Automobile Exploratory Data Analysis and Risk Classification

Christina Chang

***Abstract* — The purpose of this report is to classify cars as safe or risky using two tree based classifiers, Decision Trees and Random Forest. The performance of Decision Tree and Random Forest was compared by using cross validation accuracy scores, ROC curves, and PR curves. The five fold cross validated accuracy score of Random Forest was 80%, while the five fold cross validated accuracy score of Decision Tree was 71%. The average area under the curve for the cross validated ROC and PR curves was higher for Random Forest. Overall, the Random Forest classifier is better suited for this problem when compared to Decision Tree classifiers. Some important features for determining whether a car is safe or risky are the number of doors, height of the car, and width of the car.**

I. Introduction

The automobile data set is from the UCI Machine Learning Repository, and it includes 205 observations of cars. Each observation has 26 attributes, which are either numerical or categorical. Out of the 26 features, 24 features describe various attributes of the vehicle (wheel base, length, with, engine size, etc.). The other two features are the safety rating of the vehicle (between -3 and +3, where +3 is the riskiest and -3 is the safest) and the make of the vehicle (Audi, BMW, Toyota, etc.).

The motivation of this project is to classify cars as safe or risky based on several features of the vehicle. If the classification models perform well, it can be used by consumers when considering which car to purchase, or by insurance companies when calculating the price of auto insurance policies.

II. Data Cleaning

Some features include missing values, so the missing values were imputed. The missing values in the numerical features were imputed by using the k-Nearest Neighbor algorithm, or KNNImputer in Python. The missing values in the categorical features were imputed by using the most frequent value method. This essentially replaces missing data with the most frequent values within a column.

I removed the column ‘make’ from the data because the make does not contribute to the physical attributes of the car. Therefore, the car make will not be used in the classification algorithms. Also, some features were not the correct data type and they were converted from object type to numeric type.

III. Exploratory Data Analysis

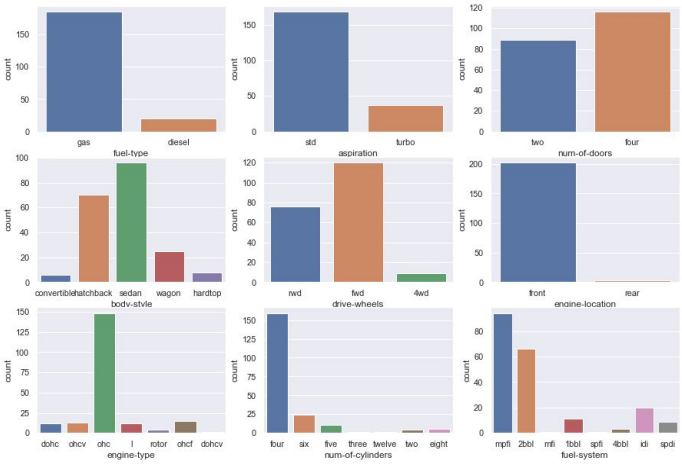

Figure 1: Barplots of counts for categorical features.

Figure 1 shows barplots of counts for each categorical feature in the data. Cars with front engines overrepresented in the data when compared to cars with rear engines. The most common body style of car is sedan. Majority of cars are gas fueled rather than diesel fueled. Also, most cars have four cylinders.

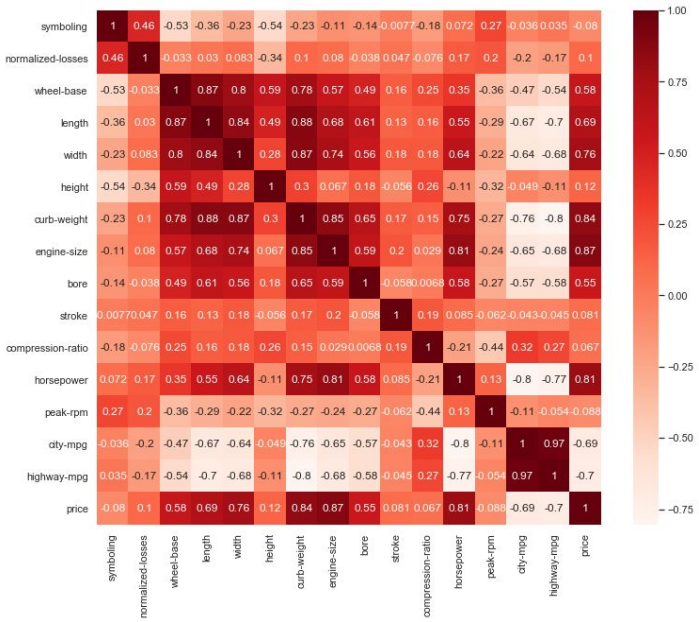

Figure 2: Heatmap of correlations between numerical variables.

Figure 2 is a heatmap of the correlations between numerical variables. Wheelbase, length, and width have a high positive correlation. City mpg and highway mpg have a high negative correlation. The features that have a positive correlation with symboling are normalized losses and peak rpm. The features that have a moderate negative correlation with symboling are wheel base, height, width and curb weight.

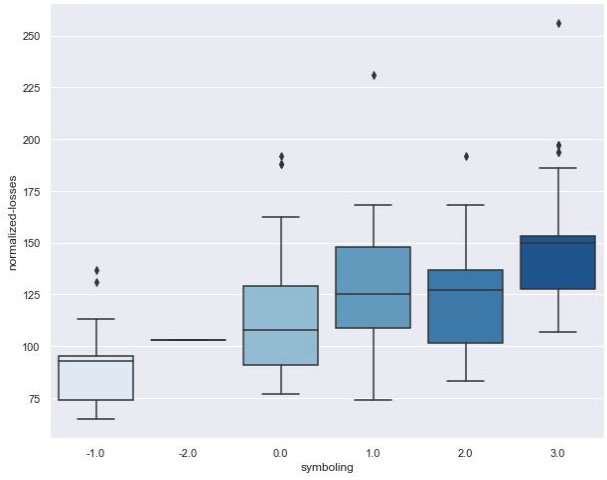

Figure 3: Grouped boxplot of normalized losses by symboling.

Figure 3 shows boxplots of normalized losses grouped by symboling. Normalized losses are the relative average loss payment per insured vehicle. Symboling is the insurance risk level of a car. A value of -3 is very safe and a value of +3 is very risky. Riskier cars seem to have a higher normalized loss. This is reasonable because safer cars should have smaller losses, while riskier cars should have larger losses.

IV. Methods

A. Classification Models

Decision Trees are a form of supervised learning algorithm that can be used for classification or regression [1]. They are tree-like structures that are easy to visualize and it is simple to understand the decision rules for prediction [1]. Each decision is based on an attribute and this forms a branch of the tree. An advantage of decision trees is that it is simple, fast, and not influenced by outliers [1]. However, it is prone to overfitting and unstable [1].

Random forests are an ensemble learning method that uses multiple decision trees to make predictions [1]. It is not as likely to overfit as it utilizes multiple trees, therefore, it yields better accuracy [1]. But, Random Forests are difficult to interpret and it is computationally more expensive.

B. Data Preprocessing

In order to use the data in Decision Tree and Random Forest classification models, the categorical variables need to be encoded. The method of encoding used was one hot encoding. One hot encoding converts categorical variables to numerical variables by representing them as binary vectors [2]. After one hot encoding, the feature space almost doubled and the dataset had 45 features. This is one disadvantage of using one hot encoding because it can lead to very high dimensional feature representations [2].

To classify cars as safe or risky, the symboling attribute was converted to a categorical variable with two classes: ‘safe’ or ‘risky’. ‘Safe’ cars have a symboling between -3 and 0. Neutral cars (symboling of 0) were considered safe because there were a large number of risky cars. ‘Risky’ cars have a symboling between +1 and +3. This is the response variable that will be the target for the classification models.

C. Feature Selection

Since the feature space increased from 26 attributes to 45 attributes, feature selection was performed on the data to remove irrelevant features and improve model performance [3]. The ExtraTreesClassifier in Python was used for feature selection by evaluating the feature importance score returned from the Extra Trees model [3]. After feature selection, the feature space was reduced to 18 attributes.

D. Training and Testing Set

The reduced dataset was split into 70% training and 30% testing.

E. Hyperparameter Tuning

Hyperparameter tuning searches the hyperparameter space for a set of hyperparameters that will optimize the model performance [4]. For both classification models, the hyperparameter tuning method used was Grid Search, or GridSearchCV in Python.

For the Decision Tree model, the parameter grid included different values for the following parameters: criterion (measures quality of split), splitter (strategy used to split nodes), max_depth (maximum depth of tree), max_leaf_nodes (maximum nodes on tree), and min_samples_split (minimum number of samples required to split a node).

For the Random Forest model, the parameter grid included different values for the following parameters: n_estimators (number of trees in the forest), max_depth (maximum depth of tree), min_samples_split (minimum number of samples required to split a node), and min_samples_leaf (minimum number of samples required to be at a leaf node).

V. RESULTS

A. *Decision Tree*

Figure 4 shows the Decision Tree classifier that achieved the best accuracy score. The best parameters determined from the grid search was criterion = 'entropy', max_depth = 5, max_leaf_nodes = 9, min_samples_split = 19, and splitter = 'best'.

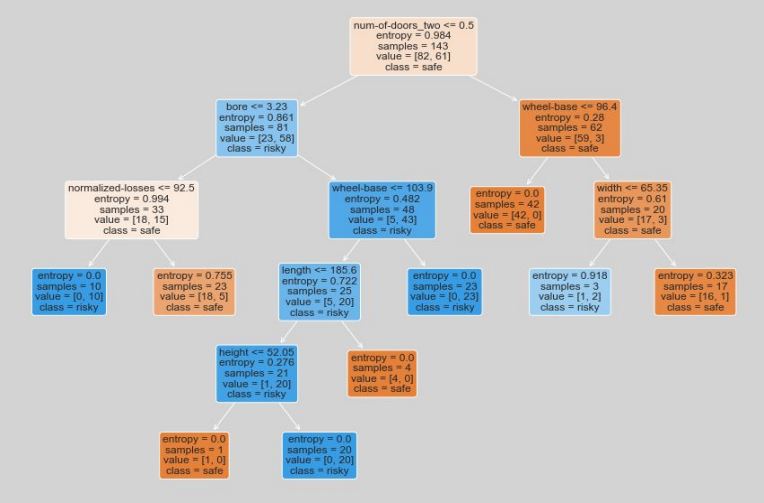

Figure 4: Decision Tree classifier model after hyperparameter tuning.

B. *Random Forest*

Figure 5 shows the feature importances from the Random Forest model that achieved the best accuracy score. The most important features in the Random Forest model were the number of doors and width. The best parameters determined from the grid search as n_estimators = 7, max_depth = 10, min_samples_leaf = 1, and min_samples_split = 5.

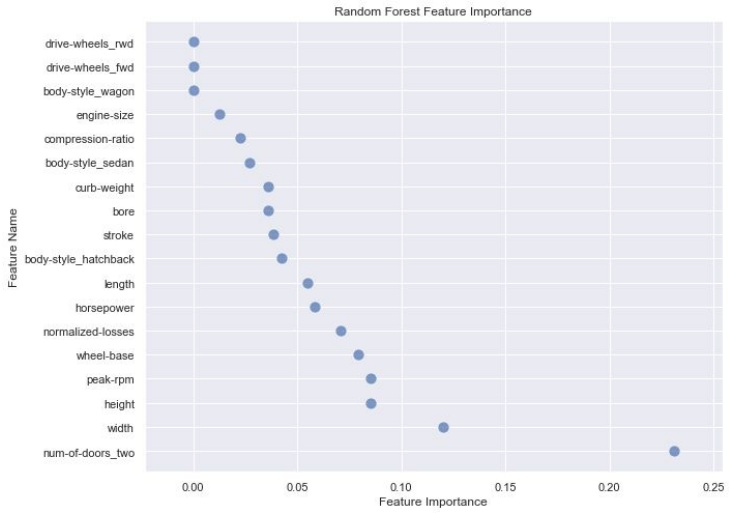

Figure 5: Feature importances obtained from Random Forest model after hyperparameter tuning.

VI. Model Evaluation and Comparison

A. *Test Set Confusion Matrix*

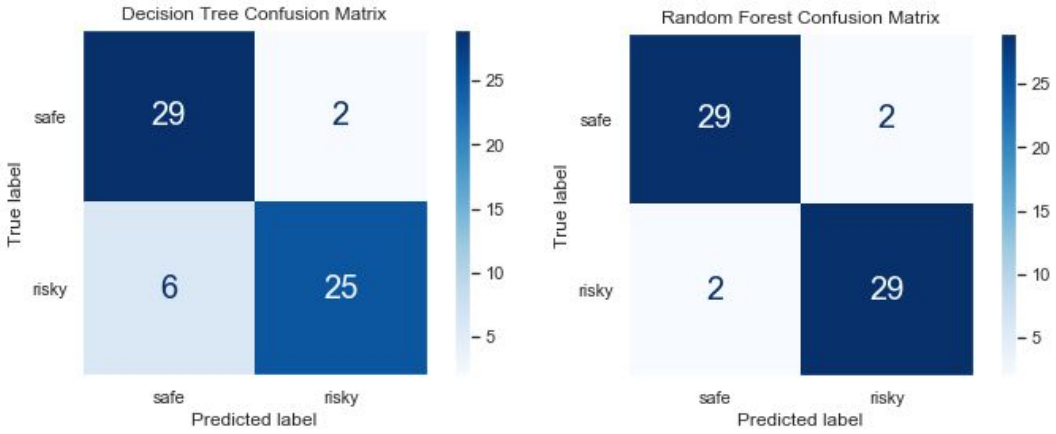

Figure 6: Confusion matrix for the best Decision Tree (left) and Random Forest (right) model.

Figure 6 shows the confusion matrix for the testing set using the best models obtained from grid search. The Decision Tree classifier misclassified 8 observations and the Random Forest classifier misclassified 4 observations.

B. *Cross Validated Accuracy Score*

For the Decision Tree classifier, the mean five fold cross validation score was 71% with a variance of 0.13%. For the Random Forest classifier, the mean five fold cross validation score was 80% with a variance of 0.12%.

C. *Receiver Operating Characteristic (ROC) Curve*

Figure 7 shows the ROC curves for both models. The ROC curve shows how well a model can distinguish between classes by plotting the true positive rate versus false positive rate [5]. The area under curve (AUC) can be used to compare model performance; the closer AUC is to 1, the better the model is [5]. For the Decision Tree classifier, the AUC for the mean ROC curve was 0.74. For the Random Forest classifier, the AUC for the mean ROC curve was 0.80.

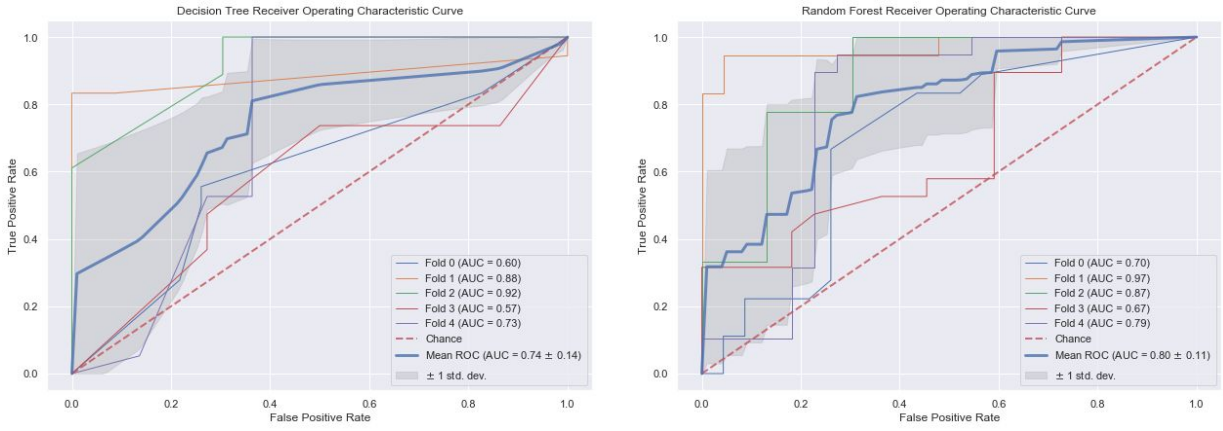

Figure 7: ROC curves for the best Decision Tree (left) and Random Forest (right) model.

D. *Precision Recall (PR) Curve*

Figure 8 shows the PR curves for both models. PR curves are another performance metric that plots positive predictive value (precision) and the true positive rate (recall) [5]. For the Decision Tree classifier, the AUC for the mean PR curve was 0.88. For the Random Forest classifier, the AUC for the mean PR curve was 0.96.

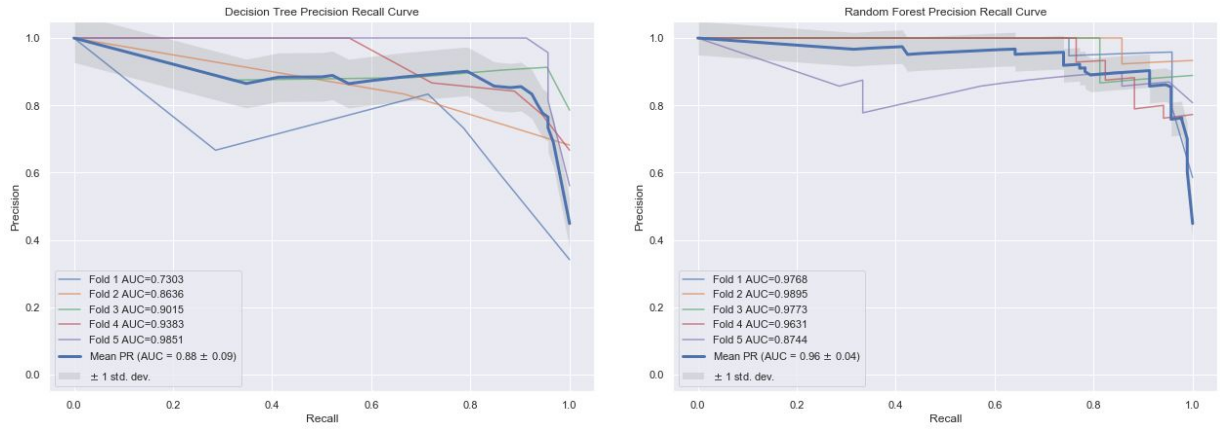

Figure 8: PR curves for the best Decision Tree (left) and Random Forest (right) model.

VII. Conclusion

Based on the model evaluation, the Random Forest classifier outperformed the Decision Tree classifier. The Random Forest classifier had a higher five fold cross validation score. Also, the AUC values for both the mean ROC curve and mean PR curve were higher for Random Forest when compared to Decision Tree.

The Random Forest model determined that the most important features were number of doors, width, height, length, peak rpm, wheel base, horsepower, and normalized losses. These features all had an importance score greater than 0.05. However, the number of doors attribute was the most important feature, with an importance score of 0.23. Thus, these attributes may be important in determining how safe a car is.

References

[1] T. Prajwala, "A Comparative Study on Decision Tree and Random Forest Using R Tool," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, iss. 1, January 2015.

[2] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing," *Degree Project in Technology*, 2018.

[3] E. Hemphill, J. Lindsay, C. Lee, I. Mandoiu, and C. Nelson, "Feature selection and classifier performance on diverse bio- logical datasets," *BMC Bioinformatics*, November 2014.

[4] P. Probst, A. Boulesteix, and B. Bischl, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms," *Journal of Machine Learning Research*, vol. 20, March 2019.

[5] P. Flach and M. Kull, "Precision-Recall-Gain Curves: PR Analysis Done Right," *Advances in Neural Information Processing Systems*, vol. 28, December 2015.