PLP FEBRUARY 2025 COHORT:
AI FOR SOFTWARE ENGINEERING

SPAM DETECTOR PROJECT REPORT

2025

COMPILED BY
MIKE, VIOLET,
LUKHANYO &
TSHIMO

TABLE OF CONTENTS

INTRODUCTION

This project explores the practical and ethical application of AI tools in tackling real-world challenges, as part of the “Mastering the AI Toolkit” assignment.

Our team set out to design and implement a machine learning model to classify and predict outcomes related to a critical social issue. In this case, we focused on spam detection, an important use case for improving communication quality and ensuring user safety.

Throughout this assignment, we applied a combination of established AI frameworks — including Scikit-learn and spaCy — to train, evaluate, and optimize a predictive model.

Our report outlines the theoretical underpinnings of these tools, presents the implementation and model outputs, and discusses ethical considerations such as data privacy, fairness, and sustainability.

Finally, we reflect on our collaborative process and lessons learned as a team, illustrating both the technical and human-centered skills required for responsible AI development.

THEORETICAL QUESTIONS

01 — What is the difference between TensorFlow and PyTorch?

- TensorFlow is developed by Google and offers high performance with strong production support (e.g., TFX, TensorFlow Lite).
- PyTorch, developed by Facebook, is more Pythonic and intuitive, widely used in research.

Key difference: PyTorch uses dynamic computation graphs (easier to debug), while TensorFlow uses static graphs (better for production deployment).

02 — When would you choose spaCy over Scikit-learn?

- spaCy is ideal for NLP tasks like Named Entity Recognition (NER), POS tagging, and dependency parsing.
- Scikit-learn is better for classical ML tasks like classification, regression, and clustering.

Choose spaCy when working with raw text and language structure.
Choose Scikit-learn when working with structured/tabular data.

03 — Define: Transfer Learning and Model Optimization

- Transfer Learning: Reusing a pre-trained model on a new, similar task to save training time and improve performance.
- Model Optimization: Techniques like pruning, quantization, and early stopping to improve model efficiency and reduce size.

THEORETICAL QUESTIONS CONTD.

04 — Pros & Cons of Jupyter Notebook vs Google Colab

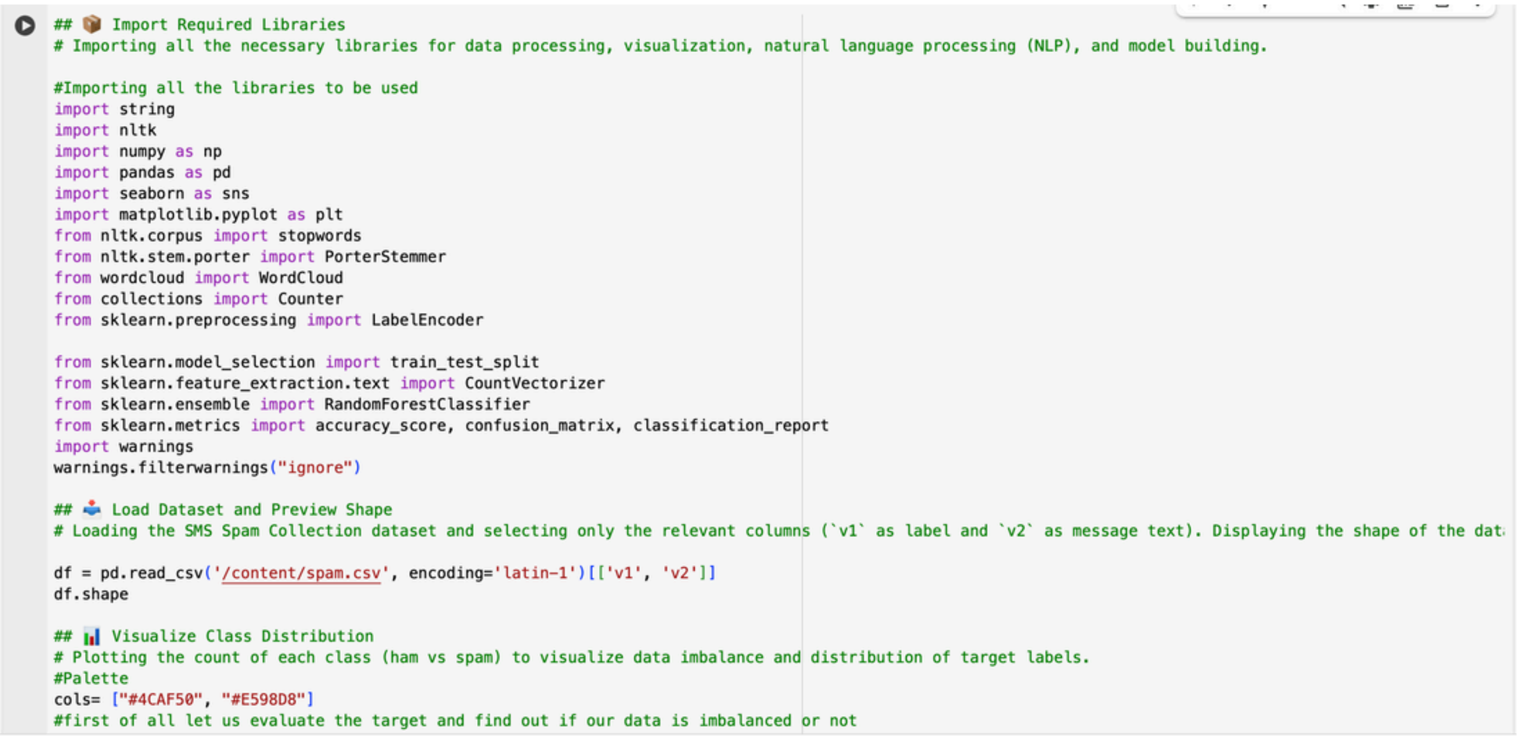
FEATURE	JUPYTER NOTEBOOK	GOOGLE COLAB
SETUP	Local setup required	No setup, runs in browser
COMPUTE POWER	Limited to your PC	Free GPU/TPU available
SHARING	Manual file sharing	Easy sharing via links
STORAGE	Local files	Google Drive integration

05 — What are APIs in AI Tools?

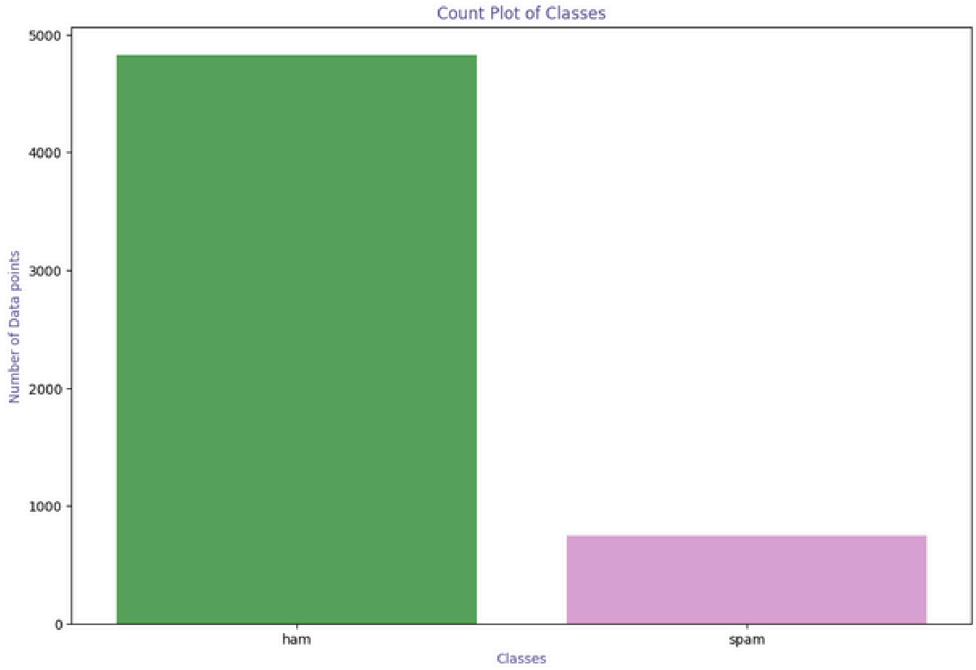
- APIs (Application Programming Interfaces) in AI allow developers to access pre-trained models or services (e.g., Google Cloud Vision, OpenAI) without rebuilding models from scratch. They help integrate AI into apps quickly.

SCREENSHOTS & OUTPUTS

Import data set and run all code for output as follows:

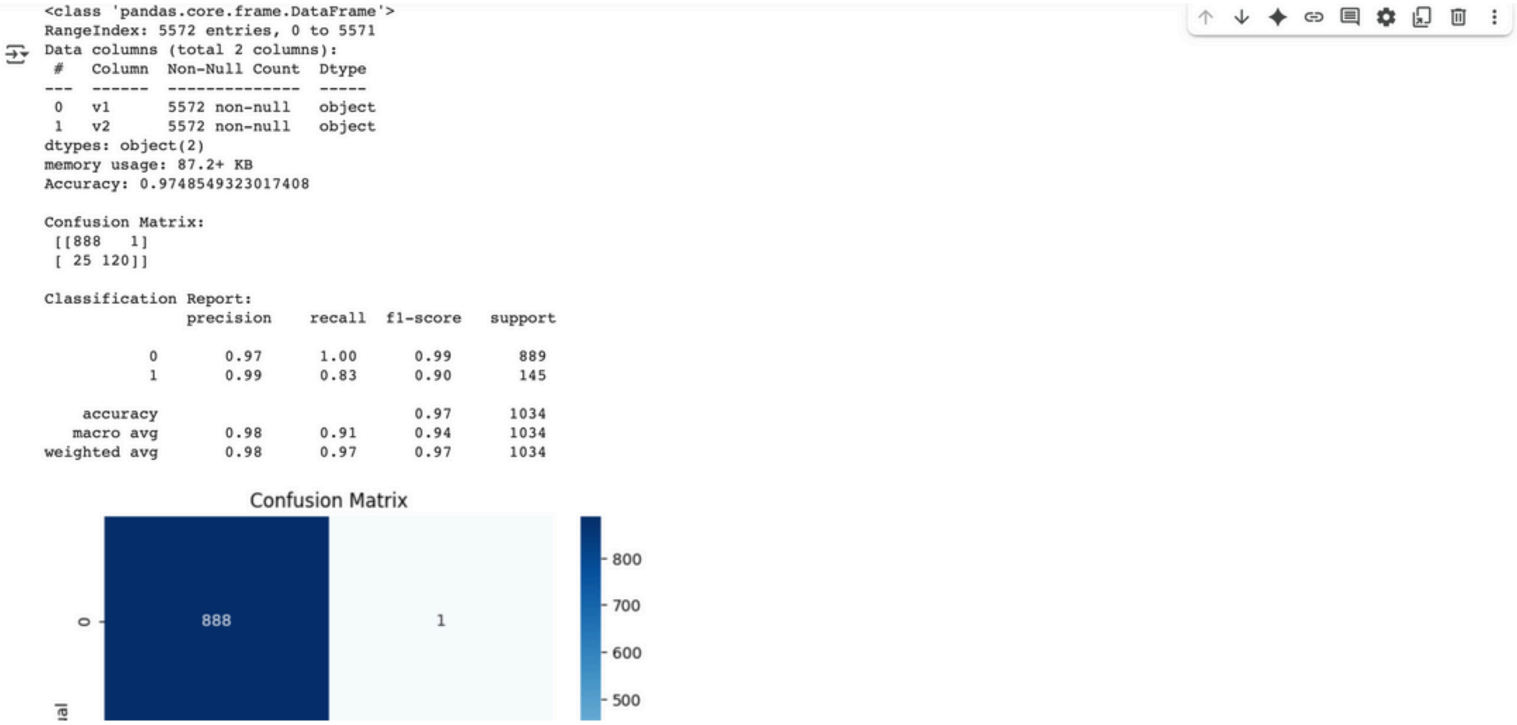


```
## 📦 Import Required Libraries
# Importing all the necessary libraries for data processing, visualization, natural language processing (NLP), and model building.

#Importing all the libraries to be used
import string
import nltk
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from wordcloud import WordCloud
from collections import Counter
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import warnings
warnings.filterwarnings("ignore")

## 📥 Load Dataset and Preview Shape
# Loading the SMS Spam Collection dataset and selecting only the relevant columns (`v1` as label and `v2` as message text). Displaying the shape of the dat

df = pd.read_csv('/content/spam.csv', encoding='latin-1')[['v1', 'v2']]
df.shape

## 📊 Visualize Class Distribution
# Plotting the count of each class (ham vs spam) to visualize data imbalance and distribution of target labels.
#Palette
cols= ["#4CAF50", "#E598D8"]
#first of all let us evaluate the target and find out if our data is imbalanced or not
```



SCREENSHOTS & OUTPUTS CONTD.

Import data set and run all code for output as follows:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
Accuracy: 0.9748549323017408

Confusion Matrix:
 [[888   1]
 [ 25 120]]

Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       889
           1       0.99      0.83      0.90       145

    accuracy                           0.97      1034
   macro avg       0.98      0.91      0.94      1034
weighted avg       0.98      0.97      0.97      1034
```

ETHICAL REFLECTION

01 —Bias in Datasets
AI models can inherit bias from historical data. For example, a spam classifier might wrongly flag messages from certain dialects or regions. This reduces fairness and user trust.

02 —Fairness and Inclusion
We must ensure the model does not disproportionately misclassify underrepresented groups. Including diverse data and applying fairness constraints helps.

03 — Privacy Concerns
Spam and text classification may involve personal communication. Data must be anonymized and handled with care to respect privacy.

04 — Environmental Consideration
Training large models consumes significant energy. Using optimized architectures and transfer learning can reduce the environmental impact.

05 — Define: Transfer Learning and Model Optimization
AI should support human decision-making, not replace it blindly. Users must understand model limitations and apply human judgment in critical cases.

CONCLUSION

This assignment provided our team with a hands-on opportunity to explore and apply modern AI tools in a structured, real-world context. From selecting the appropriate frameworks to implementing and evaluating our model, we gained valuable insights into the strengths and trade-offs of tools like Scikit-learn and spaCy.

Beyond technical implementation, we deepened our understanding of the ethical responsibilities that come with building AI systems — including fairness, transparency, privacy, and sustainability. These considerations are not just theoretical but essential to developing trustworthy and socially beneficial AI solutions.

Working collaboratively allowed us to combine our individual strengths, practice version control, and effectively communicate our findings. This project has strengthened both our technical foundation and our ability to think critically about the broader implications of AI in society.