Instructions: Replace all bracketed text with text for your project. Feel free to add. Leave the document structure in place. Delete these instructions.

[Hadi Baajour, 96316680]

CIS4930 Individual Coding Assignment
Spring 2023

1. Problem Statement

One of the most fundamental aspects of a human being is the ability to disseminate information to other human beings through a specific language. As technology advances, humans are finding ways to use technological devices in order to capture, store, and use human language for every day tasks. One of the most important machine learning technologies that has increased in popularity and necessity for understanding languages in a deeper level is natural language processing (NLP). Many of the different dialog systems used today are built of the architecture and design discovered through studies in NLP. As dialog systems advance, people need to find a way for these devices to properly understand human spoken languages and figure out whether the speech is positive or negative in connotation. This problem needs to be solved in order train machines to better understand people so that it can assist them with their daily needs. To solve this problem, we must create and train a model that will be able to determine the connotation and intention of textual speech (which has been generated by automatic speech recognition systems) in order to determine and produce an appropriate response.

2. Data Preparation

The textual data provided was prepared in a number of different ways. First, the training and testing data (which were both stored in csv files) were read by the machine and turned into data frames in order to make it easier to filter through the document. The data frames were then checked for their contents using the .head() function and all missing values were removed from the data frames to avoid the machine conflicting with data that is not even present.

After this is done, the text preprocessing stage commences. This is where the textual data is "cleaned" so that it is easier for the machine to interpret the data without having unnecessary text that won't be used by the model. The program will make all the letters lower case, remove all sorts of numerical values, expand all contraction words, remove all special characters, and remove all stop words.

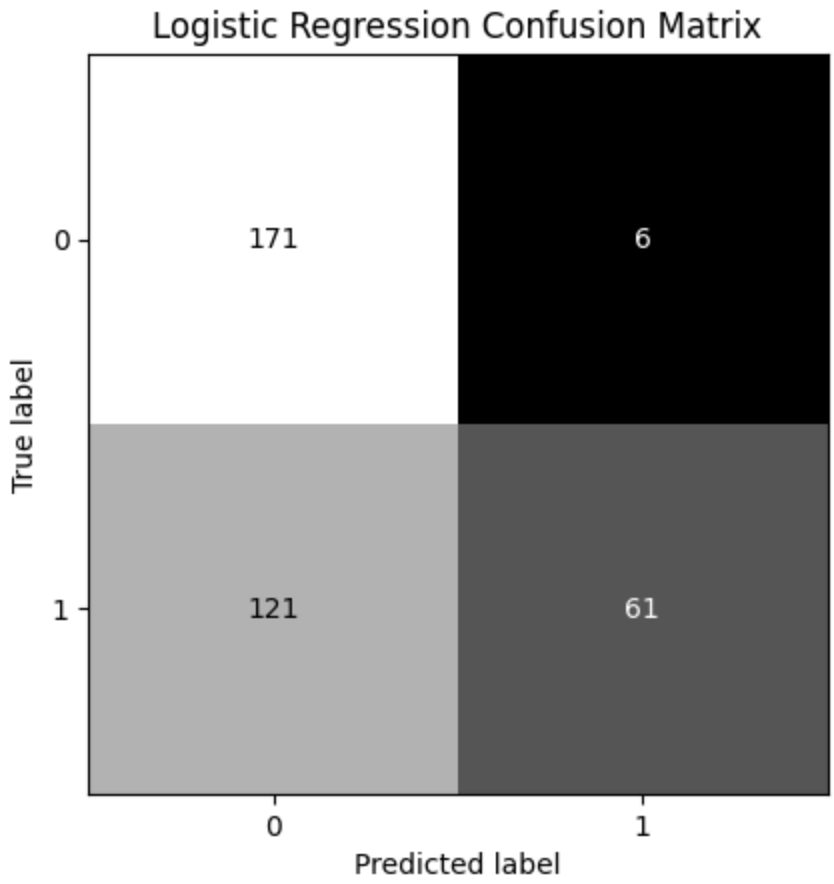
*For feature extractions, I used Bag of Words and TF*IDF which were bother used in my classification models. I also used Word2Vec, however this was not added in my classification models.*

3. Model Development

- Model Training
  - *[Describe the training phrase, which may include what models did you select, how you split training/validation/test sets, training epochs, and any other parameters.]*
  - *For the training phase, I selected five different models to use with my feature extraction algorithms. Those five models were logistic regression, k-neighbors classification, support vector classification, gaussian naïve bayes, and random forest classifier. The two features used in these models were the bag of words and tf*idf. Each of the data from both these feature extraction methods were fit and transformed in order to be places in the ML models. In this assignment, there was no need to split the testing and training data because they were already given in two different csv files and made into two separate data frames prior to using the models.*
- Model Evaluation

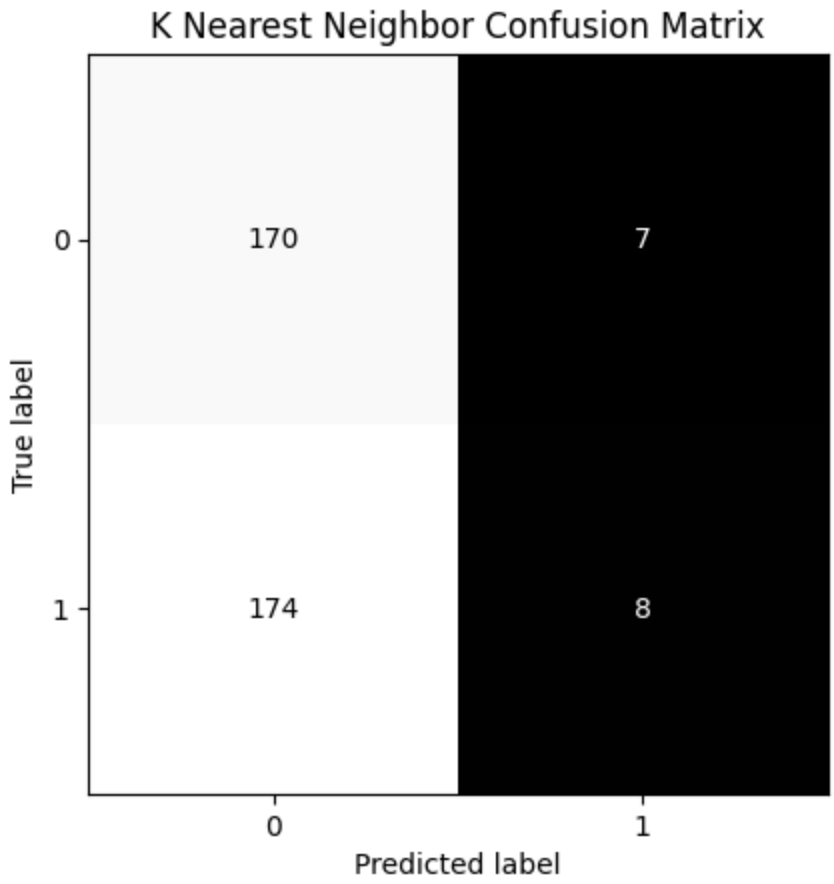
  Bag of Words
  - *Logistic Regression*

	precision	recall	f1-score	support
0	0.59	0.97	0.73	177
1	0.91	0.34	0.49	182
accuracy			0.65	359
macro avg	0.75	0.65	0.61	359
weighted avg	0.75	0.65	0.61	359



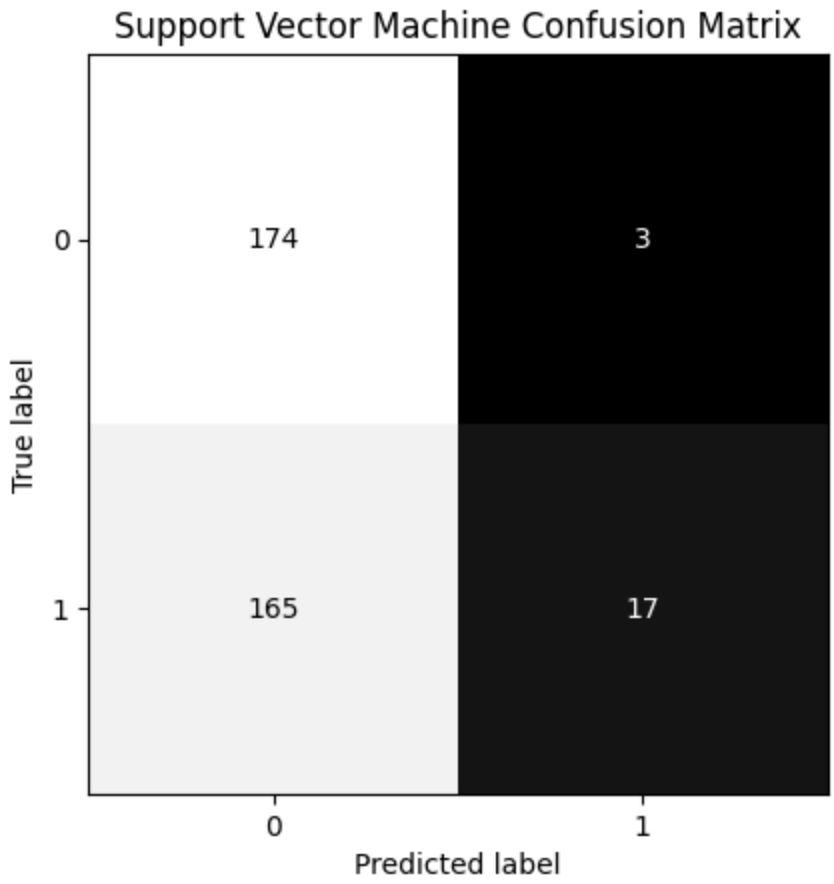
- *K Nearest Confusion Matrix*

	precision	recall	f1-score	support
0	0.49	0.96	0.65	177
1	0.53	0.04	0.08	182
accuracy			0.50	359
macro avg	0.51	0.50	0.37	359
weighted avg	0.51	0.50	0.36	359



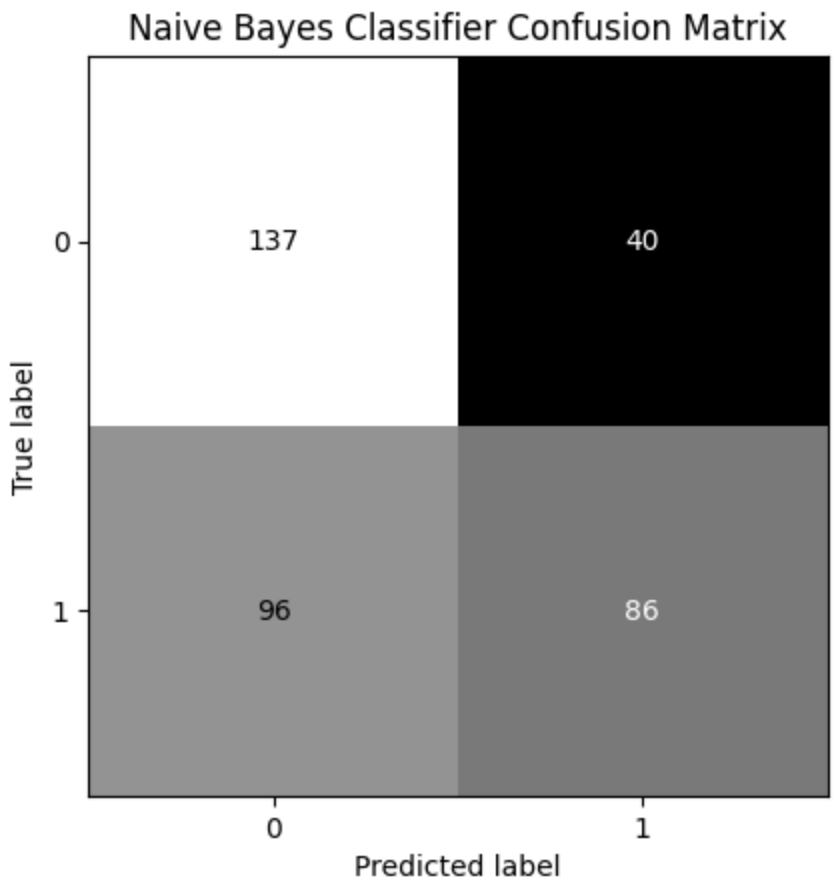
- *Support Vector Machine*

	precision	recall	f1-score	support
0	0.51	0.98	0.67	177
1	0.85	0.09	0.17	182
accuracy			0.53	359
macro avg	0.68	0.54	0.42	359
weighted avg	0.68	0.53	0.42	359



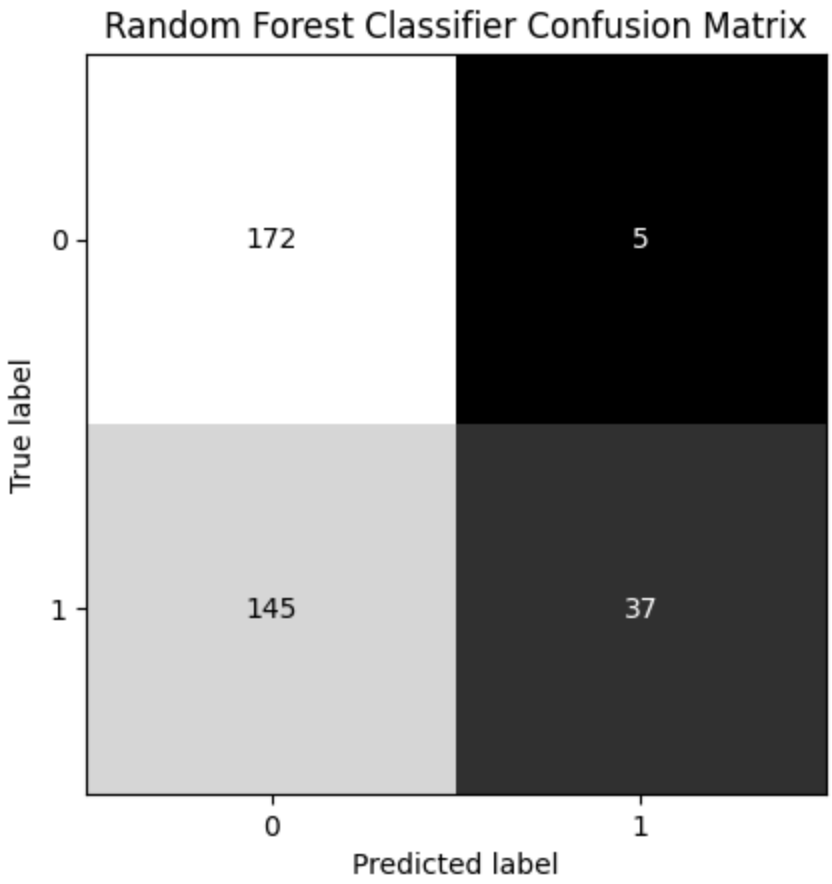
- *Naive Bayes Classifier*

	precision	recall	f1-score	support
0	0.59	0.77	0.67	177
1	0.68	0.47	0.56	182
accuracy			0.62	359
macro avg	0.64	0.62	0.61	359
weighted avg	0.64	0.62	0.61	359

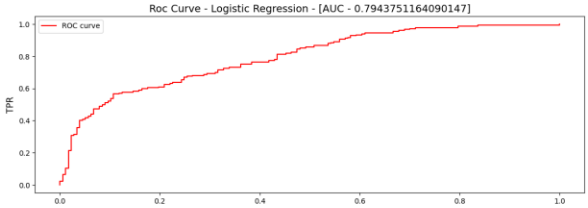


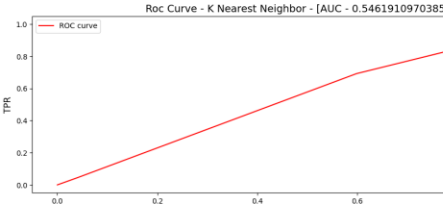
- *Random Forest Classifier*

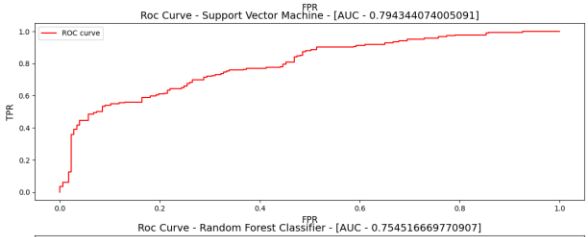
	precision	recall	f1-score	support
0	0.54	0.97	0.70	177
1	0.88	0.20	0.33	182
accuracy			0.58	359
macro avg	0.71	0.59	0.51	359
weighted avg	0.71	0.58	0.51	359

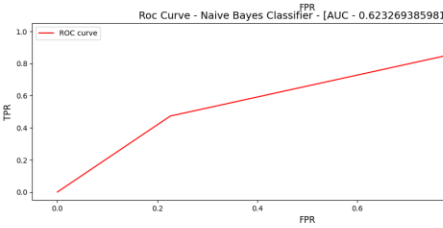

Random Forest Classifier Confusion Matrix
172
5
145
37
0
1
True label
0
1
Predicted label

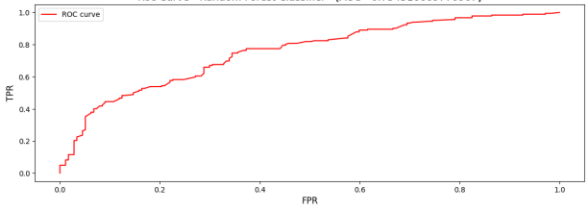
ROC Curves


Roc Curve - Logistic Regression - [AUC - 0.7943751164090147]
ROC curve
TPR
FPR


Roc Curve - K Nearest Neighbor - [AUC - 0.5461910970385
ROC curve
TPR
FPR


Roc Curve - Support Vector Machine - [AUC - 0.794344074005091]
ROC curve
TPR
FPR

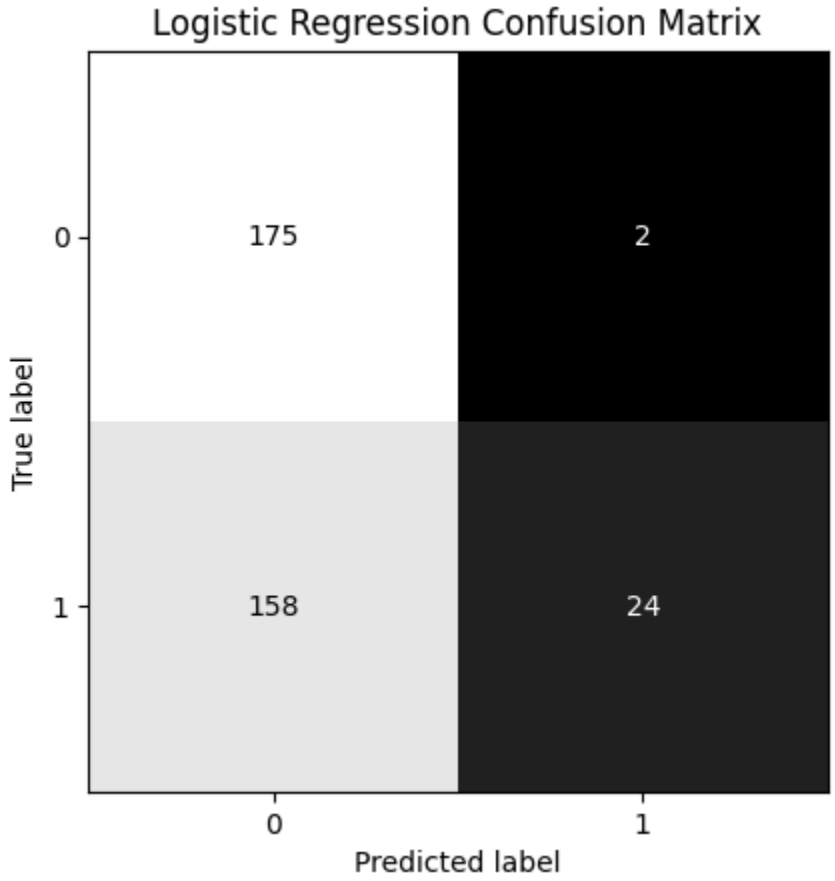

Roc Curve - Naive Bayes Classifier - [AUC - 0.623269385981
ROC curve
TPR
FPR


Roc Curve - Random Forest Classifier - [AUC - 0.754516669770907]
ROC curve
TPR
FPR

TF*IDF

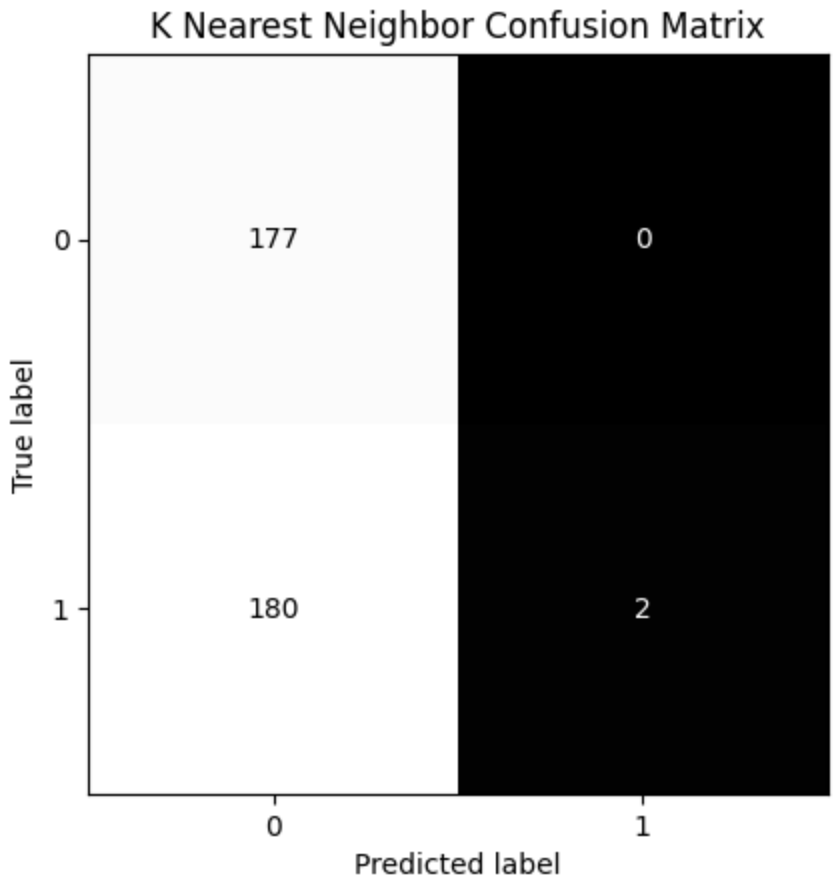
- *Logistic Regression*

	precision	recall	f1-score	support
0	0.53	0.99	0.69	177
1	0.92	0.13	0.23	182
accuracy			0.55	359
macro avg	0.72	0.56	0.46	359
weighted avg	0.73	0.55	0.46	359



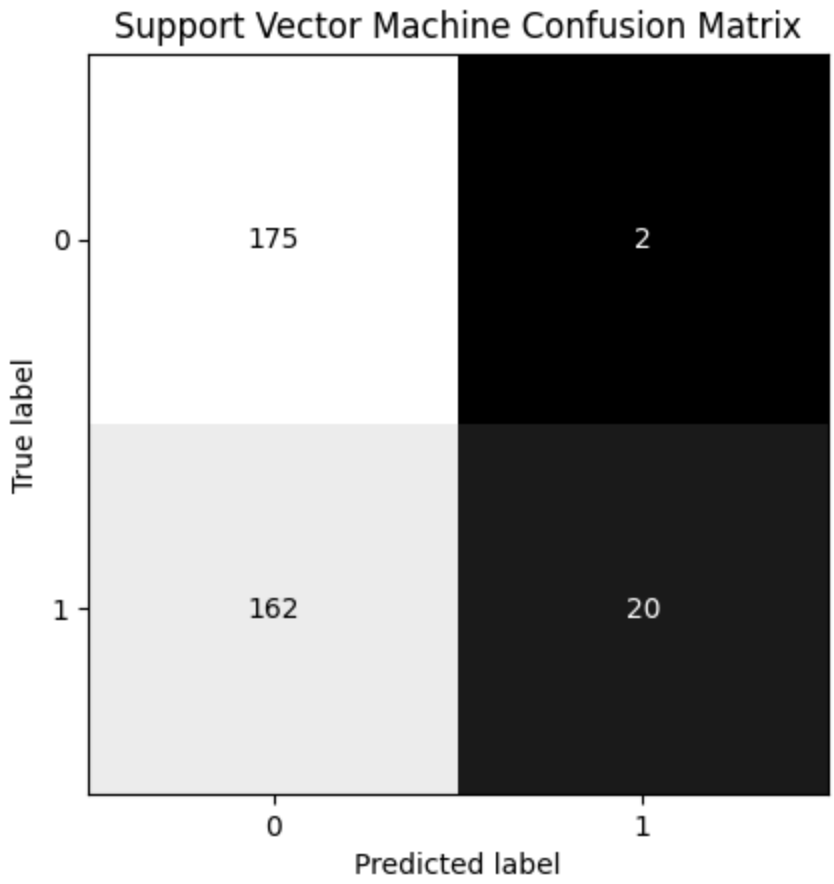
- *K Nearest Confusion Matrix*

	precision	recall	f1-score	support
0	0.50	1.00	0.66	177
1	1.00	0.01	0.02	182
accuracy			0.50	359
macro avg	0.75	0.51	0.34	359
weighted avg	0.75	0.50	0.34	359



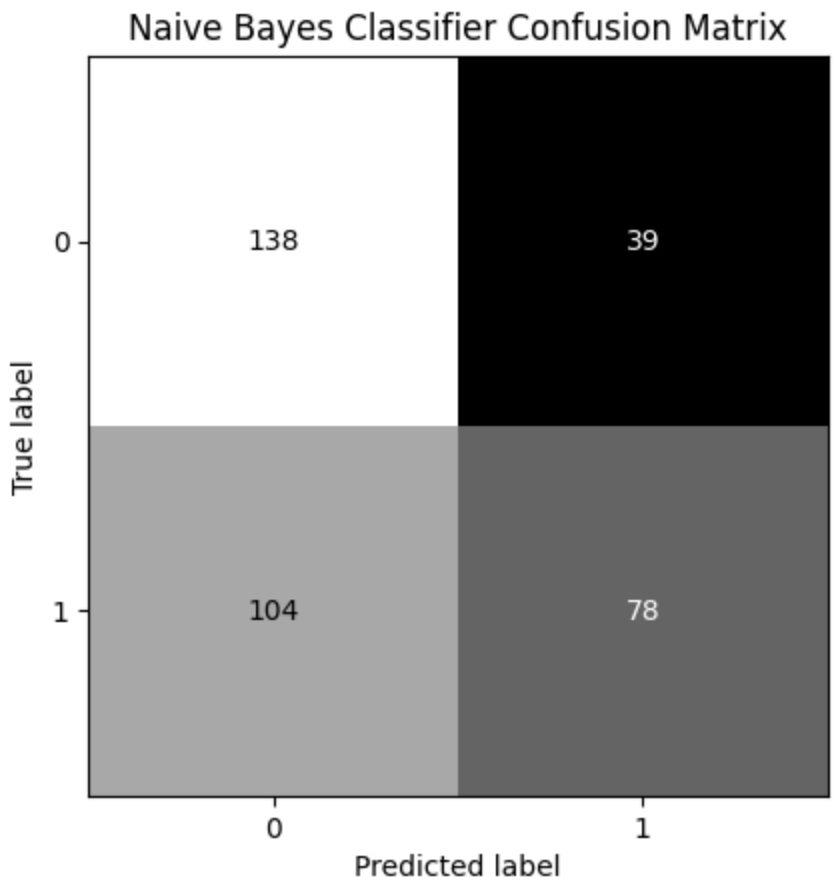
- *Support Vector Machine*

	precision	recall	f1-score	support
0	0.52	0.99	0.68	177
1	0.91	0.11	0.20	182
accuracy			0.54	359
macro avg	0.71	0.55	0.44	359
weighted avg	0.72	0.54	0.44	359



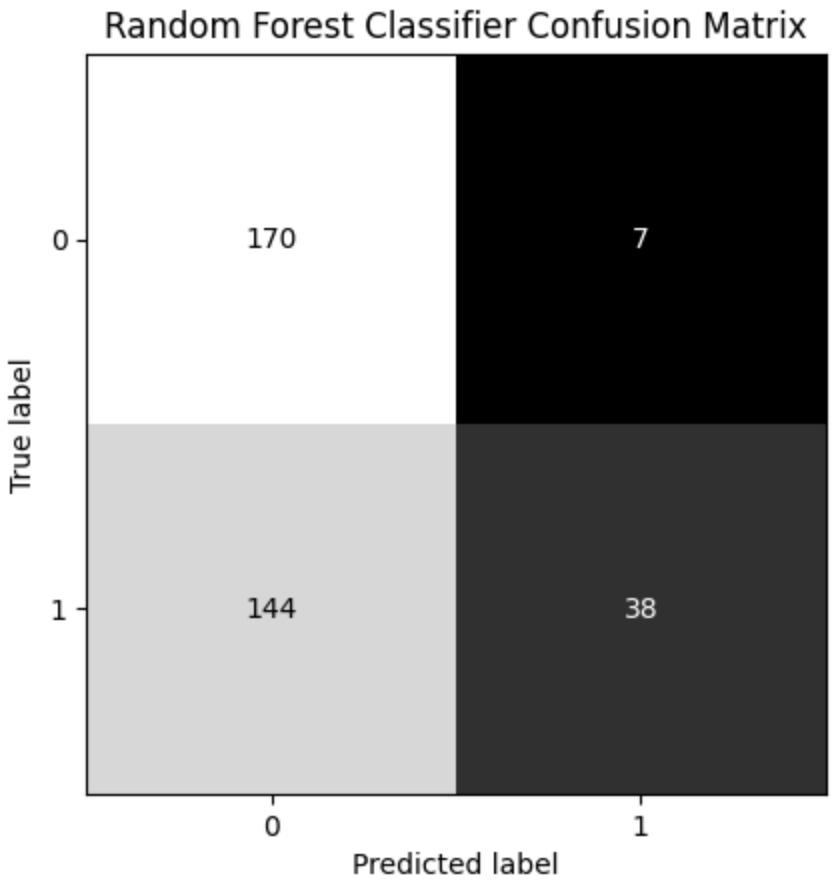
- *Naive Bayes Classifier*

	precision	recall	f1-score	support
0	0.57	0.78	0.66	177
1	0.67	0.43	0.52	182
accuracy			0.60	359
macro avg	0.62	0.60	0.59	359
weighted avg	0.62	0.60	0.59	359

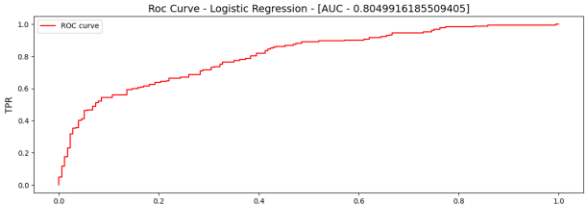


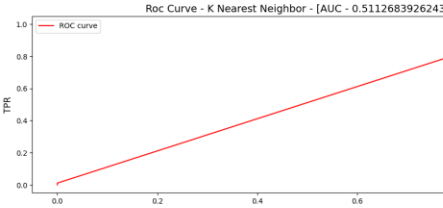
- *Random Forest Classifier*

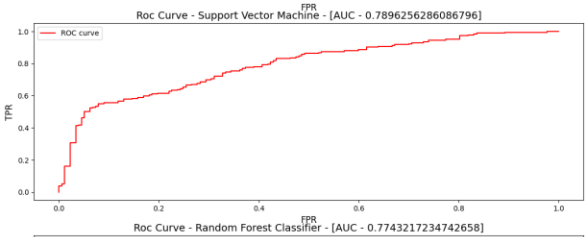
	precision	recall	f1-score	support
0	0.54	0.96	0.69	177
1	0.84	0.21	0.33	182
accuracy			0.58	359
macro avg	0.69	0.58	0.51	359
weighted avg	0.70	0.58	0.51	359

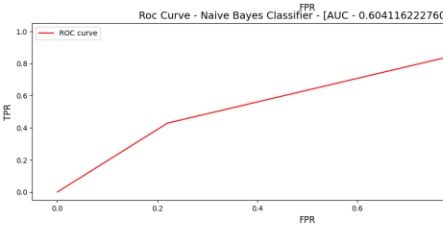

Random Forest Classifier Confusion Matrix
True label
Predicted label
0
1
170
7
144
38
0
1

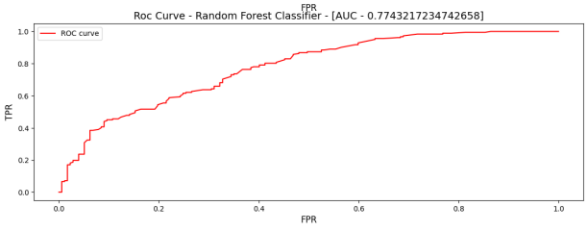
ROC Curves


Roc Curve - Logistic Regression - [AUC - 0.8049916185509405]
ROC curve
TPR
FPR


Roc Curve - K Nearest Neighbor - [AUC - 0.5112683926243
ROC curve
TPR
FPR


Roc Curve - Support Vector Machine - [AUC - 0.7896256286086796]
ROC curve
TPR
FPR


Roc Curve - Naive Bayes Classifier - [AUC - 0.604116222760
ROC curve
TPR
FPR


Roc Curve - Random Forest Classifier - [AUC - 0.7743217234742658]
ROC curve
TPR
FPR

4. Discussion

- *For the BOW and TF*IDF data, the SVM model was the one with the highest recall for finding negative text. The logistic regression model had the highest precision for positive textual data. The rain forest and naïve data both performed similarly in terms of their accuracy, f1, precision and recall. The k nearest didn't do so well compared to the rest of the models used. Overall, I believe the model did pretty well for the amount of data given to it. If it were to be given all million textual data lines as input then I believe that it would preform exceptionally well.*

- *One challenge I met was that the program was running very slow when the program began running the training models. This was because the training csv data had over a million lines of textual data, which is the most I have ever worked with. Due to this issue, I resorted to having the program randomly select 5000 lines of data to use by using the .sample() function on my training data frame.*

- *This assignment was a very good learning experience. I got to understand what natural learning processing (NLP) is, how it is used, and its importance in the real world. It also helped me understand how to capture textual data and how to clean it up in order for the machine to better understand it. I also learned how to use feature extractions so that the machine can better understand the depth of the language it receives. The information I learned in this project will definitely help me with the final project and how my group and I will go about cleaning and processing textual styled data.*

5. Appendix

- *https://github.com/hadiplays/Assignment-2*