2. Problem Statement

Predicting Survival in the Titanic Data Set

We will be using a decision tree to make predictions about the Titanic data set from

Kaggle. This data set provides information on the Titanic passengers and can be used to

predict whether a passenger survived or not.

Loading Data and modules

import numpy as np

import pandas as pd

import seaborn as sb

import matplotlib.pyplot as plt

import sklearn

from pandas import Series, DataFrame

from pylab import rcParams

from sklearn import preprocessing

from sklearn.linear\_model import LogisticRegression

from sklearn.cross\_validation import train\_test\_split

from sklearn import metrics

from sklearn.metrics import classification\_report

Url= https://raw.githubusercontent.com/BigDataGal/Python-for-Data-

Science/master/titanic-train.csv

titanic = pd.read\_csv(url)

titanic.columns =

['PassengerId','Survived','Pclass','Name','Sex','Age','SibSp','Parch','Ticket','Fare','Cabin','E

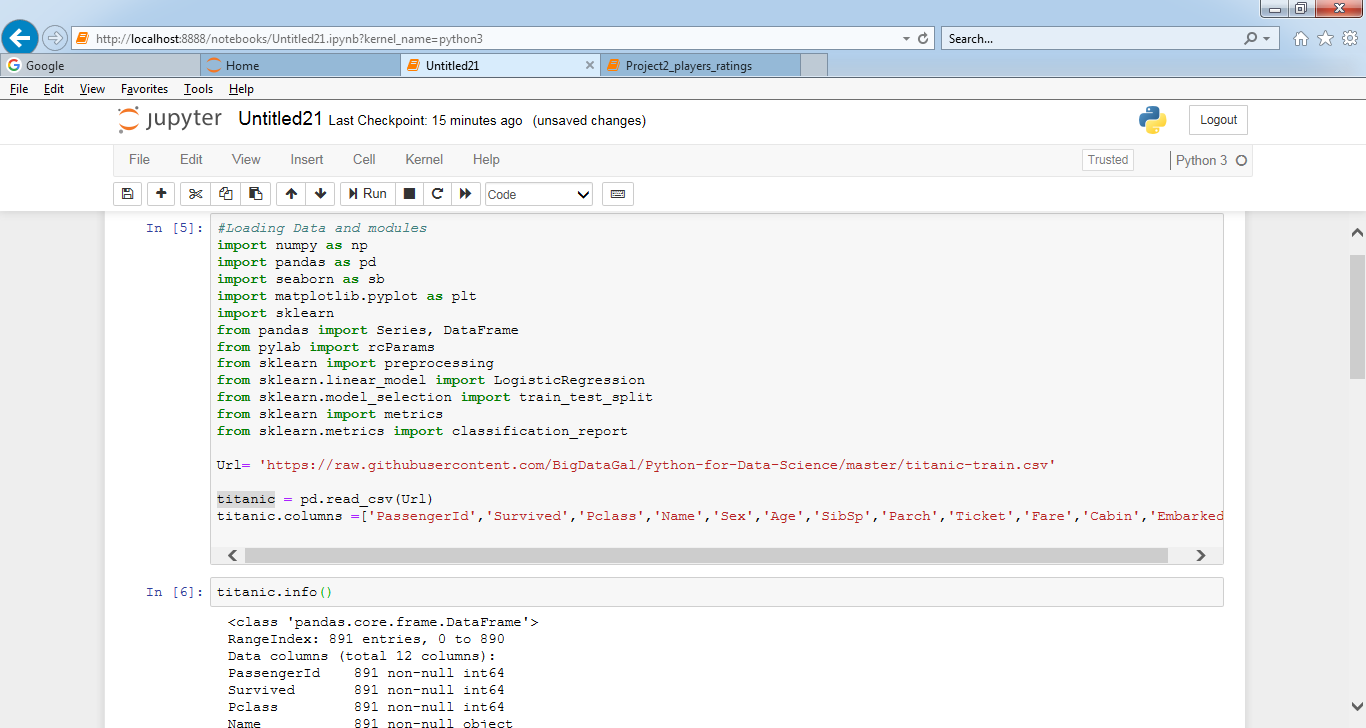
mbarked']

You use only Pclass, Sex, Age, SibSp (Siblings aboard), Parch (Parents/children aboard),

and Fare to predict whether a passenger survived.

NOTE: The solution shared through Github should contain the source code used and

the screenshot of the output.



import seaborn as sns

categ = [ 'Pclass', 'Sex', 'SibSp', 'Parch', 'Embarked']

conti = ['Fare', 'Age']

fig = plt.figure(figsize=(30, 10))

for i in range (0,len(categ)):

fig.add\_subplot(3,3,i+1)

sns.countplot(x=categ[i], data=titanic);

for col in conti:

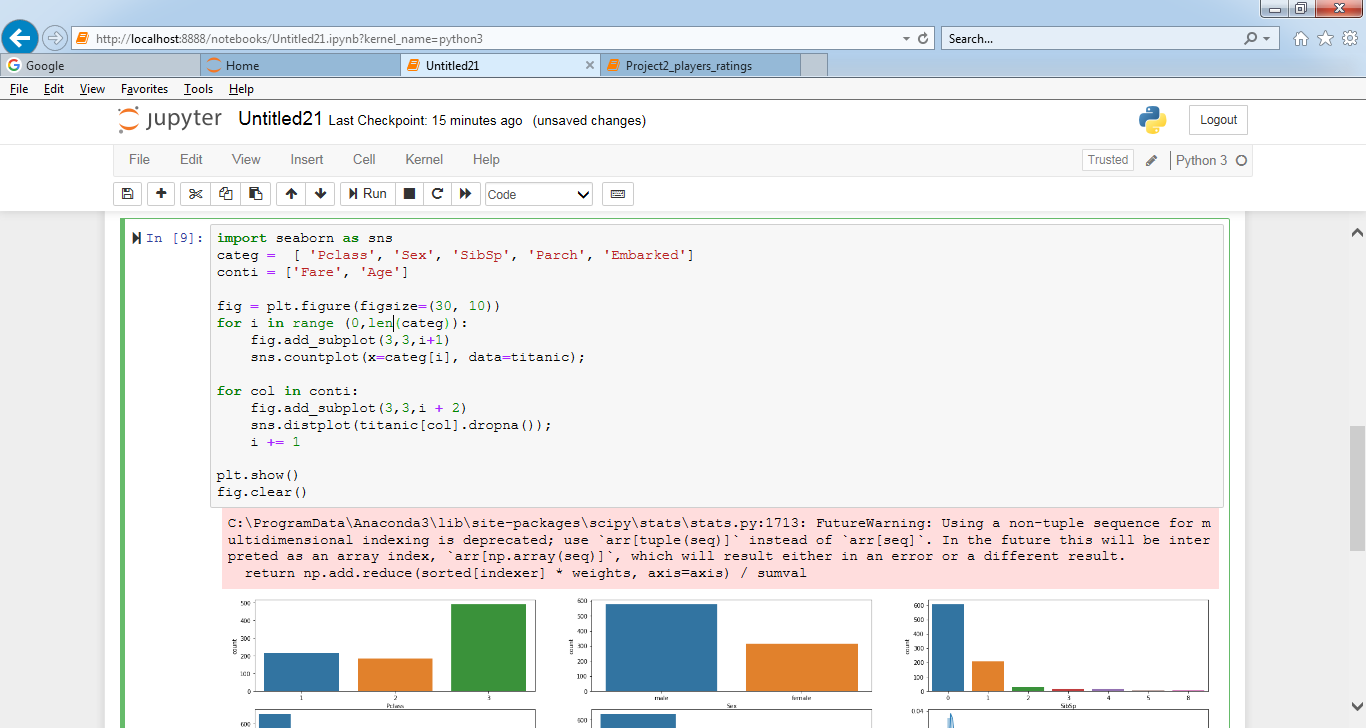
fig.add\_subplot(3,3,i + 2)

sns.distplot(titanic[col].dropna());

i += 1

plt.show()

fig.clear()



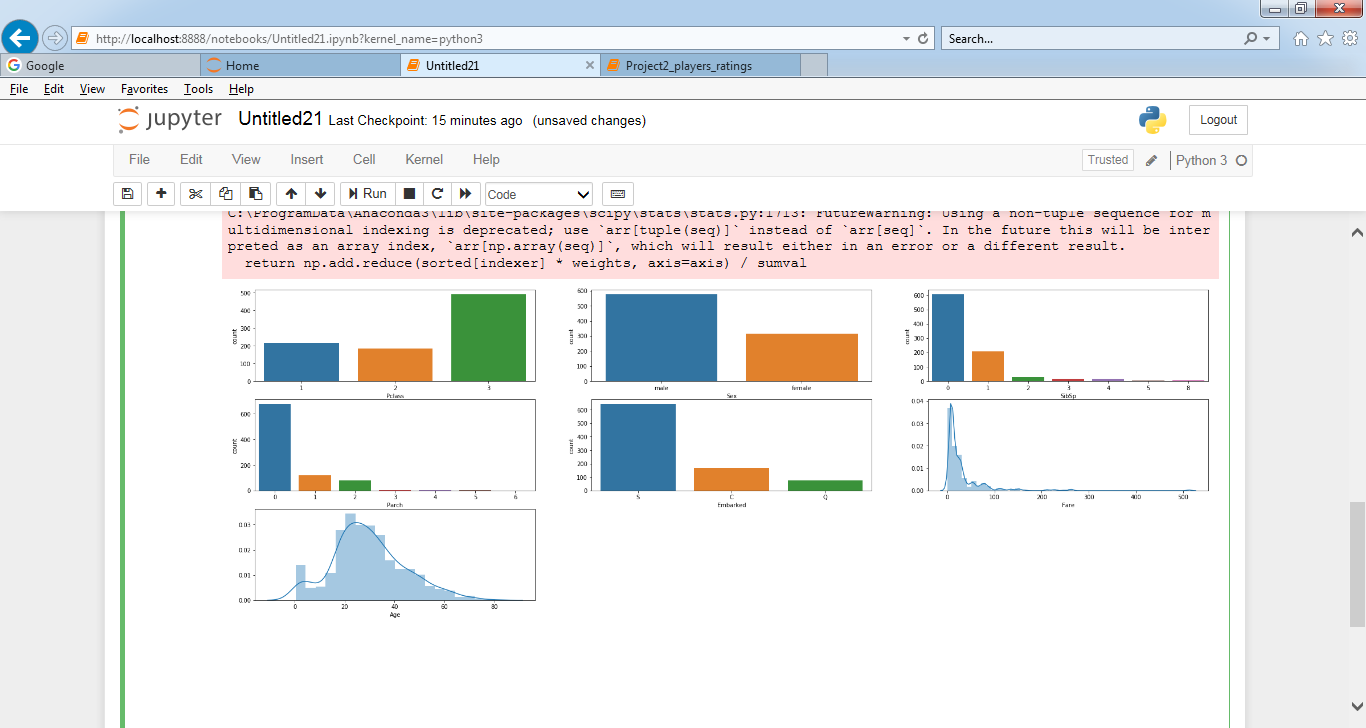


fig = plt.figure(figsize=(30, 10))

i = 1

for col in categ:

if col != 'Survived':

fig.add\_subplot(3,3,i)

sns.countplot(x=col, data=titanic,hue='Survived');

i += 1

# Box plot survived x age

fig.add\_subplot(3,3,6)

sns.swarmplot(x="Survived", y="Age", hue="Sex", data=titanic);

fig.add\_subplot(3,3,7)

sns.boxplot(x="Survived", y="Age", data=titanic)

# fare and Survived

fig.add\_subplot(3,3,8)

sns.violinplot(x="Survived", y="Fare", data=titanic)

# correlations with the new features

corr = titanic.drop(['PassengerId'], axis=1).corr()

mask = np.zeros\_like(corr, dtype=np.bool)

mask[np.triu\_indices\_from(mask)] = True

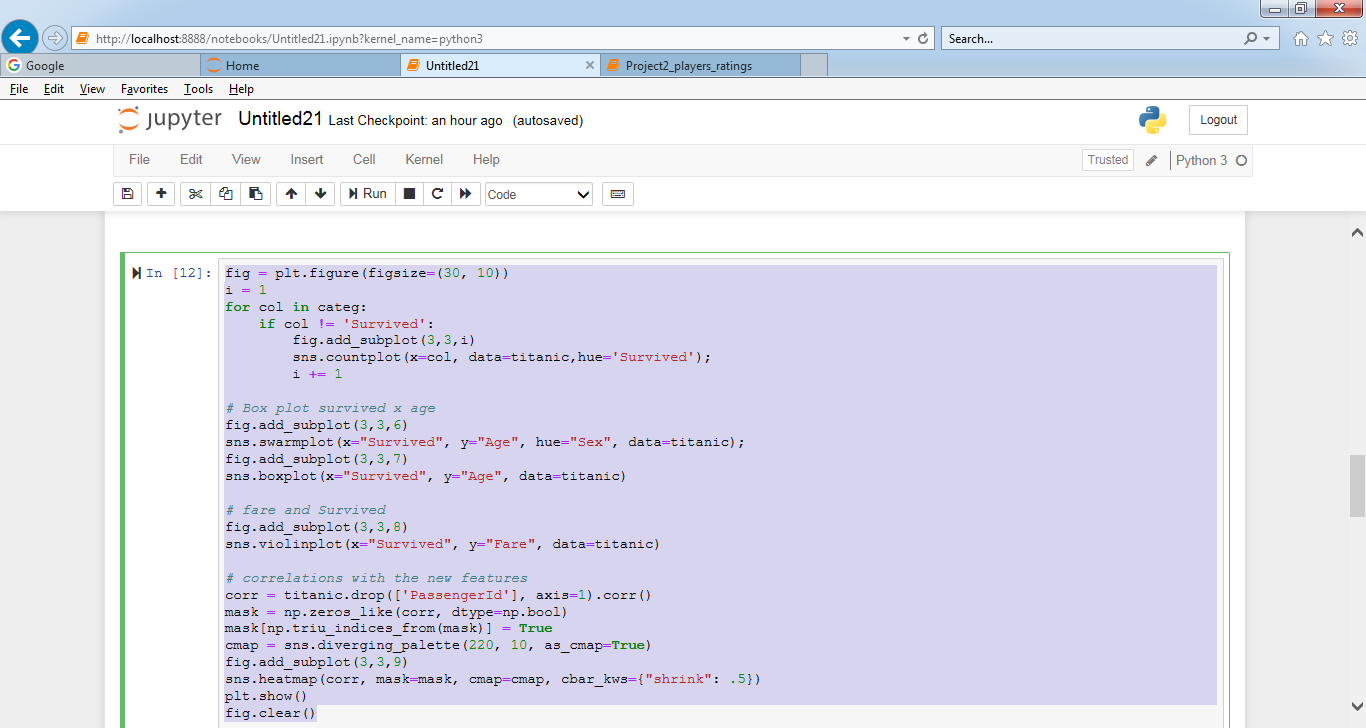
cmap = sns.diverging\_palette(220, 10, as\_cmap=True)

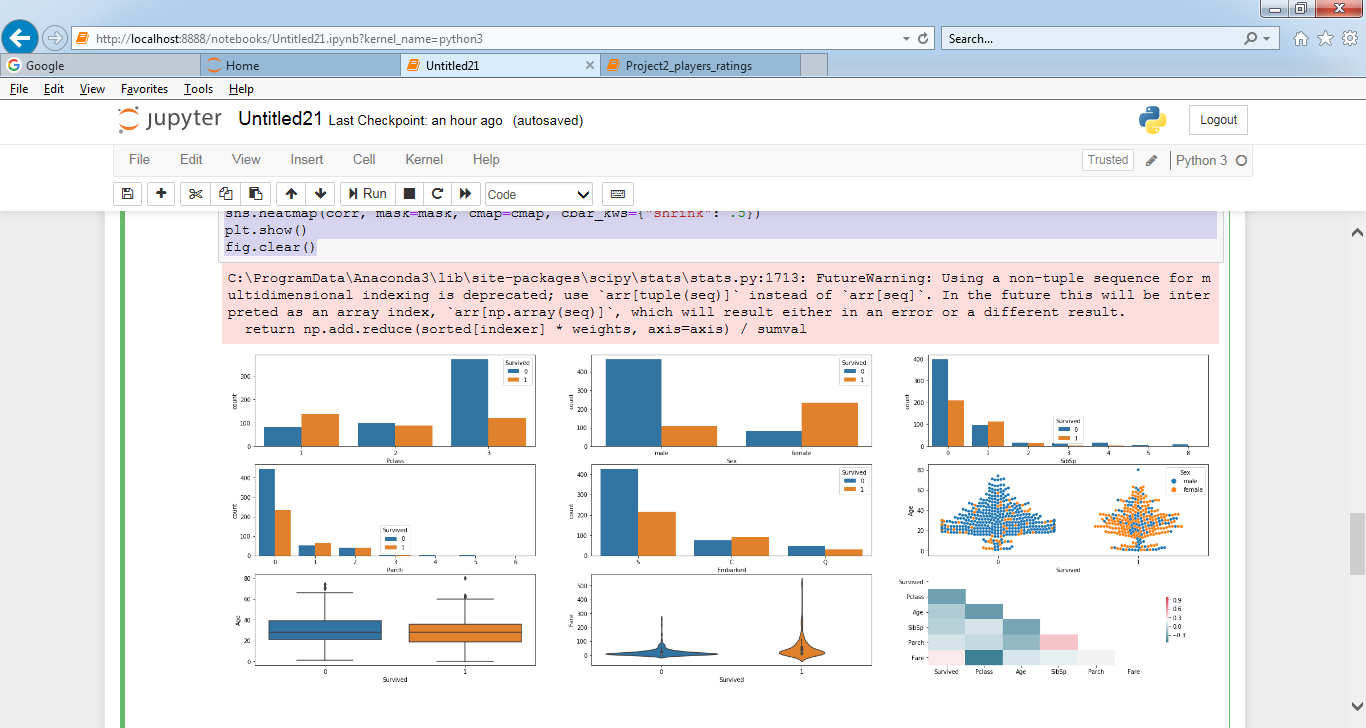
fig.add\_subplot(3,3,9)

sns.heatmap(corr, mask=mask, cmap=cmap, cbar\_kws={"shrink": .5})

plt.show()

fig.clear()





title = ['Mlle','Mrs', 'Mr', 'Miss','Master','Don','Rev','Dr','Mme','Ms','Major','Col','Capt','Countess']

def ExtractTitle(name):

tit = 'missing'

for item in title :

if item in name:

tit = item

if tit == 'missing':

tit = 'Mr'

return tit

titanic["Title"] = titanic.apply(lambda row: ExtractTitle(row["Name"]),axis=1)

plt.figure(figsize=(13, 5))

fig.add\_subplot(2,1,1)

sns.countplot(x='Title', data=titanic,hue='Survived');

