Comparing Classifiers

Chemeng 787: Machine Learning: classification models

Kyle D’mello & Obada Al baba

2020

# Introduction

The objective of this project is to compare 3 different classification methods to determine which one is better and what their discrepancies are. Gender recognition by voice data from the Kaggle Machine learning repository was used to test the classifiers. The data consist of 3,168 recorded voice samples with information such as mean frequency, standard deviation of the frequency, and other related frequency data. Each sample was classified as male or female [1]. When this was implemented in the code, the male labels were converted to 1/True and the female labels were converted to 0/False. The classifiers used in this project were K-Nearest Neighbours (KNN), Naïve Bayes (NB) and AdaBoost (AB). These were used since they are all able to classify numerical data with relatively high effectiveness.

# Procedure

The raw data was imported as a continuous sample, split by male and female. The data was first randomized, and the labels were converted from strings to zeros and ones. The data was then split into a training set of 2000 samples and a testing set of 1168. The first experiment was KNN. To first determine the value of neighbours, cross validation was used. The 2000 sample training set went through 4-fold cross validation where at each fold 4 different values of neighbours were tested, and their accuracies were stored. The k values used for this were 11, 21, 31 and 41. K values under 10 were not considered. Using a k value under 10 leads to some overlap in the scores used in ROC curves which is problematic. Once this step was completed the average of the cross validation for each neighbour was calculated, and the one with the highest average accuracy was chosen. This was then used to train and test the KNN algorithm with the full training data.

Next the NB algorithm was trained by separating the training data between male and female, and then finding the means and standard deviations of the individual variables. To test the data, a probability density function was created and the probability between male and female was compared for each sample choosing the one with the higher probability. For AdaBoost the algorithm was implemented in 2 ways the first was through Sklearn, this allowed for an ROC (Receiving Operating characteristic) curve to be created. The second was an AB code written from scratch found online [2]. This was used to create the confusion matrix.

Finally, to compare the performance of all these algorithms 3 metrics were used for each algorithm: computational times for training and testing, a confusion matrix, and an ROC curve. To compute the times, “time.pref\_counter()” was used before and after the testing and training parts of each code. An algorithm was also created to generate a confusion matrix and to plot the ROC curves.

For each of the algorithms, we compared the performance of our implementations of the algorithms to SKlearn’s implementations.

# Results

First looking at the computing times, it takes AdaBoost the longest to train since it created 10 classifiers, but it has the shortest testing time since it only must check which side of the decision boundary the test data is. KNN is the second longest to test since it must measure the distance form a test point to all the training points for all test points. KNN is also slightly different to the other algorithms since the training and testing are essentially occurring at the same time. Finally, Naive Bayes has the shortest time since to train the data, the mean and standard deviation must be calculated, and to test the data the male and female probabilities must be compared. Looking at the errors, AdaBoost had the lowest error with only 38 misclassifications, and KNN had the highest error with 372 misclassifications.

Table : Timing and error results of different classification methods

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Train Time (s) | Test Time (s) | Error |
| KNN | 7.943 | N/A | 372 |
| Naïve Bayes | 0.013 | 0.329 | 139 |
| AdaBoost | 97.096 | 0.002 | 39 |

Although the number of errors may be a good metric for the performance of an algorithm, a deeper look is necessary to fully judge the performance of a classification model. Evaluation tools such as the confusion matrix and the ROC curve are used to compare the three models. The results from the confusion matrices and ROC curves show that our AdaBoost is the best performer here.

For all the classifiers created, they were compared with Sklearns implementation to validate that the algorithms were working properly. As seen in the code, the errors and shape of the ROC curves were the same for the cases where the comparison was implemented.

# Discussion

AdaBoost is clearly the best performing classifier since it has the highest accuracy. it seems to have performed well since it combined 10 of the best weak linear classifiers and combined them to create a single boundary layer. it may have chosen the 10 classifiers on the features with the highest number of definitions which then made it easier to define the test data. Naive bayes also worked relatively well since it uses the probabilities of the features for a specific label and compares it to the probability of the other label. Since the data uses the frequency of human voices to determine their gender and based on anecdotal experience Naive Bayes should work well to discriminate lower frequency male voices from higher frequency female voices, the only problem it would face would be the overlapping region. Finally looking at KNN it underperforms significantly compared to the others. It works by looking at its closest neighbors and then takes a majority vote to predict the labels. This seems to be inaccurate for the overlapping region since depending on the number of neighbours the performance predicted value could flip-flop. Furthermore, the other methods have a higher emphasis on the feature information which helped in making a decision.

Many improvements can be implemented to improve the performance of the machine learning algorithms. Firstly, feature selection techniques and dimensional reduction can be used to improve testing/training times. When analysing the T1/T2 plot, T2/T3 plot and box/whiskers plots shown in the appendix, we can see that there is a considerable number of outliers. The outliers can be excluded during the training process or addressed using normalization techniques.

# Appendix A

## Data Visualization/Analysis:

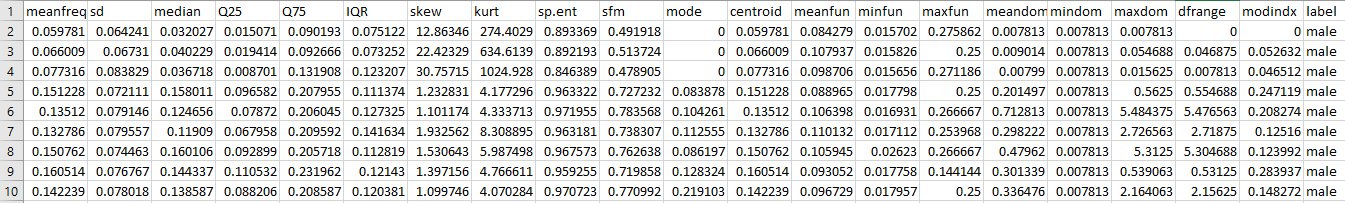


Figure : Raw Data (Data set containing 20 variables and 3168 samples) [1]

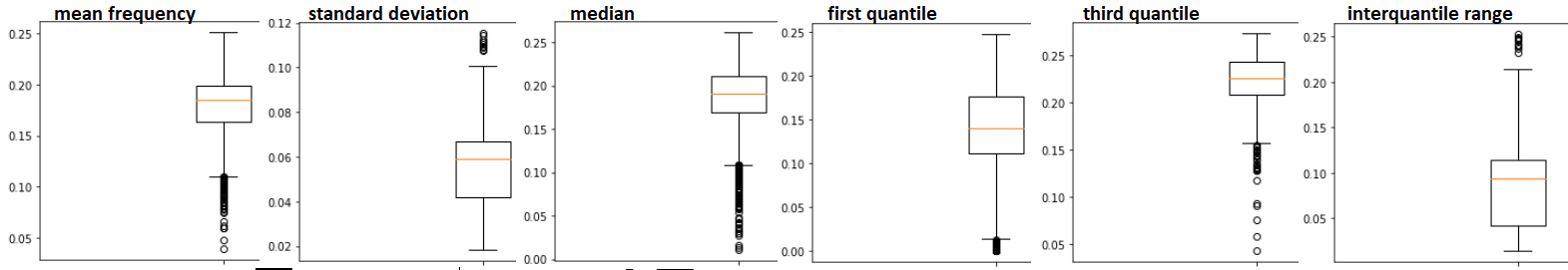


Figure :Box and Whiskers Diagrams

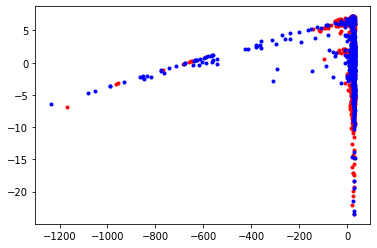


Figure :T1/T2 Plot

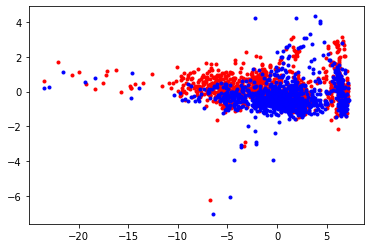


Figure :T2/T3 Plot

## Model Evaluation:

Table : Confusion Matrix for KNN

|  |  |
| --- | --- |
| TP: 407 | FP: 202 |
| FN: 154 | TN: 405 |

Table :Confusion Matrix for Naive Bayes

|  |  |
| --- | --- |
| TP: 495 | FP: 72 |
| FN: 66 | TN: 535 |

Table : Confusion Matrix for AdaBoost

|  |  |
| --- | --- |
| TP: 542 | FP: 19 |
| FN: 19 | TN: 588 |

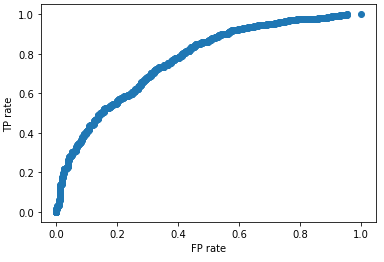


Figure :ROC curve for the KNN model using our algorithm

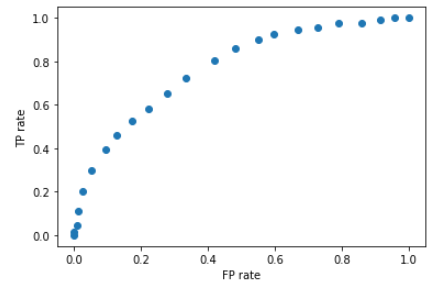


Figure :ROC curve for the KNN model using SKlearn

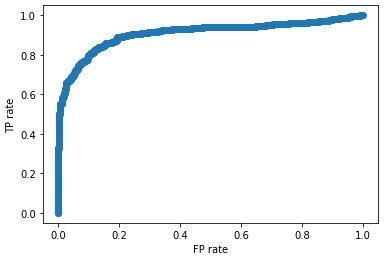


Figure :ROC curve for the Naive Bayes model using our algorithm

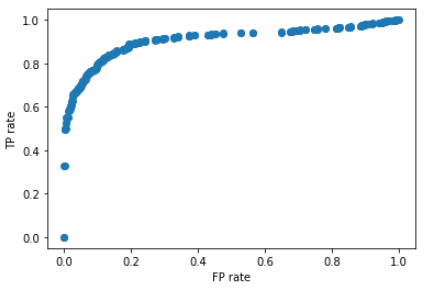


Figure :ROC curve for the Naive Bayes model using SKlearn

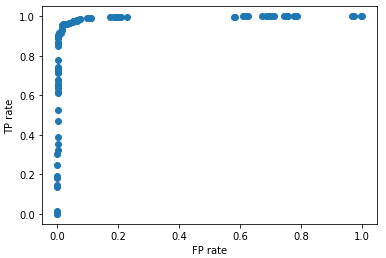


Figure : ROC curve for the AdaBoost model using Sklearn

# Bibliography

|  |  |
| --- | --- |
| [1] | K. Becker, "Gender Recognition by Voice," 2016. [Online]. Available: https://www.kaggle.com/primaryobjects/voicegender?select=voice.csv. [Accessed 14 12 2020]. |
| [2] | P. Engineer, "AdaBoost in Python - Machine Learning From Scratch 13 - Python Tutorial," Python Engineer, 16 03 2020. [Online]. Available: https://www.youtube.com/watch?v=wF5t4Mmv5us. [Accessed 19 12 2020]. |