

Final Project

Evaluating the Performance of Different Classifiers

Prepared for

Dr. Christer Karlsson

By

Raiza Soares

8th May 2020

Table of Contents

- 1. Introduction**
- 2. Documentation of Data and Information**
- 3. Data Exploration**
- 4. Preprocessing**
- 5. Hypothesis**
- 6. Model and Evaluation method Analysis**
- 7. Analysis of Classifiers and Evaluation Methods**
- 8. Comparison of Accuracies generated by Orange**
- 9. Conclusion**

Brief Verbal Description of how the study was conducted

This study was conducted using five classifier models (Naïve Bayes, Decision Tree, K Nearest Neighbors, Random Forest, and Logistic regression) to classify heart disease data. I picked these classifiers since they are often used for different kinds of data. (example, Logistic regression is suitable for mostly linear solutions, while trees and KNN do better with nonlinear solutions). The intention of performing this study was to find out which classifier would work the best for our dataset. Three valuation methods were tested because it is helpful to understand hoe these work and how they affect the accuracy of our classifiers. This report goes through the steps of preprocessing and in each section, there is an explanation of why it is vital to perform that step.

Note: The .ipynb file used for coding is included in this project. You can open this file in Kaggle or Google Collaboratory to see the implementations. A pdf of this notebook is also included.

1. Introduction

In this project, we will compare five different classifiers and evaluate them using three different evaluation methods. We will also seek to compare some of these models in different software.

For this project, all coding was done in Kaggle Notebooks. The .ipynb file is attached, along with a pdf file of the notebook.

2. Documentation of Data and Information on Data

Name of dataset: Heart Disease UCI

It was found on Kaggle.

- Number of Objects (instances): 303
- Number of attributes: 13 + 1 target
- Attribute Description:
 1. age
 2. sex (1- male, 0-female)
 3. cp - chest pain type (4 values)
 4. trestbps - resting blood pressure
 5. chol - serum cholesterol in mg/dl
 6. fbs - fasting blood sugar > 120 mg/dl
 7. restecg - resting electrocardiographic results (values 0,1,2)
 8. thalach - maximum heart rate achieved
 9. exang - exercise induced angina (1=yes, 0=no)
 10. oldpeak - ST depression induced by exercise relative to rest
 11. slope - the slope of the peak exercise ST segment
 12. ca - number of major vessels (0-4) colored by fluoroscopy
 13. thal - 3 = normal; 6 = fixed defect; 7 = reversable defect
 14. target – 1-have disease, 0- do not have disease

3. Data Exploration

In this section we will explore our data through plots to get a better understanding of the dataset we are working with.

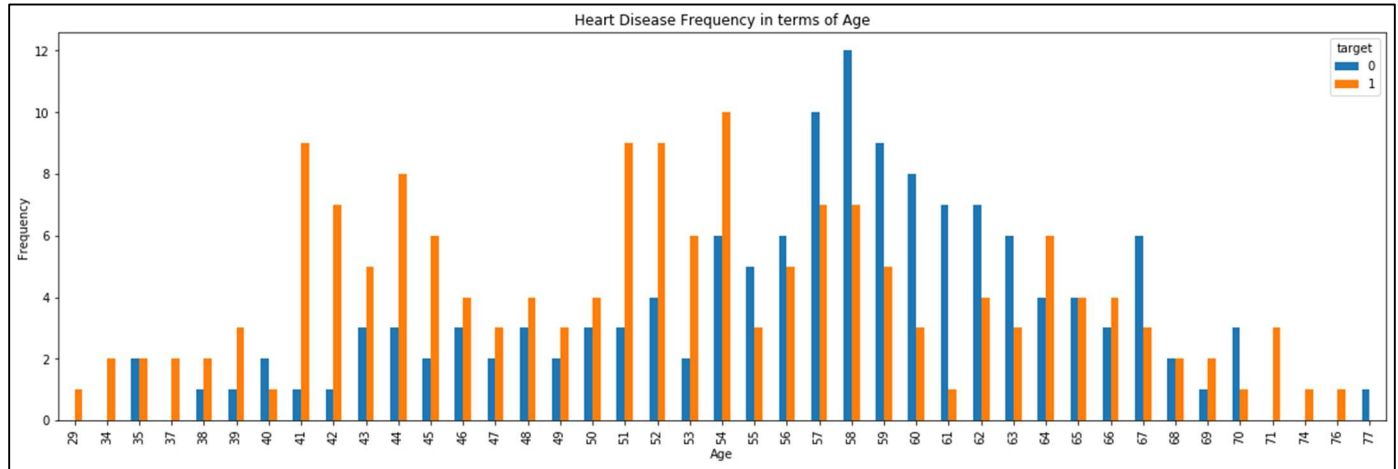


Figure 1: Heart Disease in terms of age

People 51- 55 have a higher frequency of heart disease, with the maximum frequency at age 54.

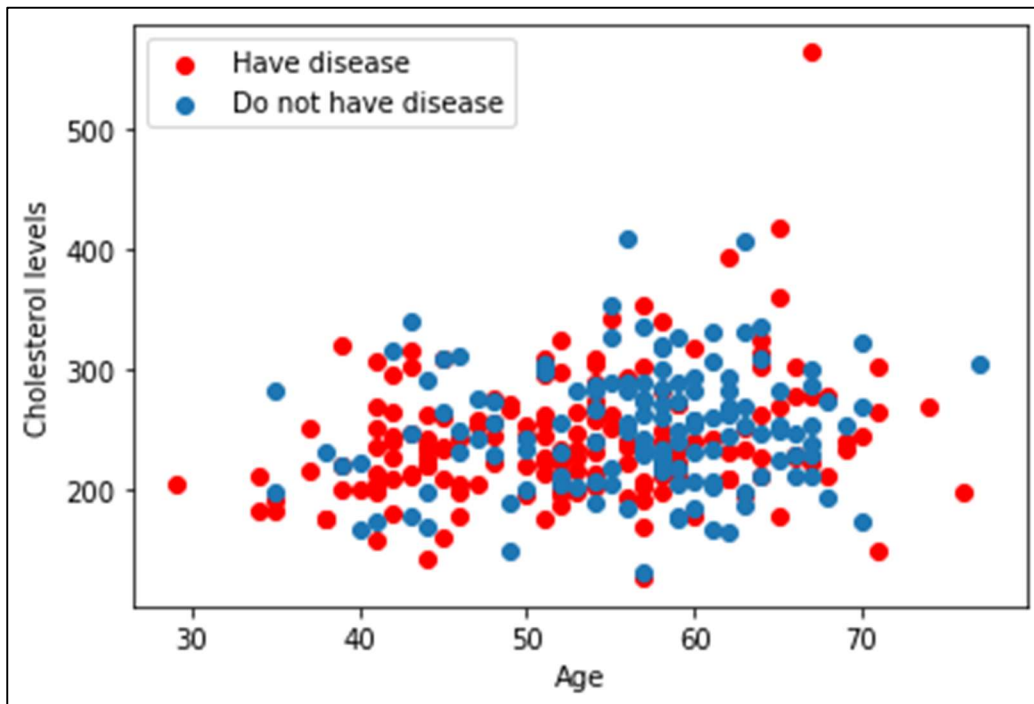


Figure 2: Scatter plot of cholesterol levels and age.

Total cholesterol levels less than 200 milligrams per deciliter (mg/dL) are considered desirable for adults. A reading between 200 and 239 mg/dL is considered borderline high and a reading of 240 mg/dL and above is considered high. As seen in the plot, most patients have cholesterol levels between 200 and 300 in the 40- 60 age range.

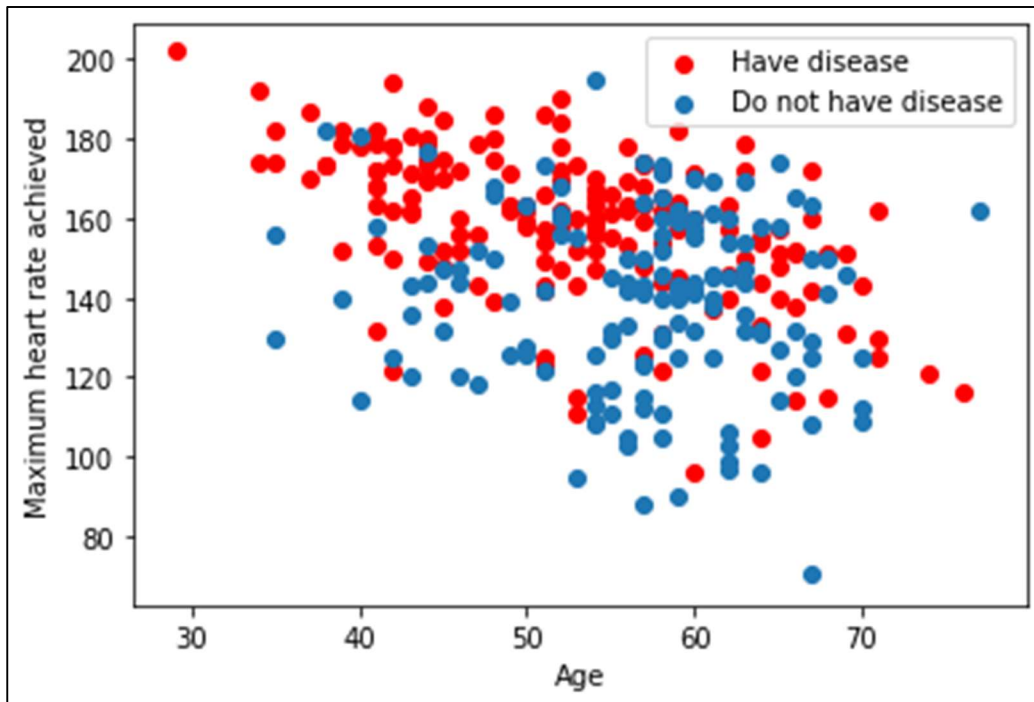


Figure 3: Scatter plot of Maximum heart rate and age.

The maximum heart rate in normal adults is 175. Most people who have a heart disease have a maximum higher than this. Most patients lie in the 40-60 age block.

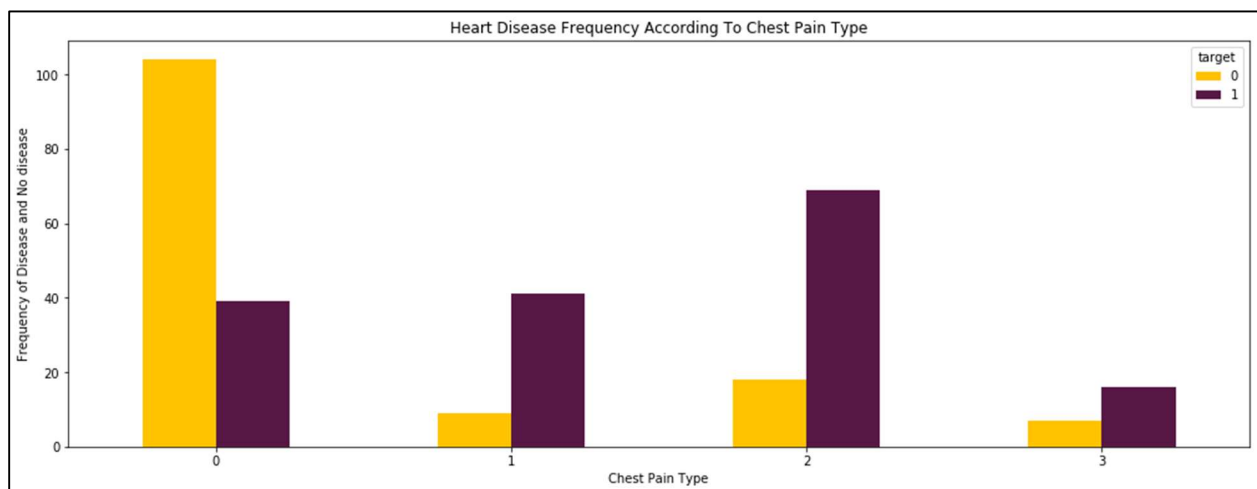


Figure 4: Frequency of heart disease based on chest pain type.

Most people with type 0 who were tested did not have a heart disease, while all other kinds of chest pain were indications of heart disease.

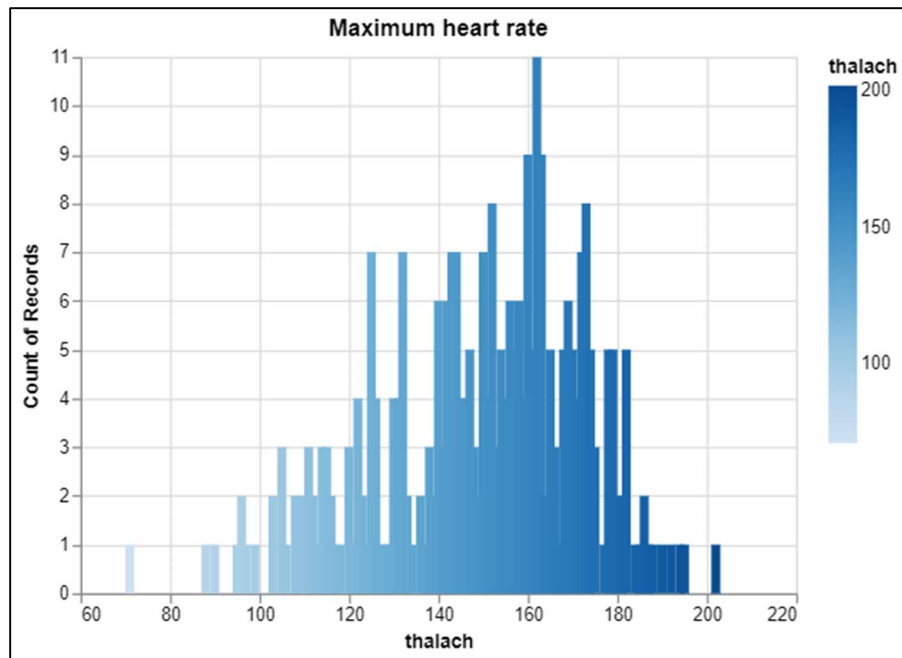


Image 5: Distribution of maximum heart rate achieved- skewed (left)

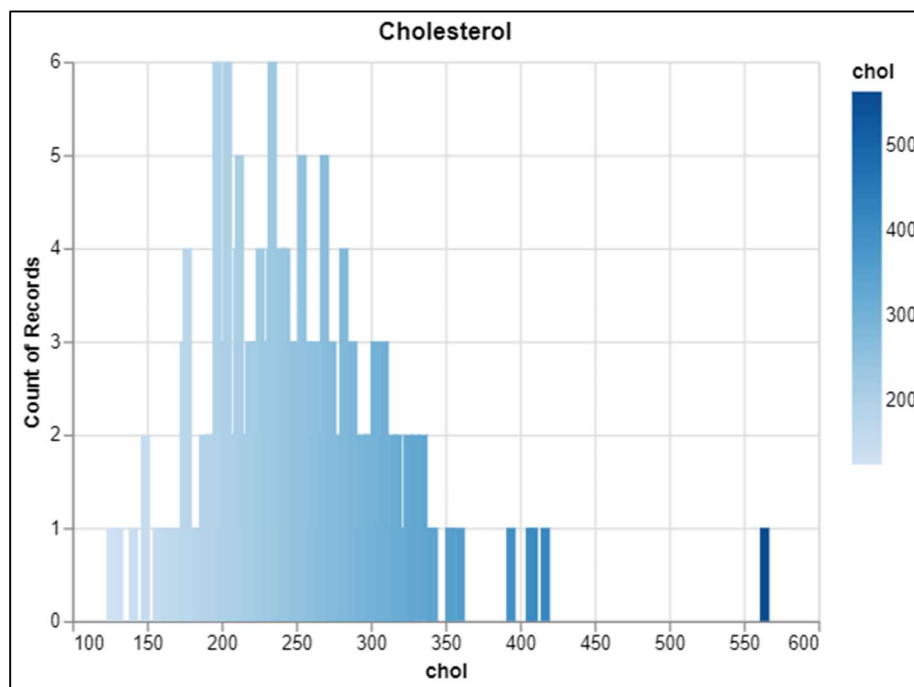


Image 6: Cholesterol distribution- slightly skewed (right)

Heat Map

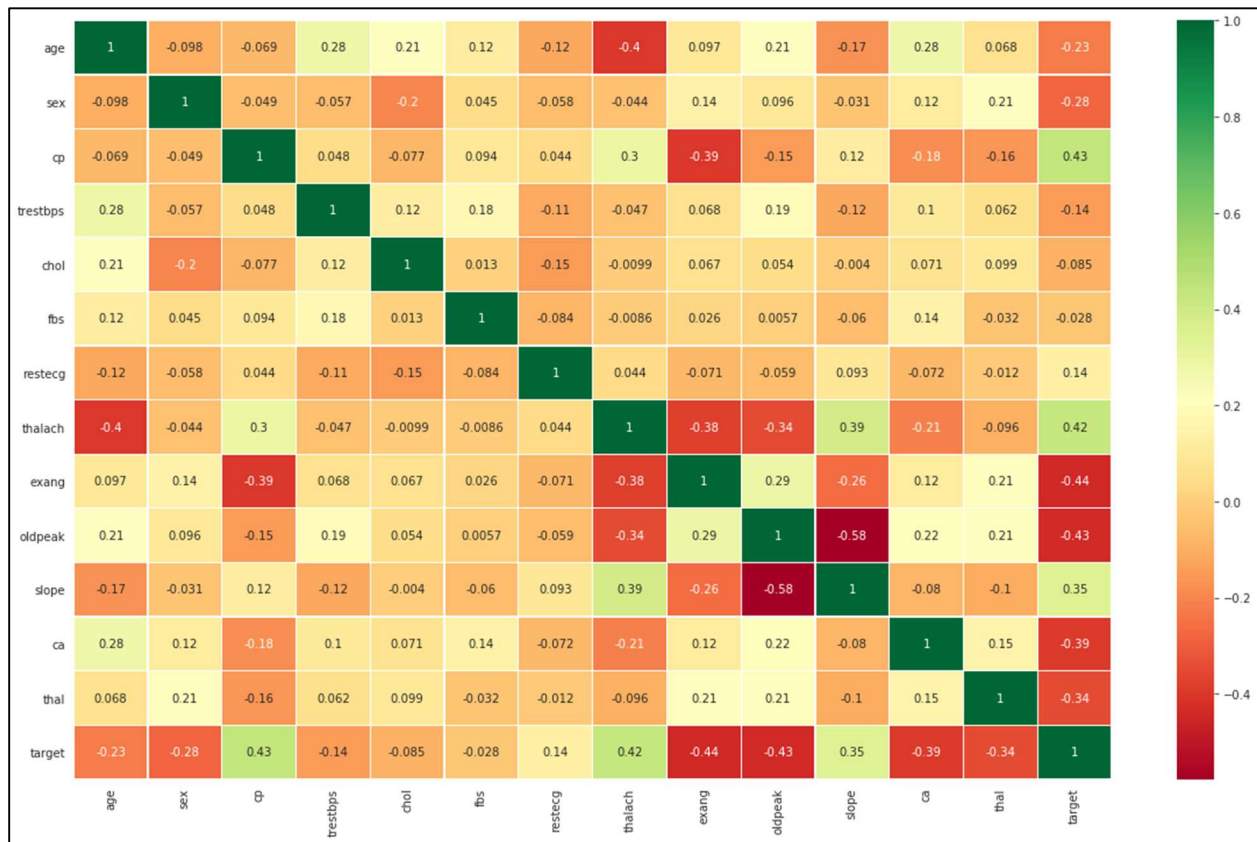


Figure7: Heat Map for Heart Disease Data

The first thing to note is that only the numeric features are compared as it is obvious that we cannot correlate between alphabets or strings. Let's look at some types of correlations.

Positive correlation: If an increase in feature A leads to increase in feature B, then they are positively correlated. A value 1 means perfect positive correlation.

Negative correlation: If an increase in feature A leads to decrease in feature B, then they are negatively correlated. A value -1 means perfect negative correlation.

If two features are highly or perfectly correlated, the increase in one leads to an increase in the other. This means that both the features have similar information and there is very little or no variance in information. While making or training models, we should try to eliminate redundant features as it reduces training time and many such advantages.

Now from the above heatmap, we can see that the features are not too correlated since all values are below 0.5.

4. Preprocessing

- **Data Quality Assessment**

Since data is taken from multiple sources which are often not too reliable, its important to consider this step. There are a few things we need to consider in this section:

- ✚ Missing Values

If we look at our dataset, we will notice that we do not have any missing values, so we do not need to worry about this area.

- ✚ Inconsistent Values

Inconsistent values can occur due to human error or maybe the information was misread while being scanned from a handwritten form. I performed a quick scan on my dataset to make sure there were no inconsistencies.

- ✚ Duplicate Values

We remove any duplicate records in order to not give that data object a bias. In my dataset I found three records that were duplicated and eliminated them.

- **Feature Aggregation**

Aggregations provide us with a high-level view of the data as the behavior of groups or aggregates is more stable than individual data objects. However, in this particular case, since we are conducting a study of which factors can best predict if someone has a heart disease or not, it doesn't make sense to aggregate on any particular attribute. We could attribute on age, creating age slots, but for now we will work on the data as in in order to get statistically unbiased results.

- **Feature Sampling**

We need to make sure that when we sample, we do not have an unbiased test set. (i.e. the data collected in the sample must be equally distributed throughout the entire dataset) We will use Kaggle to obtain an unbiased test set. (20% test and 80% train)

- **Dimensionality Reduction**

As the dimensionality of data increases, it becomes harder to perform any analysis on it. Since 'cp', 'thal' and 'slope' are categorical variables, we will turn them into dummy variables for logistic regression.

- **Feature Encoding**

Most of this data is already encoded, so we will not do any work in this sector. Example, chest pain 'cp' is encoded into 4 values (0,1, 2, 3) based on severity. Our target variable is binary, (0 or 1) based on has a disease or does not.

5. Hypothesis

To compare five different classifiers: Naïve Bayes, Decision Tree, K Nearest Neighbors, Random Forest and Logistic regression and evaluate them using three evaluation methods: Train-test set, K-fold cross validation and K-fold stratification. According to my work in data mining so far, we should see that K fold stratification should provide the highest accuracy with the random forest and KNN models. We will test this hypothesis in the next sections. Further on we will also compare some models in Orange and Kaggle and see how the accuracies vary.

6. Model and Evaluation method Analysis

We explore five different classifiers: Naïve Bayes, Decision Tree, K Nearest Neighbors, Random Forest and Logistic regression and compare which gives us the best results accuracy wise. I will also test each of these models with three evaluation methods: Train-test set, K-fold cross validation and K-fold stratification, and then compare them against each other.

6.1. Evaluation Method 1: Train-Test Set

This is a very common evaluation method and is often very good at predicting accuracies. For our dataset, we split our data into 75% train data and 25% test data. This was done using the “train-test-split” command in python. (Code can be found on the attached Kaggle workbook). Most models were used from the sk learn library and some were coded up manually.

Logistic Regression

Logistic regression provided a test accuracy of 82.89% using the LogisticRegression() function in sk learn library.

Naïve Bayes

Accuracy of Naïve Bayes was found to be 85.53% using the sk learn library.

K Nearest Neighbors

Initially I set the k value to be 2 with the sk learn library and ran the code. This provided an accuracy of only 77.63%. I then wrote some more code to find the best k value to find the highest accuracy. The highest accuracy was found to be 85.53% when k=5 and 6.

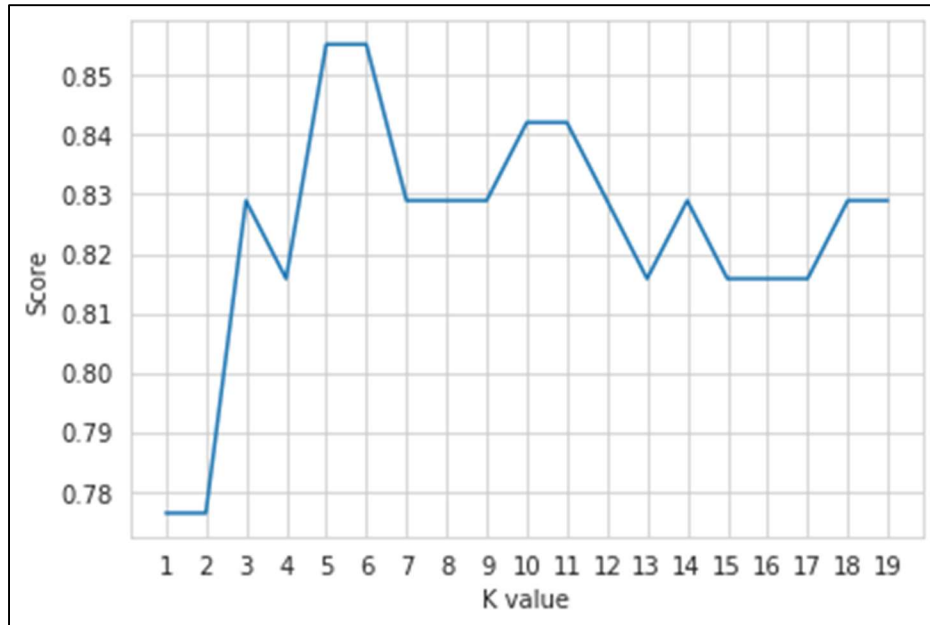


Figure 8: Finding best K value for highest accuracy

The graph peaks at k values= 5 and 6.

✚ Decision Tree

Using the decision tree classifier, from the sk learn library, accuracy was found to be only 76.32%. Giving it one of the lowest accuracies so far.

✚ Random Forest

The random forest classifier provided an accuracy of 86.84%

We now compare how each of these models performed in our first evaluation method.

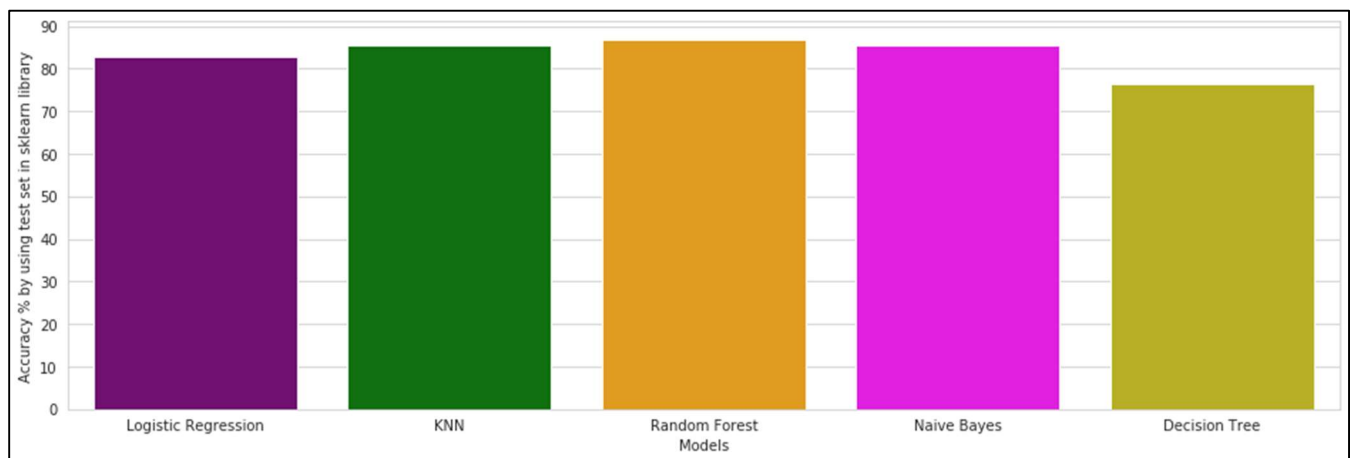


Figure 9: Comparison of Models using Train-Test evaluation method

6.1.2. Analysis

From figure 9, we realize that the Random Forest model performed the best at classifying this data, followed by K nearest neighbors. It is interesting to note that the decision tree algorithm performed the worst. Splitting the data into train and test is one of the most common approach to test the accuracy of a classifier, but it does not always do the best job of predicting results.

We will compare all evaluation methods in section 7.

6.2. Evaluation Method 2: K-fold Cross Validation

We will now code up do a 10-fold cross validation in Kaggle Notebooks using the sk learn library.

Logistic regression

```
Score from each iteration: [0.8064516129032258, 0.7419354838709677, 0.8709677419354839, 0.7741935483870968, 0.8387096774193549, 0.8387096774193549, 0.8064516129032258, 0.8709677419354839, 0.967741935483871, 0.967741935483871]
Average kfold score: 0.8483870967741935
```

Figure 10: K fold scores for Logistic regression

We note that the average K fold score is 0.848387097

Random Forest Generator

```
Score from each iteration: [0.8064516129032258, 0.8387096774193549, 0.7741935483870968, 0.7741935483870968, 0.7741935483870968, 0.8064516129032258, 0.8064516129032258, 0.967741935483871, 0.7741935483870968, 0.8064516129032258]
Average kfold score: 0.8129032258064516
```

Figure 11: K fold scores for Random Forest Generator

We note that the average K fold score is 0.8129032258

Naïve Bayes Classifier

```
Score from each iteration: [0.8709677419354839, 0.8387096774193549, 0.7096774193548387, 0.8387096774193549, 0.8709677419354839, 0.8709677419354839, 0.7741935483870968, 0.8064516129032258, 0.8064516129032258, 0.8064516129032258]
Average kfold score: 0.8193548387096774
```

Figure 12: K fold scores for Naïve Bayes Classifier

We note that the average K fold score is 0.819354838

Tree classifier

```
Score from each iteration: [0.7419354838709677, 0.7096774193548387, 0.7096774193548387, 0.9032258064516129, 0.8387096774193549, 0.7419354838709677, 0.7741935483870968, 0.6774193548387096, 0.7096774193548387, 0.6129032258064516]
Average kfold score: 0.7419354838709677
```

Figure 13: K fold scores for Tree Classifier

We note that the average K fold score is 0.74193548387

 KNN

```
Score from each iteration: [0.5483870967741935, 0.7419354838709677, 0.45161290322580644, 0.6129032258064516, 0.7096774193548387, 0.5483870967741935, 0.5483870967741935, 0.6774193548387096, 0.5161290322580645, 0.5806451612903226]
Average kfold score: 0.5935483870967742
```

Figure 14: K fold scores for KNN Classifier

We note that the average K fold score is 0.593548387

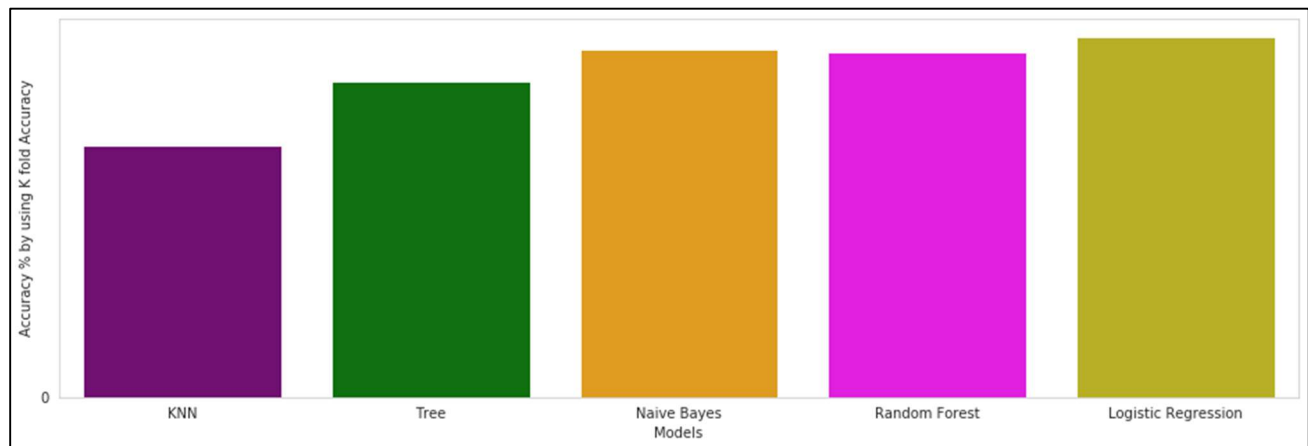


Figure 15: Comparison of classifiers using k fold evaluation method

6.2.2. Analysis

If we look at the figure above, we realize that the K fold evaluation method produced results that were fairly different than our first evaluation method. With K- folds, Logistic Regression performs the best while KNN performs the worst. In our earlier evaluation method, KNN was recorded to have a significantly higher accuracy as compared to our K fold evaluation method. We will discuss how we decide on which method to pick based on their credibility in section 5.

6.3. Evaluation Method 3: Stratified K-fold Cross Validation


We will now do a Stratified 10-fold cross validation in Kaggle Notebooks.

 KNN

```
Score from each iteration: [0.7096774193548387, 0.5806451612903226, 0.6129032258064516, 0.7666666666666667, 0.7, 0.6666666666666666, 0.7666666666666667, 0.6333333333333333, 0.6333333333333333, 0.5]
Average Stratified kfold score: 0.656989247311828
```

Figure 16: Stratified K fold scores for KNN Classifier

Average score= 0.6569892473118

 Tree

```
Score from each iteration: [0.8709677419354839, 0.7096774193548387, 0.6774193548387096, 0.8, 0.7, 0.8, 0.6, 0.7666666666666667, 0.6666666666666667, 0.7666666666666667]
Average Stratified kfold score: 0.7358064516129033
```

Figure 17: Stratified K fold scores for Tree Classifier

Average score= 0.735806451612

Naïve Bayes Classifier

```
Score from each iteration: [0.8387096774193549, 0.7741935483870968, 0.7741935483870968, 0.8333333333333334, 0.8333333333333334, 0.7333333333333333, 0.9, 0.9, 0.8, 0.8]
Average Stratified kfold score: 0.8187096774193549
```

Figure 18: Stratified K fold scores for Naïve Bayes Classifier

Average score= 0.818709677

Random Forest

```
Score from each iteration: [0.7741935483870968, 0.8387096774193549, 0.7741935483870968, 0.8333333333333334, 0.9, 0.8666666666666667, 0.8666666666666667, 0.7666666666666667, 0.8333333333333334, 0.7]
Average Stratified kfold score: 0.8153763440860213
```

Figure 19: Stratified K fold scores for Random Forest

Average score= 0.815376344

Logistic Regression

```
Score from each iteration: [0.7741935483870968, 0.9032258064516129, 0.8709677419354839, 0.8, 0.8333333333333334, 0.9, 0.8, 0.8666666666666667, 0.8, 0.8333333333333334]
Average Stratified kfold score: 0.8381720430107527
```

Figure 20: Stratified K fold scores for Logistic Regression

Average score= 0.838172043

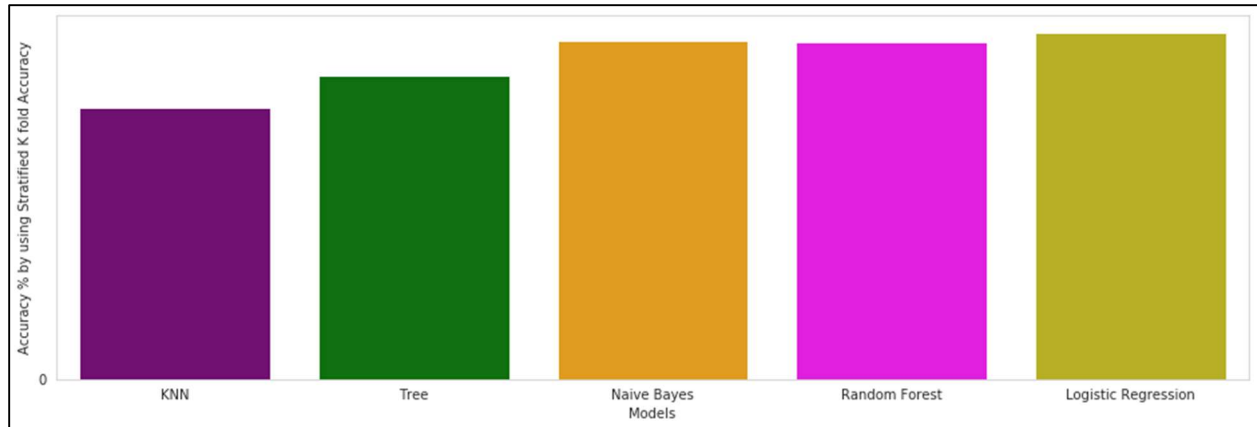


Figure 21: Comparison of classifiers using Stratified k fold evaluation method

We notice that Stratified K fold evaluation and K fold evaluation performed similarly with logistic regression performing the best, followed by Naïve Bayes and Random Forest. We see that even though the train and test evaluation method gave us a general idea of which classifiers did better than others, it produced results that were very different as compared to K fold and Stratified K-fold methods.

7. Analysis of Classifiers and Evaluation Methods

7.1. Evaluation Methods

In a train-test split, we take a subset of the data and use it as test data. (In our case 25% of the data is used for the test set). We cannot guarantee that the subset used will be a good representative of the dataset. For example, what if our test data only has people from a certain age group. This will result in overfitting. Thus, we see that this method is not completely trustworthy when gaining accuracies for our models.

In K-Folds Cross Validation we split our data into k different subsets (or folds). We use k-1 subsets to train our data and leave the last subset (or the last fold) as test data. We then average the model against each of the folds and then finalize our model. After that we test it against the test set. In this method, the test set is continuously being changes and we get a good representation of the data in all our test sets. We can then average our accuracies and find a mean accuracy. Cross validation thus is a better evaluation method since it avoids biases in the test data.

Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole. So Stratified K- Fold Cross validation made sure that each test set was a good representation of the data set. In our case, the accuracies were pretty similar in Stratified K- fold and K fold, so our test sets in K fold were pretty good as well. Stratified K Fold first shuffles your data and then splits the data into n_splits parts. Thus Stratified K fold is the best of the three evaluation methods.

7.2. Classifiers

Keeping in mind that we will use the results from the Stratified K folds evaluation method, we will attempt to discover why certain models performed better than other with respect to our data set.

Logistic regressor model is not a regression model, but a classification model. It uses a logistic function to frame binary output model. The output of the logistic regression will be a probability ($0 \leq x \leq 1$) and can be used to predict the binary 0 or 1. Logistic Regression acts somewhat very similar to linear regression. It also calculates the linear output, followed by a stashing function over the regression output. It supports only linear classification problems which may be the reason why it was performing extremely well on our data, while other classifiers like KNN and decision tree were not.

Decision trees support nonlinearity. Decision trees are very good at approximating highly nonlinear models with complex interactions. Our data has many feature variables. Decision tree pruning may neglect some key values in training data, which can lead the accuracy for a toss. When there are large number of features with less data-sets (with low noise), logistic regressions may outperform Decision trees/random forests.

The basic logic behind KNN is to explore your neighborhood, assume the test datapoint to be similar to them and derive the output. In KNN, we look for k neighbors and come up with the prediction. KNN is a non-parametric model, whereas LR is a parametric model. KNN is slow in real time as it has to keep track of all training data and find the neighbor nodes. Even though we selected K as the best possible value, this method failed since There is no training involved in KNN. During testing, k neighbors with minimum distance, will take part in classification /regression. If datapoints are fairly close to each other, it is difficult to correctly classify them using this method.

We now bring up the question as to why the random forest model performs better than the decision tree even though its basic unit is a decision tree structure. Random Forest is a collection of decision trees and average/majority vote of the forest is selected as the predicted output. Random Forest model will be less prone to overfitting than Decision tree and gives a more generalized solution. Random Forest is more robust and accurate than decision trees.

Naïve Bayes performs pretty well, but it can be improved if certain attributes are given more preference. It assumes that all predictors have an equal effect on the outcome. Our dataset is derived from a real-life situation, and in most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

Before we conclude, it is also worthwhile to not that each of these models can have varying accuracies based on what software you run them on, or if you code them up manually. We will compare K fold cross validation in Orange for our Naïve Bayes and decision tree just to get a feel of this idea.

8. Comparison of Accuracies generated by Orange

➤ Naïve Bayes

We find the overall accuracy to be 0.838

Cross validation with 20 folds:

Selection Fold	Accuracy
1	0.812
2	0.875
3	0.875
4	0.867
5	1
6	0.8
7	0.733
8	0.867
9	1
10	0.733
11	0.933
12	0.733
13	0.667
14	0.933
15	0.933
16	0.933
17	0.867
18	0.8
19	0.667
20	0.867

Mean accuracy of cross validation with 20 folds: 0.84475

Bootstrap accuracy: 0.838

➤ Decision Tree

Overall Accuracy= 0.759

Selection Fold	Accuracy
1	1
2	0.938
3	0.938
4	0.933
5	1
6	1
7	0.667
8	0.867
9	1

10	0.867
11	1
12	0.933
13	1
14	1
15	1
16	0.933
17	1
18	0.933
19	1
20	1

Mean K means accuracy= 0.8571

Bootstrap= 0.964

Although the accuracy of Naïve Bayes is similar when coded up in Kaggle and Orange, the tree K fold accuracy is drastically different. These variances exist and its important to be aware of these when picking the software, model, and evaluation method for your data set.

9. Conclusion

Based on our findings we conclude that for our data set, Stratified K folds is found to be the most accurate evaluation method, and Linear regression followed by Naïve Bayes and Random forest are the best classifiers for this model. It is important to note that each data set is different, and we need to pick models and methods to fit the needs of our data.

10. References

- Kohavi, R., A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection (1995)
- Carnegie Mellon University
<https://www.cs.cmu.edu/~schneide/tut5/node42.html>
- Towards Data Science
<https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222>
<https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>
<https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6>

