Project Aim:

Predicting the cost of used cars from the data collected from various sources and distributed across various locations in India.

Introduction:

Data was given in a '.csv' file. It was analyzed and an algorithmic ML model has been made using Python libraries of 'Pandas', 'Numpy' and 'Sklearn', whereas 'Seaborn' and 'Matplotlib.pyplot' were used for creating graphs for data exploration. The outputs are presented in terms of insight. The python program along with the output is also enclosed as an appendix to this report.

Description of Independent and Dependent Variables:

Independent Variables:

Name: The brand and model of the car.

Location: The location in which the car is being sold or is available for purchase.

Year: The year or edition of the model.

Kilometers_Driven: The total kilometers driven in the car by the previous owner(s) in KM.

Fuel_Type: The type of fuel used by the car.

Transmission: The type of transmission used by the car.

Owner_Type: Whether the ownership is Firsthand, Second hand, or other.

Mileage: The standard mileage offered by the car company in kmpl or km/kg

Engine: The displacement volume of the engine in cc.

Power: The maximum power of the engine in bhp.

Seats: The number of seats in the car.

New_Price: The price of a new car of the same model.

Dependent Variables:

Price: The price of the used car in lakhs INR.

Data Cleaning and Exploration:

The Data Cleaning part of EDA has been done by Ashwin Raj.

The list below has the name of people and the variables they have worked on, for Exploratory Data Analysis:

1. Sarthak Giri - Mileage, Engine
2. Abhipsa Panda- Location, Owner_Type
3. Ashwin Raj- Kilometers_driven
4. Muskan Khurana- Year
5. Sneha- Transmission, Seats
6. Chitrabanu-Power
7. Anshul- Fuel_Type (and all other variables)

ML Model Information

All information about the Machine Learning model to be used and the related steps are shown below.

The variables in the train data set can be considered as follows:

1. `Name`: We can divide this variable into 3 more variables (Brand, Car_Name, Model), and use label encoding or one-hot encoding to create categorical data points.
2. `Location`: We can create categorical data points in this variable using one-hot encoding or label encoding.
3. `Year`: We can change it into an ordinal variable, using label encoding or one-hot encoding.
4. `Fuel_Type`: One-hot encoding or label encoding can be used to convert it into a categorical variable.
5. `Owner_Type`: Use label encoding or one-hot encoding to convert it to a categorical variable.
6. `Transmission`: Use label encoding or one-hot encoding to convert it to a categorical variable.

The important variables are:

```
Name, Location, Year, Fuel-Type, Engine, Power, Seats, Owner-Type,
Transmission
```

The variables that are not important:

```
Kilometers_Driven, Mileage
```

Error Metrics that have been used:
R^2, Adjusted R^2, RMSE

The Methods used to clean the data and train the model:

1. Converted the `Year` variable into a label encoded variable (ordinal), and converted the `Location` variable into a one-hot encoded variable. As of now, the data set has 21 columns, since the 10 unique city-name variables were created from the `Location` variable, and the original `Location` variable was removed.
2. In the `Name` variable, separated out the brand names, and converted them into one-hot encoded variables, creating 28 more columns, and the original `Name` variable was removed.
3. Created one-hot encoded variables out of the `Fuel_Type` variable, adding 3 more columns, and the original `Fuel_Type` column has been removed.
4. Created one-hot encoded variables out of the `Seats` variable, adding 7 more columns, and the original `Seats` column has been removed.
5. The `Transmission` variable has been one-hot encoded, which has created 1 more variable, and the original `Transmission` column has been removed.
6. The `Owner_Type` variable has been one-hot encoded, which has created 3 more columns, and the original `Owner_Type` variable has been removed.

Below is a list of people who have worked on the variables mentioned in the above 6 steps:

a) Sarthak Giri - `Year`, `Location`, `Name`, `Seats`
b) Simar Preet Kaur - `Owner_Type`, `Fuel_Type`
c) Anshul Srivastava - `Transmission`

(In all one-hot encoded variables, the first variable has been dropped, to prevent the problem of multicollinearity.)

7. Removed the `Kilometers_Driven` and the `Mileage(kmpl)` variables, since they don't affect the predicted price of a car in a practical scenario. The kilometers driven is an arbitrary variable, and there is no general relationship between the value of a car and the kilometers that it has been driven for. As for the mileage, that also is an arbitrary variable, and the kmpl or km/kg mileage rating of a car is an approximate value and is not really sought after by a car buyer.

8. Feature Scaled the `Engine(CC)` and the `Power(bhp)` variable, either using min-max scaling or standard scaling.
   ● Robust Scaling has been used to scale the values in these two variables.

9. Split the training data further into train and test data sets (probably a 70-30% split).

10. Trained the model on the training data (from the train-test split), validate it using k-folds cross-validation, then apply the trained and validated model on the test data (from the train-test split), and calculate the error metrics. (The error metrics that need to be calculated are mentioned here: ML Models Information).
   ● The model used for training and testing is Linear Regression

Conclusion:

After splitting the training data set into training and test sets, and applying the linear regression model on the training set, we got these error metrics:

1. R Square Score: `0.779775569159032`
2. Adjusted R Square Score: `0.7726757193956948`

   Both the r-square and adjusted r-square values show that the model explains a good amount of the variance on the data.

3. Root Mean Square Error: `5.17564264835917 Lakh INR`

   The low value of RMSE indicates that this model has a good prediction power (lower the RMSE score, higher the prediction power)

We can thus conclude that our model can make good predictions about the prices of used cars.