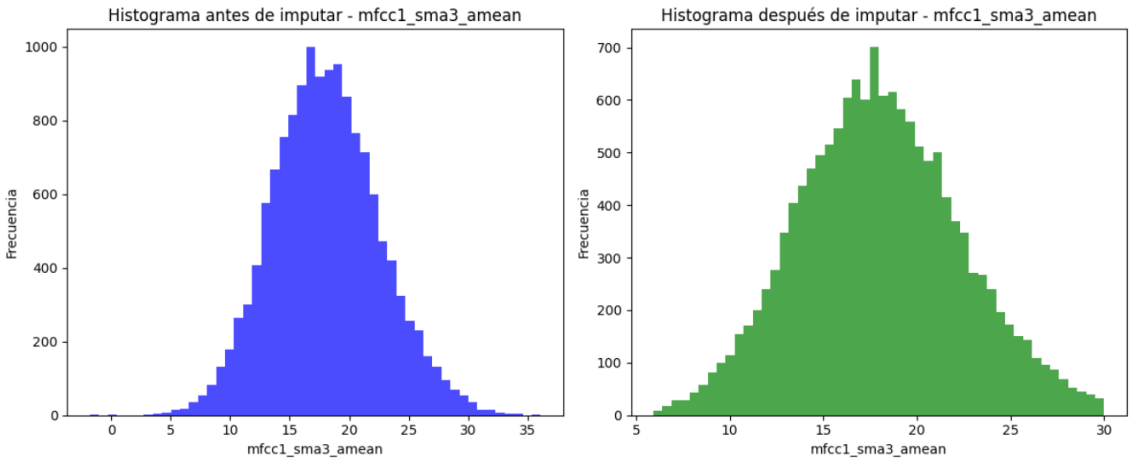
El procés utilitzat per a la identificació de valors atípics es basa en el càlcul de la distància interquartílica (IQR) per a cada variable numèrica, establint els límits màxims i mínims acceptables. Aquesta metodologia permet identificar els valors considerats atípics segons el rang interquartílic. Per exemple, per a les dues primeres variables numèriques:

```
Procesando columna: F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope
  - Q1 (25%): 29.0911435
  - Q3 (75%): 141.58102
  - IQR: 112.4898765
  - Valores fuera de rango: < -139.64367125 o > 310.31583475
```

```
  - Outliers por debajo: 0
  - Outliers por encima: 597

Procesando columna: F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope
  - Q1 (25%): 17.243138000000002
  - Q3 (75%): 35.278014999999996
  - IQR: 18.034876999999994
  - Valores fuera de rango: < -9.80917749999999 o > 62.33033049999999
  - Outliers por debajo: 316
  - Outliers por encima: 1551
```

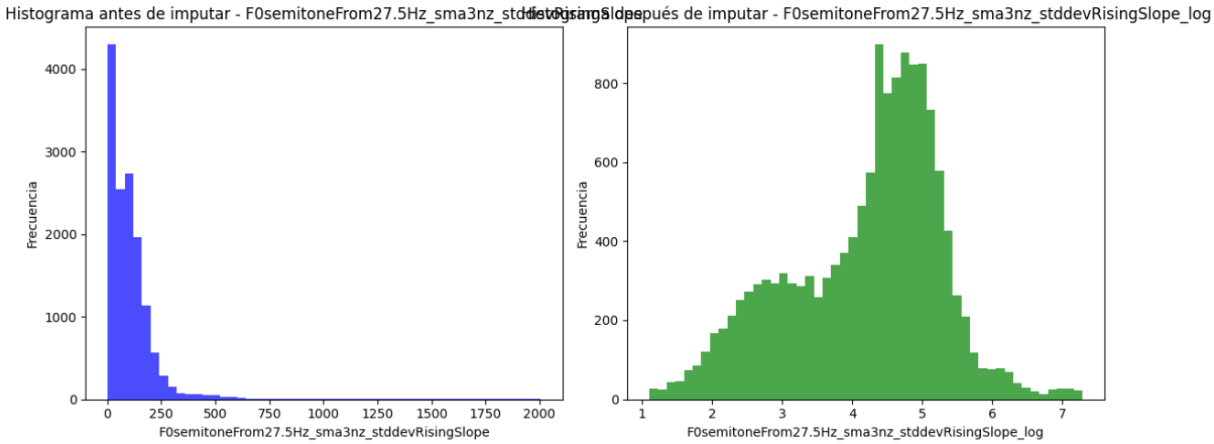
Així doncs, es van imputar aquests valors atípics utilitzant la mediana. Es va decidir això degut a la gran robustesa que té la mediana davant la presència d'outliers, ja que, a diferència de la mitjana, no es veu afectada per valors extrems, això ja s'ha comentat anteriorment per a l'apartat de la imputació de valors nuls. D'aquesta manera, es manté la distribució original de les dades sense distorsionar-les.

A més, imputar amb la mediana evita eliminar observacions del conjunt de dades, mantenint així la mida inicial del conjunt, que és essencial per a preservar la quantitat d'informació disponible per als models predictius. Aquest enfocament també assegura que els valors imputats es mantinguin dins d'un rang esperable i coherent amb la distribució de la variable, contribuint a una millor.



Cal destacar que, per a les variables que no seguien una distribució gaussiana, es va aplicar una transformació logarítmica abans de calcular els intervals interquartílics (IQR). L'objectiu d'això era normalitzar la distribució de les dades, fet que és més precís trobar els rangs màxims dels IQR per a distribucions uniformes.

Aquesta correcció prèvia no només millora l'efectivitat del procés d'identificació d'outliers, sinó que també evita l'eliminació o imputació incorrecta de valors que podrien ser rellevants en el context del model. Per tant, aplicar una transformació logarítmica assegura que el tractament dels valors atípics sigui més alineat amb la distribució real de les dades. Aquí un exemple de imputació usant la transformació logarítmica prèviament:



Com es pot observar al primer histograma, la distribució inicial no segueix una distribució gaussiana. Aquesta asimetria, amb una acumulació significativa de valors baixos i una cua llarga cap a valors alts, dificulta tant la identificació com la gestió precisa dels outliers.

Després de la transformació logarítmica i la imputació dels outliers, el segon histograma mostra una distribució més equilibrada i propera a la gaussiana. Aquest procés va permetre imputar-los de forma efectiva, utilitzant la mediana. Això assegura que la variable pugui ser utilitzada de manera òptima en les següents fases de modelització.

2.6 Recodificació de variables:

En aquesta part s'analitza la necessitat de recodificar algunes variables del conjunt de dades per tal de millorar la seva interpretabilitat, simplificar el model o adaptar-les al format requerit pels algoritmes utilitzats. Qualsevol recodificació es justifica en funció de la seva contribució a l'eficiència del procés de modelització i al potencial impacte en els resultats. Tal com s'ha mencionat anteriorment, una part de la recodificació ja realitzada ha consistit en la combinació de les variables target_sex amb sex i target_country amb country. Aquesta unió es va dur a terme per consolidar la informació de cada individu en una única variable.

D'altra banda, es va aplicar one-hot encoding a aquestes variables per transformar-les en un format numèric adequat per als models predictius. Gràcies a això, es van poder interpretar com a valors binaris, assignant una columna independent per a cada categoria. Aquest procés garanteix que els algorismes que s'utilitzen més endavant en la pràctica, puguin interpretar correctament la informació categòrica sense introduir biaixos.

Final_sex_Female	Final_sex_Male	Final_country_Argentina	Final_country_Chile	Final_country_Colombia	Final_country_Peru
0.0	1.0	0.0	1.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0

3 PREPARACIÓ DE VARIABLES

3.1 Normalització de variables:

En aquesta secció es realitza la normalització de les variables numèriques, aquest procés és especialment important en models que són sensibles a l'escala de les dades, com ara els basats en distàncies o els que utilitzen coeficients.

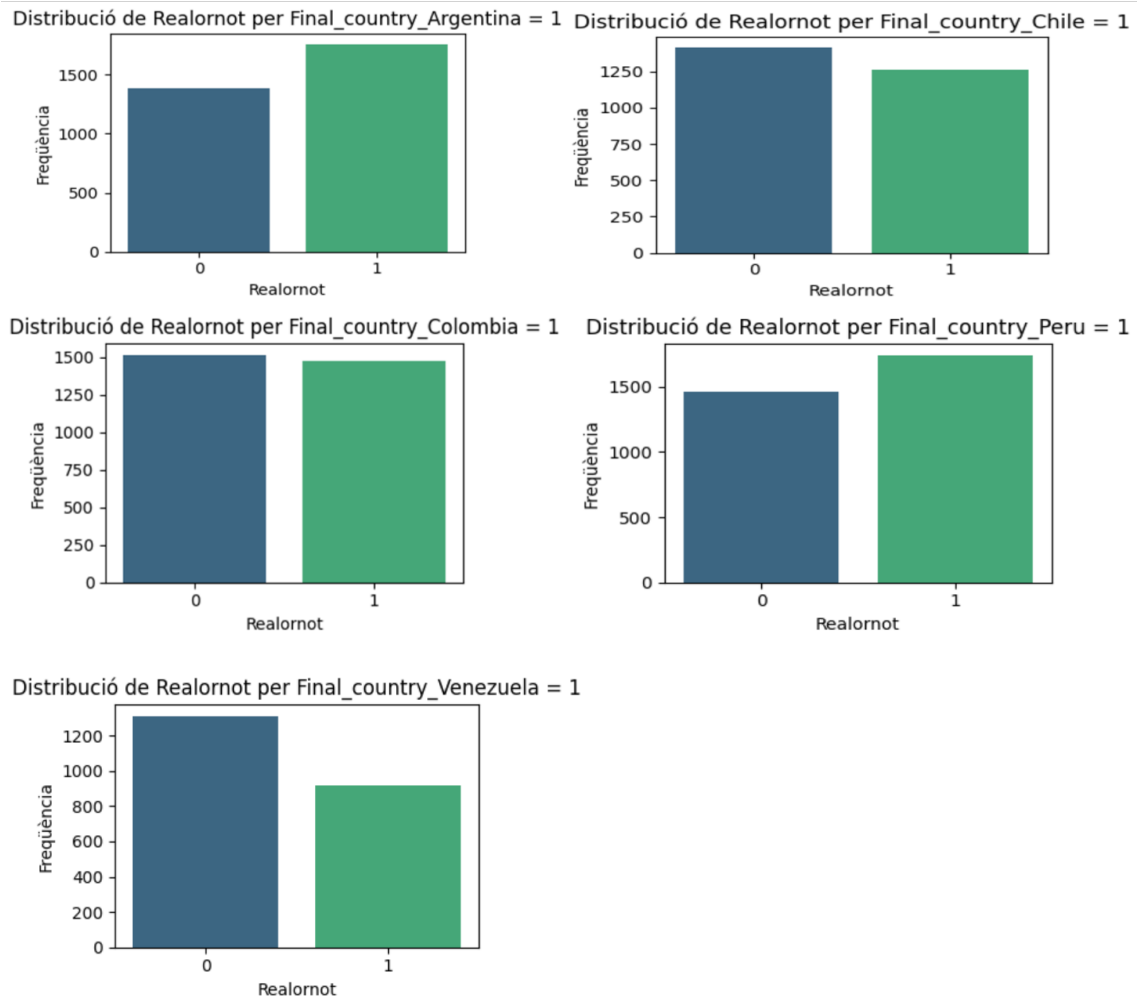
Abans de començar, aquesta normalització s'ha dut a terme utilitzant l'estandarització z-score de sklearn. Primerament, es van identificar totes les variables numèriques, excloent tant la variable objectiu, com les variables categòriques transformades prèviament amb one-hot encoding. Es va escollir aquest mètode degut a que aquest centra les dades al voltant de la mitjana (0) i les escala perquè tinguin una desviació estàndard d'1. A més, aquest té una millor elecció quan es treballa amb dades amb distribucions no uniformes. Així doncs, es va ajustar el StandardScaler() sobre les dades d'entrenament, i seguidament aquestes mateixes transformacions es van aplicar a les dades de validació. Aquest procés garanteix que les variables estiguin en la mateixa escala.

A continuació es mostra com queden les variables escalades de les dues primeres variables numèriques, això s'ha aplicat per a totes:

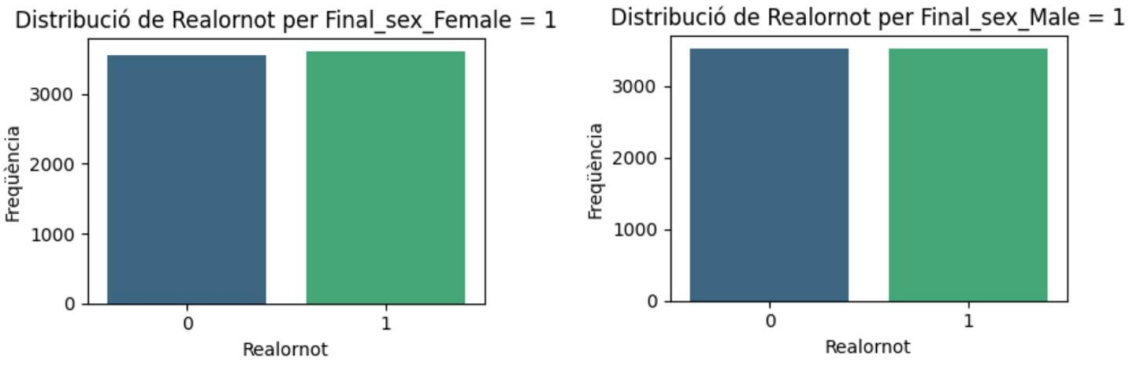
	F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope	F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope
0	0.107470	0.516185
1	-0.014251	-0.281358
2	-0.249233	-0.337932
3	-0.014251	-0.281358
4	0.234453	-0.751499

3.2 Anàlisi de variables categòriques i variable objectiu:

En aquesta part s'analitzen les variables categòriques i la variable objectiu. Les variables categòriques han estat transformades prèviament mitjançant one-hot encoding, com s'ha comentat anteriorment, pel que s'ha una columna independent per a cada categoria. Per tant, en aquest anàlisi ens centrarem únicament en els valors iguals a 1 per a cada variable, que representen la presència de la categoria corresponent, ja que no interessa analitzar pels valors iguals a 0.



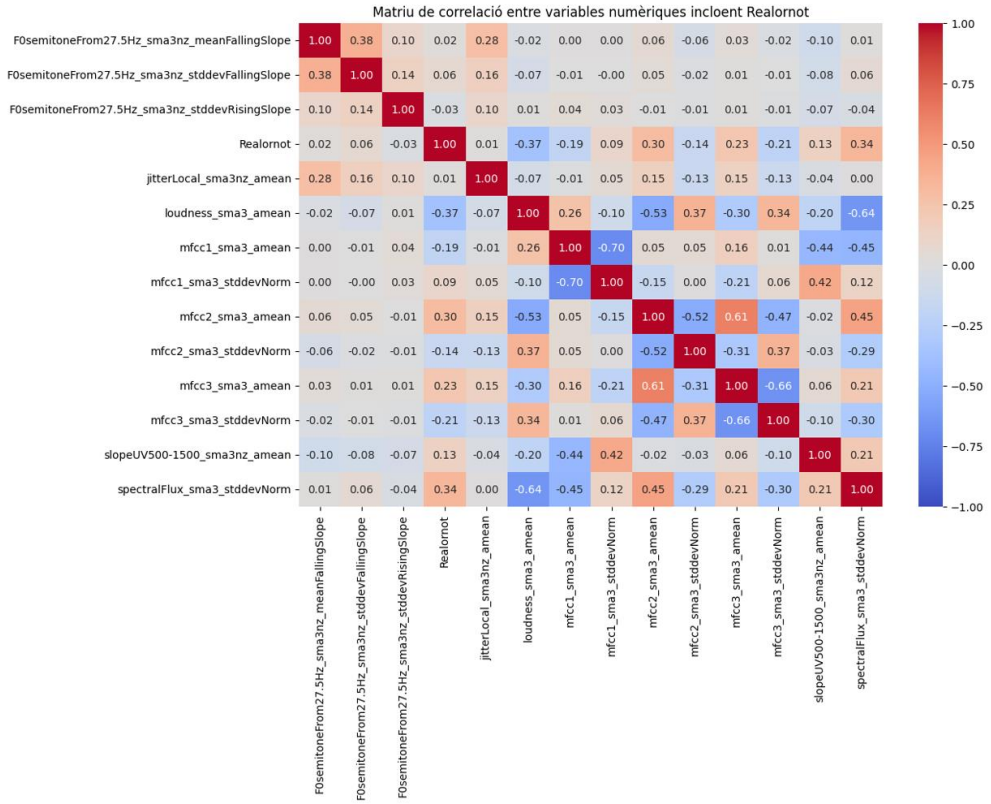
En general, es pot observar que la distribució entre les classes no real i real és equilibrada en la majoria dels països. Tanmateix, a Veneçuela s'observa una lleugera diferència, amb una major però no excessiva proporció de valors 0 en comparació amb els valors 1. Això podria indicar una variació en el comportament de la variable objectiu per a aquest país.



En quant a les distribucions de les variables del sexe, en ambdós casos les proporcions de les classes 0 i 1 són pràcticament iguals, evidenciant un equilibri notable en la distribució. Aquest comportament indica que la variable objectiu no està condicionada pel sexe, assegurant una representació uniforme i sense desequilibris.

3.3 Eliminació de variables numèriques redundants:

En aquesta part s'identifiquen i eliminen les variables numèriques que són redundants o que aporten soroll al conjunt de dades, basant-se en l'anàlisi de la correlació. Aquest procés té com a objectiu reduir la dimensionalitat i millorar l'eficiència del model. La matriu de correlació generada amb les variables numèriques és la següent:



Com es pot veure, no hi ha cap parell de variables que tinguin una correlació elevada entre elles, fet que ens faria plantejar l'eliminació d'aquestes. Tot i a aquesta afirmació, es va fer un anàlisi de la correlació de cada variable numèrica amb la variable objectiu:

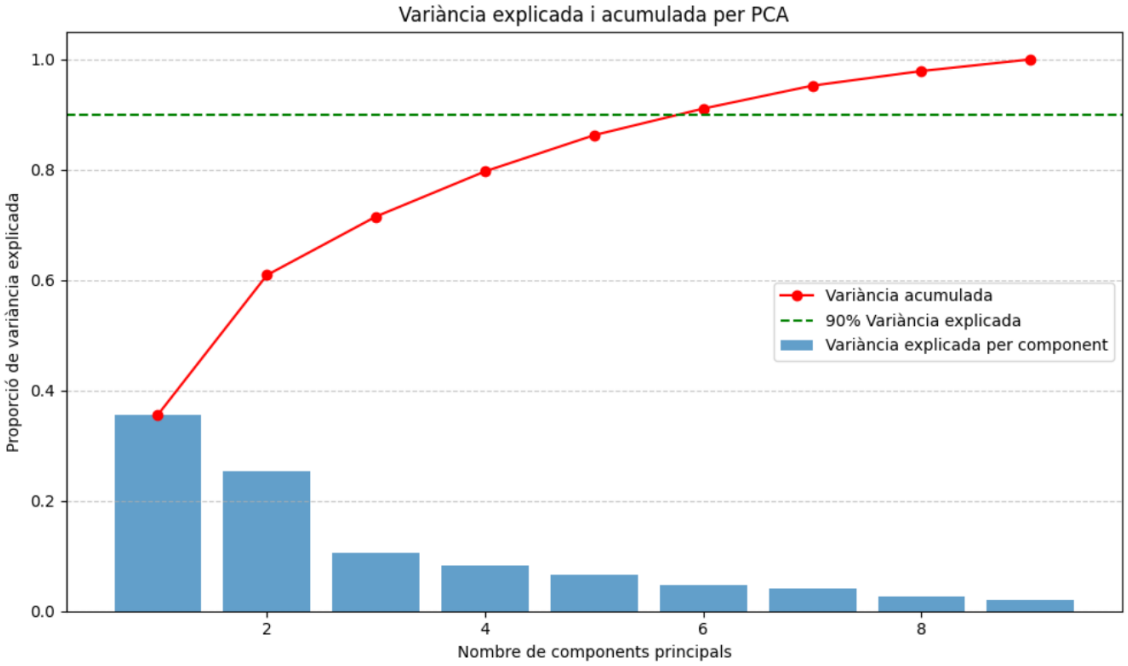
```
Correlació de cada variable amb Realornot:
```

```
Realornot                                        1.000000
spectralFlux_sma3_stddevNorm                     0.340297
mfcc2_sma3_amean                                 0.299721
mfcc3_sma3_amean                                 0.232966
slopeUV500-1500_sma3nz_amean                     0.134656
mfcc1_sma3_stddevNorm                            0.093813
F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope    0.062713
F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope      0.022092
jitterLocal_sma3nz_amean                         0.014829
F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope    -0.031183
mfcc2_sma3_stddevNorm                           -0.143122
mfcc1_sma3_amean                                -0.191197
mfcc3_sma3_stddevNorm                           -0.210893
loudness_sma3_amean                             -0.373365
dtype: float64
```

Amb aquesta sortida de la terminal es pot veure com algunes variables tenen una relació molt baixa amb 'RealOrNot', com ara *F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope, F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope, jitterLocal_sma3nz_amean i F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope*, (marcades amb groc), amb valors de correlació que oscil·len entre 0.06 i -0.03. Aquestes variables es consideren irrellevants per a la tasca predictiva, ja que tenen un impacte nul en la predicció de la variable resposta. Per aquest motiu, es decidirà eliminar-les del conjunt de dades per ja que podrien ocasionar soroll o informació redundant, i aquest fet que podria empitjorar el rendiment del model.

3.4 Estudi de dimensionalitat amb PCA:

En aquest capítol s'analitza la dimensionalitat del conjunt de dades utilitzant l'algorisme Anàlisi de Components Principals per determinar si és necessari reduir el nombre de variables. Aquest estudi es realitza sobre el conjunt de dades ja processat, després d'haver eliminat les variables redundants i sorolloses identificades en l'anàlisi de correlació anterior. Per a fer el PCA s'ha fet sense les variables categòriques, això es degut a que estan transformades usant one-hot encoding, i poden introduir complexitat addicional i s'hauria de tractar-les amb molta cura. La figura generada de la variància acumulada amb el número de dimensions és la següent:




```
Nombre de components necessaris per explicar el 90% de la variància: 6
```

Com es pot observar, es necessita sis components per explicar el 90% de la variància total, la qual cosa demostra que el conjunt de dades ja té una estructura compacta i no presenta un excés de dimensionalitat. No obstant això, donat que el conjunt de dades ja compta amb pocs features, **no** es procedirà a reduir la dimensionalitat. La pèrdua de qualsevol variable podria comprometre la informació rellevant per als models predictius. Per tant, es treballarà amb totes les variables numèriques presents al conjunt.

4 DEFINICIÓ DE MODELS

Abans de començar amb cada model entrenar, s'esmenterà quines seran les mètriques d'avaluació per a determinar si un model ajustà bé o no, per a poder comparar-los entre ells. El principal element que es mirarà serà l'accuracy, ja que permet calcular el percentatge de prediccions correctes sobre el total. Aquesta mètrica serà complementada amb la F1-Score combina la precisió i la sensibilitat, i finalment, la matriu de confusió i corba ROC-AUC proporcionaran una anàlisi detallada dels encerts i errors per cada classe, facilitant la interpretació dels resultats. També s'inclouran mapes de regions per veure si hi ha overfitting o no.

Per a trobar els paràmetres òptims de cada model, s'ha fet mitjançant una funció d'optimització basada en l'espai definit. Aquesta funció prova diverses combinacions i selecciona els paràmetres que maximitzen el rendiment del model en un conjunt de validació. Aquesta metodologia permet identificar les configuracions més eficients i assegurar que el model està ajustat de manera òptima per la tasca objectiu. La taula a continuació resumeix els hiperparàmetres i els seus valors provats durant l'optimització.

4.1 K-Nearest Neighbors

Per tal d'optimitzar el rendiment del model KNN, s'ha definit un espai de cerca d'hiperparàmetres per explorar les combinacions que millor s'adapten a les dades. L'espai inclou els següents hiperparàmetres: el nombre de veïns (n_neighbors), que abarca valors entre 1 i 30 per capturar tant relacions locals com globals; la mètrica de distància, incloent opcions com euclidiana, manhattan i minkowski, per determinar quina és la més adequada per a les dades; i els pesos (weights), provant pesos uniformes o basats en la distància.

Paràmetres provats (alguns exemples ja que n'hi ha molts):

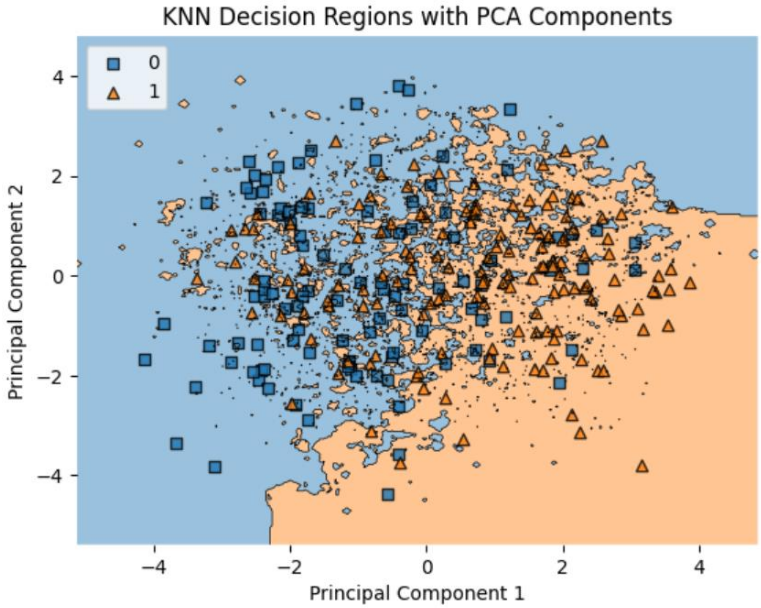
```
Accuracy de 0.840 amb els paràmetres: {'n_neighbors': 10, 'metric':
'manhattan', 'weights': 'distance'}
Accuracy de 0.834 amb els paràmetres: {'n_neighbors': 5, 'metric':
'euclidean', 'weights': 'uniform'}
Accuracy de 0.814 amb els paràmetres: {'n_neighbors': 2, 'metric':
'minkowski', 'weights': 'uniform'}
Accuracy de 0.823 amb els paràmetres: {'n_neighbors': 22, 'metric':
'manhattan', 'weights': 'uniform'}
Accuracy de 0.840 amb els paràmetres: {'n_neighbors': 10, 'metric':
'manhattan', 'weights': 'distance'}
Accuracy de 0.839 amb els paràmetres: {'n_neighbors': 10, 'metric':
'minkowski', 'weights': 'distance'}
100%|██████████| 20/20 [00:19<00:00,  1.04trial/s, best loss: -
0.8404124894995683]
Paràmetres òptims: {'metric': 'manhattan', 'n_neighbors': np.int64(10),
'weights': 'distance'}
```

Per a aquests paràmetres òptims trobats l'informe de classificació era el següent, aquests resultats són fent prediccions amb val:

```
Exactitud del model: 0.84
```

```
Informe de classificació:
              precision    recall  f1-score   support

           0       0.90      0.76      0.82      1773
           1       0.79      0.91      0.85      1788

    accuracy                           0.84      3561
   macro avg       0.85      0.84      0.84      3561
weighted avg       0.85      0.84      0.84      3561
```

I a continuació el diagrama de regions del model:

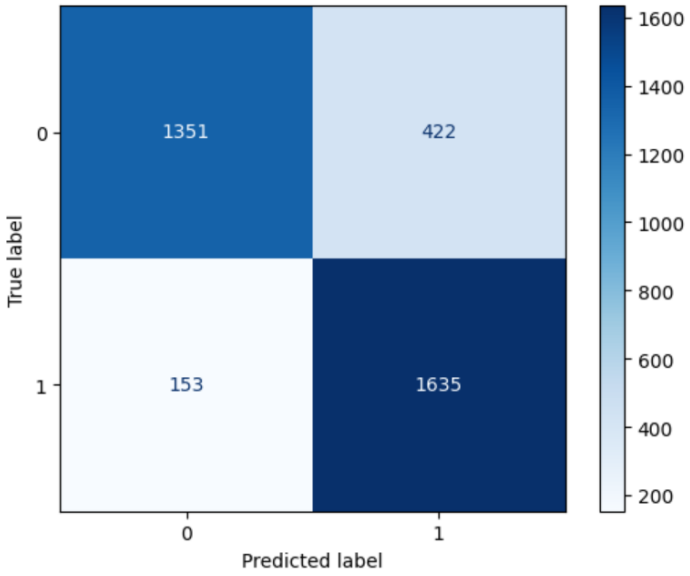


El model KNN entrenat aconsegueix una accuracy de 0.84 al conjunt de validació, amb un bon balanç entre precisió i recall. Malgrat això, el gràfic de les regions de decisió generat amb les components principals del PCA evidencia un comportament que suggereix overfitting. Les fronteres de decisió són molt irregulars i ajustades, fet que indica que el model pot estar memoritzant els patrons del conjunt d'entrenament en lloc de generalitzar-los adequadament. Això es pot comprovar al fer prediccions amb el conjunt de train:

```
Exactitud del model sobre el conjunt d'entrenament: 1.00

Informe de classificació (conjunt d'entrenament):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7092
           1       1.00      1.00      1.00      7151

    accuracy                           1.00     14243
   macro avg       1.00      1.00      1.00     14243
weighted avg       1.00      1.00      1.00     14243
```

Com es pot veure, el model encerta totes, però quan se li passen dades noves que no ha vist abans, les de validació, comet errors, significa que el model s'ha memorizat els patrons específics del conjunt d'entrenament en lloc d'aprendre a generalitzar. Això és un clar indicador de sobre-

ajustament a les dades, ja que el model no és capaç de capturar les característiques generals que permetrien fer prediccions correctes en dades desconegudes, sinó que s'ha ajustat excessivament a les dades d'entrenament, comprometent així el seu rendiment en situacions reals. Finalment, es mostra la matriu de confusió del model pel conjunt de validació:



La matriu de confusió mostra com encerta correctament en la majoria de casos, tanmateix, hi ha una quantitat elevada de falsos positius (422). Tot i tenir una accuracy global alta, aquests errors mostren que el model pot no ser completament fiable degut a que hi ha indicis d'overfitting i una quantitat alta de falsos positius.

4.2 Support Vector Machine

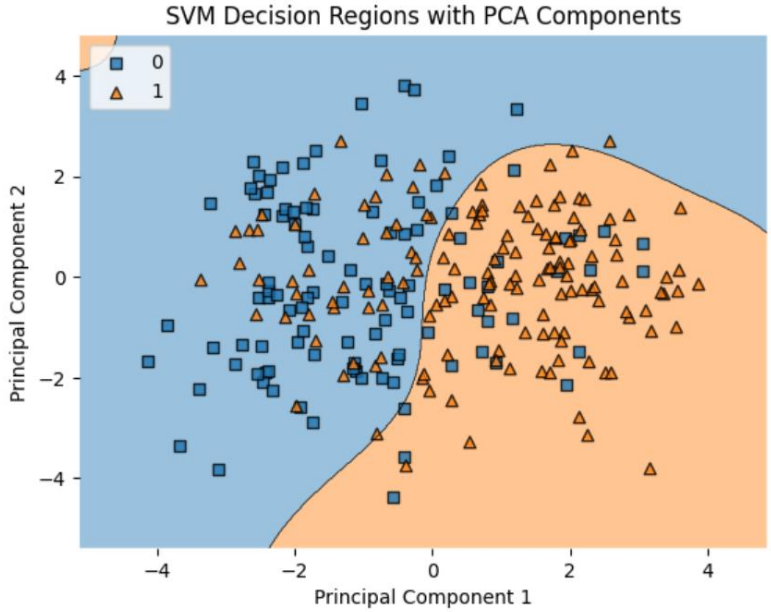
L'espai d'hiperparàmetres inclou el tipus de kernel, amb les opcions rbf i linear. També es considera el paràmetre de regularització (C), amb valors de 0.1, 1 i 10. A més, s'inclou el paràmetre gamma, opcions com scale, 0.1 i 1, per adaptar-se amb precisió a la distribució de les dades.

```
Millors paràmetres: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
Classification Report (Validation):
              precision    recall  f1-score   support

           0       0.91      0.84      0.87      1773
           1       0.85      0.92      0.88      1788

    accuracy                           0.88      3561
   macro avg       0.88      0.88      0.88      3561
weighted avg       0.88      0.88      0.88      3561
```

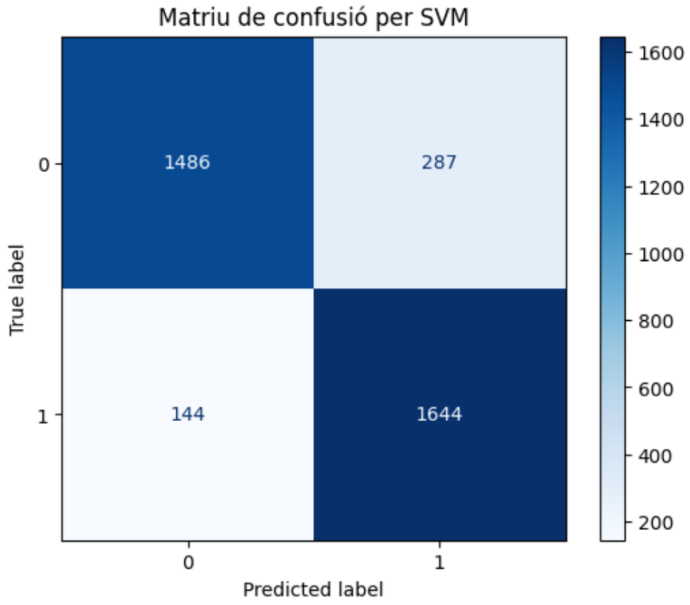
El model SVM amb els millors paràmetres (C: 10, gamma: 0.1, kernel: rbf) aconsegueix una accuracy de 0.88 en el conjunt de validació. L'informe de classificació mostra un bon balanç entre les mètriques de precisió i recall per ambdues classes. Aquest mètode té una accuracy més elevada que KNN per a les prediccions de val.

```
Exactitud del model sobre el conjunt d'entrenament: 0.92

Informe de classificació (conjunt d'entrenament):
              precision    recall  f1-score   support

           0       0.95      0.88      0.91      7092
           1       0.89      0.95      0.92      7151

    accuracy                           0.92     14243
   macro avg       0.92      0.92      0.92     14243
weighted avg       0.92      0.92      0.92     14243

Precisió: 0.916
```

A més, al fer les prediccions per al conjunt d'entrenament, aquest no dona una presició del 100%, cosa que indica que el model comet errors quan veu dades que no ha vist abans, i d'això s'intueix que aquest no s'ha memoritzat cap patró i si que aprèn realment a generalitzar. De moment, està donant indicis de que no hi ha overfitting, i amb el diagrama de regions a continuació es podrà acabar de verificar.



Aquest mostra una frontera de decisió més suau i generalitzada en comparació amb KNN. Aquesta característica és pròpia dels models SVM amb kernel rbf, que permeten una separació no lineal dels punts alhora que intenten maximitzar el marge entre les classes. A continuació la matriu de confusió per al conjunt de validació:



La matriu de confusió del model mostra una millora respecte al model KNN, tant en encerts com en l'equilibri entre les dues classes. Els encerts són molt destacats, amb 1486 mostres correctament classificades com a classe 0 i 1644 com a classe 1, cosa que evidencia una gran capacitat del model per diferenciar amb precisió entre les dues categories. A més la matriu té un nombre baix d'errors. Els falsos positius són baixos respecte als reals, i els falsos negatius són encara més baixos.

Comparant amb el model anterior de KNN, SVM aconsegueix un balanç molt favorable entre precisió i recall per ambdues classes, fent-lo un model més fiable i consistent en prediccions en comparació amb el KNN .Si es compta el bon rendiment del model en el conjunt de validació, juntament amb la l'accuracy obtinguda, i sumant proves de que no hi ha overfitting, sembla ser que SVM està ben ajustat per la tasca objectiu i ofereix un bon equilibri entre flexibilitat i capacitat de generalització en comparació amb KNN.

4.3 Arbre De Decisió

L'espai d'hiperparàmetres definit busca optimitzar la configuració d'un arbre de decisió ajustant la profunditat màxima, el nombre mínim de mostres per dividir un node i el mínim de mostres en un node fulla.

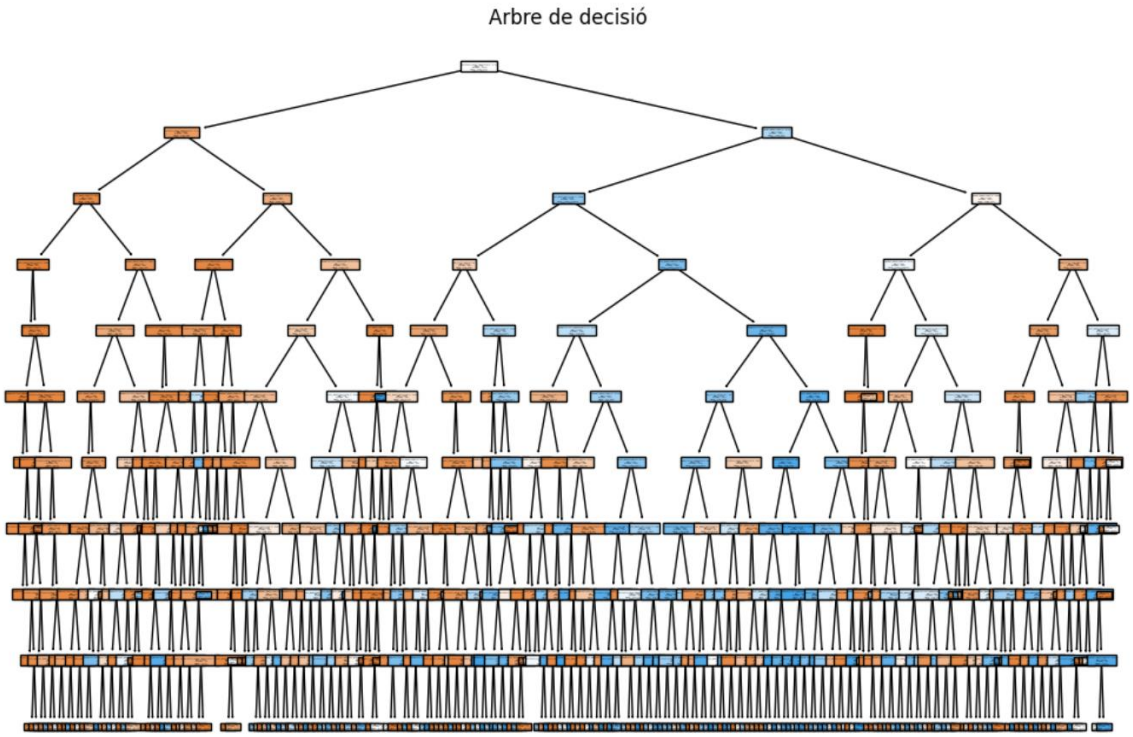
```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Millors paràmetres: {'max_depth': 10, 'min_samples_leaf': 2,
'min_samples_split': 2}
Informe de classificació:
              precision    recall  f1-score   support

           0       0.83      0.69      0.75      1773
           1       0.74      0.86      0.80      1788

    accuracy                           0.78      3561
   macro avg       0.79      0.78      0.77      3561
weighted avg       0.79      0.78      0.77      3561
```

```
Precisió: 0.777
```

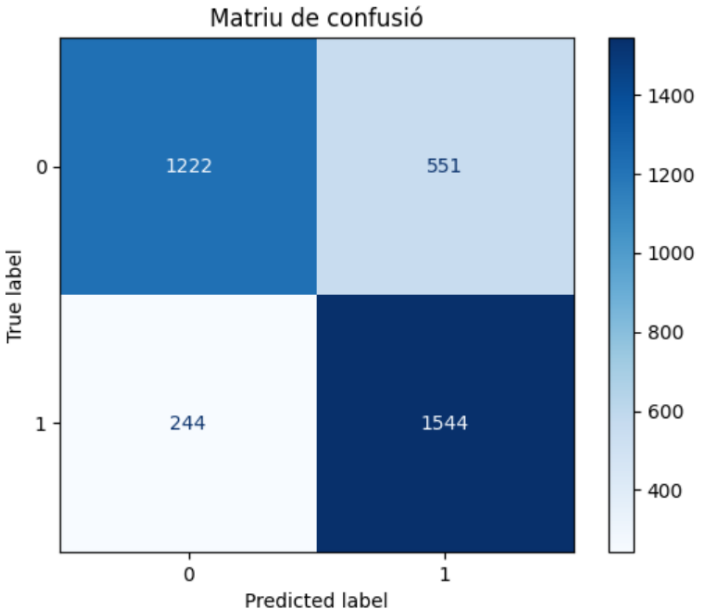
L'arbre de decisió generat amb els millors paràmetres (max_depth: 10, min_samples_leaf: 2, min_samples_split: 2) obté una accuracy de 0.777 al conjunt de validació, inferior als altres models com SVM o KNN. Tanmateix, aquest model també destaca per evitar el overfitting, com es pot observar tant a la representació de l'arbre com als resultats obtinguts.

```
Informe de classificació (conjunt d'entrenament):
              precision    recall  f1-score   support

           0       0.90      0.79      0.84      7092
           1       0.81      0.91      0.86      7151

    accuracy                           0.85     14243
   macro avg       0.85      0.85      0.85     14243
weighted avg       0.85      0.85      0.85     14243

Precisió: 0.849
```

L'arbre manté una estructura equilibrada i controlada, això, combinat amb l'accuracy al conjunt d'entrenament que no arriba a 1, indica que el model no memoritza els patrons de les dades d'entrenament, sinó que generalitza millor. Aquest comportament el fa robust en situacions reals, tot i que a costa d'un rendiment lleugerament inferior en termes d'exactitud en comparació amb altres models més complexos.



Tot i no tenir un mal rendiment, el fet de que l'accuracy sigui molt inferior a les de SVM o KNN no garanteix que aquest model faci bones prediccions i s'ajusti bé a les dades. Per finalitzar s'analitza la seva matriu de confusió corresponent:



Definitivament es mostra que el rendiment del model d'arbres de decisió es inferior al dels altres models. Els errors són més destacables, amb 551 falsos positius i 244 falsos negatius, resultats pitjors que amb els altres models.

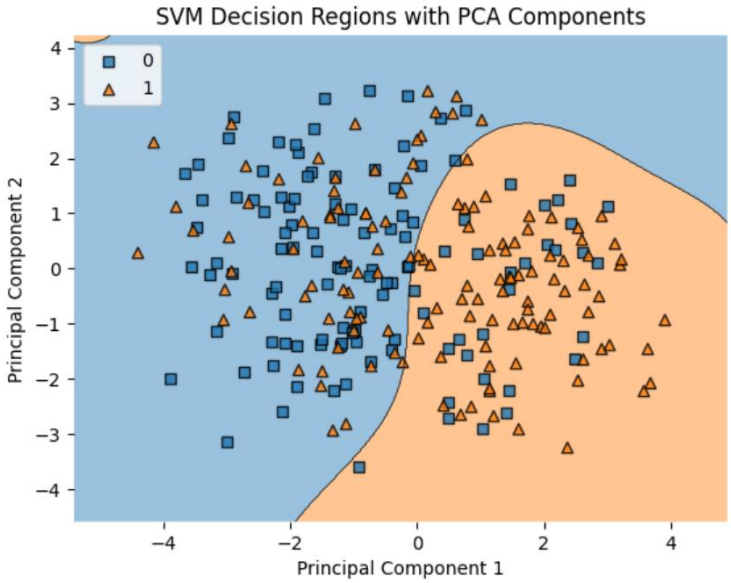
Aquest elevat nombre d'errors, especialment en els falsos positius, fa que el model sigui considerablement el menys fiable dels tres. En comparació amb el KNN i el SVM, aquest model demostra una més incapacitat per generalitzar adequadament i capturar els patrons rellevants del conjunt de dades, limitant així la seva utilitat pràctica. Tot i això, arbres de decisió no presenta overfitting, a diferència de KNN.

En definitiva, el model d'arbres de decisió presenta limitacions significatives, i el seu rendiment més baix el fa menys adequat per aquesta tasca.

5 SELECCIÓ DEL MODEL

El model escollit finalment és el que entrena un Support Vector Machine amb un kernel de RBF, un paramètre de regularització de 10 i una gamma de 0.1

Després d'analitzar els resultats comentats en la secció anterior, s'ha arribat a la conclusió que el model triat és el SVM (Support Vector Machine), justificat amb els següents aspectes; d'entrada, aquest va obtenir la accuracy més alta entre els tres models analitzats, assolint un valor de 0.88 en el conjunt de validació, 0.92 en traint i 0.80 en test. Això indica que és el que millor s'ajusta a les dades per predir la variable objectiu.



En segon lloc, no presenta overfitting, com es pot comprovar a partir del diagrama de regions de decisió, on es mostren fronteres suaus, sense ajustar-se excessivament a les dades concretes del conjunt d'entrenament. A més, això es confirma amb l'accuracy obtinguda en el conjunt d'entrenament, que no arriba a 1, indicant que el model no memoritza les dades d'entrenament i que realment aprèn generalitzant el conjunt.

Seguidament, al fer proves amb el conjunt de test, SVM era el que tenia un millor rendiment en comparació amb KNN i arbre de decisió, reforçant així la decisió d'escollir aquest com la millor opció. La combinació d'una alta accuracy, que no té sobre-ajustament i un bon rendiment en dades noves fan de l'SVM el model més robust i fiable per abordar aquest problema.

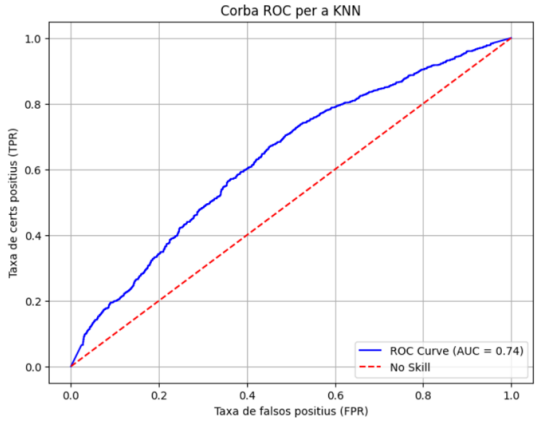
No obstant això, aquest model presenta una limitació clara. Aquesta és la seva complexitat computacional, ja que com es treballa amb un conjunt gran de dades, el model triga molt en executar-se, gairebé més del doble de temps que triga KNN.

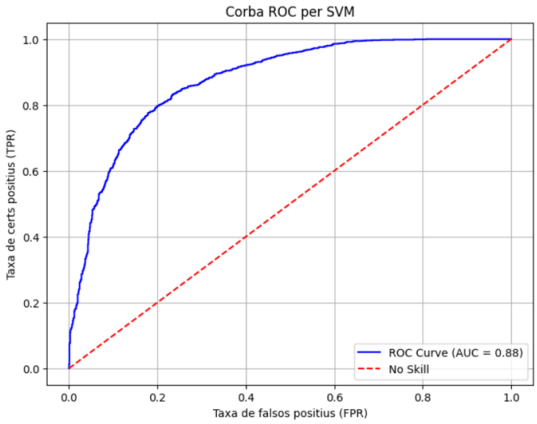
Així doncs, fent la comparació amb traint i val tenim el següent:

Validació	Train	Test
0.88	0.92	0.80

```
Classification Report (Validation):
              precision    recall  f1-score   support

           0       0.91      0.84      0.87      1773
           1       0.85      0.92      0.88      1788

    accuracy                           0.88      3561
   macro avg       0.88      0.88      0.88      3561
weighted avg       0.88      0.88      0.88      3561

Classification report (Train)
              precision    recall  f1-score   support

           0       0.95      0.88      0.91      7092
           1       0.89      0.95      0.92      7151

    accuracy                           0.92     14243
   macro avg       0.92      0.92      0.92     14243
weighted avg       0.92      0.92      0.92     14243

Classifiaction Report (Test):
              precision    recall  f1-score   support

           0       0.79      0.81      0.80      1172
           1       0.81      0.79      0.80      1200

    accuracy                           0.80      2372
   macro avg       0.80      0.80      0.80      2372
weighted avg       0.80      0.80      0.80      2372
```

Finalment, per a acabar de reforçar la decisió d'escollir el SVM es va comprarar la AUC dels models de SVM i KNN:





El KNN té un valor de 0.74 en la AUC, indicant un rendiment no tan elevat en la separació de les classes. La corba és menys pronunciada i mostra una capacitat limitada per maximitzar la taxa de certs positius quan es minimitza la de falsos positius. D'altra banda, el SVM aconsegueix un AUC de 0.88, o sigui que és millor i més robust al diferenciar entre les classes amb una millor relació entre TPR i FPR. La corba del SVM és clarament més propera a l'angle superior esquerre del gràfic, indicant un rendiment global millor que el de KNN.

El model entrenat amb SVM és millor que el de KNN per diversos motius. D'entrada, el SVM no presenta problemes d'overfitting, la seva capacitat de generalització en dades no vistes és molt millor que la de KNN, aquest és la principal raó. Seguidament, SVM ofereix un millor rendiment vist en la accuracy i la AUC. Això s'ha vist al fer experiments amb el conjunt de test, eja que aquest mostra un comportament més consistent, fent-lo el model més adequat.

6 MODEL CARDS

MODEL CARD:
Identificació entre veus reals/falses

Model Details

Overview

Aquest model s'utilitza per predir si una veu ha estat generada per intel·ligència artificial o si és real, basant-se en un conjunt de característiques específiques. El model ha estat entrenat amb un algorisme de SVM, que és un mètode no paramètric i flexible. S'ha utilitzat un kernel rbf per capturar relacions no lineals entre les característiques, amb els hiperparàmetres ajustats a C = 10 i gamma = 0.1.

Autor Daniel Álvarez Sarroca

Data 28 de desembre de 2024

Intended Users

- Estudiants de IAA
- Professorat de IAA

Use Case

Aquest model té com a objectiu identificar si una veu és real o generada de manera artificial, usant característiques acústiques. Està dissenyat exclusivament com una eina d'educació, orientada a treballar amb model cards en el marc d'una tasca d'aprenentatge supervisat, on l'objectiu és classificar la naturalesa de la veu. Aquest model no està pensat per a un ús professional ni comercial, ni per a la realització de entornsreals.

Ethical Considerations

És important considerar que les dades d'entrenament podrien contenir biaixos, els quals poden afectar negativament les prediccions del model.

Dataset

Dades

20 variables ; **12 numèriques** i 8 **de categòriques**.

Partició

Partició en train / val: 80% train 20% val, el conjunt test és el que ja venia donat amb l'enunciat.

Mida Train: 14243
Mida val: 3561
Mida test: 2372

Preprocessat

Missings i outliers: Imputats amb la mediana i outliers usant IQR també. Per a les variables que no eren gaussiantes se'ls hi ha aplicat una transformació logarítmica.

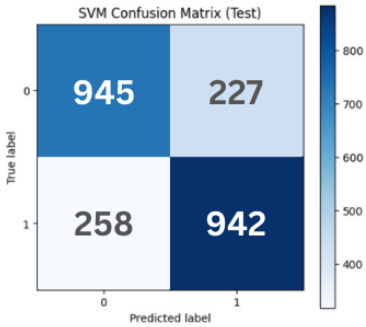
Normalització feta mitjançant amb estandarització **StandardScaler() .**

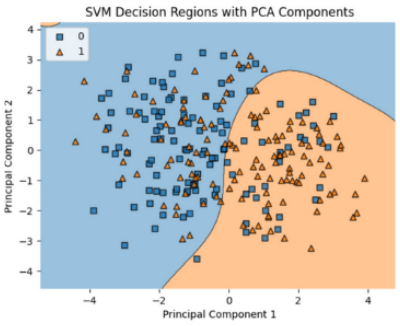
One-Hot-Encoding per a variables numèriques.

Mètriques

Conjunt Test

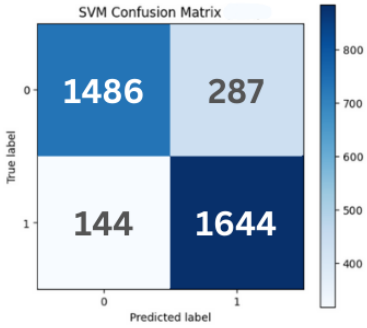
Accuracy: 0.80
F1-Score: 0.80
AUC: 0.88

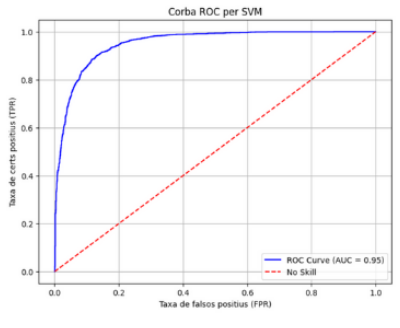




Validation:

Accuracy: 0.88
F1-Score: 0.87
AUC: 0.95





7 BONUS 1: EBM

En aquesta secció s'entrenarà un model de Explainable Boosting Machine. Es compararan els resultats obtinguts amb els models prèviament entrenats (SVM, KNN) utilitzant una taula de mètriques, i s'analitzaran les variables més importants en el conjunt d'entrenament i validació mitjançant una figura que n'indiqui la contribució relativa al model.

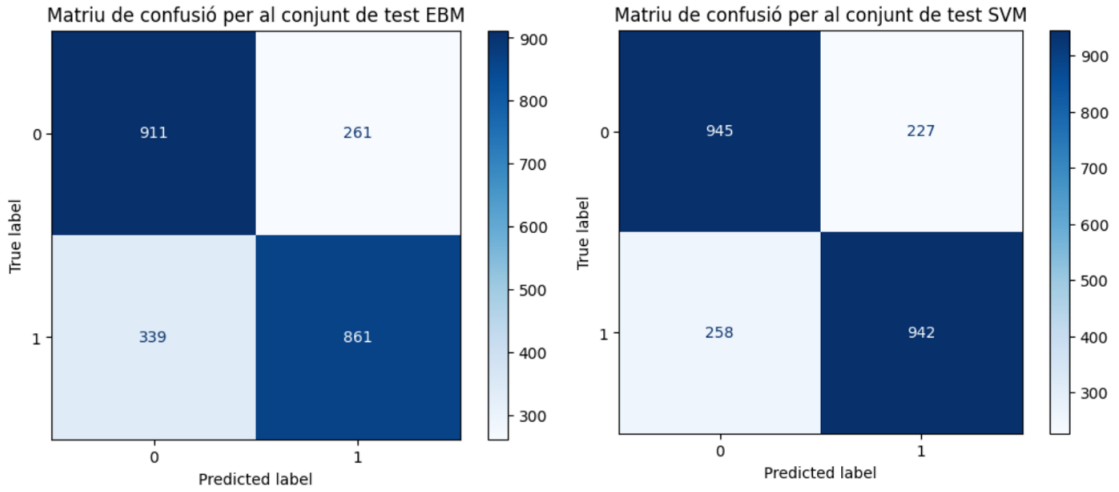
Un cop entrenat el model i fent prediccions amb test:

```
Exactitud del model en el conjunt de test: 0.75

Informe de classificació per al conjunt de test:
              precision    recall  f1-score   support

           0       0.73      0.78      0.75      1172
           1       0.77      0.72      0.74      1200

    accuracy                           0.75      2372
   macro avg       0.75      0.75      0.75      2372
weighted avg       0.75      0.75      0.75      2372
```

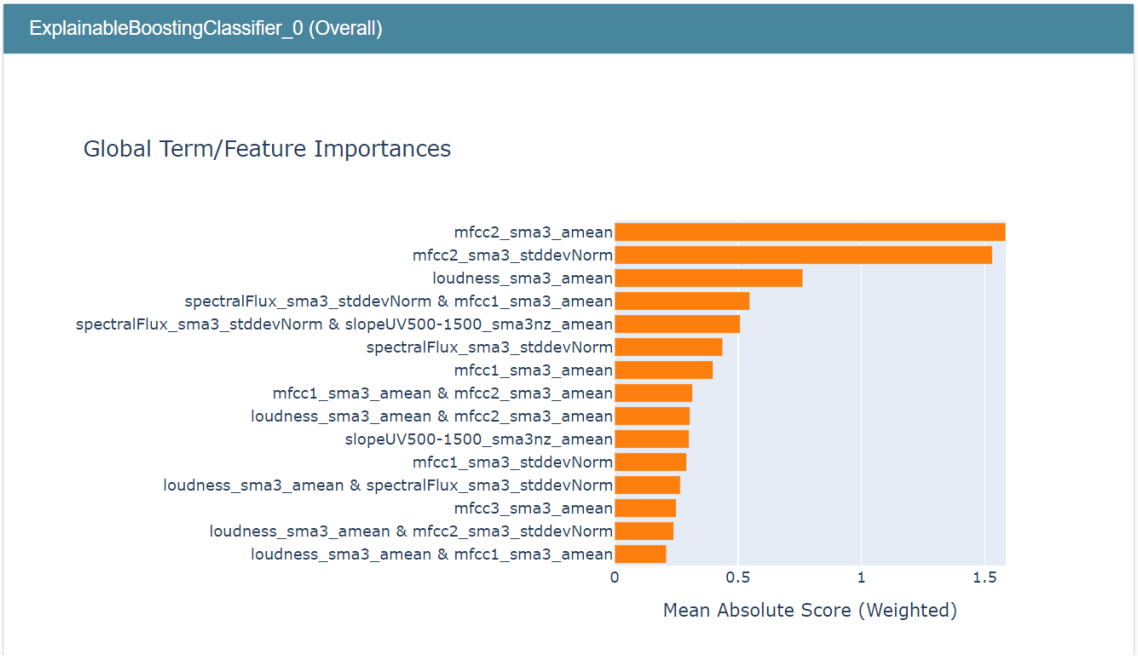
Aquest obté una accuracy de 0.75 en el conjunt de test, amb un rendiment equilibrat entre les dues classes. SVM va obtenir la millor accuracy per al conjunt de test i va demostrar una excel·lent capacitat de generalització, mentre que l'EBM mostra un rendiment inferior tant en precisió com en recall. Pel que fa al KNN, tot i aconseguir una accuracy de 0.73 per al conjunt de test, va presentar signes d'overfitting.

SVM	KNN	EBM
0.80	0.73	0.75

Ara es procedirà a fer un anàlisi de les matrius de confusió per a EBM i SVM:



El SVM supera l'EBM en termes de rendiment, amb menys errors en totes dues classes. La seva capacitat per reduir tant els falsos positius com els falsos negatius el fa més fiable. Un cop feta la taula de l'informe de classificació, es procedeix a fer un anàlisi de la importància dels features per a fer prediccions.



Com es pot veure, les variables **mfcc2_sma3_amean** i **mfcc2_sma3_stddevNorm** són les que més contribueixen al rendiment del model, degut a que tenen un pes molt elevat. D'aquí s'intuiex que són les que tenen una relació molt propera amb la variable resposta i són les principals mostres per a predir el model. Es podria dir també que **loudness_sma3_amean** proporciona informació rellevant per complementar les decisions del model, ja que aquesta es la que està en tercera posició, però molt per sota de les dues primeres. A més, el model aprofita també les relacions entre característiques per millorar les prediccions. Per exemple com passa amb spectralFlux_sma3_stddevNorm & mfcc1_sma3_amean.

8 CONCLUSIONS GENERALS

Abans de començar a parlar de les conclusions sobre els models, cal tenir en compte que aquests no haurien donat tan bons resultats si no fos per la importància del preprocés de dades. En el que s'ha fet imputació de valors nuls amb la mediana, normalització amb estandarització i recodificació de variables categòriques mitjançant one-hot encoding, sense deixar de banda les imputacions amb la mediana dels outliers. També s'ha dut a terme una eliminació de variables en base a les seves correlacions amb la variable resposta, assegurant així que només es treballi amb els features més rellevants per a la tasca objectiu.

Després d'haver realitzat un preprocessat, es va realitzar un procés d'experimentació amb diferents models per tal de poder decidir el millor; un KNN, un arbre de decisió, un EBM i un SVM. Després de fer proves amb els conjunts de validació i test es va acabar determinant que el SVM amb és el model més adequat per a poder predir quines veus eren generades sintèticament. Aquest ha demostrat ser consistent i fiable, amb un alt rendiment en ambdós conjunts, sense signes d'overfitting. El model aconsegueix una bona capacitat de generalització.

En quant a les mètriques, els models han estat avaluats utilitzant la accuracy, f1-score, corbes ROC, matrius de confusió i diagrames de regió. Tot i que els altres dos models han obtingut resultats acceptables, han presentat més errors de classificació, sobretot en falsos positius i falsos negatius, en comparació amb el SVM, que ha mostrat un millor equilibri i menys errors. A més, el KNN té un alt indici de sobre-ajustament a les dades concretes. D'altra banda s'ha observat els arbres de decisió pateixen problemes de rendiment i no aconsegueixen la mateixa robustesa que SVM.

Finalment, tot i els bons resultats del SVM, en un futur, com a continuació de la pràctica es podrien explorar tècniques com l'ajustament de més hiperparàmetres o l'augment del conjunt de dades. Cal dir que aquest projecte ha estat molt interessant de realitzar, i ajuda a entendre millor com funcionen els models predictius aplicats a la realitat.

9 REFERÈNCIES

Google Model Cards. (s. f.). https://modelcards.withgoogle.com/about

1.4. Suport a màquines vectorials. (s. f.). Scikit-aprendre. https://scikitlearn.org/1.5/modules/svm.html

KNeighborsClassifier. (s. f.). Scikit-aprendre. https://scikit-learn.org/1.5/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

1.10. Arbres de decisió. (s. f.). Scikit-aprendre. https://scikit-learn.org/1.5/modules/tree.html

PCA. (s. f.). Scikit-aprendre. https://scikit-learn.org/1.5/modules/generated/sklearn.decomposition.PCA.html

StandardScaler. (s. f.). Scikit-aprendre. https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.StandardScaler.html

Explainable Boosting Machine. (s. f.). Interpret-ml. https://interpret.ml/docs/ebm.html

roc_auc_score. (s. f.). Scikit-aprendre. https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.roc_auc_score.html