Pràctica 2: Classificació

Aprenentatge Computacional (102787) - Grau en Enginyeria Informàtica [MO52745]

Enrique Gómez Becerra - 1566725

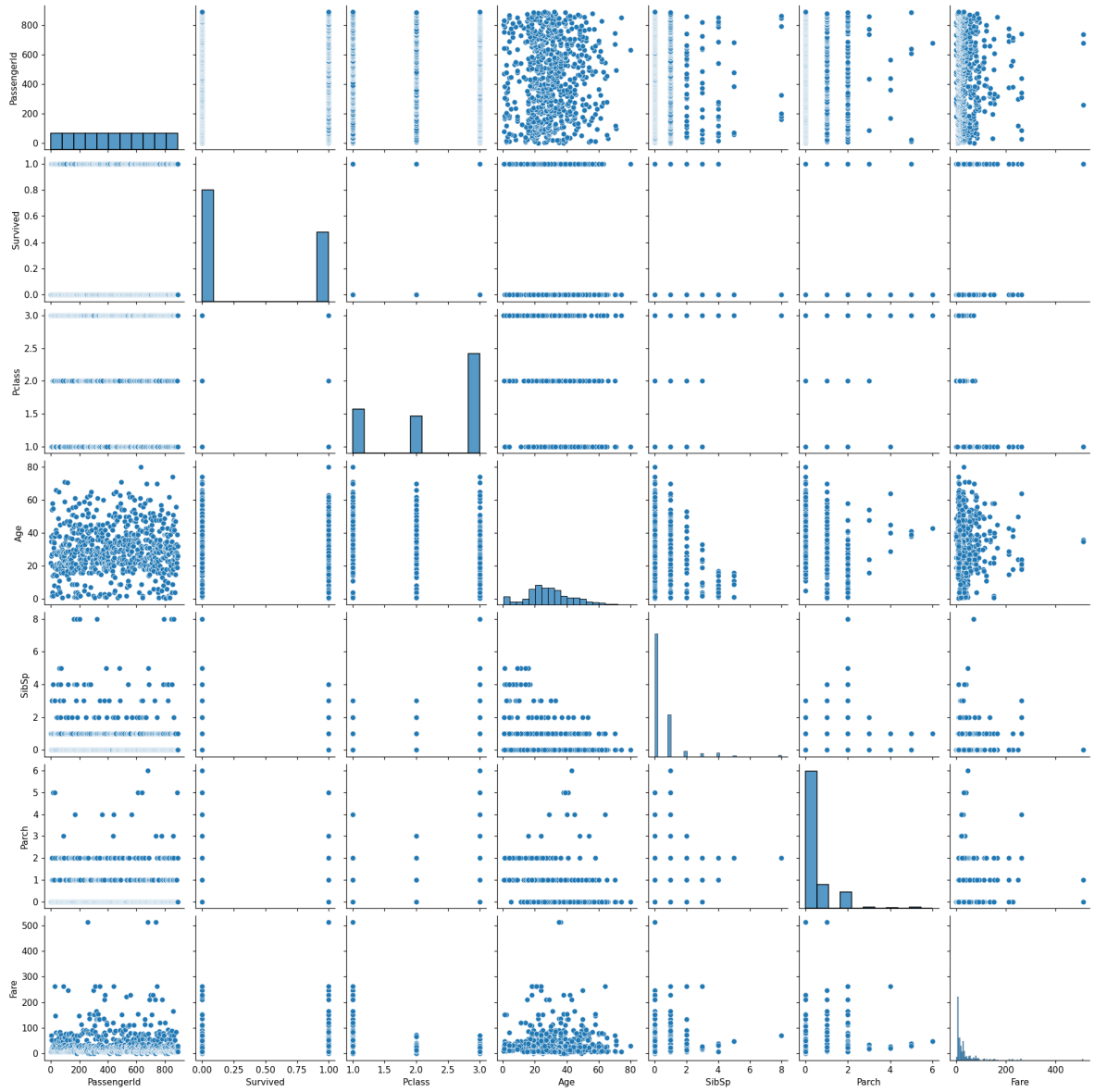
Borja Arcos Comas - 1568307

Javier Méndez Leiva - 1496052

Index

Introducció

En aquest document es detallen les tècniques emprades per la classificació de les dades obtingudes en l'estudi del gran desastre que va suposar l'accident del Titanic. Ens hem basat en les dades proporcionades pels professors del següent enllaç: Titanic - Machine Learning from Disaster | Kaggle. Utilitzarem com a pedra angular l'atribut survived per fer la classificació en el model, utilitzant altres paràmetres com sexe, la quantitat pagada per l'entrada o la edad.



Altres dades com poden ser el nom, la cabina on s'hospedaven o l'id en la base de dades són atributs que no es tindran en compte a l'hora de fer la classificació.

Objectius

Amb aquest projecte el que es busca és posar en pràctica els conceptes estudiats a classe i aprendre a aplicar models de classificació sobre dades reals. A més, serà necessari assolir el coneixement necessari per aplicar les tècniques de classificació i anàlisi necessàries per poder treure els resultats desitjats, tot amb la búsqueda d'informació per internet i el nostre propi criteri. Per acabar, volem treure conclusions pròpies de l'estudi realitzat a partir del coneixement adquirit mentre es realitza la pràctica.

Descripció de la base de dades

Dintre de la base de dades podem veure tres tipus de valors.

Per una banda, els que descriuen directament a la persona, com pot ser el sexe o l'edat.

Per altre, podem veure descripcions indirectes com el que van pagar pel ticket (això ens ajuda a tenir una estimació de la posició económica de la persona en qüestió).

Finalment tenim com va ser el seu desenllaç en el Titanic. Aquest atribut amb nom Survived podrà ser 0 si no van sobreviure o 1 si van sobreviure.

Per fer una classificació que ens ha interessat utilitzarem aquest últim valor descrit, ja que arran d'altres atributs com el sexe o el que van pagar pel ticket podrem fer una classificació acurada.

A l'hora de suprimir altres dades vam trobar que, a la nostre base de dades, hi havien certs valors NaN a la columna de l'edat. Per poder fer una classificació correcte, hem decidit calcular la mitja de les edats, substituint aquests valors NaN per aquesta mitja.

```
Per comptar el nombre de valors no existents:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Com les variables de sexe estan en text caldrà normalitzar-les i fer una conversió a numeric. Sera un tipus binari (0 o 1).

Preprocessat

Un cop hem agafat les dades de la base de dades veiem que hi han dades que necessiten un processament previ per ser utilitzades. Per aquestes dades és necessari fer una codificació als atributs *Sex* i *Embarked*, és a dir, convertir-los d'un element de texte a un valor numèric discret. Per fer-ho utilitzem *LabelEncoder()* de la llibreria *sklearn* sobre els atributs esmentats i obtenim una columna binària per a *Sex* i una columna amb els valors 0, 1 i 2 per a *Embarked*.

També hi han valors buits, llavors cal fer un reemplaçament de les celes d'aquests per valors coherents. En el nostre cas hi han valors *nan* que substituïm per la mitjana de la resta de valors del l'atribut, per així evitar una influència d'aquestes dades a la classificació. Fent ús de la funció *SimpleImputer()* de *sklearn* omplim els buits tal com hem dit.

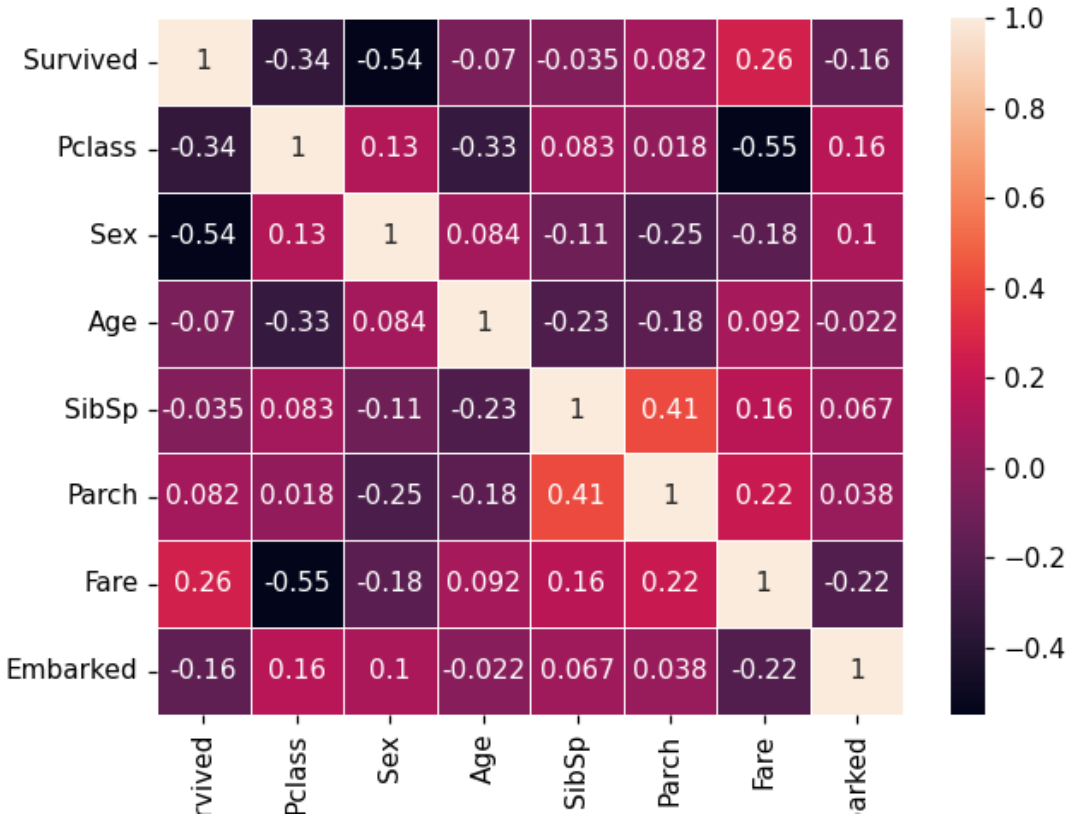
En el nostre cas no creiem que calgui aplicar PCA, ja que les nostres dades són en gran part discretes i categòriques. A més, el PCA assumeix una correlació lineal entre les dades i, com hem vist al *pairplot*, el atributs de la base de dades no comparteixen correlacions lineals. Per tant, l'aplicació del PCA no portaria cap benefici significatiu.

També és possible aplicar PolynomialFeatures per millorar el model en casos en que les dades es relacionen de forma no-lineal. En el nostre cas, si observem el *pairplot*, no observem cap mena de relació no-lineal entre les dades i, per tant, no aportaria gaire millora al model.

Selecció de models

Durant la selecció de models s'han considerat, nearest neighbours, SVM amb diferents kernels, logistic regression i el perceptró.

En base a les següents correlacions observades podem concluir les següents afirmacions.



L'algorisme més precís, probablement, serà SVM amb diferents kernels degut a l'ús no tan sols del sexe sinó també de la classe dels passatgers per determinar la classificació, nearest neighbours en trobar-se amb una correlació tan dolenta,

encara que sense dubte sería el més ràpid, troba la seva fiabilitat molt menguada, per altra banda el perceptró mancaria de una major disponibilitat de capes de dades, on precisament SVM al ser unes dades tan limitades es on precisament més pot aportar. Finalment, s'ha de dir que regressió logística és un algorisme que sota correlacions tan baixes perd enormement efectivitat sobre la classificació a fer.

Finalment, degut a la poca correlació de les nostres dades ens sembla que l'ensemble no es una bona idea, ja que la nostre predicció, pressuposem, no serà gaire encertada. La unió de moltes prediccions dolentes amb diferents seleccions, que també poden estar malament seleccionades, arribarà a ser una predicció encara menys fiable. Els desavantatges, a part dels ja mencionats, serien:

- Ensembling és menys interpretable, la sortida del model d'ensembled és difícil de predir i explicar.
- Ensembling és costos en termes d'espai i temps.

Crossvalidation

La crossvalidation és un mètode útil perquè ens permet utilitzar tot el dataset com a training i així no separar-ho en tres parts(train, validation, test). També ens permet saber quina part del nostre dataset és més eficaç a l'hora de fer prediccions, ja que fem validació de tot el dataset i podem veure la seva accuracy, fent així una mitja de tots. En el nostre cas, com tenim poques dades ens resulta de molta utilitat, ja que tenir 3 datasets ens faria la feina molt difícil, i acabaria sent un model poc fiable degut a les poques dades que tindriem. Amb crossvalidation podem agafar la millor part del nostre (petit) dataset i fer una predicció més fiable.

Per altre banda LeaveOneOut seria una bona opció a contemplar en el nostre cas, ja que el numero de dades son poques. Tot i això, creiem que és més convenient el crossvalidation, ja que considerem que la quantitat que tenim de dades és suficient.

L'accuracy score és utilitzat, sobretot, quan la distribució de classes és similar i els casos de vertader positiu i verdader negatiu son més important, que això és el que més ens importa a l'hora de fer el nostre estudi. Per tant, hem decidit escollir

aquesta mètrica per fer l'entrenament.

```
0.79 accuracy with a standard deviation of 0.02
0.71 f1-score with a standard deviation of 0.03
0.63 average-precision with a standard deviation of 0.03
```

Per veure les principals mètriques de classificació hem utilitzat classification_report per així poder escollir quina ens va millor. En el nostre cas ens fixem en la métrica accuracy, perquè com hem dit abans és la més interessant en el nostre cas i per tant farem la optimització de la classificació.

Hyperparameter search

A l'hora d'escollir l'hiperparàmetre, buscarem el que sigui òptim. Existeixen dos tipus de cerca basiques, gridSearch i randomizedSearch. El gridSearch buscarà totes les combinacions possibles amb els valors dels hyperparàmetres i retornarà la combinació amb l'*score* més elevat, és a dir, aquella combinació que obté millors resultats de precisió i recall. Aquesta característica el converteix en una técnica computacionalment molt costosa, al contrari que el randomized que fa una cerca randomitzada dels paràmetres per trobar la millor combinació que trobi en un número d'intents. Els dos tipus de cerques ens poden fer bé el treball, ja que disposem de bastant temps actualment per la cerca dels hyperparameters, però tampoc necessitem un resultat òptim, ja que aquest estudi no serà pel benefici de cap persona o empresa. Si tinguéssim un temps limitat optariem per el randomizedSearch, mentre que amb uns recursos computacionals limitats fariem ús de gridSearch.

Existeixen molts altres mètodes de cerca que no són necessàriament més eficients, però ens permeten tenir un control més exhaustiu dels recursos que volem destinar a aquesta cerca. Alguns exemples podrien ser BayesianSerach o els Halving Methods.

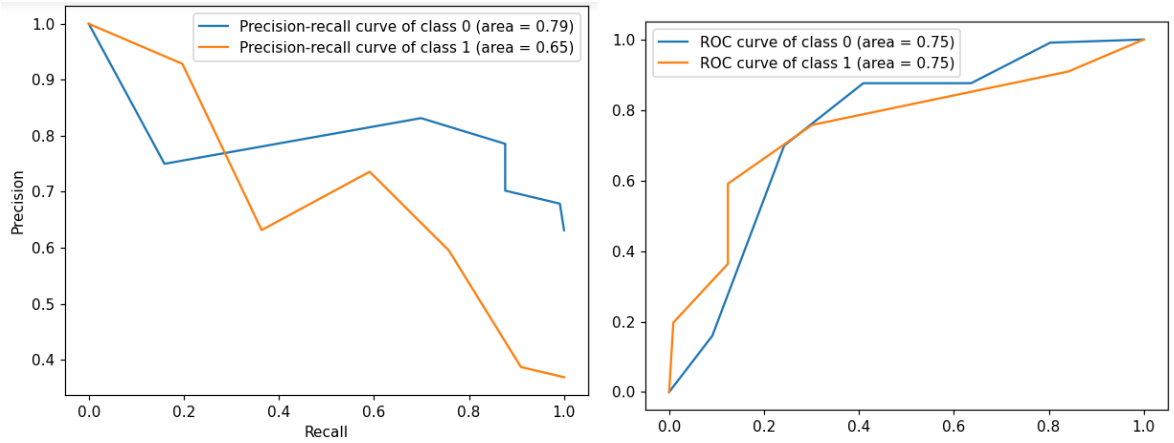
Anàlisi apartat A

A la primera secció de l'apartat a observem la comparativa entre un model de regressió logística i un SVC. Es mostren els resultats obtinguts fent ús del 50%, 70% i 80% del dataset com a conjunt de dades d'entrenament.

```
Correct classification Logistic  0.5 % of the data:  0.7959641255605381
Correct classification SVM       0.5 % of the data:  0.7713004484304933
Correct classification Logistic  0.7 % of the data:  0.8022388059701493
Correct classification SVM       0.7 % of the data:  0.8022388059701493
Correct classification Logistic  0.8 % of the data:  0.770949720670391
Correct classification SVM       0.8 % of the data:  0.770949720670391
```

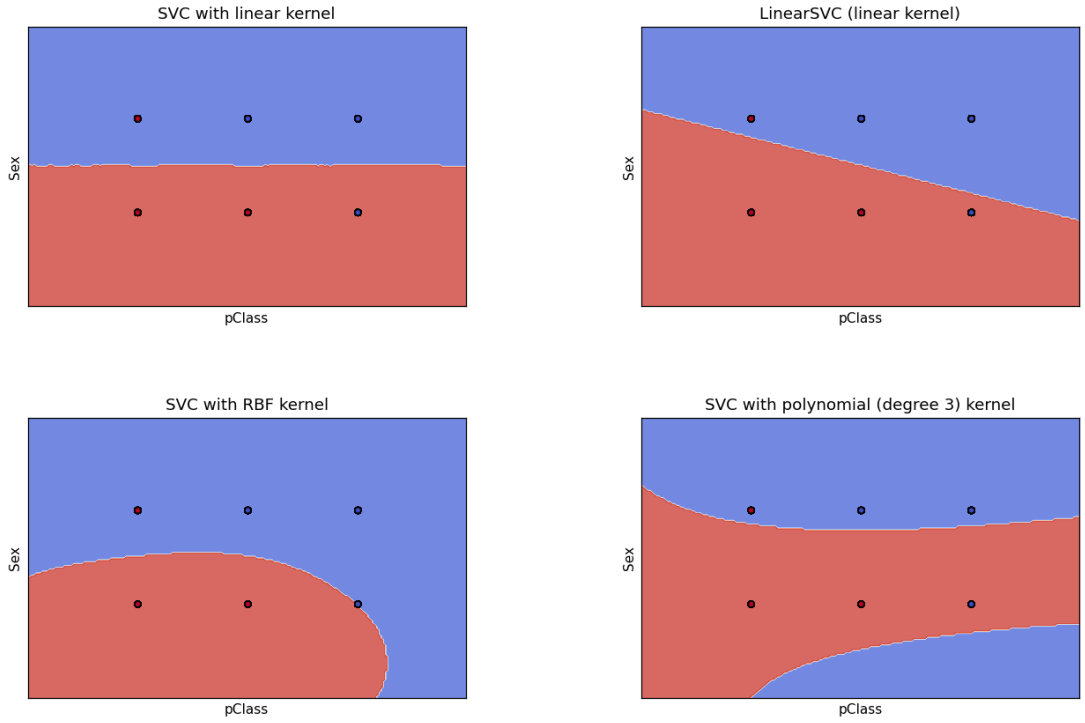
Observem que en ambdós casos la puntuació máxima s'obté destinant el 70% de les dades a l'entrenament del model i el 30% restant al testejat.

A la secció següent obtenim 2 figures referents a les corbes precision-recall i ROC.



A partir de les corbes de precision-recall podem dir que els nostre model obté una precisió i recall de més qualitat en les prediccions sobre la classe 0, en aquest cas referent a les persones que no sobreviuen (*Survived* = 0). Tot i així, si observem les corbes de ROC, aquesta diferència en la precisió de les prediccions no és produ significativa ja que obtenim un rendiment idèntic en les prediccions d'ambdues classes.

Per acabar, a la darrera secció obtenim una sèrie de gràfiques indicant els classificadors definits per 4 models diferents sobre el nostre dataset.



En el cas de les nostres dades, la interpretació d'aquestes gràfiques és més complicada, ja que treballem amb dades categòriques per obtenir les prediccions. Cada punt representa un subjecte del dataset, encara que amb les nostres dades tots aquests punts es concentren en 6 punts diferenciats. Amb la visualització d'aquests gràfics no podem deduir quin dels models és més encertat, ja que els colors assignats als punts no representen realment la seva classe degut a que es troben superposats entre ells i només veiem l'últim representat a cada grup.

Conclusions

A arrel del model de classificació fet podem dir que em comprengut les diferents condicionants que poden facilitar i ampliar l'eficiència d'un clasificador. Per altra banda, també em obtingut detall en diferents indicadors i medidors que poden impactar el nostre punt de vista a l'hora d'analitzar i conèixer els paràmetres més adequats per acabar d'optimitzar i puntualitzar el nostre model.