Car Evaluation
Random Forest Classifier

Devcharan K
Indian Institute of Technology, Madras

***Abstract*—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. In this paper, data about car safety evaluation was analyzed using the Random Forest classifier with varying number of decision trees. The classifier is used to predict the safety of a given car based on its features.**

I. Introduction

Cars are essentially part of our regular day to day life. Car safety is extremely important because the roads which we drive on are much more dangerous than we think. When an individual considers buying a car, there are numerous aspects that could influence their choice on which kind of car they are keen on. The aim of this paper is to be able to predict the safety level of the car with the data provided.

Random forest is a supervised learning algorithm. It has two variations – one is used for classification problems and other is used for regression problems. It is one of the most flexible and easy to use algorithm. It creates decision trees on the given data samples, gets prediction from each tree and selects the best solution by means of voting. It is also a pretty good indicator of feature importance. Random forest algorithm combines multiple decision-trees, resulting in a forest of trees, hence the name Random Forest. In the random forest classifier, the higher the number of trees in the forest results in higher accuracy.

In this paper, the data about cars such as price, maintenance cost, number of doors, seating capacity, storage space is considered for evaluation. The aim is to classify the car based on its safety.

In Section II, Random Forest Classifier and techniques used are explained. In Section III, the problem and the method to solve it is shown. In Section IV, the concepts used to prepare the model is evaluated. Finally the inferences of the analysis are presented in Section V.

II. Random Forest Classifier

Random forest algorithm intuition can be divided into two stages. In the first stage, "k" features out of total m features are randomly selected and the random forest is built. The first stage is as follows:-

1) Randomly select k features from a total of m features where $k < m$.
2) Among the k features, calculate the node d using the best split point.
3) Split the node into daughter nodes using the best split.
4) Repeat 1 to 3 steps until l number of nodes has been reached.
5) Build forest by repeating steps 1 to 4 for n number of times to create n number of trees.

In the second stage, predictions are made using the trained random forest algorithm.

1) The test features are taken and the rules of each randomly created decision tree is used to predict the outcome and stores the predicted outcome.
2) Then, the votes for each predicted target is calculated.
3) Finally, the highest voted predicted target is considered as the final prediction from the random forest algorithm.

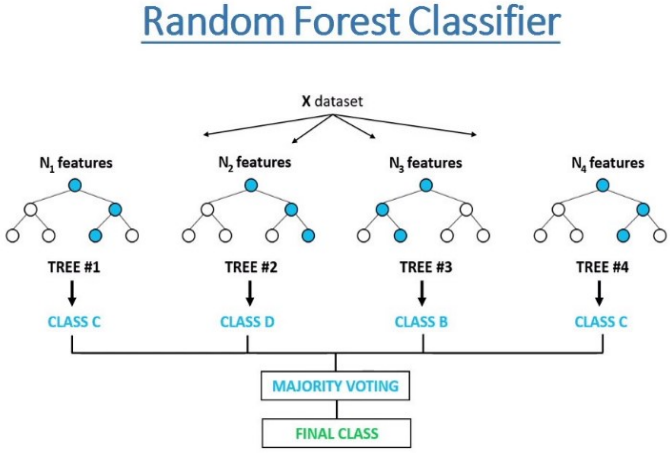

Fig. 1. Random Forest Classifier

A. Advantages and disadvantages of Random Tree Classifier

The advantages of Random forest algorithm are as follows:-

- Random forest algorithm can be used to solve both classification and regression problems.
- It is considered as very accurate and robust model because it uses large number of decision-trees to make predictions.
- Random forests takes the average of all the predictions made by the decision-trees, which cancels out the biases. So, it does not suffer from the over-fitting problem.

- Random forest classifier can handle the missing values. There are two ways to handle the missing values. First is to use median values to replace continuous variables and second is to compute the proximity-weighted average of missing values.
- Random forest classifier can be used for feature selection. It means selecting the most important features out of the available features from the training dataset.

The disadvantages of Random Forest algorithm are listed below:-

- The biggest disadvantage of random forests is its computational complexity. Random forests is very slow in making predictions because large number of decision-trees are used to make predictions. All the trees in the forest have to make a prediction for the same input and then perform voting on it. So, it is a time-consuming process.
- The model is difficult to interpret as compared to a decision-tree, where we can easily make a prediction as compared to a decision-tree.

B. Difference between Random Forests and Decision Trees

Some salient features of comparison are as follows:-

- Random forests is a set of multiple decision-trees.
- Decision-trees are computationally faster as compared to random forests.
- Deep decision-trees may suffer from overfitting. Random forest prevents overfitting by creating trees on random forests.
- Random forest is difficult to interpret. But, a decision-tree is easily interpretable and can be converted to rules.

C. Feature selection with Random Forests

Random forests algorithm can be used for feature selection process. This algorithm can be used to rank the importance of variables in a regression or classification problem. The variable importance in a dataset is measured by fitting the random forest algorithm to the data. During the fitting process, the out-of-bag error for each data point is recorded and averaged over the forest. The importance of the j-th feature was measured after training. The values of the j-th feature were permuted among the training data and the out-of-bag error was again computed on this perturbed dataset. The importance score for the j-th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences.

Features which produce large values for this score are ranked as more important than features which produce small values. Based on this score, we will choose the most important features and drop the least important ones for model building.

III. The Problem

The car details such as price, maintenance cost, seating capacity, luggage space etc is considered. The aim of this exercise is to classify the cars based on their safety level which can be analyzed with the help of the other data provided. This model can also be used to predict the safety level of other cars provided the appropriate data.

A. Data Cleaning and Exploration

The dataset has 7 columns but no column names. The columns are named as follows: *'buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class'*. There are no missing values in the 1728 data samples. *'Class'* is the target variable and the remaining variables belong to the feature vector. All the variables are categorical in nature

IV. Model Evaluation

The model to be created is a Random Forest Classifier. The above mentioned techniques are used to create the model.

A. Feature Engineering

Feature Engineering is the process of transforming raw data into useful features that help us to understand our model better and increase its predictive power. In this case that is done by encoding the categorical variables using the package *'category_encoders'* which is used to convert categorical variables to numeric for ease of usage in modeling.

B. Test Train Split

The dataset is first split into two parts called training data and testing data. The training data is used to train the model and then the model is applied on the testing data to see its accuracy.

C. Random Forest with default parameters

First the model is built with default parameters of $n\_estimators = 10$. So, 10 decision trees are used to build the model. The model accuracy score obtained with this setting is 0.9650 (or 96.5%)

D. Random Forest with 100 Decision trees

The model is rebuilt with $n\_estimators = 100$. In this case 100 decision trees are used to build the model. The model accuracy score obtained with this setting is 9.9685 (or 96.85%). There is a slight improvement in model accuracy when a greater number of decision trees are used in the model.

E. Feature Importance

Until now, all the features were included in the analysis.But, as shown in Figxxx, the feature "$doors$" has very low importance while the feature "$safety$" is of the highest importance. So, the models are rebuilt after dropping the "doors" variable. The model accuracy score obtained after removing the "doors" variable is 0.9764 (or 97.64%). There is even more improvement in the model accuracy when the least important feature is ignored.

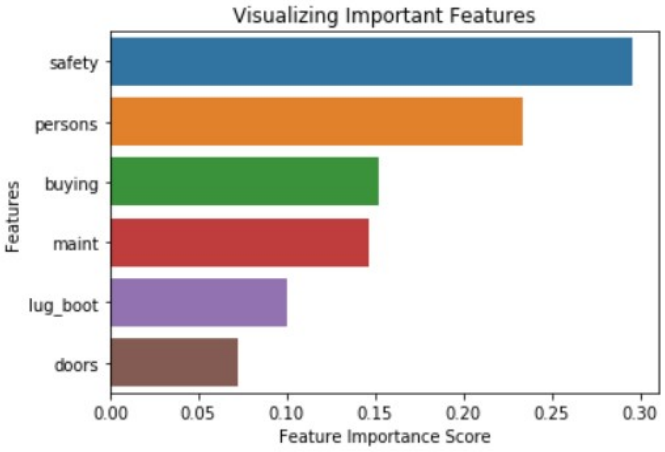

Fig. 2. Importance of each feature in the dataset

F. Confusion Matrix

A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give a clear picture of classification model performance and the types of errors produced by the model. It gives a summary of correct and incorrect predictions broken down by each category.

The confusion matrix or the Random Tree classifier model is $\begin{bmatrix} 104 & 12 & 10 & 3 \\ 0 & 18 & 0 & 2 \\ 10 & 0 & 387 & 0 \\ 3 & 2 & 0 & 20 \end{bmatrix}$

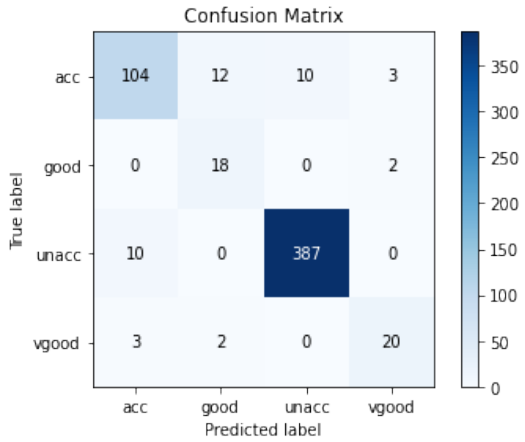

Fig. 3. Confusion matrix for Random Forest Classifier

V. Inferences

The Random Forest classifier to predict the safety of a car yields a very good performance as indicated by the model accuracy score of 97%. There is no sign of over-fitting in both the cases of decision trees implemented. The confusion matrix yields a very good performance.

VI. Conclusion

The Random Forest Classifier model is a very powerful classifier as indicated by the accuracy scores above. The model with 10 decision trees had an accuracy score of 96.5% The model with 100 decision trees had an accuracy score of 96.85%. The model after removing the least important feature "doors" had an accuracy score of 97.64%.

References

[1] V. Y. Kulkarni and P. K. Sinha, "Pruning of Random Forest classifiers: A survey and future directions," 2012 International Conference on Data Science Engineering (ICDSE), 2012, pp. 64-68, doi: 10.1109/ICDSE.2012.6282329.

[2] Aurelien Geron,"Hands on Machine Learning with Scikit-Learn and Tensorflow", 2017.

[3] "Random Forest" Wikipedia [Accessed on 10 November] Available: https://en.wikipedia.org/wiki/Random_forest

[4] Understanding Random Forests Classifiers in Python Available: https://www.datacamp.com/community/tutorials/random-forests-classifier-python

[5] Evaluating Machine Learning Models using Hyperparameter Tuning Available: https://www.analyticsvidhya.com/blog/2021/04/evaluating-machine-learning-models-hyperparameter-tuning/