CAR PRICE PRESECTION REPORT

ABSTRACT:

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.

➤ There is a need of accurate price prediction mechanism for used cars. Prediction techniques of machine learning can be helpful in this regard.

➤ So, we used the car dataset from Kaggle website and built a model for predicting the price of used cars using Random Forest Regressor algorithm.

➤ Many companies are making ads for such kind of things and we are using the dataset provided by the KAGGLE website for our prediction.

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.

We are basically combining the following car brands (audi, bmw,ford, hyundi, merc, skoda, toyota, Vauxhall, vw) into a single dataset.

Algorithms:

Feature Engineering

Exploratory Data Analysis: We 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 we 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.

MODELS:

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

After fitting our model into this ExtraTreeRegressor, we 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.

ACCURACY:

R2 score = 0.92
MAE: 1705.0133318061835
MSE: 7715862.161463453

TOOLS USED:

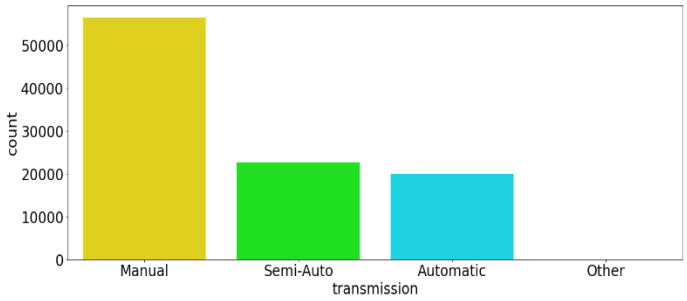
➤NumPy and Pandas for data manipulation.

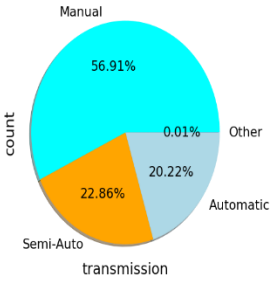
➤Matplotlib and seaborn for Data Visualization and Plotting.

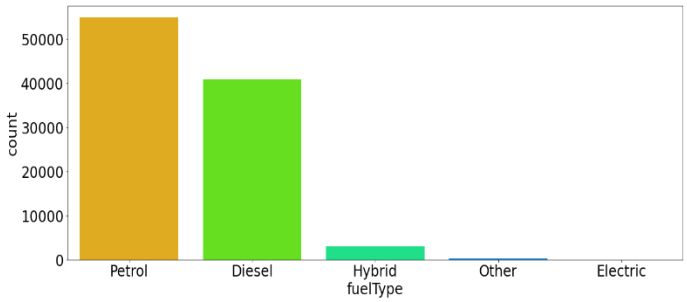
➤Scikitlearn for Modelling.

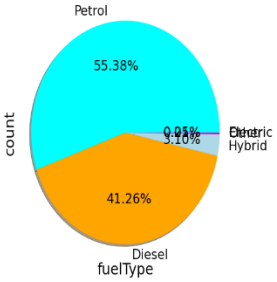
➤Random Forest Regression algo for prediction.

Communication:









PREDECTION:

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.