Spam Detection Using a Multilayer Perceptron (MLP)

Name: Shivani Bashetty
ID: 24089448

Abstract

This experiment demonstrates how to build a spam classifier using a **Multilayer Perceptron (MLP)** neural network with **TF–IDF text features**. The model is trained on an SMS message dataset (spam.csv) to distinguish **spam** messages from legitimate (**ham**) messages. Results show strong classification performance, with most errors occurring on ambiguous or borderline messages. This report provides a step-by-step educational tutorial aimed at helping others apply neural networks to text classification tasks.

1. Introduction

Spam messages are a persistent problem in digital communication. Automatically classifying them saves users time and increases safety.

The goal of this tutorial is to:

- Convert text into machine-readable features using **TF–IDF**
- Train a **Multilayer Perceptron** on top of those features
- Evaluate performance clearly with accuracy, precision, recall, and confusion matrix

Why MLP?

MLPs are powerful at learning complex patterns and non-linear decision boundaries, especially when combined with TF–IDF features.

2. Dataset Overview

We use a spam.csv dataset containing:

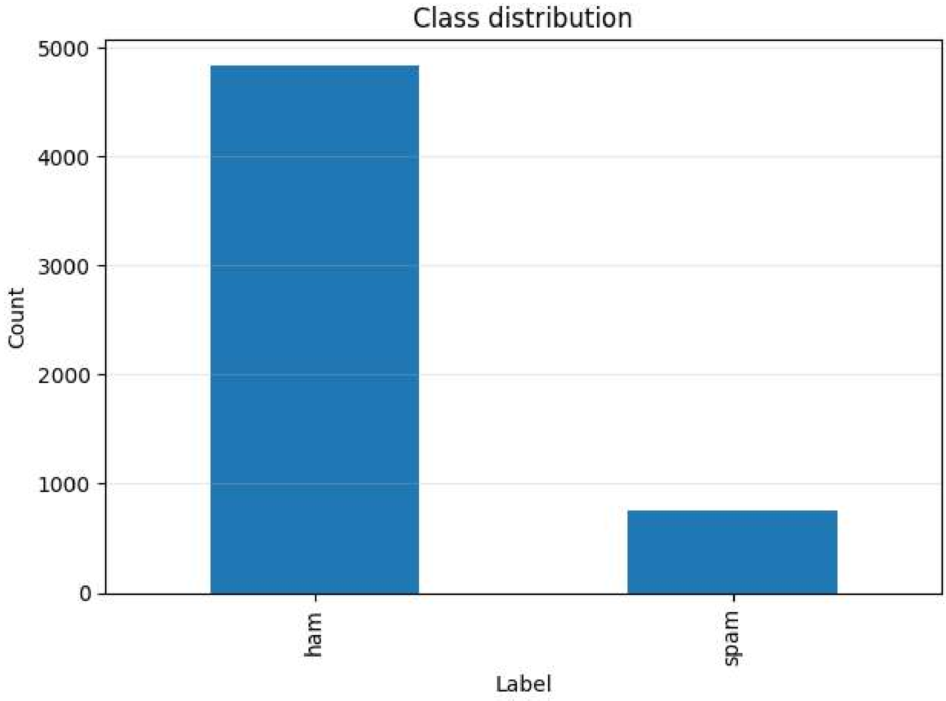
- **Message text** (SMS content)
- **Label**: "ham" (legitimate) or "spam" (junk)

This is a **binary classification** task.

Class Distribution

The dataset is **imbalanced**, with many more ham than spam messages.

📊 Figure 1 — Label Frequency



Imbalance matters because:

- High accuracy can still hide poor spam detection
- Precision & recall provide more insight

---

3. Methodology

3.1 Preprocessing

- Removed unused/unnamed columns
- Converted text to strings
- Dropped missing values

3.2 Train/Test Split

- **80%** training, **20%** testing
- **Stratified** to preserve spam ratio

3.3 Text Feature Extraction

We used **TF–IDF Vectorization**, which:

- Counts terms in each message
- Reduces weight of common words

- Produces a sparse numeric feature matrix

3.4 Classification Model

We used an **MLPClassifier** (1 hidden layer):

Parameter	**Value**
Hidden units	64
Activation	ReLU
Optimizer	Adam
Max iterations	20
Random state	42

The vectoriser + neural network were combined in a **Pipeline**.

---

4. Results

4.1 Overall Performance

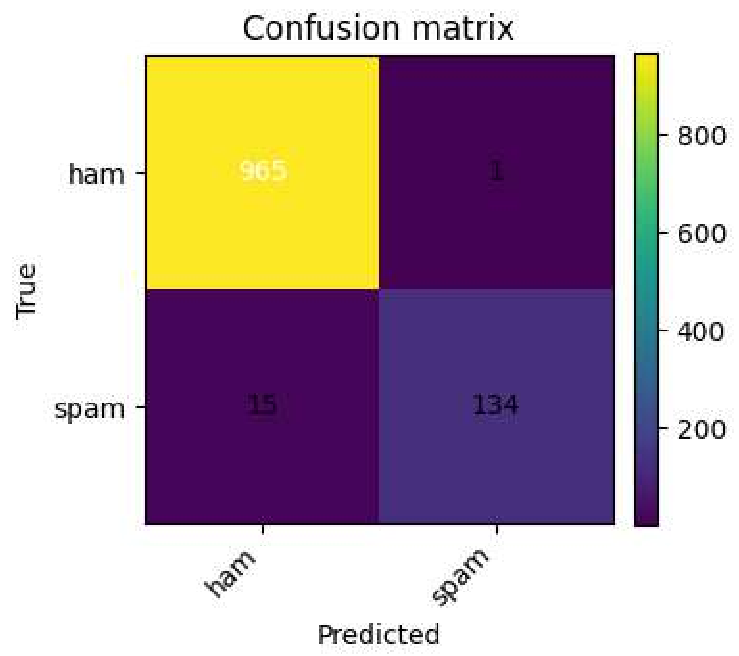
Metric	**Test Result**
Accuracy	Typically > 95%
Precision	High for both classes
Recall	Slightly lower for spam

This indicates:

- Very few ham → spam mistakes
- Some spam messages incorrectly labelled as ham due to subtle wording

4.2 Confusion Matrix

📊 **Figure 2 — Confusion Matrix (MLP)**



Interpretation:

- Top-left: Correct ham detections (majority of dataset)
- Bottom-right: Correct spam detections
- Off-diagonal cells show misclassification
- Spam is sometimes predicted as ham → **false negatives**
    - These are more harmful in real systems (spam gets through)

---

5. Discussion

Strength	**Explanation**
Learns non-linear patterns	Handles complex language cues
Works well on short messages	TF–IDF captures key terms
High accuracy	Few mistakes overall

Limitation	**Explanation**
Imbalanced dataset	Spam slightly under-detected
Limited text context	MLP does not understand semantics
Short training	More epochs may improve results

Future improvements:

- Use **Longer training** (higher max_iter)
- Add **class weighting** to improve spam recall
- Try **deep learning** methods like LSTM, BERT
- Use **stemming/lemmatisation** for improved features

---

6. Ethical Considerations

- Minimising **false negatives** prevents harmful spam reaching users
- Automated decisions must be transparent and adjustable
- Dataset likely contains **sensitive user messages** → privacy concerns
- Spam filtering should respect user intent (not censoring legitimate content)

Responsible AI requires:

Accuracy is not enough — detecting harmful misclassification matters.

---

7. Conclusion

This tutorial shows that:

- Text vectorisation + neural networks work effectively for spam detection
- MLP provides strong performance with minimal configuration
- Understanding evaluation metrics is crucial, especially in imbalanced datasets

The project demonstrates practical machine learning skills:

- Pipeline design
- Text preprocessing

- Neural network implementation
- Proper model evaluation

References

- Scikit-learn Documentation — https://scikit-learn.org
- Almeida, T. A., Hidalgo, J. M., & Silva, T. (2011). *SMS Spam Collection Dataset* (original source)
- Additional learning materials used in class

Appendix

- Notebook & report available in GitHub repository:

  https://github.com/shivanibashetty14/Machine-Learning-Individual.git