# Comparison of Machine Learning for Spam – Final Results

Ethan Massingill




# Introduction

- With the growth of spam and the use of internet communications it is essential to maintain proper filters
- Emerging Machine Learning can be utilized to perhaps better filter spam as hardware increases.

# Value

- Provides technical coding for models using Python Libraries data cleaning experience
- ELM is a newer front-loaded neural network
    - Provides a unique coding opportunity  to understand how it can be utilized for spam detection
- Allows for deeper understanding of Machine Learning for security and Language processing

# Literature Review – Research Papers

- Going over previous research in the field for spam detection provided insight into the process to conduct the research
- Provided insight into how the algorithms determine if it is spam or not along with the best processes to tune them.
- Showed promising results for both model types under large datasets.

# Methodology

- Process the data to prepare it for testing and training.
- Write out the models in python.
- Extract the model's metrics and compare them.

# Algorithm Refresher

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

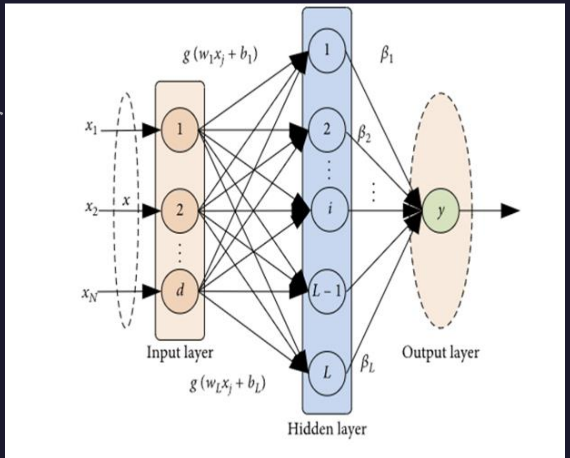
$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

## EXTREME LEARNING MACHINE

- Feedforward neural network
    - Single hidden layer of neurons
- Weights connecting input layer to hidden layer are random and fixed
- Weights connecting hidden to output layer are learned through linear regression
    - Activation function is applied to weighted sum (sigmoid)
- Output layer determines if spam or not spam
    - Single pass learning
- Once trained it can be used to make predictions on new messages. Input vector is fed through the hidden layer, and the output is predicted using learned weight vector

## NAÏVE BAYES

- Based on Bayes Theorem
    - Probability of an event happening based on prior knowledge of conditions that might be related
- Calculated based on the probability of each word in the message occurring in spam and non-spam emails in a training dataset.
- The output of the Naive Bayes model is a binary prediction of whether an email message is spam or not.
- Probability of specific words used in spam.
- Assumes words are independent of each other
- Single pass probability learning



# Implementation

- Python 3.9
  - PyCharm IDE
  - Scikit-learn
  - Pandas
  - Numpy
  - Matplotlib
  - Seaborn
  - Time

- Hardware
  - Ryzen 7 3750Hz 4 core processor
  - NVIDIA GeForce GTX 1660 Ti with Max-Q
  - 16 gig ddr4 ram
- Data
  - Spam Assassin Corpus
    - Greater than 5000 emails
    - Processed data before running into 3 columns
      - 0 or 1 spam/not spam labels
      - Email Body
      - Email number (Label Key)

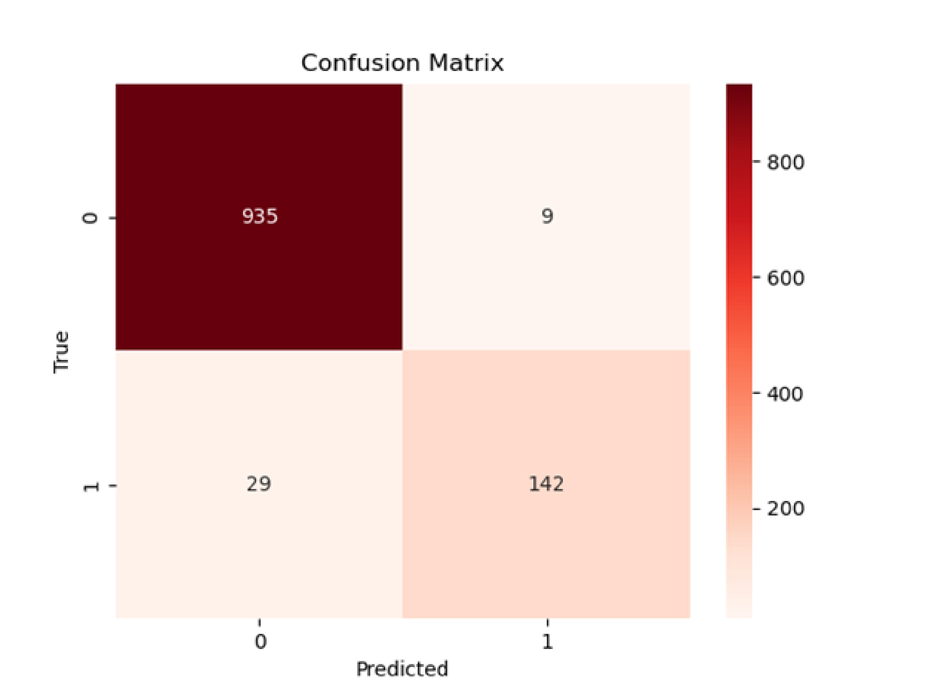
# Demo Code

```
#this is the set-up to define the neural network itself. The input size is the s
#The hidden_size is the number of hidden neurons in the dense single layer
#the input weights are randomly determined based on the hidden size and input si
#the biases are also randomly picked using np as defined by an elm
input_size = X_train_vec.shape[1]
hidden_size = 1000
input_weights = np.random.normal(size=[input_size,hidden_size])
biases = np.random.normal(size=[hidden_size])
# Activation function. This is using sigmoid but can be change if needed

Ethan019
def sigmoid(x):
    return 1/(1+ np.exp(-x))

# Applies the hidden nodes with the input weights, input and dot product.
#then applies the biases
#finally runs through sigmoid activation

#******this is the only learning phase for the model******
Ethan019
def hidden_nodes(X):
    G = np.dot(X, input_weights)
    G = G + biases
    H = sigmoid(G)
    return H
```

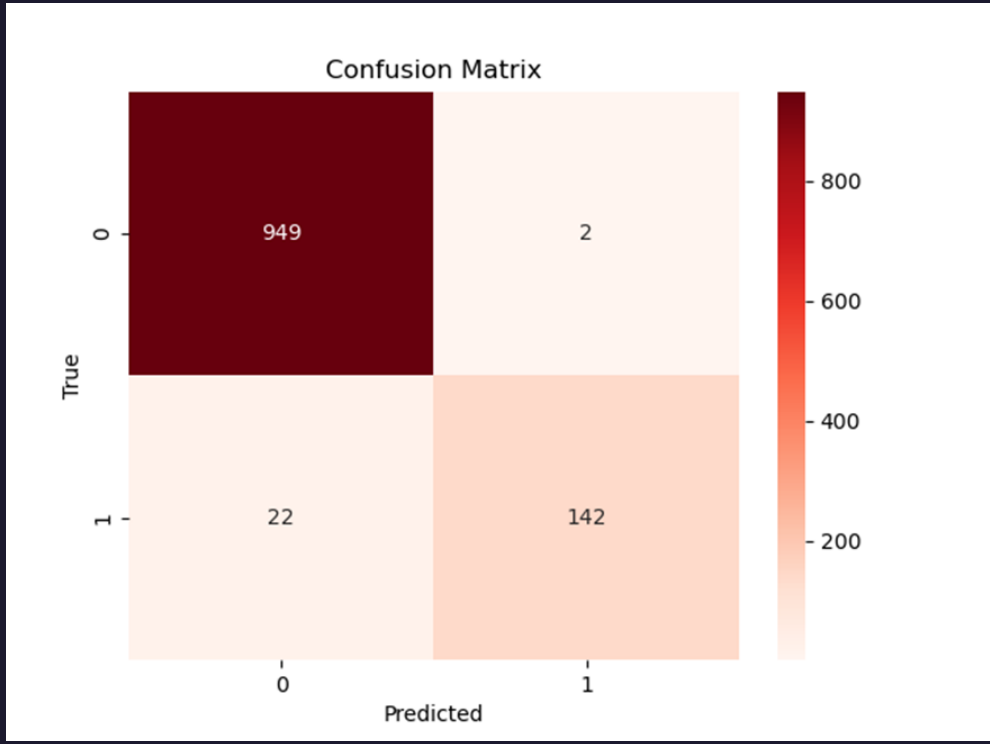
```
ELM Model Results:
*********************
Time running: 1.057401180267334 seconds
Number of nodes used: 1000
*********************
Number of spam emails:  747
Number of non-spam emails:  4825
*********************
Metrics
Precision: [0.96991701 0.94039735]
Recall: [0.9904661  0.83040936]
F1 Score: [0.98008386 0.88198758]
Accuracy: 96.59192825112108
```

# Initial Results

## ELM




## NAÏVE BAYES

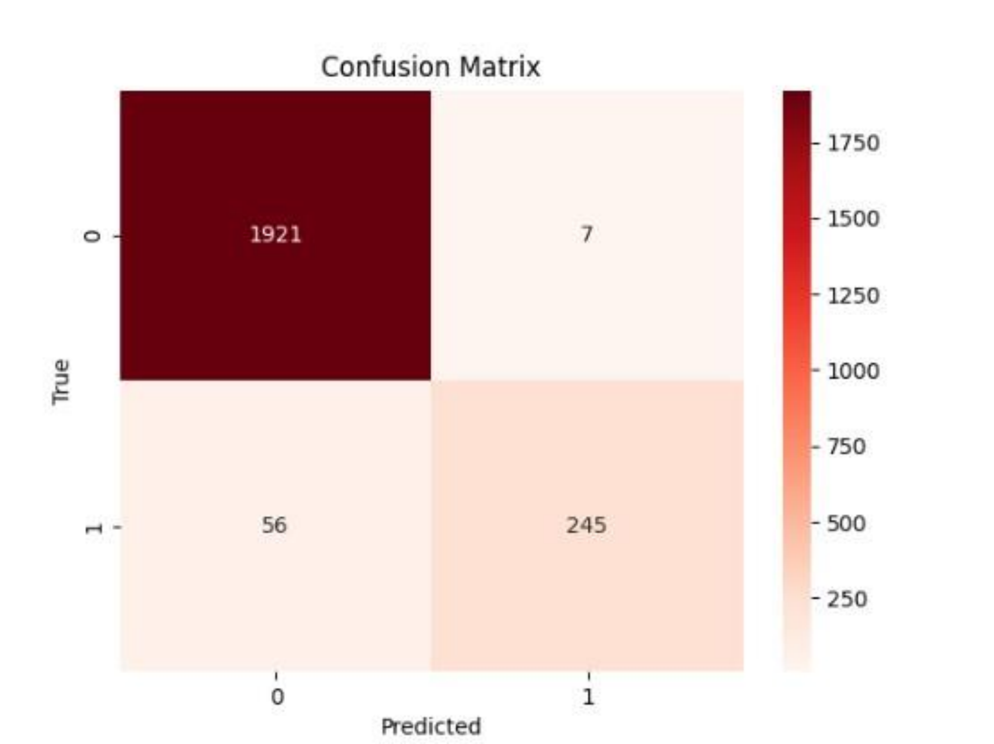


# Test Modifications

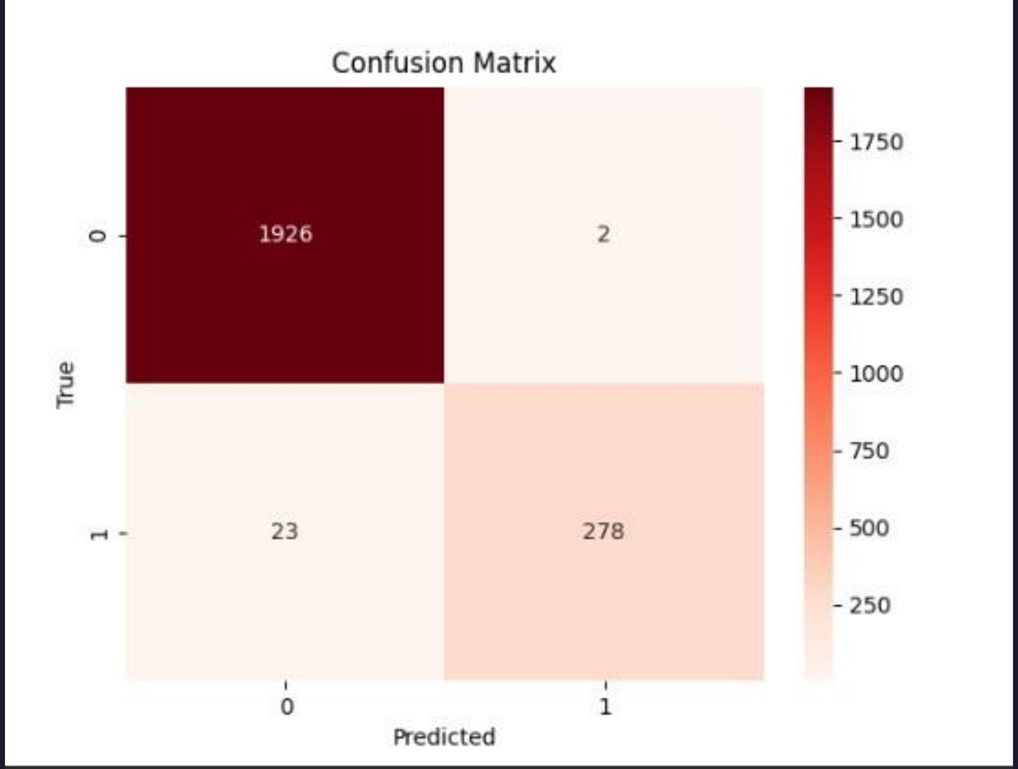
- Increased nodes for hidden layer in ELM from 1000 to 1500
- Combined ELM and NB models into one file for unified running environment
- Added data visualization to metrics
- Doubled the test set from 20% to 40%

# Final Confusion Matrix Results

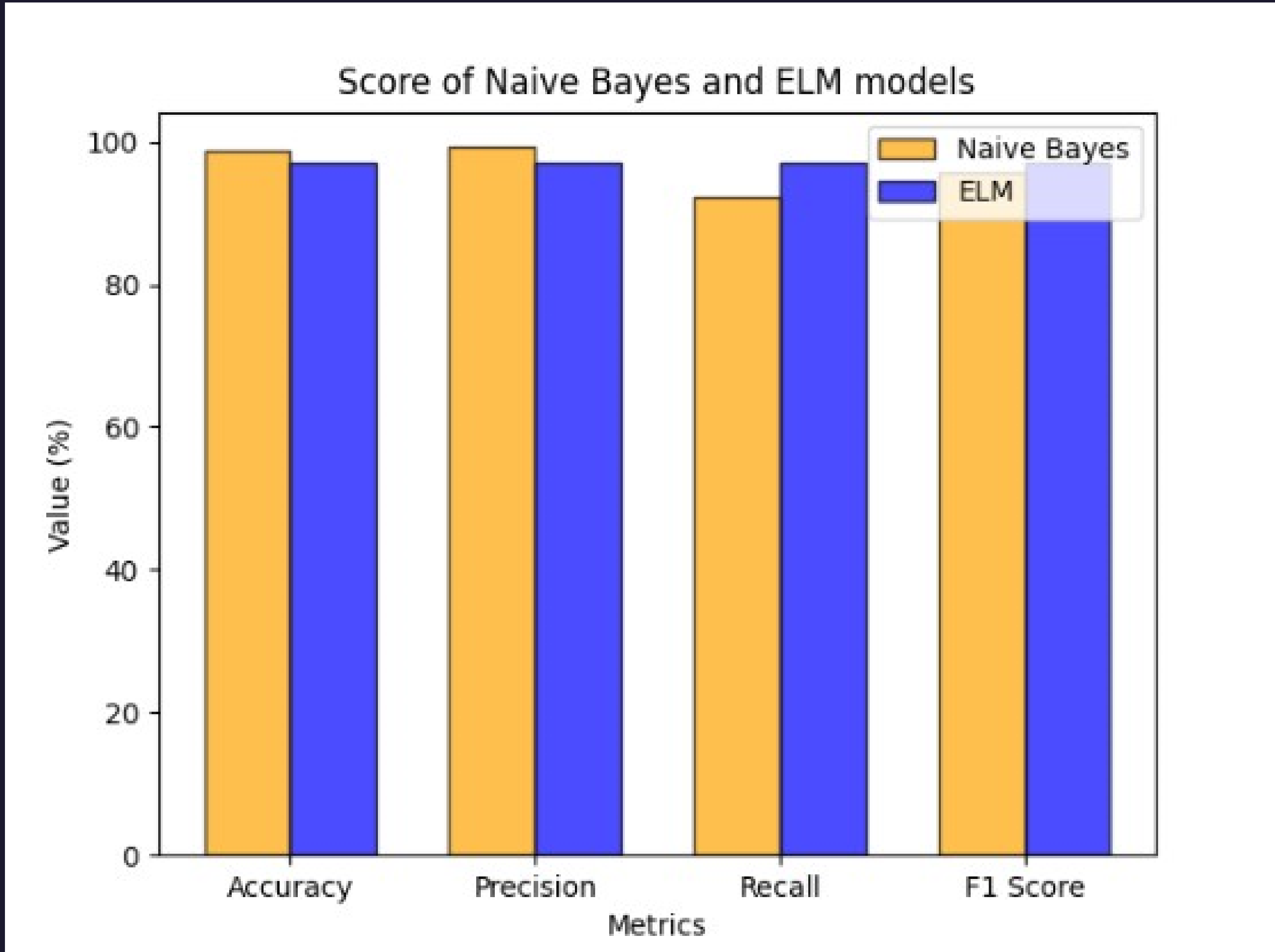
## ELM RESULTS - MATRIX



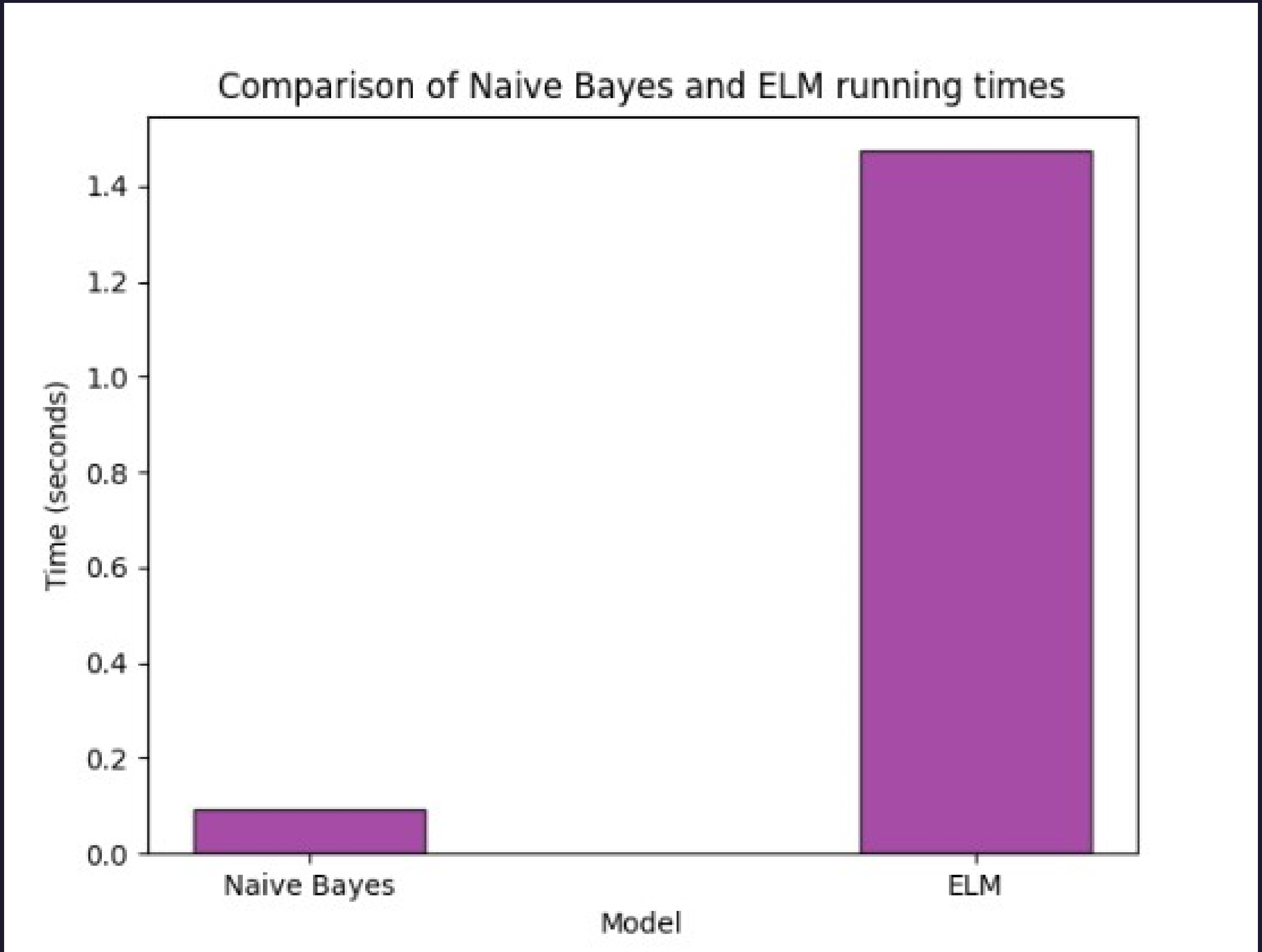

## NB RESULTS - MATRIX



# Final Accuracy Comparison



# Final Time Comparison



# Summary Discussion

Naïve Bayes and ELM both show similar scores for accuracy of single pass models even after increasing the testing size. However, NB shows a drastic reduction in time for the model to correctly pass through the data set. This could be due to hardware of the machine; however, it seems to confirm other published research that shows NB out preforms complicated models. This is likely due to it being a statistical model instead of having to pass through neurons in a hidden layer. For implementation, these factors should be considered when choosing which model is suitable for live production.

# References


1. Wang, Jian, et al. "A Review on Extreme Learning Machine - Multimedia Tools and Applications." SpringerLink, Springer US, 22 May 2021, link.springer.com/article/10.1007/s11042-021-11007-7#Sec45.
2. Webb, Geoffrey I., Eamonn Keogh, and Risto Miikkulainen. "Naïve Bayes." Encyclopedia of machine learning 15 (2010): 713-714.
3. Rusland, Nurul Fitriah, et al. "IOPscience." *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 1 Aug. 2017, iopscience.iop.org/article/10.1088/1757-899X/226/1/012091.
4. Zhang, Zhijie, et al. "Directory of Open Access Journals." IEEE Access, IEEE, 1 Jan. 2020, doaj.org/article/4ca7c055a470444893367ec06baf02c3.
5. Kaya, Yilmaz, et al. *An Expert Spam Detection System Based on Extreme Learning Machine*. Computer Science - Research and Development, July 2014, www.researchgate.net/publication/268025901_An_Expert_Spam_Detection_System_Based_on_Extreme_Learning_Machine.
6. Eberhardt, Jeremy. "Bayesian Spam Detection ." *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*, University of Minnesota , Mar. 2015, digitalcommons.morris.umn.edu/cgi/viewcontent.cgi?article=1028&context=horizons.
7. Metsis, Vangelis, et al. "Spam Filtering with Naive Bayes – Which Naive Bayes?" *Conference: CEAS 2006 - The Third Conference on Email and Anti-Spam*, CEAS , Jan. 2006, www.researchgate.net/publication/221650814_Spam_Filtering_with_Naive_Bayes_-_Which_Naive_Bayes.
8. Dixon, s. "Global Average Daily Spam Volume 2021." *Statista*, Statista, 28 Apr. 2022, www.statista.com/statistics/1270424/daily-spam-volume-global/.
9. Erdem, K. E. (burnpiro). (2020, May 29). *Introduction to extreme learning machines*. Medium. Retrieved March 16, 2023, from https://towardsdatascience.com/introduction-to-extreme-learning-machines-c020020ff82b
10. C.D. Manning, P. Raghavan and H. Schütze (2008). Introduction to Information Retrieval. Cambridge University Press, pp. 234-265.