Title:

Detecting Spam Emails Using Structured Metadata

KIET GROUP OF INSTITUTIONS

Name: Jahnavi Singh
Roll Number: 202401100300131(57)
Course: Artificial Intelligence
Exam: MSE 2
Platform Used: Google Colab
Tools & Libraries: Python, Scikit-learn, Pandas, RandomForestClassifier

Introduction

Email spam is a significant challenge in digital communication, leading to wasted time, security risks, and data overload. While traditional approaches use text classification, this project focuses on structured metadata for detecting spam. This method mimics the approach used by modern email servers where email logs are analyzed based on patterns and characteristics like number of recipients, attachments, links, etc.

Example metadata fields:

- Number of recipients
- Email length
- Presence of attachments or links
- Time of sending
- Common spam words or flags

Methodology

The steps followed in the project are:

1. **Data Creation**: A sample dataset is created with features such as num_recipients, has_attachment, email_length, num_links, and contains_free_offer.
2. **Preprocessing**: No missing data cleanup needed due to synthetic data.
3. **Feature Selection**: Structured fields are chosen based on relevance to spam detection.
4. **Model Selection**: A **Random Forest Classifier** is selected due to its robustness and good performance on tabular data.
5. **Training and Testing**: The data is split (80% training, 20% testing) to evaluate accuracy.
6. **Evaluation**: Accuracy score and classification report are used to assess model performance.

Code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Step 1: Structured Metadata Sample
data = {
    'num_recipients': [1, 10, 1, 3, 25, 2],
    'has_attachment': [0, 1, 0, 0, 1, 0],
    'email_length': [120, 5000, 300, 150, 4000, 180],
    'num_links': [0, 10, 1, 0, 8, 0],
    'contains_free_offer': [0, 1, 0, 0, 1, 0],
    'is_spam': [0, 1, 0, 0, 1, 0]
}

df = pd.DataFrame(data)

# Step 2: Feature Selection
X = df.drop('is_spam', axis=1)
y = df['is_spam']

# Step 3: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Train the model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Step 5: Prediction and Evaluation
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Output/Result

```
Accuracy: 1.0
          precision    recall  f1-score   support


          0       1.00      1.00      1.00         1

          1       1.00      1.00      1.00         1


    accuracy                          1.00         2

   macro avg       1.00      1.00      1.00         2

weighted avg       1.00      1.00      1.00         2
```

References/Credits

- Scikit-learn Documentation: https://scikit-learn.org/
- Email Spam Dataset ideas: UCI Machine Learning Repository
- Random Forest Algorithm: https://en.wikipedia.org/wiki/Random_forest
- Python Pandas Documentation
- Google Colab Platform