# Random Forest for Classification to evaluate Customer acceptability of Cars

Jeshlin Donna J
*Department of Metallurgical and Materials Engineering*
*Indian Institute of Technology Madras*
Chennai, India
mm20b029@smail.iitm.ac.in

***Abstract*—The key task of this paper is to evaluate customer acceptability of Cars. The aim is use and apply Random Forest and perform Exploratory data analysis to uncover previously unknown or hidden facts in the data set available, visualize them using Data Visualization and make predictions on the validation data. This is followed by model evaluation on the Random Forest model. The task is to conclude the study of which types of cars are more likely to have higher customer acceptability.**

***Index Terms*—Random Forest model, Exploratory data analysis, Data Visualization, Model Evaluation**

## I. INTRODUCTION

The data considered was extracted from the UCI Machine learning repository. The key task is to evaluate customer acceptability of Cars. The aim is use and apply Random Forest and perform Exploratory data analysis to uncover previously unknown or hidden facts in the data set available, visualize them using Data Visualization and make predictions on the validation data. The task is to conclude the study of which types of cars are more likely to have a higher customer acceptability level.

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

The goal of this research paper is to correctly predict which kind of cars are more likely to have a higher customer acceptability given a set of information. A predictive classification model based on Random Forest was built using the given data so that the 'Buying price', 'Price of the maintenance', 'Number of doors', 'Capacity', 'Estimated safety of the car' and other features contributed to their customer acceptability. The output of a cars customer acceptability is predicted using the Random Forest based on different feature combinations of the data.

In this paper, we conducted exploratory data analysis to excavate different knowledge existing in the given data set and to perceive the impact of every field with respect to the customer acceptability by the use of 'Customer Acceptability' field analysis in between each field of the data set. Data pre-processing was done, data analysis was performed and, a Random Forest Classifier model was built and trained on the training data, and the accuracy was tested on the validation dataset. After analyzing the dataset, we discovered insights and information on what the cars with high customer acceptability had in common that helped them do so. We were also able to predict if a particular car with certain feature values would have a good customer acceptability level by applying the tools of machine learning and statistical analysis.

## II. RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance. Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, i.e. have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. Though not quite similar, forests give the effects of a K-fold cross validation.

### A. Bagging

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly ('B' times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$ :
1. Sample, with replacement, n training examples from X, Y; call these $X_b, Y_b$.
2. Train a classification or regression tree $f_b$ on $X_b, Y_b$.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':
$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$ or by taking the majority vote in the case of classification trees.

This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.

Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on x': $\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x')-\hat{f})^2}{B-1}}$.

The number of samples/trees, B, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimal number of trees B can be found using cross-validation, or by observing the 'out-of-bag error': the mean prediction error on each training sample $x_i$, using only the trees that did not have $x_i <$ in their bootstrap sample. The training and test error tend to level off after some number of trees have been fit.

### B. From bagging to random forests

The above procedure describes the original bagging algorithm for trees. Random forests also include another type of bagging scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the B trees, causing them to become correlated.

Typically, for a classification problem with p features, $\sqrt{p}$ (rounded down) features are used in each split. For regression problems the inventors recommend p/3 (rounded down) with a minimum node size of 5 as the default. In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.

### C. ExtraTrees

Adding one further step of randomization yields "extremely randomized trees", or ExtraTrees. While similar to ordinary random forests in that they are an ensemble of individual trees, there are two main differences: first, each tree is trained using the whole learning sample (rather than a bootstrap sample), and second, the top-down splitting in the tree learner is randomized. Instead of computing the locally 'optimal' cut-point for each feature under consideration (based on, e.g., information gain or the Gini impurity), a 'random' cut-point is selected. This value is selected from a uniform distribution within the feature's empirical range (in the tree's training set). Then, of all the randomly generated splits, the split that yields the highest score is chosen to split the node. Similar to ordinary random forests, the number of randomly selected features to be considered at each node can be specified. Default values for this parameter are $\sqrt{p}$ for classification and $p$ for regression, where $p$ is the number of features in the model.

## III. Using Random Forest for Classification to evaluate Customer Acceptability of Cars

### A. Description of the Dataset

The data considered was is the Car evaluation dataset is taken from UCI Machine learning repository derived from simple hierarchical decision model.

The dataset comprises of 1728 observations and 7 variables. Out of which one variable('Customer Acceptability') is the dependent variable and the rest 6 are independent variables.

It was noted that all the variables in the considered dataset are categorical in nature and had values of 'object' datatype(Fig.1. and Fig.2.).

```
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Buying price                 1728 non-null   object
 1   Price of the maintenance     1728 non-null   object
 2   Number of doors              1728 non-null   object
 3   Capacity                     1728 non-null   object
 4   The size of luggage boot     1728 non-null   object
 5   Estimated safety of the car  1728 non-null   object
 6   Customer Acceptability       1728 non-null   object
```

Fig. 1. Information about the dataset

| | Buying price | Price of the maintenance | Number of doors | Capacity | The size of luggage boot | Estimated safety of the car | Customer Acceptability |
|---|---|---|---|---|---|---|---|
| count | 1728 | 1728 | 1728 | 1728 | 1728 | 1728 | 1728 |
| unique | 4 | 4 | 4 | 3 | 3 | 3 | 4 |
| top | med | med | 2 | 2 | med | med | unacc |
| freq | 432 | 432 | 432 | 576 | 576 | 576 | 1210 |

Fig. 2. Statistics of the dataset

### *B. Data Cleaning*

The raw dataset was found to be having vague column/variable names, these variables were renamed with meaningful variable names after performing a background research on the dataset.

### *C. Visualizations and insights from the analysis*


Fig. 3. Average Customer Acceptability level for different levels of Buying Price

From (Fig.3.) we infer that, lower the buying price, higher is the average customer acceptability level.

From (Fig.4.) we infer that, lower the price of maintenance, higher is the average customer acceptability level.

From (Fig.5.) we can conclude that, higher the number of doors in the car, higher is the average customer acceptability level.

From(Fig.6.) we can conclude that, cars with a capacity of 2 have zero customer acceptability level. And, cars with a capacity of 4 people was most preferred and had the highest

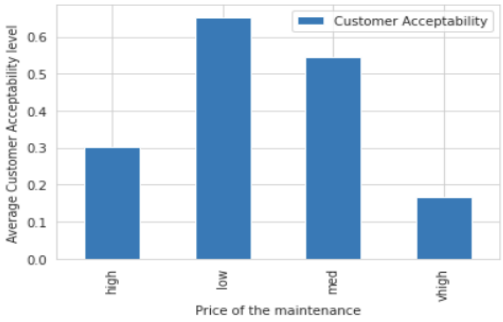


Fig. 4. Average Customer Acceptability level for different levels of Price of Maintenance

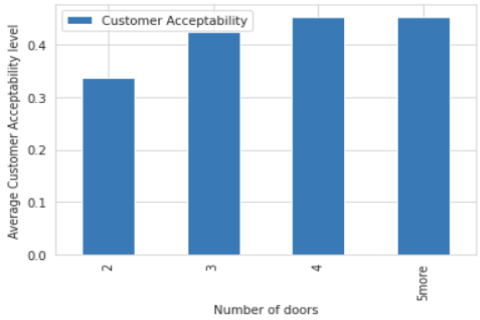


Fig. 5. Average Customer Acceptability level for different number of doors in the car

average customer acceptability level.

From(Fig.7.) we can infer that the higher the size of luggage boot, higher is the average customer acceptability level.

From (Fig.8.), it can be said that, higher the estimated safety of the car, higher is the average customer acceptance level. Cars with 'low' safety estimation, had zero customer acceptability level.

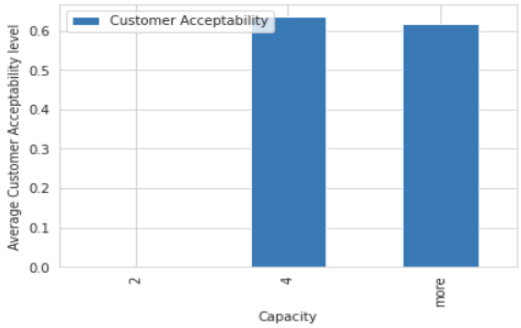


Fig. 6. Average Customer Acceptability level for different values of Car Capacity

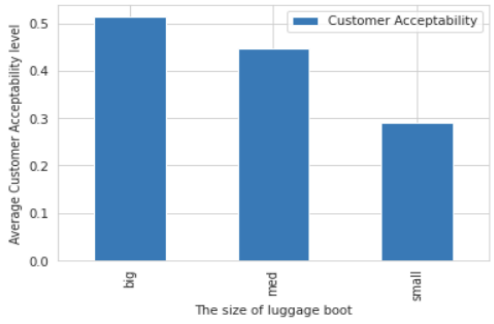


Fig. 7. Average Customer Acceptability level for different levels of Luggage Boot Size

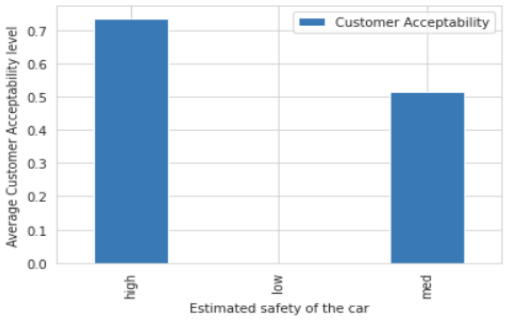


Fig. 8. Average Customer Acceptability level for different levels of Estimated Car Safety

### D. Data Preprocessing

Categorical variables were created from the given variables of 'object' type. Since all the categories in the variables had a naturally increasing order, they were converted to numbers (e.g. 1, 2, 3, 4, 5) preserving the order. These are called ordinals.

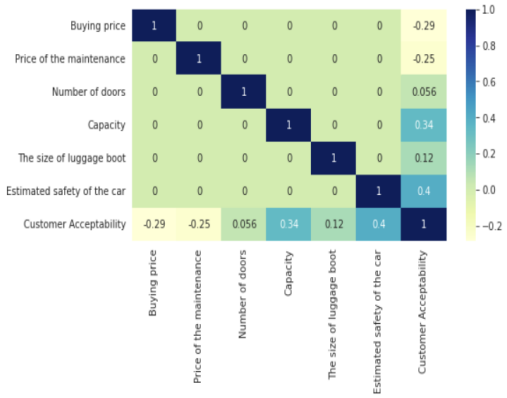


Fig. 9. Correlation heat-map between the various features of data

The correlation between no variables were found to be significant(Fig.9.), hence there was no requirement to drop any variable before training to remove redundancy.

The dataset was also split into the training and validation parts with the validation data being used to calculate the accuracy and performance of the model before making predictions on the test data.

### E. Building the Random Forest Model

A Random Forest Model has been built using the inbuilt 'RandomForestClassifier' from the Skikit-Learn library. The variables- 'Buying price', 'Price of the maintenance', 'Number of doors', 'Capacity', 'The size of luggage boot', 'Estimated safety of the car' were taken as the categorical inputs and were used to predict the categorical output variable-'Customer Acceptability'.

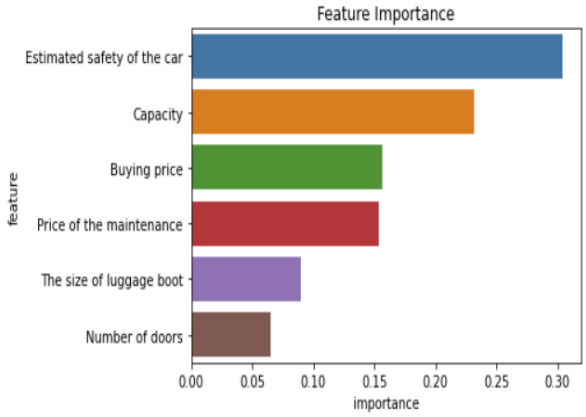


Fig. 10. Feature Imporatance

It was found out(Fig.10.) that the Capacity and Estimated Safety of the car were amongst the features/variables of utmost importance in the decision tree model.

### F. Model performance

The dataset was split into the training and validation parts and the training dataset was used to train the model to fit the given data. The random forest model was found to fit the training data with an astounding 100% accuracy.

The performance of the Random Forest model was also evaluated on the validation dataset and, the accuracy of the model was found to be 96.3%. A confusion matrix of our model(Fig.11.) on the validation dataset was plotted to visualize the miss-classified datapoints.

## IV. CONCLUSIONS

We analyzed the role of various features-'Buying price', 'Price of the maintenance', 'Number of doors', 'Capacity', 'The size of luggage boot', 'Estimated safety of the car' in estimating the 'Customer Acceptability' of cars. It was inferred that lower the buying price, higher was the

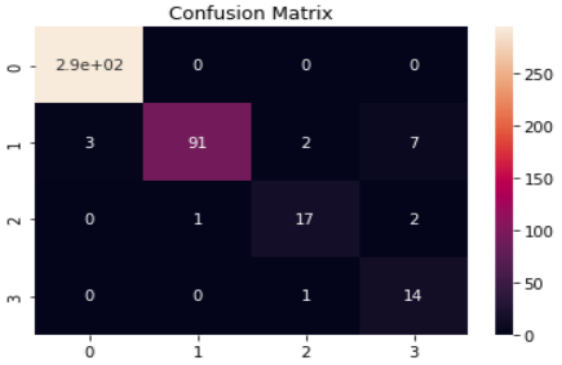


Fig. 11. Confusion matrix of our model on the validation dataset

average customer acceptability level. And, lower the price of maintenance, higher was the average customer acceptability level. It was also discovered that higher the number of doors in the car, higher was the average customer acceptability level. Cars with a capacity of 2 were found to have zero customer acceptability level. And cars with a capacity of 4 people was most preferred and had the highest average customer acceptability level. It was also found that the customer acceptability level was higher for higher the sizes of luggage boot. Safety of the car was found to be one of the features of utmost importance in customer acceptability. Cars with low safety estimation, had zero customer acceptability level. And cars with high estimated safety had a high average customer acceptance level.

Despite the high accuracy of the model on the validation dataset(96.3%), further improvements in the model are possible. In order to further improve the overall result, an extensive hyper-parameter tuning can be done to improve the accuracy of the model for it to better fit the data. It also is possible to further improve it by possibly finding even better machine learning algorithms for the concerned task.

## REFERENCES

[1] Ali, Jehad Khan, Rehanullah Ahmad, Nasir Maqsood, Imran. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues(IJCSI). 9.
[2] Berk, Richard. (2020). Random Forests. 10.1007/978-3-030-40189-4_5.
[3] Boehmke, Brad Greenwell, Brandon. (2019). Random Forests. 10.1201/9780367816377-11.
[4] Rebala, Gopinath Ravi, Ajay Churiwala, Sanjay. (2019). Random Forests. 10.1007/978-3-030-15729-6_7.