import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import re

import numpy as np

from sklearn import tree

from sklearn.model\_selection import train\_test\_split

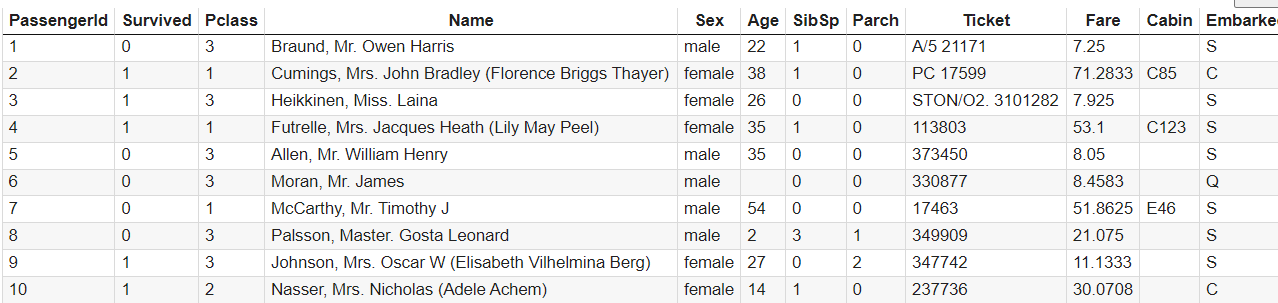
%matplotlib inline

sns.set()

test\_df = pd.read\_csv('/content/titanic-test.csv')

train\_df = pd.read\_csv('/content/titanic-train.csv')

train\_df.head()

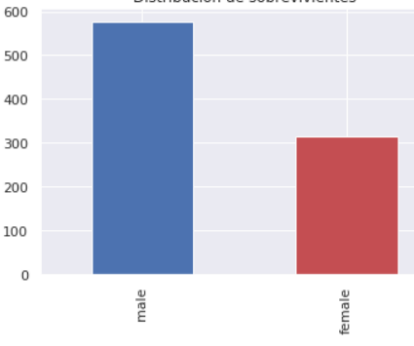


train\_df.info()

train\_df.Sex.value\_counts().plot(kind = 'bar', color = ['b', 'r'])

plt.title('Distribucion x sexo')

plt.show()



#Esta seria otra forma de filtrar los sobrevivientes y separarlos por sexo

df\_train[ df\_train['Survived'] == 1 ]['Sex'].value\_counts().plot(kind='bar', color=['b','r'])

plt.title('Distribucion de sobrevivientes')

Chart, bar chart

Description automatically generated

Separar mujeres / hombers

men=train\_df.loc[train\_df['Sex']=='male']

women=train\_df.loc[train\_df['Sex']=='female']

men.Survived.value\_counts().plot(kind='bar', color=['b','r'])

plt.title('Distribucion de hombre sobrevivientes')

plt.show()

women.Survived.value\_counts().plot(kind='bar', color=['b','r'])

plt.title('Distribucion de mujeres sobrevivientes')

plt.show()

train\_df['Age'] = train\_df['Age'].fillna(train\_df['Age'].median())

train\_df['Embarked'] = train\_df['Embarked'].fillna('S')

train\_predictors = train\_df.drop(['PassengerId', 'Survived', 'Name', 'Ticket', 'Cabin'], axis = 1)

categorical\_cols = [cname for cname in train\_predictors.columns if

                        train\_predictors[cname].nunique() < 10 and

                        train\_predictors[cname].dtype == 'object'

numerical\_cols = [cname for cname in train\_predictors.columns if

                   train\_predictors[cname].dtype in ['int64', 'float64']

                 ]

my\_cols = categorical\_cols + numerical\_cols

train\_predictors = train\_predictors[my\_cols]

dummy\_encoded\_train\_predictors = pd.get\_dummies(train\_predictors)

train\_df['Pclass'].value\_counts()

3 491

1 216

2 184

Name: Pclass, dtype: int64

y\_target = train\_df['Survived'].values

x\_features\_one = dummy\_encoded\_train\_predictors.values

x\_train, x\_validation, y\_train, y\_validation = train\_test\_split(x\_features\_one, y\_target, test\_size = .25, random\_state = 1)

tree\_one = tree.DecisionTreeClassifier()

tree\_one = tree\_one.fit(x\_features\_one, y\_target)

tree\_one\_accuracy = round(tree\_one.score(x\_features\_one, y\_target), 4)

print('Accuracy: %0.4f' % (tree\_one\_accuracy))

Accuracy: 0.9798

from io import StringIO

from IPython.display import Image, display

import pydotplus

out = StringIO()

tree.export\_graphviz(tree\_one, out\_file = out)

graph = pydotplus.graph\_from\_dot\_data(out.getvalue())

graph.write\_png('titanic.png')

True