Artificial Intelligence Bootcamp

SMS Spam Detection: Machine Learning in Action

Nathan Hull

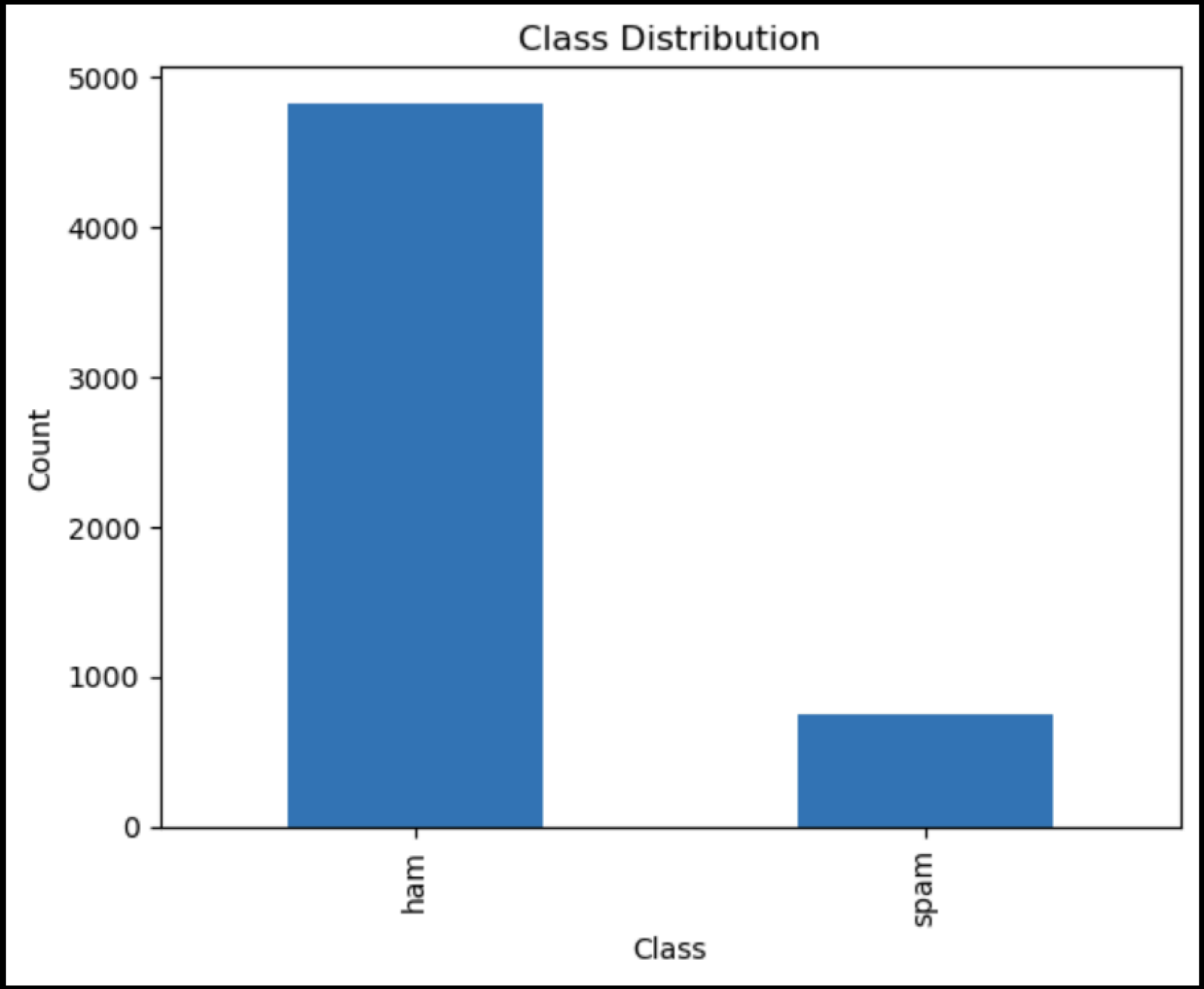
PROJECT OVERVIEW

- **Objective: Develop a model to classify SMS messages as spam or ham**
- **Dataset: SMS Spam Collection Dataset**
- **Approach: Data preprocessing, model training, and evaluation**

DATA EXPLORATION

- /uciml/sms-spam-collection-dataset
- 5,572 text samples
- 87% ham 13% spam

```
Class distribution:
label
ham     0.865937
spam    0.134063
Name: proportion, dtype: float64
```



TEXT PREPROCESSING

STEPS:

- **Lowercase conversion**
- **special character removal**
- **tokenization**
- **stop word removal**
- **lemmatization**

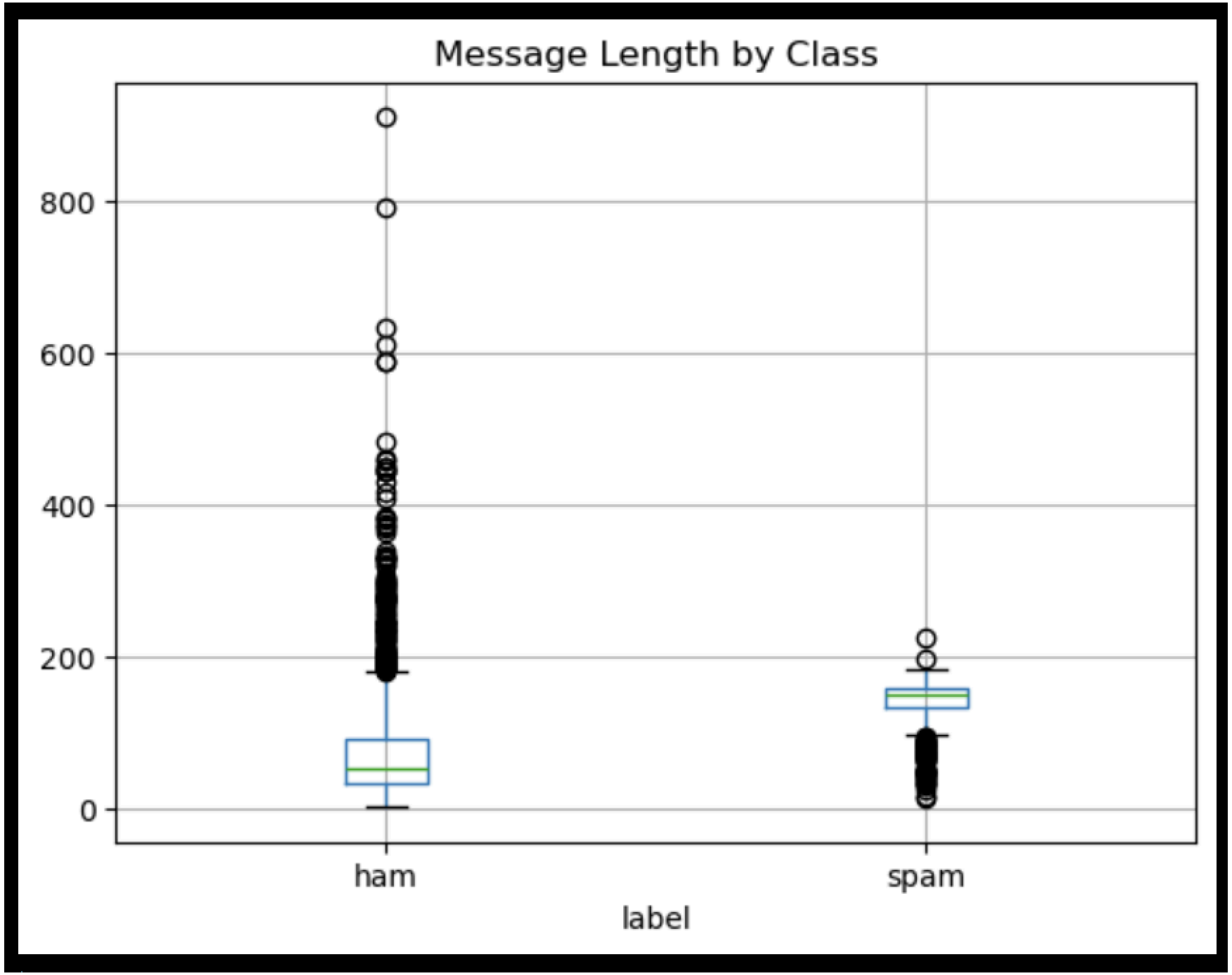
```
Sample cleaned data:
```

	text	cleaned_text
0	Go until jurong point, crazy.. Available only ...	go jurong point crazy available bugis n great ...
1	Ok lar... Joking wif u oni...	ok lar joking wif u oni
2	Free entry in 2 a wkly comp to win FA Cup fina...	free entry wkly comp win fa cup final tkts st ...
3	U dun say so early hor... U c already then say...	u dun say early hor u c already say
4	Nah I don't think he goes to usf, he lives aro...	nah dont think go usf life around though

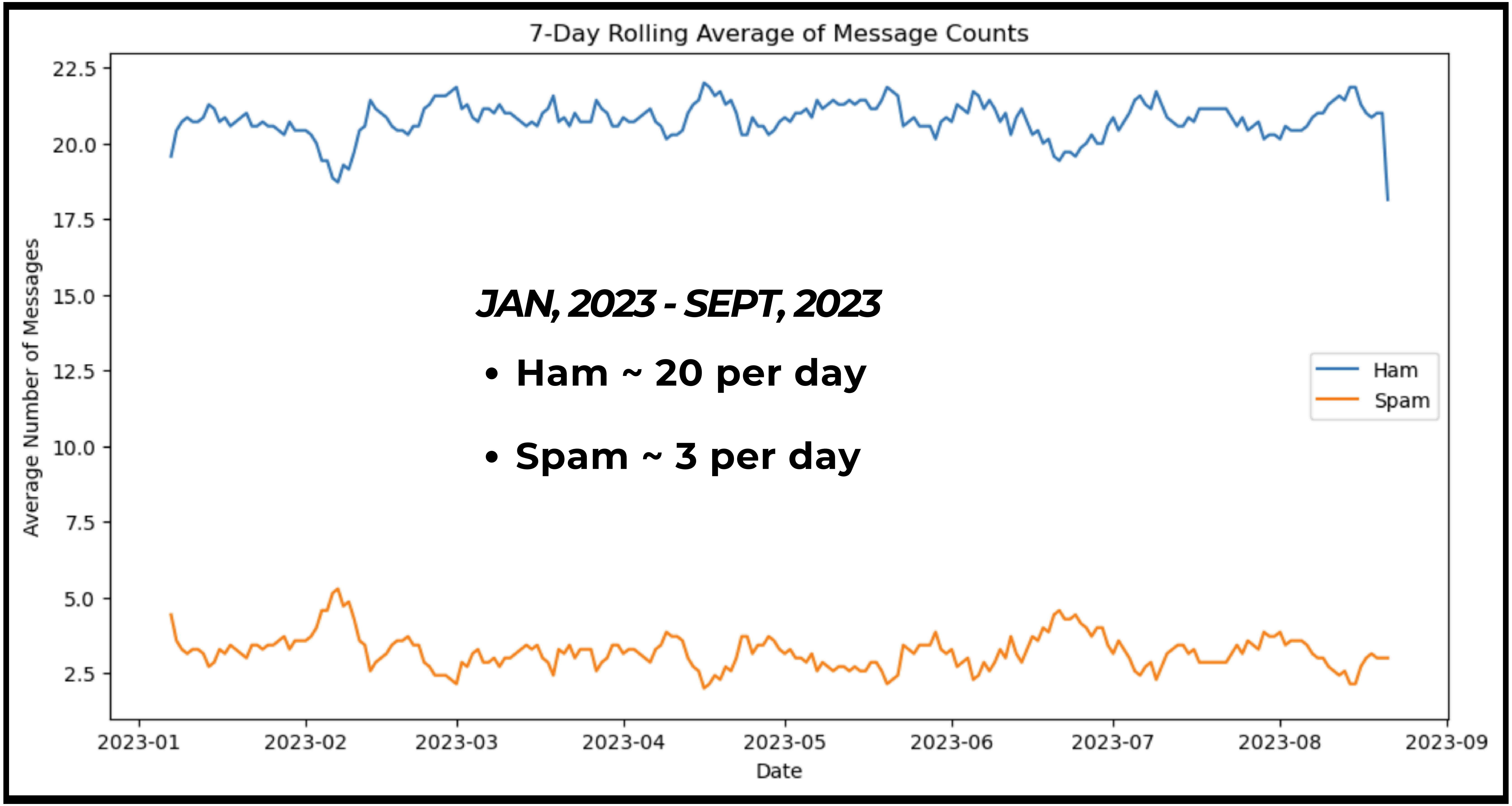
FEATURE ENGINEERING

MESSAGE LENGTH ANALYSIS:

- **Ham - a larger distribution of short messages with extremely long outliers**
- **Spam - Message length on average is roughly double Ham. Extremely short outliers.**



TIME SERIES ANALYSIS



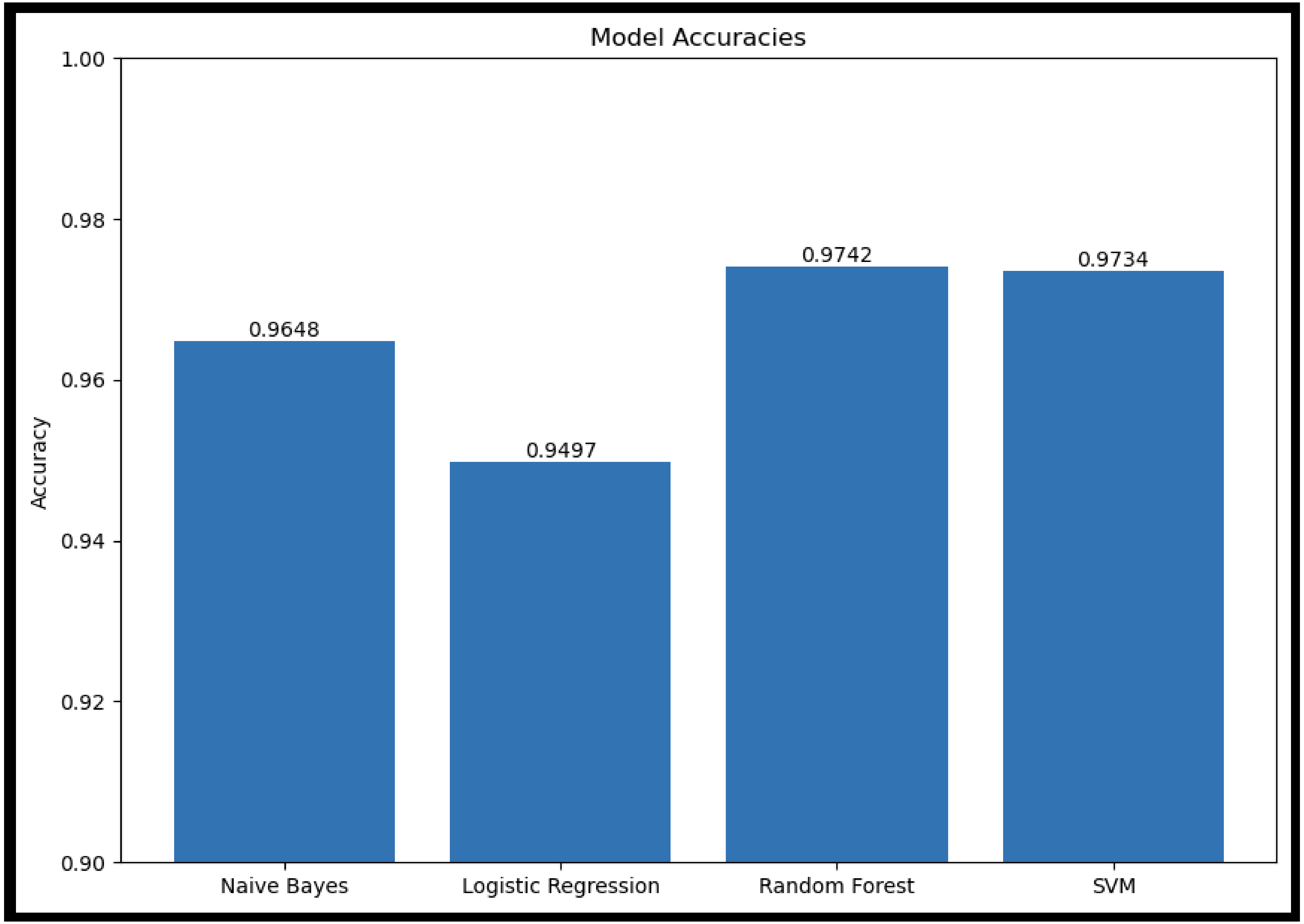
MODEL TRAINING PROCESS

- **Vectorization: TF-IDF**
- **Models tested: Naive Bayes, Logistic Regression, Random Forest, SVM**
- **Training/Test split: 75/25**

MODEL COMPARISON

PERFORMANCE IN %

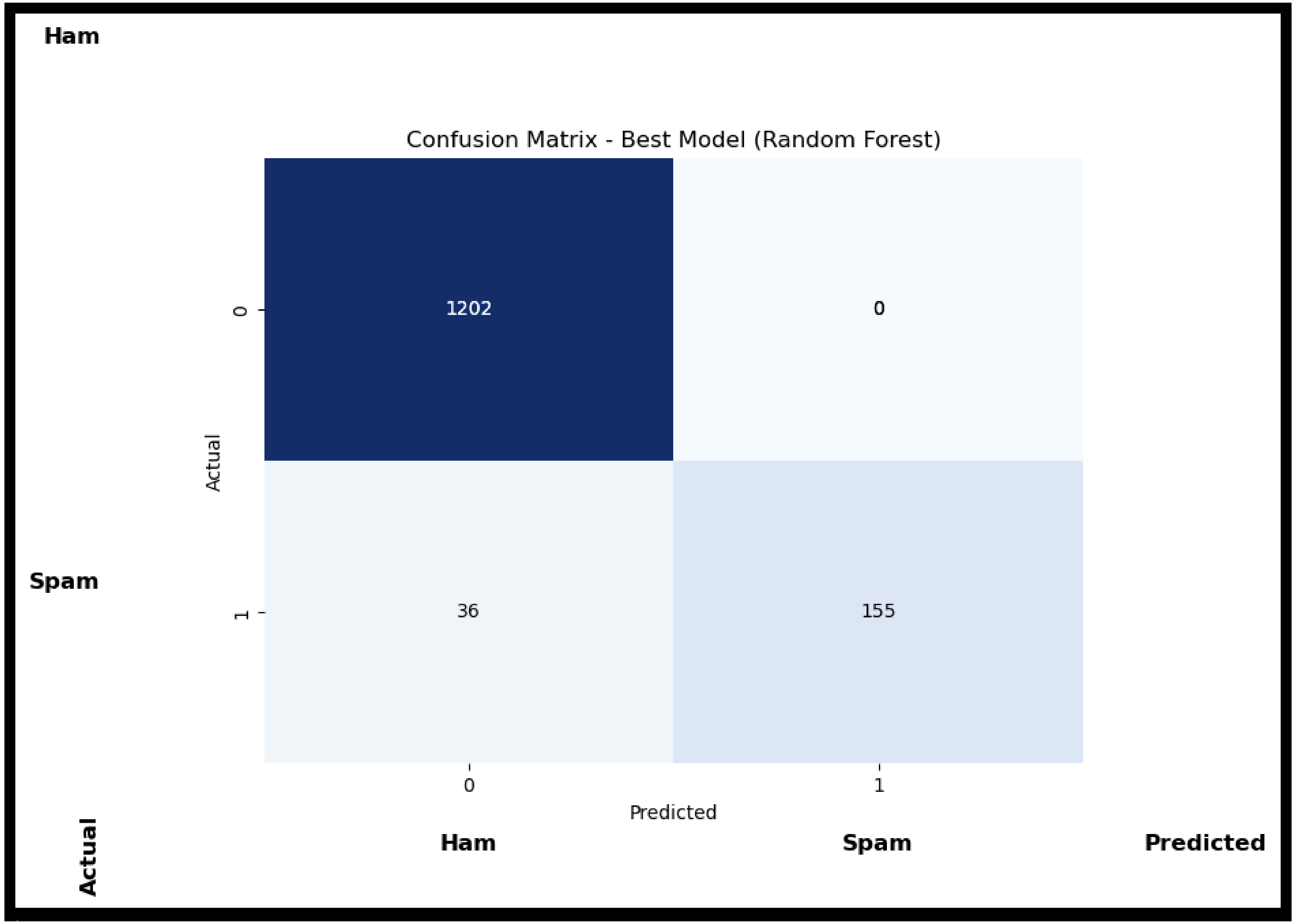
- Naive Bayes: 96.48%
- Logistic Regression: 94.97%
- Random Forest: 97.42%
- SVM: 97.34%



BEST PERFORMANCE

RANDOM FOREST

- **True Positives = 1202**
- **False Negatives = 0**
- **False Positives = 36**
- **True Negatives = 155**



DETAILED METRICS

```
Final Evaluation on Test Set:
Classification Report:
              precision    recall  f1-score   support

         ham       0.97      1.00      0.99      1202
        spam       1.00      0.81      0.90       191

    accuracy                           0.97      1393
   macro avg       0.99      0.91      0.94      1393
weighted avg       0.97      0.97      0.97      1393
```

KEY FINDINGS

1. **High Model Accuracy: The best-performing model achieved an accuracy of 97.42% on the test set. This significantly exceeds the project requirement of 75%. All models tested had accuracies above 94%**
2. **Excellent Ham Detection: The final evaluation shows perfect recall (1.00) for ham messages, meaning the model correctly identified 100% of legitimate messages.**
3. **Strong Spam Precision: The model achieved perfect precision (1.00) for spam messages. This means that when the model classified a message as spam, it was correct 100% of the time.**

MODEL TUNING

HYPER PERAMETER TUNING BEST MODEL - RANDOM FOREST

```
Hyperparameter Tuning Results:
Best parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 300}
Best cross-validation score: 0.9744
Original model score: 0.9742
Improvement over original model: 0.0002
```

- **A whopping 00.02% (Better than nothing)**
- **The model already performed extremely well.**
- **Little room for improvement**

FUTURE IMPROVEMENT

ADDITIONAL TUNING

- Feature Importance Analysis
- Cross-validation
- Threshold adjustment - optimize the model's precision recall trade-off

DEEP LEARNING APPROACHES

- Experiment with neural networks, particularly recurrent neural networks (RNNs) or transformers, which can capture sequential information in text.

PRACTICAL EXAMPLE

- ***SPAM MESSAGE ANALYSIS (DMS, FB, IG, ETC, NOT JUST SMS)***
- ***IMAGE NOT TEXT ANALYSIS***
- ***DEEP LEARNING - LLM***

1. **Image Upload to Web Interface**
2. **Image to OCR API to extract text**
3. **Return**
4. **Text to LLM for Analysis**
5. **Return Safe Output and Print Analysis**

THANK YOU