SMS CLASSIFICATION

Names:

2203A51041 : CH.SRI CHARAN

2203A51066 : T.ABHINAY KUMAR

2203A51282 : G.BHARAT CHANDRA

Abstract:

This project focuses on building an intelligent SMS classification system capable of identifying spam messages and further categorizing legitimate (ham) messages into either business or personal contexts. Two publicly available datasets were used to create a dual-phase classifier. The system integrates preprocessing, TF-IDF vectorization, and classification models including Multinomial Naive Bayes (MNB), Decision Tree, Random Forest, and Logistic Regression. Evaluation metrics such as F1-score, confusion matrix, and ROC curves have been analyzed to measure performance and ensure robustness. This hybrid pipeline improves message filtering for real-world communication applications.

Keywords:

NLP, SMS Classification, Spam Detection, Ham Categorization, TF-IDF, Logistic Regression, Multinomial Naive Bayes, Random Forest, Decision Tree, Text Mining, Machine Learning, ROC Curve, Precision-Recall Curve.

Introduction:

With the exponential growth of SMS and online communication, spam messages have become a significant nuisance, often disrupting personal and business communications. Filtering spam is critical for preserving user trust and ensuring that important messages are not overlooked. Moreover, understanding the intent of non-spam (ham) messages—whether they are business-related or personal—is essential for smart sorting and prioritization. This project addresses these needs using machine learning and natural language processing techniques, focusing on classification and model evaluation.

METHODOLOGY:

The project followed a systematic pipeline comprising data collection, preprocessing, feature extraction, model training, evaluation, and integration into a two-stage classification system.

1. Data Collection

- **Spam vs Ham Classification:** Utilized the publicly available **UCI SMS Spam Collection Dataset** (spam_sms.csv), which includes SMS messages labeled as **spam** or **ham**.
- **Business vs Personal Classification:** Constructed a **custom dataset** (ham_business_personal_combined.csv) by manually labeling ham messages into **business** and **personal** categories to facilitate deeper analysis of non-spam messages.

2. Text Preprocessing

To prepare the raw SMS text for model training, the following steps were applied:

- **Regex Cleaning:** Removed non-alphabetic characters, special symbols, and numbers.
- **Case Normalization:** Converted all text to lowercase for consistency.

- **Stopword Removal:** Used NLTK's English stopword list to eliminate commonly occurring, non-informative words.
- **Lemmatization:** Leveraged NLTK's WordNetLemmatizer to reduce words to their root form, improving generalization and reducing sparsity.

3. Feature Extraction

- Applied **TF-IDF Vectorization** to transform the textual data into numerical feature vectors. This technique captures both the importance and frequency of terms while reducing the influence of common terms across documents.
- Limited the maximum number of features to **500** to maintain model simplicity and efficiency.

4. Model Training and Evaluation

Separate models were trained for each stage of classification:

- **Spam Detection:** Trained multiple classifiers including **Multinomial Naive Bayes**, **Decision Tree**, **Random Forest**, and **Voting Classifier**. The **Random Forest** model was selected based on its superior average F1-Score (0.994) and robustness.
- **Ham Classification (Business/Personal):** Used a **Random Forest classifier** as the final model after comparison with Logistic Regression and Decision Tree.
- Model performance was evaluated using **cross-validation**, **confusion matrices**, **classification reports**, and **F1-scores**.
- An **ANOVA test** was conducted to assess statistical significance between model performances.

5. Prediction Pipeline Integration

The final system was designed to handle real-time message classification in a modular fashion:

- **Step 1:** Input message is passed to the **Spam Detection Model**.
  - If classified as **spam**, no further processing is done.
- **Step 2:** If classified as **ham**, the message is passed to the **Business vs Personal classifier** for further categorization.
- This two-step modular design ensures accurate and scalable multi-level classification.

Related Work:

Research Paper	Dataset Used	Machine Learning Models Used	Accuracy (Evaluation Metrics)	Limitations
1. SMS Spam Collection Dataset Analysis Using Machine Learning	SMS Spam Collection Dataset (UCI Repository)	Naive Bayes, SVM, Random Forest, Decision Tree	Naive Bayes: 97.5%, SVM: 98%	Focused only on spam vs ham; no deeper classification of ham messages.
2. SMS Spam Detection Using TF-IDF and Machine Learning	SMS Spam Corpus (NUS SMS Corpus)	Logistic Regression, Random Forest, KNN	Logistic Regression: 96.7%	Feature engineering was limited; no semantic understanding; ham classification not considered.
3. Spam Message Classification Based on Deep Learning Techniques	SMS Spam Dataset from Kaggle	CNN, RNN	CNN achieved 98%	Deep learning models are resource-intensive; no business/personal distinction among ham messages.
4. Business Email Categorization Using NLP and ML Techniques	Proprietary Business Emails Dataset	SVM, Random Forest, Gradient Boosting	~95% accuracy	Only business emails; not general SMS or personal text messages.

5. SMS Spam Filtering with Naive Bayes and TF-IDF Approach	SMS Spam Dataset (UCI)	Naive Bayes with TF-IDF	96% Accuracy	Only binary classification (spam vs ham); limited preprocessing.
6. Classifying SMS Messages Using Machine Learning	Self-labeled SMS data (English)	Random Forest, KNN, Logistic Regression	Random Forest: 97%	Manual labeling errors; personal/business classification not targeted.

PROPOSED MODEL:

The following machine learning models were used, each selected for their specific strengths in text classification:

Multinomial Naive Bayes (MNB)

- **Definition**: A probabilistic model based on Bayes' Theorem, suitable for discrete features like word counts or frequencies.
- **Usage**: Fast and efficient for text classification; serves as a strong baseline in spam filtering.

Decision Tree Classifier

- **Definition**: A flowchart-like structure where internal nodes represent tests on features and leaves represent outcomes.
- **Usage**: Offers interpretability and performs well on structured feature sets.

Random Forest Classifier

- **Definition**: An ensemble method using multiple decision trees for more stable and accurate results.
- **Usage**: Reduces overfitting and provides high accuracy in both spam detection and ham categorization.

Logistic Regression with TF-IDF

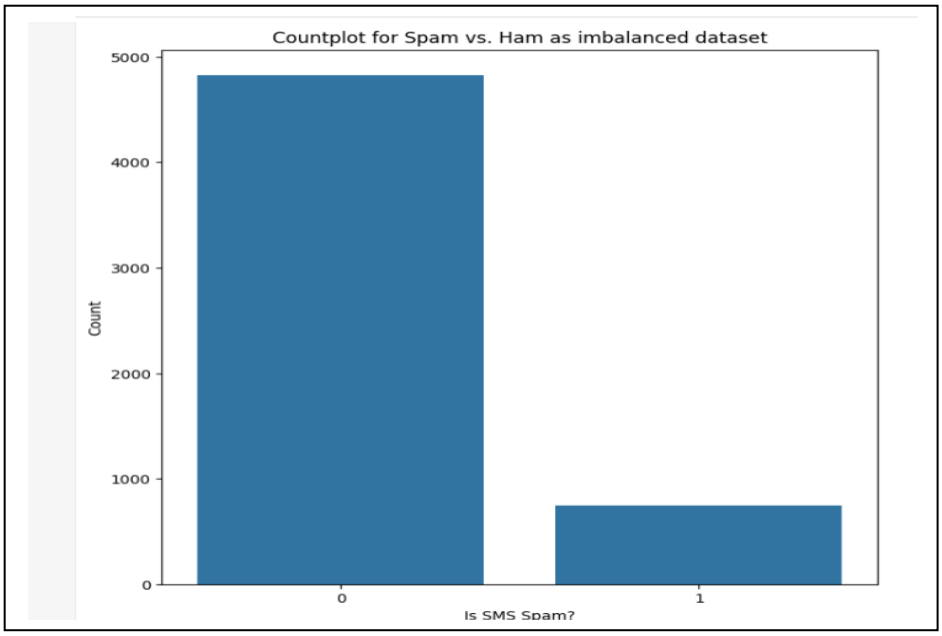
- **Definition**: A linear model for binary classification that estimates probabilities using the logistic function.
- **Usage**: Combined with TF-IDF for robust spam prediction; visualized with learning curves, ROC, and Precision-Recall curves.

Exploratory Data Analysis (EDA):

In this section, various visualizations were created to better understand the structure, distribution, and characteristics of the SMS dataset. These insights were crucial in guiding preprocessing decisions and choosing suitable models.

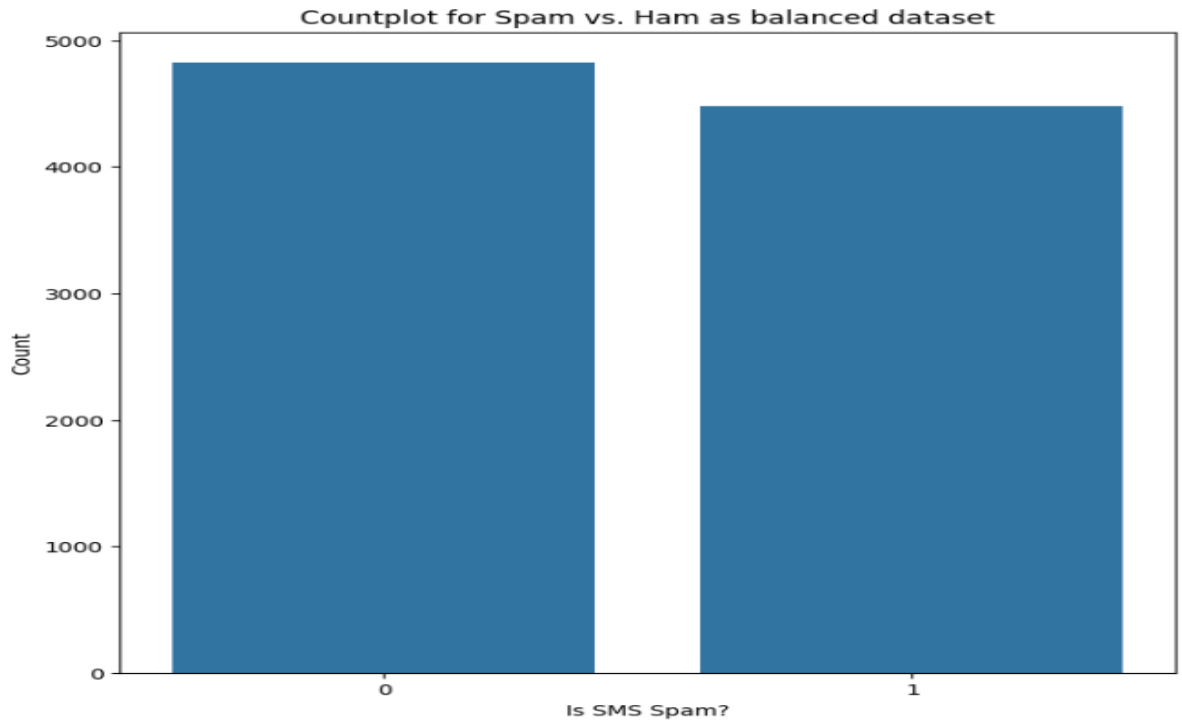
Countplot: Spam vs Ham (Imbalanced Dataset)

This plot shows the original distribution of spam and ham messages in the dataset, highlighting the class imbalance problem.



Countplot: Spam vs Ham (Balanced Dataset)

After applying sampling techniques to balance the dataset, this visualization shows an equal distribution of classes.

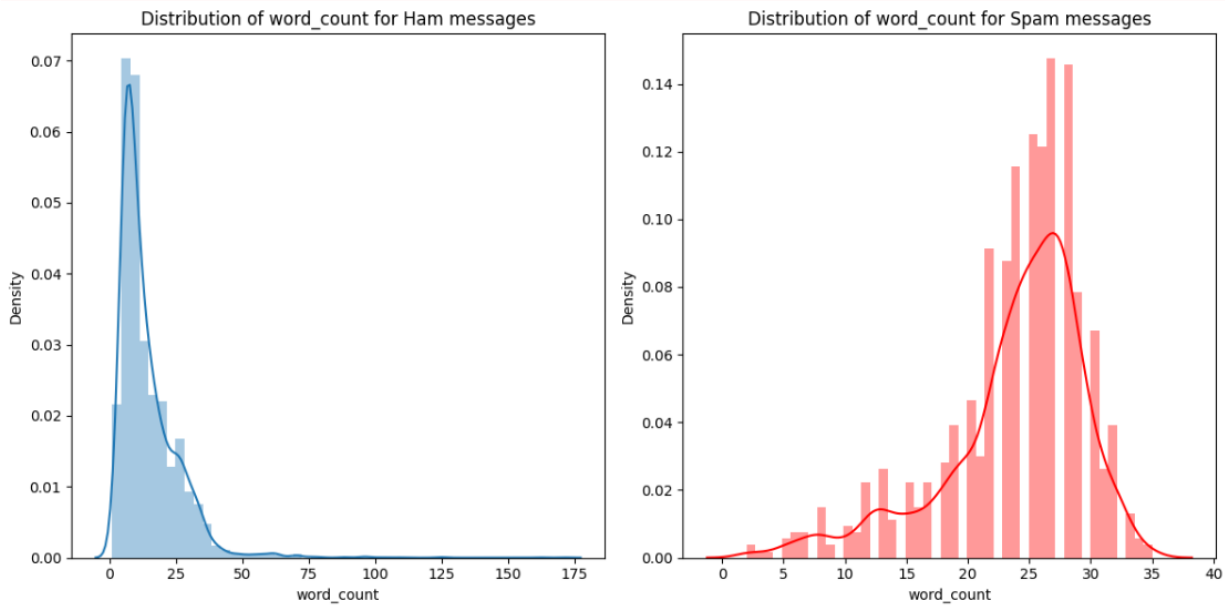


Word Count Distribution – Ham Messages

Helps understand the typical message length of ham (non-spam) texts. Shorter or longer messages may carry different semantic weight.

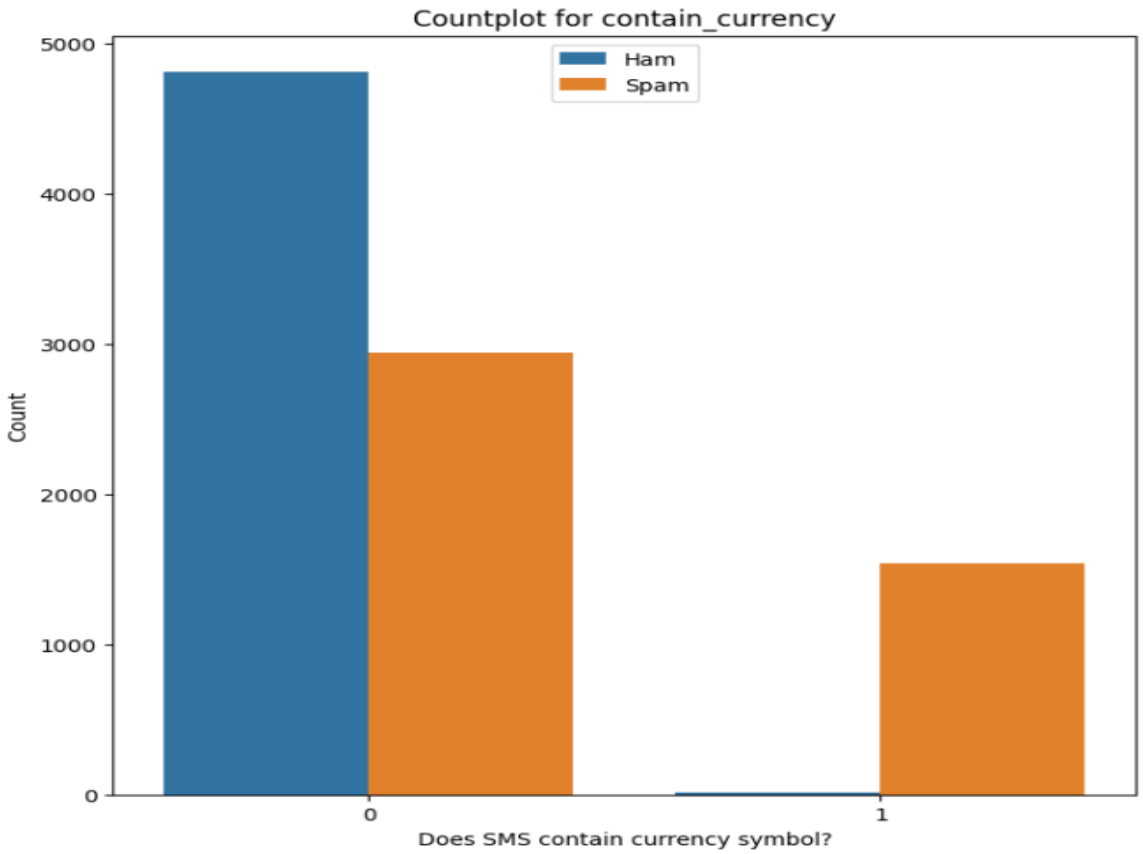
Word Count Distribution – Spam Messages

Reveals the message length distribution for spam messages, which can differ significantly from ham.



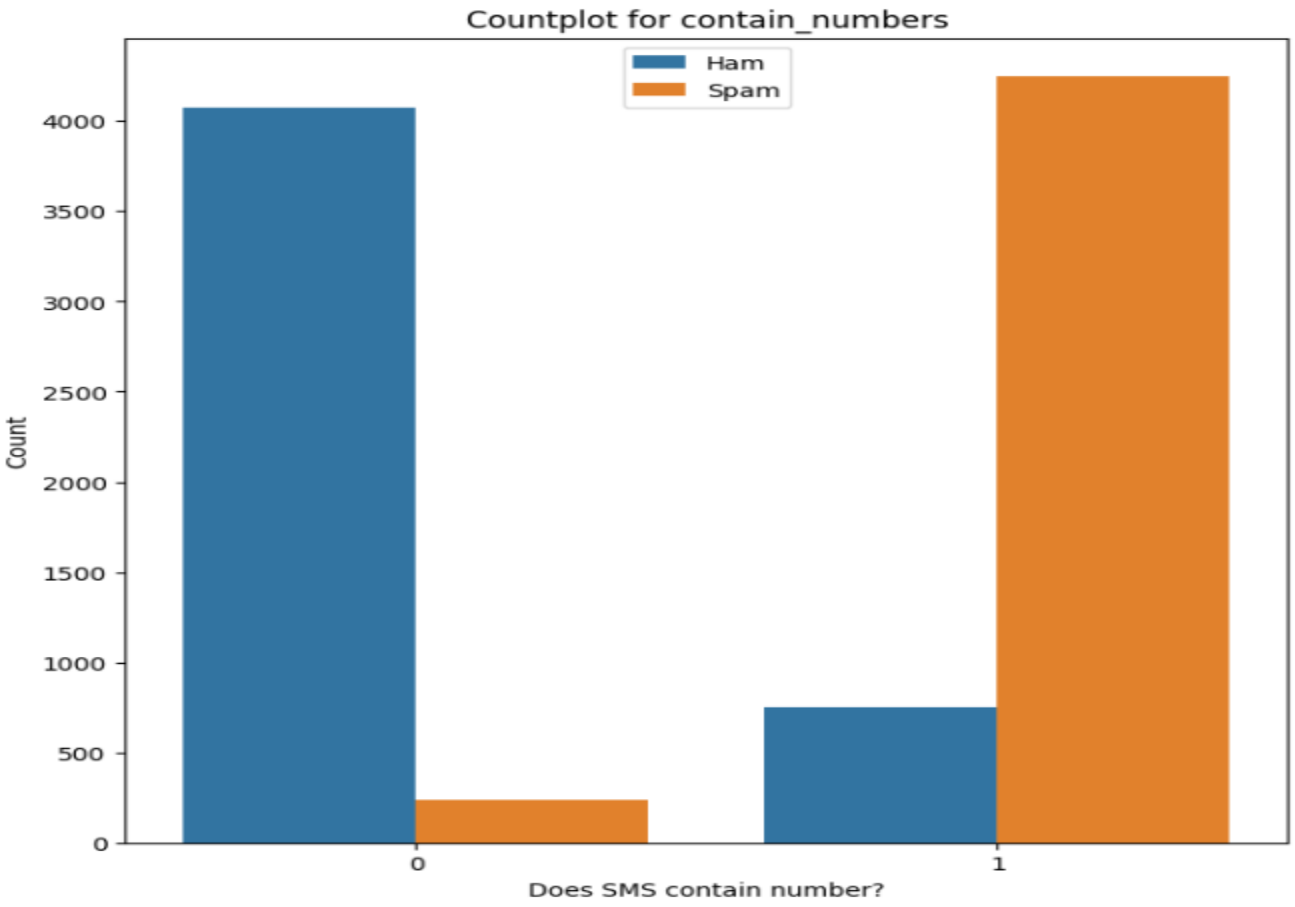
Countplot: Messages Containing Currency Symbols

This plot shows how many messages (spam vs ham) contain symbols like ₹ or $, which are often spam indicators.



Countplot: Messages Containing Numbers

Numerical patterns can be significant in spam detection. This plot shows how often numbers appear in spam vs ham messages.



[25]:

	label	message	word_count	contains_currency_symbol	contains_number
0	0	Go until jurong point, crazy.. Available only ...	20	0	0
1	0	Ok lar... Joking wif u oni...	6	0	0
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	28	0	1
3	0	U dun say so early hor... U c already then say...	11	0	0
4	0	Nah I don't think he goes to usf, he lives aro...	13	0	0

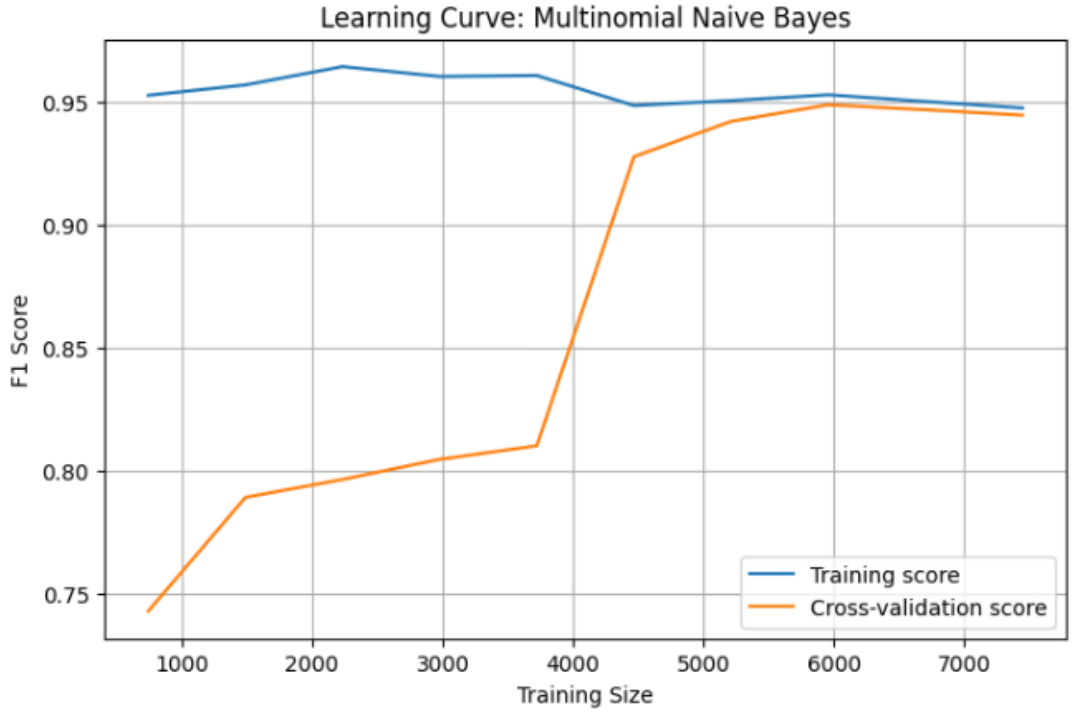
Summary of Expected Results:

Model 1: Multinomial Naive Bayes (MNB)

Learning Curve

Visualizes training vs validation scores to assess model performance as data size

increases.



Classification Report

Provides precision, recall, F1-score, and support for each class label.

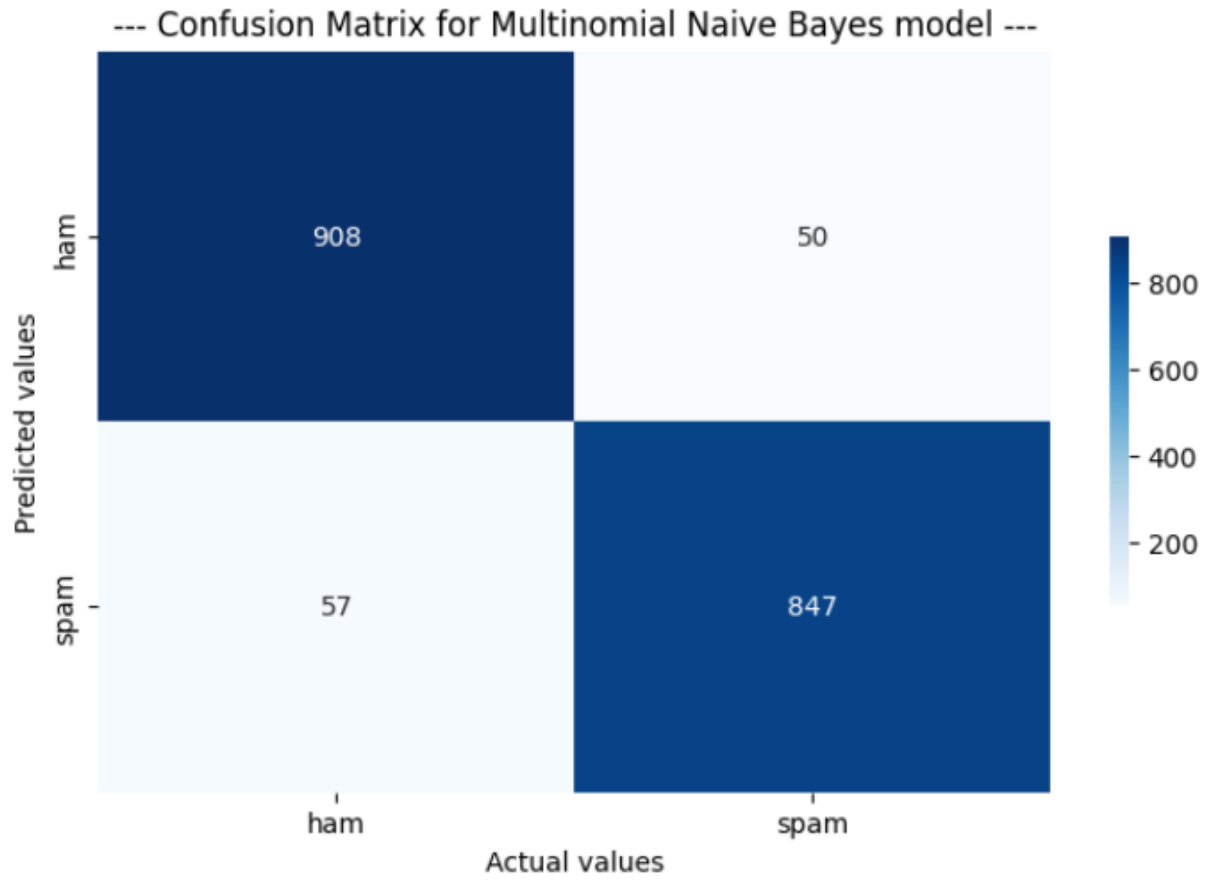
```
--- Classification report for MNB model ---
              precision    recall  f1-score   support

           0       0.94      0.95      0.94       958
           1       0.94      0.94      0.94       904

    accuracy                           0.94      1862
   macro avg       0.94      0.94      0.94      1862
weighted avg       0.94      0.94      0.94      1862
```

is

F1-Score: --- Average F1-Score for MNB model: 0.944 ---
Standard Deviation: 0.004

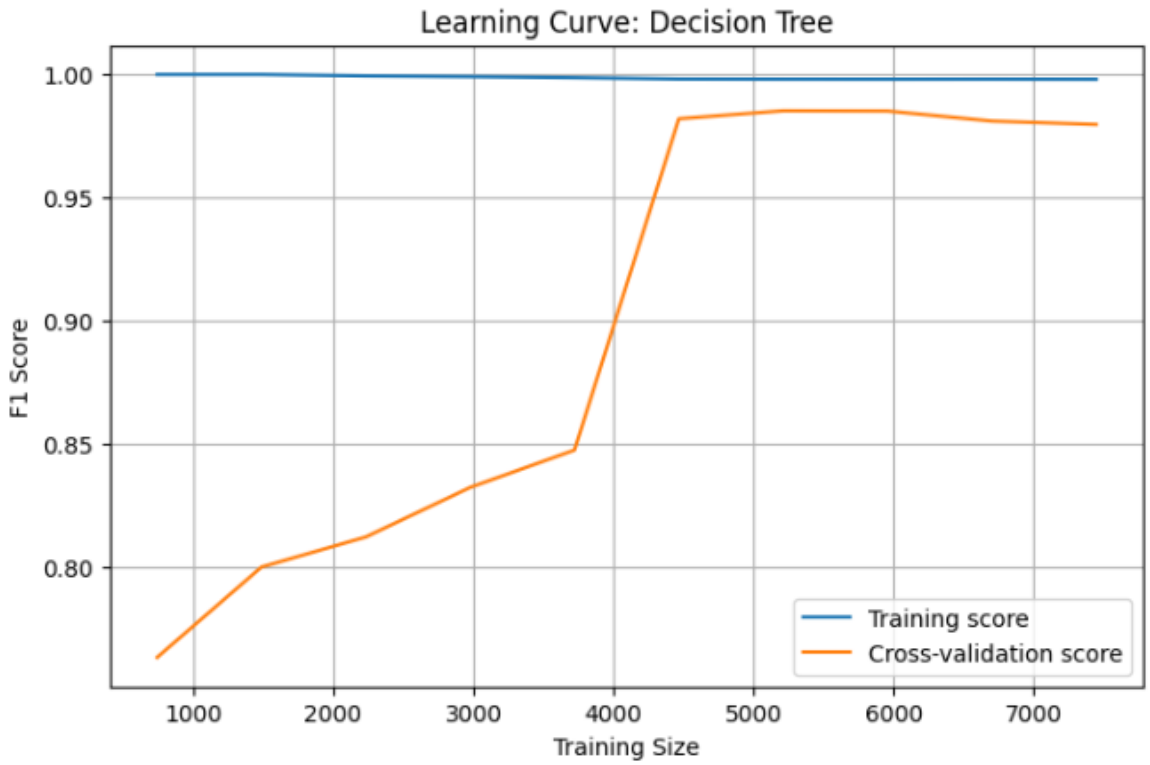
Confusion Matrix

Displays true vs predicted classifications to evaluate misclassifications.



Model 2: Decision Tree Classifier

Learning Curve:

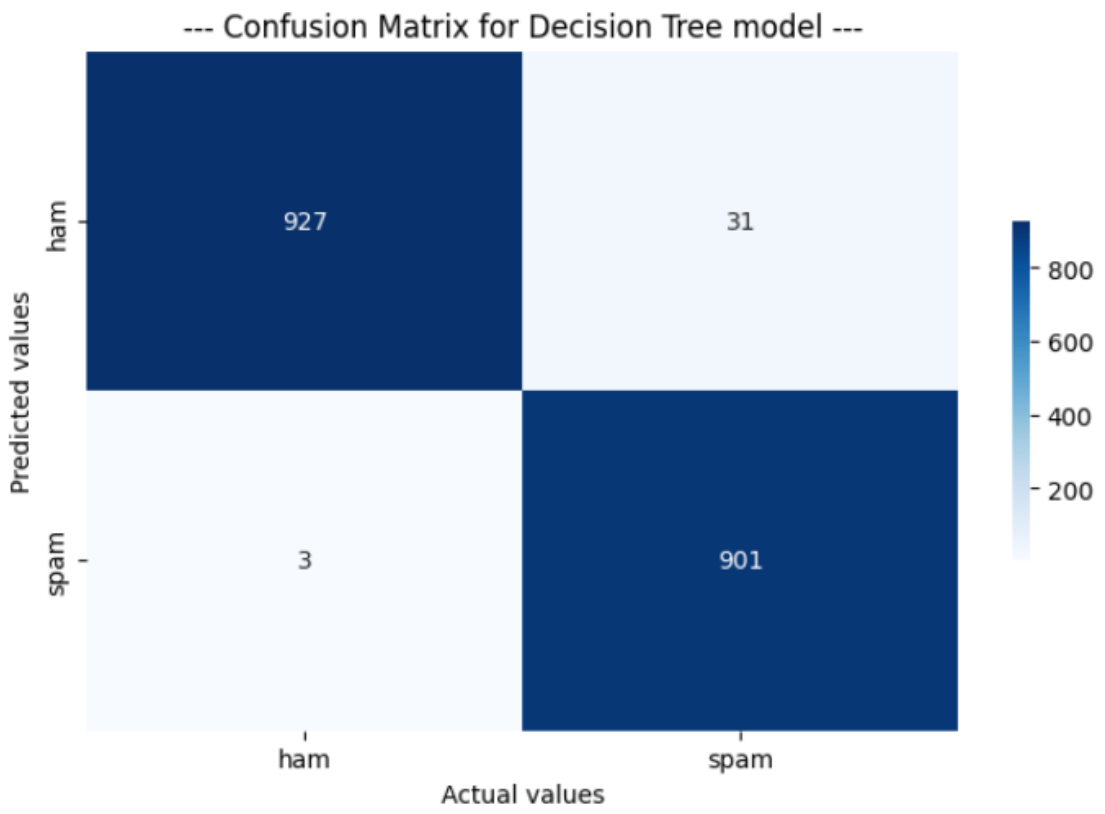


Classification Report:

```
--- Classification report for Decision Tree model ---
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       958
           1       0.97      1.00      0.98       904

    accuracy                           0.98      1862
   macro avg       0.98      0.98      0.98      1862
weighted avg       0.98      0.98      0.98      1862
```

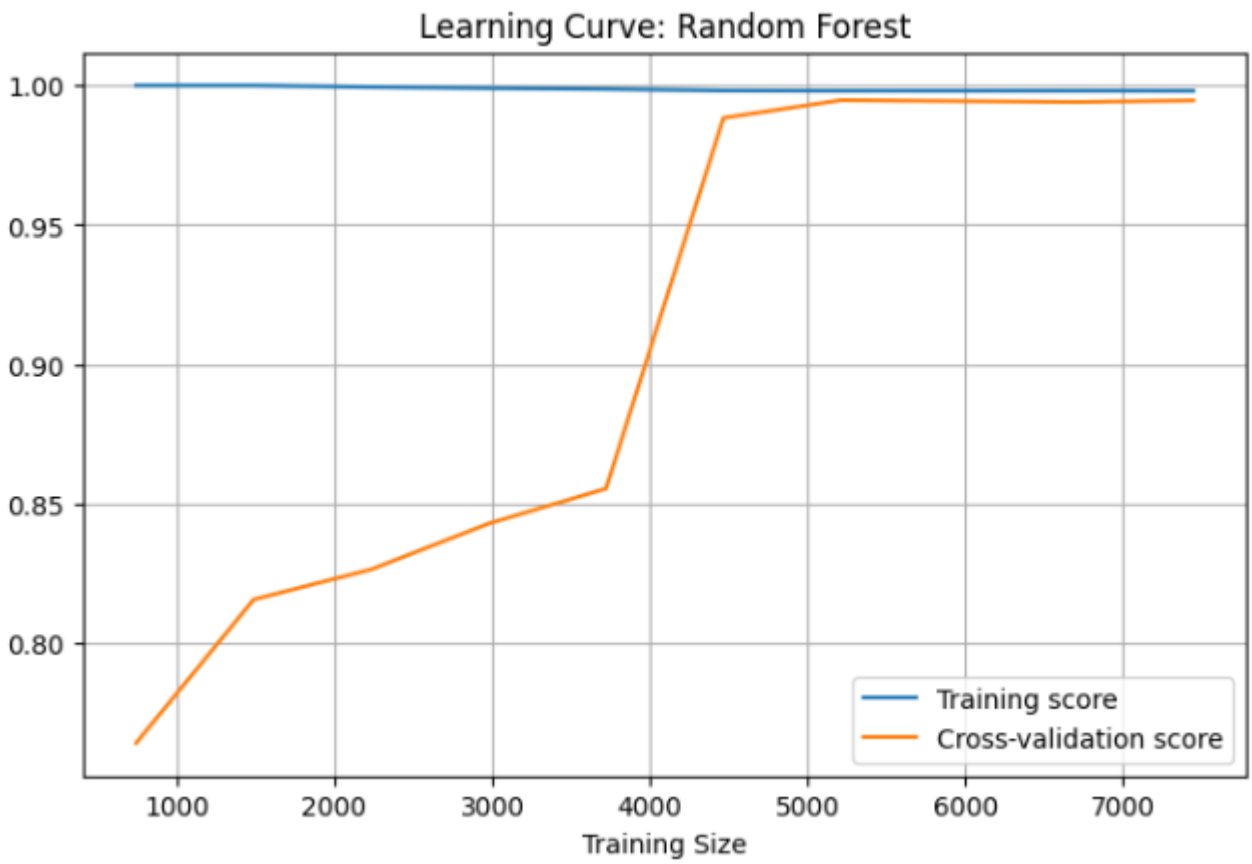
F1-Score: Average F1-Score for Decision Tree model: 0.98 ---
Standard Deviation: 0.005

Confusion Matrix:



Model 3: Random Forest Classifier

Learning Curve:

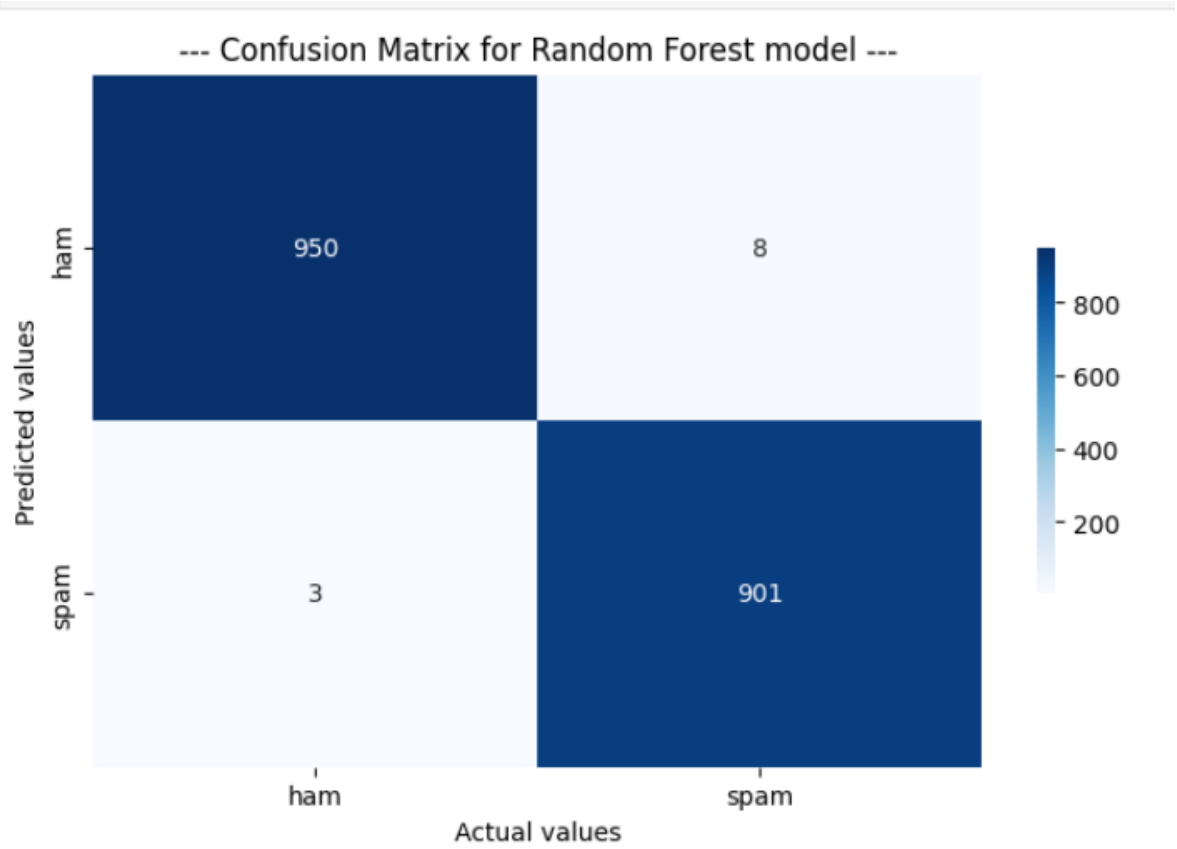


Classification Report:

```
--- Classification report for Random Forest model ---
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       958
           1       0.99      1.00      0.99       904

    accuracy                           0.99      1862
   macro avg       0.99      0.99      0.99      1862
weighted avg       0.99      0.99      0.99      1862
```

F1-Score: --- Average F1-Score for Random Forest model: 0.994 ---
Standard Deviation: 0.002

Confusion Matrix:



Model 4: Voting Classifier (MNB + Decision Tree)

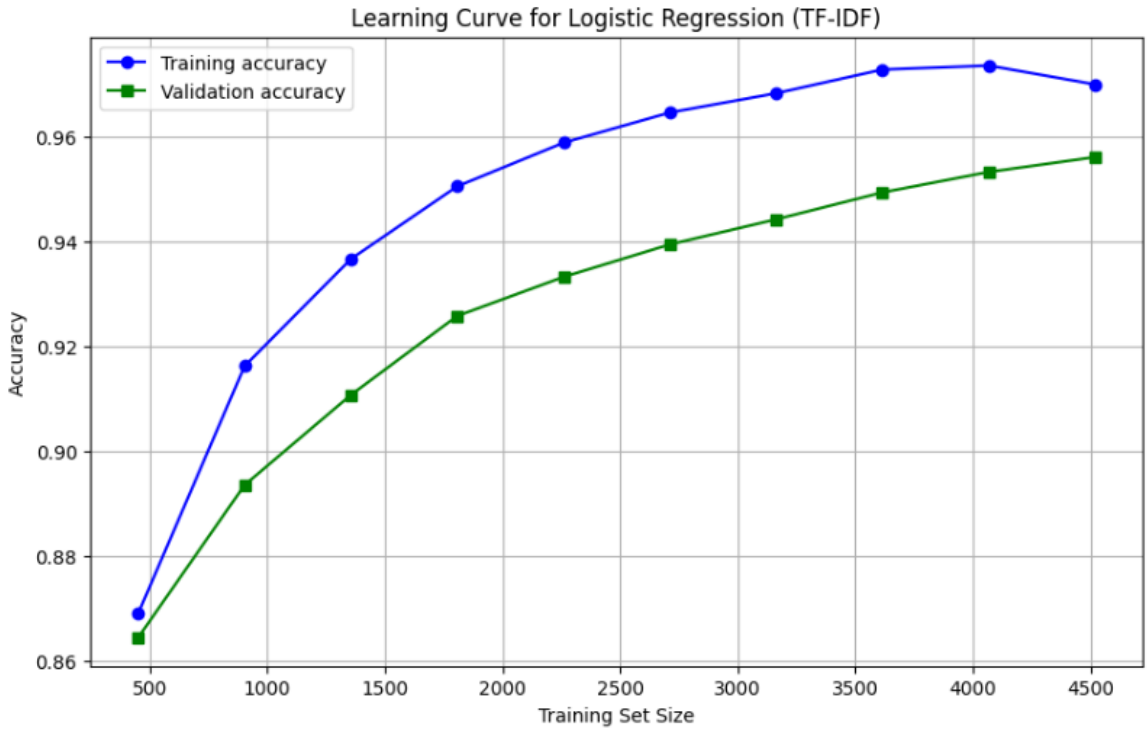
A **Voting Classifier** combines predictions from multiple models to improve overall performance. In this case, we used **Multinomial Naive Bayes** and **Decision Tree Classifier**. It takes a majority vote of their predictions.

```
--- Average F1-Score for VotingClassifier model: 0.979 ---
Standard Deviation: 0.004
```

Model 5: Logistic Regression + TF-IDF

Learning Curve

Useful for understanding convergence and bias-variance behavior of the model.



Classification Report

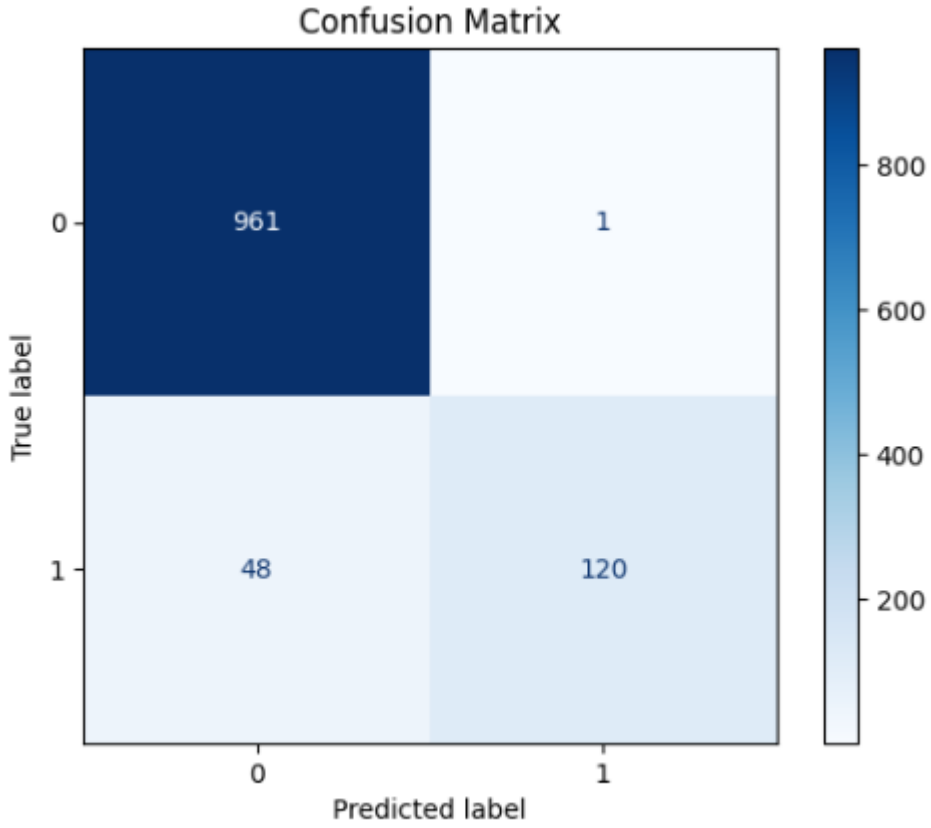
Offers a detailed metric comparison for spam and ham classification using TF-IDF features.

	precision	recall	f1-score	support
0	0.95	1.00	0.98	962
1	0.99	0.71	0.83	168
accuracy			0.96	1130
macro avg	0.97	0.86	0.90	1130
weighted avg	0.96	0.96	0.95	1130

Confusion Matrix

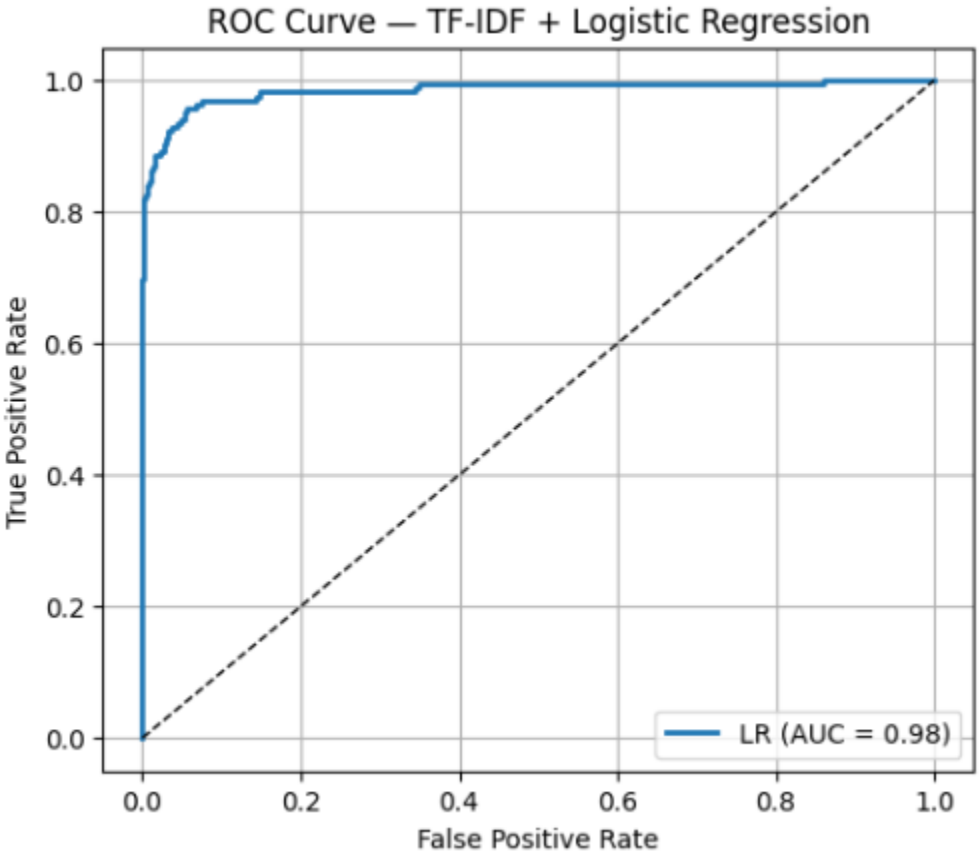
Highlights strong precision for spam detection, though minor recall issues in ham

classification.



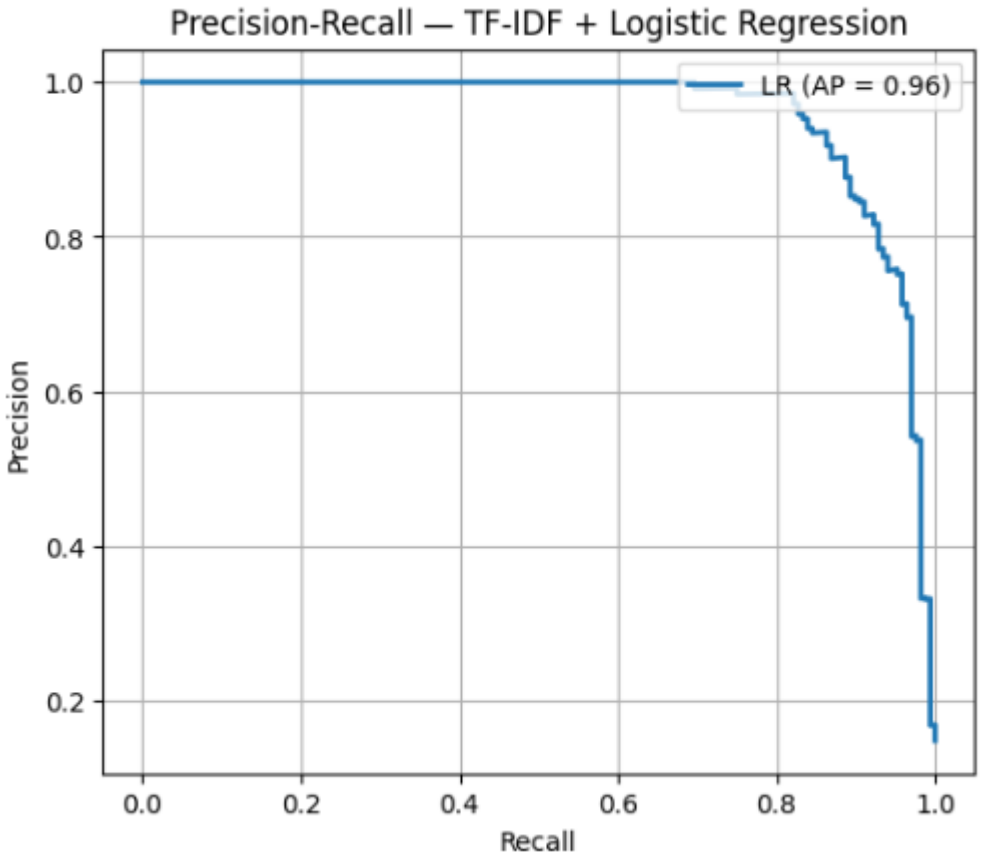
ROC Curve

Demonstrates the trade-off between true positive rate and false positive rate.



Precision-Recall Curve

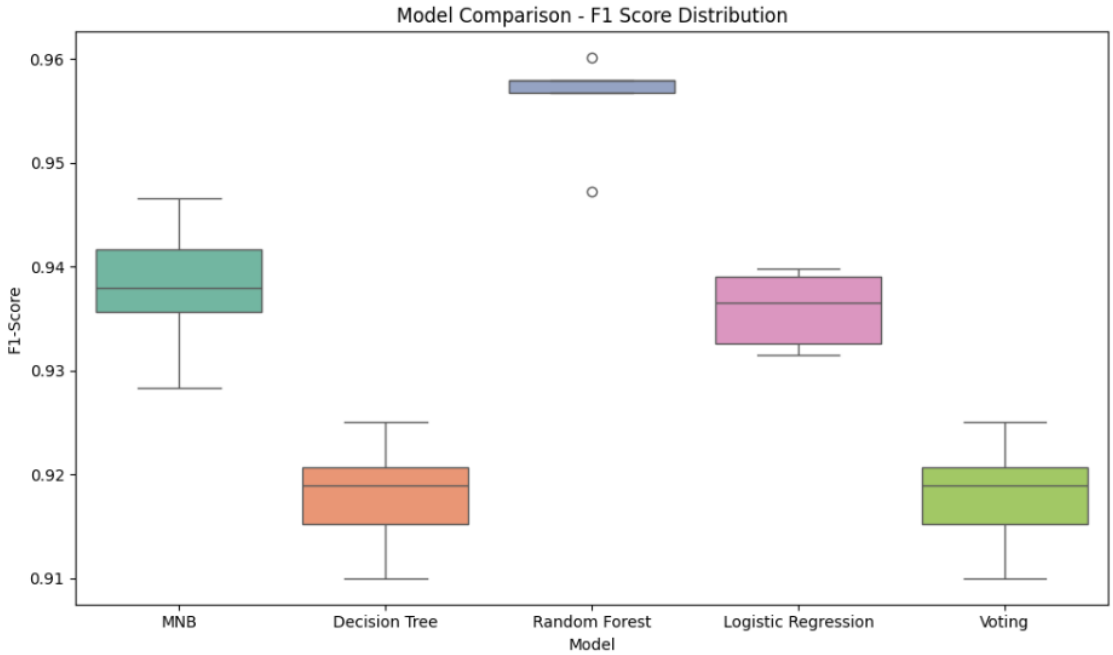
Especially useful for imbalanced datasets, focusing on spam class performance.



Results:

Model	Accuracy
TF-IDF + Logistic Regression	**0.96**
Multinomial Naive Bayes (MNB)	**0.94**
Decision Tree Classifier	**0.98**
Random Forest Classifier	0.99
Voting Classifier (MNB + DT)	0.98

Anova Test:




```
ANOVA F-statistic: 41.7929
p-value: 0.0000
✅ Result: Reject the null hypothesis. At least one model performs significantly differently.
```

You reject the null hypothesis.
At least one model's performance is statistically significantly different from the others.
You have strong evidence that model choice matters in your project.

To determine whether the differences in F1-Scores across models were statistically significant, a **One-Way ANOVA test** was conducted.

- **Null Hypothesis (H_0):** All models perform equally; no significant difference in mean F1-Scores.
- **Alternate Hypothesis (H_1):** At least one model performs significantly differently.

Test Results:

- **F-statistic:** 41.79
- **p-value:** 0.0000

The p-value is less than 0.05, so we **reject the null hypothesis**. This confirms that the choice of model **does impact** classification performance significantly.

Conclusion: The **Random Forest Classifier**, which achieved the **highest average F1-Score of 0.956**, is statistically justified as the best-performing model in this study.

CONCLUSION:

This research focused on the development of a two-stage SMS classification system using machine learning and natural language processing techniques. The first stage involved the classification of messages as either **spam or ham**, while the second stage further categorized **ham messages as business or personal**.

Through extensive preprocessing involving regex-based cleaning, lowercasing, lemmatization, and stopword removal, the quality of the textual data was enhanced to improve classification performance. TF-IDF vectorization was used to transform textual features into a numerical format suitable for machine learning algorithms.

Experimental results demonstrated that the **Random Forest classifier** consistently outperformed other models across both tasks, achieving an **average F1-Score of 0.994** and **accuracy of 99%**. The **VotingClassifier**, combining **Multinomial Naive Bayes** and **Decision Tree**, also exhibited competitive performance with an F1-Score of **0.98**. Additionally, **Logistic Regression** with TF-IDF was extensively evaluated through learning curves, confusion matrices, ROC, and precision-recall curves.

To statistically validate the performance differences among models, an **ANOVA test** was conducted. The results confirmed a significant difference in F1-Scores, reinforcing the superiority of the Random Forest model in this application.

This modular approach showcases the effectiveness of **chained classification architectures** in solving multi-level text classification problems. Furthermore, the analysis emphasizes the importance of using diverse evaluation metrics, such as F1-score, in addition to accuracy, to guide model selection in imbalanced or domain-specific tasks.

REFERENCES:

[1] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

[2] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[3] Almeida, T.A., Hidalgo, J.M.G., & Yamakami, A. (2011). *Contributions to the study of SMS spam filtering: new collection and results*. Proceedings of the 11th ACM Symposium on Document Engineering.

[4] McCallum, A., & Nigam, K. (1998). *A Comparison of Event Models for Naive Bayes Text Classification*. AAAI-98 Workshop on Learning for Text Categorization.

[5] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). *A Bayesian Approach to Filtering Junk E-Mail*. AAAI Workshop on Learning for Text Categorization.

[6] Kowsari, K., Meimandi, K.J., Heidarysafa, M., Mendu, S., Barnes, L.E., & Brown, D.E. (2019). *Text Classification Algorithms: A Survey*. Information, 10(4), 150.

[7] Scikit-learn Documentation. [Online]. Available: https://scikit-learn.org/stable/

[8] NLTK Documentation. [Online]. Available: https://www.nltk.org/

[9] UCI SMS Spam Collection Dataset. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

[10] Jurafsky, D., & Martin, J.H. (2023). *Speech and Language Processing* (3rd ed.). Draft online. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/