Import Library

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Load Dataset from Local Directory

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files CarPrice_Assignment.csv
• CarPrice_Assignment.csv(text/csv) - 26717 bytes, last modified: 4/9/2023 - 100% done
Saving CarPrice_As ment.csv to CarPrice_Assignment.csv
```

Importing the Dataset

```
dataset = pd.read_csv('CarPrice_Assignment.csv')
dataset = dataset.drop(['car_ID'], axis=1)
Xdata = dataset.drop('price', axis='columns')
numericalCols = Xdata.select_dtypes(exclude=['object']).columns
X = Xdata[numericalCols]
y = dataset.iloc[:, -1].values
ysvm = y.reshape(len(y), 1)
print(y)
print(ysvm)
```

```
[ 7957.   ]
[ 6229.   ]
[ 6692.   ]
[ 7609.   ]
[ 8921.   ]
[12764.   ]
[22018.   ]
[32528.   ]
[34028.   ]
[37028.   ]
[31400.5  ]
[ 9295.   ]
[ 9895.   ]
[11850.   ]
[12170.   ]
[15040.   ]
[15510.   ]
[18150.   ]
[18620.   ]
[ 5118.   ]
[ 7053.   ]
[ 7603.   ]
[ 7126.   ]
[ 7775.   ]
[ 9960.   ]
[ 9233.   ]
[11259.   ]
[ 7463.   ]
[10198.   ]
[ 8013.   ]
[11694.   ]
[ 5348.   ]
[ 6338.   ]
[ 6488.   ]
[ 6918.   ]
[ 7898.   ]
[ 8778.   ]
[ 6938.   ]
[ 7198.   ]
[ 7898.   ]
[ 7788.   ]
[ 7738.   ]
[ 8358.   ]
[ 9258.   ]
[ 8058.   ]
[ 8238.   ]
[ 9298.   ]
[ 9538.   ]
[ 8449.   ]
[ 9639.   ]
[ 9989.   ]
[11199.   ]
[11549.   ]
[17669.   ]
[ 8948.   ]
[10698.   ]
[ 9988.   ]
[10898.   ]
[11248.   ]
```

Splitting the dataset into the 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 = 0.20, random_state = 0)
X_trainsvm, X_testsvm, y_trainsvm, y_testsvm = train_test_split(X, ysvm, test_size = 0.20, random_state = 0)
```

Importing Machine Learning Algorithms

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
```

Initializing different Regression algorithms

```
from sklearn.preprocessing import StandardScaler

modelLR = LinearRegression()

poly_reg = PolynomialFeatures(degree=4)
X_poly = poly_reg.fit_transform(X_train)
modelPLR = LinearRegression()

modelRFR = RandomForestRegressor(n_estimators=10, random_state=0)

modelDTR = DecisionTreeRegressor(random_state=0)

modelSVR = SVR(kernel='rbf')

sc_X = StandardScaler()
sc_y = StandardScaler()
X_trainsvm = sc_X.fit_transform(X_trainsvm)
y_trainsvm = sc_y.fit_transform(y_trainsvm)
```

Training Regression algorithm

```
modelLR.fit(X_train, y_train)
modelPLR.fit(X_poly, y_train)
modelRFR.fit(X_train, y_train)
modelDTR.fit(X_train, y_train)
modelSVR.fit(X_trainsvm, y_trainsvm)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was pas
  y = column_or_1d(y, warn=True)
▾ SVR
SVR()
```

Predicting the test set for validation

```
modelLRy_pred = modelLR.predict(X_test)
modelPLRy_pred = modelPLR.predict(poly_reg.transform(X_test))
modelRFRy_pred = modelRFR.predict(X_test)
modelDTRy_pred = modelDTR.predict(X_test)
# modelSVRy_pred = sc_y.inverse_transform(modelSVR.predict(sc_X.transform(X_test)))
```

Evaluating the Model Performance

```
from sklearn.metrics import r2_score
print("Linear Regression Accuracy: {}".format(r2_score (y_test, modelLRy_pred)))
print("Polynomial Regression Accuracy: {}".format(r2_score(y_test, modelPLRy_pred)))
print("Random Forest Regression Accuracy: {}".format(r2_score(y_test, modelRFRy_pred)))
print("Decision Treee Regression Accuracy: {}".format(r2_score(y_test, modelDTRy_pred)))
# print("Support Vector Regression Accuracy: {}".format(r2_score(y_test, modelSVRy_pred)))
```

```
Linear Regression Accuracy: 0.815461783189196
Polynomial Regression Accuracy: -795.2039031869009
Random Forest Regression Accuracy: 0.898293680543916
Decision Treee Regression Accuracy: 0.8558001106879025
```

✓ 0s completed at 6:48 AM