

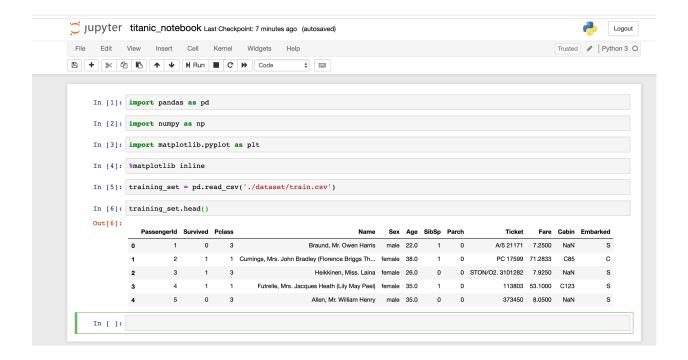
Requerimientos técnicos sobre Mac OS

Lo primero que se debe tener en cuenta es instalar las siguientes dependencias:

- brew install python3
- pip3 install jupyter jupyter_kernel_gateway
- pip3 install matplotlib
- pip3 install pandas
- pip3 install sklearn

Procesamiento del conjunto de datos.

Lo primero que se debe realizer es leer el dataset, para lo cual se empleara la función read_csv de pandas, la cual cargará el archive train.csv. Con la función head se podrán pre-visualizar algunas de las filas del conjunto de datos como se puede apreciar en la siguiente imagen.





Información básica del conjunto de datos

La estructura training_set (pandas.core.frame.DataFrame) tiene varias propiedades entre ellas shape, la cual muestra el número filas y columnas. Por ejemplo la siguiente imagen indica que tiene 891 filas y 12 columnas.

```
In [7]: training_set.shape
Out[7]: (891, 12)
```

Si se quiere averiguar los títulos de las columnas se puede acceder a la propiedad columns

Si se quiere averiguar el numero de valores nulos por columna se debe llamar a la función isnull y luego sumarizar todos los valores con la función sum

En la imagen anterior se puede apreciar que en la columna Age hay 177 registros nulos de la misma forma en la columna Cabin 687.

Preparando el conjunto de datos

Se deben crear nuevas clases, en este caso se creará una nueva columna Deck, la cual nos indica si la columna



```
In [17]: def assignDeckValue(CabinCode):
               if pd.isnull(CabinCode):
                    category = 'Unknow
               else:
                    category = CabinCode[0]
               return category
           deck = np.array([assignDeckValue(cabin) for cabin in training_set['Cabin'].values])
           training_set = training_set.assign(Deck= deck)
In [18]: training_set.head()
Out[18]:
              PassengerId Survived Pclass
                                                Name
                                                         Sex Age SibSp Parch
                                                                                   Ticket
                                                                                           Fare Cabin Embarked
                                                                                                                    Deck
                                            Braund, Mr.
                                                                             0 A/5 21171 7.2500
                                                                                                               S Unknow
                                            Owen Harris
                                          Cumings, Mrs.
John Bradley
                                                                             0 PC 17599 71.2833
                                                                                                                      С
                                1
                                                       female 38.0
                                                                                                  C85
                                                                       1
                                            (Florence
Briggs Th...
                                            Heikkinen,
Miss. Laina
                                                                             0 STON/O2.
3101282
                                                                                          7.9250
                                                      female 26.0
                                                                                                  NaN
                                                                                                               S Unknow
                                           Futrelle, Mrs.
                                                                                                                      С
                                                       female 35.0
                                                                                  113803 53.1000
                                                                                                  C123
                                                                                                               S
                                          (Lily May Peel)
                                              Allen, Mr.
                                0
                                       3
                                                        male 35.0
                                                                       0
                                                                                  373450
                                                                                          8.0500
                                                                                                  NaN
                                                                                                               S Unknow
                                           William Henry
```

Ahora se procede a crear la clase Family size, la cual indicará el número de hermanos y padres en conjunto.

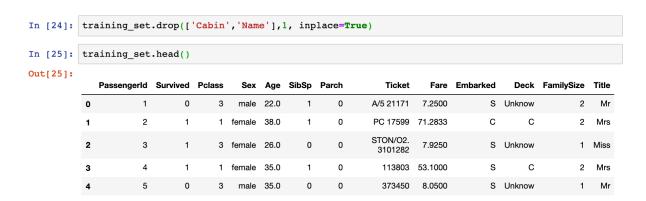
n [19]:	trainin	g_set['I	FamilyS	Size'] =	traini	ing_s	et['S	ibSp']	+ train	ing_set	t['Par	ch'] + 1		
n [20]:	trainin	g_set.he	ead()											
Out[20]:	assengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Deck	FamilySize
	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	Unknow	2
	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	1	0	PC 17599	71.2833	C85	С	С	2
	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	Unknow	1
	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	С	2
	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	Unknow	1



Ahora se creará la clase Title la cual indicara si es Sr, Ms, entre otros.

23]: 23]:	<pre>training_set.head()</pre>														
)	erld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Deck	FamilySize	Title
	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	Unknow	2	Mr
	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	1	0	PC 17599	71.2833	C85	С	С	2	Mrs
	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	Unknow	1	Miss
	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	s	С	2	Mrs
	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	Unknow	1	Mr

Ahora que se proceso la información como se requiere se proce a borrar las columnas Cabin y Name



Controlando valores Faltantes

Para la clase Embarked, faltan altunos valores, los valores que puede tomar esta son C=Cherbourg, Q=Queenstown, S = Southampton, en este caso se procede para los valores



faltantes asignar S que es el mas comun. Luego se realiza una contabilidad para verificar que no hay valores nulos para esta columna.

```
In [26]: # Returns count for each category
    training_set['Embarked'].value_counts()

# Fills null values with 'S'-most common occurence
    common = 'S'
    training_set['Embarked']=training_set['Embarked'].fillna('S')

# Checking the no of null values now
    training_set['Embarked'].isnull().sum()
Out[26]: 0
```

Para la clase Age, se realiza un procedimiento similar, estableciendo el valor medio a partir del titulo.

```
n [27]: means = training_set.groupby('Title')['Age'].mean()

title_list = ['Master', 'Miss', 'Mr', 'Mrs', 'Others']

def age_missing_replace(means, dframe, title_list):
    for title in title_list:
        temp = dframe['Title'] == title
        dframe.loc[temp, 'Age'] = dframe.loc[temp, 'Age'].fillna(means[title])

age_missing_replace(means, training_set, title_list)
```

Codificación de características categóricas.

Muchos algoritmos de aprendizaje automático no pueden admitir valores categóricos sin convertirse en valores numéricos. Para lo cual se encarga de usar la función map la cual asignara un valor numeric a cada columna dependiendo cómo se haya definido.



	training_set	['Sex']	= trai	ining	_set	['Sex].map	rked'].map({'(('male':0, ': .map({'Master	female'	:1})	• /	rs':3,'Ot	hers'
In [30]:	training_se	.head()											
Out[30]:	Passengerld	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Deck	FamilySize	Title
	0 1	0	3	0	22.0	1	0	A/5 21171	7.2500	2	Unknow	2	2
	1 2	1	1	1	38.0	1	0	PC 17599	71.2833	0	С	2	3
	2 3	1	3	1	26.0	0	0	STON/O2. 3101282	7.9250	2	Unknow	1	1
	3 4	1	1	1	35.0	1	0	113803	53.1000	2	С	2	3
	4 5	0	3	0	35.0	0	0	373450	8.0500	2	Unknow	1	2

Asignar valores de forma manual, no es la mejor opción ya que se puede dejar datos de forma inconsistente y el tiempo empleado puede ser alto depende del volumen de datos a procesar. En este caso se empleara LabelEncoder de sklear para realizar la conversión de la columna Deck. En la siguiente imagen se aprecia la asignación de valores numéricos por categoría de forma aleatoria a la columna Deck.



[30]:	trainin	g_set	.head()											
[30]:	Passe	ngerld	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Deck	FamilySize	Title
	0	1	0	3	0	22.0	1	0	A/5 21171	7.2500	2	Unknow	2	! 2
	1	2	1	1	1	38.0	1	0	PC 17599	71.2833	0	С	2	! 3
	2	3	1	3	1	26.0	0	0	STON/O2. 3101282	7.9250	2	Unknow	1	1
	3	4	1	1	1	35.0	1	0	113803	53.1000	2	С	2	! 3
	4	5	0	3	0	35.0	0	0	373450	8.0500	2	Unknow	1	2
[31]:			<pre>import essing.</pre>											
[sform	(train	ning_set['Deck'	1)				
[34]:	trainin	g_set	.head()											
:[34]	Passe	ngerld	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Deck	FamilySize	Title
												_	2	
	0	1	0	3	0	22.0	1	0	A/5 21171	7.2500	2	8	2	2
			0	3	0	22.0 38.0	1	0	A/5 21171 PC 17599	7.2500 71.2833	0	2	2	3
	0	1			1			0				_		
	0	1 2	1	1	1	38.0 26.0	1	0	PC 17599	71.2833	0	2	2	3

Después de realizar este paso ya se encuentra lista la información sin ningún valor faltante.