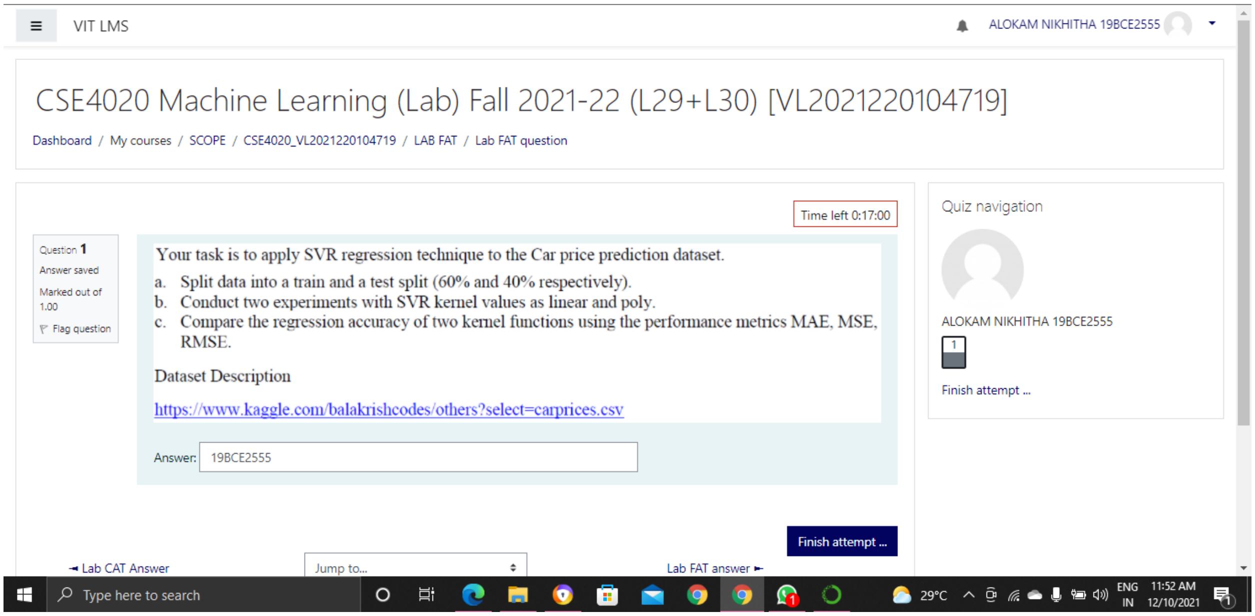
CSE 4020 - MACHINE LEARNING

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 :

```
In [55]: #Importing the libraries
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
```

We imported the necessary libraries

```
In [56]: #Importing the Dataset
         dataset = pd.read_csv("CarPrices (1).csv")
```

We are importing data to the workspace using pandas

```
In [57]: dataset.head()
```

Out[57]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	fuelsystem	boreratio	st
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	

5 rows × 26 columns

We are printing First Few data of the dataset

```
In [179]: #Defining set of Dependent and Independent Attributes
          X = dataset.loc[:, ['horsepower', 'peakrpm', 'citympg']]
          y = dataset['price']
```

Defining set of Dependent and Independent Attributes

In [180]: X

Out[180]:

	horsepower	peakrpm	citympg
0	111	5000	21
1	111	5000	21
2	154	5000	19
3	102	5500	24
4	115	5500	18
...	...	...	...
200	114	5400	23
201	160	5300	19
202	134	5500	18
203	106	4800	26
204	114	5400	19

205 rows × 3 columns

Printing Dependent variables

In [181]: y

```
Out[181]: 0      13495.0
1      16500.0
2      16500.0
3      13950.0
4      17450.0
        ...
200    16845.0
201    19045.0
202    21485.0
203    22470.0
204    22625.0
Name: price, Length: 205, dtype: float64
```

Printing Independent variables

In [182]:

```
#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)
```

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

```
In [183]: #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))
```

Feature Scaling the Attributes

```
In [198]: from sklearn.svm import SVR
          y = y.ravel()
          regressor1 = SVR(kernel = 'linear')
          regressor1.fit(X, y)

Out[198]: SVR(kernel='linear')
```

Defining Kernal Type as Linear

```
In [199]: y_pred = regressor1.predict(X_test)

In [200]: y_pred = sc_y.transform(y_pred.reshape(-1, 1))

In [201]: from sklearn.metrics import mean_absolute_error
          mean_absolute_error(y_pred, y_test)

Out[201]: 0.6101587894993993
```

Calculating the Mean Absolute error of the linear kernel model

```
In [202]: from sklearn.metrics import mean_squared_error
          mean_squared_error(y_pred, y_test)

Out[202]: 0.9554010401918693
```

Calculating the Mean squared error of the linear kernel model

```
In [203]: from math import sqrt
          sqrt(mean_squared_error(y_pred, y_test))

Out[203]: 0.9774461827598844
```

Calculating the Root Mean Square error of the linear kernel model

```
In [205]: from sklearn.svm import SVR
          y = y.ravel()
          regressor2 = SVR(kernel = 'poly')
          regressor2.fit(X, y)

Out[205]: SVR(kernel='poly')
```

Defining poly Kernel

```
In [209]: y_pred = regressor2.predict(X_test)

In [210]: y_pred = sc_y.transform(y_pred.reshape(-1, 1))

In [211]: mean_absolute_error(y_pred, y_test)

Out[211]: 0.6280083817836325
```

Calculating the Mean Absolute error of the POLY kernel model

```
In [212]: mean_squared_error(y_pred, y_test)

Out[212]: 0.9737241220806779

In [213]: sqrt(mean_squared_error(y_pred, y_test))

Out[213]: 0.9867746055106393
```

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