# Car Price Prediction

## Problem

The Indian used car market is vast. Accurately pricing a used car is crucial. The goal is to build a machine learning model to predict the selling price of a used car based on its key features.

## Dataset

Dataset was downloaded from Kaggle on Indian Car Prices in second hand market by CarDekho.
Link: https://www.kaggle.com/datasets/manishkr1754/cardekho-used-car-data

The original dataset contains following features:

- Car_name
- Brand
- Model
- Vehicle_age
- Km_driven
- Seller_type
- Fuel_type
- Transmission_type
- Mileage
- Engine
- Max_power
- Seats
- Selling_price

**Total rows: 15411**

## EDA (Exploratory Data Analysis)

Steps taken for EDA:

- Reading the CSV dataset in pandas dataframe
- Dropping unnecessary columns (brand, model), and duplicate rows / null value rows
- Basic dataframe operations (df.head() , df.info(), df.describe(), df.shape, df.isnull().sum() )
- Univariate Analysis
    - Bar plots for discrete features

    - KDE plots for continuous numeric features.
- Outlier Analysis and Removal.
- Bivariate Analysis
    - Scatter plots (Other features vs selling price)
    - Bar chart (Other features vs selling price)
    - Box plots (Other features vs selling price)
- Correlation Heatmap

## EDA Insights & Findings

- The dataset has no null values or duplicate rows.
    - This means dataset providers have cleaned the dataset.
- ‘Brand’ and ‘model’ these 2 columns can be dropped for our objective because the car name already contains this data, so no information loss.
- Count plots reveal that:
    - Majority of the cars use Petrol or Diesel as fuel. CNG, LPG and electric cars are very less.
    - On their platform, there are more dealers than individual sellers.
    - Majority of the cars are Manual transmission cars rather than Automatic.
- KDE (Kernel Density) plots reveal that:
    - Selling price distribution is very skewed due to some very expensive cars
    - Vehicle age is also skewed due to some cars older than 15 years. Legally, this should not be allowed as these vehicles should be scrapped after 15 years. But it seems they must have refurbished the vehicles.
    - Km_driven is also skewed due to some outliers
- Outliers
    - As seen from the various charts, many features are skewed due to some readings and this may affect our model performance.
    - I have removed the rows with outliers in following columns: Selling_price, km_driven, vehicle_age.
    - The decision of removing the rows rather than capping them was taken because the dataset already consists of a lot of rows. And capping might create false information.
    - After removing the outliers, The KDE shows a near-normal distribution of these features.
- Bivariate Analysis reveal that:
    - Cars between the age range of 1 year to 4 years have higher selling prices than others.

        - Selling price gradually decreases with vehicle_age.
        - Diesel cars are generally more expensive than petrol cars.
        - Automatic cars are more expensive than Manual cars.
        - Selling price does not vary too much depending on the type of seller.
        - Petrol cars are on average cheaper than diesel cars but there are a lot of petrol cars being more expensive than diesel cars, probably these cars are luxury cars.
- Correlation Heatmap
    - Heatmap shows most features are not dependent on each other.
    - Features like Engine BHP, Max power and km_driven are the most critical features affecting the selling price of the car.

# Machine Learning

I was assigned to make 2 machine learning models:

- Linear Regression
- Random Forest Regressor.

## Approach 1: Using the cleaned dataset.

**Preprocessing**

- In this approach, I used the cleaned dataset from the EDA notebook.
- Mapped 92 unique cars to numbers because car_name was string.
- Dropped columns of engine_power, seats. As most cars have 5 or 7 seats.
- Performed One-Hot encoding on Fuel_type, Seller_type and Transmission_type features.
- Performed standard scaling on numeric features.
- Split the dataset into X and y. X means input features and y means selling_price.
- Split the dataset into a training set and testing set. (80-20 split with random_state)

**Model**

- Linear Regression model
    - **R2 score: 0.6338318359571987**
    - Mean Absolute Error: 0.47139769343469
- Polynomial Regression model
    - **R2 score: 0.7095315262640465**
    - Mean Absolute Error: 0.4019474634082295
- Random Forest Regressor
    - **R2 score: 0.878099518924329**
    - Mean Absolute Error: 0.253339180640986

    - Root mean squared error: 0.35344380742518183

**Summary**

- Linear regression model performed poorly.
- It did not pass the criteria of 80% r2_score.
- Tried Polynomial Regression but it did not perform well.
- Trained Random Forest Regressor. It performed better than earlier models and passed 80% criteria of r2_score.

## Approach 2: Added more columns

**Preprocessing**

- Due to the last model's poor performance, I decided not to drop engine_power and seats columns from the dataset.
- The reason behind it is that these are quantitative data which helps a lot in model training and we had more qualitative columns than quantitative columns.
- Did not scale the selling price to get more ideas from MAE and RMSE errors.

**Model**

- Linear Regression model
    - **R2 score: 0.6651094164667595**
    - Mean Absolute Error: 278351.6282614204
- Polynomial Regression model
    - **R2 score: 0.560180478279164**
    - Mean Absolute Error: 194337.91357969135
- Random Forest Regressor
    - **R2 score: 0.9239521186971583**

**Summary**

- Model performance improved for Random Forest.
- Model performance worsened for Regression models.

## Approach 3: Car_name feature handling

**Preprocessing**

- With Car_name being binary_encoded
- It means this time I have not dropped the column of car name but instead converted it into binary encoding.
- Binary encoding is not as sparse as one-hot encoding.
- It was important not to drop the car name from the dataset as Car name is very important feature
- In the real world, the selling price would heavily depend on the car model.

**Model**

- Linear Regression model
    - **R2 score: 0.667671661220389**
    - Mean Absolute Error: 273522.354208716
    - Root Mean Squared Error: 500170.6927588081
- Polynomial Regression model
    - **R2 score: 0.5154347721380046**
    - Mean Absolute Error: 179529.70532382157
    - Root Mean Squared Error: 603963.47558098
- Random Forest Regressor
    - **R2 score: 0.9295917681749251**
    - Mean Absolute Error: 101394.95823549716
    - Root Mean Squared Error: 230221.70796438114

**Summary**

- Model does not improve.
- In fact, the performance gets worse for Polynomial Regression.

## Approach 4: Capping outliers instead of removing

**Preprocessing**

- With original dataset and improved cleaning.
- This time, I have handled outliers from every continuous numeric column.
- Instead of removing the outliers, I have capped these outliers to make sure I don't lose any data.

**Model**

- Linear Regression Model
    - **R2 score on test set: 0.8343334568648284**
    - Mean Absolute Error: 118417.15082832648
    - Root Mean Squared Error: 155478.8344624024
- Random Forest Regressor
    - **R2 score on test set: 0.9321263650265217**
    - Mean Absolute Error: 65743.98005673103
    - Root Mean Squared Error: 99518.63840469492

**Summary**

- Finally got the required r2_score in both the models.
- The MAE in Random Forest Model exceeds expectations.
- The training score and testing score does not vary in any models which shows there was **no overfitting.**

## Model Selection

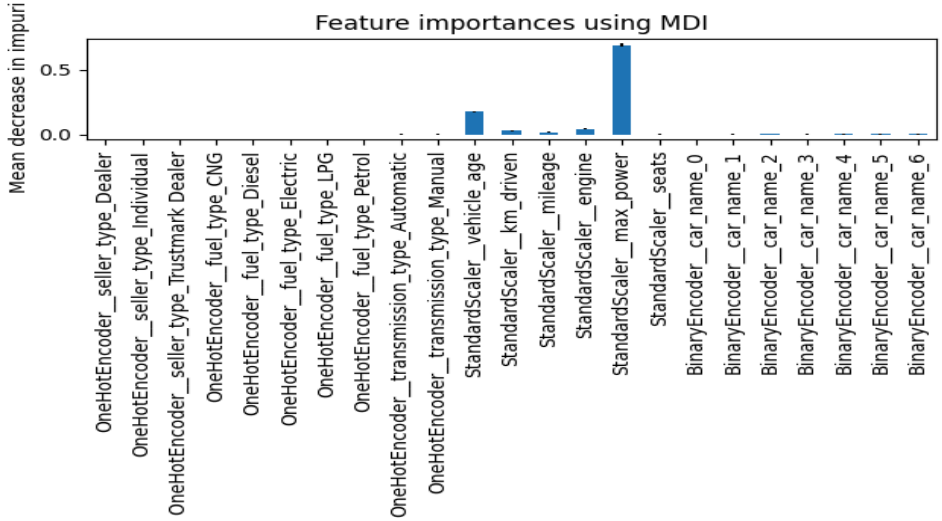
- As seen in above results, **Random Forest Regressor** outperforms Linear Regression every time.
- Finally, on the last approach, Linear Regression gives r2_score upwards of 80%.

## Feature Importance

- Feature Importance means analysis of which feature is more important in decision trees.
- Performed Feature importance analysis using these 2 methods:
    - *Built-in Feature Importance: This method utilizes the model's internal calculations to measure feature importance, such as Gini importance and mean decrease in accuracy. Essentially, this method measures how much the impurity (or randomness) within a node of a decision tree decreases when a specific feature is used to split the data.*
    - *Permutation feature importance: Permutation importance assesses the significance of each feature independently. By evaluating the impact of individual feature permutations on predictions, it calculates importance.*

## Insights from Feature Importance

- Performed Feature Importance analysis on Random Forest model to determine which features influence selling_price the most.
- Turns out that Engine_max_power and vehicle_age are the most influential features for selling_price.



## Limitations

- The dataset is complex, but the Linear regression model is not complex.
- Linear Regression model assumes the linear relationship between independent and dependent features, which is not the case here.
- Additionally, we capped the outliers for every numerical feature. Which is risky because:
    - Suppose a car is 25 years old. Our preprocessing makes it 15-17 years old. Which is a huge difference.
    - Similar capping was done on selling_price as well.
- In a way, we have given wrong information to get better r2_scores.
- Models may be limited to the capped max limit of selling_price because no training data exceeds that.
- Models may also perform poorly on very old cars or very powerful cars.

## Potential next steps

- Neural networks may perform better than these models.
- Better feature engineering may also help boost the performance. Like adding a brand value section, current market price of the car.
- Current condition score determined from various parameters like regular servicing, accidents or not, etc.