USED CAR PRICE PREDICTION

USING MACHINE LEARNING

APSTRACT

- Car price prediction especially when the car is used and not coming directly from the showroom, is both critical and important task. With increase in demand for used cars more and more car buyers are finding alternatives of buying cars.

DESIGN

- This project is taken from the Kaggle website: https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes

- This dataset contains information about already used cars. This dataset can be used for the price prediction to represent the use of regression Machine Learning model like Random Forest Regressor.

DATA

The dataset consists of 99187 rows with 11 featured including 11 categorical features as described below:

More about Features:

- Model: Model type.
- Year: Registration Year.
- Price: Price in euros.
- Transmission: Type of Gearbox.
- Mileage: Distance Used.
- Fuel Type: Engine Fuel.
- Tax: Road Tax.
- mpg: Miles per Gallon.
- Engine Size: Size in litres.
- Brand: Name of the car brand.

ALGORITHM

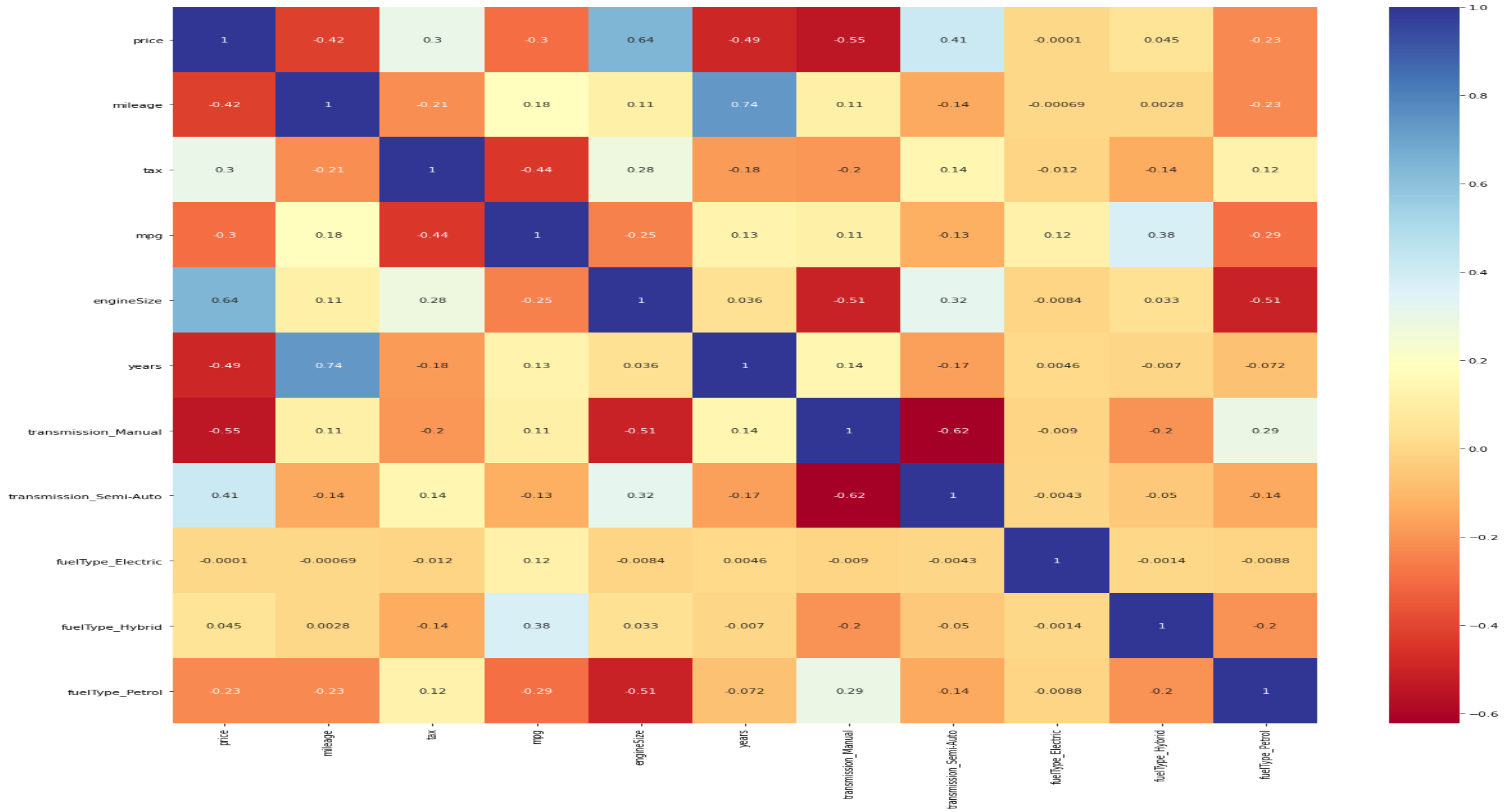
- Feature Engineering

Exploratory Data Analysis:

I have loaded the dataset and want to deal with all the categorical features and perform Exploratory Data Analysis(EDA) on the given dataset. For that I have used a technique called One-Hot Encoding to convert the categorical features into numerical ones. I have used the get_dummies algorithm to avoid dummy variable trap and perform one hot encoding. Then the libraries like seaborn and matplotlib are used for data visualization of the features. Correlation matrix also was built to understand how one feature relates to another.

CORRELATION MATRIX

- This matrix helps us in finding out how one feature is correlated to another using heatmap The thick blue indicates that it is positively correlated that means increase in the value of one feature results in increment of the value of the corresponding feature and the red indicates that the features are negatively correlated that is increase in value of one feature corresponds to decrease in the value of the corresponding feature. The value ranges from -1 to +1.


price
1
-0.42
0.3
-0.3
0.64
-0.49
-0.55
0.41
-0.0001
0.045
-0.23
mileage
-0.42
1
-0.21
0.18
0.11
0.74
0.11
-0.14
-0.00069
0.0028
-0.23
tax
0.3
-0.21
1
-0.44
0.28
-0.18
-0.2
0.14
-0.012
-0.14
0.12
mpg
-0.3
0.18
-0.44
1
-0.25
0.13
0.11
-0.13
0.12
0.38
-0.29
engineSize
0.64
0.11
0.28
-0.25
1
0.036
-0.51
0.32
-0.0084
0.033
-0.51
years
-0.49
0.74
-0.18
0.13
0.036
1
0.14
-0.17
0.0046
-0.007
-0.072
transmission_Manual
-0.55
0.11
-0.2
0.11
-0.51
0.14
1
-0.62
-0.009
-0.2
0.29
transmission_Semi-Auto
0.41
-0.14
0.14
-0.13
0.32
-0.17
-0.62
1
-0.0043
-0.05
-0.14
fuelType_Electric
-0.0001
-0.00069
-0.012
0.12
-0.0084
0.0046
-0.009
-0.0043
1
-0.0014
-0.0088
fuelType_Hybrid
0.045
0.0028
-0.14
0.38
0.033
-0.007
-0.2
-0.05
-0.0014
1
-0.2
fuelType_Petrol
-0.23
-0.23
0.12
-0.29
-0.51
-0.072
0.29
-0.14
-0.0088
-0.2
1
price
mileage
tax
mpg
engineSize
years
transmission_Manual
transmission_Semi-Auto
fuelType_Electric
fuelType_Hybrid
fuelType_Petrol
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

MODELS

- Linear regression, k-nearest neighbors, and random forest Regressor were used before settling on random forest as the model with high accuracy.

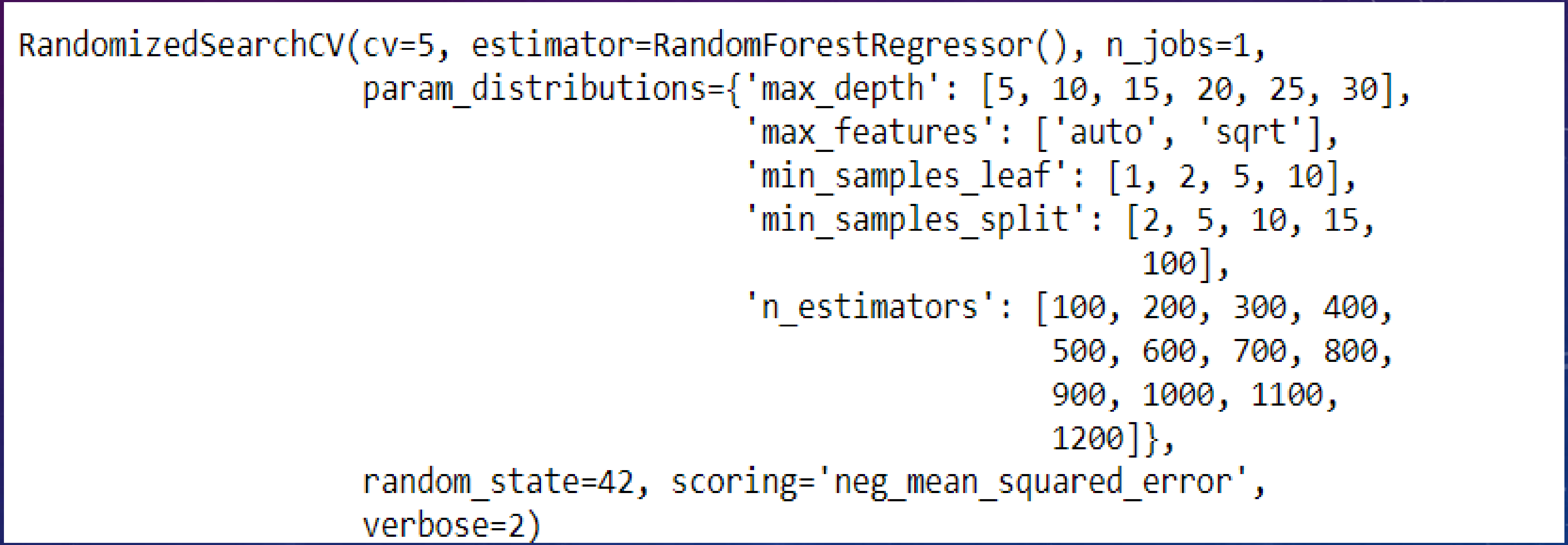
- After fitting our model into this ExtraTreeRegressor, I can find the feature importances of each one. The default values for the parameters max_depth, min_samples_leaf, etc , leads to a fully grown and unpruned trees which can potentially very large on some data sets. To reduce memory consumption, the complexity and the size of trees should be controlled by setting those parameter values.

MODEL EVALUATION AND SELECTION

- The entire training dataset of 59,400 records was split into 80/20 train vs. holdout.

- RandomizedSearchCV : RandomizedSearchCV implements a “fit” and a “score” method. It is the best algorithm for hyper parameter tuning that is for choosing the best parameters. This method have parameters like estimator (Random Forest Regressor), Param_distributions(RandomGrid) which is used to find best hyper parameters, Scoring (negative meansquared error). It helps us to find the best parameters.

RESULT OF HYPER PARAMETER TUNING



```
RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=1,
                   param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                        'max_features': ['auto', 'sqrt'],
                                        'min_samples_leaf': [1, 2, 5, 10],
                                        'min_samples_split': [2, 5, 10, 15,
                                                              100],
                                        'n_estimators': [100, 200, 300, 400,
                                                         500, 600, 700, 800,
                                                         900, 1000, 1100,
                                                         1200]},
                   random_state=42, scoring='neg_mean_squared_error',
                   verbose=2)
```

TOOLS USED

- ➤Numpy and Pandas for data manipulation.
- ➤Matplotlib and seaborn for Data Visualization and Plotting .
- ➤Scikitlearn for Modelling.
- ➤Random Forest Regression algo for prediction.

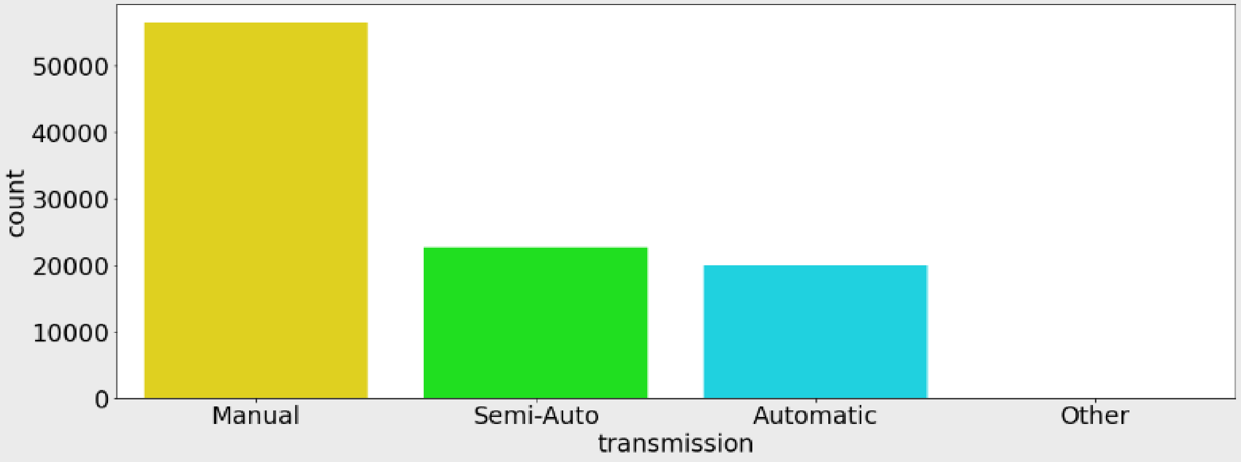
ACCURACY:

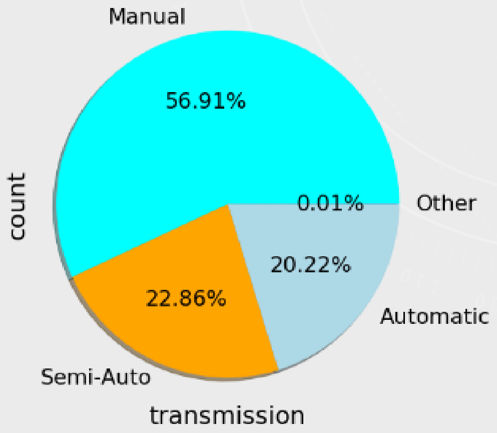
R2 score = 0.92

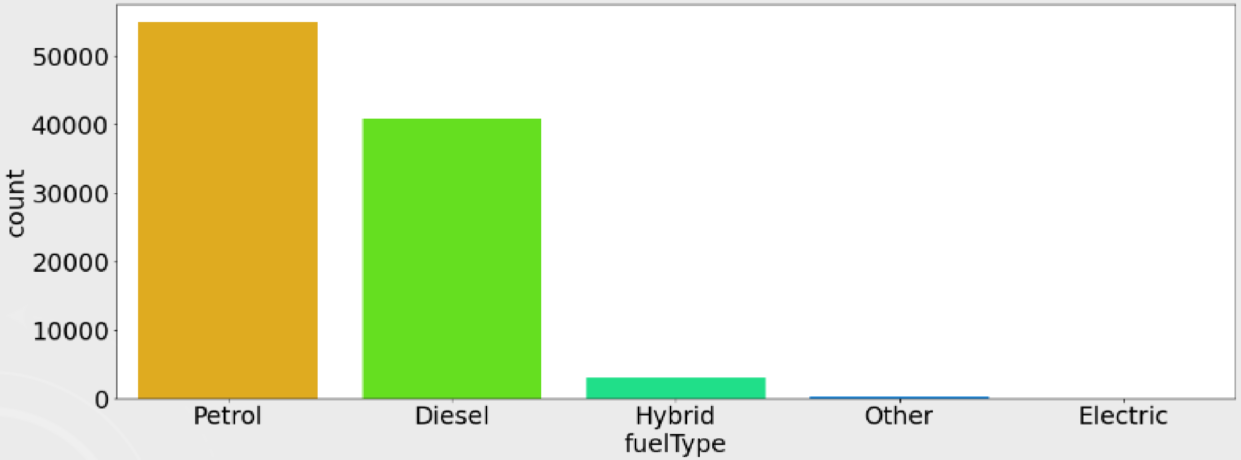
MAE: 1705.0133318061835

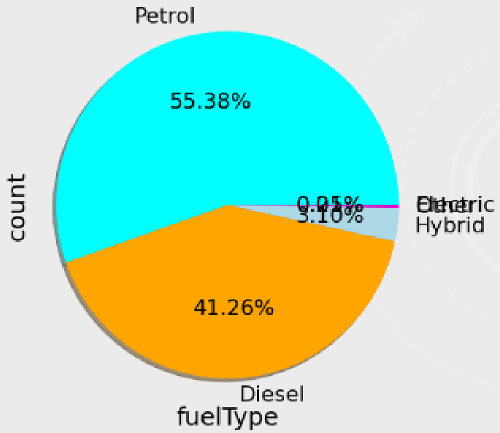
MSE: 7715862.161463453

COMMUNICATION



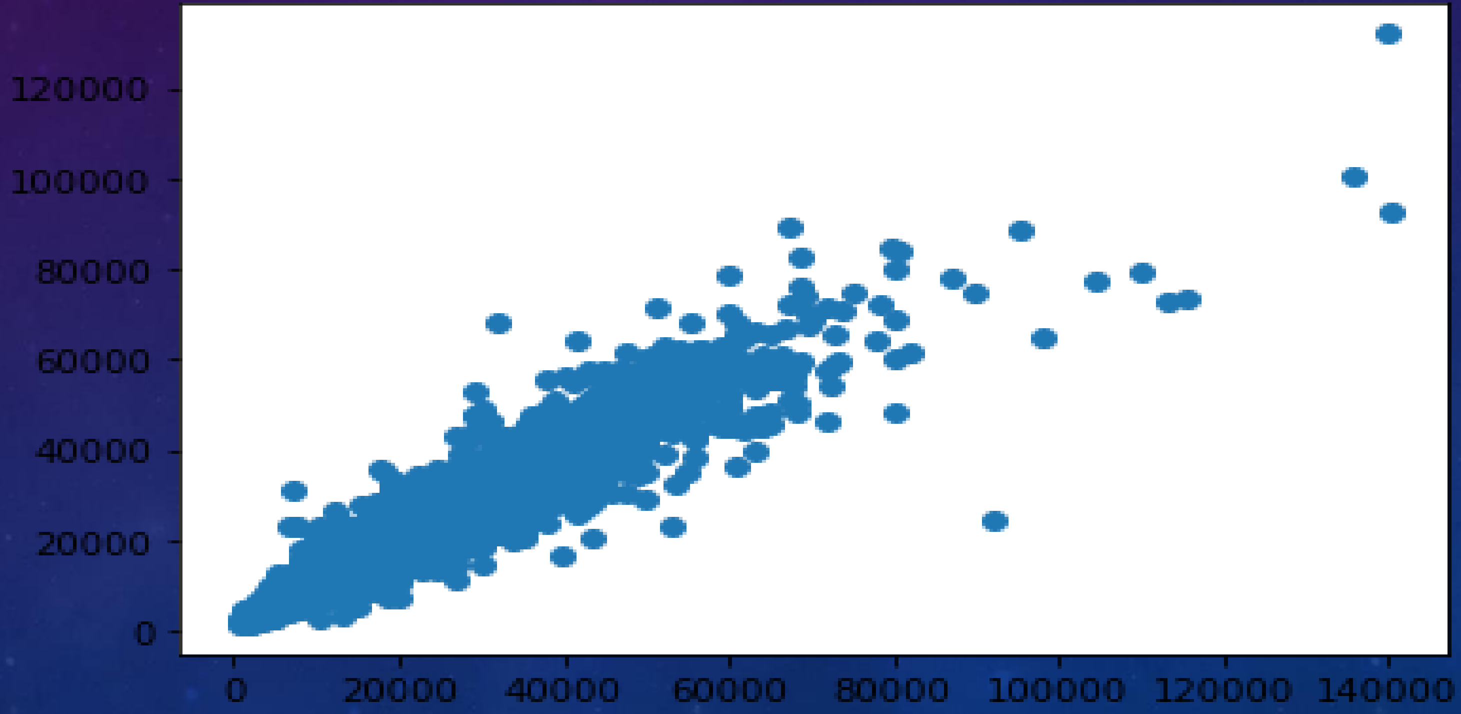






PREDICTION

- The linear plot that we got after creating the model. Thus shows the linear behaviour of the model which indicates that this prediction is pretty much good.



THANK YOU