ECE 9063 Data Analytics Foundations

Assignment 1: Forecasting

Student Name: Jianping Ye
Student Number: 250887769
Instructor: Katarina Grolinger

Problem Statement

The used car market is a perfect place for finding cars in decent conditions and with fair prices. It is also the reason that the market has been growing in recent years. However, it is difficult to choose the opportune moment to buy or sell as the price fluctuates constantly. And there are many factors contributing to the price fluctuations. For instance, cars have diverse conditions and the market trend is not stationary all the time. It will be beneficial for both buyers and sellers if we could make a model to predict the value of cars such that they can make a more confident decision. With the help of a suitable model, buyers will be able to make sure the car is worthy of its price, and sellers can get a more accurate price estimation in accordance with other cars having similar conditions. In this report, the forecasting problem is defined as follow: predict the price of a used car in the current year given a set of relevant information.

Dataset Description

Link to the data: https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes

These datasets list scraped data of used cars in the British market and are separated into files specific for each car manufacturer. In this report, the dataset selected is "Audi.csv". It contains 9 attributes and 10668 samples. The dataset is suitable for this assignment as it has adequate attributes and samples. With over 10,000 samples, it is easier to strike a balance between computational time and reliability of the model . The attributes are listed below:

- Model: The model code of the car
- Year: registration year of the car
- Price: price on the market
- Transmission: type of gearbox, either manual, automatic, or semi-auto
- Mileage: distance used so far
- fuelType: type of fuel the engine uses, either diesel, petrol, hybrid, or other
- tax: road tax
- mpg: miles per gallon
- engineSize: size of engine in litres

Noticeably, model, transmission, and fuelType have nominal data that needs to be transformed into numerical values. All the attributes in the dataset are considered in the model as they are all important factors while estimating the price of cars in the real-world.

Algorithms Overview

The first algorithm is support vector regression (SVR). Support vector regression adheres to the basic principle of support vector machine, which is the maximum margin characteristic, but it is used for regression instead. Linear kernel is used.

The second algorithm is decision tree regression. The algorithm proceeds incrementally as breaking down the data into smaller subsets and build the associated sub-trees from them. At the end, a tree structure with decision nodes and leaf nodes is constructed. However, one major issue with decision tree regression is that it is very prone to overfitting. At the result comparison section, we will inspect whether this problem arises.

The third algorithm is random forest regression. Random forest regression utilizes the idea of ensemble learning, which is a technique that can take advantages from multiple machine learning algorithms such that it can produce a more accurate prediction.

Before applying the above-mentioned algorithms, it is necessary to normalize our dataset as it is a common requirement of many machine learning algorithms and it is also considered good practice. The normalization technique used is standardization, which will make the data have zero mean and unit variance.

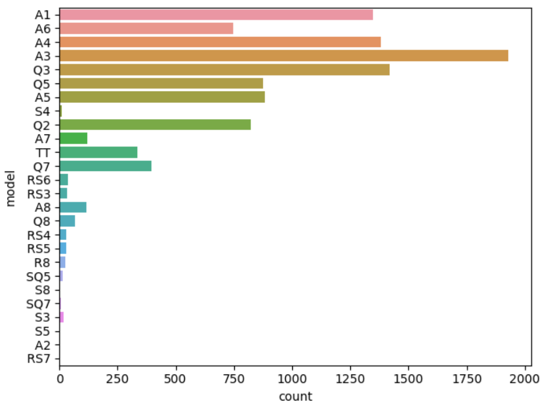
Detailed Procedures

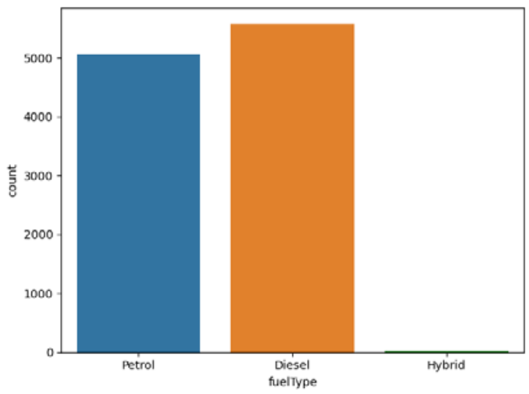
1. <u>Exploratory Data Analysis</u>

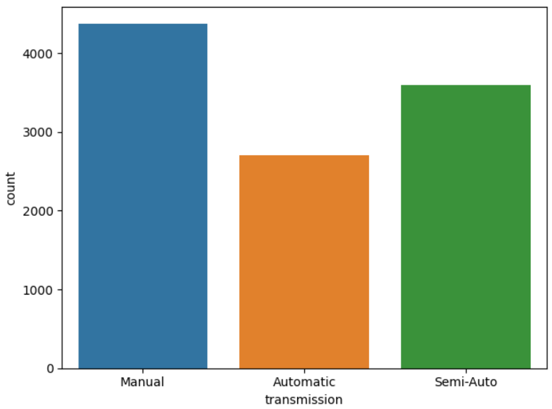
```
      model  year  price transmission  mileage fuelType  tax   mpg  engineSize
0        A1  2017  12500       Manual    15735   Petrol  150  55.4         1.4
1        A6  2016  16500    Automatic    36203   Diesel   20  64.2         2.0
2        A1  2016  11000       Manual    29946   Petrol   30  55.4         1.4
3        A4  2017  16800    Automatic    25952   Diesel  145  67.3         2.0
4        A3  2019  17300       Manual     1998   Petrol  145  49.6         1.0
...     ...   ...    ...          ...      ...      ...  ...   ...         ...
10663    A3  2020  16999       Manual     4018   Petrol  145  49.6         1.0
10664    A3  2020  16999       Manual     1978   Petrol  150  49.6         1.0
10665    A3  2020  17199       Manual      609   Petrol  150  49.6         1.0
10666    Q3  2017  19499    Automatic     8646   Petrol  150  47.9         1.4
10667    Q3  2016  15999       Manual    11855   Petrol  150  47.9         1.4
```

After importing the csv file, I printed the data to have an overview of its structure.

Clearly, model, transmission, and fuelType has nominal values. I plotted the counts of each type of observation versus that attribute.

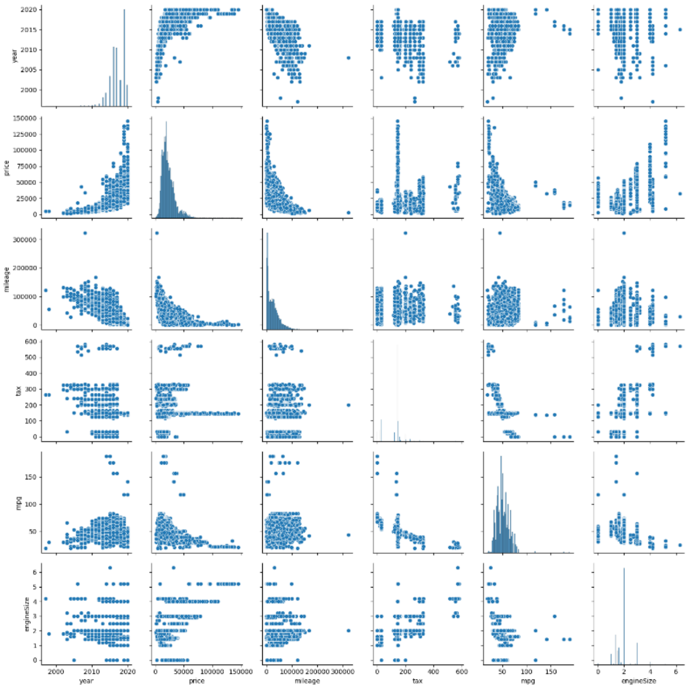







```
model           0
year            0
price           0
transmission    0
mileage         0
fuelType        0
tax             0
mpg             0
engineSize      0
```

It is observed that there are finite number of possibilities in each attribute, so one-hot encoding is selected as encoding scheme. And there is no null data in each attribute, which is very clean.



From the pair plot graph, we can see the data is dense (No zeros). Therefore, standardization will be used to normalize the data.

2. <u>Preprocessing</u>

	model	price	transmission	mileage	...	tax	mpg	engineSize	age_of_car
0	A1	12500	Manual	15735	...	150	55.4	1.4	3
1	A6	16500	Automatic	36203	...	20	64.2	2.0	4
2	A1	11000	Manual	29946	...	30	55.4	1.4	4
3	A4	16800	Automatic	25952	...	145	67.3	2.0	3
4	A3	17300	Manual	1998	...	145	49.6	1.0	1
...	...	...	...	...	...	...	...	...	...
10663	A3	16999	Manual	4018	...	145	49.6	1.0	0
10664	A3	16999	Manual	1978	...	150	49.6	1.0	0
10665	A3	17199	Manual	609	...	150	49.6	1.0	0
10666	Q3	19499	Automatic	8646	...	150	47.9	1.4	3
10667	Q3	15999	Manual	11855	...	150	47.9	1.4	4

Firstly, compute a new attribute called “age_of_car” by subtracting 2020 from the ‘year’ attribute. The result is number of years that car has been used since its registration. I believe the ‘age of car’ will be more informative and intuitive than its original.

In addition, use one-hot encoding to transform categorical features into numerical features. Moreover, separate the dataset into X (features) and Y (target) set. X set is independent variables including all the attributes except ‘price’. Y set is the target variable --‘price’ attribute. Lastly, apply standard scaling (Standardization) on X set to normalize the input data. At this stage, we can split the normalized features and target into X_train, X_test, Y_train, and Y_test. The test set size is 20% of the entire data.

3. Modeling

By using sklearn, we can easily create regression model to fit the training data. When creating the regression model instance, all the optional arguments are set to default to keep simplicity. All three algorithms will expect two arguments to pass into the function, one being X_train and the other being Y_train. This corresponds to the idea of train the model on the training set. After fitting the model, we can predict the corresponding value of X_test by applying the fitted parameters, then compare the predicted results with the actual target values, Y_test.

4. Accuracy & Evaluation

Different metrics are applied to reveal how well the models fit the data. Firstly, I will compute the mean absolute error (MAE). The rational is that MAE gives an intuitive measurement of the distance between predicted and actual values. To change the perspective, I also used the mean absolute percentage error (MAPE) to show the error as percentage values. However, these two metrics do not penalize errors that are bigger than others. Therefore, root mean squared error (RMSE) will be our primary error metric. In RMSE, bigger errors are penalized much heavier than the smaller errors as the error is squared. Both training errors and test errors are reported. Generally, it is anticipated that the test error will be larger than the training error. We will use test error as an indication of how well the model fits.

5-fold cross validation is also used to evaluate how well the model generalize on new data. The error metric used is RMSE. The entire dataset is split into 5 subsets and in each cross-validation iteration, one non-repeatable subset is selected as validation and the other 4 subsets is used for training. The validation error in each iteration is stored to allow computation of overall error for the 5 folds by simply taking average value.

Comparison of Results

```
      predicted  actual
2049    13990.0   14998
5609    24995.0   21950
7638    26990.0   28990
1603    26995.0   25489
5953    32490.0   30950
...         ...     ...
49      31500.0   23700
9999    17498.0   18000
2580    46500.0   45995
4139    30990.0   30500
9795    10495.0    8400

[2134 rows x 2 columns]
Train Set MAE: 53.28
Train Set RMSE: 393.37
Train Set MAPE: 0.17%

Test Set MAE: 1905.37
Test Set RMSE: 3337.47
Test Set MAPE: 8.89%

Cross Validation RMSE: [2801.9  3516.5  3451.44 3621.47 3352.77]
Cross Validation Overall RMSE: 3348.82
```

Decision Tree Regression

```
       predicted  actual
10442   10173.80    9990
2907    21970.77   22382
7388    27073.72   28990
3016    25747.18   30777
7890    16811.65   14950
...          ...     ...
8606    27558.58   31450
8977    15860.76   12900
3673    17311.25   16750
1034    22107.21   21996
6867    11499.67    9547

[2134 rows x 2 columns]
Train Set MAE: 3166.68
Train Set RMSE: 6084.54
Train Set MAPE: 12.72%

Test Set MAE: 3420.31
Test Set RMSE: 6690.45
Test Set MAPE: 12.94%

Cross Validation RMSE: [5031.92 6492.19 8603.58 5587.89 5068.19]
Cross Validation Overall RMSE: 6156.75
```

Support Vector Regression

Random Forest Regression

```
      predicted  actual
2049   14524.90   14998
5609   23116.90   21950
7638   27593.79   28990
1603   26962.43   25489
5953   32277.05   30950
...         ...     ...
49     32990.44   23700
9999   17077.75   18000
2580   45753.96   45995
4139   31138.60   30500
9795   10328.72    8400

[2134 rows x 2 columns]
Train Set MAE: 595.64
Train Set RMSE: 960.6
Train Set MAPE: 2.79%

Test Set MAE: 1503.79
Test Set RMSE: 2429.17
Test Set MAPE: 7.01%

Cross Validation RMSE: [2257.31 2336.83 2805.59 2653.35 2983.32]
Cross Validation Overall RMSE: 2607.28
```

We can see that random forest regression has the lowest RMSE, MAE and MAPE in the test set. It also has the lowest overall RMSE in cross validation. The lowest RMSE indicates that large errors are fewer in random forest regression. In contrast, support vector regression has very high RMSE which means large errors are more prevalent.

Decision tree regression has stunningly low RMSE, MAE and MAPE (0.17%) in the training data. In comparison, RMSE in test set is almost 10 times of that in training set. They are likely to be the evidences of overfitting. In the test set and cross validation set, the error becomes larger and closer to results obtained from other two models.

As anticipated, all error metrics in test set are larger than those of the training set in all three algorithms. Since the model is not fitted to the test data, then it may not perform as good as it was in the training stage.

To sum up, random forest regression has the overall best performance. What's better still, its cross-validation error is stable, which makes it a suitable model on this dataset and our forecasting problem. Support vector regression has very high RMSE as it is penalized more by larger errors. And decision tree regression potentially suffers from overfitting, thus it is not recommended on this problem.