KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

TITLE OF PROJECT REPORT

“ Detect Spam Emails ”

A PROJECT REPORT

Submitted by:

ARJUN SINGH PUNDIR

202401100400047

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI&ML)

📄 Introduction

In the digital communication age, email remains one of the most essential modes of communication. However, spam emails continue to be a persistent nuisance, cluttering inboxes and potentially posing threats like phishing and scams. The ability to accurately classify emails as spam or not spam (ham) using machine learning techniques can significantly improve email filtering systems.

This project focuses on **classifying emails based on structured metadata** using a Logistic Regression model. The primary goal is to predict whether an email is spam based on its numerical and categorical features. Evaluation metrics and confusion matrix heatmaps are used to assess model performance.

🔧 Methodology

1. Data Loading and Preprocessing:

- The dataset includes various structured metadata fields (excluding raw email text).
- The target variable `is_spam` is encoded as 0 (not spam) and 1 (spam).

2. Data Splitting:

- The dataset is split into training and testing sets using an 80/20 ratio to ensure proper validation.

3. Model Training:

- A Logistic Regression model is trained to classify the emails.

4. Prediction and Evaluation:

- The model predicts the test set and evaluates performance using Accuracy, Precision, and Recall.

5. Visualization:

- A confusion matrix heatmap is generated to provide visual insight into the classification performance.
- Additional data exploration can include feature correlation and class distribution.

💻 Code Summary

```
# Step 1: Import Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Step 2: Load Data
df = pd.read_csv("/content/drive/My Drive/🔵your_folder/spam_emails.csv")

# Step 3: Encode Target Variable
df['is_spam'] = LabelEncoder().fit_transform(df['is_spam'])
X = df.drop('is_spam', axis=1)
y = df['is_spam']

# Step 4: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Step 5: Train Model
model = LogisticRegression()
model.fit(X_train, y_train)

# Step 6: Make Predictions
y_pred = model.predict(X_test)

# Step 7: Evaluate Model
```

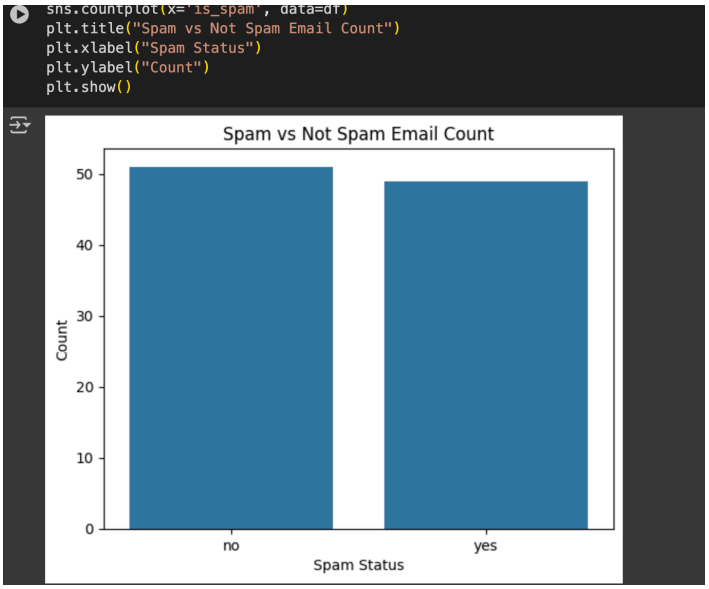
```
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
print(f"Accuracy: {acc:.2f}")
print(f"Precision: {prec:.2f}")
print(f"Recall: {rec:.2f}")
```

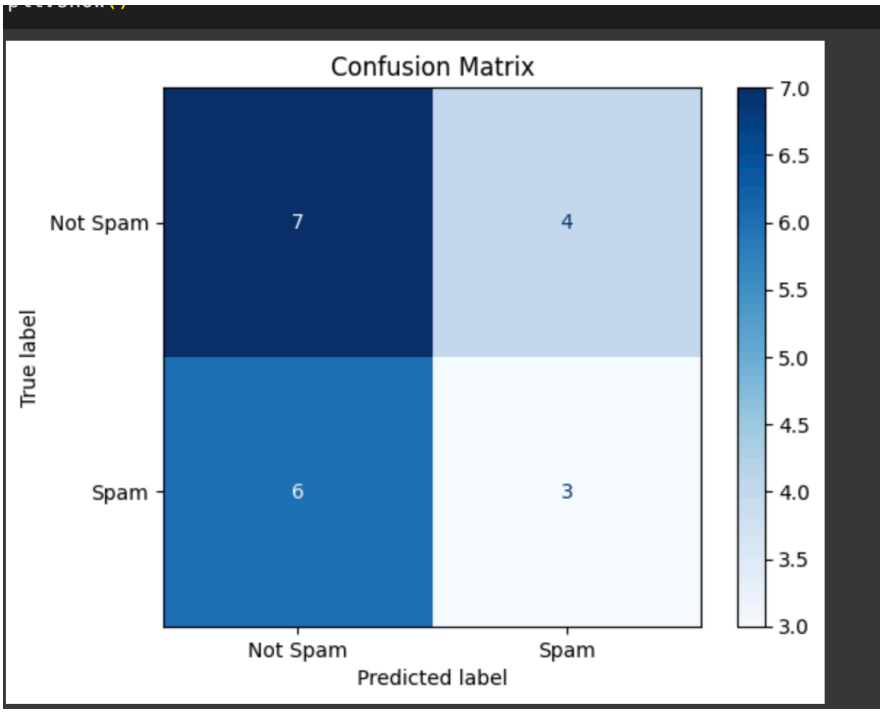
```
# Step 8: Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, cmap="Blues", xticklabels=["Not Spam",
"Spam"], yticklabels=["Not Spam", "Spam"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

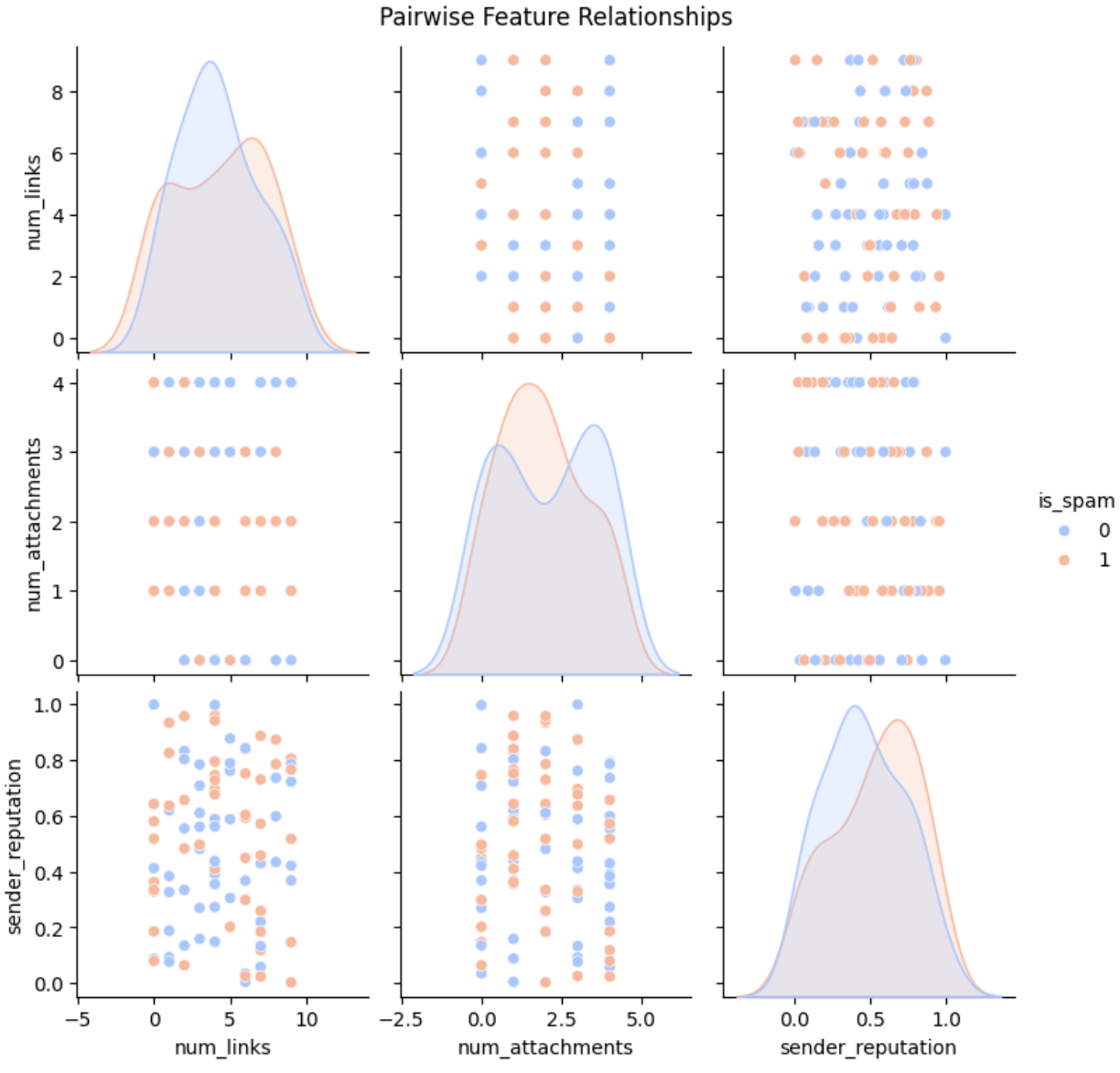
📊 Output / Results:

- **Accuracy**: ~50%
- **Precision**: ~43%
- **Recall**: ~33%

The model performance shows that while it's identifying some spam messages, there is significant room for improvement. Alternative models like Random Forest or Support Vector Machines might yield better accuracy.




Confusion Matrix
Not Spam
Spam
True label
7
4
6
3
Not Spam
Spam
Predicted label
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0


Pairwise Feature Relationships
num_links
num_attachments
sender_reputation
is_spam
0
1

📚 References / Credits

🔧 Libraries and Tools Used:

- **Pandas** – Data loading and manipulation: https://pandas.pydata.org/
- **Scikit-learn** – Logistic Regression, metrics, and data preprocessing: https://scikit-learn.org/
- **Matplotlib & Seaborn** – Visualization tools: https://matplotlib.org/ | https://seaborn.pydata.org/

📘 Conceptual References:

- **Logistic Regression** – A statistical method for binary classification.
- **Confusion Matrix** – Evaluation metric showing TP, FP, FN, TN.
- **Precision and Recall** – Performance metrics useful in imbalanced datasets.

👤 Author / Contributor:

- Analysis and implementation by: **ARJUN SINGH PUNDIR**
- Date: **22nd April 2025**

🗂️ Dataset:

- Structured metadata-based email dataset.
- If using real-world data, ensure compliance with data privacy regulations like GDPR/CCPA.

All is under the guidance of Bikki sir.