Car Resale Value Prediction

Model Building

Check the Metrics of the Model

```
[34] from sklearn.linear_model import LinearRegression
```

```
[35] from sklearn.preprocessing import OneHotEncoder
     from sklearn.compose import make_column_transformer
     from sklearn.pipeline import make_pipeline
     from sklearn.metrics import r2_score
```

Creating an OneHotEncoder object to contain all the possible categories

```
[36] ohe=OneHotEncoder()
     ohe.fit(X[['name','company','fuel_type']])
```

```
OneHotEncoder()
```

Creating a column transformer to transform categorical columns

```
[37] column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_),['name','company','fuel_type']),
                                          remainder='passthrough')
```

Linear Regression Model

```
[38] lr=LinearRegression()
```

Making a pipeline

```
[39] pipe=make_pipeline(column_trans,lr)
```

Fitting the model

```
[43] pipe.fit(X_train,y_train)
```

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(categories=[array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6 2.0',
       'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series',
       'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d',
       'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat...
                                                                      array(['Audi', 'BMW', 'Chevrolet', 'Datsun', 'Fiat', 'Force', 'Ford',
       'Hindustan', 'Honda', 'Hyundai', 'Jaguar', 'Jeep', 'Land',
       'Mahindra', 'Maruti', 'Mercedes', 'Mini', 'Mitsubishi', 'Nissan',
       'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'],
      dtype=object),
                                                                      array(['Diesel', 'LPG', 'Petrol'], dtype=object)]),
                                                  ['name', 'company',
                                                   'fuel_type'])])),
                ('linearregression', LinearRegression())])
```

```
[41] y_pred=pipe.predict(X_test)
```

Checking R2 Score

```
[42] r2_score(y_test,y_pred)
```

```
0.6547602819063847
```

Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2_score

```
[43] scores=[]
     for i in range(1000):
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
         lr=LinearRegression()
         pipe=make_pipeline(column_trans,lr)
         pipe.fit(X_train,y_train)
         y_pred=pipe.predict(X_test)
         scores.append(r2_score(y_test,y_pred))
```

```
[44] np.argmax(scores)

     655
```

```
[45] scores[np.argmax(scores)]

     0.920087093218515
```

```
[46] pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))

     array([400642.51767152])
```