Classification and Regression Tasks

By: Ruba Othman

Course: Concepts in Artificial Intelligence

Tutor: Mr. Ahmed Khubaib

Contents

1. Introduction

Classification and Regression are both known to be supervised learning algorithms. They are used in a number of Machine Learning applications such as forecasting and labeling.

Classification is mainly used for dividing data into different categories, an example is classifying genders of workers in a company as Male or Female or labeling different emails as Spam or Not Spam. Regression on the other hand is used in the process of predicting data of continuous types such as stock prices or age.

In both Classification and Regression, there are different types of algorithms. This report aims to implement the different types of algorithms as two separate tasks and compare their results with each other.

The first part of the report will investigate the Regression models, in that task the Boston Housing dataset was used for training the Linear Regression and Random Forest Regressor and their results were compared.

The second part of the report will look into the Classification models, the White Wine Quality dataset was utilized for the Decision Tree Classifier and Random Forest Classifier. The end results were also compared.

2. Regression Task:

For the regression task, the Kaggle Boston House Pricing dataset was used to predict the prices of houses. Two algorithms were used in this task and their accuracies were compared to each other.

The first algorithm implemented was Linear Regression, and it obtained an overall accuracy of 45% as seen in below Figure 1.

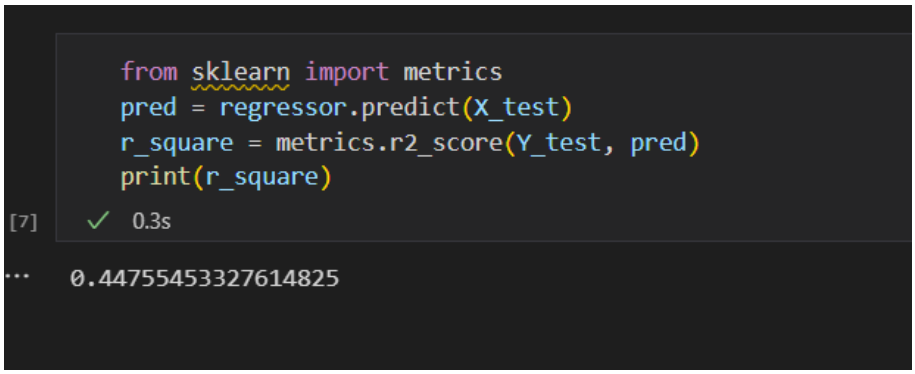

Figure 1. Accuracy of Linear Regression model

The second algorithm was Random Forest Regressor and it achieved a better accuracy of 80%.

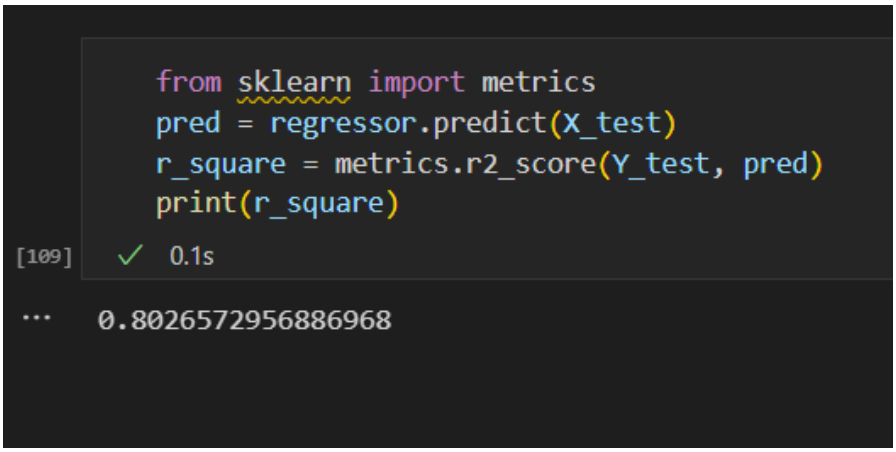

Figure 2. Accuracy of Random Forest Regressor model

In Linear Regression, a line is drawn through the data points of a target and predictor graph to showcase a linear relationship between dependent and independent variables. The relationship between the target and predictor could either be a positive relationship or a negative one (Guwali, 2021). Plotting as example the relationship between MEDV (X-axis) and RM (Y-axis)

below in Figure 3, we can visually plot a straight line that goes through the majority of datapoints whilst ignoring the outliers. This case is a positive relationship as the value of RM increases with MEDV.

In contrast, plotting the relationship between MEDV (X-axis) and LSTAT (Y-axis) in Figure 4, a negative relationship is shown as the value of LSTAT decreases with MEDV. Positive or negative relationships in general do not determine how well a model will predict as Linear Regression essentially works to find the best fit line calculated as below:

$$y = mx+b \quad \Longrightarrow \quad y = a_0+a_1x$$

Where y is the dependent variable, x is the independent variable, a0 is the intercept of the line and a1 is the coefficient.

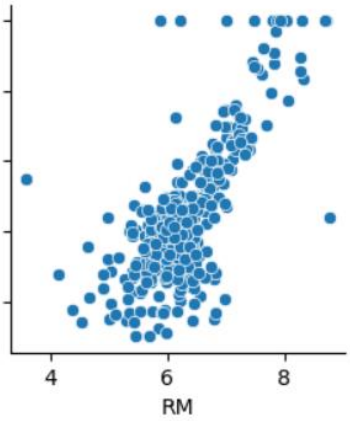

Figure 3. MEDV vs. RM

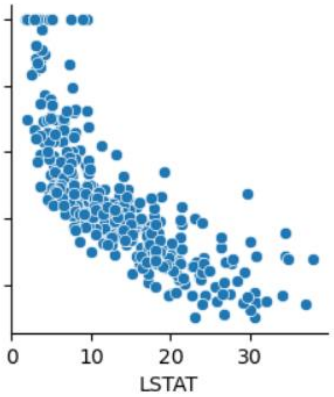

Figure 4. MEDV vs. LSTAT

In the case of the Boston housing dataset, the price of the house is taken as the dependent variable to be predicted while the rest of the features are considered to be independent. The first accuracy obtained of 45% is a very poor accuracy and that is because all the features were used as inputs. To better enhance the performance of the models, the features relationship with the price is to be first visualized. Below Figure 5 shows the correlation between all variables in the dataset.

The proposed model aims to predict the price of the house, hence a focus will be put on the relationship between MEDV and other features. It is noticeable that the data points of MEDV vs. CRIM, TAX, B and ZN is clustered at one side or the other and that is due to their data not being normally distributed.

To enhance the performance of the model, a MinMax Scaler function is applied to the data of CRIM, TAX, B and ZN to normalize their distribution. The normalized dataset will then be trained and tested. This improved the accuracy of the Linear Regression model to 78% as seen in below Figure 5.

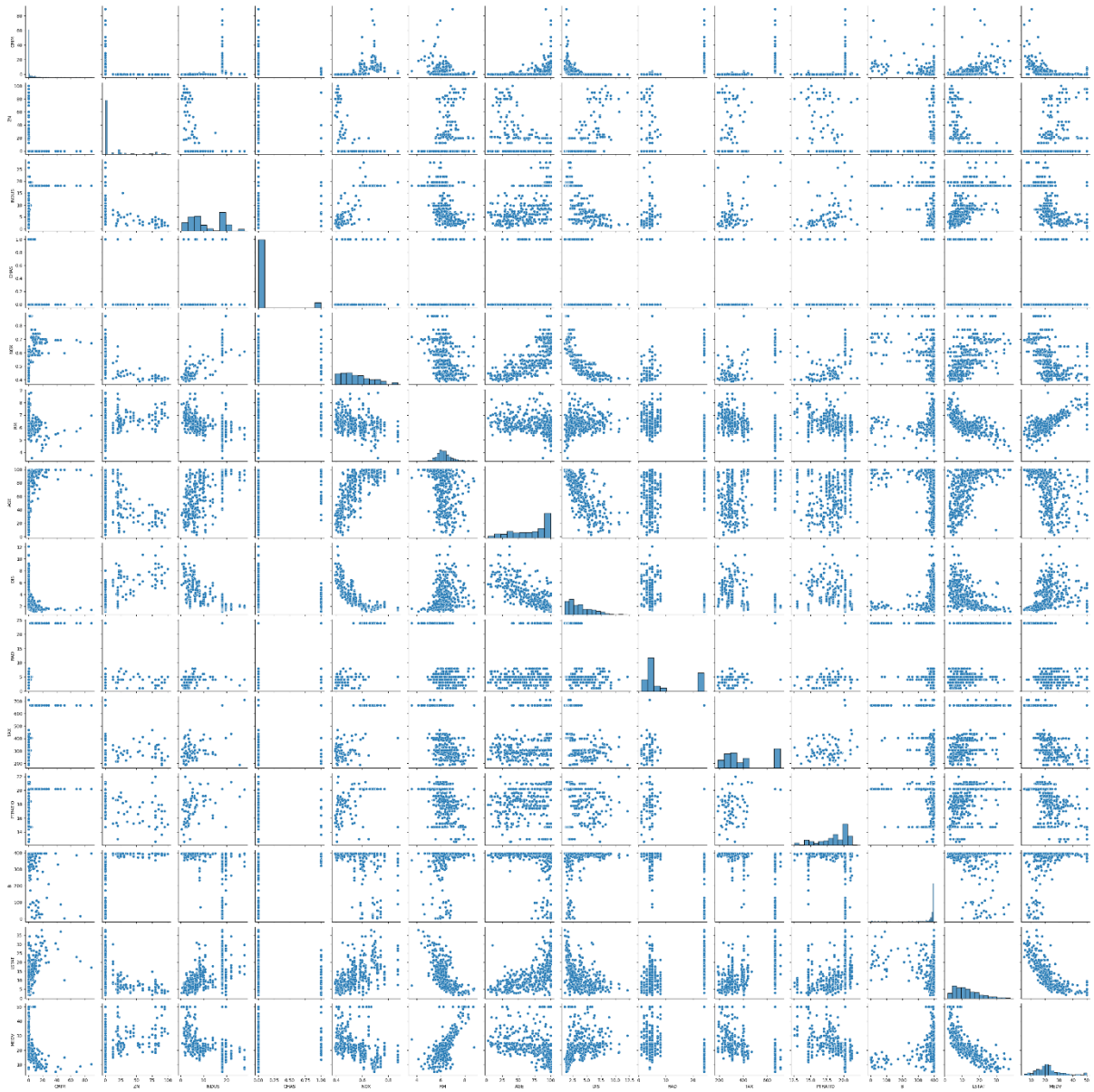

Figure 5. Correlation of Boston Housing Dataset

```
from sklearn import metrics
pred = regressor.predict(X_test)
r_square = metrics.r2_score(Y_test, pred)
print(r_square)
```

```
[99]
...  0.781497569267829
```

Figure 6. Accuracy of Linear Regressor post normalization

The second algorithm used was the Random Forest Regressor, this algorithm utilizes a number of techniques. Firstly, the ensemble learning aggregates predictions of a number of machine learning algorithms to bring about a higher prediction than each of the combined models. It also uses techniques such as boosting where the weak learners are strengthened to bring about one learner that is stronger through correcting previous model errors and resulting in one highly accurate model. The random forest combines these methods in classification and regression where a number of decision trees are formed and the average mean accuracy of all decision trees is the final result (Raj, 2020).

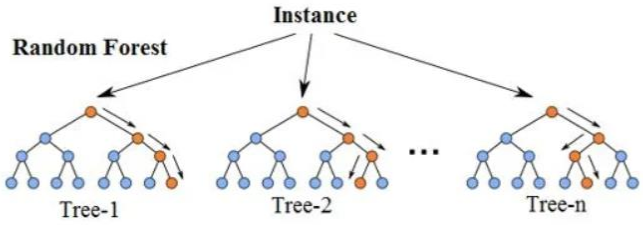

Figure 7. Random Forest Regression Tree (Source: Raj, 2020)

When using the normalized dataset to enhance the performance of the Random Forest Regressor model the accuracy is improved to 92% as seen below.

```
from sklearn import metrics
pred = regressor.predict(X_test)
r_square = metrics.r2_score(Y_test, pred)
print(r_square)
```

```
13]
.  0.926635205391494
```

Figure 8. Random Forest Regressor accuracy post normalization

3. Classification Task:

For the classification task the Kaggle White Wine Quality dataset was used to propose a model that will classify if the wine is of “high” or “low” quality. A “qual” column was first added in the csv file to set a range for the quality, if quality is equal to or greater than 7 “qual” is set to 1, otherwise it is set to 0.

First algorithm implemented was Decision Tree Classifier, which resulted in a macro avg precision of 75%.

```
              precision    recall  f1-score   support

         0.0       0.90      0.87      0.89       768
         1.0       0.59      0.66      0.62       212

    accuracy                           0.83       980
   macro avg       0.75      0.76      0.75       980
weighted avg       0.83      0.83      0.83       980

[[671  97]
 [ 73 139]]
```

Figure 9. Decision Tree Classifier Accuracy

The second algorithm implemented was Random Forest Classifier, that has resulted in a higher macro avg precision of 88%.

```
              precision    recall  f1-score   support

         0.0       0.90      0.97      0.93       768
         1.0       0.86      0.59      0.70       212

    accuracy                           0.89       980
   macro avg       0.88      0.78      0.82       980
weighted avg       0.89      0.89      0.88       980

[[747  21]
 [ 87 125]]
```

Figure 10. Random Forest Classifier Accuracy

To further enhance the model performances, the feature importance of each model is investigated. Below Figure 11 shows the feature importance of the Decision Tree model.

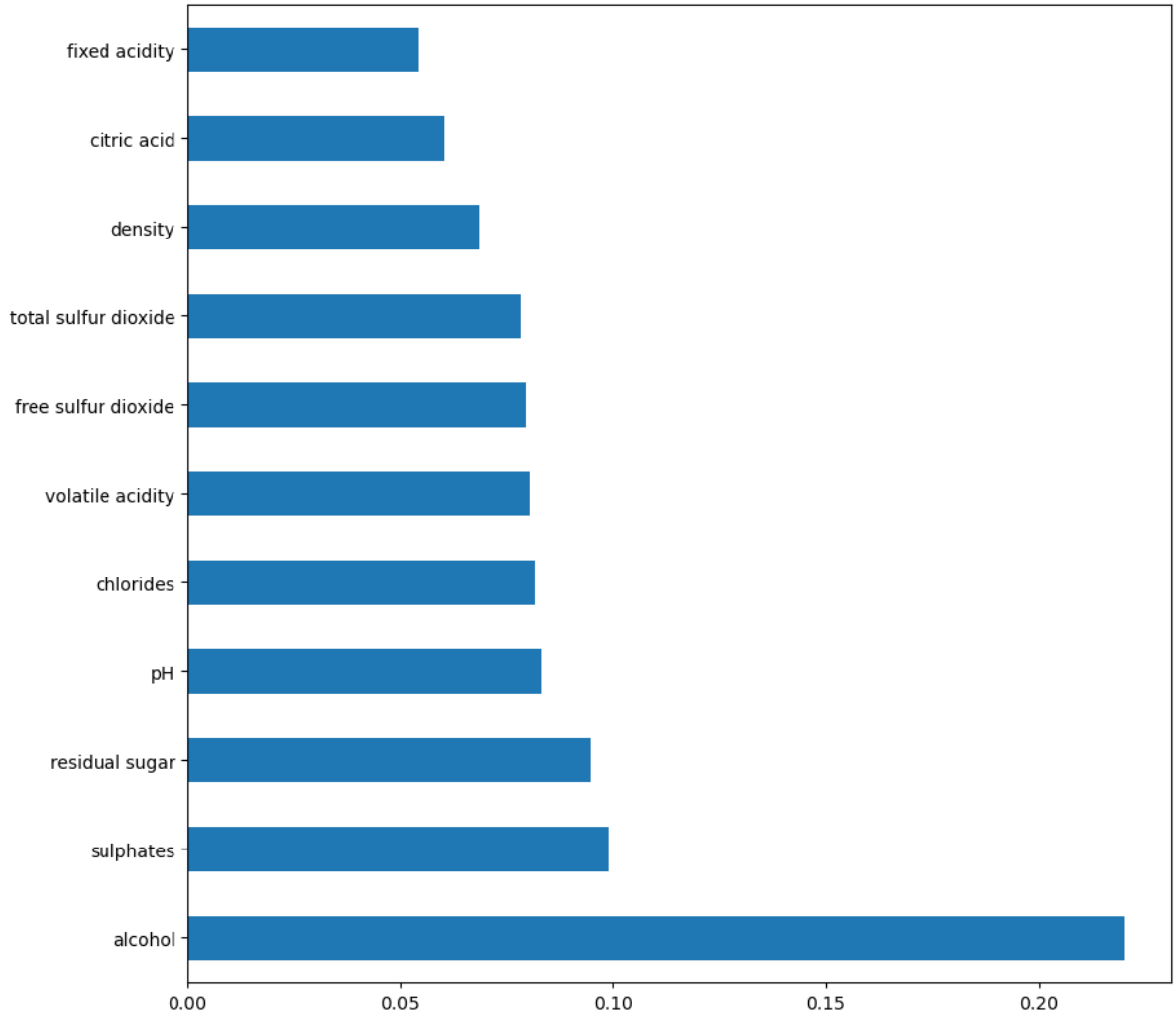

Figure 11. Decision Tree Model Feature Importance

As seen from the bar chart, the alcohol feature has the highest importance when it comes to determining the quality of the wine. The least 4 important features are fixed acidity, citric acid, density and total sulfur dioxide. When removing these features from the dataset and train the model again, the precision of the Decision Tree model considerably increases to 100%.

```
    print(classification_report(Y_test, prediction))
    print(confusion_matrix(Y_test, prediction))
[31] ✓ 0.1s
...                 precision    recall  f1-score   support

           0.0       1.00      1.00      1.00       768
           1.0       1.00      1.00      1.00       212

      accuracy                           1.00       980
     macro avg       1.00      1.00      1.00       980
  weighted avg       1.00      1.00      1.00       980

  [[768   0]
   [  0 212]]
```

Figure 12. Decision Tree Accuracy post feature filtering

In the case of the Random Forest model, the feature importance differs slightly from the Decision Tree. As seen in below bar chart, the density has a high importance unlike in the previous model where it was lower. It similarly has low importance for fixed acidity, citric acid, total sulfur dioxide with the addition of sulphates. When removing these features and training the model again, the precision of the Random Forest Classifier also increases to 100%.

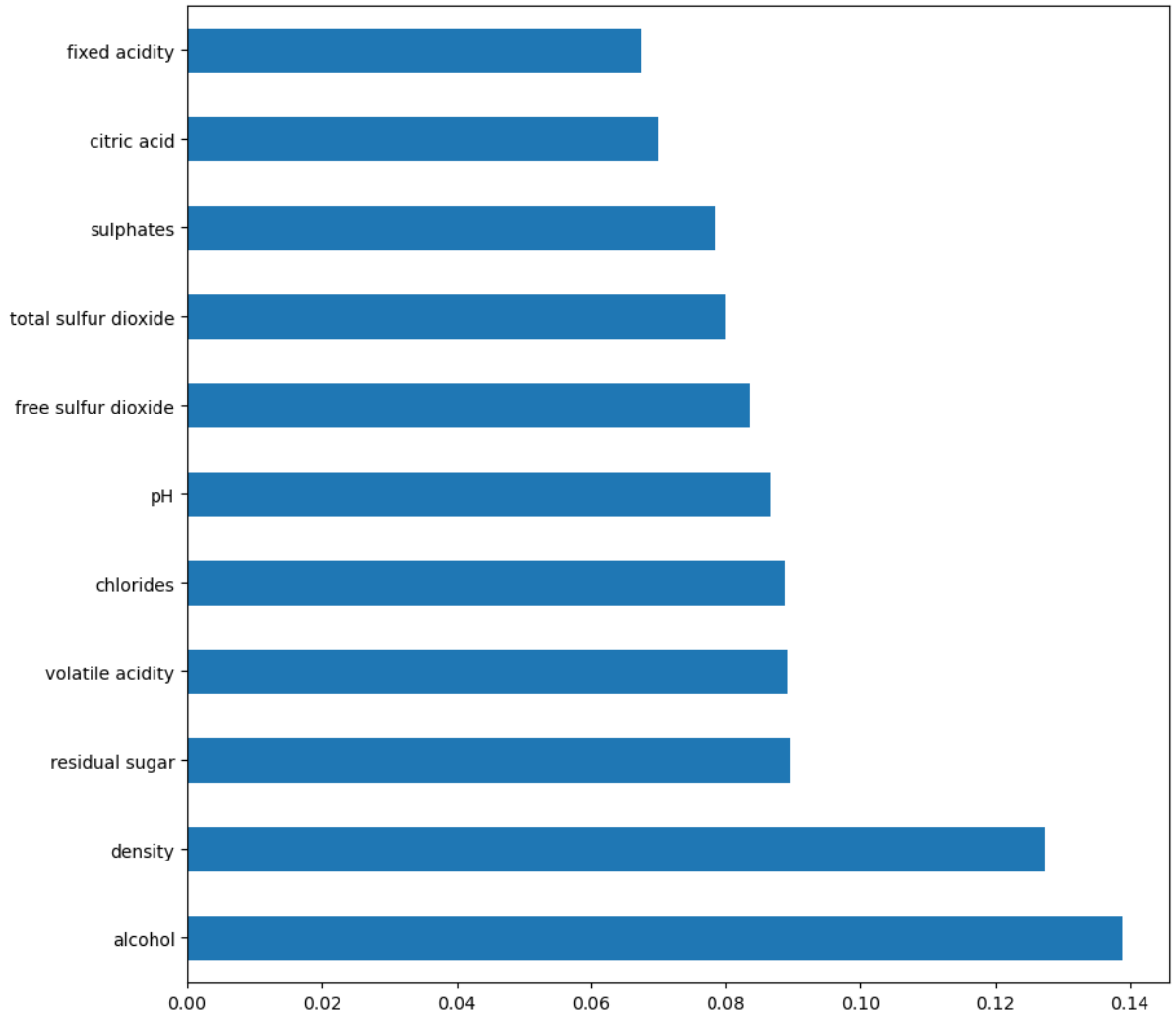

Figure 13. Feature Importance of Random Forest Classifier

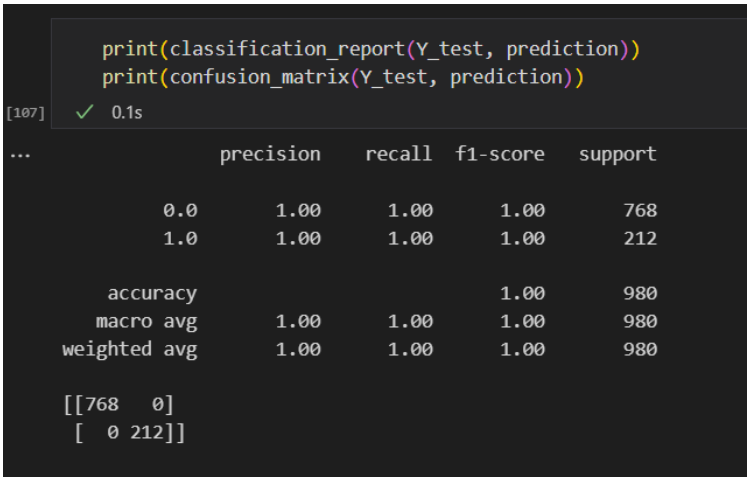

Figure 14. Random Forest Accuracy post feature filtering

In the decision tree model a subset of each feature is created at the root node, the pruning part then begins by looking at predictions of each decision nodes and removing the unwanted branches until the final prediction is obtained. The Entropy, which is a measure of uncertainty, decides which features are at the root nodes or decision nodes. The information gain also aids in the decision process of root/decision nodes and it calculates the feature's reduction in uncertainty (Saini, 2021).

Before removing the low importance features, the Random Forest algorithm generally outperformed the Decision Tree model since the Random Forest algorithm does not entirely depend on one single prediction rather it aggregates several predictions from a number of decision trees to reach the best result.

A similar comparison can be made in the case of Random Forest and Linear Regression, for the regression task, the Random Forest crowd approach can result in better performance with large datasets as Linear Regression uses a more general equation based approach while Random Forest can split the data into categories more accurately.

References:

GUWALI, S. 2021. *Linear Regression in machine learning* [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/linear-regression-in-machine-learning/ [Accessed].

RAJ, A. 2020. *A Quick and Dirty Guide to Random Forest Regression* [Online]. Available: https://towardsdatascience.com/a-quick-and-dirty-guide-to-random-forest-regression-52ca0af157f8 [Accessed].

SAINI, A. 2021. *Decision Tree Algorithm – A Complete Guide* [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/ [Accessed].