DATA SCIENCE ESSENTIALS

(CSC570)

FINAL PROJECT ON

CHOOSING THE BEST CAR

By

SAI KIRAN REDDY VAJRALA

UIN: 670965547

**INTRODUCTION:**

My final project is about finding the best car from different automobile companies.

I have taken a dataset containing information about different cars and implemented the below things in my project work.

**Ipython code**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

**Import the csv file**

import csv

cars = pd.read\_csv('carsdata.csv', sep=',')

# please keep the csv file in your ipython directory.

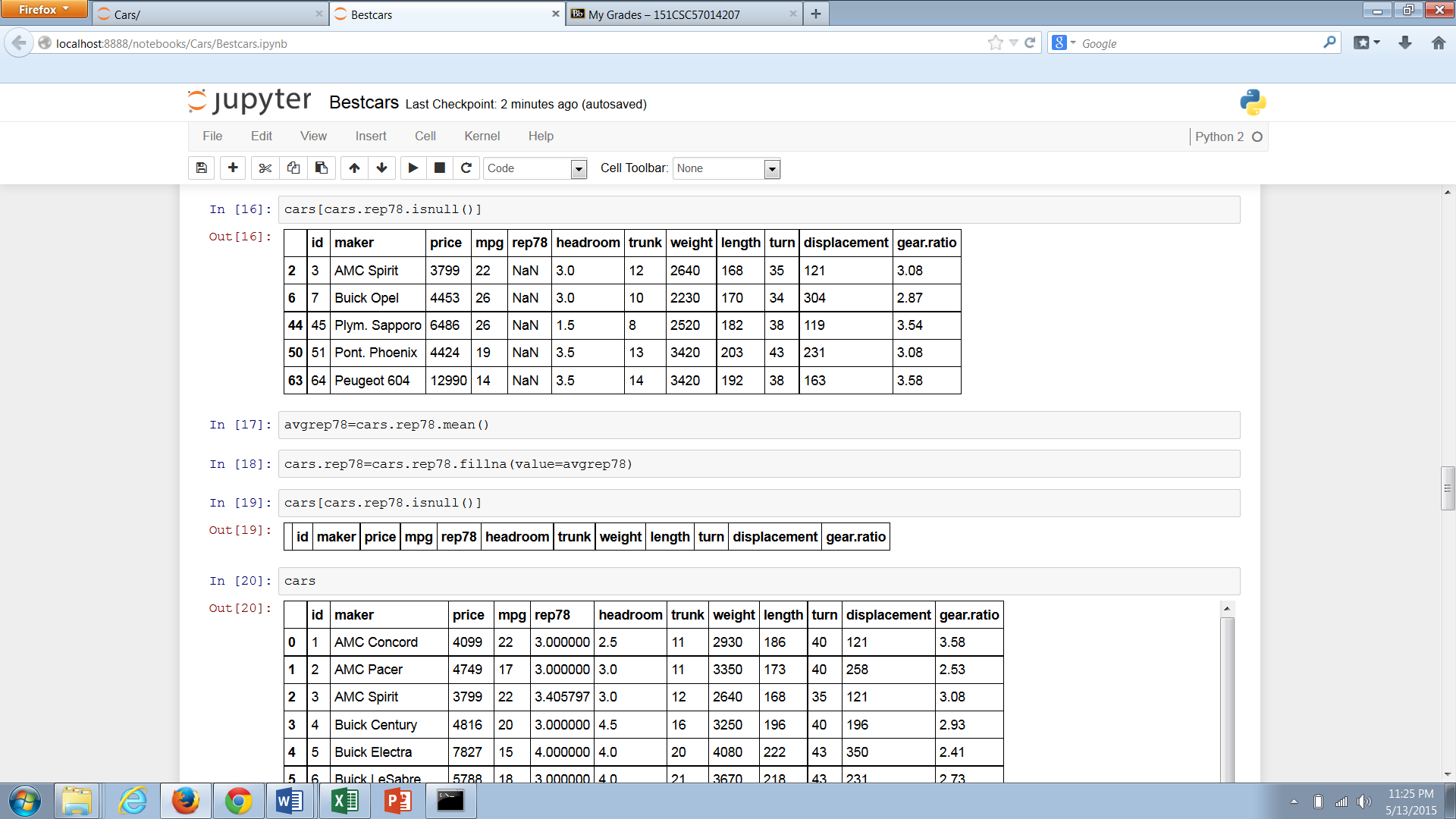
**Check the data**

print(cars)

**Missing values**

avgrep78=cars.rep78.mean()

cars.rep78=cars.rep78.fillna(value=avgrep78)



**Calculate the basic descriptive statistics for some major variables**

cars.price.describe()

cars.mpg.describe()

cars.trunk.describe()

cars.length.describe()

cars.weight.describe()

**Histogram**

plt.hist(cars.length)

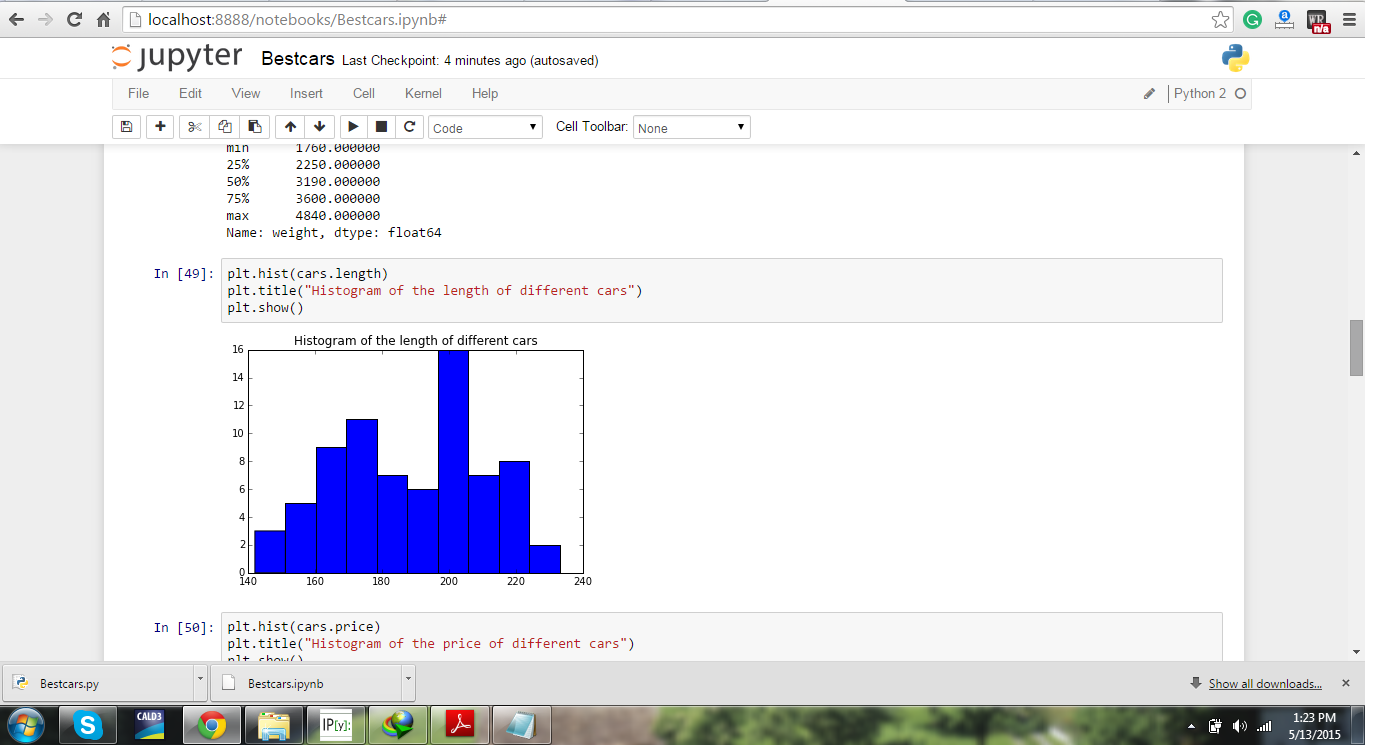
plt.title("Histogram of the length of different cars")

plt.show()

plt.hist(cars.price)

plt.title("Histogram of the price of different cars")

plt.show()



**Regression model**

from \_\_future\_\_ import print\_function

from statsmodels.compat import lzip

import statsmodels

import statsmodels.formula.api as smf

import statsmodels.stats.api as sms

**fitting the model**

regcar=smf.ols("(price~mpg+weight+length+trunk)", data=cars).fit()

print(regcar.summary())

**Diagonostics of the model**

**Independence**

np.linalg.cond(regcar.model.exog)

**BPG test**

name = ['Lagrange multiplier statistic', 'p-value',

'f-value', 'f p-value']

test = sms.het\_breushpagan(regcar.resid, regcar.model.exog)

lzip(name, test)

**Goldfeld test**

name = ['F statistic', 'p-value']

test = sms.het\_goldfeldquandt(regcar.resid, regcar.model.exog)

lzip(name, test)

**Implementation of machine learning in ipyhton a logistic regression approach**

# I have tested two different categories of car( 0= foreign car, 1= domestic car)

import numpy as np

import pandas as pd

import sklearn

import sklearn.linear\_model as lm

import sklearn.cross\_validation as cv

import sklearn.grid\_search as gs

import matplotlib.pyplot as plt

%matplotlib inline

data = cars[['price', 'mpg', 'weight', 'length', 'fd']].copy()

data['fd'] = data['fd'] == 1

data = data.dropna()

data = cars[['price', 'mpg', 'weight', 'length', 'fd']].copy()

data['fd'] = data['fd'] == 1

data = data.dropna()

data\_np = data.astype(np.int32).values

X = data\_np[:,:-1]

y = data\_np[:,-1]

(X\_train, X\_test,

y\_train, y\_test) = cv.train\_test\_split(X, y, test\_size=.05)

logreg = lm.LogisticRegression()

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

y\_predicted = logreg.predict(X\_test)

plt.figure(figsize=(8, 3));

plt.imshow(np.vstack((y\_test, y\_predicted)),

interpolation='none', cmap='bone');

plt.xticks([]); plt.yticks([]);

plt.title(("home and domestic cars"

" on the test set"));

**Cross validation check**

cv.cross\_val\_score(logreg, X, y)