TAA - Rentrega 1

Sébastien EYSSERIC

May 2023

1 Taller 1 - Titanic

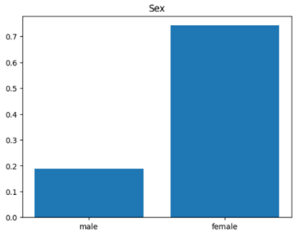
1. Los atributos ordinales: PassengerId, Name, Ticket
Los atributos numéricos: Age, Fare, Sibsp, Parch
Los atributos categóricos: Sex, Cabin, Embarked, Survived, Pclass

PassengerId, Name, Ticket no van a permitir de predictir si una persona sobreviva, sino que permite de identificar cada persona es porque son considerados como atributos ordinales.
Age y Fare son atributos continuos y de tipos floats es porque son considerados como atributos numéricos.
Sibsp, Parch son attributos discretos pero que no tiene un maximo pues no hay un quantidad definitivas de valores que se puede tomar estos attributos es porque son considerados como attributos numéricos.
Sex, Cabin, Embarked, Survived, Pclass, pueden tomar un nombre finito y predeterminado de valores es porque son considerados como atributos categóricos.
Survival puede ser 0 o 1, Pclass puede ser 1, 2 o 3, sex puede ser 'male' o 'female' y en fin embarked puede ser C, Q, S.

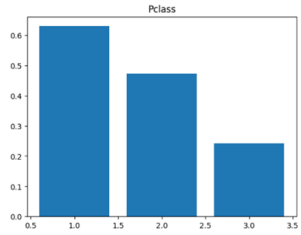
2. Hay una correlacion importante entre Sex y Survived, o entre Pclass y Survived :

Pclass	-0.338481
Age	-0.077221
SibSp	-0.035322
PassengerId	-0.005007
Parch	0.081629
Embarked	0.169718
Fare	0.257307
Sex	0.543351
Survived	1.000000

(a) Correlación con survived de cada atributo



(b) Proporción de sobrevivientes por cada género



(c) Proporción de sobrevivientes por cada PClass

Podemos ver que un hombre tiene 19% de sobrevivencia en comparación a una mujer que tiene 74%.
También podemos ver la correlación con Pclass en el ultimó grafo. En efecto parece que tenía más suerte de sobre vivir en primera clase que en segunda y tercera, y también más suerte de sobre vivir cuando está en segundo que en tercero.

3.

Table 1: **Comparación de los pipelines**

Pipeline	1 (Parte 4)	2 (Parte 5)	3 (Parte 8)
Accuracy con 5 CV	0.794	0.787	0.801

Para comparar estos diferentes pipelines utilizamos cross validation que consiste en dividir los datos en 5 "folds" en este caso y de entrenar y evaluar el modelo 5 veces, utilizando cada fold como conjunto de "validation" en una iteración y el resto como conjunto de "train". Para medir el error utilizamos "accuracy" que representa la proporción de predicciones correctas realizadas por el modelo en relación con el total de predicciones.

4.

Table 2: **Pipeline 1 : Pclass como dato**

Pipeline	numérico	ordinal	categórico
Accuracy con 5 CV	0.787	0.788	0.794

La Pipeline 1 se emperora si cambiamos el atributo Pclass de tipo.
Si hacemos la misma comparación de tipo sobre Embarked y Sex podemos ver que también no mejorar la Pipeline 1 y que debemos considerar Embarked y Sex como atributos categoricos.

Table 3: **Pipeline 1 mejorada con atributos escalado**

Scale	Ningun	Age	Fare	SibSp	Parch
Accuracy con 5 CV	0.794	0.794	0.794	0.794	0.794

Ningun escalado sirve en los atributos numéricos porque no cambia el accuracy score.

Si reemplazamos SibSp y Parch por la suma tenemos un mejor score de : **0.797**, contra 0.794 por el mejor score actual. Este score no cambia si aplicamos a la suma de estos dos atributos scaler.

Podemos también probar de discretizar la edad con différente pasos y cambiar el atributo numérico edad en un atributo catégorico.

Table 4: **Pipeline con la categorica edad**

Paso entre dos categoricos de edad	num	20	30	40
Accuracy con 5 CV	0.797	0.795	0.795	0.800

Discretizar la edad en la poniendo en categoricas mejora un poco el score de la pipeline a **0.800**.

Hacemos lo mismo por el atributo Fare

Table 5: **Pipeline con la categorica Fare**

Paso entre dos categoricos de Fare	num	100	200	250	300
Accuracy con 5 CV	0.800	0.799	0.799	0.799	0.798

Discretizar fare en la poniendo en categoricas empeora un poco el accuracy, pues no vamos a cambiar el tipo de Fare.

Vamos a hacer una comparacion entre logarithmic regression, svm y RandomForestClassifier

Table 6: **Differente modélos**

Modelo	LogReg	SVM	RandomForestClassifier
Accuracy con 5 CV	0.800	0.787	0.787

Conclusion : La pipeline 3 para tener el mejor score debe tener las categoricas :
numeric cols = ['sum col','Fare']
cat cols3 = ['Age range','Pclass', 'Sex']
Con 'sum col' = 'Parch' + 'SibSp'y con un paso de 40 para Age range.

El modelo eligido es : Logarithmic Regression
Por fin si cambiamos el hyperparametro C por su optimo 1.5, el mejor accuracy que tenemos es **0.801**.
Si miramos el table 1. podemos ver que este accuracy a un poco aumentado. El impacto del cambio de mi pipeline me parece muy poco por el esfuerzo puesto en mejorar la.

Figure 2: Resultado obtenido con el pipeline propuesto

No es realmente lo que fue esperado como es quasi el peor score que tenemos pero es siempre un bueno score para tener resultado más o menos fiables. Me parece que hemos ido overfetting o sea ajustamos demasiado precisamente a los datos de entrenamiento, llegando incluso a memorizar el ruido o las variaciones aleatorias de los datos. Como resultado, el modelo puede tener dificultades para generalizar y comportarse adecuadamente en nuevos datos.

2 Taller 2 - Criticas de cine

1. En primer lugar limpiezamos los datos. Despues separamos los datos en 30 000 de entrenamiento y 5 000 de test. Separamos los datos de entrenamiento en 25 000 de train y 5 000 de validation.
El pipeline basica genere el modelo bag-of-words y realice una clasificación de los mismos utilizando el clasificador de regresión logística.

Table 7: **Comparación de los pipelines**

Name	Basic pipeline	Stop Word pipeline	Tf-Idf pipeline	Bigrama pipeline
Parte	4	5	6	7
Precision	+ : 0.87 - : 0.89	+ : 0.84 - : 0.86	+ : 0.88 - : 0.90	+ : 0.88 - : 0.90
Recall	+ : 0.89 - : 0.87	+ : 0.87 - : 0.83	+ : 0.91 - : 0.87	+ : 0.90 - : 0.87
Accuracy	0.88	0.85	0.89	0.89

Stop Word pipeline: Accuracy se empeora mucho con Stop Word, no vamos a guardarlo en el proximo pipeline pero debemos todavía encontrar una maneja para determinar palabras que no son relevantes y Tf-Idf puede ser una solución para eso.
Tf-Idf pipeline : La técnica tf-idf da un coeficiente de importancia a las palabras lo que permite de obtener una mejora previsión.
Bigram pipeline : Esta técnica tiene más o menos el mismo score que el de Tf-Idf.

Conclusion : Los mejores pipelines son el de la parte 6 Tf-Idf pipeline y de la parte 7 Bigram pipeline.

2. Para mejorar los resultados podemos añadir un modelo que incluya Lemmatization a los dos mejores pipelines.

Observación : Si juntamos Tf-Idf y Bigram empeora el accuracy de la pipeline (accuracy = 0.79).

Table 8: **Lemmanization**

Pipeline	Tf-Idf	Tf-Idf con lemmanization	Bigrama	Bigrama con lemmanization
Precision	+ : 0.88 - : 0.90	+ : 0.61 - : 0.63	+ : 0.88 - : 0.90	+ : 0.80 - : 0.81
Recall	+ : 0.91 - : 0.87	+ : 0.63 - : 0.61	+ : 0.90 - : 0.87	+ : 0.81 - : 0.80
Accuracy	0.89	0.62	0.89	0.81

La lemmanization no mejora los mejores pipelines, pues no vale la pena de añadir la a los mejores pipelines actual.

Limpieza y preprocesamiento de los datos (ej. Contracciones don't → do not)

Table 9: **Limpieza y preprocesamiento de los datos**

Pipeline	Tf-Idf	Tf-Idf con preprocesamiento	Bigrama	Bigrama con preprocesamiento
Precision	+ : 0.88 - : 0.90	+ : 0.88 - : 0.90	+ : 0.88 - : 0.90	+ : 0.88 - : 0.90
Recall	+ : 0.91 - : 0.87	+ : 0.91 - : 0.87	+ : 0.90 - : 0.87	+ : 0.90 - : 0.87
Accuracy	0.89	0.89	0.89	0.89

No cambia nada a los dos pipelines pero, pienso que puede solamente mejorar la pipeline actual pues podemos añadir este procesamiento al pipeline final.

Modelo n-gramas

Table 10: **Modelo n-gramas**

Pipeline	Bigrama	3-grama	4-grama
Precision	+ : 0.88 - : 0.90	+ : 0.82 - : 0.88	+ : 0.75 - : 0.84
Recall	+ : 0.90 - : 0.87	+ : 0.89 - : 0.81	+ : 0.86 - : 0.71
Accuracy	0.89	0.85	0.78

Parece que cuando n sube el accuracy se empeora, pero podemos tomar en cuenta más de un n-grama a la vez.

Table 11: **Modelo n-gramas**

ngram range	(1,2)	(1,3)	(1,4)	(2,3)
Precision	+ : 0.90 - : 0.91	+ : 0.90 - : 0.92	+ : 0.90 - : 0.91	+ : 0.87 - : 0.90
Recall	+ : 0.91 - : 0.90	+ : 0.92 - : 0.90	+ : 0.91 - : 0.90	+ : 0.90 - : 0.87
Accuracy	0.91	0.91	0.91	0.88

El mejor pipeline es el que utiliza 1-grama 2-grama y 3-grama al mismo tiempo. Este modélo tiene una accuracy de 0.91.

Conclusion : Al final utilizamos la pipeline basica y le añadi la limpieza y preprocesamiento de los datos (ej. Contracciones don't → do not) y 1-grama 2-grama y 3-grama al mismo tiempo.

3. El accuracy evaluado con el conjunto de test es de 0.902 que es más o menos lo que tenemos con el validation data. Eso es un bueno desempeño.