2. Problem Statement In this assignment, students will be using the K-nearest neighbors algorithm to predict how many points NBA players scored in the 2013-2014 season. A look at the data Before we dive into the algorithm, let’s take a look at our data. Each row in the data contains information on how a player performed in the 2013-2014 NBA season. Download 'nba\_2013.csv' file from this link: https://www.dropbox.com/s/b3nv38jjo5dxcl6/nba\_2013.csv?dl=0 Here are some selected columns from the data: player - name of the player pos - the position of the player g - number of games the player was in gs - number of games the player started pts - total points the player scored There are many more columns in the data, mostly containing information about average player game performance over the course of the season. See this site for an explanation of the rest of them. We can read our dataset in and figure out which columns are present: import pandas with open("nba\_2013.csv", 'r') as csvfile: nba = pandas.read\_csv(csvfile)

Solution:

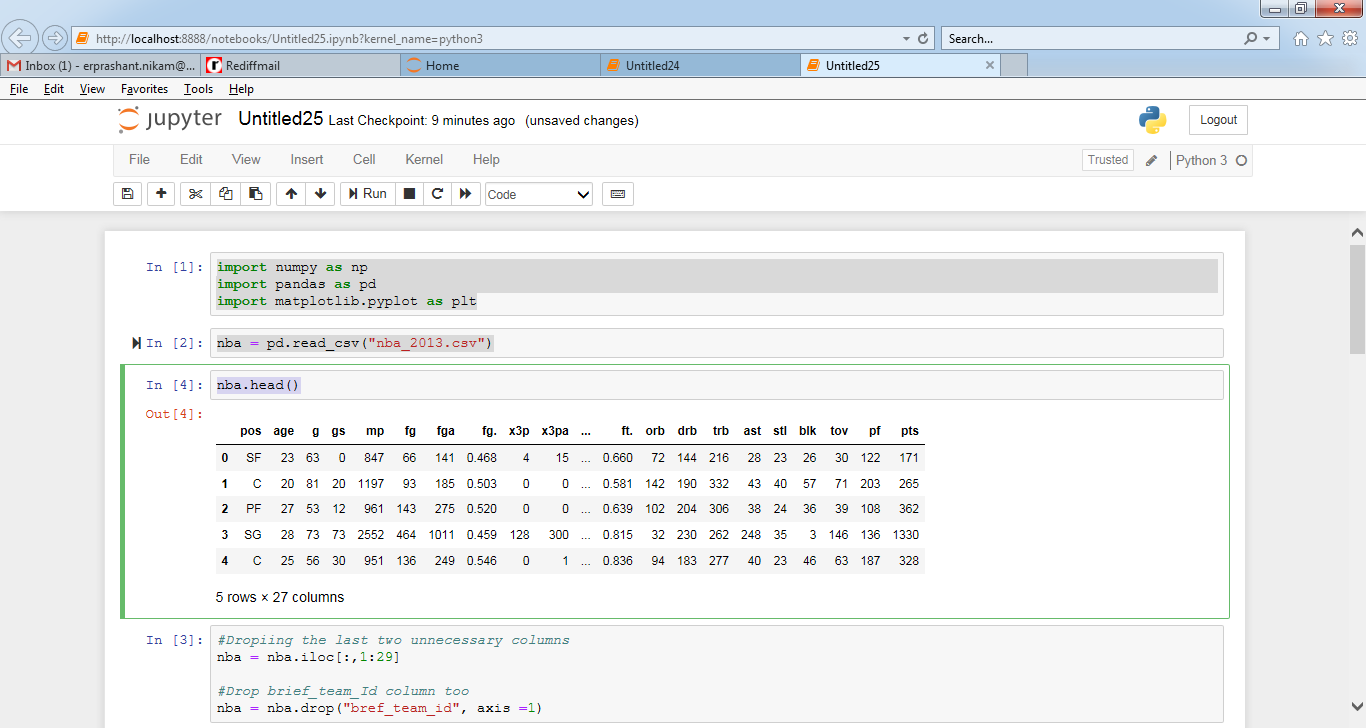
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

nba = pd.read\_csv("nba\_2013.csv")

nba.head()



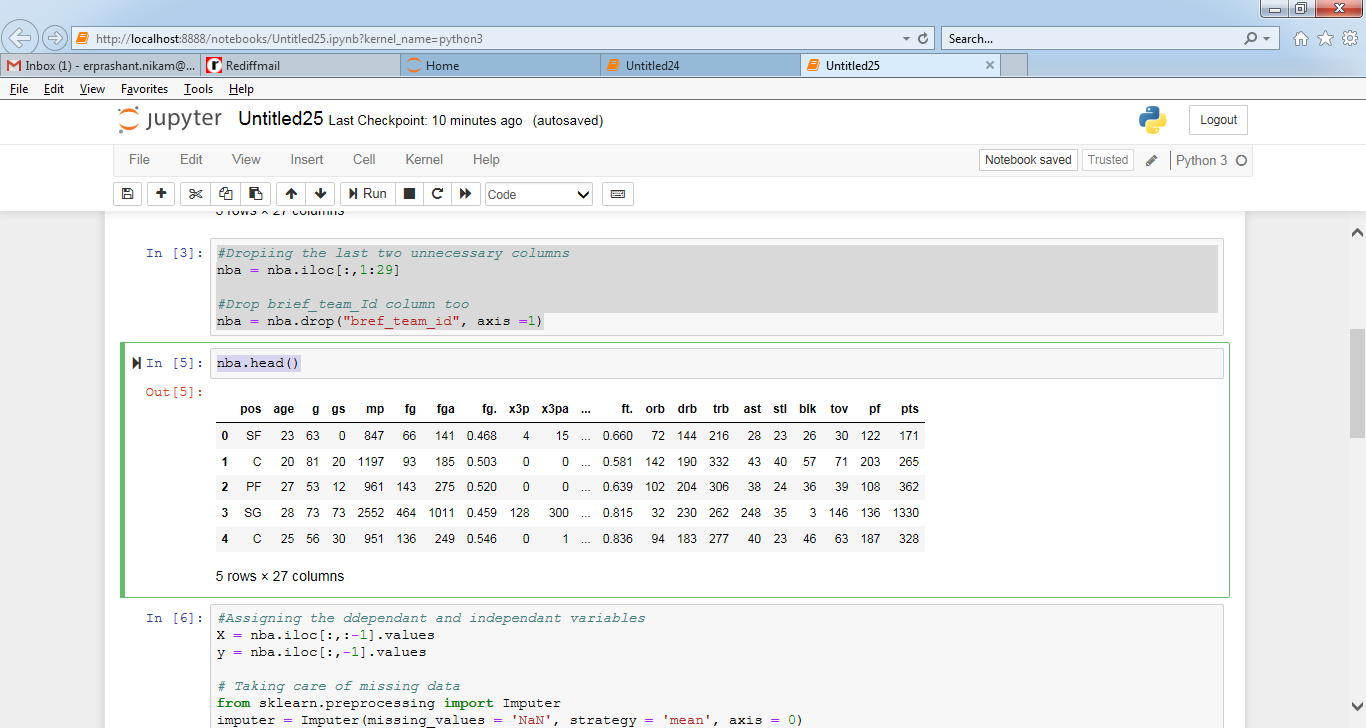
#Dropiing the last two unnecessary columns

nba = nba.iloc[:,1:29]

#Drop brief\_team\_Id column too

nba = nba.drop("bref\_team\_id", axis =1)

nba.head()



#Assigning the ddependant and independant variables

X = nba.iloc[:,:-1].values

y = nba.iloc[:,-1].values

# Taking care of missing data

from sklearn.preprocessing import Imputer

imputer = Imputer(missing\_values = 'NaN', strategy = 'mean', axis = 0)

imputer = imputer.fit(X[:,[7,10,13,14,17]])

X[:,[7,10,13,14,17]] = imputer.transform(X[:,[7,10,13,14,17]])

#Label encoding categorical value

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder = LabelEncoder()

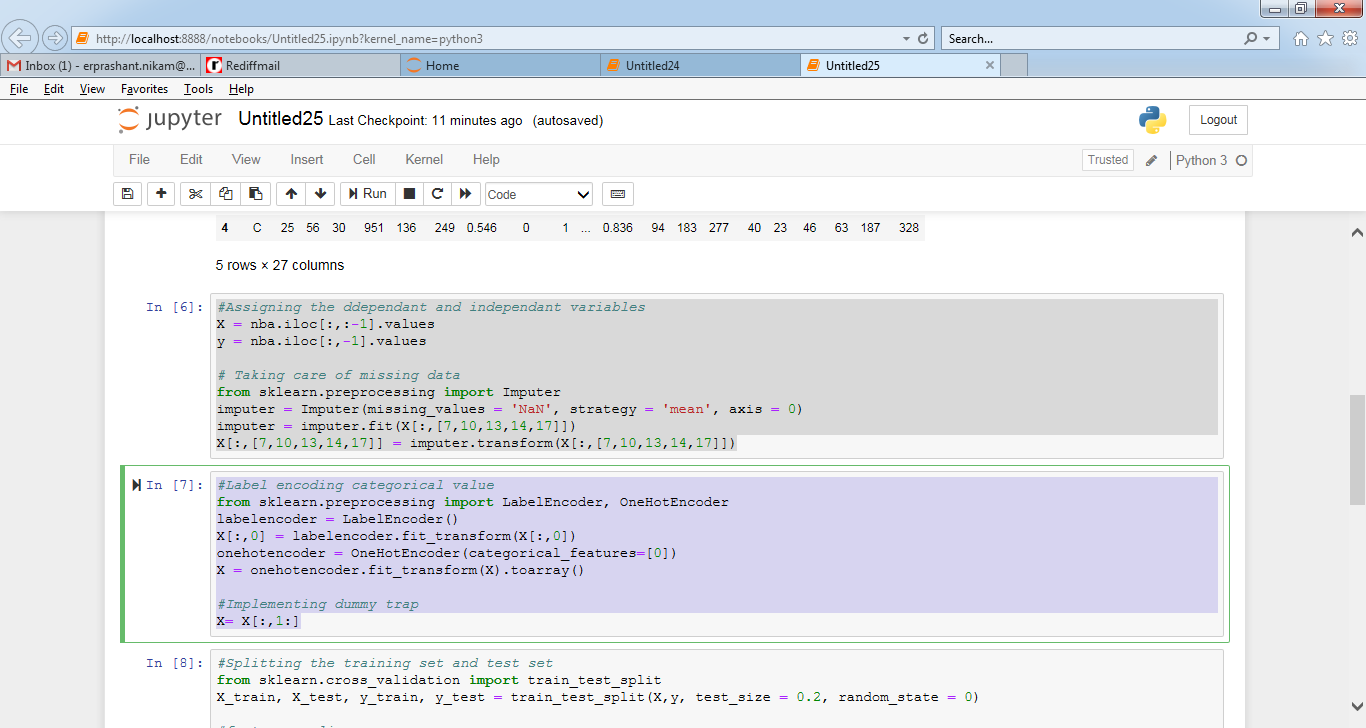
X[:,0] = labelencoder.fit\_transform(X[:,0])

onehotencoder = OneHotEncoder(categorical\_features=[0])

X = onehotencoder.fit\_transform(X).toarray()

#Implementing dummy trap

X= X[:,1:]



#Splitting the training set and test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size = 0.2, random\_state = 0)

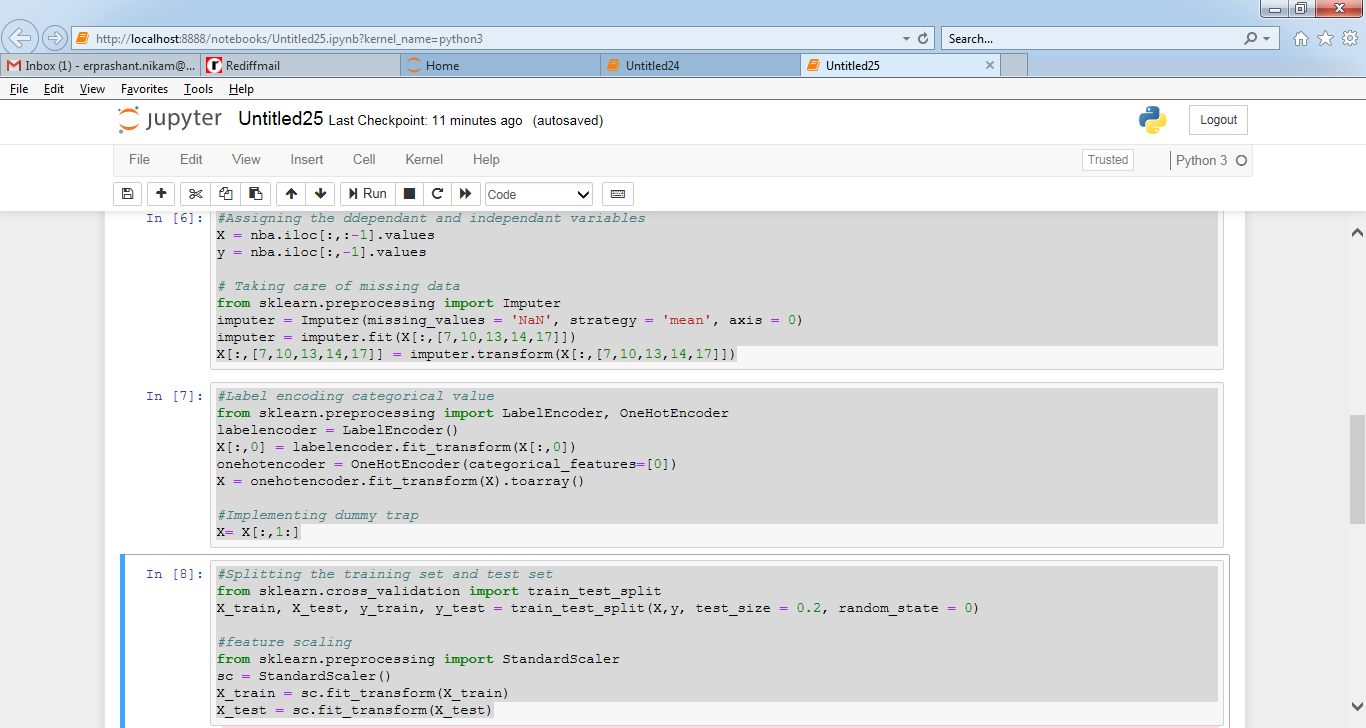
#feature scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.fit\_transform(X\_test)



#Applying Algorithm KNN to the training dataset

from sklearn.neighbors import KNeighborsRegressor

classifier = KNeighborsRegressor(n\_neighbors=5, metric= "minkowski", p =2)

classifier.fit(X\_train, y\_train)

#Predicting the test data

y\_pred = classifier.predict(X\_test)

#Plotting the scatter plots

plt.scatter(y\_test, y\_pred)

plt.xlabel("Actual Pts")

plt.ylabel("Predicted pts")

plt.title("Actual Pts vs Predicted Pts")

