Project 1: Email Spam Detection

The Team

Angel Argueta

Creston Altiere

Sahar Sheikholeslami

Wessley Dennis

Problem Definition (5 points)

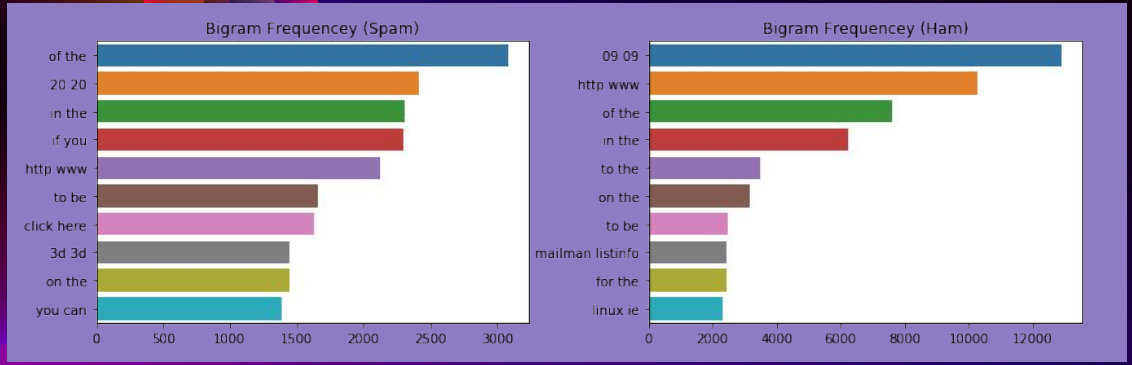
- Goals:
  - Leverage contemporary Natural Language Processing techniques to construct and train a Spam Email detector
- Problem Relevance:
  - Whether or not an email is spam can depend largely on:
    - If you're using a work account
    - if the email is a reply to a previous email
    - if the user is interested in a particular product
  - NLP and machine learning allows
    - Identification of emails likely to be spam
    - Pretraining of model that can be employed out of the box

Data Collection (5 points):

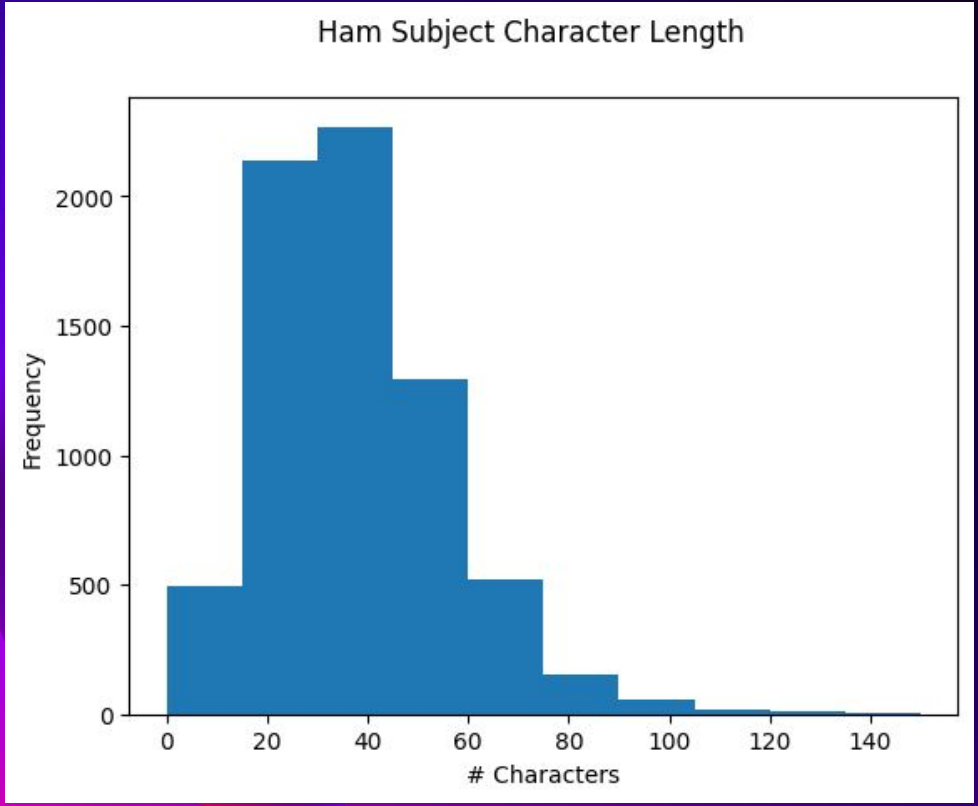
- Dataset: SpamAssassin public mail corpus from 2004 (9351 emails).
  - Advantages
    - Large popular dataset utilized to train some of the earliest spam detectors
      - Widely used
    - Contains ham and Spam emails of varying difficulty
    - Contains 9351 emails
  - Disadvantages
    - It's likely outdated - (Updated as of 2004)
    - data is asymmetrical (only 2399 Spam vs 6952 Hams)
  - Overall outlook
    - We may have a chance to evaluate how Spam has improved in it's solicitation methods between 2004 and today

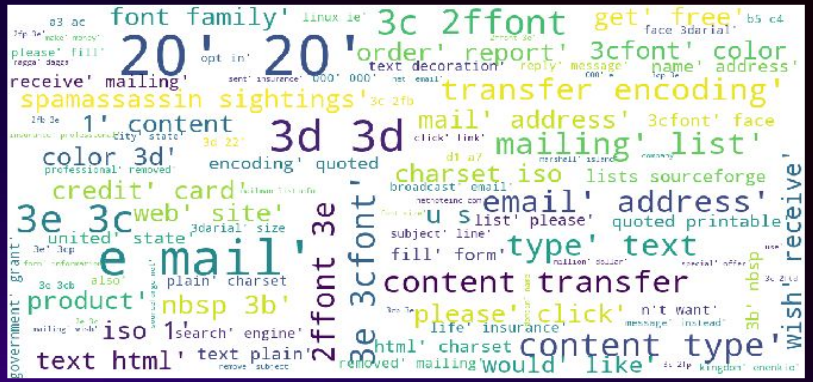
Data Analysis (5 points):

- Dataset - First look:
  - Spam
    - Phrase “Click Here”, seems to be more popular in spam emails
    - larger significantly larger amount of second hand reference (greater appearance of the word “you” as if requiring something of the user)
  - Ham
    - More specific subject related terms
    - Slightly greater amount of longer words
- The Above findings might help give insight into understanding our data, and what areas we might be able to effectively employ it in training.



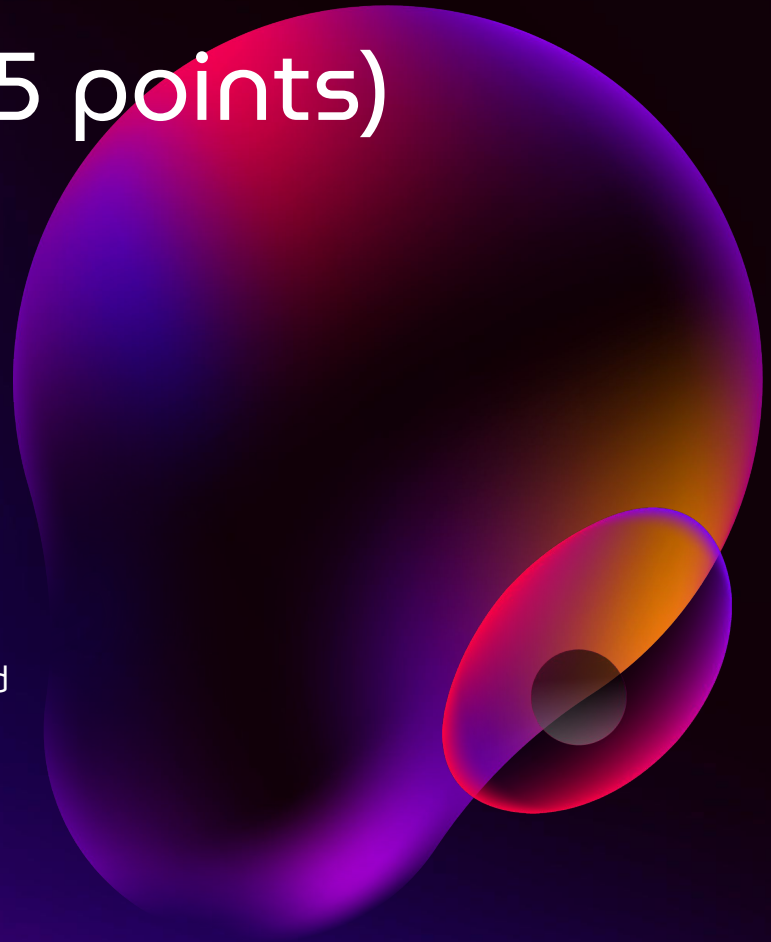
Data Analysis (5 points):





Data Preprocessing (5 points)

- Removal of:
  - Stopwords
  - Duplicates
  - Punctuations (Non-alphabet Characters)
- Tokenization
- Missing values investigated
- Duplicate values removed
- Outliers Detection
  - Feature Correlation Studied
  - Bar plots were studied
  - Count of words in Spam vs. Ham were studied
- Normalization:
  - Stemming
  - Lemmatization



Feature Engineering (5 points):

The following tasks were performed and a pipeline to be fed to our models was created

- Vectorization
  - Label Encoding
  - Word embeddings
  - Count (BoW)
  - TF-IDF
- Punctuation Count Ham vs. Sam
  - Bar plots were studied to understand possible correlation
  - Correlations were studied to understand possible correlation
- Email Text Length Ham vs. Sam
  - Bar plots were studied to understand possible correlation
  - correlation were studied to understand possible correlation

Model Selection and Architecture (10 points):

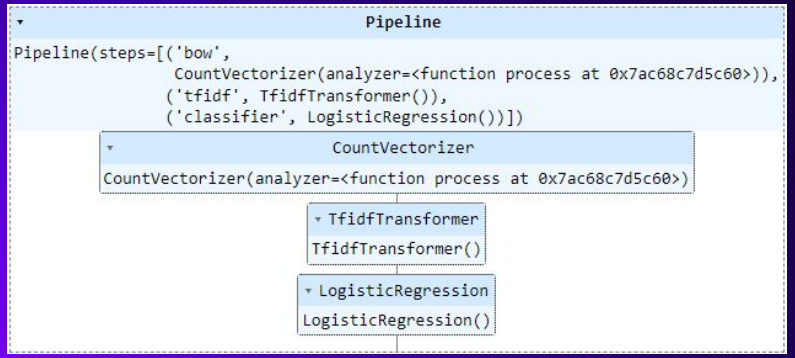
- We experimented with a mixture of 6 different model architects listed below. Model Architects are shown in form of a diagram in the next slide
    - Deep Neural Network
        - Best model in terms of accuracy score. This was to be expected given the complex nature of spam emails. This is our chosen model based on accuracy scores
    - Naive Bayes
        - This was our least accurate model. We believe the model performed poorly because the data is too complex to be able to employ a simple architect such as NB
    - SVM Accuracy
        - This was a tie for our second best model. By using sigmoid kernel trick we were able able to capture details of text data with this mode.

Model Selection and Architecture (10 points):

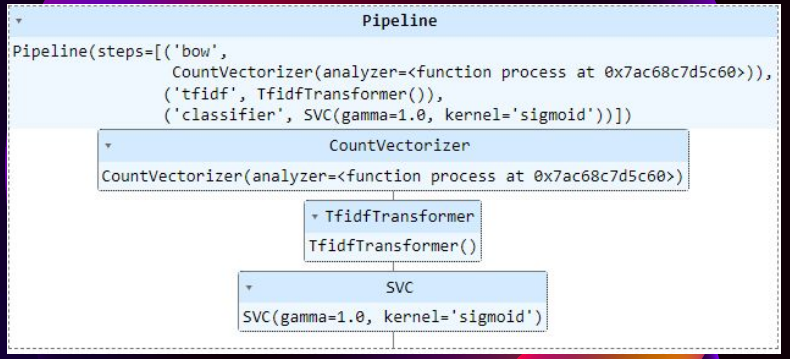
- KNN
  - This was our second worst performing model. We believe the metric of “nearest neighbor” in terms of L2 norm distance of words with each other may not be a good representation of the distance between spam and ham emails. Because words maybe close enough but the context and semantic may not be. We believe this was the reason for this model’s poor performance
- Logistic Regression
  - This was our second tie for second best performing model. This method is statistical and commonly used for binary classification problems. Spam and Ham are a perfect example of binary problems and the model performed very well
- Random Forest
  - Random Forest is an ensemble learning algorithm and ruled based. Much like KNN, it may not have the necessary robustness to understand the intricacies of context of a text that represents Spam vs. Ham

Model Selection and Architecture

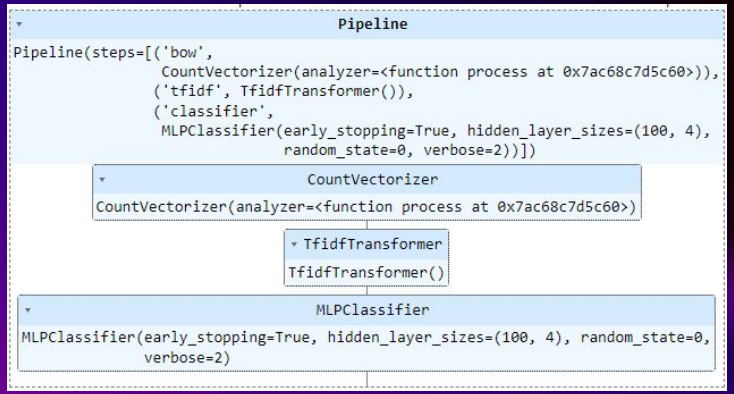
- Logistic Regression



- SVM

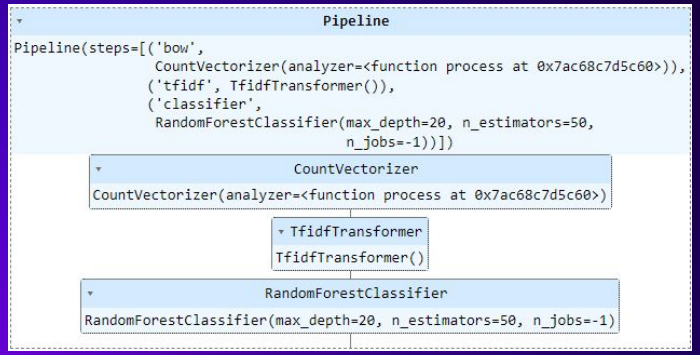


- Deep Neural Network

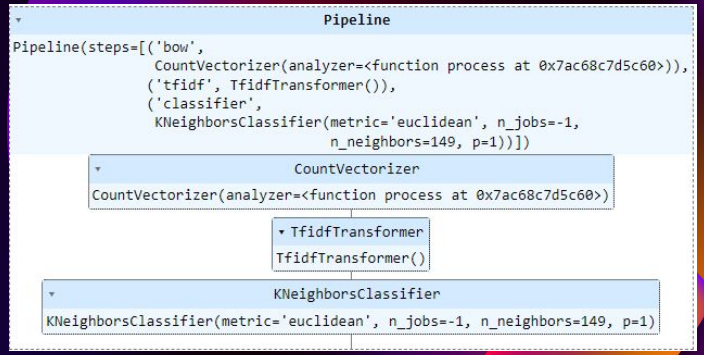


Model Selection and Architecture (10 points):

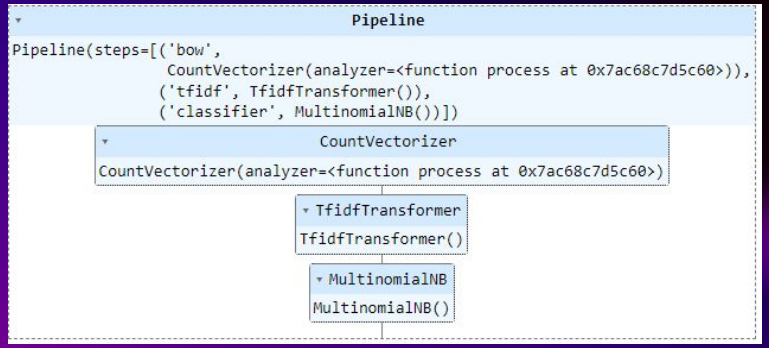
- Random Forest



- KNN



- Naive Bayes



Training and Evaluation (10 points):

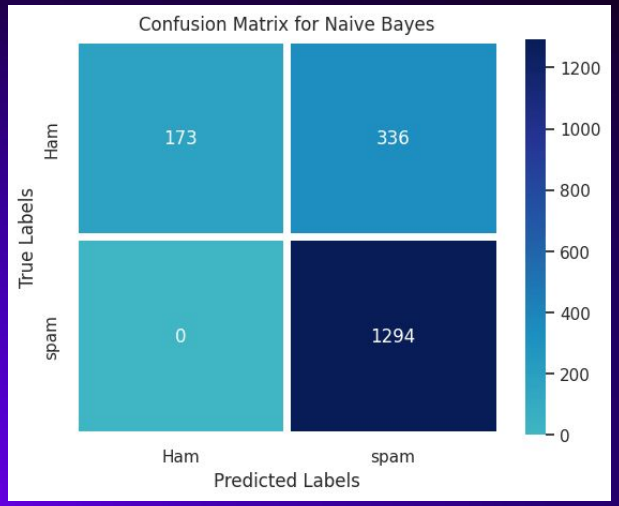
- Deep Neural Network - experimented with a grid of hyper parameters :
  - hidden_layer_sizes, activation=relu, tanh, logistic, solver=adam, sgd, alpha=0.01, 0.001, and 0.0001, verbose=True, momentum=0.1, 0.5, 0.9, early_stopping=True, validation_fraction=0.1
- Naive Bayes - experimented with a grid of hyper parameters :
  - alpha = 0.001, 0.01,0.1
- SVM - experimented with a grid of hyper parameters :
  - C=1.0, 0.5, 0.6, kernel=sig, rbf, degree=1,3,5 , gamma=1,auto

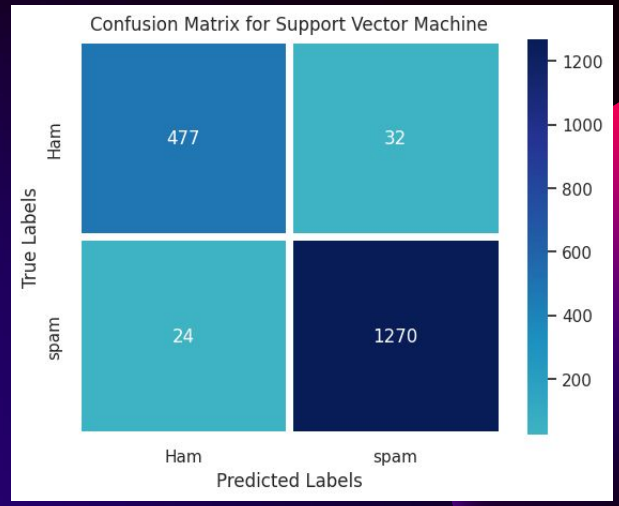
Training and Evaluation (10 points):

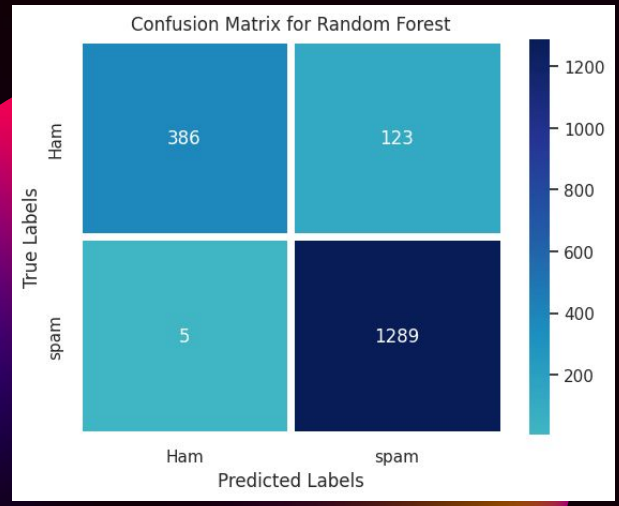
- KNN - experimented with a grid of hyper parameters :
  - n_neighbors=5, 20, 100, 149, leaf_size=10, 20, 30, metric='minkowski', Euclidean, n_jobs=None, -1
- Logistic Regression - experimented with a grid of hyper parameters :
  - penalty=l2,l1 *, C=0.5, 1.0, 5, solver=lbfgs,adam
- Random Forest - experimented with a grid of hyper parameters :
  - n_estimators=10,50, 100, criterion=gini, max_depth=10,20,50,100, min_impurity_decrease=0.0, bootstrap=True, n_jobs=-1

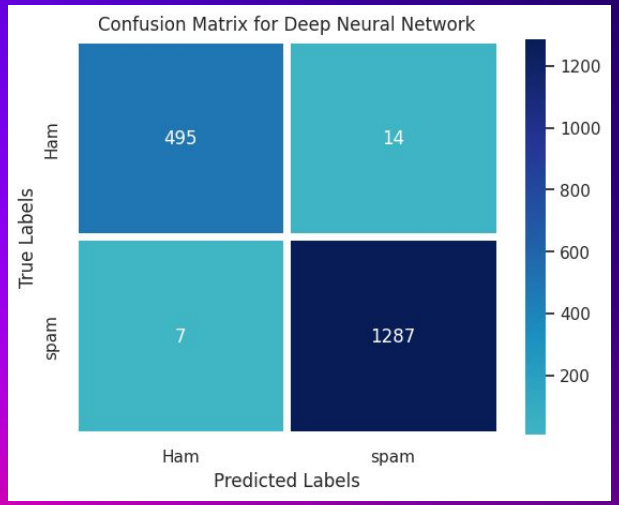
Results and Discussion during Presentation (5 points):

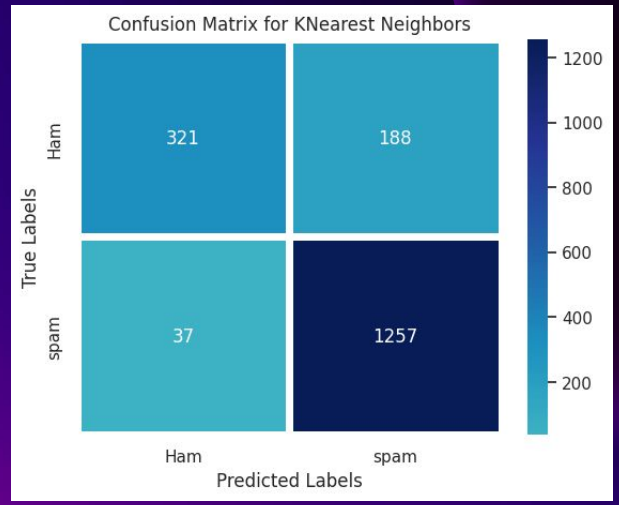
- To analyze our models, we looked at several measurements:
  - The Confusion Matrix for each model
  - Overall accuracy
    - Percentage of predictions that were correct
    - Not completely indicative of performance due to disparity in category sizes
  - Precision, Recall, and F1 for Ham and Spam
    - Precision
      - Percentage of positive predictions which were true
    - Recall
      - Percentage of Ham or Spam samples correctly labelled
    - F1 Score
      - Harmonic mean of precision and recall

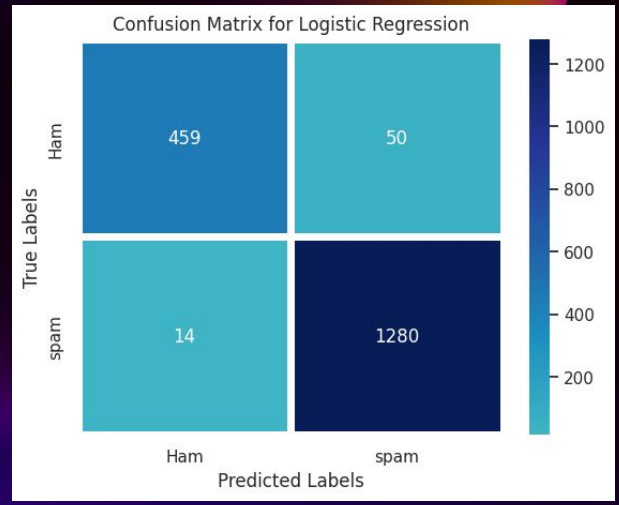

Confusion Matrix for Naive Bayes
True Labels
Ham
spam
173
336
0
1294
Ham
spam
Predicted Labels
1200
1000
800
600
400
200
0

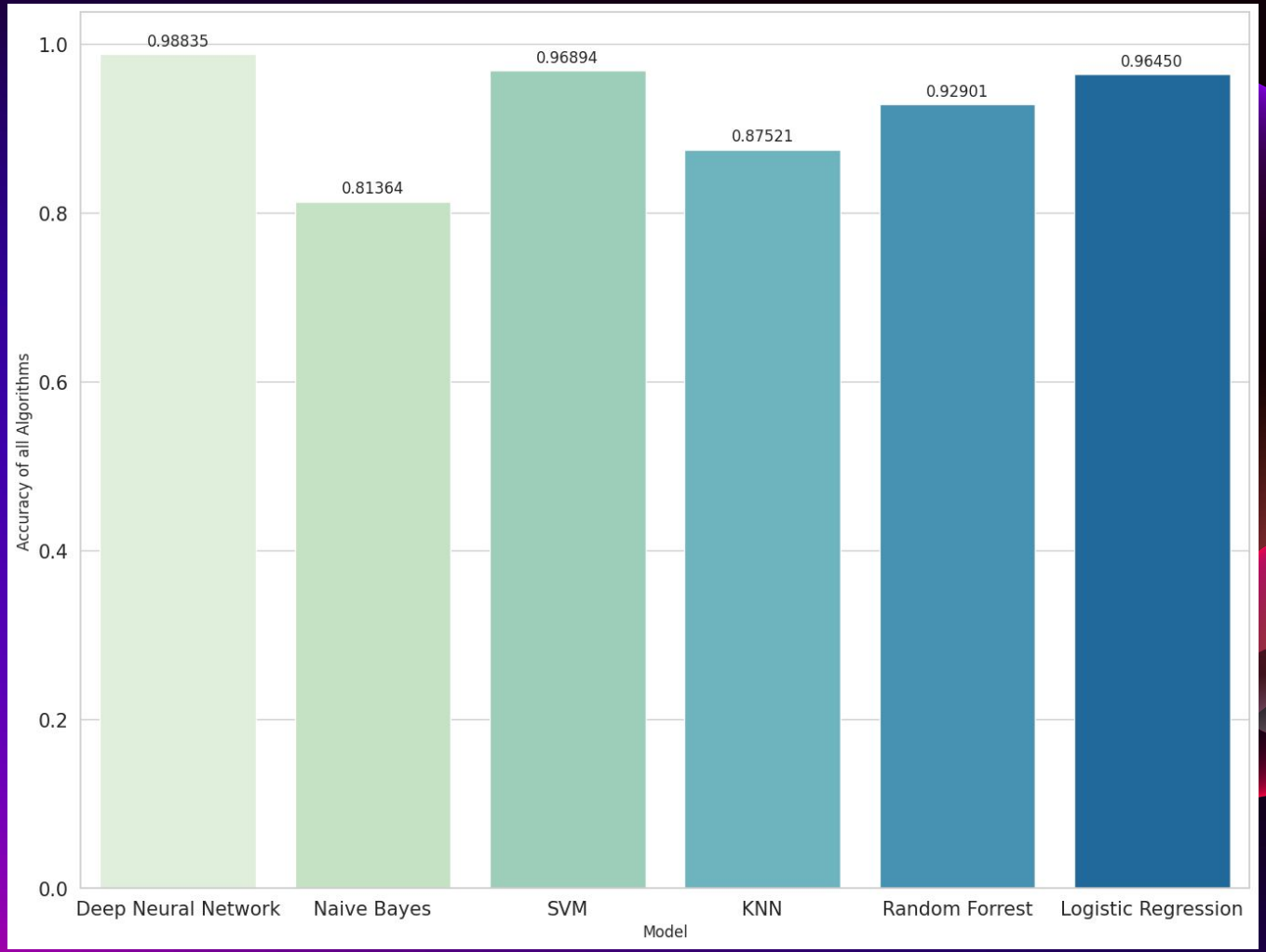

Confusion Matrix for Support Vector Machine
True Labels
Ham
spam
477
32
24
1270
Ham
spam
Predicted Labels
1200
1000
800
600
400
200

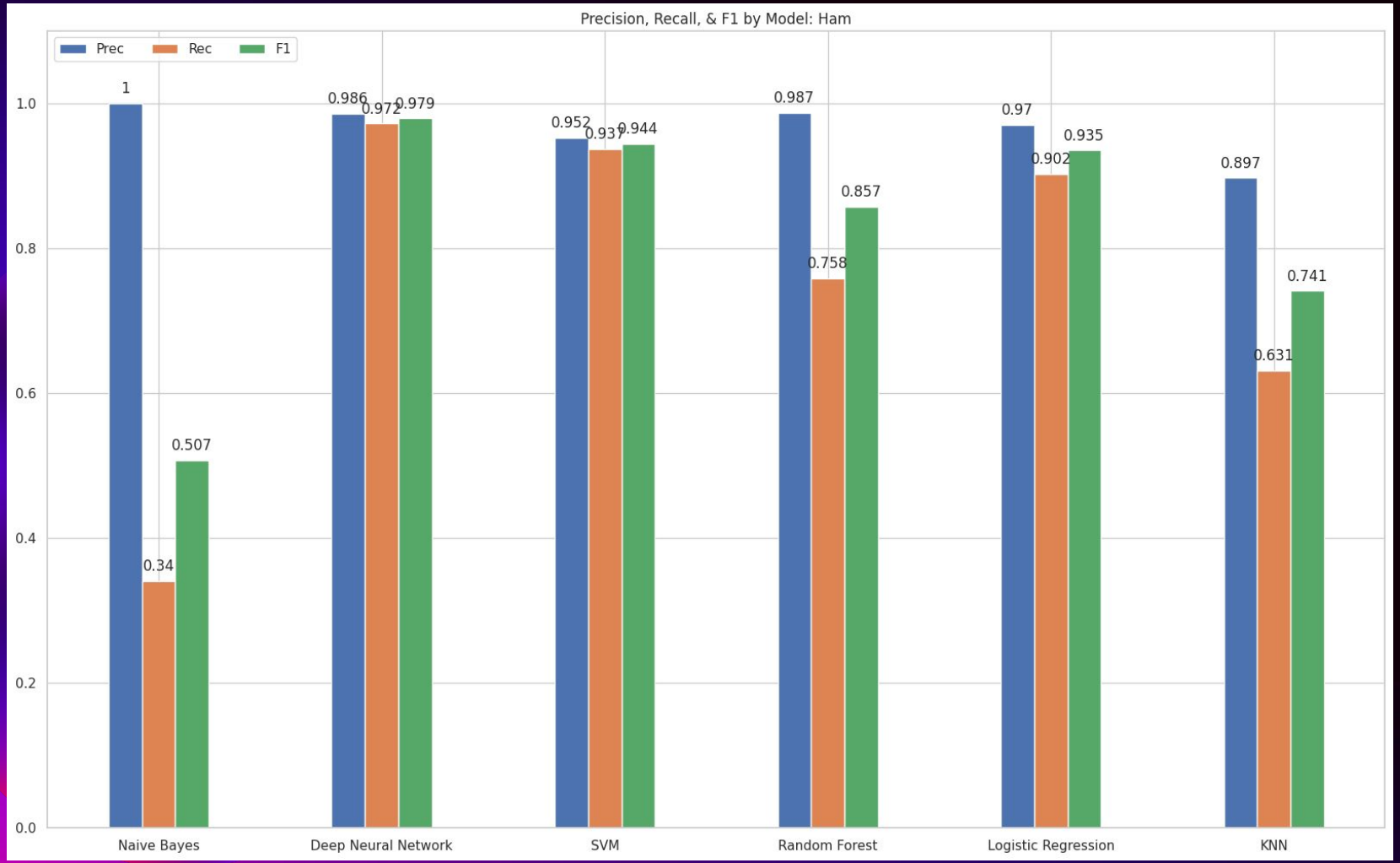

Confusion Matrix for Random Forest
True Labels
Ham
spam
386
123
5
1289
Ham
spam
Predicted Labels
1200
1000
800
600
400
200

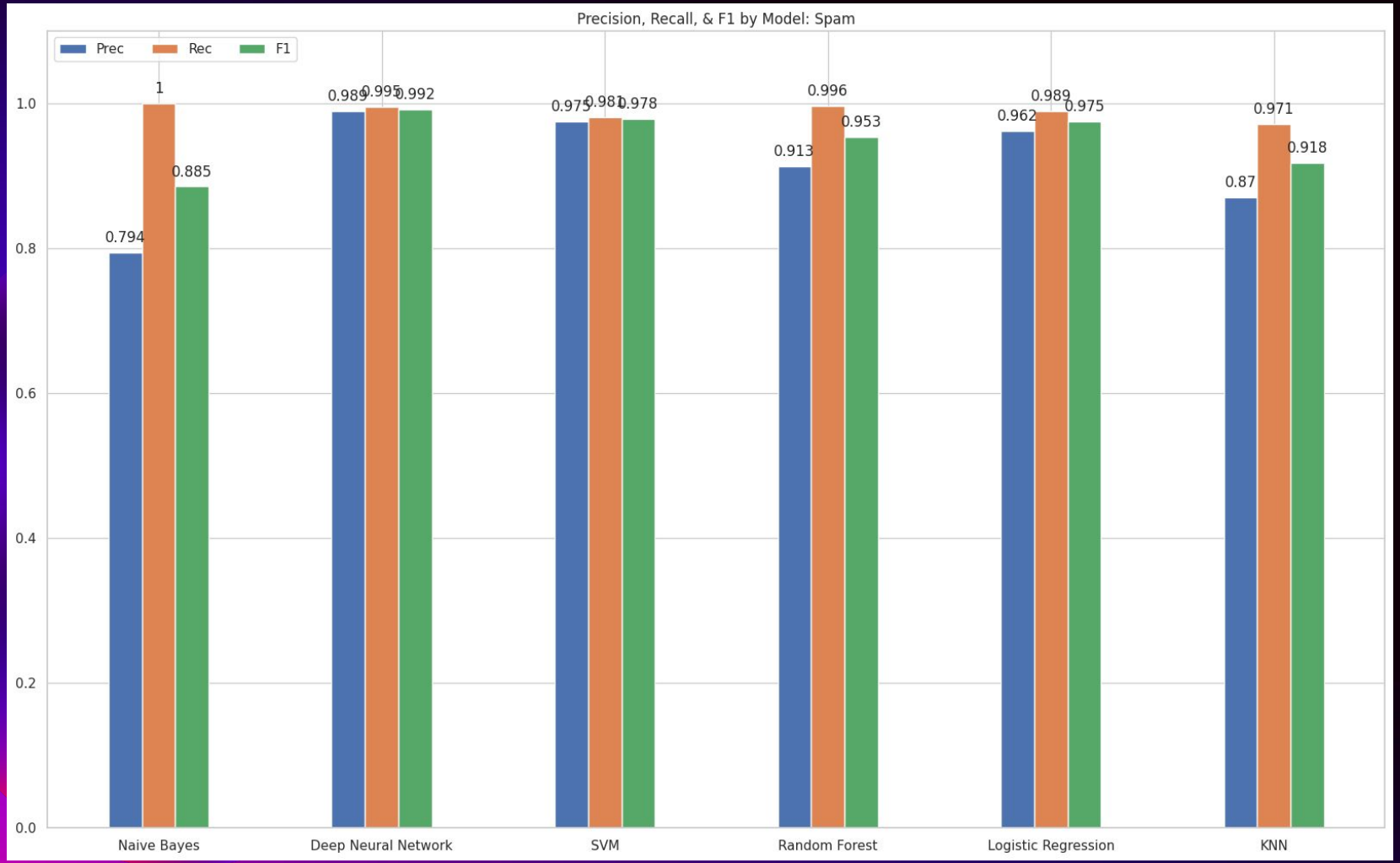

Confusion Matrix for Deep Neural Network
True Labels
Ham
spam
495
14
7
1287
Ham
spam
Predicted Labels
1200
1000
800
600
400
200


Confusion Matrix for KNearest Neighbors
True Labels
Ham
spam
321
188
37
1257
Ham
spam
Predicted Labels
1200
1000
800
600
400
200

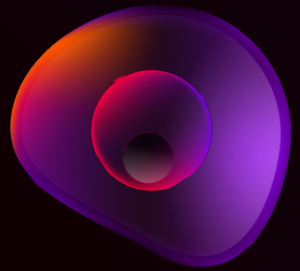

Confusion Matrix for Logistic Regression
True Labels
Ham
spam
459
50
14
1280
Ham
spam
Predicted Labels
1200
1000
800
600
400
200


1.0
0.8
0.6
0.4
0.2
0.0
Accuracy of all Algorithms
0.98835
0.81364
0.96894
0.87521
0.92901
0.96450
Deep Neural Network
Naive Bayes
SVM
KNN
Random Forrest
Logistic Regression
Model


Precision, Recall, & F1 by Model: Ham
Prec
Rec
F1
1.0
0.8
0.6
0.4
0.2
0.0
1
0.34
0.507
0.986
0.972
0.979
0.952
0.937
0.944
0.987
0.758
0.857
0.97
0.902
0.935
0.897
0.631
0.741
Naive Bayes
Deep Neural Network
SVM
Random Forest
Logistic Regression
KNN


Precision, Recall, & F1 by Model: Spam
Prec
Rec
F1
1.0
0.8
0.6
0.4
0.2
0.0
0.794
1
0.885
0.989
0.995
0.992
0.975
0.981
0.978
0.913
0.996
0.953
0.962
0.989
0.975
0.87
0.971
0.918
Naive Bayes
Deep Neural Network
SVM
Random Forest
Logistic Regression
KNN

Github Link



- https://github.com/NLP-Projects-CAP6640/Porject1/tree/main