Tipología y ciclo de vida de los datos - PRAC2
Limpieza y análisis de datos

David Marín Sánchez
Junio de 2020

Descripción del dataset
El dataset a analizar es el de la competición de Kaggle Titanic: *Machine Learning from Disaster*. Tanto el dataset como su descripción se puede obtener de la sección *data* de la web de la competición.

El problema planteado en esta competición es el de predecir los pasajeros que sobrevivieron al hundimiento del Titanic, acontecido en la fatídica madrugada del 14 al 15 de abril de 1912 durante su viaje inaugural, a partir de los datos disponibles en el dataset.

La importancia de este análisis recae principalmente en que:
- Puede ayudar a comprender qué factores influyeron en el hecho de que un pasajero sobreviviera: si tener en cuenta un factor influye en la precisión del modelo, entonces se puede considerar que ese factor muy probablemente tuvo una influencia significativa.
- Puede ser útil para validar el uso de distintos métodos de análisis predictivo porque la solución al problema es conocida: existen registros de los pasajeros que sobrevivieron y los que no, por lo que es sencillo medir la precisión del modelo.

Integración y limpieza de datos
Este apartado se corresponde con los apartados Adquisición de datos (1) y Limpieza de datos (3) del *notebook* incluido en el proyecto de GitHub de la práctica.

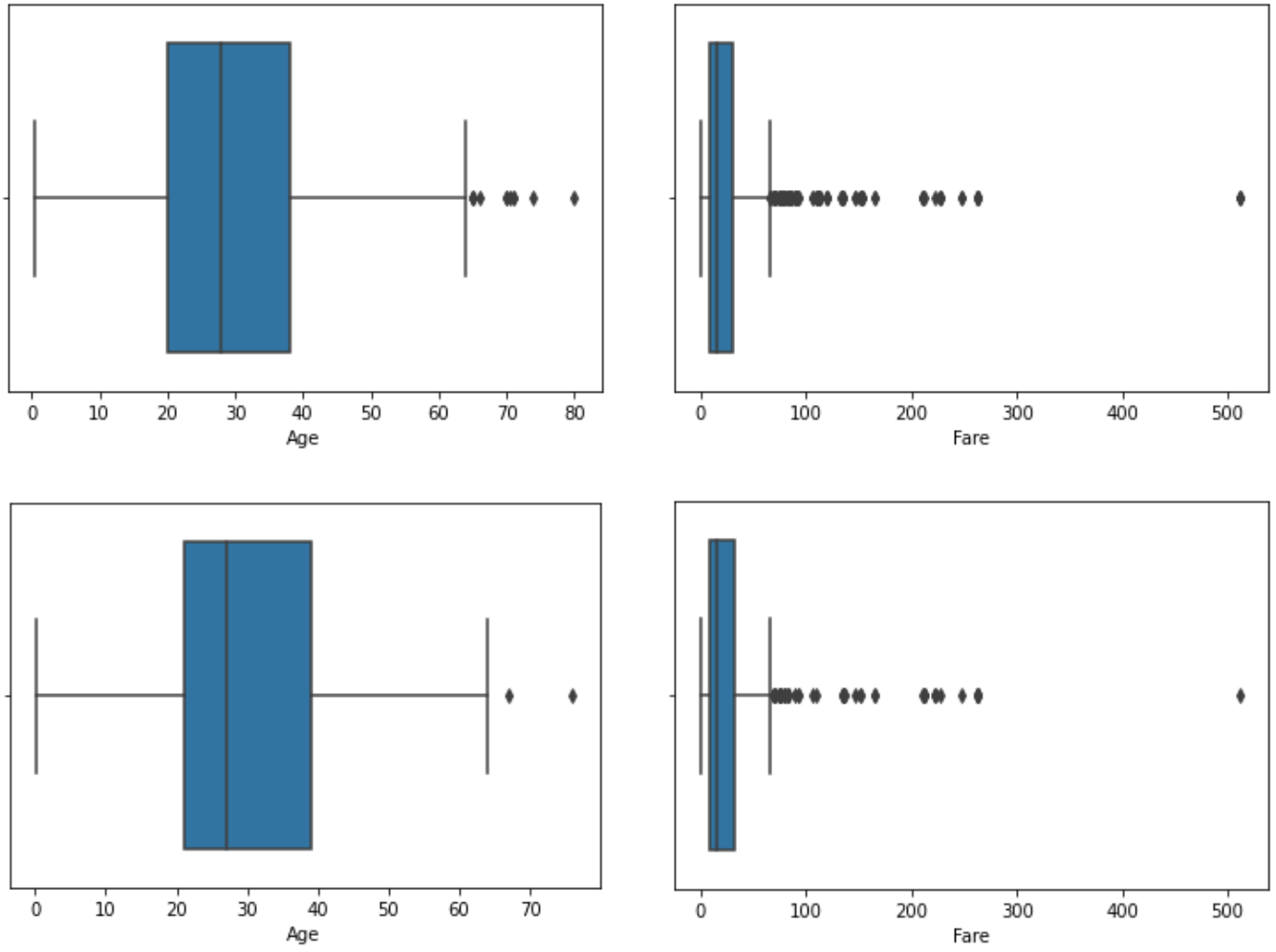
Adquisición de datos
La adquisición de datos se realiza mediante la descarga de los datasets de entrenamiento y test del repositorio de Kaggle. Para ello se utiliza la librería propia de Kaggle y las credenciales de acceso que se almacenan en Google Drive, montado como filesystem en el notebook.

El siguiente paso consiste en cargar el dataset de entrenamiento como DataFrame de Pandas y separar el conjunto de características de la variable objetivo *Survived*. Se hace lo mismo con el dataset de test, pero en este caso no es necesario separar la variable objetivo porque este dataset no la contiene.

Limpieza de datos

Después se realiza el tratamiento de los valores extremos de las características no categóricas (*Age* y *Fare*), que es para las que tiene sentido este tratamiento. Para visualizar los *outliers* se muestran gráficas tipo *boxplot* donde los valores extremos se representan como puntos.

En las siguientes imágenes se pueden ver las gráficas tipo *boxplot* para la variable *Age* (izquierda) y *Fare* (derecha) para los datos de entrenamiento (superior) y de test (inferior):



En el caso de querer reducir o eliminar los valores extremos, se seleccionan las muestras inferiores a un percentil inferior y superiores a un percentil superior y se eliminan sustituyendolos por el valor NaN.

Se han hecho distintas pruebas eliminando mayor o menor cantidad de valores extremos y, aunque los resultados son muy similares, parecen algo mejores cuando se ha reducido la cantidad de valores extremos. Concretamente, la opción escogida ha sido eliminar los valores por debajo del percentil 10 y por encima del percentil 90.

El siguiente paso consiste en extraer nuevas características a partir de las características existentes. En particular se obtiene una nueva variable categórica *Title,* a partir de la variable *Name*, que contiene el título de la persona (*Mr*, *Mrs*, *Miss*, ...), donde cada título tiene asociado un identificador numérico.

Aunque a priori pudiera parecer que el nombre de la persona no debería influir en que sobreviviera o no, el hecho es que el título contenido en el nombre si influye en la precisión de las predicciones.

El siguiente paso consiste en eliminar características que se cree no influyen de forma significativa en los resultados y que, por lo tanto, no se utilizan para realizar el análisis. Las características eliminadas son *Name* (ya se aprovecha su utilidad en la característica *Title*) y las características *Ticket* y *Cabin*, que se pueden considerar simples identificadores que, en principio, no deberían aportar información.

Después de esto se procede a convertir las variables *Sex* y *Embarked* en variables categóricas donde las clases se identifican con valores numéricos. De esta manera podrán ser utilizadas en el análisis.

A continuación se discretizan las variables continuas *Age* y *Fare*, agrupandolas en un número definido de grupos, de manera que cada grupo contenga el mismo número de muestras. Después de realizar distintas pruebas, se ha escogido agrupar las muestras de estas variables en 10 grupos.

El siguiente paso consiste en imputar los valores perdidos (*NaN*) utilizando el método de *k-Nearest Neighbors*. Al finalizar este paso todas las características contienen valores numéricos y sin valores perdidos. A continuación se muestran los primeros valores de las distintas características para el conjunto de datos de entrenamiento (izquierda) y de test (derecha).

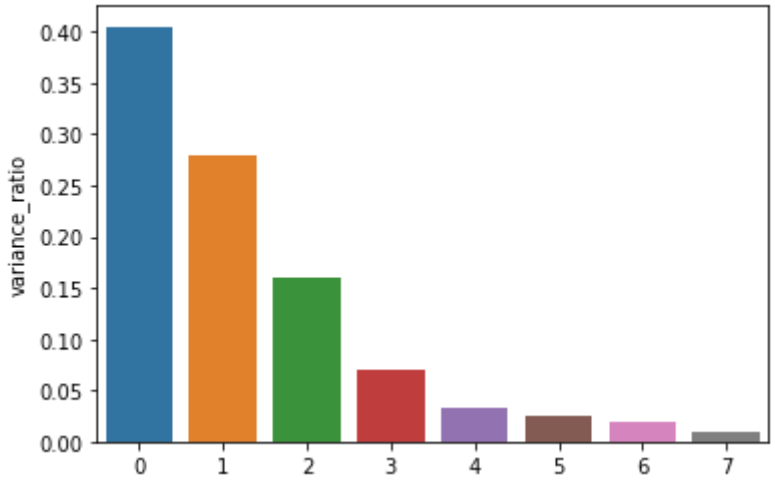
	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
0	3.0	0.0	0.0	1.0	0.0	5.0	0.0	0.0
1	1.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0
2	3.0	1.0	2.0	0.0	0.0	1.0	0.0	2.0
3	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0
4	3.0	0.0	1.0	0.0	0.0	2.0	0.0	0.0

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
0	3.0	0.0	1.0	0.0	0.0	1.0	2.0	0.0
1	3.0	1.0	9.0	1.0	0.0	4.8	0.0	1.0
2	2.0	0.0	4.4	0.0	0.0	4.0	2.0	0.0
3	3.0	0.0	3.0	0.0	0.0	4.0	0.0	0.0
4	3.0	1.0	0.0	1.0	1.0	8.0	0.0	1.0

En el conjunto de datos utilizado los ceros se consideran valores válidos, por lo que no se realiza ningún tratamiento especial sobre ellos.

Después de esto se realiza un análisis de componentes principales, representando el espacio de características en una nueva base para poder escoger, si así se quiere, solo aquellas componentes

que explican mayor variabilidad, es decir, contienen mayor información. En el siguiente gráfico se puede ver la proporción de variabilidad explicada por cada una de estas nuevas componentes:



Aunque se han hecho pruebas reduciendo el número de componentes del espacio de características, los mejores resultados se han obtenido al no reducir el número componentes. Por lo tanto, la opción elegida ha sido no reducir el número de componentes para no perder información.

Finalmente, el último paso de la limpieza de datos consiste en estandarizar el espacio de características (aproximar la media de las variables a 0 y la desviación a 1). Este paso es necesario porque en el análisis se utiliza un modelo supervisado de red neuronal artificial, que es especialmente sensible al escalado de los datos. Después de este paso los primeros valores del conjunto de datos de entrenamiento (superior) y test (inferior) son los siguientes:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
0	0.827377	-0.737695	-1.961031	0.432793	-0.473674	0.268258	-0.571322	-0.542822
1	-1.566107	1.355574	-1.529242	0.432793	-0.473674	-1.533202	1.002728	0.037099
2	0.827377	1.355574	-1.097452	-0.474545	-0.473674	-1.172910	-0.571322	0.617020
3	-1.566107	1.355574	-1.529242	0.432793	-0.473674	-1.533202	-0.571322	0.037099
4	0.827377	-0.737695	-1.529242	-0.474545	-0.473674	-0.812618	-0.571322	-0.542822

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
0	0.827377	-0.737695	-1.529242	-0.474545	-0.473674	-1.172910	2.576777	-0.542822
1	0.827377	1.355574	1.925073	0.432793	-0.473674	0.196199	-0.571322	0.037099
2	-0.369365	-0.737695	-0.061158	-0.474545	-0.473674	-0.092034	2.576777	-0.542822
3	0.827377	-0.737695	-0.665663	-0.474545	-0.473674	-0.092034	-0.571322	-0.542822
4	0.827377	1.355574	-1.961031	0.432793	0.767630	1.349134	-0.571322	0.037099

Análisis de datos

Este apartado se corresponde con los apartados Análisis de datos preliminar (2) y Análisis de datos (4) del *notebook* incluido en el proyecto de GitHub de la práctica.

Análisis de datos preliminar

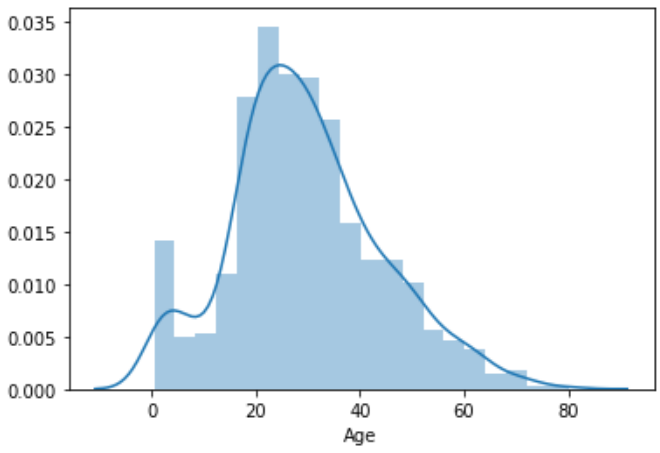
El análisis exploratorio preliminar se realiza justo después de la adquisición de datos y antes de la limpieza de datos. De esta manera es posible realizar el análisis estadístico descriptivo y los análisis de normalidad y homocedasticidad de las variables originales, antes de ser manipuladas en la limpieza de datos.

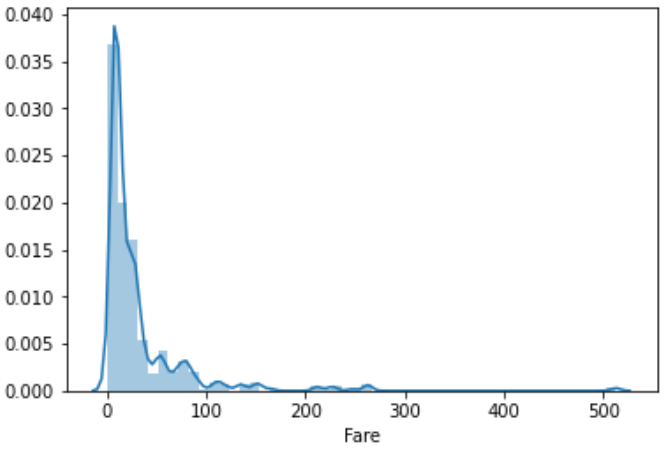
Los resultados de este primer análisis estadístico descriptivo, para los conjuntos de datos de entrenamiento (izquierda) y test (derecha), se pueden observar en las siguientes tablas:

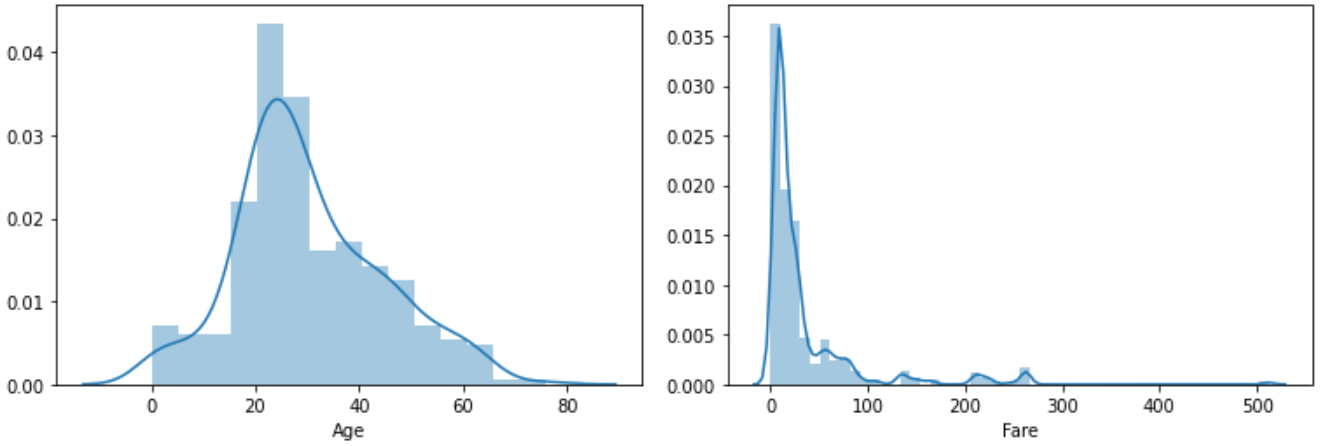
	Pclass	Age	SibSp	Parch	Fare
count	891.000000	714.000000	891.000000	891.000000	891.000000
mean	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.420000	0.000000	0.000000	0.000000
25%	2.000000	20.125000	0.000000	0.000000	7.910400
50%	3.000000	28.000000	0.000000	0.000000	14.454200
75%	3.000000	38.000000	1.000000	0.000000	31.000000
max	3.000000	80.000000	8.000000	6.000000	512.329200

	Pclass	Age	SibSp	Parch	Fare
count	418.000000	332.000000	418.000000	418.000000	417.000000
mean	2.265550	30.272590	0.447368	0.392344	35.627188
std	0.841838	14.181209	0.896760	0.981429	55.907576
min	1.000000	0.170000	0.000000	0.000000	0.000000
25%	1.000000	21.000000	0.000000	0.000000	7.895800
50%	3.000000	27.000000	0.000000	0.000000	14.454200
75%	3.000000	39.000000	1.000000	0.000000	31.500000
max	3.000000	76.000000	8.000000	9.000000	512.329200

Para realizar el análisis de normalidad de las variables continuas se ha utilizado la representación gráfica del histograma de la variable y el test estadístico de Shapiro-Wilk, obteniendo como resultado que la variable *Age* (izquierda) se ajusta a una distribución normal para los conjuntos de datos de entrenamiento (superior) y test (inferior), pero la variable *Fare* (derecha) no lo hace para el conjunto de datos de entrenamiento (superior) pero sí para el de test (inferior):

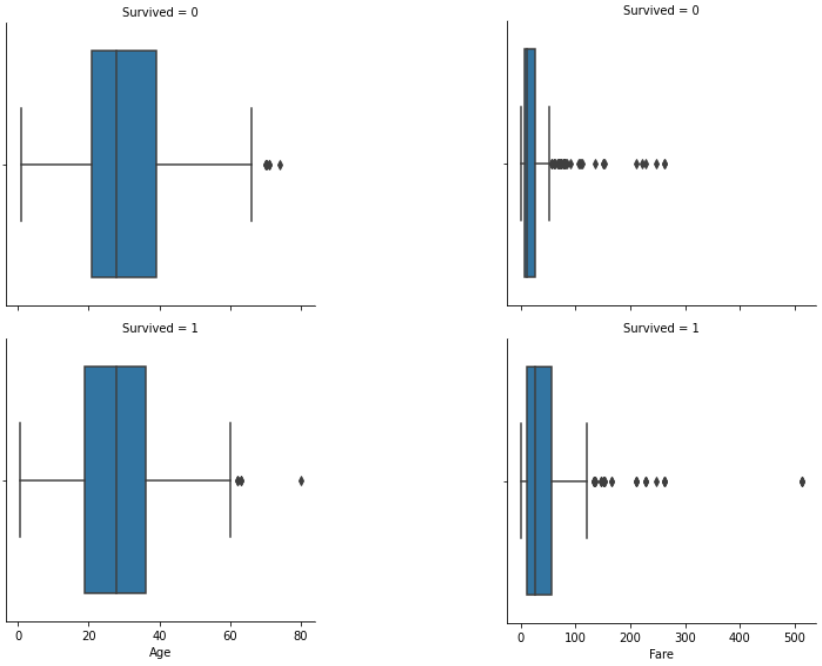






Para comprobar la homocedasticidad de las variables continuas en función de si la persona sobrevivió o no, es decir, si la varianza es significativamente diferente entre los casos de supervivientes y no supervivientes, se utiliza el test de Fligner-Killeen. El resultado obtenido es que la varianza de las variables *Age* y *Fare* son significativamente diferentes para los casos de superviviente y no superviviente.

A continuación se pueden ver los gráficos de cajas para las variables *Age* (izquierda) y *Fare* (derecha) que muestran las diferencias estadísticas para los casos en que la persona sobrevivió (inferior) o no sobrevivió (superior):



Análisis de datos

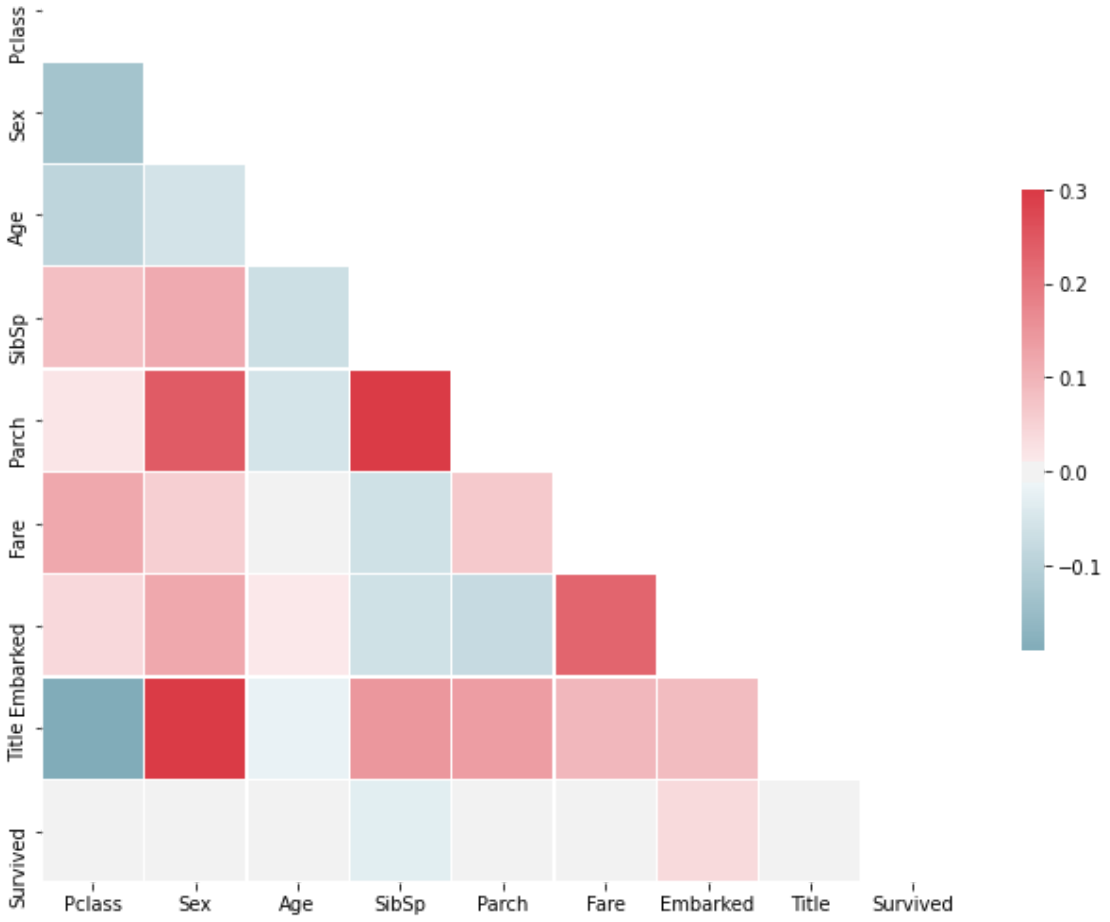
Después de todo el proceso de limpieza de datos, justo a continuación de la estandarización de las características, se vuelve a realizar un análisis estadístico descriptivo de las variables. Los resultados de este análisis, para los datos de entrenamiento (superior) y test (inferior), se pueden ver a continuación:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
count	8.910000e+02	8.910000e+02	8.910000e+02	8.910000e+02	8.910000e+02	8.910000e+02	8.910000e+02	8.910000e+02
mean	-2.031048e-16	3.162453e-16	1.869062e-16	3.456519e-16	6.716164e-17	-1.265978e-16	6.678783e-17	2.367479e-17
std	1.000562e+00	1.000562e+00	1.000562e+00	1.000562e+00	1.000562e+00	1.000562e+00	1.000562e+00	1.000562e+00
min	-1.566107e+00	-7.376951e-01	-1.961031e+00	-4.745452e-01	-4.736736e-01	-1.533202e+00	-5.713215e-01	-5.428218e-01
25%	-3.693648e-01	-7.376951e-01	-6.656631e-01	-4.745452e-01	-4.736736e-01	-8.126182e-01	-5.713215e-01	-5.428218e-01
50%	8.273772e-01	-7.376951e-01	2.519983e-02	-4.745452e-01	-4.736736e-01	-9.203418e-02	-5.713215e-01	-5.428218e-01
75%	8.273772e-01	1.355574e+00	6.297049e-01	4.327934e-01	-4.736736e-01	7.366374e-01	1.002728e+00	6.170204e-01
max	8.273772e-01	1.355574e+00	1.925073e+00	6.784163e+00	6.974147e+00	1.709426e+00	2.576777e+00	8.735916e+00

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000
mean	-0.051570	0.023493	-0.019012	-0.068631	0.013345	0.065874	0.159218	-0.067508
std	1.007462	1.008165	1.034796	0.813665	1.218251	1.043931	1.079036	0.794538
min	-1.566107	-0.737695	-1.961031	-0.474545	-0.473674	-1.533202	-0.571322	-0.542822
25%	-1.566107	-0.737695	-0.665663	-0.474545	-0.473674	-0.812618	-0.571322	-0.542822
50%	0.827377	-0.737695	0.025200	-0.474545	-0.473674	-0.092034	-0.571322	-0.542822
75%	0.827377	1.355574	0.629705	0.432793	-0.473674	0.988842	1.002728	0.559028
max	0.827377	1.355574	1.925073	6.784163	10.698058	1.709426	2.576777	6.996152

Se observa como la estandarización ha funcionado mejor para los datos de entrenamiento, es decir, la media se acerca más a 0 y la desviación más a 1, que para los datos de test. Esto es así porque para realizar la estandarización, igual que para el resto de transformaciones realizadas en la limpieza de datos (eliminación de *outliers*, extracción de nuevas características y categorización y discretización de variables), se han utilizado únicamente los datos de entrenamiento. El objetivo es no aprovechar la información existente en los datos de test para que los modelos obtenidos sean lo más generalizables posibles y la precisión de estos modelos, calculada a partir de las predicciones obtenidas para los datos de test, sea lo más realista posible.

El siguiente paso consiste en un análisis de correlación. Este análisis es útil para ver si existe una correlación fuerte entre alguna de las características y la variable objetivo, o entre las propias características. El siguiente gráfico muestra las correlaciones entre las variables para el conjunto de datos de entrenamiento:



Se observa como existen características que se correlaciona de forma significativa entre sí, como por ejemplo *Sex* y *Title* o *SibSp* y *Parch*. Este resultado verifica que el espacio de características puede ser expresado en unas nuevas componentes no correlacionadas, tal como se vió en el análisis de componentes principales realizado anteriormente.

También se puede ver que no existe una característica que esté correlacionada significativamente más que las demás con la variable objetivo *Survived*. Por lo tanto, parece acertado utilizar todas las características disponibles para la predicción.

Los siguientes pasos consisten en la realización del análisis predictivo utilizando primero un modelo de regresión logística y luego un modelo supervisado de red neuronal artificial. Antes de implementar estos modelos se opta por separar parte de las muestras del conjunto de datos de entrenamiento generando un subconjunto de datos de validación. Este subconjunto se utilizará para la validación de los modelos antes de subir los resultados definitivos a Kaggle (apartado 5 del *notebook*), donde se obtendrá la precisión de los modelos a partir de las predicciones realizadas para el conjunto de datos de test.

Se ha escogido utilizar un modelo de regresión logística porque es útil para resolver el problema de clasificación binaria planteado.

El modelo de red neuronal se ha implementado utilizando una capa de entrada de 8 neuronas (número de características) con una función de activación ReLU. La capa oculta se ha implementado con 32 neuronas, también con la función de activación ReLU. Finalmente se ha implementado la capa de salida con una sola neurona y la función de activación Sigmoide, para que la red neuronal funcione como un clasificador binario.

La función de coste utilizada ha sido la de *Binary Cross Entropy*, útil para clasificadores binarios, y el algoritmo de optimización escogido ha sido Adam, con un ratio de aprendizaje de 0,0002.

Para el entrenamiento de la red se ha elegido un máximo de 200 ciclos de entrenamiento o épocas, con la posibilidad de parar el entrenamiento antes si el valor del coste, para las muestras de validación, crece durante diez épocas seguidas. Se utiliza como modelo final el de la época con menor coste de validación.

En las diferentes pruebas realizadas, el modelo de regresión logística normalmente ha obtenido una precisión algo inferior que el modelo de red neuronal, tanto para los datos de validación como para los datos de test. También se ha observado que los valores de precisión obtenidos para los datos de validación, normalmente superiores al 80%, han sido siempre algo mejores que para los datos de test, normalmente inferiores al 80%.

La mejor precisión para los datos de test ha sido del 80,38% y se se ha obtenido para el modelo de red neuronal. Al ser el mejor resultado, es el utilizado para la competición de Kaggle.

A modo de ejemplo, se muestran a continuación la precisión de los modelos para la prueba final que se puede ver en *notebook*, tanto para los datos de validación como para los de test:

Modelo	**Regresión logística**	**Red neuronal artificial**
Precisión de validación	81,11%	84,44%
Matriz de confusión de validación	55 | 8 9 | 18	58 | 5 9 | 18
Precisión de test	74,64%	77,99%

Conclusiones

Se puede concluir que, tanto el modelo de regresión logística como el modelo de red neuronal pueden ser útiles para resolver el problema de clasificación binaria planteado, porque la precisión obtenida, cercana siempre al 80%, es muy superior al 50%, valor esperado para predicciones realizadas de forma totalmente aleatoria en el caso de una variable dicotómica.

Pero, a la vista de la precisión obtenida por otros competidores en Kaggle, aún sería posible mejorar la precisión de los resultados. Para ello, y como propuesta de mejora para tratar de obtener resultados más precisos, sería necesario extraer nuevas características a partir de las disponibles, revisar la elección de parámetros, el procesado de datos y la estructura de la red neuronal y plantear el uso de otros modelos supervisados distintos al de red neuronal.

Referencias

https://www.kaggle.com/ouyangg/titanic
https://triangleinequality.wordpress.com/2013/09/08/basic-feature-engineering-with-the-titanic-data/
https://www.kaggle.com/parthsuresh/binary-classifier-using-keras-97-98-accuracy
https://machinelearningmastery.com/binary-classification-tutorial-with-the-keras-deep-learning-library
https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping
https://github.com/Kaggle/kaggle-api
https://scikit-learn.org/stable/modules/impute.html
https://scikit-learn.org/dev/modules/generated/sklearn.impute.KNNImputer.html
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
https://scikit-learn.org/stable/modules/neural_networks_supervised.html#mlp-tips
https://nextjournal.com/schmudde/how-to-remove-outliers-in-data
https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.bar.html
https://seaborn.pydata.org/examples/many_pairwise_correlations.html

Licencia