**BUILDING AI POWERED SPAM CLASSIFIER**

**TEAM MEMBER**

**211121104032 : KEVIN . B**

**Phase 5 submission document**

**Project Title : AI Spam Classifier**

**Phase 5 :Project Documentation And Submission**

**Topic : In This section we will document the complete project and prepare It for submission**

**Introduction to Spam Classifier: Taming Unwanted Digital Clutter**

In today's digital age, our communication channels are overflowing with information, messages, and emails. Amid the vast sea of digital content, one persistent nuisance remains—spam. Spam refers to unsolicited and irrelevant messages, often laden with advertisements, phishing attempts, or other forms of malicious content. Managing and filtering this barrage of spam has become a crucial part of maintaining a clean and secure digital environment. Enter the Spam Classifier, a powerful tool that plays a pivotal role in identifying and segregating spam from legitimate messages.

**The Growing Spam Epidemic**

As the internet has evolved, so too has the proliferation of spam. What started as simple email chain letters and dubious product promotions has morphed into a sophisticated array of fraudulent schemes, malware delivery mechanisms, and deceptive tactics. Spam targets email inboxes, instant messaging platforms, comment sections, and social media channels, cluttering our digital lives and posing significant threats to privacy and security.

**The Need for Spam Classification**

To combat the ever-expanding onslaught of spam, spam classification systems have become essential. These systems employ advanced algorithms and machine learning techniques to automatically detect and classify messages as either spam or non-spam (often referred to as "ham"). The goal is not only to alleviate the nuisance of unwanted messages but also to protect users from potentially harmful or fraudulent content.

**How Spam Classifiers Work**

Spam classifiers employ a diverse set of techniques to analyze the content, structure, and metadata of messages to determine their authenticity. They may examine factors such as sender information, message content, URLs, attachments, and user interactions. By assessing patterns and features, these classifiers assign a likelihood score to each message, indicating whether it is likely to be spam or not. Advanced classifiers adapt over time, continuously learning from new data to improve their accuracy.

**Key Challenges in Spam Classification**

While spam classifiers have come a long way, they still face several challenges. Spammers continually devise new tactics to evade detection, requiring classifiers to remain vigilant and adaptable. There's also the issue of false positives and negatives, where legitimate messages may be mistakenly classified as spam, or spam messages may slip through the filter. Striking the right balance between aggressive filtering and preserving user-relevant content is an ongoing challenge.

**Application Beyond Email**

Spam classification is not limited to email. It has found applications in various domains, including text message filtering, comment moderation on websites, and even voice-based spam detection for phone calls. The need to maintain digital hygiene and protect users from unwanted and potentially harmful content extends to a wide array of digital communication channels.

**Conclusion**

In an age of information overload and digital clutter, the spam classifier stands as a guardian of our digital spaces. It's a technological sentinel that tirelessly sifts through vast volumes of data, identifying and containing the relentless tide of spam. As spammers continue to innovate, so too do these classifiers, evolving to keep our digital lives clean, secure, and efficient.

The spam classifier is a testament to the power of technology in solving everyday challenges, providing users with a safer and more streamlined online experience. As the digital landscape continues to evolve, the role of the spam classifier remains pivotal in preserving the integrity and utility of our online interactions.

**Tools And Softwares Used In This Model :**

The process of building, loading, and preprocessing datasets in data science, machine learning, and data analysis projects involves a variety of tools and software. Here is a list of commonly used tools and software for each stage of this process:

**Data Collection:**

1. \*\*Web Scraping Tools\*\*:

- Beautiful Soup

- Scrapy

2. \*\*Data Extraction Tools\*\*:

- Selenium

- Apache Nutch

3. \*\*APIs\*\*:

- Requests (Python library)

- Postman

4. \*\*Databases\*\*:

- MySQL

- PostgreSQL

- MongoDB

- SQLite

5. \*\*Data Integration Tools\*\*:

- Apache NiFi

- Talend

**Data Loading:**

1. \*\*Pandas\*\* (Python Library): Widely used for loading, manipulating, and analyzing data from various file formats (e.g., CSV, Excel, JSON).

2. \*\*NumPy\*\* (Python Library): Useful for handling numerical data and arrays.

3. \*\*Apache Spark\*\*: Ideal for big data processing and distributed computing.

4. \*\*SQL Databases Tools\*\*:

- SQL Server Management Studio

- MySQL Workbench

- pgAdmin (for PostgreSQL)

5. \*\*ETL Tools\*\*:

- Apache Nifi

- Talend

- Apache Beam

**Data Preprocessing:**

1. \*\*Pandas\*\* (Python Library): For data cleaning, transformation, and manipulation.

2. \*\*Scikit-Learn\*\* (Python Library): Includes tools for feature extraction, scaling, and transformation.

3. \*\*OpenRefine\*\*: A powerful tool for cleaning and transforming messy data.

4. \*\*Apache OpenNLP\*\*: Useful for natural language processing (NLP) tasks in text data preprocessing.

5. \*\*Data Wrangling Tools\*\*:

- Trifacta

- KNIME

- RapidMiner

6. \*\*Data Visualization Tools\*\*:

- Tableau

- Power BI

- Matplotlib and Seaborn (Python libraries)

**Data Exploration:**

1. \*\*Jupyter Notebook\*\*: Ideal for interactive data exploration with code, visualizations, and explanations.

2. \*\*RStudio\*\*: Popular for data analysis in R language.

3. \*\*Data Visualization Tools\*\*:

- Matplotlib and Seaborn (Python libraries)

- ggplot2 (R library)

4. \*\*Statistical Software\*\*:

- SPSS

- SAS

**Data Export:**

1. \*\*Pandas\*\* (Python Library): Easily export data to various formats.

2. \*\*NumPy\*\* (Python Library): Useful for saving numerical data.

3. \*\*SQL Tools\*\*:

- SQL Server Management Studio

- MySQL Workbench

- pgAdmin

4. \*\*ETL Tools\*\*:

- Apache Nifi

- Talend

- Apache Beam

It's important to note that the choice of tools and software may vary depending on the specific project, dataset, and your personal preferences or organization's requirements. Data scientists and analysts often use a combination of these tools to complete the data preparation process effectively.

**Importing the Dependencies**

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

Data Collection & Pre-Processing

# loading the data from csv file to a pandas Dataframe  
raw\_mail\_data = pd.read\_csv('/content/mail\_data.csv')

print(raw\_mail\_data)

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]

# replace the null values with a null string  
mail\_data = raw\_mail\_data.where((pd.notnull(raw\_mail\_data)),'')

# printing the first 5 rows of the dataframe  
mail\_data.head()

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...

# checking the number of rows and columns in the dataframe  
mail\_data.shape

(5572, 2)

Label Encoding

# label spam mail as 0; ham mail as 1;  
  
mail\_data.loc[mail\_data['Category'] == 'spam', 'Category',] = 0  
mail\_data.loc[mail\_data['Category'] == 'ham', 'Category',] = 1

spam - 0

ham - 1

# separating the data as texts and label  
  
X = mail\_data['Message']  
  
Y = mail\_data['Category']

print(X)

0 Go until jurong point, crazy.. Available only ...  
1 Ok lar... Joking wif u oni...  
2 Free entry in 2 a wkly comp to win FA Cup fina...  
3 U dun say so early hor... U c already then say...  
4 Nah I don't think he goes to usf, he lives aro...  
 ...   
5567 This is the 2nd time we have tried 2 contact u...  
5568 Will ü b going to esplanade fr home?  
5569 Pity, \* was in mood for that. So...any other s...  
5570 The guy did some bitching but I acted like i'd...  
5571 Rofl. Its true to its name  
Name: Message, Length: 5572, dtype: object

print(Y)

0 1  
1 1  
2 0  
3 1  
4 1  
 ..  
5567 0  
5568 1  
5569 1  
5570 1  
5571 1  
Name: Category, Length: 5572, dtype: object

Splitting the data into training data & test data

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=3)

print(X.shape)  
print(X\_train.shape)  
print(X\_test.shape)

(5572,)  
(4457,)  
(1115,)

Feature Extraction

from sklearn.feature\_extraction.text import TfidfVectorizer  
  
# Create a TF-IDF vectorizer  
feature\_extraction = TfidfVectorizer(min\_df=1, stop\_words='english', lowercase=True)  
  
# Transform the training and test data  
X\_train\_features = feature\_extraction.fit\_transform(X\_train)  
X\_test\_features = feature\_extraction.transform(X\_test)  
  
# Convert Y\_train and Y\_test values to integers  
Y\_train = Y\_train.astype('int')  
Y\_test = Y\_test.astype('int')

print(X\_train)

3075 Don know. I did't msg him recently.  
1787 Do you know why god created gap between your f...  
1614 Thnx dude. u guys out 2nite?  
4304 Yup i'm free...  
3266 44 7732584351, Do you want a New Nokia 3510i c...  
 ...   
789 5 Free Top Polyphonic Tones call 087018728737,...  
968 What do u want when i come back?.a beautiful n...  
1667 Guess who spent all last night phasing in and ...  
3321 Eh sorry leh... I din c ur msg. Not sad alread...  
1688 Free Top ringtone -sub to weekly ringtone-get ...  
Name: Message, Length: 4457, dtype: object

print(X\_train\_features)

(0, 5413) 0.6198254967574347  
 (0, 4456) 0.4168658090846482  
 (0, 2224) 0.413103377943378  
 (0, 3811) 0.34780165336891333  
 (0, 2329) 0.38783870336935383  
 (1, 4080) 0.18880584110891163  
 (1, 3185) 0.29694482957694585  
 (1, 3325) 0.31610586766078863  
 (1, 2957) 0.3398297002864083  
 (1, 2746) 0.3398297002864083  
 (1, 918) 0.22871581159877646  
 (1, 1839) 0.2784903590561455  
 (1, 2758) 0.3226407885943799  
 (1, 2956) 0.33036995955537024  
 (1, 1991) 0.33036995955537024  
 (1, 3046) 0.2503712792613518  
 (1, 3811) 0.17419952275504033  
 (2, 407) 0.509272536051008  
 (2, 3156) 0.4107239318312698  
 (2, 2404) 0.45287711070606745  
 (2, 6601) 0.6056811524587518  
 (3, 2870) 0.5864269879324768  
 (3, 7414) 0.8100020912469564  
 (4, 50) 0.23633754072626942  
 (4, 5497) 0.15743785051118356  
 : :  
 (4454, 4602) 0.2669765732445391  
 (4454, 3142) 0.32014451677763156  
 (4455, 2247) 0.37052851863170466  
 (4455, 2469) 0.35441545511837946  
 (4455, 5646) 0.33545678464631296  
 (4455, 6810) 0.29731757715898277  
 (4455, 6091) 0.23103841516927642  
 (4455, 7113) 0.30536590342067704  
 (4455, 3872) 0.3108911491788658  
 (4455, 4715) 0.30714144758811196  
 (4455, 6916) 0.19636985317119715  
 (4455, 3922) 0.31287563163368587  
 (4455, 4456) 0.24920025316220423  
 (4456, 141) 0.292943737785358  
 (4456, 647) 0.30133182431707617  
 (4456, 6311) 0.30133182431707617  
 (4456, 5569) 0.4619395404299172  
 (4456, 6028) 0.21034888000987115  
 (4456, 7154) 0.24083218452280053  
 (4456, 7150) 0.3677554681447669  
 (4456, 6249) 0.17573831794959716  
 (4456, 6307) 0.2752760476857975  
 (4456, 334) 0.2220077711654938  
 (4456, 5778) 0.16243064490100795  
 (4456, 2870) 0.31523196273113385

Training the Model

Logistic Regression

model = LogisticRegression()

# training the Logistic Regression model with the training data  
model.fit(X\_train\_features, Y\_train)

LogisticRegression()

Evaluating the trained model

# prediction on training data  
  
prediction\_on\_training\_data = model.predict(X\_train\_features)  
accuracy\_on\_training\_data = accuracy\_score(Y\_train, prediction\_on\_training\_data)

print('Accuracy on training data : ', accuracy\_on\_training\_data)

Accuracy on training data : 0.9670181736594121

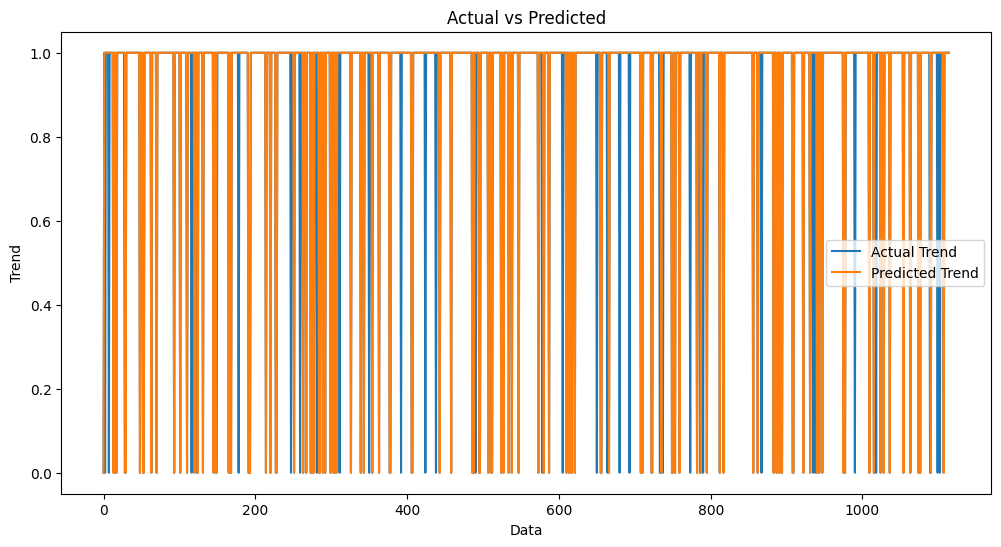
# prediction on test data  
  
prediction\_on\_test\_data = model.predict(X\_test\_features)  
accuracy\_on\_test\_data = accuracy\_score(Y\_test, prediction\_on\_test\_data)

print('Accuracy on test data : ',accuracy\_on\_test\_data)

Accuracy on test data : 0.9659192825112107

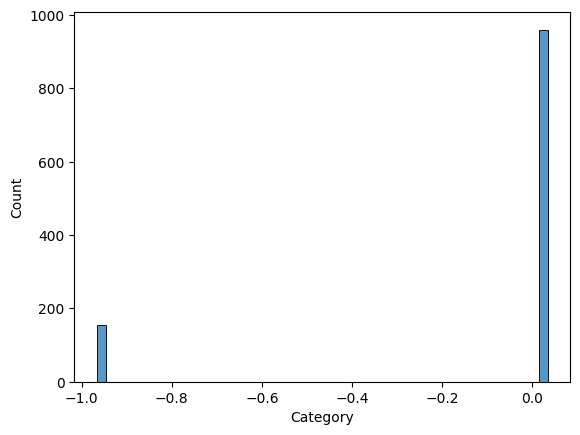
import matplotlib.pyplot as plt  
plt.figure(figsize=(12,6))  
plt.plot(np.arange(len(Y\_test)), Y\_test, label='Actual Trend')  
plt.plot(np.arange(len(Y\_test)), prediction\_on\_test\_data, label='Predicted Trend')  
plt.xlabel('Data')  
plt.ylabel('Trend')  
plt.legend()  
plt.title('Actual vs Predicted')

Text(0.5, 1.0, 'Actual vs Predicted')



import seaborn as sns  
sns.histplot((Y\_test-accuracy\_on\_test\_data), bins=50)

<Axes: xlabel='Category', ylabel='Count'>



Building a Predictive System

input\_mail = ["I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times"]  
  
# convert text to feature vectors  
input\_data\_features = feature\_extraction.transform(input\_mail)  
  
# making prediction  
  
prediction = model.predict(input\_data\_features)  
print(prediction)  
  
  
if (prediction[0]==1):  
 print('Ham mail')  
  
else:  
 print('Spam mail')

[1]  
Ham mail

input\_mail = ["Urgent UR awarded a complimentary trip to EuroDisinc Trav, Aco&Entry41 Or Â£1000. To claim txt DIS to 87121 18+6\*Â£1.50(moreFrmMob. ShrAcomOrSglSuplt)10, LS1 3AJ"]  
  
# convert text to feature vectors  
input\_data\_features = feature\_extraction.transform(input\_mail)  
  
# making prediction  
  
prediction = model.predict(input\_data\_features)  
print(prediction)  
  
  
if (prediction[0]==1):  
 print('Ham mail')  
  
else:  
 print('Spam mail')

[0]  
Spam mail

**Conclusion: Empowering Data Workflows with Essential Tools**

In the realm of data science, machine learning, and data analysis, the journey from raw data to valuable insights is a complex and multifaceted one. The robustness and efficiency of this journey often hinge on the tools and software leveraged at each stage. As we conclude our exploration of the tools commonly used in data collection, loading, and preprocessing, it becomes evident that these tools are the unsung heroes behind the scenes of data-driven success.

Data Collection, the initial phase, finds its allies in web scraping tools, data extraction utilities, APIs, and databases. These tools enable the extraction of data from a multitude of sources, ranging from websites and cloud-based platforms to local databases and APIs. They form the gateway to an extensive world of data, making it accessible for further analysis and transformation.

Data Preprocessing is where the magic happens. This is the phase where the raw data undergoes a transformation from its unpolished state to an analysis-ready format. Data cleaning, feature engineering, and data transformation are some of the critical operations conducted during this phase. Tools such as Pandas, Scikit-Learn, OpenRefine, and data wrangling platforms play a vital role in shaping the data into a form that's conducive to accurate analysis and modeling.

In conclusion, the tools and software mentioned in this list are the backbone of modern data workflows. They empower data professionals to make sense of vast amounts of information, clean and transform it, explore its nuances, and ultimately derive valuable insights and informed decisions. These tools represent a harmonious synergy between human intellect and technological innovation.

As the landscape of data science continues to evolve, these tools adapt and improve, opening new horizons for data-driven discovery. They enable data professionals to navigate the intricate journey from raw data to actionable insights, contributing to advancements in various fields, from business analytics to scientific research.

Whether you are a seasoned data scientist or a newcomer to the world of data analysis, the utilization of these tools is a fundamental step on the path to unlocking the full potential of data, and it is a testament to the ever-expanding possibilities that technology offers in the quest for knowledge and understanding.