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

#datos prueba / entrenamiento

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

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

#Exploracion

train\_df.info()

train\_df.head()

train\_df.dtypes

#Si desea explorar por Sexo sin grafica

#  train\_df.Sex.value\_counts()

# Explore visualizacion de sobrevivientes x Sexo

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

#  plt.title('Distribucion de sobrevivientes')

Chart, bar chart

Description automatically generated

#Para ser entendido un dato

#por la maquina, se crea un label encoder

#el modelo usara campo Sex de tipo 'object'

#el cual puede ser modificado a tipo categoria

from sklearn import preprocessing

label\_encoder = preprocessing.LabelEncoder()

encoder\_sex = label\_encoder.fit\_transform(train\_df['Sex'])

train\_df['Sex'] = encoder\_sex

#Limpiar informacion

#===================

#Manejo de nulos

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

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

#Los campos que no se pueden predecir eliminelos

#y cree nuevo dataframe

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

#Crear lista de columnas categoricas del dataFrame

#Condiciones unicos < de tamano menor a 3 characteres y tipo objeto

#Sino ponemos esta condicion incluye ticket, o bien eliminamos la columna ticket antes

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

                         train\_predictors[cname].nunique() < 3 and

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

                    ]

print(categorical\_cols)

# -> Resultado ['Embarked']

#Si desea explorar valores posibles de Embarked

#  train\_df.Embarked.value\_counts()

#Crear lista de columnas numericas del dataFrame

#Condicion int64, float64

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

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

                 ]

#print(numerical\_cols)

# -> Resultado ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']

#union de los datos

my\_cols = categorical\_cols + numerical\_cols

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

#crea nuevo dataFrame con las columnas que se escogieron

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

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

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

#verificar el nivel que tendra de exactitud modelo

# aun no se usa dummy\_encoded\_train\_predictors

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

tree\_one = tree.DecisionTreeClassifier()

tree\_one = tree\_one.fit(x\_train,y\_train)

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

tree\_one\_accuracy

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())

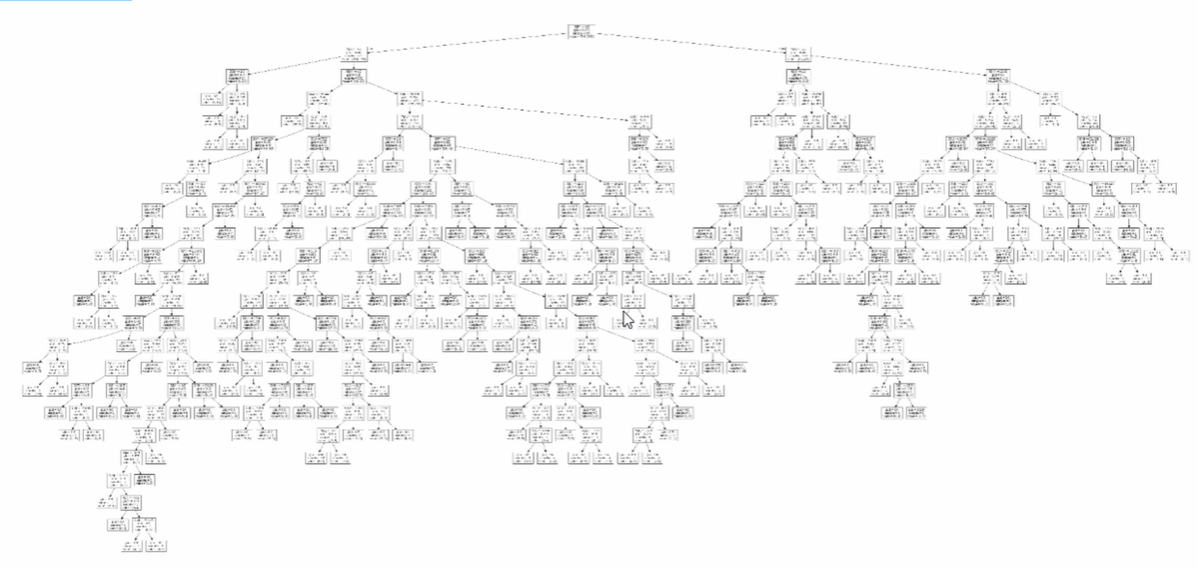
# EN

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

Refresque donde estas los archivos y aparecera archive titanic.png

Le da click y aparecera el arbol de decisiones

El arbol de decision para saber si sobrevive o no



Como es muy grande…. Esta sobreentrenado

<https://colab.research.google.com/drive/1jMtnmQySIXXBNSvAzsehucPO3j43OqLC?usp=drive_open>

Se obtiene el árbol pero hubiese sido interesante se analizara el árbol y mencionar cosas al menos ligeramente sobre el indice de Gini que es el que por defecto asume sklearn, pero también se hubiera podido usar “entropy” para hablar del Gain\_info. tambien faltó mencionar que una vez obtenido se puede pensar en hacer una “poda” del árbol que mejore el modelo. y de los diagramas de complejidad que te permite ayudarte en este proceso.