



Assessment Report

on

“Detect Spam Emails”

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY
DEGREE**

SESSION 2024-25

in

INTRODUCTION TO AI

By

ANKIT JAISWAL (202401100400037, CSE-AIML A)

Under the supervision of

“MR.BIKKI GUPTA”

KIET Group of Institutions, Ghaziabad

Introduction

Spam emails are unwanted or unsolicited messages that clutter inboxes and pose security risks.

This project focuses on classifying emails as *spam* or *not spam* using machine learning techniques.

We utilize structured metadata features (like word frequency, presence of links, etc.) from emails.

The dataset is processed and fed into a supervised learning algorithm (Random Forest Classifier).

The model is trained on a portion of the data and tested on the rest to evaluate its performance.

We compute important evaluation metrics such as accuracy, precision, and recall.

A confusion matrix heatmap is generated to visualize classification performance.

This helps identify how well the model distinguishes between spam and legitimate emails.

Such automated systems can improve email filtering and reduce risks of phishing attacks.

The project demonstrates the practical application of AI in cybersecurity and communication.

PROBLEM STATEMENT

The project aims to classify emails as spam or not spam using machine learning.

It uses structured metadata features to train a classification model.

The goal is to automate spam detection and evaluate the model's accuracy and effectiveness.

Methodology

- **Data Collection:** A CSV dataset containing email metadata and spam labels is used.
- **Data Preprocessing:** Handle missing values and encode categorical variables if present.
- **Feature Selection:** Separate input features (X) and target label (`is_spam`) for modeling.
- **Model Training:** Use a Random Forest Classifier to learn patterns from the training data.
- **Evaluation:** Test the model using accuracy, precision, recall, and confusion matrix.
- **Visualization:** Generate a heatmap to clearly represent classification performance.

CODE

```
from google.colab import files
uploaded = files.upload()

# Use the correct file name
import pandas as pd
file_path = "spam_emails.csv"
data = pd.read_csv(file_path)
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import
confusion_matrix, accuracy_score,
precision_score, recall_score
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import
LabelEncoder
```

```
# -----  
# STEP 1: Load Your CSV File  
# -----  
# Make sure your CSV is in the same  
# directory or provide full path  
file_path = "spam_emails.csv" # <--  
# Replace with your actual filename  
data = pd.read_csv(file_path)  
  
# -----  
# STEP 2: Preprocessing (if needed)  
# -----  
# Check if there are any categorical  
# columns to encode  
for col in  
data.select_dtypes(include=['object'])  
.columns:  
    le = LabelEncoder()  
    data[col] =  
le.fit_transform(data[col])
```

```
# Ensure the label column is named
'is_spam' (you can change this below
if needed)
label_column = 'is_spam'
X = data.drop(label_column, axis=1)
y = data[label_column]

# -----
# STEP 3: Split Data
# -----
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=42)

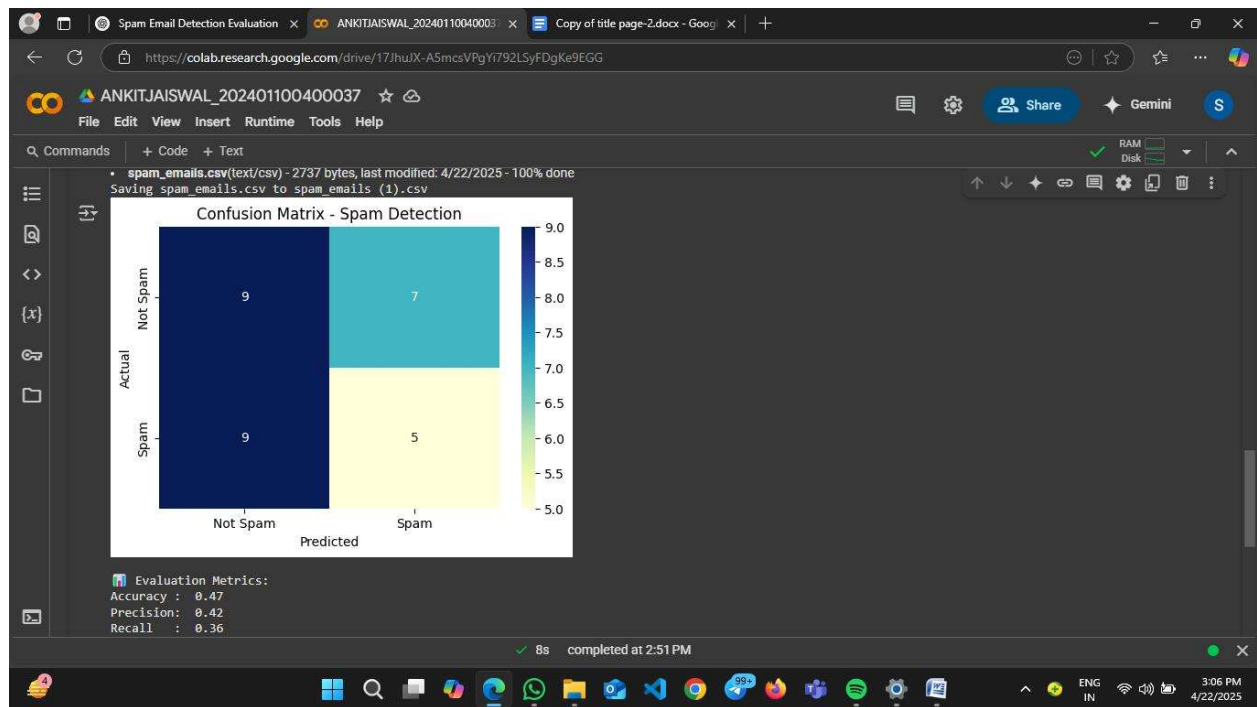
# -----
# STEP 4: Train the Classifier
# -----
model =
RandomForestClassifier(n_estimators=1
00, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```



```
# -----  
# STEP 5: Evaluation Metrics  
# -----  
cm = confusion_matrix(y_test, y_pred)  
accuracy = accuracy_score(y_test,  
y_pred)  
precision = precision_score(y_test,  
y_pred)  
recall = recall_score(y_test, y_pred)  
  
# -----  
# STEP 6: Heatmap  
# -----  
plt.figure(figsize=(6, 4))  
sns.heatmap(cm, annot=True, fmt='d',  
cmap='YlGnBu', xticklabels=['Not  
Spam', 'Spam'], yticklabels=['Not  
Spam', 'Spam'])  
plt.xlabel('Predicted')  
plt.ylabel('Actual')
```

```
plt.title('Confusion Matrix - Spam  
Detection')  
plt.show()  
  
# -----  
# STEP 7: Print Evaluation Metrics  
# -----  
print(f"\n  Evaluation Metrics:")  
print(f"Accuracy :   {accuracy:.2f}")  
print(f"Precision:   {precision:.2f}")  
print(f"Recall      :   {recall:.2f}")
```

OUTPUT



References and Credits

1. Scikit-learn Documentation – <https://scikit-learn.org>
2. Pandas Documentation – <https://pandas.pydata.org>
3. Matplotlib and Seaborn Libraries – For data visualization
4. Dataset Source – Provided/uploaded CSV file for spam email classification
5. Project developed using Google Colab and Python

