A Project Report

On

Email Spam Detection Using ML

Submitted in partial fulfillment of the requirement

for the award of the degreeof

MASTER OF COMPUTER APPLICATION

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

DEGREE

Session 2023-24

Submitted By

Abhishek Kumar (23SCSE2030018)

Aradhya Parashar (23SCSE2030056)

Umashankar Mahato (23SCSE2030019)

Under the Supervision of

[Dr. Pratima Singh]

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

June 2024

ABSTRACT

Now a days, all the people are communicating official information through emails.Spam mails are the major issue on the internet. It is easy to send an email which contains spam message by the spammers. Spam fills our inbox with several irrelevant emails. Spammers can steal our sensitive information from our device like files, contact. Even we have the latest technology it is challenging to detect spam emails. This paper aims to propose a term frequency inverse Document frequency (TFIDF) approach by implementing the support vector machine algorithm. The results are compared in terms of the confusion matrix, accuracy, and data achieved by using the term frequency inverse document frequency(TFIDF) based support vector machine (SVM) system People are using them for illegal and unethical conducts, phishing, and fraud. Sending malicious link throughspam emails which can harm our system and can also seek in into your system. Creating a fake profile and email account is much easy for the spammers.

INTRODUCTION

Today, Spam has become a major problem in communication over internet. It hasbeen accounted that aound 55% of all emails are reported as spam and the number has been growing steadily. Spam which is also known as unsolicited bulk email has led to the increasing use of email as email provides the perfect ways to send the unwanted advertisement or junk newsgroup posting at no cost for the sender. This chance has been extensively exploited by irresponsible organizationsand resulting to clutter the mailboxes of millions of people all around the world.

Spam has been a major concern given the offensive content of messages; spam isa waste of time. End user is at risk of deleting legitimate mail by mistake.
Moreover, spam also impacted the economical which led some countries to adoptlegislation.

Text classification is used to determine the path of incoming mail/message eitherinto inbox or straight to spam folder. It is the process of assigning categories to text according to its content. It is used to organic, structures and categorize text. It can be done either manually or automatically, Machine learning, automaticallyclassifies the text in a much lister way than manual technique. Machine learning uses pre- labelled text to learn the different associations between pieces of text and it output. It used feature extraction to transform each text to numerical representation in form of vector which represents the frequency of word in predefined dictionary.

Text classification is important to structure the unstructured and messy nature oftext such as documents and spam messages in a in a cost-effective way. Machinelearning can make more accurate precisions in real-time and help to improve the
manual slow process to much better and faster analyzing big data. It is importantespecially to a company to analyze text data, help inform business decisions and even automate business processes.
In this project, machine learning techniques are used to detect the spam messageof a mail. Machine learning is where computers can learn to do something.

without the need to explicitly program them for the task. It uses dataand produce a program to perform a task such as classification.
Compared to knowledge engineering, machine learning techniques require messages that have been successfully pre-classified. The pre- classified messages make the training dataset which will be used to fitthe learning algorithm to the model in machine learning studio.

A combination of algorithm is used to learn the classification rules frommessages. These algorithms are used for classification of objects of different classes. These algorithms are provided with pre labelled data and an unknown text. After learning from the prelabelled data each of these algorithms predict which class the unknown text may belong to, and the category predicted by majority is considered as final.

OBJECTIVES AND SCOPE

Problem Statement

Spammers are in continuous war with email service providers. Email service providers implement various span flaring methods to retain their users, and spammers are continuing changing patterns, using veins embedding tricks to get through filtering. These filters can never be too aggressive because a slight misclassification may lead to important information loss for consumer. A rigid filtering method withadditional reinforcements is needed to tackle this problem.

Objectives

The objectives of this project are

- To create an ensemble algorithm for classification of spamwith highest possible accuracy
- To study on how to use machine learning for spam detection
- To study how natural language processing techniques canbe implemented in-spam detection.
- To provide user with insights of given text leveraging the created algorithm and NL. P

Project Scope

This project needs a coordinated scope of work.

- Combine existing machine learning algorithms to form a better ensemble algorithm.
- Clear, processing and make use of the dataset for trainingand testing the model created.
- Analyze the tests and extract entities for presentation.

TECHNOLOGIES USED

(i) Python

Python, renowned for its simplicity, readability, and versatility, stands as a preeminent programming language the vast landscape of software development. Originating from the guiding principles of clarity and ease of use, Python has grown into a dynamic and powerful tool embraced by developers, data scientists, and engineers across diverse domains. Its clean syntax facilitates rapid development, making it an ideal language for both beginners and experienced programmers. Python's extensive standard library and rich ecosystem of third- party package contribute to its adaptability for a python myriad of applications, from web development and scientific computing to artificial intelligence and automation. As an open-source language, Python encourages collaboration and innovation, fostering a vibrant community that continually expands its capabilitiesWhether employed for crafting robust web applications or implementing complexmachine learning algorithms, Python's inherent and simplicity belies its ability to tackle intricate tasks, making it an indispensable asset in the toolkit of modern- day developer.

(ii) PICKLE

Pickle, in the realm of Python of programming, refers to a module that provides a mechanism for serializing and deserializing Python objects. Serialization involves converting complex data structures, such as lists or dictionaries, into a format thatcan be easily stored or transmitted, often in the form of a binary file. Pickle, as a module excels at this task, allowing Python developers to save the state of their programs or store objects persistently. The versatility of Pickle extends beyond simple data structures to include custom classes and instances. This capability makes it a valuable tool for tasks such as saving machine learning models, caching, or transferring data between different Python applications. Despite its utility, developers should exercise caution when unpickling data from untrusted sources, as Pickle can execute arbitrary code during deserialization Nevertheless,in controlled environments, Pickle stands as a convenient and efficient.

(iii) PANDAS

Pandas, a powerful and widely used library in the Python programming language, has become synonymous with data manipulation and analysis. Developed to address the challenges of working with structured data, Pandas provides high- performance, easy-to-use data structures, and data analysis tools. At its core, Pandas revolves around two primary data structures Series and Data Frame whichenable users to efficiently organize, manipulate, and analyze tabular data. With a syntax that emphasizes simplicity and expressiveness, Pandas has become the go- to tool for data scientists, analysts, and developers working on tasks such as data cleaning, exploration, and transformation. Its seamless integration with other Python libraries, coupled with its ability to handle diverse data types and missing values, makes Pandas an indispensable asset in the data science toolkit. Whether dealing with large datasets or performing intricate data wrangling tasks,

Pandas provides a versatile and efficient solution, contributing significantly to the accessibility and effectiveness of data analysis on the Python ecosystem.

(iv) SCIKITE-LEARN

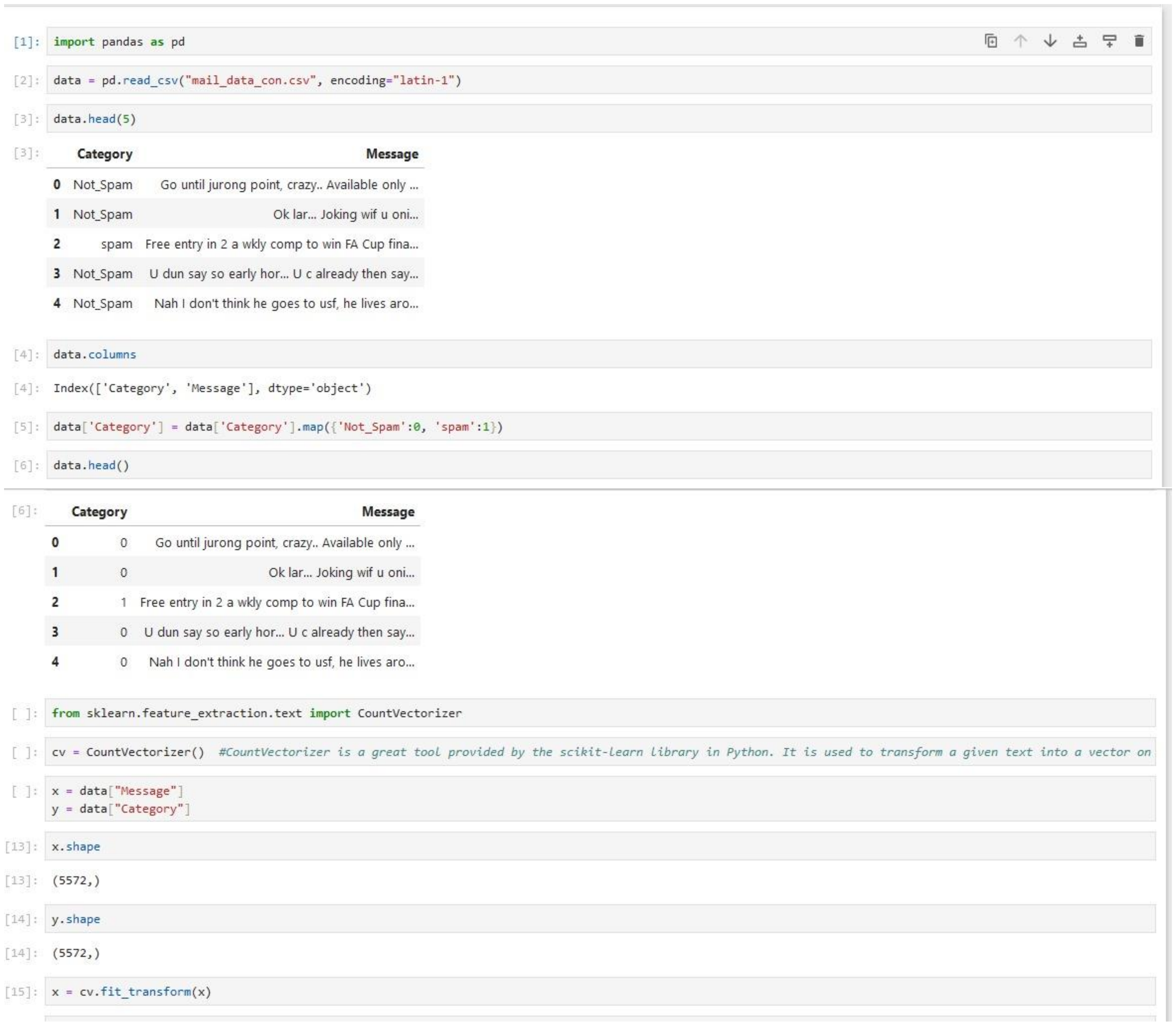
Scikit-learn, a cornerstone in the realm of machine learning and data science, has established itself as a go-to library for practitioners seeking a powerful and user- friendly toolset. Developed on the principles of simplicity and accessibility, scikit- learn is an open-source Python library that provides an extensive array of algorithms for tasks such as classification, regression, clustering, and dimensionality reduction. With a focus on case of use and consistency, scikit-learnsimplifies the implementation of complex machine learning models, enabling developers and researchers to quickly experiment with and deploy sophisticated solutions. Its comprehensive documentation and straightforward API design make it particularly suitable for users ranging from novices to seasoned experts. From preprocessing data to evaluating model performance, scikit-learn seamlessly integrates into the broader Python ecosystem, making if an indispensable resource for those engaged in the exploration and application of machine learningtechniques. As an enduring force in the data science landscape, scikit-learn continues to empower practitioners, fostering innovation and advancements in the ever-evolving field of machine learning.

(v) JUPYTER NOTEBOOK

Jupyter Notebook, an interactive and open-source web application, has become an indispensable tool in the realm of data science, research, and education.

Named after the three core programming languages it supports-Julia, Python, and R-Jupyter provides an innovative environment that seamlessly combines code, visualizations, and narrative text. With its user-friendly interface, Jupyter Notebook allows users to create and share documents containing live code, equations, plots, and explanatory text. This integration of computational elements and documentation makes it particularly well-suited for tasks such as data analysis, machine learning, and scientific research. The real-time feedback provided by Jupyter's interactive environment enhances the exploratory and collaborative aspects of coding, fostering a more intuitive and dynamic workflow. Its versatility extends beyond the data science community as Jupyter Notebooks find application in various fields where code, data, and insights converge, making it a cornerstone tool in modern programming and research practices.

SOURCE CODE



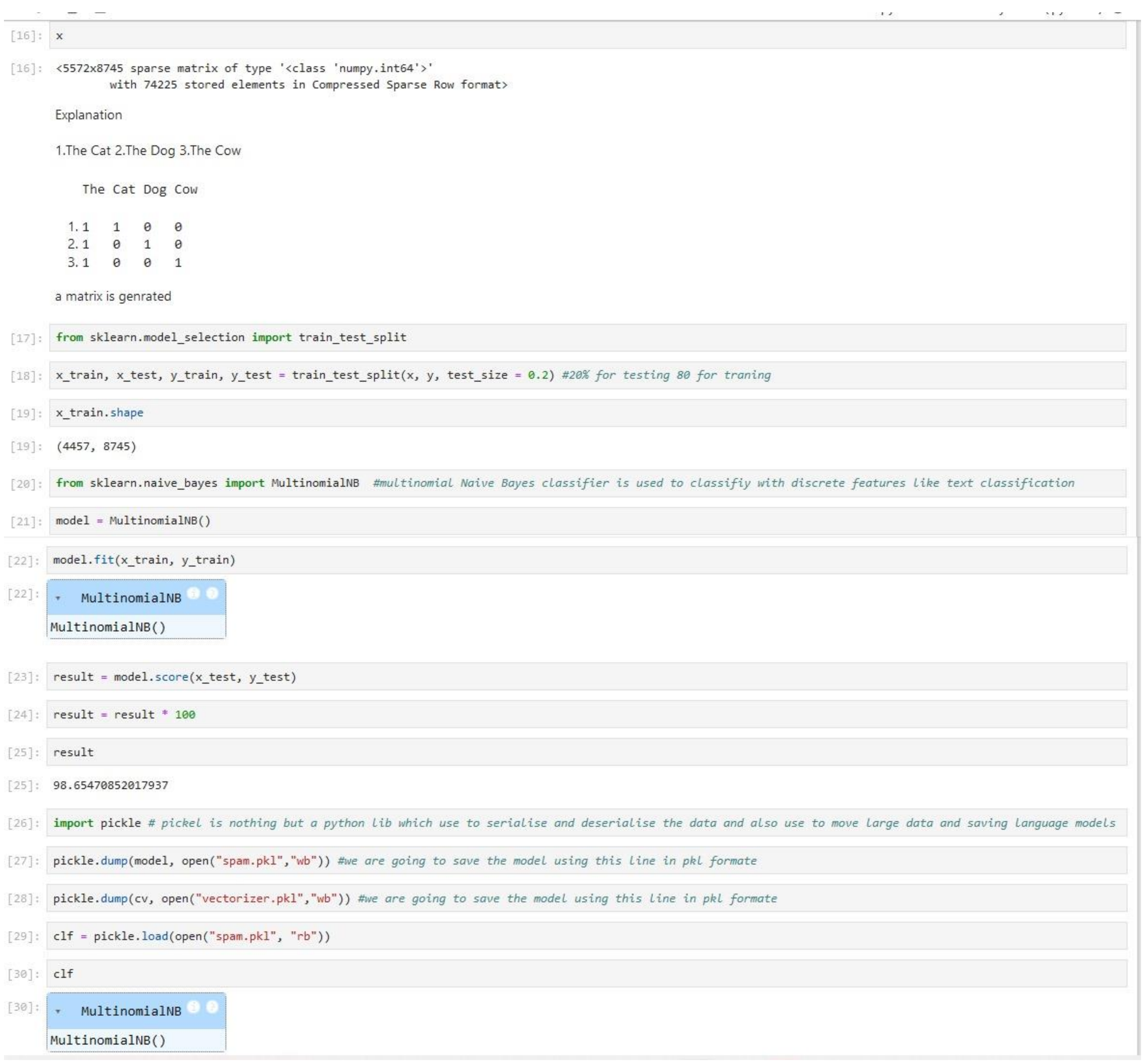
```
[1]: import pandas as pd

[2]: data = pd.read_csv("mail_data_con.csv", encoding="latin-1")

[3]: data.head(5)

[3]:
```

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

```
[4]: data.columns

[4]: Index(['Category', 'Message'], dtype='object')

[5]: data['Category'] = data['Category'].map({'Not_Spam':0, 'spam':1})

[6]: data.head()

[6]:
```

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

```
[ ]: from sklearn.feature_extraction.text import CountVectorizer

[ ]: cv = CountVectorizer()  #CountVectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on

[ ]: x = data["Message"]
     y = data["Category"]

[13]: x.shape

[13]: (5572,)

[14]: y.shape

[14]: (5572,)

[15]: x = cv.fit_transform(x)
```



```
[16]: x

[16]: <5572x8745 sparse matrix of type '<class 'numpy.int64'>'
              with 74225 stored elements in Compressed Sparse Row format>
```

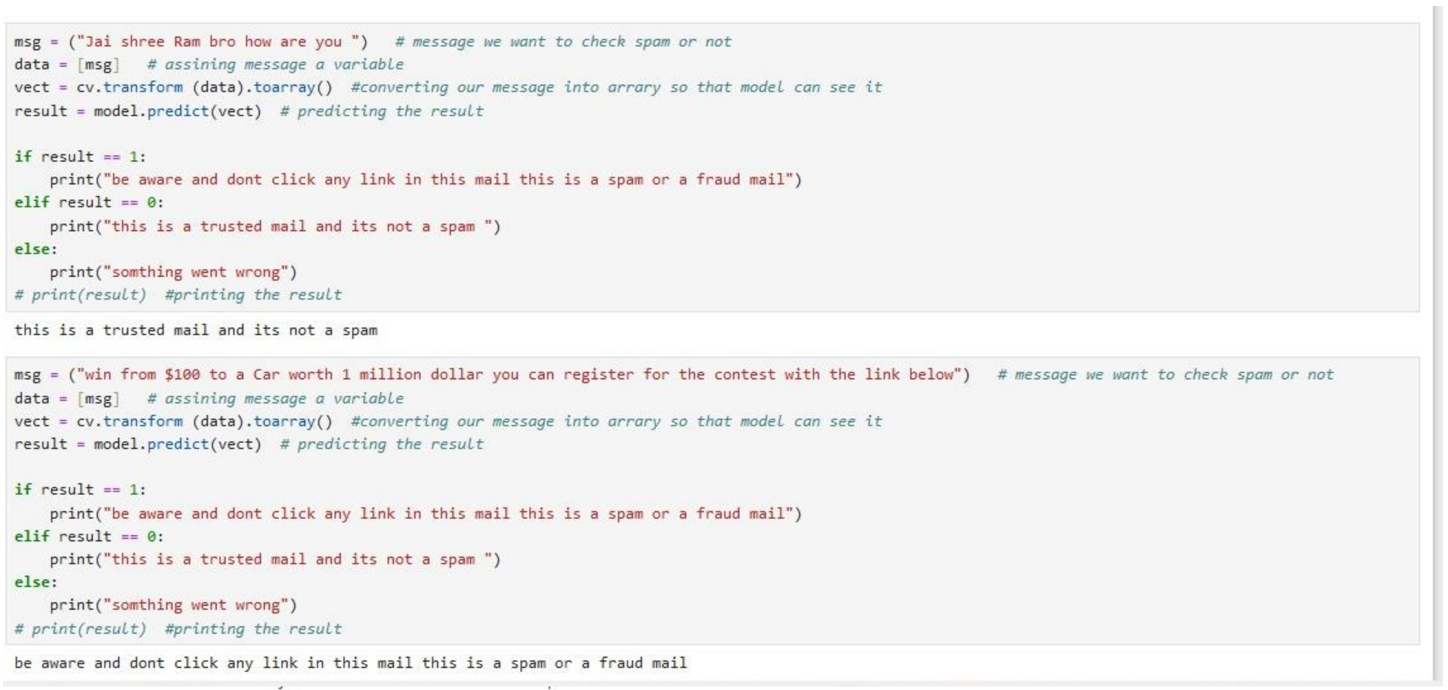
Explanation

1.The Cat 2.The Dog 3.The Cow

```
    The Cat Dog Cow

  1. 1   1   0   0
  2. 1   0   1   0
  3. 1   0   0   1
```

a matrix is genrated

```
[17]: from sklearn.model_selection import train_test_split

[18]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2) #20% for testing 80 for traning

[19]: x_train.shape

[19]: (4457, 8745)

[20]: from sklearn.naive_bayes import MultinomialNB  #multinomial Naive Bayes classifier is used to classifiy with discrete features like text classification

[21]: model = MultinomialNB()

[22]: model.fit(x_train, y_train)

[22]: ▾ MultinomialNB
      MultinomialNB()

[23]: result = model.score(x_test, y_test)

[24]: result = result * 100

[25]: result

[25]: 98.65470852017937

[26]: import pickle # pickel is nothing but a python lib which use to serialise and deserialise the data and also use to move large data and saving language models

[27]: pickle.dump(model, open("spam.pkl","wb")) #we are going to save the model using this line in pkl formate

[28]: pickle.dump(cv, open("vectorizer.pkl","wb")) #we are going to save the model using this line in pkl formate

[29]: clf = pickle.load(open("spam.pkl", "rb"))

[30]: clf

[30]: ▾ MultinomialNB
      MultinomialNB()
```

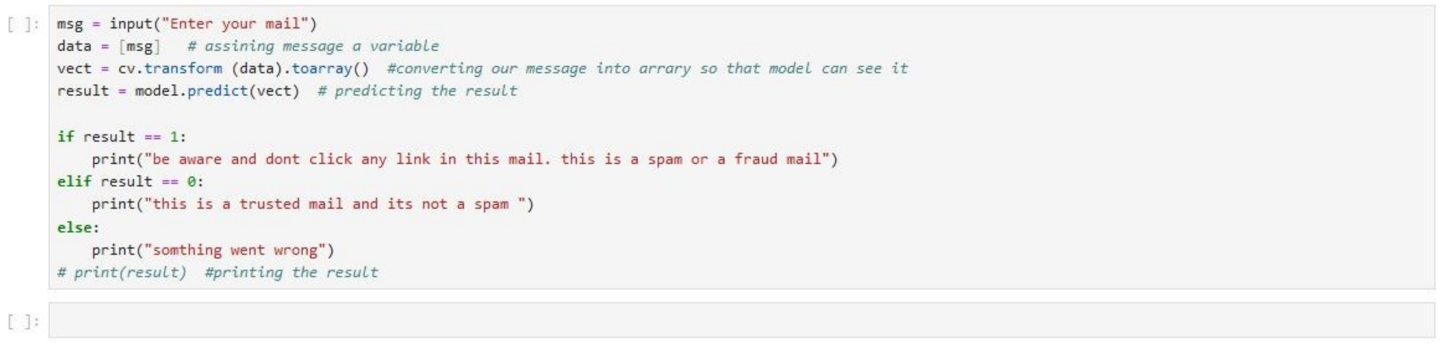


```
msg = ("Jai shree Ram bro how are you ")   # message we want to check spam or not
data = [msg]   # assining message a variable
vect = cv.transform (data).toarray()  #converting our message into arrary so that model can see it
result = model.predict(vect)  # predicting the result

if result == 1:
    print("be aware and dont click any link in this mail this is a spam or a fraud mail")
elif result == 0:
    print("this is a trusted mail and its not a spam ")
else:
    print("somthing went wrong")
# print(result)  #printing the result
```

```
this is a trusted mail and its not a spam
```

```
msg = ("win from $100 to a Car worth 1 million dollar you can register for the contest with the link below")   # message we want to check spam or not
data = [msg]   # assining message a variable
vect = cv.transform (data).toarray()  #converting our message into arrary so that model can see it
result = model.predict(vect)  # predicting the result

if result == 1:
    print("be aware and dont click any link in this mail this is a spam or a fraud mail")
elif result == 0:
    print("this is a trusted mail and its not a spam ")
else:
    print("somthing went wrong")
# print(result)  #printing the result
```

```
be aware and dont click any link in this mail this is a spam or a fraud mail
```



```
[ ]: msg = input("Enter your mail")
     data = [msg]   # assining message a variable
     vect = cv.transform (data).toarray()  #converting our message into arrary so that model can see it
     result = model.predict(vect)  # predicting the result

     if result == 1:
         print("be aware and dont click any link in this mail. this is a spam or a fraud mail")
     elif result == 0:
         print("this is a trusted mail and its not a spam ")
     else:
         print("somthing went wrong")
     # print(result)  #printing the result
```

```
[ ]:
```