A
Project Report
On

" Spam Mail Predictio n "

Submitted To

D Y PATIL INTERNATIONAL UNIVERSITY

AKURDI PUNE

School of Computer Science, Engineering and Applications

Program: MCA SEM I

2023 Batch

Submitted By

Miss. Prajakta Bote - 20230804067
Mr. Prit Mahant - 20230804059
Mr. Pranav Madaghe - 20230804037
Mr. Harshilkumar Munjapara - 20230804004

Subject on

Artificial intelligence & Machine learning

Guidance of

Prof. Kapil Sharma
Miss. Ramya Praveen

Index

Abstract

The aim of this project is to develop a machine learning model that can accurately predict whether an email is spam or not. The model was trained on a dataset consisting of thousands of emails, labeled as either spam or not spam.

Various feature engineering techniques were applied to the input data to improve the model's accuracy, including the use of natural language processing and the extraction of features such as the presence of certain keywords or the length of the email.

The resulting model was able to achieve high accuracy in predicting whether an email is spam or not, with a precision of 96%. This project has the potential to be used in real-world scenarios to prevent users from receiving unwanted junk mail, saving time and increasing productivity.

Introduction

I. Dataset

We was download this dataset from Kaggle.com.This dataset have 5572 Uniques entries.Dataset have 2 columns that is **'Category'** and **'Message'**. In category column two value 'spam' and 'ham'.And message column content email they are spam and ham both.A size of dataset is 5.5mb and format is **'csv'**.

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

[5572 rows x 2 columns]

Fig. : Sample of dataset

We don't required any data prepossessing.First replace category column to 0 and 1,0 stand for spam and 1 stand for ham mails.Second split dataset in 60-40% and also use random state 3.Training dataset have 3343 and test set have 2229 records.

	Category	Message
0	1	Go until jurong point, crazy.. Available only ...
1	1	Ok lar... Joking wif u oni...
2	0	Free entry in 2 a wkly comp to win FA Cup fina...
3	1	U dun say so early hor... U c already then say...
4	1	Nah I don't think he goes to usf, he lives aro...
5	0	FreeMsg Hey there darling it's been 3 week's n...
6	1	Even my brother is not like to speak with me. ...
7	1	As per your request 'Melle Melle (Oru Minnamin...

Fig. : Change Category Values

Introduction

II. Algorithm

Logistic regression is a popular machine learning algorithm used for binary classification problems. It works by fitting a logistic function to the input data and determining the probability of the input belonging to one of two classes.

When using logistic regression, it is important to preprocess the data to remove any noise or irrelevant information. One common technique is to use a TF-IDF vectorizer, which stands for "term frequency-inverse document frequency". This technique calculates a weight for each term in a document, based on how often the term appears in the document and how often it appears in the entire dataset. This helps to identify the most important terms in each document, and can improve the accuracy of the logistic regression model.

The TfidfVectorizer class from the sklearn.feature_extraction.text module in Python provides a simple way to apply TF-IDF vectorization to text data. It can be customized with various parameters, such as the minimum and maximum document frequency, to adjust its behavior for specific use cases.We use **'min_df=10, stop_words="english", lowercase=1'**.

One of its parameters, min_df=10, specifies that any term that appears in less than 10 documents will be ignored. This can help to eliminate noise in the data and improve the performance of the model. Another parameter, stop_words="english", specifies that common English words such as "the" and "and" should be removed from the data. This can help to improve the quality of the model by reducing the number of irrelevant words that are included in the analysis. Finally, lowercase=1 is a parameter that specifies whether or not the text should be transformed into lowercase letters before being analyzed. This can help to ensure that the model is able to recognize words regardless of their capitalization, which can be useful in cases where capitalization is inconsistent. Overall, these parameters can be used to improve the performance of the TfidfVectorizer and ensure that it is able to accurately represent the text data that it is analyzing.

Code & Results

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Fig. : Dependencies

```python
raw_data = pd.read_csv("spam.csv")
```

```python
raw_data.info()
```

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

Fig. : Load and Describe dataset

```python
raw_data.isna().sum()
```

```
Category    0
Message     0
dtype: int64
```

Fig. : Check dataset quality

```
raw_data.loc[raw_data['Category'] == 'spam', 'Category'] = 0
raw_data.loc[raw_data['Category'] == 'ham', 'Category'] = 1
```

```
raw_data.head(20)
```

	Category	Message
0	1	Go until jurong point, crazy.. Available only ...
1	1	Ok lar... Joking wif u oni...
2	0	Free entry in 2 a wkly comp to win FA Cup fina...
3	1	U dun say so early hor... U c already then say...
4	1	Nah I don't think he goes to usf, he lives aro...
5	0	FreeMsg Hey there darling it's been 3 week's n...

Fig. : Replace text data to numeric

```
X_train, X_test, Y_train, Y_test = train_test_split(raw_data.Message, raw_data.Category, test_size=0.4, random_state=3)
```

```
Y_train.shape
```

```
(3343,)
```

```
Y_test.shape
```

```
(2229,)
```

Fig. : Split dataset in train and test set

```
feature_extraction = TfidfVectorizer(min_df=10, stop_words="english", lowercase=1)

X_train_feature = feature_extraction.fit_transform(X_train)
X_test_feature = feature_extraction.transform(X_test)
```

```
C:\Users\admin\anaconda3\lib\site-packages\sklearn\utils\_param_validation.py:591: FutureWarn
parameter is deprecated in version 1.2 and won't be supported anymore in version 1.4.
  warnings.warn(
```

```
Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

```
print(X_train_feature)
```

```
  (0, 467)      0.5874783690622778
  (0, 131)      0.809239869188318
  (1, 77)       0.23719473543344657
  (1, 497)      0.29144569047904656
  (1, 249)      0.18698287905940753
  (1, 248)      0.2437179602985452
  (1, 201)      0.521207236727189
  (1, 136)      0.29887887966435067
  (1, 406)      0.2701067484175382
  (1, 470)      0.5156139382825599
  (1, 58)       0.2542362455124458
  (2, 228)      0.24421436676111594
  (2, 474)      0.34097604903224193
  (2, 404)      0.7517680836158547
  (2, 422)      0.3962879422314371
  (2, 306)      0.31921042018134016
  (3, 262)      0.29016338785793916
  (3, 305)      0.34673819194906935
```

Fig. : Text feature extraction and vectorization

```
model = LogisticRegression()

model.fit(X_train_feature, Y_train)
```

```
LogisticRegression()
```

Fig. : Train model

```python
prediction = model.predict(X_train_feature)
accuracy = accuracy_score(Y_train, prediction)

print("Model Accuracy :",accuracy)
```

```
Model Accuracy : 0.9727789410708944
```

```python
prediction_test = model.predict(X_test_feature)
accuracy_test = accuracy_score(Y_test, prediction_test)

print("Test Model Accuracy :",accuracy_test)
```

```
Test Model Accuracy : 0.9676985195154778
```

Fig. : Model accuracy

```python
user_input = input("Your Mail : ")

# print(list(user_input))
user_input = feature_extraction.transform([user_input])

result =  model.predict(user_input)


if result[0] == 1:
    print("Your mail is ham mail.")
else:
    print("Alert!! Changes to your mail is spam.")
```

```
Your Mail : 'Will u meet ur dream partner soon? Is ur career off 2 a flyng start? 2 find out free, txt HORO followed by ur star sign, e. g. HORO ARIES'
Alert!! Changes to your mail is spam.
```

Fig. : Final Output