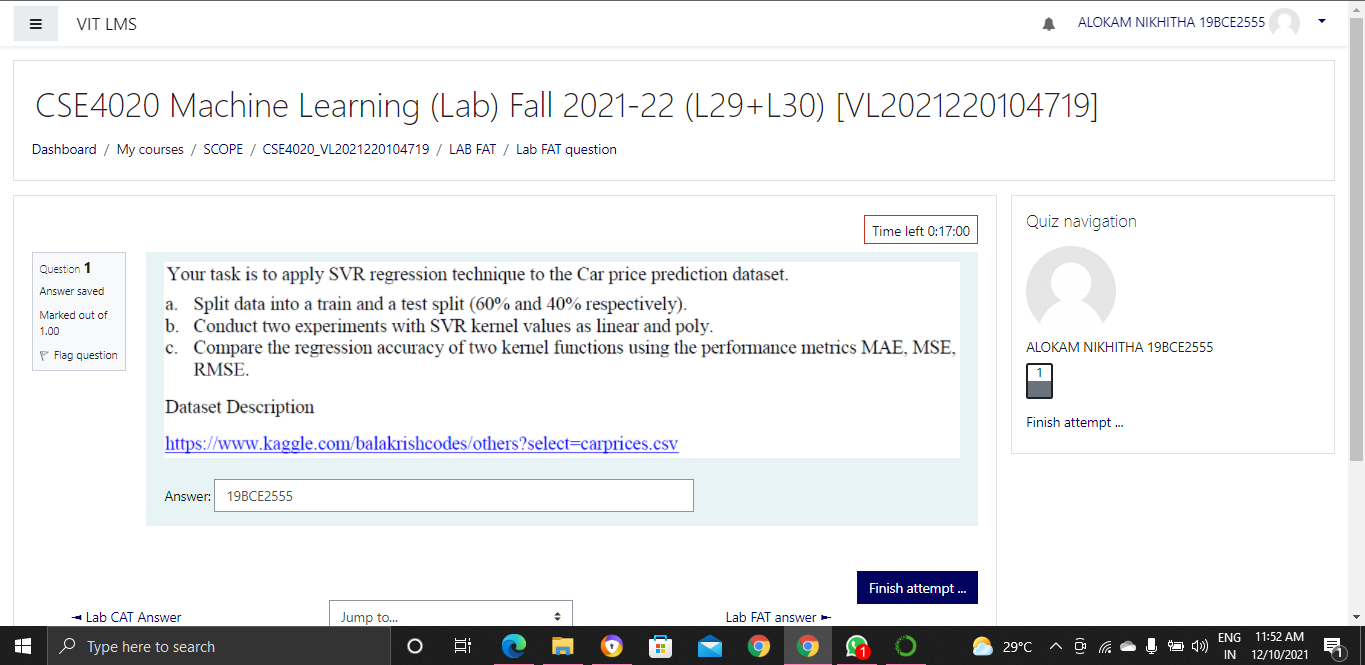
# CSE 4020 - MACHINE LEARNING

# Lab 29+30

# Lab FAT

# Submitted by: Alokam Nikhitha(19BCE2555)

**Question:**



**Dataset Used:**

Since the data set provide is very small . I used this data set given below

https://www.kaggle.com/katarzynanecka/carprices

**Procedure:**

* Firstly we are importing the Libraries
* We are importing the dataset using pandas
* Next we displayed the first few rows of the dataset.
* We identified Dependent and Independent variables in the dataset.
* Splitting the dataset in to Training and Testing sets(60% and 40% respectively).
* Feature Scalling the attributes.
* Next we have to find the MAE,MSE,RMSE of the with Linear Kernel of SVR
* Later we have to find the MAE,MSE,RMSE of the with Poly Kernel of SVR

**Code**

#Importing the libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

#Importing the Dataset

dataset = pd.read\_csv("CarPrices (1).csv")

dataset.head()

#Defining set of Dependent and Independent Attributes

X = dataset.loc[:, ['horsepower', 'peakrpm', 'citympg']]

y = dataset['price']

#printing Dependent Variables  
X

#printing Independent Variables

Y

#Splitting the dataset into training and test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.4,

random\_state=42)

#Feature Scaling

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

sc\_y = StandardScaler()

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

X\_test = sc\_X.transform(X\_test)

y\_train = sc\_y.fit\_transform(y\_train.values.reshape(-1, 1))

y\_test = sc\_y.transform(y\_test.values.reshape(-1, 1))

from sklearn.svm import SVR

y = y.ravel()

regressor1 = SVR(kernel = 'linear')

regressor1.fit(X, y)

y\_pred = regressor1.predict(X\_test)

y\_pred = sc\_y.transform(y\_pred.reshape(-1, 1))

from sklearn.metrics import mean\_absolute\_error

mean\_absolute\_error(y\_pred, y\_test)

from sklearn.metrics import mean\_squared\_error

mean\_squared\_error(y\_pred, y\_test)

from math import sqrt

sqrt(mean\_squared\_error(y\_pred, y\_test))

from sklearn.svm import SVR

y = y.ravel()

regressor2 = SVR(kernel = 'poly')

regressor2.fit(X, y)

y\_pred = regressor2.predict(X\_test)

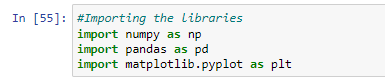
y\_pred = sc\_y.transform(y\_pred.reshape(-1, 1))

mean\_absolute\_error(y\_pred, y\_test)

mean\_squared\_error(y\_pred, y\_test)

sqrt(mean\_squared\_error(y\_pred, y\_test))

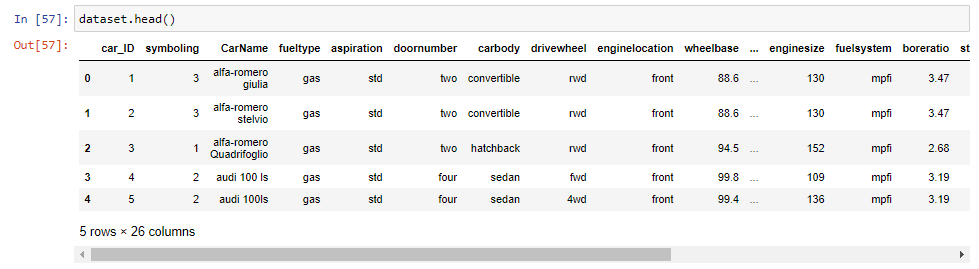
**Code Snippets and Output :**



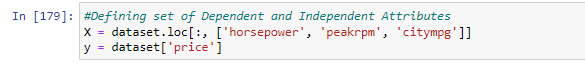
We imported the necessary libraries



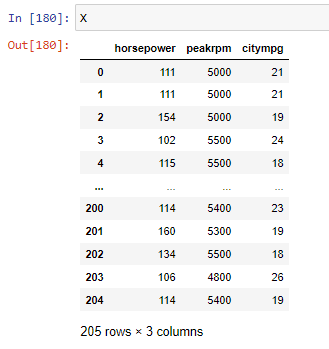
We are importing data to the workspace using pandas



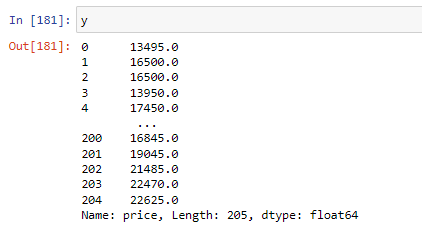
We are printing First Few data of the dataset



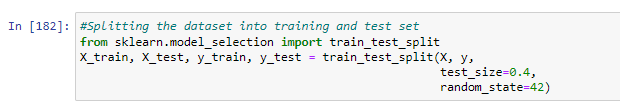
Defining set of Dependent and Independent Attributes



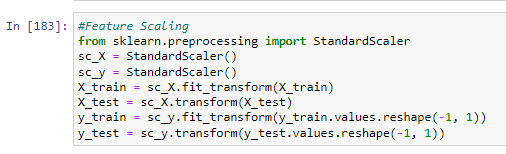
Printing Dependent variables



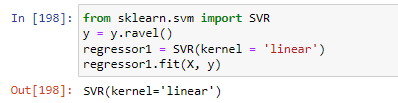
Printing Independent variables



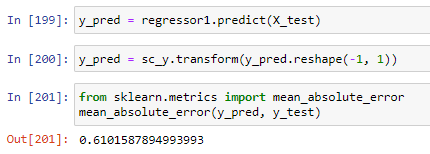
Splitting the dataset into Training and Testing dataset 60% and 40%.



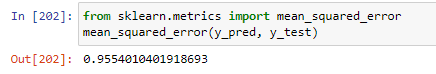
Feature Scaling the Attributes



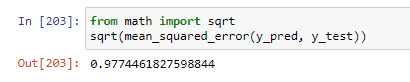
Defining Kernal Type as Linear



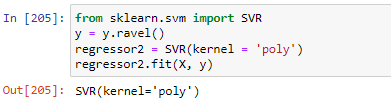
Calculating the Mean Absolute error of the linear kernel model



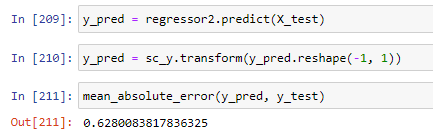
Calculating the Mean squared error of the linear kernel model



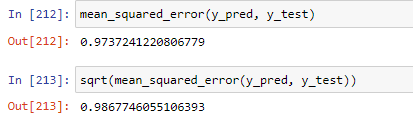
Calculating the Root Mean Square error of the linear kernel model



Defining poly Kernel



Calculating the Mean Absolute error of the POLY kernel model



Similarly Calculating the Meansquared error and Root mean squared error of the POLY kernel model

**Results and Conclusion**

SVR Linear

Mean absolute Error= 0.6101587894993993

Mean Squared error = 0.9554010401918693

Root mean Square error =0.9774461827598844

SVR Kernal

Mean absolute Error= 0.6280083817836325

Mean Squared error = 0.9737241220806779

Root mean Square error =0.9867746055106393