Detection of SPAM in Emails

Arti Ravi Garg
Ryan Richardson
Nolan Arendt

Introduction

Our Goals:

- Accurately detect SPAM emails in the digital communication era.

Objective:

- Develop a machine learning classifier that is capable of distinguishing between SPAM and authentic emails.

Justification:

- Enhancing email filtering processes that can protect users from malicious content within SPAM emails.

Dataset Overview & Preprocessing

Utilizing the Enron Email Dataset which contains 3,672 non-SPAM emails, and 1,500 SPAM emails.

- Balanced for model efficiency.

Preprocessing steps include tokenization, stop word removal, and normalization of text data.

Preprocessing was determined based on the initial analysis conducted to further improve classifier performance.

Feature Extraction and Selection

Converted emails into vector representations using a bag-of-words approach, focusing on word frequencies to construct the initial feature set.

Experimented with bi-gram features to capture contextual information by considering pairs of consecutive words.

Implemented part-of-speech tagging to differentiate words based on their grammatical roles.

Selected top 2000 most frequent words to further reduce the dimensionality and only focus on the most relevant and influential features.

Classifier Overview and Selection Criteria

Classifiers Evaluated:

- NLTK's Naïve Bayes
- Gradient Boosting
- Random Forest
- Logistic Regression
- Support Vector Machine

Selection Criteria: Accuracy, Precision, Recall, and F1-Score

Evaluation: Cross-validation to ensure reliability across different data partitions (5)

NAIVE BAYES CLASSIFIER PERFORMANCE

Unigram Approach: involves using individual words (unigrams) as features for classification tasks.

- Accuracy: 97.06%

Cross Folds: 5 Folds

- Each fold test set 600
- Mean Accuracy 96.7%

Confusion Matrix: Cross-validation was performed using five-folds, a common technique for assessing the model's generalization performance.

	Ham	Spam
Ham.	46.7%.	3.3%
Spam.	0.0%	50.0%

Ham:
Precision: 93.5% Recall 99.9%. F1=96.6%

Spam:
Precision: 99.9% Recall 93.9%. F1=96.8%

Stop words Removal Approach

Removed Stop words Approach: stop words are often considered noise in text data and removing them can lead to improved classification performance.

- Accuracy: 96.67%
- Mean Accuracy (5 folds): 97.03%

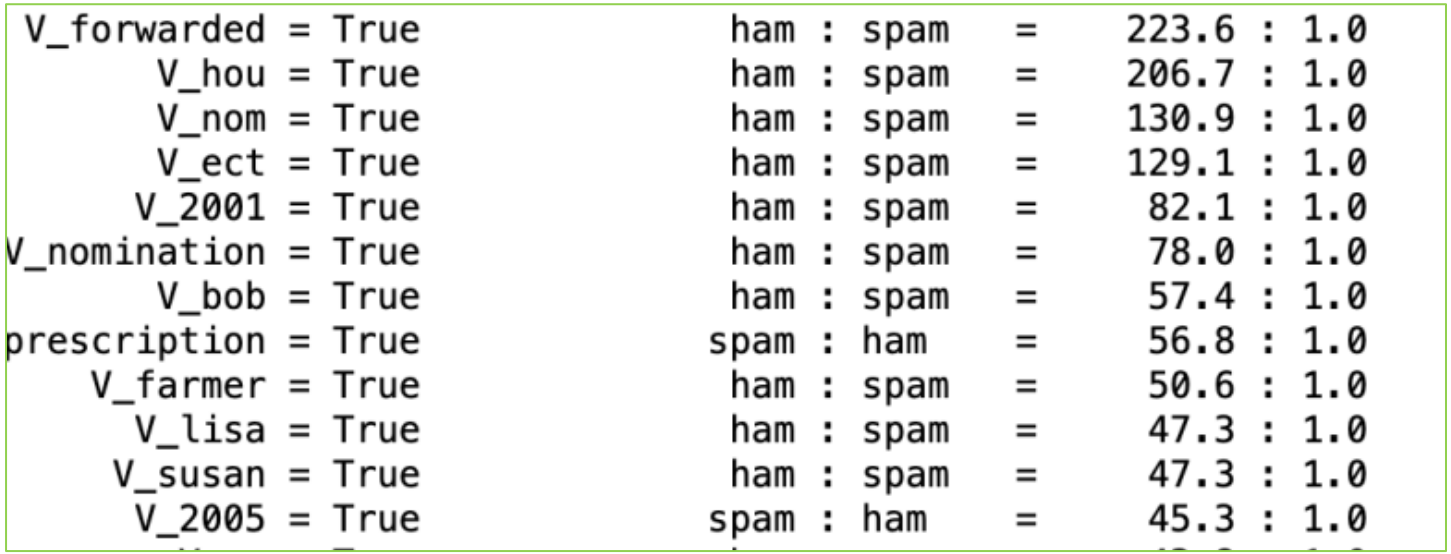
Confusion Matrix: Cross-validation was performed using five-folds, a common technique for assessing the model's generalization performance.

	Ham	Spam
Ham.	47.2%.	2.8%
Spam.	0.1%	49.90%

Ham:
Precision: 94.3% Recall 99.7%. F1=97.00%

Spam:
Precision: 99.7% Recall 94.6%. F1=97.10%

Most Informative Features:



```
 V_forwarded = True              ham : spam   =    223.6 : 1.0
       V_hou = True              ham : spam   =    206.7 : 1.0
       V_nom = True              ham : spam   =    130.9 : 1.0
       V_ect = True              ham : spam   =    129.1 : 1.0
      V_2001 = True              ham : spam   =     82.1 : 1.0
V_nomination = True              ham : spam   =     78.0 : 1.0
       V_bob = True              ham : spam   =     57.4 : 1.0
prescription = True             spam : ham    =     56.8 : 1.0
    V_farmer = True              ham : spam   =     50.6 : 1.0
      V_lisa = True              ham : spam   =     47.3 : 1.0
     V_susan = True              ham : spam   =     47.3 : 1.0
      V_2005 = True             spam : ham    =     45.3 : 1.0
```

Logistic Regression Classifier Performance

Logistic Regression is a linear classifier used for binary classification tasks.

Optimized GridSearchCV Results:

- Accuracy: 98%
- Precision
  - Ham: 99%, Spam: 97%
- Recall
  - Ham: 97%, Spam: 99%
- F1-Score
  - Ham: 98%, Spam: 98%

Best Parameters Found

- {'max_iter': 100, 'tol': 0.01}

Test set accuracy: 0.98

	precision	recall	f1-score	support
ham	0.99	0.97	0.98	291
spam	0.97	0.99	0.98	309
accuracy			0.98	600
macro avg	0.98	0.98	0.98	600
weighted avg	0.98	0.98	0.98	600

Random Forest Classifier Performance

Random Forest is an ensemble method of decision trees, highlighting its approach to reduce overfitting by averaging multiple decision trees.

Optimized GridSearchCV Results:

- Accuracy: 97%
- Precision
  - Ham: 99%, Spam: 94%
- Recall
  - Ham: 94%, Spam: 99%
- F1-Score
  - Ham: 96%, Spam: 97%

Best Parameters Found

- {'criterion': 'gini', 'max_features': 'sqrt', 'min_samples_split': 0.1, 'n_estimators': 500}

Test set accuracy: 0.9666666666666667

	precision	recall	f1-score	support
ham	0.99	0.94	0.96	291
spam	0.94	0.99	0.97	309
accuracy			0.97	600
macro avg	0.97	0.97	0.97	600
weighted avg	0.97	0.97	0.97	600

Gradient Boosting Classifier Performance

Gradient Boosting is an ensemble method that builds strong classifiers from a sequence of weak classifiers.

Optimized GridSearchCV Results:

- Accuracy: 98%
- Precision
  - Ham: 99%, Spam: 97%
- Recall
  - Ham: 97%, Spam: 99%
- F1-Score
  - Ham: 98%, Spam: 98%

Best Parameters Found

- {'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 300, 'subsample': 0.8}

```
Test set accuracy:  0.9816666666666667
              precision    recall  f1-score   support

         ham       0.99      0.97      0.98       291
        spam       0.97      0.99      0.98       309

    accuracy                           0.98       600
   macro avg       0.98      0.98      0.98       600
weighted avg       0.98      0.98      0.98       600
```

Support Vector Machine Classifier Performance

Support Vector Machines are suitable for high-dimensional text classification.

Optimized GridSearchCV Results:

- Accuracy: 98%
- Precision
  - Ham: 99%, Spam: 97%
- Recall
  - Ham: 96%, Spam: 99%
- F1-Score
  - Ham: 98%, Spam: 98%

Best Parameters Found

- {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}

Test set accuracy: 0.9783333333333334

	precision	recall	f1-score	support
ham	0.99	0.96	0.98	291
spam	0.97	0.99	0.98	309
accuracy			0.98	600
macro avg	0.98	0.98	0.98	600
weighted avg	0.98	0.98	0.98	600

Key Findings and Insights

Advanced feature sets significantly improve classifier performance over the baseline model.

Feature combination strategy yields the best results, underscoring the importance of both content and context in the domain of SPAM detection in emails.

Gradient Boost, optimized through the removal of stop words and GridSearch, outperformed all other classifiers... but in some of our runs, we experienced near perfect precision on ham emails with SVM

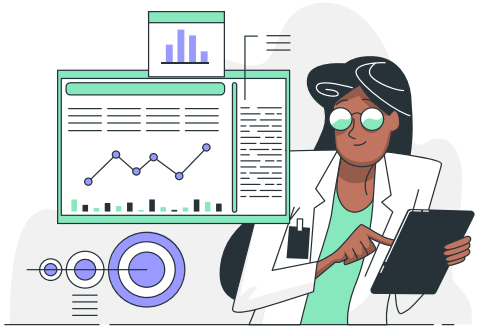
Challenges and Solutions

Challenge 1:

- High dimensionality of feature space led to computational inefficiency, especially with GridSearch optimization.
- We implemented feature selection to focus on the most informative attributes, reducing the model's complexity and reducing run time / training time.

Challenge 2:

- Classifiers were overfitting with highly specific SPAM indicators.
- We utilized cross-validation and regularization techniques to ensure model robustness.



Conclusion

Successfully developed a SPAM email classifier with enhanced accuracy through advanced feature engineering.

Future steps would include exploring deep learning techniques for automatic feature extraction and further model improvements such as accuracy.

The model has potential for real-world application in email filtering to reduce SPAM threats and SPAM email impact.

Thanks!

Do you have any questions?

Arti Ravi Garg : aravigar@syr.edu
Ryan Richardson : ryrichar@syr.edu
Nolan Arendt : nnarendt@syr.edu

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**