Spam or Not Spam: An Email Classification Deep Dive

This documentation outlines the development and evaluation of a robust spam email classifier. We explore various machine learning models and their efficacy in distinguishing legitimate emails from unsolicited messages, providing a comprehensive guide for data scientists and ML engineers.

Project Overview: Building a Reliable Spam Classifier

Objective

To develop a highly accurate spam email classifier using the Spambase dataset, comparing the performance of multiple machine learning algorithms.

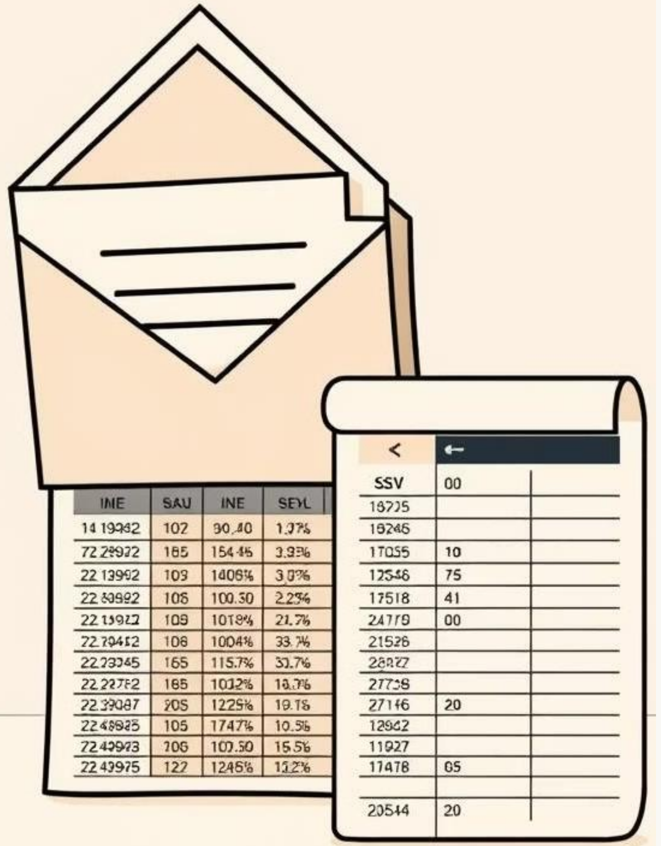
Dataset

The UCI Spambase dataset provides email messages labelled as spam (1) or not spam (0), with numerical features derived from word and character frequencies.

Approach

Training and evaluating a diverse suite of ML models to identify the most effective solution for spam detection.

This project aims to leverage established machine learning techniques to address the persistent challenge of spam email, enhancing digital communication security and user experience.



The Spambase Dataset: Understanding Our Data

Source & Structure

The dataset originates from the UCI Machine Learning Repository (spambase.csv), containing 4601 samples and 58 features.

Target Variable

A binary classification problem, where 'spam' is the target variable (1 for spam, 0 for legitimate).

Feature Details

Features include frequencies of specific words and characters within email bodies, alongside other statistical attributes like capitalisation patterns.

A thorough understanding of this dataset is crucial for building a classifier that accurately captures the nuances of spam characteristics.

Rigorous Data Preprocessing Pipeline

01

Initial Data Inspection

Checked dataset structure using `df.info()` and `df.shape` to understand dimensions and data types.

02

Missing Value Verification

Ensured data integrity by confirming the absence of missing values with `df.isnull().sum()`.

03

Feature Correlation Analysis

Generated a correlation matrix to pinpoint features most strongly associated with the 'spam' label, aiding in feature selection.

04

Feature Selection & Scaling

Selected the top 32 most impactful features and applied `StandardScaler` for robust normalisation.

05

Dataset Splitting

Divided the dataset into an 80% training set and a 20% testing set to ensure unbiased model evaluation.

This systematic preprocessing ensures the data is clean, relevant, and prepared for optimal model training and evaluation.

Feature Extraction: Identifying Key Spam Indicators

Targeted Feature Selection

From the original 58 features, we meticulously extracted the 32 features demonstrating the highest correlation with the 'spam' label.

Constructing Data Matrices

The selected features formed the feature matrix `X`, while the 'spam' labels constituted the target vector `y`.

Standardisation for Consistency

All features were standardised using `StandardScaler`, ensuring consistent scaling across all classifiers and preventing bias towards features with larger numerical ranges.

This focused approach to feature extraction ensures that our models are trained on the most informative aspects of the email data, leading to more accurate spam detection.

Comprehensive Model Implementation: A Comparative Study

- Classical Machine Learning
  - Logistic Regression
  - Support Vector Machine (SVC)
  - K-Nearest Neighbors (KNN)
  - Decision Tree
  - Random Forest
  - Naive Bayes (GaussianNB)
  - XGBoost
  - AdaBoost
  - Gradient Boosting
- Ensemble & Deep Learning
  - Stacking Classifier (combining multiple models)
  - Neural Network (implemented via TensorFlow/Keras)

This diverse array of models provides a robust framework for comparing algorithmic strengths and weaknesses in spam classification.

Model Evaluation: Key Performance Metrics

→ **Accuracy**

Overall correctness of predictions.

→ **Precision**

Proportion of correctly identified spam emails among all positive predictions.

→ **Recall**

Proportion of actual spam emails correctly identified.

→ **F1-Score**

Harmonic mean of precision and recall, providing a balanced measure.

→ **Confusion Matrix**

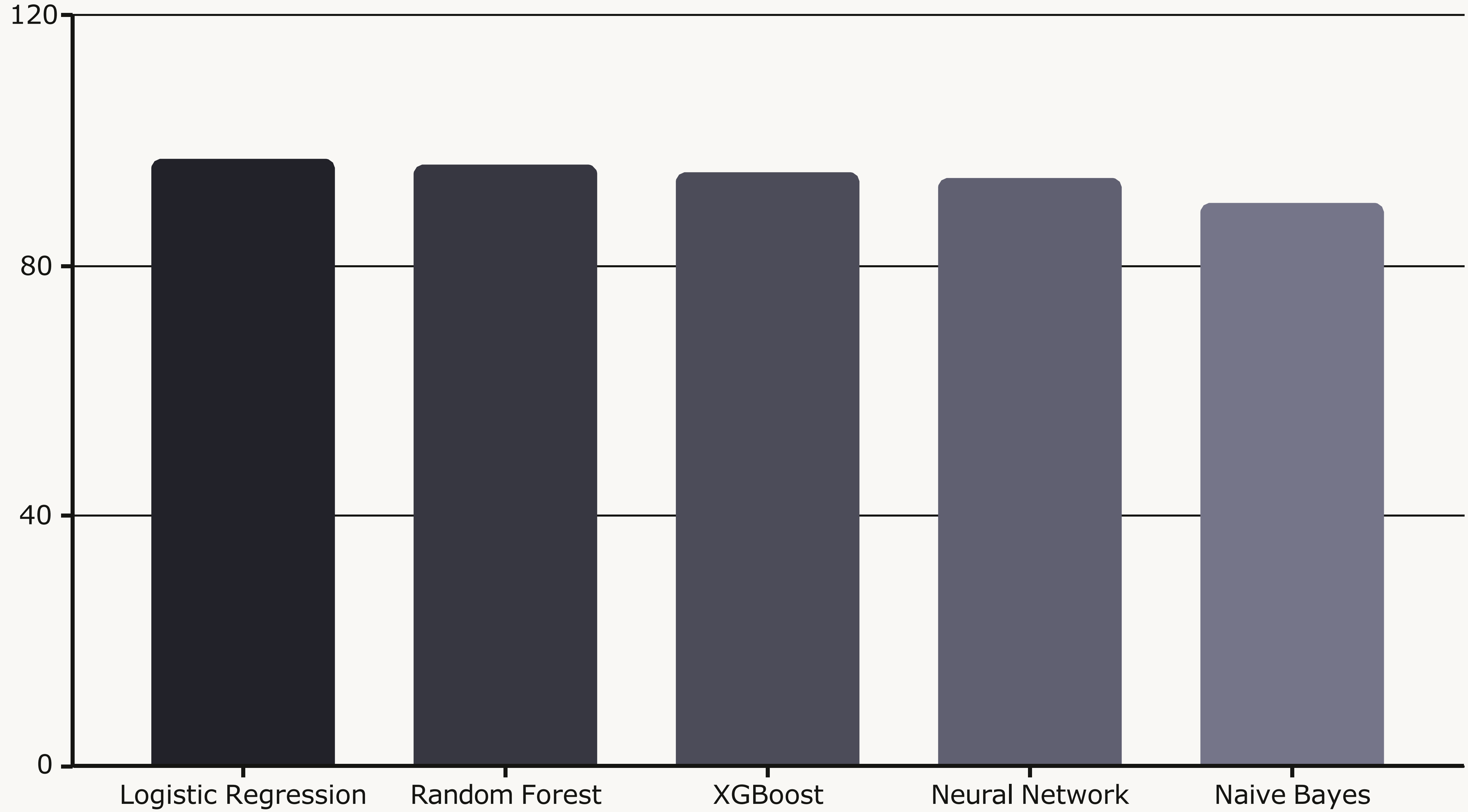
Detailed breakdown of true positives, true negatives, false positives, and false negatives.

→ **Cross-validation**

Stratified K-Fold used to ensure robust and generalisable performance estimates.

These metrics collectively offer a comprehensive view of each model's effectiveness and reliability in classifying spam.

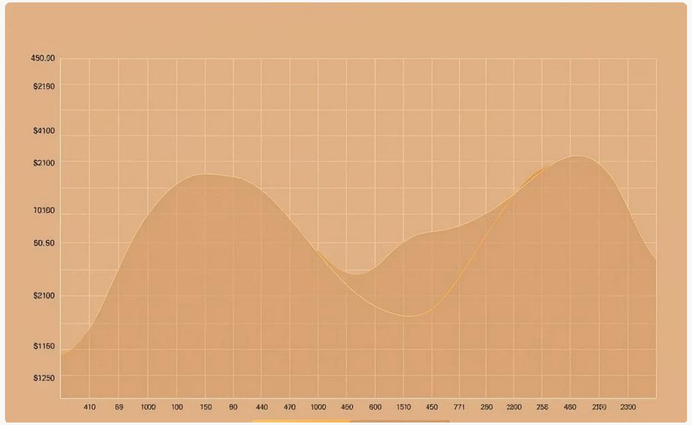
Results: Top Performers and Insights



- **Best Models:** Logistic Regression, Random Forest, and XGBoost consistently achieved the highest accuracy, ranging from ~95–97%.
- **Naive Bayes:** Showed decent performance, though slightly lower than the top models.
- **Neural Network:** Delivered comparable accuracy but required significantly more computational resources.
- **Confusion Matrix:** All top models demonstrated strong classification, minimising false negatives (critical for not missing spam).

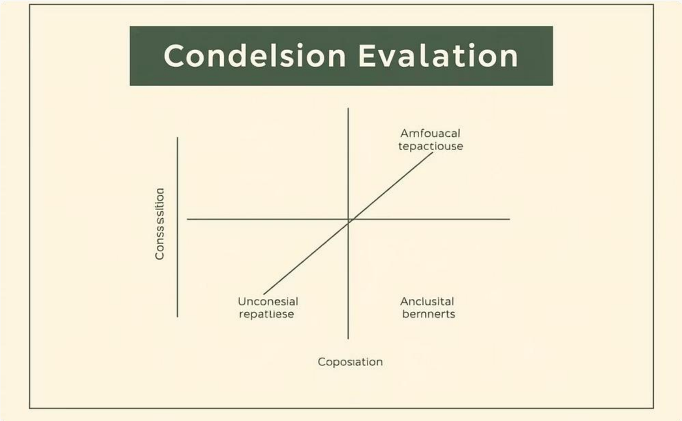
These results highlight the efficiency of classical ML approaches for this specific classification task.

Visualisations: Unveiling Data Patterns



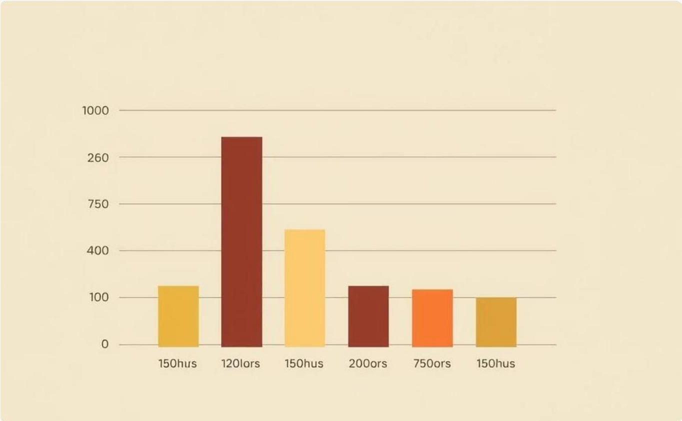
Correlation Heatmap

Clearly depicted relationships between features and the spam label, guiding feature engineering and selection.



Confusion Matrices

Provided detailed visual comparisons of misclassifications for each model, aiding in error analysis.



Accuracy Comparison Bar Chart

Offered an immediate visual understanding of each model's performance relative to others.

These visualisations were instrumental in interpreting model performance and understanding underlying data structures.

Conclusion & Future Work: Advancing Spam Detection

1

Key Conclusion

Classical ML models such as Logistic Regression, Random Forest, and XGBoost proved highly effective for spam detection, often outperforming deep learning approaches for this dataset.

2

Advanced Deep Learning

Experiment with more sophisticated deep learning architectures, including Recurrent Neural Networks (RNNs) and Transformer models, for potential improvements.

3

Real-world Deployment

Develop and integrate the classifier into a web application or email filtering system for practical usage.

4

Dataset Expansion

Enhance the dataset with more contemporary spam examples to ensure the model remains robust against evolving spam tactics.

The current models offer a strong foundation, but continuous improvement and adaptation are essential for long-term effectiveness in the dynamic landscape of email spam.