Informe de la Pràctica: Detecció de Parla Falsa mitjançant Intel·ligència Artificial

Pau Escobar Asensio

28/12/2024

Universitat Politècnica de Catalunya
Grau en Intel·ligència Artificial
IAA

Índex:

Introducció

Aquesta pràctica de l'assignatura d'Introducció a l'Aprenentatge Automàtic es centra en desenvolupar un model capaç de predir si un àudio és autèntic o ha estat generat mitjançant intel·ligència artificial. Per a aquest objectiu, s'utilitza un conjunt de dades que inclou característiques acústiques i demogràfiques derivades dels àudios. La variable objectiu és 'Realornot', que indica si un àudio és real (1) o generat artificialment (0).

El treball segueix les següents etapes:

- Anàlisi i preprocessat de dades
- Preparació de variables
- Definició i entrenament de models
- Selecció del model final
- Documentació mitjançant una Model Card

1. Anàlisi i preprocessat de dades

1.1 Anàlisi estadístic de les variables de manera independent

El primer que s'ha fet és analitzar les dos bases de dades inicials proporcionades, *full_data_updated_metadata.csv* i *smile_feature_selected.csv*, en la primera hi consta informació de variables categòriques (es decidirà quines seleccionem i quines no) i a la segona hi ha característiques dels audios extretes amb l'eina opensmile. Per tal d'aconseguir-ho hem utilitzat, primerament, el mètode *describe()* per veure una taula amb les mitjanes, valors mínims i màxims, error estàndard, etc i s'ha creat una funció *data_explore()* que ens dona informació estadística sobre un dataframe. A més a més, s'han fet plots per veure les distribucions tant de les variables numèriques com de les categòriques. Alguns resultat interessants a comentar després d'aplicar aquests dos mètodes i de veure les distribucions han estat els següents:

Per a la variable numèrica *F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope* s'ha observat una amplitud considerable, ja que té un valor mínim de -2448.99 i un valor màxim de 4659.96. Això suggereix que possiblement hi hagi outliers en aquesta variable i utilitzem l'histograma que hem generat (on l'eix x són els valors que pren la variable en intervals i l'eix y la freqüència amb la que els valors cauen dins de cada interval) per veure-ho:

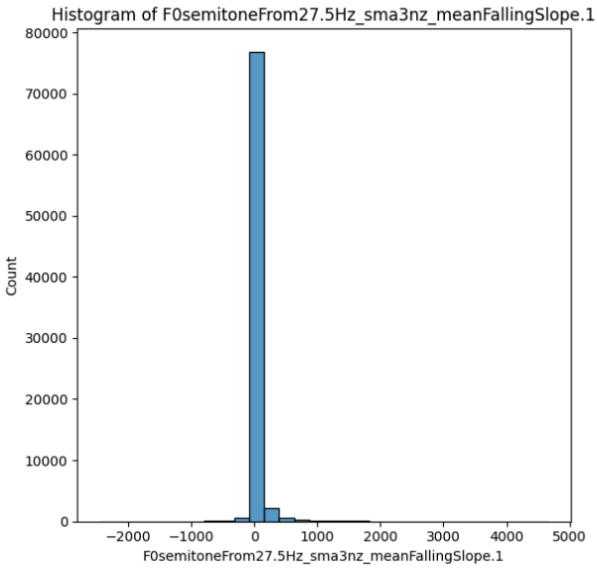

Figura 1: histograma d'una variable numèrica amb molta amplitud de valors

Una altre variable bastant interessant per comentar és *mfcc1_sma3_amean*, que pren un valor mitjà de 18.377609 amb un desviació estàndard de 4.111353. Fet que suggereix segueix una distribució bastant gaussiana sense cap preprocessat, ho podem veure a l'histograma corresponent:

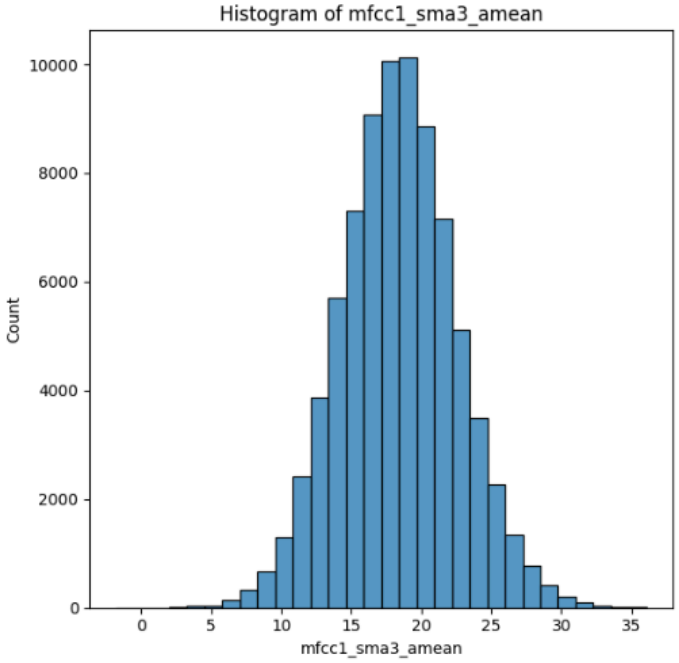

Figura 2: histograma d'una variable numèrica que segueix una distribució gaussiana

Cal comentar què a partir d'aquest punt, per tal de simplificar aquesta primer part de la pràctica, s'ha treballat directament sobre el *training_set.csv* proporcionat. Aquesta decisió ha estat presa perquè (després de l'anàlisis estadístic independent de les variables del *full_data_updated_metadata.csv*) s'ha considerat que les variables que constaven a la base

de dades original i que no estaven presents a la base de dades d'entrenament proporcionada no eren rellevants:

- *Filename*, *F1*, *F2*, *F3*, *F4*, *Transcription* i *Times* no aporten al model informació significativa per tal de predir si l'audio és generat artificialment, per tant es considerem irrellevants (el soroll que poden generar pot ser una font de problemes).
- S'ha pogut observar amb la funció *data_explore()* que la variable *Variant* té casi un 97% de missings, cosa que la fa irrellevant.
- *File_Target_ID* i *File_ID* ens dona informació que ja tenim present al dataframe mitjançant altres variables.
- *Utterance, Source_Utterance* i *Target_Utterance* també les considerem irrellevants per la predicció.

No obstant, també s'han realitzat canvis al *training_set.csv* i al *smile_feature_selected.csv*:

Al primer, s'ha fet una fusió de les variables *Sex* amb *Target_Sex* i *Country* amb *Target_Country* de la qual han resultat les variables *Final_sex* i *Final_country*, a més a més, s'ha fet *drop()* de les variables utilitzades per fer la fusió i de *Source_Sex* i *Source_Country* perquè ja no són necessàries. Això s'ha fet perquè hagués sigut un greu error conservar aquestes variables que ens donaven informació sobre el sexe i el país ja que les variables que contenen aquest Target i Source eren les que es feien servir per generar el audio artificialment i les variables *Sex* i *Country* només eren presents a les característiques dels audios reals, per tant el model s'hagués pogut basar en aquesta presència o absència dels features que acabem de comentar per resoldre el problema exposat, cosa que faria que el model no fos vàlid. Per aquest mateix motiu, s'ha fet *drop()* sobre la variable *Category*. També s'han eliminat les variables *Source_ID*, *Target_ID*, *ID* i *F_path* perquè es consideren irrellevants.

Al segon, hem fet *drop()* d'una variable que s'ha observat que estava duplicada.

Tots aquests canvis que hem realitzar al *training_set.csv*, que ens han donat la base de dades d'entrenament sobre la que treballarem a continuació (a part de amb la base de dades de l'smile), els hem replicat al *test_set.csv*.

1.2 Estudi de balanceig de classe objectiu

No s'ha observat cap desequilibri a la variable objectiu 'Realornot' en cap de les bases de dades del *training_set.csv* i el *test_set.csv*. Per validar-ho hem utilitzat el mètode *value_counts()* que ens ha donat aquest resultat pel dataframe d'entrenament: 'Realornot' 1: 8939, 0: 8865; i aquest pel dataframe de test: 'Realornot' 1: 1200, 0: 1172. Es pot observar que està ben balancejada, tot i així, per veure-ho de manera més clara ho validem gràficament amb un plot on podem veure a l'eix x els valors que pren la variable 'Realornot' (0 i 1) i a l'eix y la freqüència d'aquests pel dataframe d'entrenament:

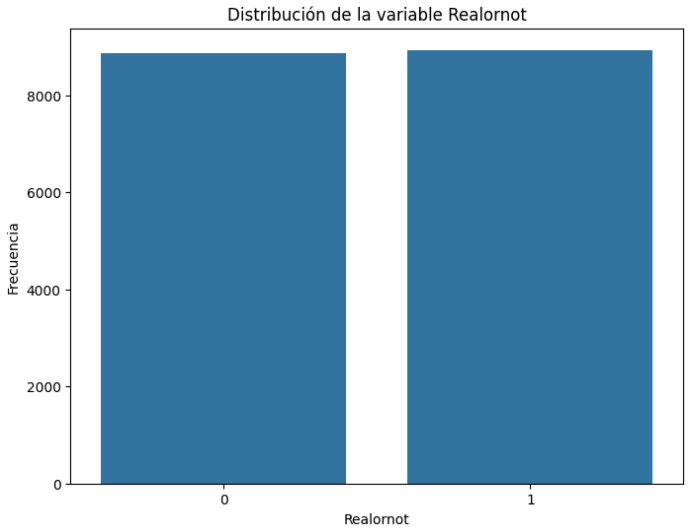

Figura 3: Plot de la distribució de la variable *Realornot* per validar que està balancejada

1.3 Particionat del dataset

Fem la partició de la base de dades d'entrenament en entrenament/validació en aquest punt de la pràctica ja que es gestionen els outliers com a missings i els s'imputen (tal i com es pot veure en els següents apartats). Així doncs s'ha particionat assignant un 80% de les dades a l'entrenament i un 20% a la validació. S'ha fet d'aquesta manera perquè aquesta divisió proporciona un equilibri entre entrenar bé el model i avaluar la seva capacitat per generalitzar a nous dades, a més a més, ens ajudarà a prevenir el overfitting en un futur.

Mida de la base de dades d'entrenament: (17804, 4)

Mida de la base de dades d'entrenament després de la parició: (14243, 4)

Mida de la base de dades de validació: (3561, 4)

1.4 Gestió de missings

Hem utilitzat un altre cop el mètode *data_explore()* que ha permès veure si les variables del dataframe d'entrenament i de l'smile contenien missings i el percentatge d'aquests. Després d'aplicar-ho s'ha observat que no constaven missings a cap variable:

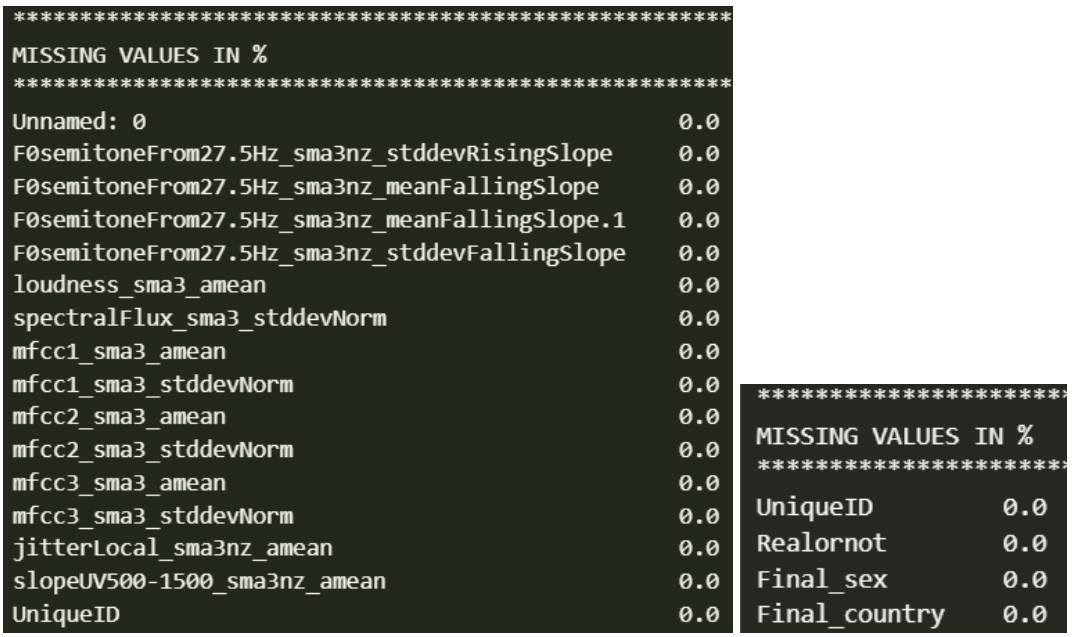

Figura 4: Percentatge de missings pels dataframes d'entrenament i de l'smile

Tot i així, tal i com veurem al següent apartat, es tracten els outliers com a missings. Per tractar aquests missings s'han considerat dos possibles estratègies:

1. Eliminar-los: Implicaria suprimir més de 30000 files amb informació, cosa que resultaria en una gran pèrdua de les dades. Per tant, descartem aquesta opció
2. Imputar: Substituir les dades mancants per certs valors té com a objectiu conservar totes les observacions evitant així la pèrdua d'informació. Així doncs seleccionem aquesta tècnica per tractar els missings encara que hi hagi un risc d'introduir biaix a l'anàlisi.

Així doncs, per imputar els outliers (marcats com a NA) també es van considerar diferents opcions amb les que s'ha experimentat: primerament es van reemplaçar els valors dels outliers de cada variable pel valor mitjà d'aquestes, no va ser una bona opció ja que la mitjana es veu molt afectada pels outliers. Seguidament, es va plantejar fer-ho substituint-los pel valor de la mediana (que no es veu tant afectada pels outliers) però va resultar que, després de la imputació, hi havia massa concentració en aquest valor i per tant estem introduint massa biaix el model. Per últim, es va utilitzar imputació mitjançant KNN que és la que ha donat millors resultats, sent l'opció finalment escollida.

1.5 Normalització de les variables

Motiu pel qual es normalitza abans de tractar els outliers:

S'ha decidit que el més adient era normalitzar les variables abans de tractar el outliers. L'ordre en què es normalitzen les variables i es tracten els outliers afecta significativament la qualitat del model. Fer-ho al revés, és a dir, tractar els outliers abans de normalitzar pot augmentar el risc de **overfitting** ja que distorsiona l'escala de les variables, influint en la mitjana i desviació estàndard de la normalització posterior. A més, els outliers no normalitzats tenen un pes desproporcionat, el que pot portar a decisions esbiaixades.

Normalitzant primer, les variables queden a una escala comuna, millorant la detecció d'outliers i evitant la propagació de soroll, resultant en models més robustos i generalitzables.

Ho hem experimentat avaluant el rendiment del model final canviant l'ordre corresponent i ho hem pogut validar. Els resultats corresponents a l'experimentació es poden trobar a l'apartat 7.1 del report

Distribucions de les variables:

Per dur a terme això s'ha utilitzat una còpia del dataframe en la que descartem algunes variables com *Unnamed: 0* i *UniqueID* que no té sentit incluir-les i s'ha descartat els valors extrems de cada variable amb el objectiu de poder apreciar millor la distribució de cada variable sense que es vegi afectada pels outliers més grans. Per visualitzar la distribució s'han utilitzant diferents mètodes com histogrames amb superposició de la curva de densitat d'una distribució normal, qq plots o KDE (Kernel Density Estimation) que ha donat el següent resultat:

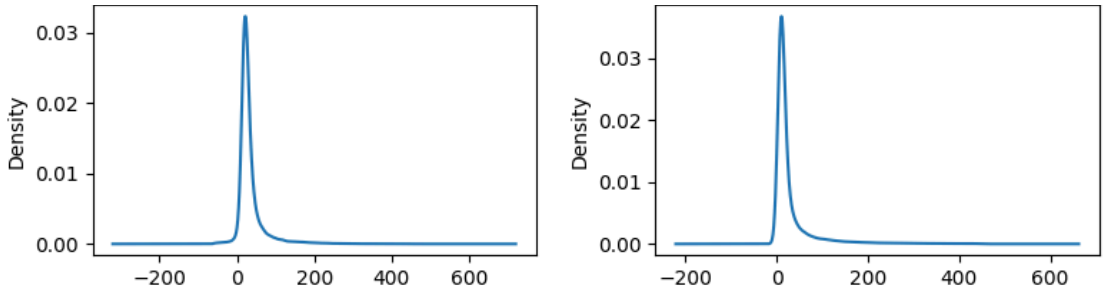

Figura 5: Distribucions de dues les variables visualitzades mitjançant KDE

Estratègia seguida per normalitzar i motivació:

Per a les variables *F0semitone*, on els valors tenen una gran dispersió, es fa una transformació logarítmica seguida d'un escalat robust amb *RobustScaler()*, que és menys sensible als outliers i ajuda a controlar els valors atípics sense distorsionar la distribució. Les variables com *loudness_sma3_amean* i els coeficients *MFCC*, que tenen una distribució aproximadament normal, es normalitzen amb *StandardScaler()* per garantir que tinguin mitjana 0 i desviació estàndard 1, facilitant així la comparació directa entre elles. Per les variables amb rangs petits i distribucions similars, com *jitterLocal_sma3nz_amean*, s'utilitza *MinMaxScaler()* per escalar-les en un interval [0,1], la qual cosa ajuda a millorar l'eficàcia de les tècniques estadístiques. Finalment, les variables que ja estan normalitzades, com *stddevNorm*, es mantenen sense canvis.

Observem els resultats de la normalització mitjançant KDE, que estima la densitat de probabilitat d'una variable contínua, amb l'eix X com els valors de la variable i l'eix Y com la densitat. És útil per comprovar la gaussianitat perquè permet veure si la distribució té forma de campana, pròpia d'una distribució normal. Els plots que podem veure són un cop s'ha realitzat la imputació del outliers ja que ens permet veure molt millor la distribució de les variables:

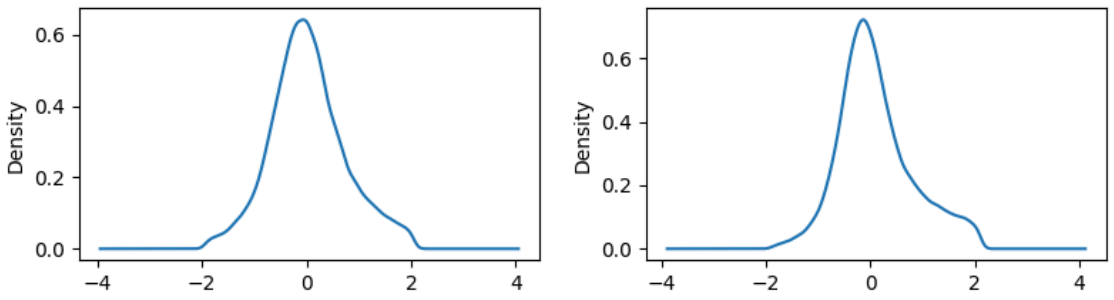

Figura 6: KDE de les mateixes variables que hem visualitzat abans un cop normalitzades i sense outliers

1.6 Identificació i tractament d'outliers

Els outliers s'han identificat utilitzant el criteri del rang interquartílic (IQR), que és una mesura estadística que descriu la dispersió d'un cojnunt de dades i que es calcula com la diferència entre el tercer quartil (Q3) i el primer quartil (Q1). Sent el primer quartil el valor que deixa per sota el 25% de les dades i sent el tercel quartil el valor que deixa per sota el 75% de les dades. Per tant, es marquen com a outliers els valors més petits que Q1 - 1,5*IQR i els valors més grans que Q3 + 1,5*IQR.

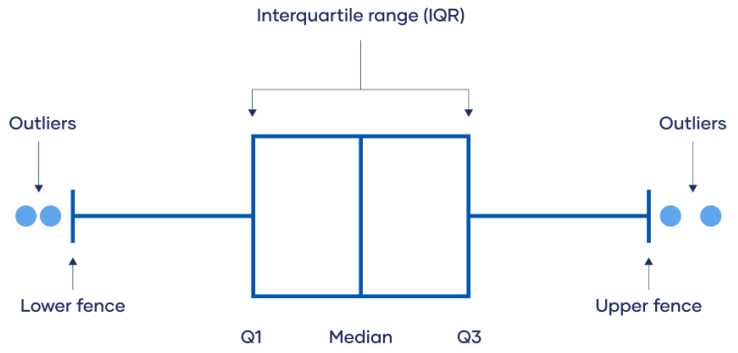

Figura 7: Descripció visual del criteri que hem seguit per detectar outliers

Així doncs, primerament, s'ha calculat el percentatge d'outliers de cada variable utilitzant aquest mètode en una funció python *calculate_outliers_percentage()* sobre la base de dades d'entrenament (0% d'outliers a totes les variables); i sobre l'smile (aquest últim ens ha donat outliers a quasi totes les variables), on hi ha una variable amb un 13% d'outliers, tal i com podem veure a la següent imatge:

	Variable	Outliers (%)
0	Unnamed: 0	0.000000
1	F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope	1.199020
2	F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope	7.545536
3	F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope	8.420362
4	loudness_sma3_amean	3.556226
5	spectralFlux_sma3_stddevNorm	1.383389
6	mfcc1_sma3_amean	1.053009
7	mfcc1_sma3_stddevNorm	3.292665
8	mfcc2_sma3_amean	1.067858
9	mfcc2_sma3_stddevNorm	13.409473
10	mfcc3_sma3_amean	1.085181
11	mfcc3_sma3_stddevNorm	8.598545
12	jitterLocal_sma3nz_amean	4.040042
13	slopeUV500-1500_sma3nz_amean	3.050139
14	UniqueID	0.000000

Figura 8: Percentatge d'outliers per cada variable de la base de dades de l'smile

Així doncs hem procedit, tal i com hem comentat a l'apartat 1.4 de gestió de missings, a marcar els valors que es detecten com a outliers mitjançant IQR com a NA i imputant aquests NA mitjaçant KNN. Hem visualitzat els resultats mitjançant histogrames, a continuació podem veure l'estat final de la variable *F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope* que haviem vist a l'apartat 1.1 de la pràctica:

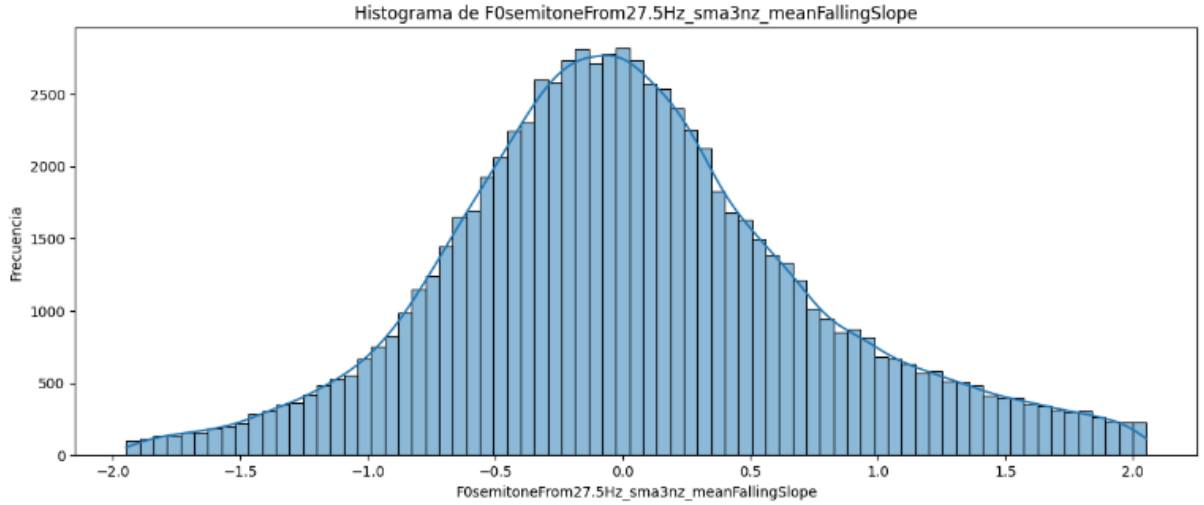

Figura 9: Distribució de la variable mitjançant un histograma un cop normalizada i sense outliers

1.7 Recodificació de variables

Abans de recodificar les variables, apliquem el mètode *merge()* per unir els datasets d'entrenament, test i validació amb el dataset d'smile preprocessat corresponentment, guinat-se mitjançant la variable *UniqueID* per fer-ho.

Ara sí, recodifiquem:

- Per la variable *Final_country* utilitzem one hot encoding mitjançant el mètode *get_dummies()*.

- Per la variable *Final_sex,* com que ja és binària, utilitzem un map per codificar 'Male' com a 0 i 'Female' com a 1.
-

Finalment, utilitzem el mètode *astype(int)* per convertir els valors de les columnes recodificades amb one hot encoding a nombres enters.

Motivació de la recodificació:
La recodificació converteix les variables categòriques Final_country i Final_sex en variables numèriques utilitzant one-hot encoding i mapatge binari, respectivament, per preparar les dades per a l'aprenentatge automàtic. Això assegura que les dades estiguin en un format adequat per als models, millorant la seva precisió i eficiència.

2. Preparació de variables

2.1 Anàlisi de variables categòriques i variable objectiu:

Al fer la recodificació de les variables categòriques a variables numèriques anteriorment ens hem quedat sense variable categòriques. Tot i això, s'utilitzen les mateixes que s'han recodificat per tal de poder assolir aquesta tasca

A continuació es pot veure en bar plots la distribució de cada variable per cada valor de la variable objectiu, on a l'eix X es veuen els valors que prenen les variables categòriques i a l'eix Y la freqüència d'aquests. Sent el color blau el corresponent al valor 0 de la variable objectiu i sent el color taronja el corresponent al valor 1:

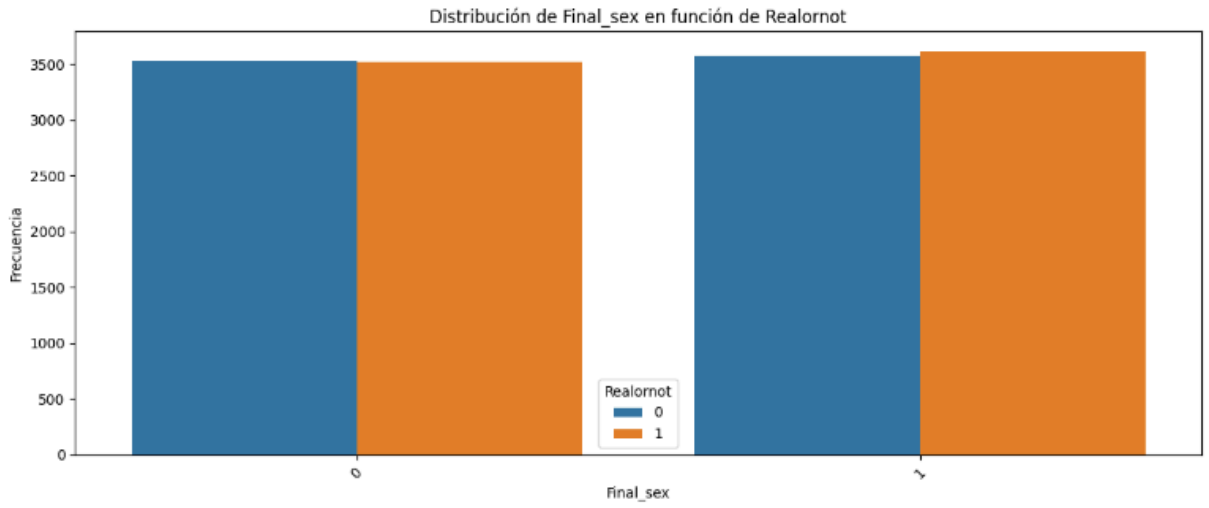

Figura 10: Valors de la variable categòrica Final_sex per cada valor de la variable objectiu

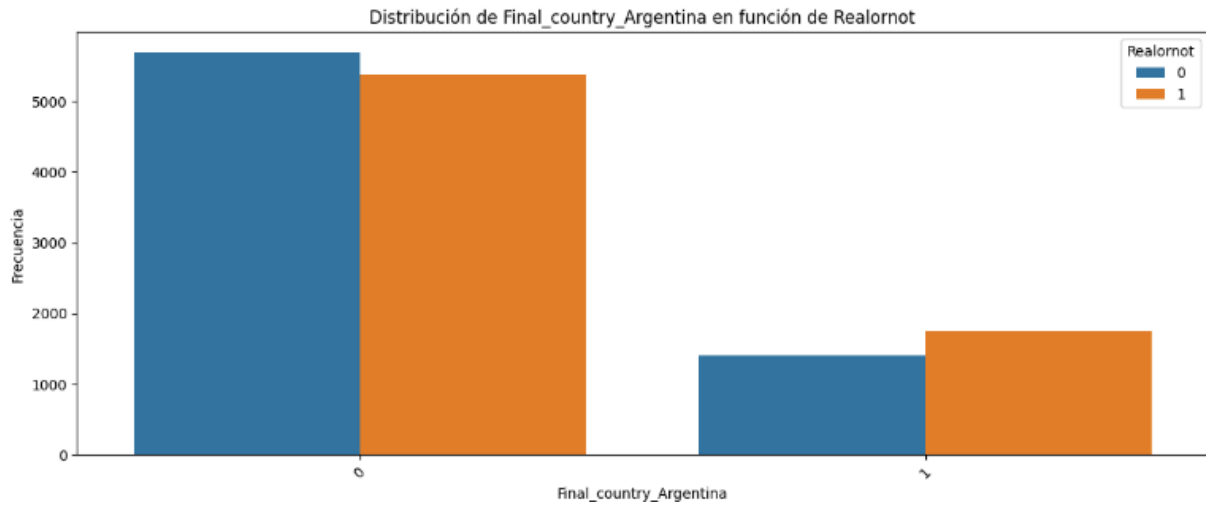

Figura 11: Valors de la variable categòrica Final_country_Argentina per cada valor de la variable objectiu

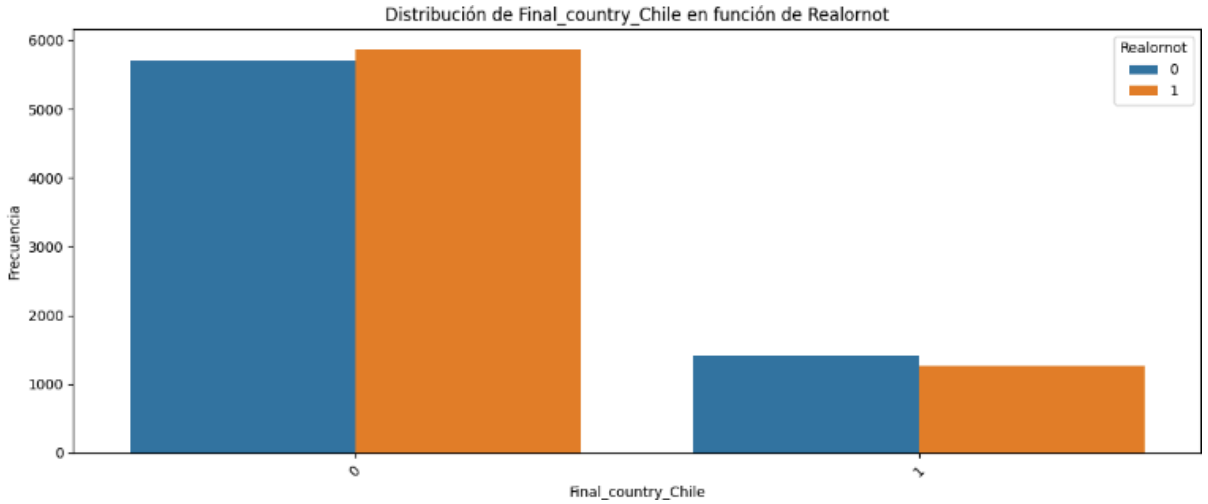

Figura 12: Valors de la variable categòrica Final_country_Chile per cada valor de la variable objectiu

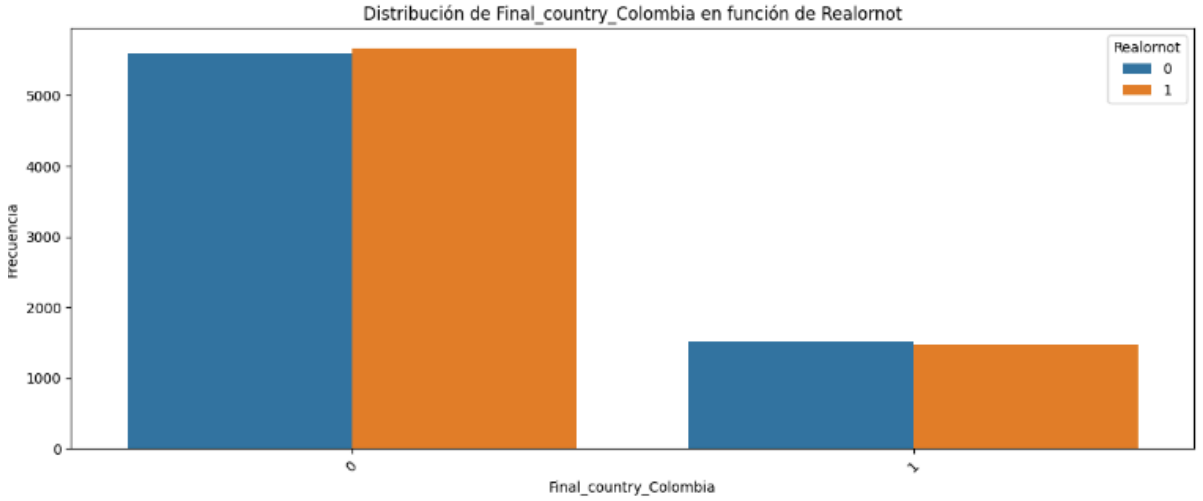

Figura 13: Valors de la variable categòrica Final_country_Colombia per cada valor de la variable objectiu

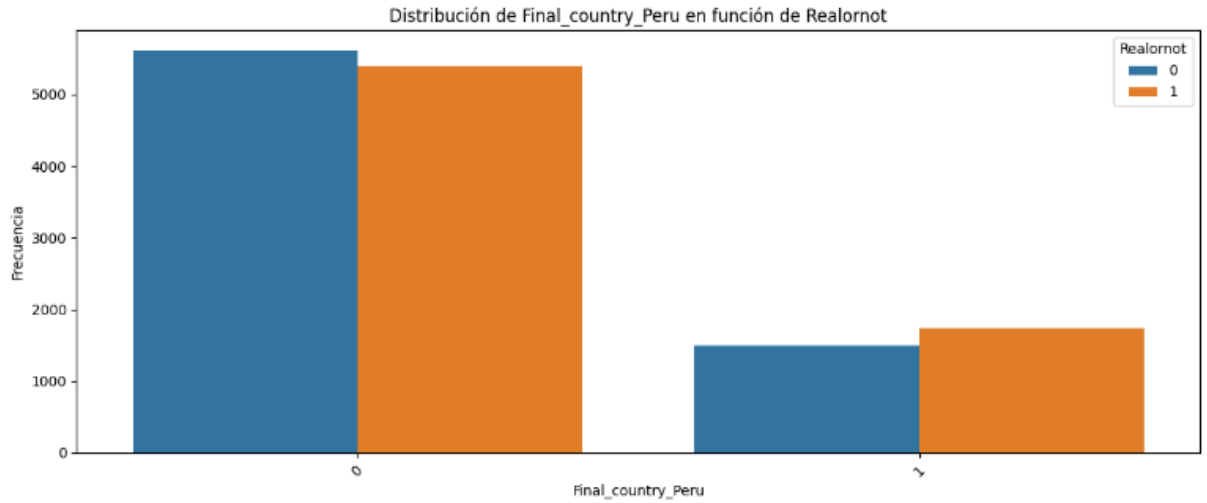

Figura 14: Valors de la variable categòrica Final_country_Peru per cada valor de la variable objectiu

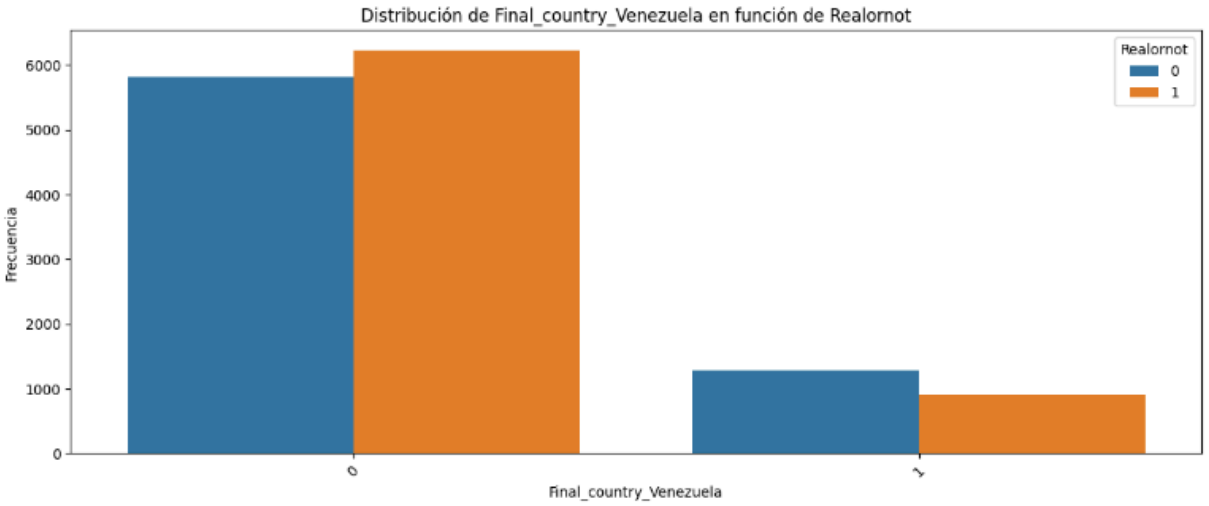

Figura 15: Valors de la variable categòrica Final_country_Venezuela per cada valor de la variable objectiu

Primerament es pot veure que en tots els casos, la variable objectiu està balancejada, fet que és clau. A més a més, es pot observar que la variable *Final_sex* pren la mateixa freqüència de valors mentre que totes les variables del tipus *Final_country_* prenen amb bastanta més freqüència el valor 0 que l'1, fet que no considerem trascedent pel correcte funcionament del model, ja que com acabem de comentar, tan pels valors 1 o 0 d'aquestes variables la classe objectiu continua estant balancejada.

2.2 Anàlisi de correlació

Per realitzar l'anàlisi de correlació primerament s'ha mirat com correlaciones les variables numèriques amb la variable objectiu. Hem fet un plot d'un Heatmap que ho permet veure, on a l'eix X trobem 'Realornot' i a l'eix Y la resta de variables:

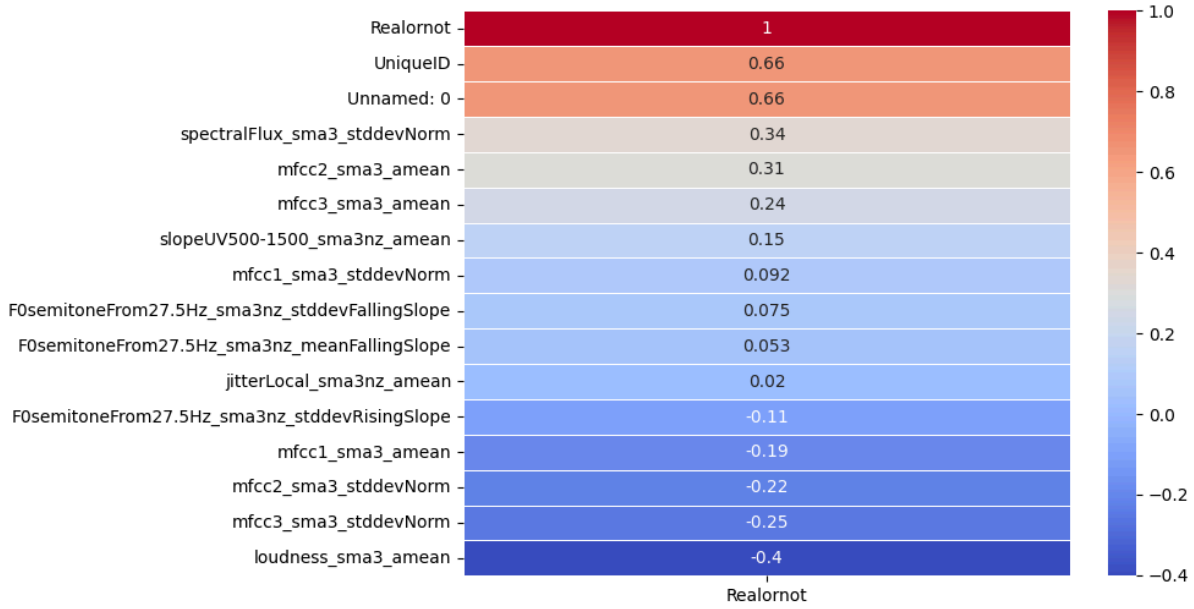

Figura 16: Correlacions de les variables numèriques amb la variable objectiu

La motivació d'aquest Heatmap és descartar les 5 variables que menys correlacionen amb 'Realornot', ja que es considera que no seran rellevants a l'hora d'entrenar el model. No descartem més seguint aquest criteri perquè sinó podriem estar perdent bastanta explicabilitat. Les variables descartades han estat aquestes:

- 'F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope'
- 'F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope'
- 'jitterLocal_sma3nz_amean'
- 'F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope'
- 'mfcc1_sma3_stddevNorm'

A més a més, al notebook també es fa el mapa de correlacions de totes les variables amb totes. D'aquesta manera, si es veu que dues variables tenen una correlació rellevant (signficant que amb una variable ja pots explicar l'altre), es pot descartar una de les dues. Així doncs, seguint aquest altre criteri s'ha decidit descartar aquestes dues variables ja que estaven altament relacionades amb la variable corresponent a la seva mitjana, i a més a més, són les dues variables que tenen unes distribucions que s'allunyen més de la gaussiana:

- 'mfcc2_sma3_stddevNorm'
- 'mfcc3_sma3_stddevNorm'

Finalment, aprofitem per fer descartar ''Unnamed: 0' i 'UniqueID' ja que seria un greu error conservar-les en els nostres datasets ja que el model podria arrivar a una resposta creuant aquests índexs entre les diferents bases de dades i, per tant, no seria valid.

2.3 Estudi de dimensionalitat

Per realitzar aquest estudi s'ha implementat un codi aplica l'Anàlisi de Components Principals (PCA) per reduir la dimensionalitat sense tenir en compte les variables 'Realornot', 'Final_sex' que són binaries ni les variables recodificades amb one hot encoding. Primer, selecciona les columnes numèriques especificades, les estandarditza amb `StandardScaler` per garantir que totes tinguin mitjana 0 i desviació estàndard 1, i després ajusta el model de PCA sobre les dades escalades. Es calcula la variància explicada per cada component principal i la variància acumulada, que es representen gràficament:

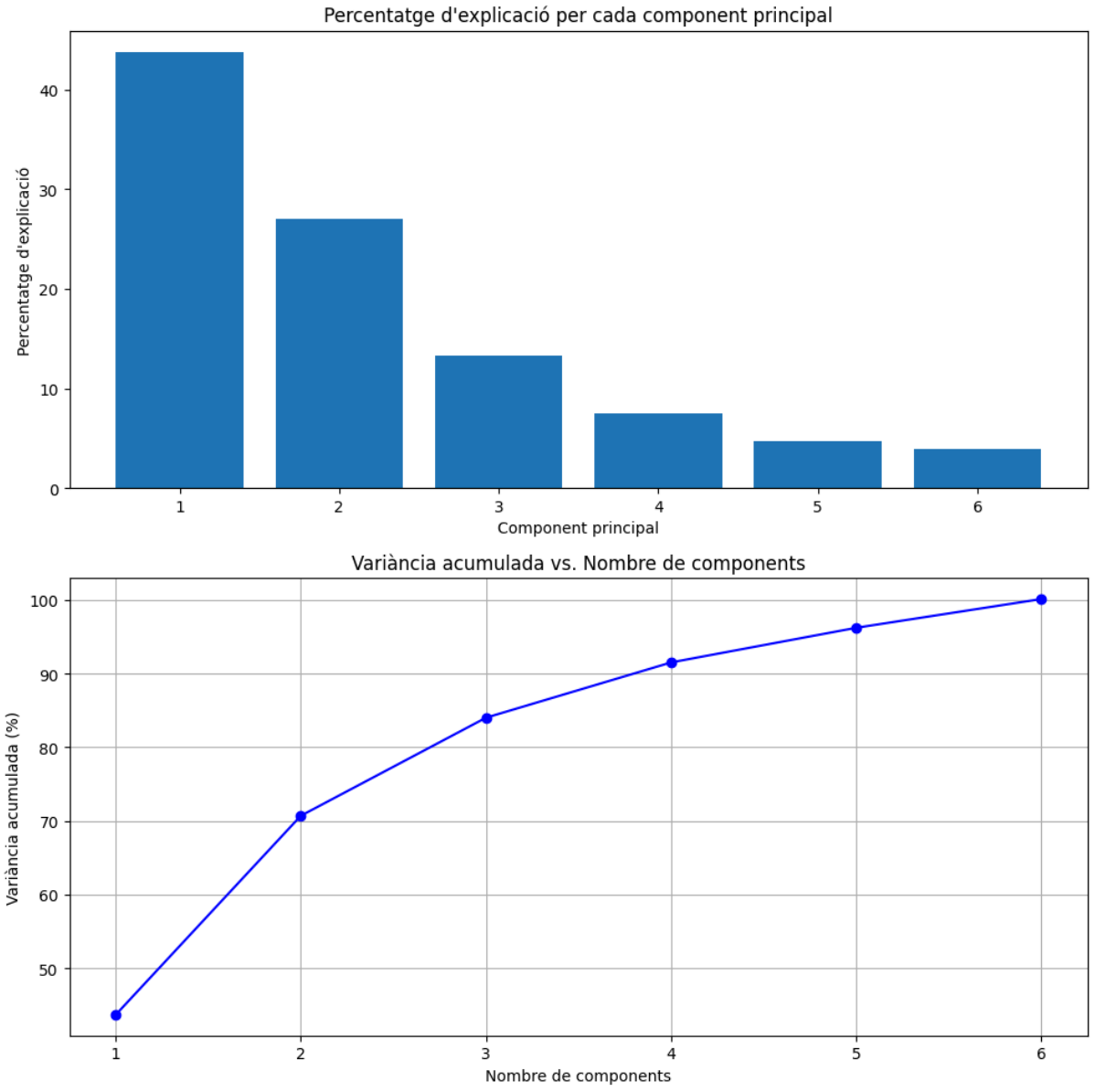

Figura 17: Percentatge d'explicació i variància acumulada per cada component

Un cop vistos els resultats sembla ser que, a priori, sortiria a compte reduir la dimensionalitat del model, ja que amb 3 components es pot capturar una variança acumulada de més del 80%, tot i així, potser no sortia tan rentable tenint en compte que ja tenim molt pocs features. És per aquest debat que es va decidir avaluar el rendiment dels 3 models proposats havent reduït la dimensionalitat i sense haver-ho fet i es va poder

observar que el rendiment del model final empitjorava al reduir la dimensionalitat, molt possiblement per això que es comentava anteriorment que no surt a compte perdre l'explicabilitat de certs features. Per tant, finalment no s'ha reduït la dimensionalitat del model.

Els resultats d'aquesta experimentació estan a l'apartat 7.2 del report.

3. Definició de models

3.1 Mètriques de rendiment

Primer de tot, hem definit quines seran les mètriques que s'utilitzaran per avaluar el rendiment del model. Però abans és important primer entendre què vol dir cada una de les següents coses segons el nostre model:

• True Positives (TP): Audios predits com a reals que sí que ho són.
• True Negatives (TN): Audios predit com a falsos que sí que ho són.
• Falsos Positius (FP): Audios predits com a reals que no ho són.
• Falsos Negatius (FN): Audios predits com a falsos que no ho són.

Accuracy

L'**accuracy** mesura el percentatge de prediccions correctes del model sobre el total de prediccions. És una mètrica global útil per avaluar el rendiment general, però pot ser enganyosa en conjunts de dades desequilibrats. Per exemple, si el 90% de les dades pertanyen a una classe i el model prediu sempre aquesta classe, l'accuracy serà elevada (90%), tot i que el model no detecta bé la classe minoritària.

Precision

La **precisió** és la proporció de prediccions positives correctes respecte al total de prediccions positives fetes pel model. És especialment important en escenaris on els falsos positius tenen un cost alt, com en diagnòstics mèdics o detecció de fraus. Una precisió elevada indica que el model no comet molts errors quan classifica una instància com a positiva.

Recall (o Sensibilitat)

El **recall** calcula la proporció de veritables positius detectats pel model respecte al total d'instàncies positives reals. Aquesta mètrica és crítica quan els falsos negatius tenen conseqüències greus, com en la detecció de malalties o anomalies. Un valor alt de recall assegura que el model identifica correctament la major part dels casos positius.

F1-Score

L'**F1-Score** és la mitjana harmònica entre la precisió i el recall. Aquesta mètrica equilibra ambdues mesures i és especialment útil quan hi ha un compromís entre precisió i

sensibilitat. És ideal en conjunts de dades desequilibrats, on cal una visió equilibrada del rendiment del model.

ROC Curve i AUC

La corba **ROC (Receiver Operating Characteristic)** mostra la relació entre el **True Positive Rate** (recall) i el **False Positive Rate** a diferents llindars de classificació. Una corba que s'acosta a la cantonada superior esquerra indica un model excel·lent. L'**AUC (Area Under the Curve)** és una mètrica numèrica que quantifica aquesta àrea, amb valors propers a 1 que indiquen un bon rendiment. És útil per comparar diferents models de classificació, especialment quan es treballa amb conjunts de dades desequilibrats.

Confusion Matrix

La **matriu de confusió** ofereix una anàlisi detallada de les prediccions del model. Presenta quatre valors: **veritables positius (TP)**, **veritables negatius (TN)**, **falsos positius (FP)** i **falsos negatius (FN)**. Aquesta matriu permet identificar errors específics del model, com ara si confon sistemàticament una classe amb una altra. És una eina visual molt útil per entendre el comportament del model en profunditat.

3.2 Mètrica a maximitzar

És essencial pel correcte entrenament del models que en nostre cas la mètrica a maximitzar és la **Precision**. Això es deu a què la pràctica consisteix en detectar parla falsa, i per tant, maximitzant la precision minimitzarem els Falsos Positus (FP), això vol dir que estarem minimitzant els audios que s'estan predint com a positius però realment són negatius, aconseguint un millor rendiment així detectant la parla falsa.

3.3 Possibles hiperparàmetres i selecció dels òptims

Per a tots els models s'ha utilitzat **GridSearchCV** per trobar la millor combinació d'hiperparàmetres mitjançant validació creuada amb **10 folds** i optimitzant la mètrica de precisió (**precision**), el motiu d'això està exposat anteriorment a l'apartat 3.2 del report. Es defineix una graella (param_grid) amb diferents valors per als hiperparàmetres. Després d'entrenar el model amb totes les combinacions possibles, es retorna la configuració òptima i es calcula la precisió mitjana i la seva desviació estàndard en les particions. Això assegura un rendiment òptim i consistent del model sobre les dades d'entrenament.

S'ha decidit fer 10 folds ja que, a diferència de fer-ho amb 5, cada model s'entrena en un subconjunt lleugerament més gran de dades, la qual cosa proporciona una millor estimació de la capacitat del model de generalitzar a noves dades. Amb menys folds, com ara 5, cada model tindria menys dades per aprendre, la qual cosa augmentaria el risc de biaix.

KNN (K-Nearest Neighbors)

Possibles hiperparàmetres:

- **'n_neighbors': [3, 5, 7, 8, 9, 11]**: El nombre de veïns més propers té una influència directa sobre la classificació. Valors més baixos poden fer que el model sigui

sensible al soroll, mentre que valors massa alts poden suavitzar massa les prediccions. Es s'ha explorat un rang relativament estret per equilibrar aquestes opcions.
- **'weights': ['uniform', 'distance']**: Aquest hiperparàmetre controla com es ponderen els veïns. La configuració 'uniform' dóna el mateix pes a tots els veïns, mentre que 'distance' dóna més pes als veïns més propers.
- **'metric': ['euclidean', 'manhattan', 'minkowski']**: Les diferents mètriques de distància poden influir en els resultats segons la natura de les dades. 'Manhattan' és una opció més robusta per a dades no lineals.

Hiperparàmetres òptims:

- **'metric': 'manhattan', 'n_neighbors': 8, 'weights': 'uniform'**

Anàlisi: Els valors òptims suggereixen que el model KNN es beneficia d'una mètrica de distància 'manhattan', que és menys sensible a les distàncies molt llargues, i una configuració de veïns uniformes (sense ponderar per distància). El valor de 'n_neighbors' a 8 indica un bon balanç entre un model prou específic (evitant el sobreajustament) i prou general (evitant el subajustament).

Arbre de Decisió

Possibles hiperparàmetres:

- **'criterion': ['gini', 'entropy']**: Aquest paràmetre controla la funció utilitzada per dividir els nodes de l'arbre. 'gini' és més eficient computacionalment, mentre que 'entropy' pot ser millor en situacions on les classes són més complexes.
- **'max_depth': [3, 5, 7, 10]**: El límit màxim de profunditat de l'arbre controla la complexitat del model. Un arbre massa profund pot sobreajustar-se, mentre que un arbre poc profund pot ser massa senzill. S'estableix la profunditat màxima del model a 10 per evitar overfitting.
- **'min_samples_split': [2, 5, 7, 10]**: Aquest paràmetre controla el nombre mínim de mostres requerides per dividir un node. Augmentar aquest valor pot fer l'arbre més general.
- **'min_samples_leaf': [2, 5, 10]**: Controla el nombre mínim de mostres que un node full ha de tenir. Un valor alt pot ajudar a evitar el sobreajustament, per aquest motiu s'ha inclòs aquest hiperparàmetre.

Hiperparàmetres òptims:

- **'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 10, 'min_samples_split': 2**

Anàlisi: Els resultats òptims indiquen que l'ús de l'entropia com a criteri de divisió és més efectiu per a les dades en qüestió, ja que pot manejar millor la complexitat de les classes. La profunditat màxima de 10 sembla ser una bona opció per evitar tant el sobreajustament com el subajustament. Els valors de 'min_samples_leaf' i 'min_samples_split' més alts poden ajudar a fer el model més robust, evitant la creació d'arbres excessivament complexos.

SVM (Support Vector Machine)

Possibles hiperparàmetres:

- **'C': [0.1, 1, 10]**: Aquest paràmetre controla el compromís entre maximitzar el marge i minimitzar l'error de classificació. Un valor més alt de C implica una penalització més gran per als errors de classificació.
- **'kernel': ['linear', 'rbf']**: El tipus de nucli determina la forma de la frontera de decisió. 'rbf' pot gestionar més bé les dades no lineals, mentre que 'linear' és més simple i adequat per a dades lineals.
- **'gamma': ['scale', 'auto']**: Controla la influència dels punts de dades en la creació de la frontera de decisió. 'scale' ajusta automàticament aquest valor, mentre que 'auto' utilitza un valor fix.

Hiperparàmetres òptims:

- **'C': 10, 'gamma': 'scale', 'kernel': 'rbf'**

Anàlisi: Els valors òptims suggereixen que un valor alt de **C (10)** és adequat per minimitzar els errors de classificació en aquest conjunt de dades. L'ús del nucli **'rbf'** (funció de base radial) indica que les dades no són lineals i requereixen una transformació per a una millor separabilitat. L'opció **'gamma'** en **'scale'** ajuda a ajustar de manera òptima la influència de cada punt de dades en la frontera de decisió.

3.4 Anàlisi general derivat dels hiperparàmetres obtinguts:

Els nostres conjunts de dades probablement presenten una combinació de relacions no lineals i estructures complexes entre les característiques i les classes. Els models seleccionats, com el SVM amb nucli 'rbf' i l'ús d'arbres de decisió profunds, indiquen que les relacions entre les variables no són senzilles i requereixen models capaços de captar aquesta complexitat. En general, sembla que les dades són suficientment variades per necessitar models robustos que no només s'ajustin bé, sinó que també siguin capaços de generalitzar correctament.

3.5 Entrenament dels models i validació

S'entrena cada model utilitzant els hiperparàmetres òptims seleccionats prèviament. A continuació, cada model es fa servir per ajustar-se a les dades d'entrenament *X_train* i *y_train*. Posteriorment, es separen les característiques i les etiquetes de validació del conjunt *val_df*. Els models realitzen prediccions tant per a les dades d'entrenament com per a les de validació, generant prediccions clàssiques (*train_pred*, *val_pred*) i probabilitats associades (*train_prob*, *val_prob*). A continuació, s'utilitzen dues funcions *evaluate_model* i *plot_confusion_matrix* per avaluar el rendiment dels models tant en el conjunt d'entrenament com en el de validació, mostrant les mètriques definides anteriorment per a cada conjunt. A continuació es veuen i s'interpreten els resultats obtinguts per a cada un d'ells

Cal recordar que busquem maximitzar la precision, concretament amb la que es prediu el valor 0 de la variable objectiu

KNN

Dades d'entrenament:

	Precision	Recall	F1-score
0	0,87	0,83	0,85
1	0,84	0,88	0,86

Accuracy: 0,85

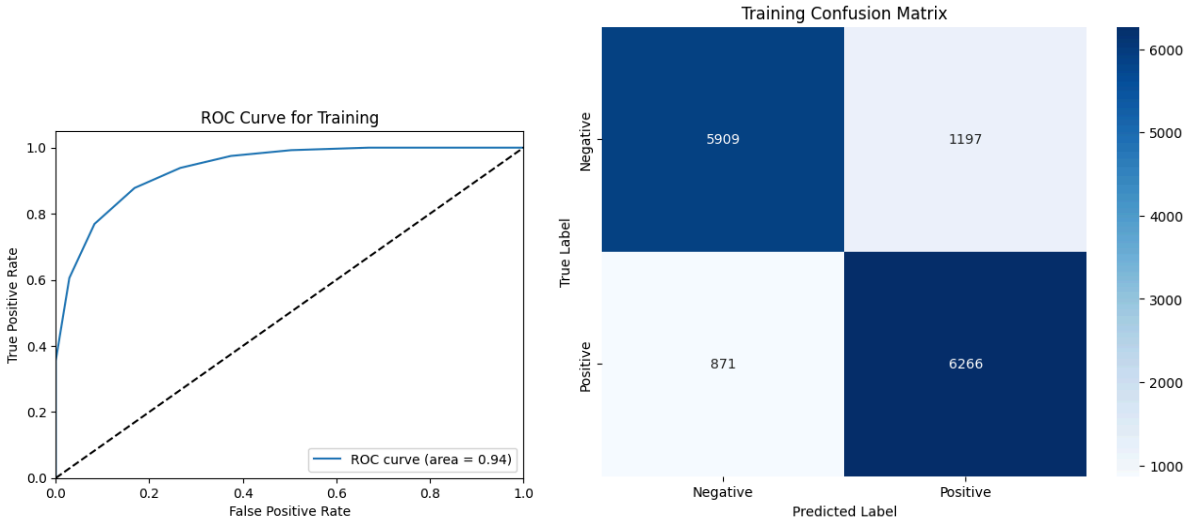

Figura 18: ROC curve i matriu de confusió pels resultats obtinguts amb les dades d'entrenament per KNN

Dades de validació:

	Precision	Recall	F1-score
0	0,83	0,79	0,81
1	0,81	0,85	0,83

Accuracy: 0,82

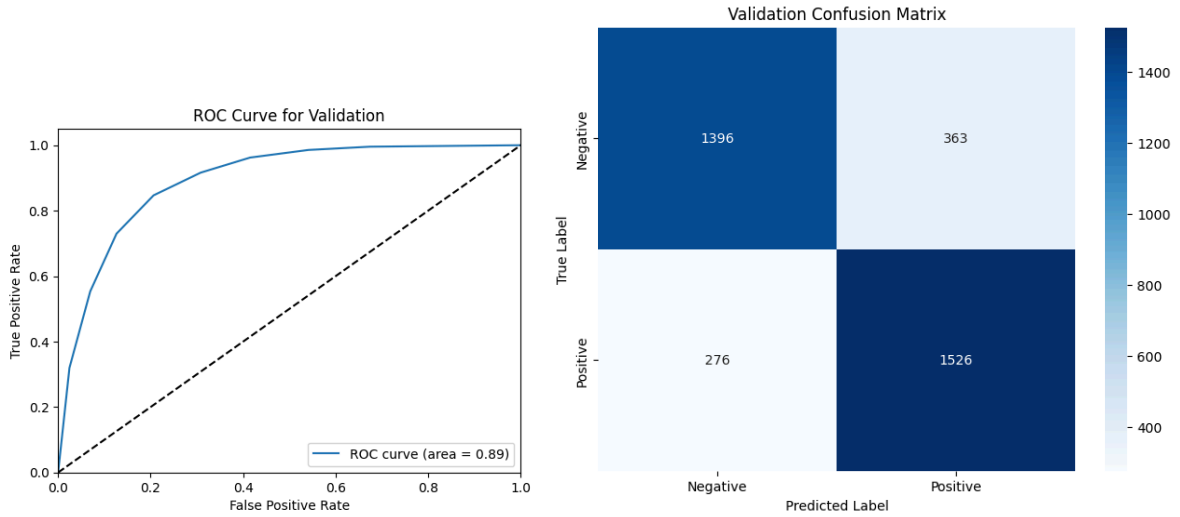

Figura 19: ROC curve i matriu de confusió pels resultats obtinguts amb les dades de validació per KNN

Anàlisi general: Els resultats obtinguts amb el model KNN mostren un bon rendiment, amb una **accuracy** de 0,85 en entrenament i 0,82 en validació, indicant que el model generalitza bé. La **precision** per la classe 0 és alta (0,87 en entrenament i 0,83 en validació), però la **recall** és una mica més baixa (0,83 en entrenament i 0,79 en validació), suggerint que el model perd algunes instàncies negatives. La lleugera disminució en les dades de validació podria indicar un lleuger **subajustament**, però no sembla haver-hi **sobreajustament** important, ja que la caiguda en les mètriques no és substancial. El model aconsegueix un bon equilibri entre **precision** i **recall** per la classe 0, amb espai per a millores en la identificació de totes les instàncies negatives.

Arbres de Decisió

Dades d'entrenament:

	Precision	**Recall**	**F1-score**
0	0,86	0,78	0,82
1	0,80	0,88	0,84

Accuracy: 0,83

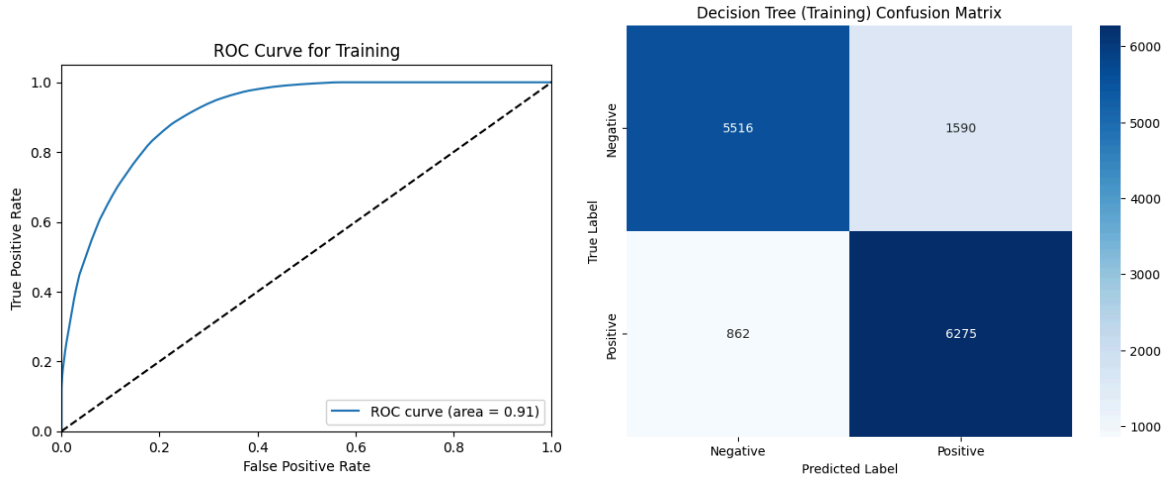

Figura 20: ROC curve i matriu de confusió pels resultats obtinguts amb les dades d'entrenament per DT

Dades de validació:

	Precision	**Recall**	**F1-score**
0	0,80	0,73	0,76
1	0,76	0,82	0,79

Accuracy: 0,78

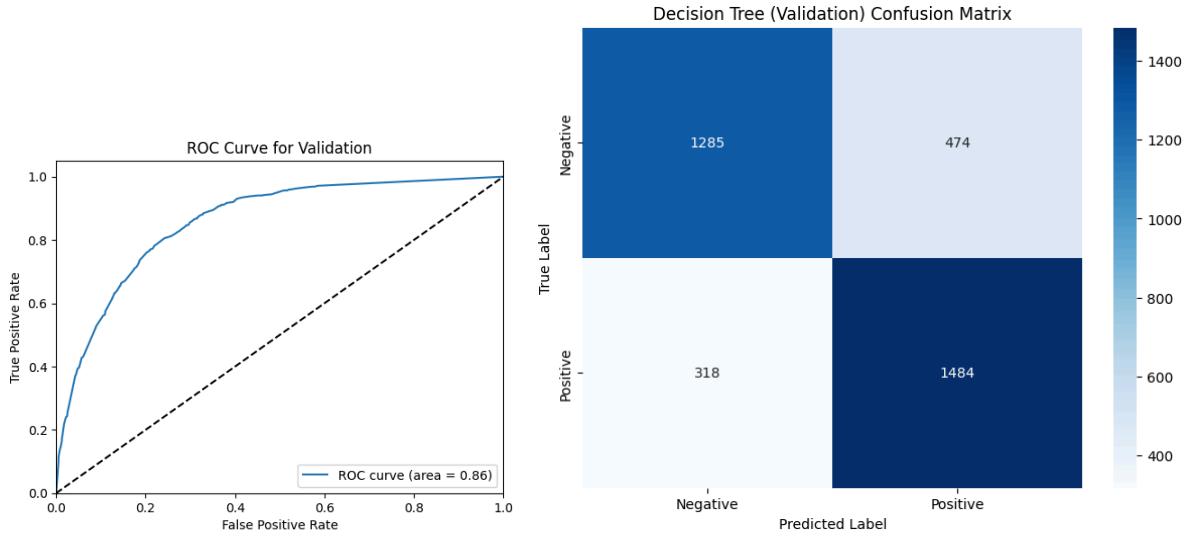

Figura 21: ROC curve i matriu de confusió pels resultats obtinguts amb les dades de validació per DT

Anàlisi general: Els resultats obtinguts amb el model d'**arbre de decisió** mostren un rendiment decent, amb una **accuracy** de 0,83 en entrenament i 0,78 en validació, indicant una certa caiguda en el rendiment a mesura que el model generalitza. La **precision** per la classe 0 (no real) és alta en entrenament (0,86), però disminueix a 0,80 en validació, mentre que la **recall** per la classe 0 és de 0,78 en entrenament i 0,73 en validació, mostrant una petita caiguda en la capacitat del model per identificar les instàncies negatives. La **precision** i **recall** per la classe 1 (real) són relativament equilibrades, amb una **F1-score** de 0,84 en entrenament i 0,79 en validació. Aquesta disminució en la **precision** i **recall** de la

classe 0 a la validació, juntament amb la lleugera caiguda en l'**accuracy**, pot indicar un cert **subajustament** del model. No sembla haver-hi **sobreajustament**, ja que les mètriques de validació no mostren una caiguda dramàtica. El model aconsegueix un bon compromís entre **precision** i **recall** per a la classe 1, però hi ha marge de millora en la identificació de la classe 0.

SVM

Dades d'entrenament:

	Precision	**Recall**	**F1-score**
0	0,88	0,79	0,84
1	0,81	0,89	0,85

Accuracy: 0,84

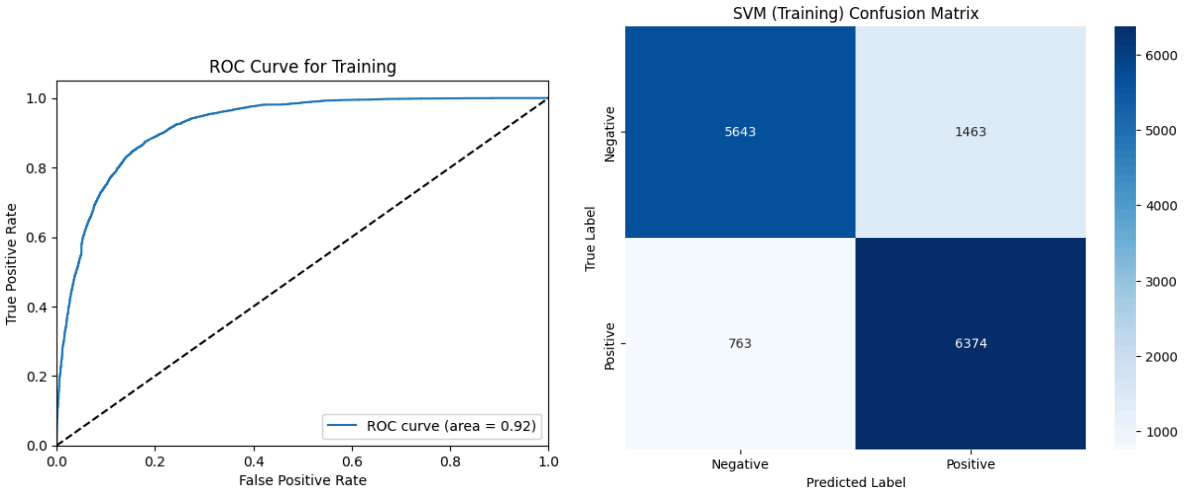

Figura 22: ROC curve i matriu de confusió pels resultats obtinguts amb les dades d'entrenament per SVM

Dades de validació:

	Precision	**Recall**	**F1-score**
0	0,88	0,80	0,84
1	0,82	0,89	0,86

Accuracy: 0,85

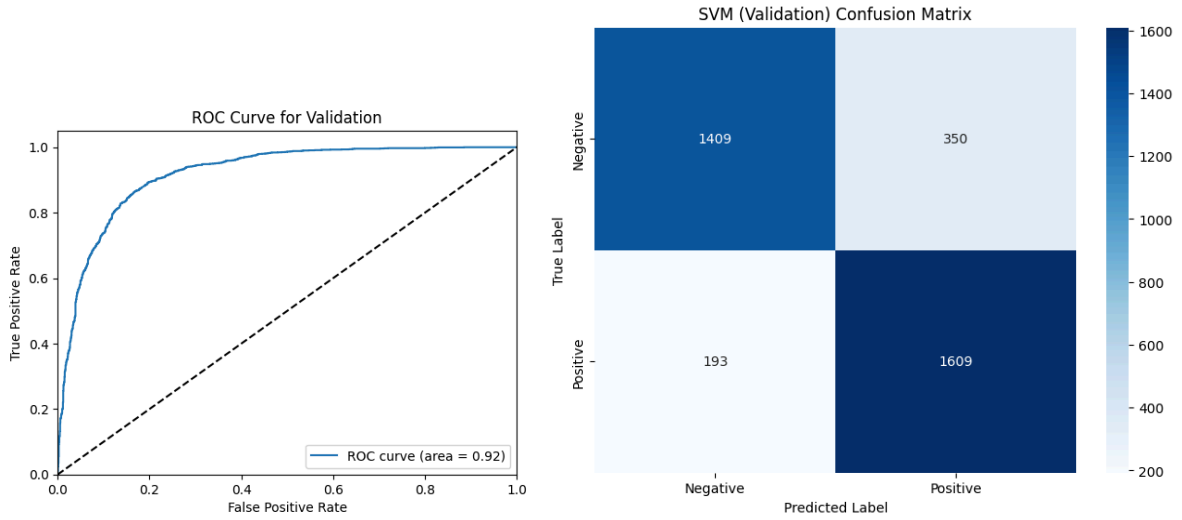

Figura 23: ROC curve i matriu de confusió pels resultats obtinguts amb les dades de validació per SVM

Anàlisi general: Els resultats obtinguts amb el model **SVM** mostren un bon rendiment, amb una **accuracy** de 0,84 en entrenament i 0,85 en validació, indicant que el model generalitza bé i no presenta **sobreajustament**. La **precision** per la classe 0 (no real) és alta (0,88 en ambdós conjunts), mentre que la **recall** per la classe 0 és lleugerament més baixa (0,79 en entrenament i 0,80 en validació), suggerint que el model pot perdre algunes instàncies negatives. Per la classe 1 (real), la **precision** i **recall** són força equilibrades, amb un **F1-score** de 0,85 en entrenament i 0,86 en validació, mostrant que el model és eficaç a identificar les instàncies positives. La lleugera caiguda en la **precision** per la classe 0 a la validació, però l'alta **accuracy** i el bon **F1-score** global, suggereixen que el model està ben ajustat i generalitza de manera eficient. En general, el model SVM aconsegueix un bon compromís entre **precision** i **recall**, amb un rendiment consistent en ambdós conjunts de dades.

4. Selecció del model

El model **SVM** s'escull com a model final per diversos motius:

1. **Rendiment consistent:** Els resultats obtinguts amb el SVM mostren una **accuracy** elevada tant en les dades d'entrenament (0,84) com en les de validació (0,85), indicant que el model generalitza bé sense signes clars de **sobreajustament** o **subajustament**.
2. **Bona precision i recall per la classe 0 (no real):** El model SVM aconsegueix una **precision** de 0,88 per la classe 0 tant en entrenament com en validació, destacant-se en identificar correctament les instàncies negatives, que és un dels objectius principals. La **recall** també es manté elevada, amb 0,79 en entrenament i 0,80 en validació, assegurant que el model no passa per alt moltes instàncies negatives.
3. **Equilibri en la classe 1 (real):** Per a la classe 1, el model SVM també mostra un bon rendiment amb **precision** i **recall** força equilibrades (0,81 i 0,89 en entrenament, 0,82 i 0,89 en validació), aconseguint un **F1-score** elevat (0,85 en entrenament i

0,86 en validació), la qual cosa reflecteix una bona identificació de les instàncies positives.

4. **Resultats en el conjunt de validació:** El SVM manté un alt rendiment en el conjunt de validació, amb una **accuracy** de 0,85 i un **F1-score** per a la classe 0 i la classe 1 que continua sent força alt, demostrant que el model és capaç de generalitzar bé a dades no vistes.
5. **Visualització dels resultats:** La **ROC curve** i la **matriu de confusió** mostren un bon comportament global del model, amb poques falses positives i falses negatives, indicant que el model té un bon control sobre les prediccions, especialment per a la classe 0.

En conjunt, l'SVM és seleccionat per ser un model robust, que aconsegueix un bon equilibri entre **precision** i **recall**, especialment per la classe 0, i que generalitza bé a dades noves, sent adequat per l'objectiu de maximitzar la identificació de la classe negativa. A més a més, es pot suportar la desició del model escollit amb el següent gràfic de comparació de les corves ROC pels tres models en el conjunt de validació:

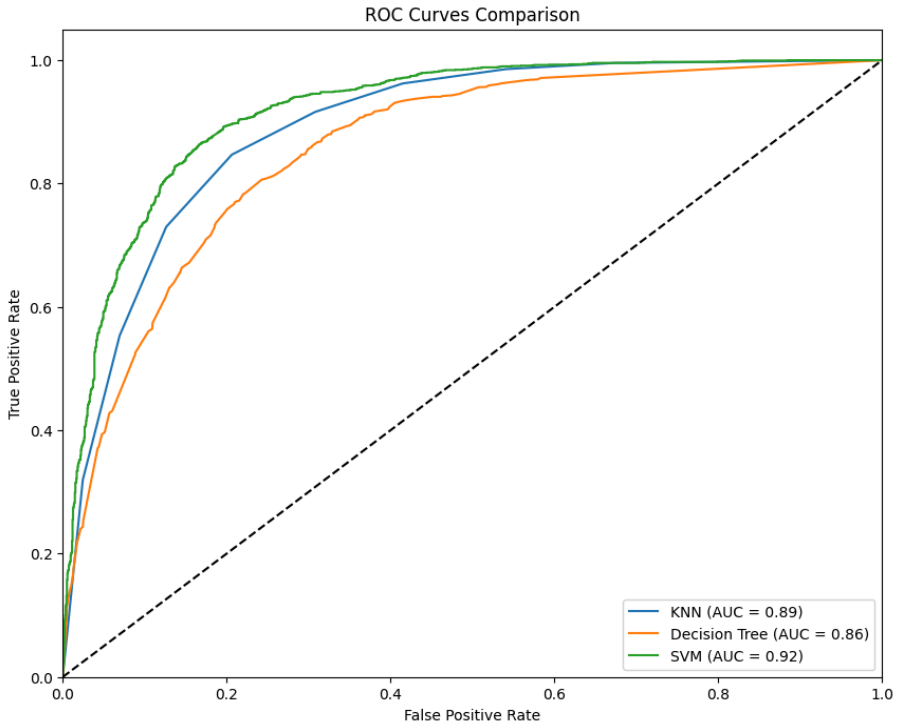

Figura 24: Comparació de Corves ROC pels 3 models en el conjunt de validació

A continuació hem procedit a provar el model amb la partició de test i els resutats han estat els següents:

	Precision	**Recall**	**F1-score**
0	0,78	0,68	0,73
1	0,72	0,82	0,77

Accuracy: 0,85

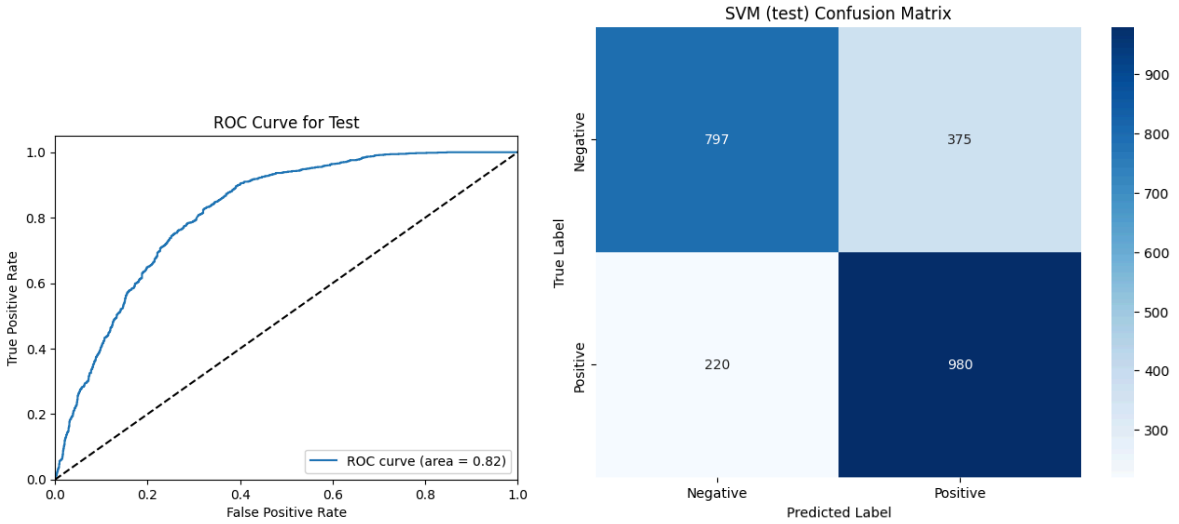

Figura 25: ROC curve i matriu de confusió pels resultats obtinguts amb les dades de test en SVM

Anàlisi: Els resultats obtinguts pel model **SVM** en el conjunt de test mostren un **accuracy** de 0,85, el que indica que el model manté una bona capacitat de generalització. Tanmateix, observem una certa disminució en les mètriques de **precision** i **recall** per la classe 0 (no real) en comparació amb les dades d'entrenament i validació. La **precision** per la classe 0 és de 0,78 i la **recall** de 0,68, la qual cosa suggereix que el model té certes dificultats per identificar correctament totes les instàncies negatives, perdent algunes d'elles (baix **recall**). Per la classe 1 (real), la **precision** és de 0,72 i la **recall** de 0,82, amb un **F1-score** de 0,77, indicant que el model segueix sent més eficaç per identificar les instàncies positives, tot i que amb una lleugera disminució en **precision** respecte a les dades d'entrenament.

Aquest anàlisi condueix a pensar que si haguéssim buscat maximitzar una altre mètrica com F1-score potser ens hagués quedat un model que s'adaptaria millor al conjunt de test ja que no és tan discriminador i possiblement no ens haguéssim trobat amb aquest sobreajustament moderat final.

5. Model Card

Nom del Model: Support Vector Machine (SVM)

Data de Creació: 28/12/2024

Creador: Pau Escobar Asensio

Descripció del Model: Aquest model SVM s'ha seleccionat per a la tasca de detecció de parla falsa, tenint en compte el seu rendiment consistent i la capacitat de generalitzar bé. El model utilitza un nucli "rbf" per adaptar-se a les relacions no lineals entre les característiques i les classes, i un paràmetre C alt per penalitzar els errors de classificació. L'objectiu principal és maximitzar la precisió (precision) per a la classe negativa (0), reduint els falsos positius, i mantenir un bon equilibri entre la precisió i recall per a la classe positiva (1).

Entrenament:

- **Dades d'Entrenament:** Les dades d'entrenament es van utilitzar per ajustar el model, amb l'objectiu de maximitzar la precisió de la classe 0 (parla no real).
- **Hipòtesis i Hiperparàmetres:** El model es va entrenar amb els següents hiperparàmetres seleccionats:
    - **C:** 10
    - **Gamma:** 'scale'
    - **Kernel:** 'rbf'

Mètriques de Rendiment:

- **Precisió (Precision):**
    - Classe 0: 0,88 (entrenament) / 0,88 (validació)
    - Classe 1: 0,81 (entrenament) / 0,82 (validació)
- **Recall:**
    - Classe 0: 0,79 (entrenament) / 0,80 (validació)
    - Classe 1: 0,89 (entrenament) / 0,89 (validació)
- **F1-Score:**
    - Classe 0: 0,84 (entrenament) / 0,84 (validació)
    - Classe 1: 0,85 (entrenament) / 0,86 (validació)
- **Accuracy:**
    - 0,84 (entrenament) / 0,85 (validació)

Avaluació de Rendiment: Els resultats obtinguts amb el model SVM mostren un rendiment robust amb una alta precisió per a la classe negativa (0), que és essencial per a la tasca de detecció de parla falsa. La capacitat del model per identificar instàncies positives (classe 1) també és bona, amb un F1-Score equilibrat. La caiguda lleugera en les mètriques de precisió i recall en el conjunt de test no implica un gran desajust, ja que el model continua generalitzant bé, amb una **accuracy** de 0,85 en el conjunt de test.

Mètrica a Maximitzar: Es va triar maximitzar la **precision** per a la classe 0 (parla no real), ja que el cost dels falsos positius és especialment important en aquest context. Això assegura que el model minimitza els errors en la classificació d'audios com a reals quan no ho són.

Anàlisi de Resultats: El model SVM aconsegueix un bon equilibri entre **precision** i **recall**, especialment per la classe negativa (0), amb una **precision** elevada (0,88) i una **recall** alta (0,79 a 0,80). Els resultats obtinguts en el conjunt de validació són consistents, la qual cosa mostra que el model no està sobreajustat i generalitza bé.

Limitacions: Encara que els resultats són molt bons, hi ha una lleugera disminució en les mètriques de **precision** i **recall** per la classe 0 en el conjunt de test. Això indica que el model pot tenir dificultats per identificar correctament algunes instàncies negatives, tot i que la **accuracy** global del model es manté elevada.

Selecció del Model: S'ha seleccionat el model **SVM** com a model final per la seva capacitat per generalitzar bé a dades no vistes, el bon equilibri entre les classes, i la seva robustesa en la detecció de parla no real (classe 0). La seva consistència en el conjunt de validació el fa una bona elecció per aquest problema.

Comparativa entre Models: El model SVM ha superat altres models com KNN i arbres de decisió en termes de **precision** per la classe 0 i en la capacitat de generalitzar. La comparació de les corbes

ROC entre els models en el conjunt de validació mostra que l'SVM és el model amb el millor rendiment global.

Conclusió: L'SVM s'ha seleccionat com a model final per a la detecció de parla falsa, ja que aconsegueix un bon compromís entre **precision** i **recall**, maximitzant la identificació de la classe negativa (parla no real) sense perdre eficiència en la classe positiva (parla real).

6. Bonus 1

Per tal d'implementar el EBM hem seguit els mateixos passos que amb els altres models:

Possibles hiperparàmetres:

- 'learning_rate': [0.01, 0.1]
- 'max_bins': [128]

Hiperparàmetres òptims:

- 'learning_rate': 0.01, 'max_bins': 128

Anàlisi: Learning rate (0.01): Controla la velocitat d'aprenentatge. Un valor baix (0.01) implica un aprenentatge més lent però estable. Max bins (128): Defineix el nombre de intervals per discretitzar les dades contínues. Un valor alt (128) permet una representació més detallada. Els valors obtinguts 'learning_rate': 0.01 i 'max_bins': 128 indiquen que el millor rendiment es va aconseguir amb un aprenentatge lent i una discretització precisa de les dades.

Resultats entrenament:

	Precision	**Recall**	**F1-score**
0	0,84	0,79	0,82
1	0,8	0,85	0,83

Accuracy: 0,82

Resultats validació per test:

	Precision	**Recall**	**F1-score**
0	0,75	0,68	0,72
1	0,72	0,78	0,74

Accuracy: 0,73

Conclusions: El model EBM mostra un cert **overfitting**, ja que els resultats d'entrenament són significativament millors que els de validació. Mentre que en entrenament, l'**accuracy** és del 82% amb uns valors de **precision** i **F1-score** bastant equilibrats per a les dues classes, en la validació per test es produeix una caiguda en el rendiment, especialment per la classe 0. La **precision** per la classe 0 baixa a 0,75 (enfront de 0,84 en entrenament), i el **recall** també disminueix, passant de 0,79 a 0,68. Això indica que el model és més sensible a la classe 1 durant la validació i tendeix a identificar menys bé els casos de la classe 0, probablement per la seva menor representació en el conjunt de test o per una tendència a generalitzar millor la classe més predominant. El **overfitting** podria estar causat per una complexitat excessiva del model o una manca de regularització, que fa que el model s'adapti massa als detalls del conjunt d'entrenament, afectant negativament la seva capacitat de generalitzar als nous dades.

7. Experimentació

7.1 Normalitzar abans o després de tractar els outliers?

Aquests han estat els resultats d'avaluar el model final havent tractat els outliers abans de normalitzar les variables:

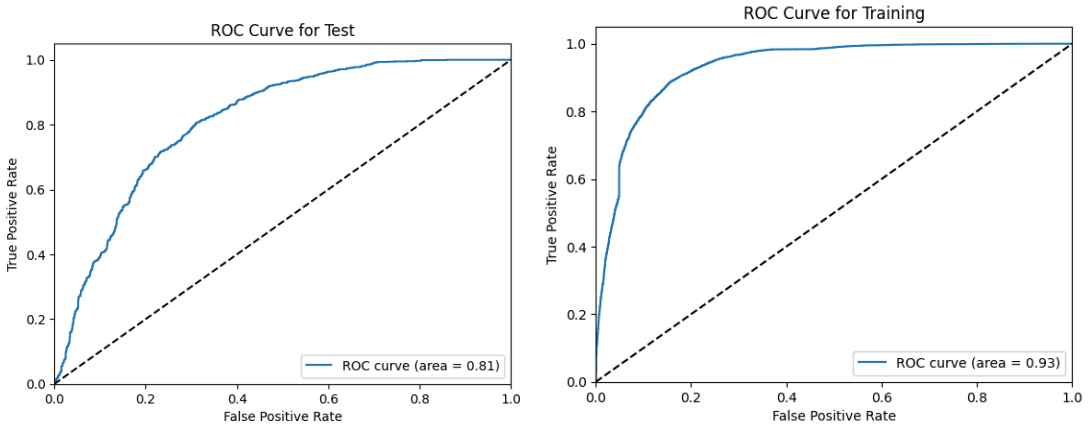

Figura : ROC curves del training i del validation pel model normalitzat després de tractar outliers

Com podem observar a les curves ROC, pel training tenim una AUC (àrea per sota de la curva) de 0,93 mentre que al validation és de 0,81. Aquest fet suggereix que el model té overfitting, i per tant, afirmem una de les premises que consideravem a l'apartat 1.5 del report. Així doncs, finalment es farà la normalització abans del tractament dels outliers

7.2 Reduir dimensionalitat del model?

A continuació s'observen els resultats abans i després d'aplicar la reducció de dimensionalitat als 3 models:

1. **KNN:**
    - **Abans de PCA:** F1-score global de 0.85 amb precisió mitjana de 0.79. El model generalitza bé.

    - **Després de PCA:** F1-score global de 0.77 amb precisió mitjana de 0.73. Hi ha pèrdua de capacitat predictiva i discriminativa.
    - **Anàlisi:** KNN depèn molt de l'estructura original; reduir dimensions simplifica en excés l'espai i afecta el rendiment.
2. **Arbre de decisió:**
    - **Abans de PCA:** F1-score global de 0.78 amb precisió mitjana de 0.73. Bon equilibri entre ajust i generalització.
    - **Després de PCA:** F1-score global de 0.74 amb precisió mitjana de 0.70. Reducció moderada en el rendiment.
    - **Anàlisi:** Els arbres són robustos, però PCA elimina característiques clau que ajuden a discriminar classes.
3. **SVM:**
    - **Abans de PCA:** F1-score global de 0.78 amb precisió mitjana de 0.71. Bon rendiment amb dependència dels paràmetres.
    - **Després de PCA:** F1-score global de 0.77 amb precisió mitjana de 0.71. Canvi mínim en el rendiment.
    - **Anàlisi:** SVM és robust a PCA, ja que les components principals retenen informació discriminativa suficient.

Conclusió General:

- **KNN:** PCA afecta significativament el rendiment; cal evitar-lo amb aquest model.
- **Arbre de Decisió:** Impacte moderat, però PCA pot eliminar informació clau.
- **SVM:** PCA té un impacte mínim; el model manté el rendiment gairebé intacte.

Per tant, no aplicarem reducció de dimensionalitat al nostre model

8. Conclusions

El treball ha aconseguit desenvolupar amb èxit un model de detecció de parla falsa utilitzant tècniques d'aprenentatge automàtic. Les conclusions principals són:

1. El model SVM seleccionat ha demostrat ser el més eficaç, amb una precisió del 88% en la detecció de parla falsa i una accuracy global del 85% en el conjunt de validació.
2. La decisió de normalitzar les variables abans de tractar els outliers ha estat crucial per evitar el sobreajustament del model.
3. S'ha demostrat que la reducció de dimensionalitat mitjançant PCA no era necessària ni beneficiosa pel rendiment del model final, ja que les variables originals contenien informació rellevant per a la classificació.
4. El preprocessament acurat de les dades, incloent la gestió d'outliers mitjançant KNN i la normalització adequada de les variables, ha estat fonamental per aconseguir aquests bons resultats.
5. Tot i els bons resultats generals, s'ha observat una lleugera disminució en el rendiment amb el conjunt de validació i de test, suggerint que encara hi ha marge de millora en la generalització del model.