SMS Spam Classification

Name – Manish Meena

Introduction

With the rapid proliferation of mobile communication, the influx of Short Message Service (SMS) messages has become an integral part of our daily lives. However, along with the convenience of SMS comes the nuisance of spam messages. To address this issue, this project focuses on developing a machine learning model using Natural Language Processing (NLP) techniques to automatically classify SMS messages as either SPAM or HAM (normal). The primary goal is to provide users with a tool that efficiently filters and cleans their SMS inbox, enhancing the overall mobile communication experience.

Project Overview

In this project, a robust SMS spam classification model was constructed, leveraging advanced NLP techniques and machine learning algorithms. The classification process relies on analyzing the textual content of SMS messages to distinguish between legitimate and spam messages.

Key Features and Contributions

Feature Engineering: The project employed feature engineering techniques, creating insightful features such as word_count, contains_currency_symbol, and contains_number, to enrich the model's predictive capabilities.

Project Benefits

The significance of this project lies in its potential to streamline and enhance the SMS experience for users. By automatically identifying and segregating spam messages, users can enjoy a clutter-free SMS inbox, thereby saving time and mitigating the annoyance caused by unwanted messages.

Resources Used

Packages

- pandas
- numpy

- sklearn
- matplotlib
- seaborn
- nltk

Dataset

The project utilized the SMS Spam Collection Dataset from UCI Machine Learning, available on Kaggle: SMS Spam Collection Dataset

Project Lifecycle

Exploratory Data Analysis (EDA)

- Explored NaN values in the dataset.
- Plotted a countplot to visualize the distribution of SMS labels (Spam vs. Ham).

Feature Engineering

Addressed the issue of imbalanced datasets using oversampling techniques.

Created new features, such as word_count, contains_currency_symbol, contains_numbers, to improve model performance.

Data Cleaning

1. Removed special characters and numbers using regular expressions.
2. Converted the entire SMS content to lowercase.
3. Tokenized SMS messages into words.
4. Removed stop words.
5. Lemmatized words to simplify and normalize the text.
6. Constructed a corpus of cleaned messages.

Model Building and Evaluation

Metric: F1-Score

1. Multinomial Naive Bayes: 0.943
2. Decision Tree: 0.98
3. Random Forest: 0.994
4. Ensemble (Voting) - Decision Tree + Multinomial Naive Bayes: 0.98

Output

Prediction 1 - Lottery text message

sample_message = 'IMPORTANT - You could be entitled up to £3,160 in compensation from mis-sold PPI on a credit card or loan. Please reply PPI for info or STOP to opt out.'

if predict_spam(sample_message):

print('Gotcha! This is a SPAM message.')

else:

print('This is a HAM (normal) message.')

In [132]:
```
# Prediction 1 - Lottery text message
sample_message = 'IMPORTANT - You could be entitled up to £3,160 in compensation from mis-sold PPI on a credit card or loan.

if predict_spam(sample_message):
  print('Gotcha! This is a SPAM message.')
else:
  print('This is a HAM (normal) message.')
```

```
Gotcha! This is a SPAM message.
```

Prediction 2 - Casual text chat

sample_message = 'Came to think of it. I have never got a spam message before.'

if predict_spam(sample_message):

print('Gotcha! This is a SPAM message.')

else:

print('This is a HAM (normal) message.')

In [133]:
```
# Prediction 2 - Casual text chat
sample_message = 'Came to think of it. I have never got a spam message before.'

if predict_spam(sample_message):
  print('Gotcha! This is a SPAM message.')
else:
  print('This is a HAM (normal) message.')
```

```
This is a HAM (normal) message.
```

Prediction 3 - Transaction confirmation text message

sample_message = 'Sam, your rent payment for Jan 19 has been received. $1,300 will be drafted from your Wells Fargo Account ******0000 within 24-48 business hours. Thank you!'

if predict_spam(sample_message):

print('Gotcha! This is a SPAM message.')

else:

print('This is a HAM (normal) message.')

```
In [134]: # Prediction 3 - Transaction confirmation text message
          sample_message = 'Sam, your rent payment for Jan 19 has been received. $1,300 will be drafted from

          if predict_spam(sample_message):
            print('Gotcha! This is a SPAM message.')
          else:
            print('This is a HAM (normal) message.')
```

```
This is a HAM (normal) message.
```

Predicting values 4 - Feedback message

sample_message = 'Tammy, thanks for choosing Carl's Car Wash for your express polish. We would love to hear your thoughts on the service. Feel free to text back with any feedback. Safe driving!'

```
In [135]: # Predicting values 4 - Feedback message
          sample_message = 'Tammy, thanks for choosing Carl's Car Wash for your express polish. We wou

          if predict_spam(sample_message):
            print('Gotcha! This is a SPAM message.')
          else:
            print('This is a HAM (normal) message.')
```

```
Gotcha! This is a SPAM message.
```

Github link - https://github.com/Manishmeena10038/SMS-Spam-Classification

Notebook link - http://localhost:8888/notebooks/Downloads/Spam%20SMS%20Classication.ipynb#