

A Project/Dissertation ETE Report

on

“Email/SMS Spam Classifier”

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

B.Tech in Computer Science & Engineering



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

Under The Supervision of

Ms. Kiran Singh

Professor

Submitted By: BT4109

MAHESH KUMAR SHARMA -18SCSE1010357

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA- 2021-22

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled “**Email/SMS Spam Classifier**” in partial fulfillment of the requirements for the award of the **Bachelor of Technology** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July, 2021 to December, 2021 under the supervision of **Ms. Kiran Singh , Professor** in Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

MAHESH KUMAR SHARMA

18SCSE1010357

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

MS. KIRAN SINGH

PROFESSOR

TABLE OF CONTENTS

S. No.	Particulars	Page No.
1	Abstract	1
2	List of Tables	2
3	Introduction	3
4	Literature Survey	4
5	Requirements, Feasibility & Objective	5
6	Design	6
7	Diagrams	8-10
8	Testing	11
9	Output	12-13

Abstract

Spams are nowadays part of one's life. It is very difficult sometimes to differentiate between the spam and genuine messages. Online fraudster use this lack of knowledge to their benefit by taking improper advantage of innocent people.

SMS spam classifier come in your help, by using the extensive database and using machine learning , it helps one to differentiate between spam and genuine message. One just need to visit our website and copy the suspicious content and get the result whether it's a spam or not.

One can stay safe in this technological era and keep their near and dear ones safe from any online fraud and threat.

It is also used by google and amazon for classifying genuine comments from the false one, which is there to degrade the product review.

The two common approaches used for filtering spam mails are knowledge engineering and machine learning. Emails are classified as either spam or ham using a set of rules in knowledge engineering. ... A particular machine learning algorithm is then used to learn the classification rules from these email messages.

List of Tables

Table for Student Data:

S. No	Name	Enrollment Number	Admission Number	Program / Branch	Sem
1	MAHESH KUMAR SHARMA	18021011589	18SCSE1010357	B.Tech CSE	7

Faculty Data:

Guide Name: Ms. Kiran Singh

Designation: Professor

Introduction

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. The person sending the spam messages is referred to as the spammer. Such a person gathers email addresses from different websites, chatrooms, and viruses . Spam prevents the user from making full and good use of time, storage capacity and network bandwidth. The huge volume of spam mails flowing through the computer networks have destructive effects on the memory space of email servers, communication bandwidth, CPU power and user time . The menace of spam email is on the increase on yearly basis and is responsible for over 77% of the whole global email traffic . Users who receive spam emails that they did not request find it very irritating. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, Bank Verification Number (BVN) and credit card numbers.

To effectively handle the threat posed by email spams, leading email providers such as Gmail, Yahoo mail and Outlook have employed the combination of different machine learning (ML) techniques such as Neural Networks in its spam filters. These ML techniques have the capacity to learn and identify spam mails and phishing messages by analyzing loads of such messages throughout a vast collection of computers. Since machine learning have the capacity to adapt to varying conditions, Gmail and Yahoo mail spam filters do more than just checking junk emails using pre-existing rules. They generate new rules themselves based on what they have learnt as they continue in their spam filtering operation. The machine learning model used by Google have now advanced to the point that it can detect and filter out spam and phishing emails with about 99.9 percent accuracy. The implication of this is that one out of a thousand messages succeed in evading their email spam filter. Statistics from Google revealed that between 50-70 percent of emails that Gmail receives are unsolicited mail. Google's detection models have also incorporated tools called Google Safe Browsing for identifying websites that have malicious URLs.

Literature Survey

Bo Yu and Zong-ben Xu (2008) performed a comparative analysis on content-based spam classification using four different machine learning algorithms. This paper classified spam emails using four different machine learning algorithms viz. Naïve Bayesian, Neural Network, Support Vector Machine and Relevance Vector Machine. The analysis was performed on different training dataset and feature selection. Analysis results demonstrated that NN algorithm is no good enough algorithm to be used as a tool for spam rejection. SVM and RVM machine learning algorithms are better algorithms than NB classifier. Instead of slow learning, RVM is still better algorithm than SVM for spam classification with less execution time and less relevance vectors.

This paper proposed an efficient spam classification method along with feature selection using content of emails and readability. This paper used datasets such as UMI MACHINE LEARNING in Kaggle.com

The proposed approach is able to classify emails of any language because the features are kept independent of the languages.

Firstly it uses certain steps like data cleaning, EDA(Exploratory Data Analysis),Text preprocessing, Model building, Evaluation, Improvement, Website, Deployment.

REQUIREMENTS

Coding Platforms:

➤ Sublime Text 3

Sublime Text 3 (ST3) is a **lightweight, cross-platform code editor** known for its speed, ease of use, and strong community support. It's an incredible editor right out of the box, but the real power comes from the ability to enhance its functionality using Package Control and creating custom settings.

➤ Jupyter Notebook

The Jupyter Notebook is a web application for creating and sharing documents that contain code, visualizations, and text. It can be used for data science, statistical modeling, machine learning, and much more.

Languages Used:

➤ Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at

a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Deployment Platform Used:

- Sublime Text 3 And Stream Lit

Streamlit is **an open-source python framework for building web apps for Machine Learning and Data Science**. ... Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

DESCRIPTION OF REQUIREMENTS

MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

Some Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

Streamlit

Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

Installing Streamlit

1. Make sure you have python installed in your system.
2. Use the following command to install streamlit.

Development flow

If the source code of the streamlit's python script changes the app shows whether to rerun the application or not in the top-right corner. You can also select the 'Always rerun' option to rerun always when the source script changes.

This makes our development flow much easier, every time you make some changes it'll reflect immediately in your web app. This loop between coding and viewing results live makes you work seamlessly with streamlit.

Data flow

Streamlit allows you to write an app the same way you write a python code. The streamlit has a distinctive data flow, any time something changes in your code or anything needs to be updated on the screen, streamlit reruns your python script entirely from the top to the bottom. This happens when the user interacts with the widgets like a select box or drop-down box or when the source code is changed.

If you have some costly operations while rerunning your web app, like loading data from databases, you can use streamlit's `st.cache` method to cache those datasets, so that it loads faster.

Displaying the data

Streamlit provides you with many methods to display various types of data like arrays, tables, and data frames.

To write a string simply use, `st.write("Your string")`

To display a data frame use, `st.dataframe` method

Widgets

There are several widgets available in streamlit, like `st.selectbox`, `st.checkbox`, `st.slider`, and etc.

Layout

You can easily arrange your widgets or data seamlessly using the `st.sidebar` method. This method helps you to align data in the left panel sidebar. All you have to do is simply use `st.sidebar.selectbox` to display a selectbox in the left panel.

Working of streamlit

- Streamlit runs the python script from top to bottom
- Each time the user interacts the script is a rerun from top to bottom
- Streamlit allows you to use caching for costly operations like loading large datasets.

PyCharm Community

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!

It supports two versions: v2.x and v3.x.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm:

1. Intelligent Code Editor:

- It helps us write high-quality codes!
- It consists of color schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- It helps identify errors easily.
- It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- It helps developers in editing and enhancing the code with less effort and time.

- With code navigation, a developer can easily navigate to a function, class, or file.
- A programmer can locate an element, a symbol, or a variable in the source code within no time.
- Using the lens mode, further, a developer can thoroughly inspect and debug the entire source code.

3. Refactoring:

- It has the advantage of making efficient and quick changes to both local and global variables.
- Refactoring in PyCharm enables developers to improve the internal structure without changing the external performance of the code.
- It also helps split up more extended classes and functions with the help of the extract method.

4. Assistance for Many Other Web Technologies:

- It helps developers create web applications in Python.
- It supports popular web technologies such as HTML, CSS, and JavaScript.
- Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.
- The developers can follow the changes directly on a web browser.
- PyCharm also supports AngularJS and NodeJS for developing web applications.

5. Support for Popular Python Web Frameworks:

- PyCharm supports web frameworks such as Django.
- It provides the autocomplete feature and suggestions for the parameters of Django.
- It helps in debugging the codes of Django.
- It also assist web2py and Pyramid, the other popular web frameworks.

6. Assistance for Python Scientific Libraries:

- PyCharm supports Python's scientific libraries such as Matplotlib, NumPy, and Anaconda.
- These scientific libraries help in building projects of Data Science and Machine Learning.
- It consists of interactive graphs that help developers understand data.
- It is capable of integrating with various tools such as IPython, Django, and Pytest. This integration helps innovate unique solutions.

- **Heroku**

- Heroku is a cloud service platform whose popularity has grown in recent years. Heroku is so easy to use that it's a top choice for many development projects.

- With a special focus on supporting customer-focused apps, it enables simple application development and deployment. Since the Heroku platform manages

- hardware and servers, businesses that use Heroku are able to focus on perfecting their apps. And not the infrastructure that supports them.

- Heroku, a Platform-as-a-Service solution, is generally easy-to-use. But it's most beneficial to businesses in specific situations. Heroku has a free service model for small projects. Also, tiered service packages exist for cases where more complex business needs must be addressed.

- The Heroku cloud service platform is based on a managed container (called dynos within the Heroku paradigm) system. It has integrated data services and a powerful ecosystem for deploying and running modern applications.

-

- **Features of Heroku:**

- **1. Heroku Accommodates Many Development Languages:**

- Heroku supports several programming languages that are used as a web application

- deployment model. As one of the first cloud platforms, Heroku has been in development since June 2007. Back then, it supported only the Ruby programming language.

- But now it also supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. This

-

- means a variety of developers can look to Heroku for an inexpensive way to scale their application, no matter their preferred development language.

-

- **2. Heroku Supports Diverse Solutions:**

- Heroku also provides custom buildpacks, where developers can deploy apps in any other programming language. For this reason, Heroku is a polyglot platform. It lets the developer build, run, and scale applications in a similar manner across all programming languages.

-

- Polymorphism and scalability are reasons why Heroku is often seen as a preferred platform amongst developers.

- **3. Heroku Dynos Enable Easy Development and Better Usability:**

- Applications that are run on Heroku typically have unique domain names, which are used to route HTTP requests to the correct container. Applications as services use application containers. Containers are designed to package and run services. Each of the application containers is a smart container on a reliable, fully-managed runtime environment.

- **4. Heroku Lets Developers Scale Applications Instantly:**

- This is accomplished either by increasing the number of dynos or by changing the type of dyno in which the app runs. When the application can scale so easily, the user can always expect more speed when using that application.

Jupyter Notebook

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

The Jupyter Notebook is quite useful not only for learning and teaching a programming language such as Python but also for sharing your data.

Project Jupyter recently launched their latest product, JupyterLab. JupyterLab incorporates Jupyter Notebook into an Integrated Development type Editor that you run in your browser. You can kind of think of JupyterLab as an advanced version of Jupyter Notebook. JupyterLab allows you to run terminals, text editors and code consoles in your browser in addition to Notebooks.

Python

Nowadays, Python is in great demand. It is widely used in the software development industry. There are 'n' number of reasons for this.

High-level object-oriented programming language: Python includes effective symbolism.

Rapid application development: Because of its concise code and literal syntax, the development of applications gets accelerated. The reason for its wide usability is its simple and easy-to-master syntax. The simplicity of the code helps reduce the time and cost of development.

Dynamic typescript: Python has high-level incorporated data structures blended with dynamic typescript and powerful binding.

Features of Python:

- Python supports code reusability and modularity.
- It has a quick edit-inspect-debug cycle.
- Debugging is straightforward in Python programs.
- It has its own debugger written in Python itself, declaring to Python's reflective power.
- Python includes a plethora of third-party components present in the Python Package Index (PyPI).

MODULES

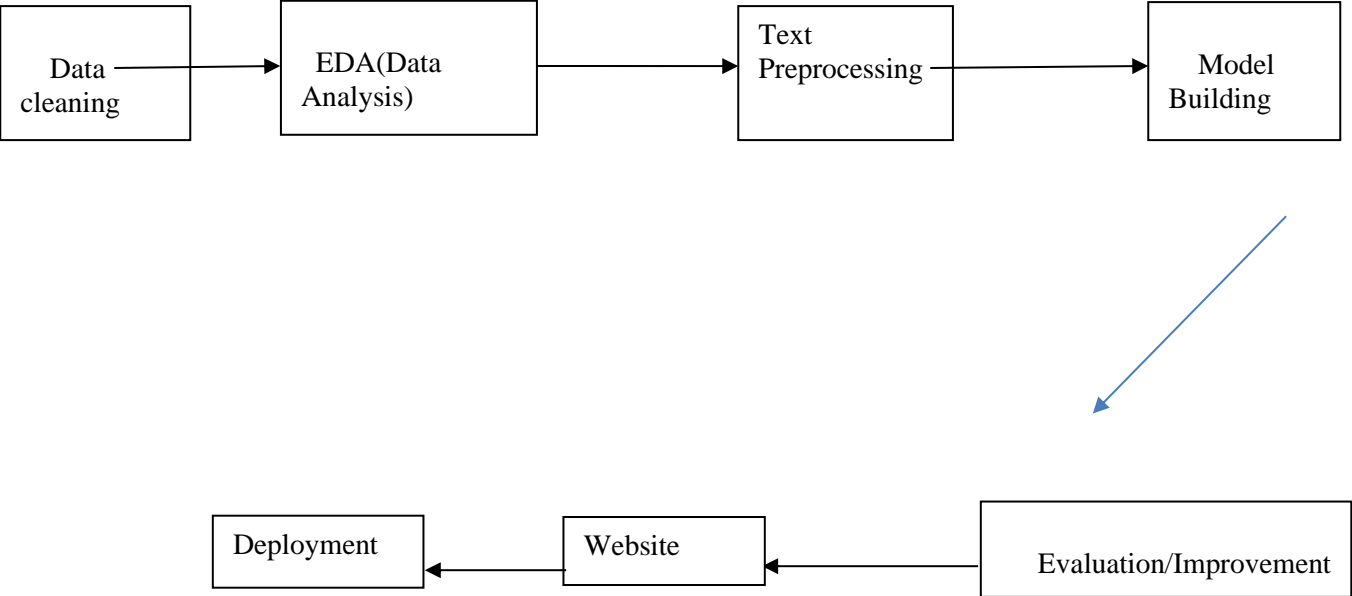
User Part:

- The User has to type the spam message in the Search Box.
- When the user enters the message, it will show whether its spam and ham.

The Project, Email/SMS spam classifier system can be divided into 5 modules, namely, **Collection of Data, Data Pre-processing, Model, Website, and Deployment.**

1. **Collection of Data:** The required data is collected from the UCI spam dataset.
2. **Data Pre-processing:** The collected is then pre-processed and manipulated to remove the unnecessary content, to meet the requirements of the Project.
3. **Model:** The pre-processed data is then again manipulated with the required tools and functions, so that it provides the required output.
4. **Website:** The prepared model is then converted to a website with the help of PyCharm Community platform and other required tools.
5. **Deployment:** The prepared website is then finally Deployed with the help of Sublime Text 3 And Stream lit.

Design of Project



Data Flow Diagram

Data Flow Diagram:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

Rules for creating DFD

- The name of the entity should be easy and understandable without any extra assistance(like comments).
- The processes should be numbered or put in ordered list to be referred easily.
- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes upto 9 and minimum 3 processes.

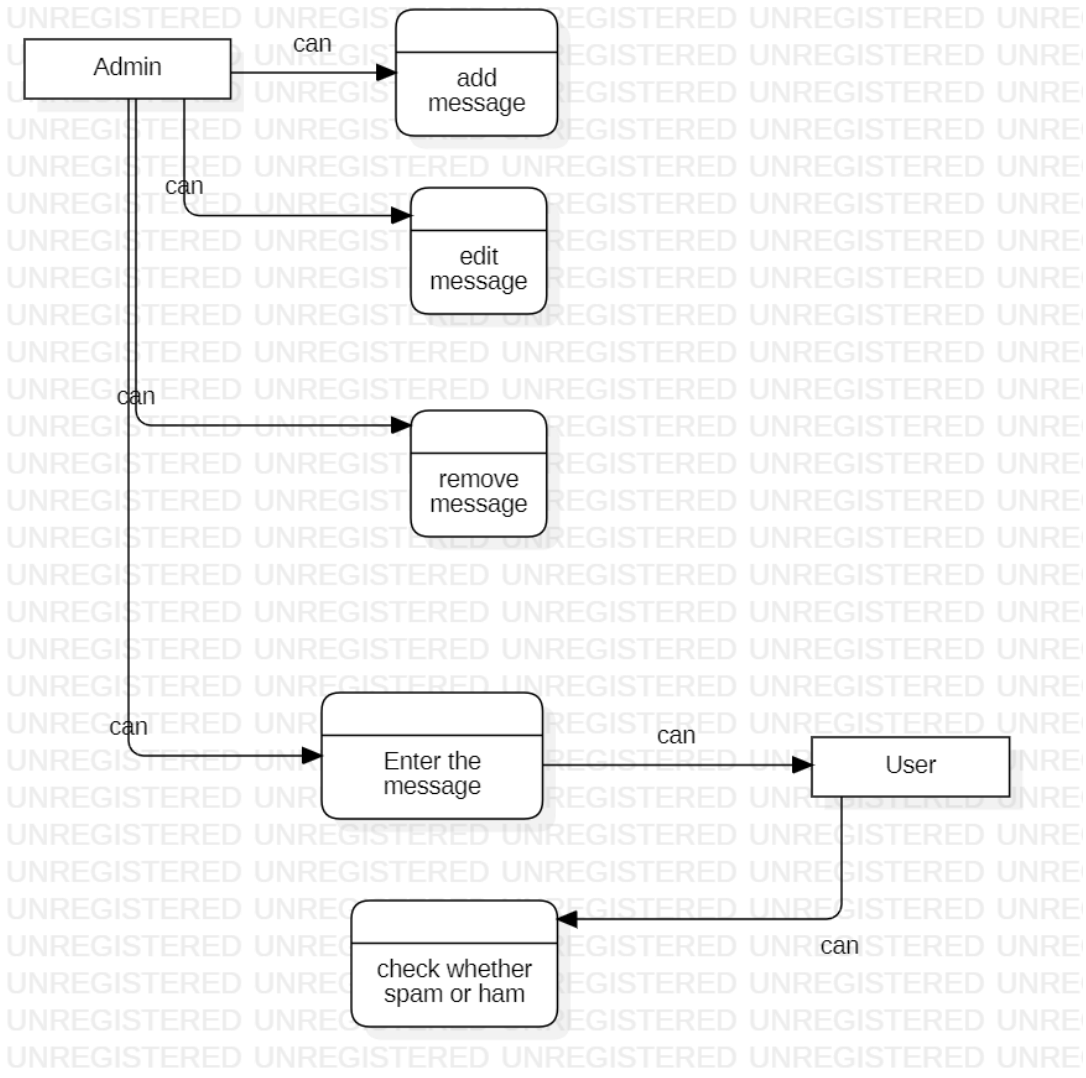
Advantages of DFD

- It helps us to understand the functioning and the limits of a system.

- It is a graphical representation which is very easy to understand as it helps visualize contents.
- Data Flow Diagram represent detailed and well explained diagram of system components.
- It is used as the part of system documentation file.
- Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.

Disadvantages of DFD

- At times DFD can confuse the programmers regarding the system.
- Data Flow Diagram takes long time to be generated, and many times due to this reasons analyst are denied permission to work on it.



Activity Diagram:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.

- Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements –

- Activities
- Association
- Conditions
- Constraints

Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

Activity diagram can be used for –

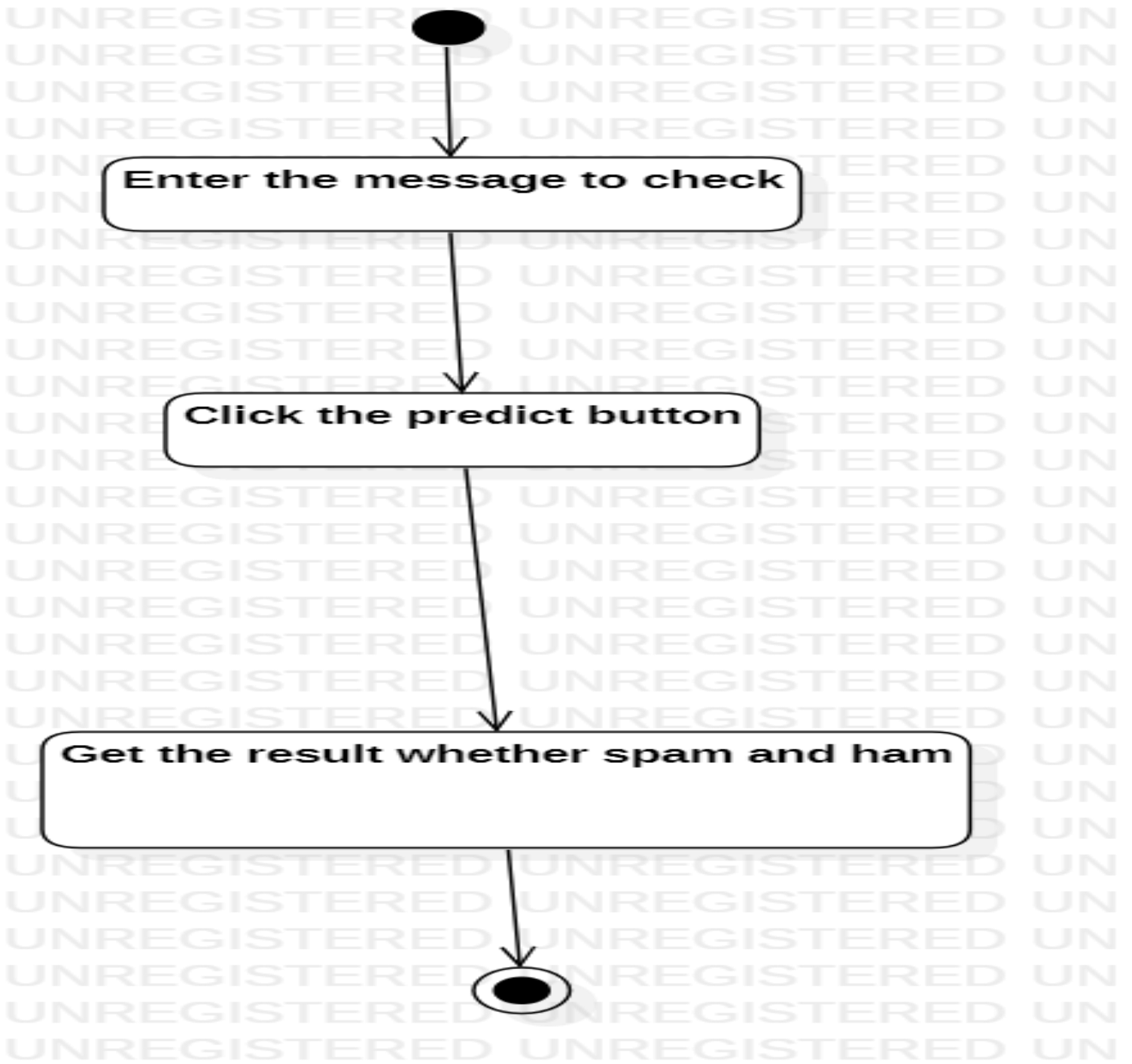
- Modeling work flow by using activities.
- Modeling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.



Enter the message to check

Click the predict button

Get the result whether spam and ham



Use Case Diagram:

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, a number of use case diagrams are used.

Purpose of Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements and actors.

How to Draw a Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

Functionalities to be represented as use case

- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system.

Where to Use a Use Case Diagram?

There are five diagrams in UML to model the dynamic view of a system. Now each and every model has some specific purpose to use. Actually these specific purposes are different angles of a running system.

To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

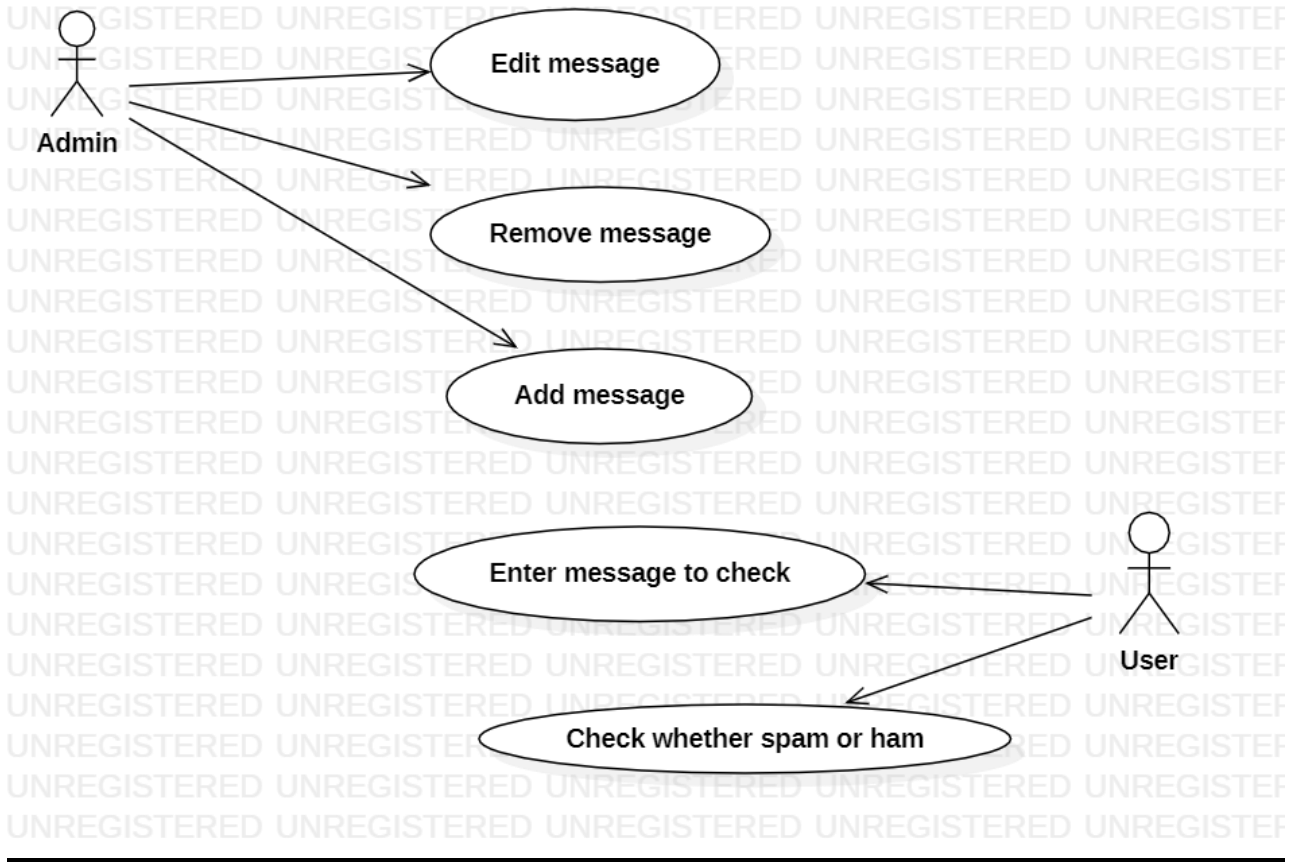
These diagrams are used at a very high level of design. This high-level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

Use case diagrams can be used for –

- Requirement analysis and high-level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.



TESTING

The **Test Cases** below give an idea of what result must be obtained on performing a particular task.

- **Pre-processing of Data:** The test case involved is to check whether the data is pre-processed as required or not.
- **Model for website:** The test case involved is to check whether the model prepared is returning the output as required or not.
- **Website Deployment:** The test case involved is to check whether the website is working as per need or not.

PRE-PROCESSING:

S.No.	Test Case	Excepted Result	Test Result
1	The Data is pre-processed according to the requirement.	The output should display the Pre-processed data.	Successful

MODEL:

S.No.	Test Case	Excepted Result	Test Result
1	The Model is prepared according to the requirement.	The output should display 1 for spam message and 0 for ham message.	Successful

WEBSITE DEPLOYMENT:

S.No.	Test Case	Excepted Result	Test Result
1	The Website is prepared according to the requirement.	The output should predict the spam or ham message .	Successful

OUTPUT

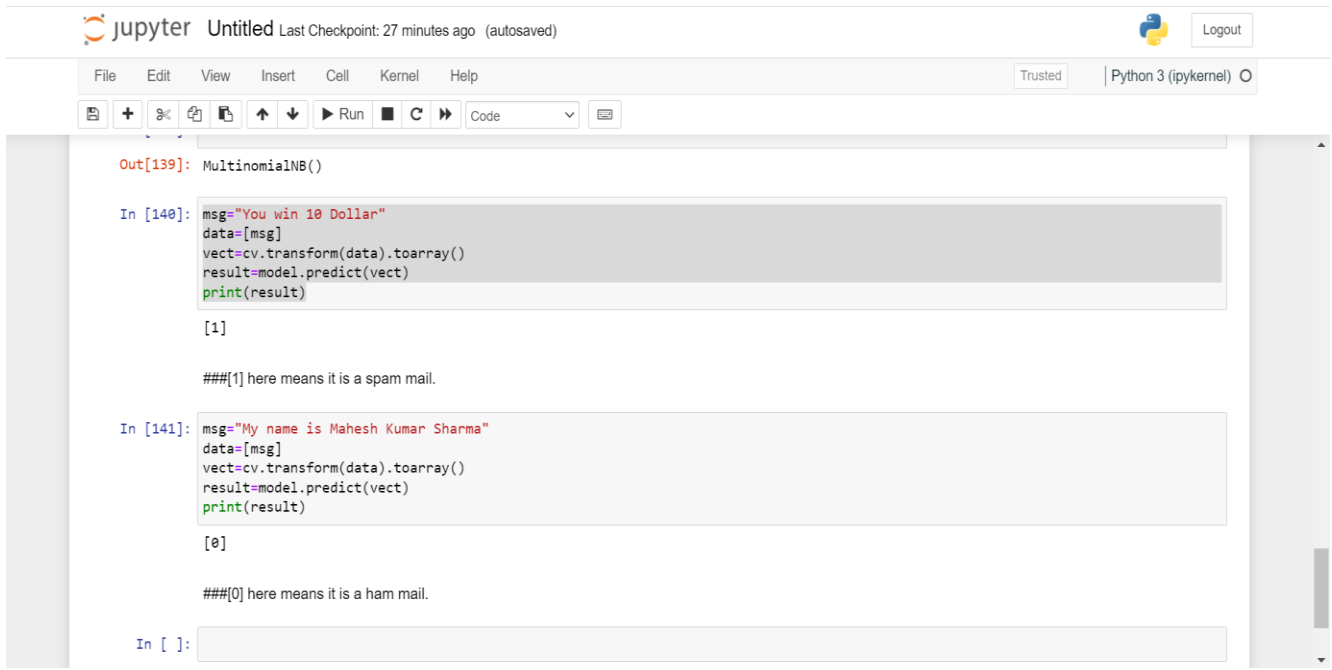
```
v1          v2  Unnamed: 2  Unnamed: 3  Unnamed: 4
0  ham  Go until jurong point, crazy.. Available only ...  NaN  NaN  NaN
1  ham                Ok lar... Joking wif u oni...  NaN  NaN  NaN
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...  NaN  NaN  NaN

In [114]: data.columns
Out[114]: Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')

In [115]: data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)

In [116]: data.head()
Out[116]:
   v1          v2
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...
```

- **The dataset has been pre-processed, and only useful data that is needed is kept in data set.**
- **V1= ham and spam**
- **V2=message**
- **Again we will convert ham =0 and spam =1, using nlp technique and store in sparse matrix format.**



Jupyter Untitled Last Checkpoint: 27 minutes ago (autosaved) Python 3 (ipykernel)

```
Out[139]: MultinomialNB()

In [140]: msg="You win 10 Dollar"
data=[msg]
vect=cv.transform(data).toarray()
result=model.predict(vect)
print(result)

[1]

###[1] here means it is a spam mail.

In [141]: msg="My name is Mahesh Kumar Sharma"
data=[msg]
vect=cv.transform(data).toarray()
result=model.predict(vect)
print(result)

[0]

###[0] here means it is a ham mail.

In [ ]:
```

- **Model is created and we can see the output .**
- **That when msg="You won a dollar“, result = 1 i.e spam**
- **And in second when msg="My name is Mahesh“, result=0 i.e ham.**

The screenshot shows a Jupyter Notebook with the following code and output:

```
In [50]: x.shape
Out[50]: (5572,)
```

```
In [51]: y.shape
Out[51]: (5572,)
```


```
In [52]: x=cv.fit_transform(x)
```

```
In [53]: x
Out[53]: <5572x8672 sparse matrix of type '<class 'numpy.int64''>
        with 73916 stored elements in Compressed Sparse Row format>
```

```
1. The Cat
2. The Dog
3. The Bird

The Cat Dog Bird
4. 1 1 0 0
5. 1 0 1 0
6. 1 0 0 1
```

The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar at the bottom showing system information like network speed, temperature, and time.

 Fig: The data has been converted into Compresses Sparse Row Format.

 What is a sparse row matrix?

A sparse matrix is **a matrix that is comprised of mostly zero values**. Sparse matrices are distinct from matrices with mostly non-zero values, which are referred to as dense matrices.

```
C:\Users\hp\machine-learning-project\MaheshspamDetector.py (hp) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
  hp
  .android
  .gradle
  .idlerc
  .ipymb_checkp
  .ipython
  .jupyter
  .streamlit
  .vscode
  3D Objects
  AndroidStudi
  AppData
  Application D
  bluej
  Contacts
  Cookies
  Desktop
  Documents
  Downloads
  Favorites
  hello
  IBA_IOAPDAT
  IntelGraphicsf
  Links
  Local Settings
  machine-learn
  Music
  My Document
  Mahesh

MaheshspamDetector.py x
1  import pickle
2  import streamlit as st
3
4
5  model=pickle.load(open("spam.pkl", "rb"))
6  cv=pickle.load(open("vectorizer.pkl", "rb"))
7
8
9  def main():
10     st.title("Email/SMS Spam Classifier Website")
11     st.subheader(":Made By Mahesh With Python & Streamlit")
12     msg=st.text_input("Enter the Text : ")
13     if st.button("Predict"):
14         data=[msg]
15         vect=cv.transform(data).toarray()
16         prediction=model.predict(vect)
17         result=prediction[0]
18         if result==1:
19             st.error("This is Spam Message")
20         else:
21             st.success("This a Ham Message")
22
23     main()

Line 21, Column 43      7%      Tab Size: 4      Python
```

Fig: Website Deployment using Sublime Text.

Here, We have used streamlit module to convert our pickle file “spam.pkl” and opened it in read mode, and then make a main function in which we make the heading of our file;

“Email/SMS spam classifier” and sub heading “made by Mahesh with streamlit and python.”

Then we have added st.error: which we show red message when we have the spam message

And st.success : which shows green message when we have ham message.


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\machine-learning-project>streamlit run MaheshspamDetector.py
```

Fig: We use cmd to run streamlit and a local host will be provided.

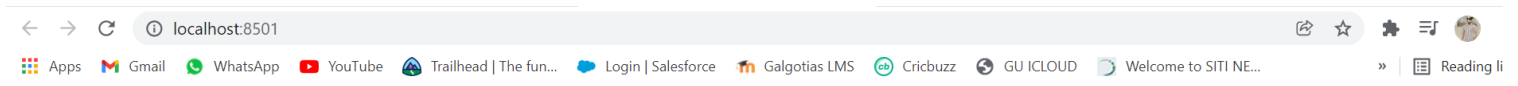
```
C:\Windows\System32\cmd.exe - streamlit run MaheshspamDetector.py
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\machine-learning-project>streamlit run MaheshspamDetector.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.29.183:8501
```

Fig: Now this local host can be used for viewing the website(MaheshspamDetection.py).



Email/SMS Spam Classifier Website

:Made By Mahesh With Python & Streamlit

Enter the Text :

Predict

Fig: After the loading of host , this is the first look of our website.

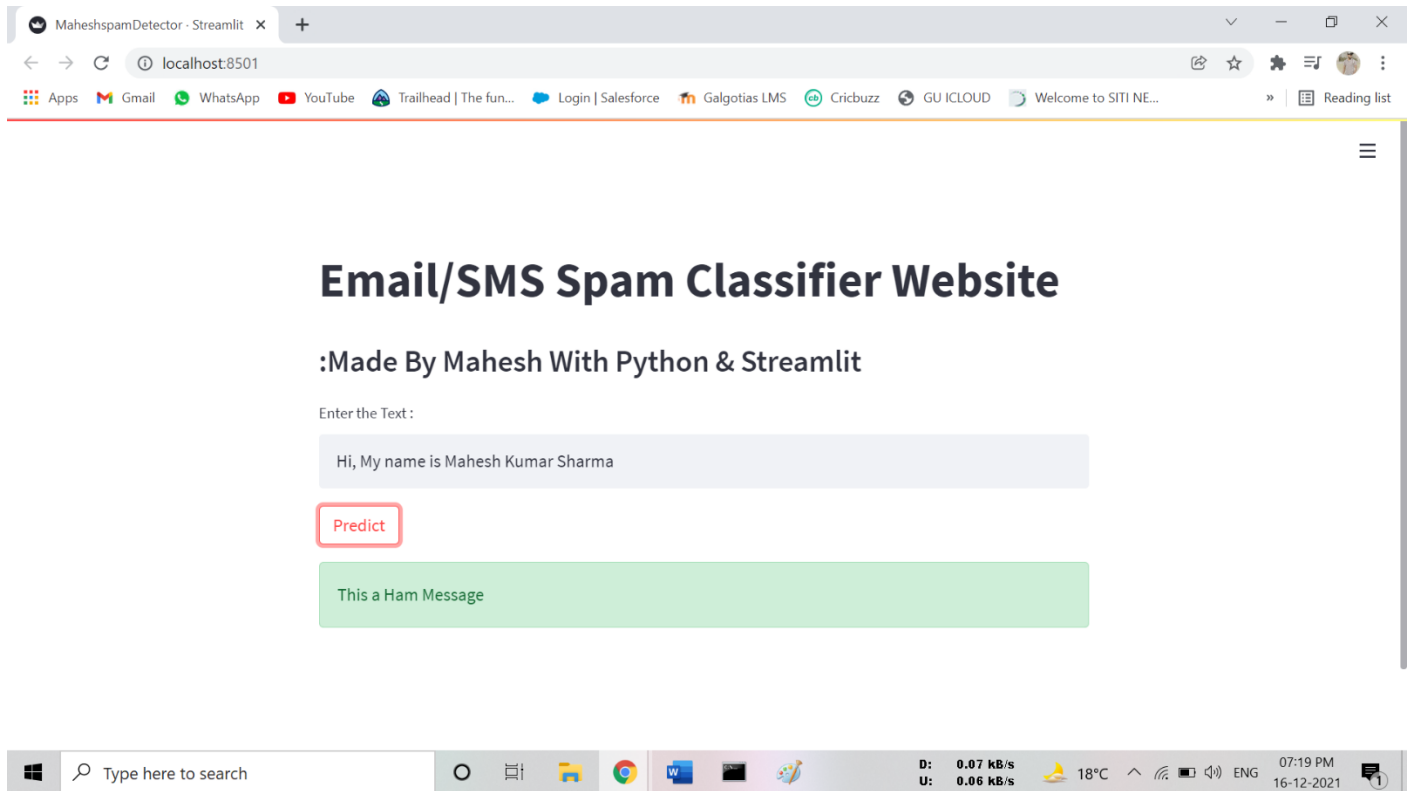


Fig: Successful detection of Ham message(Shown in green!!)

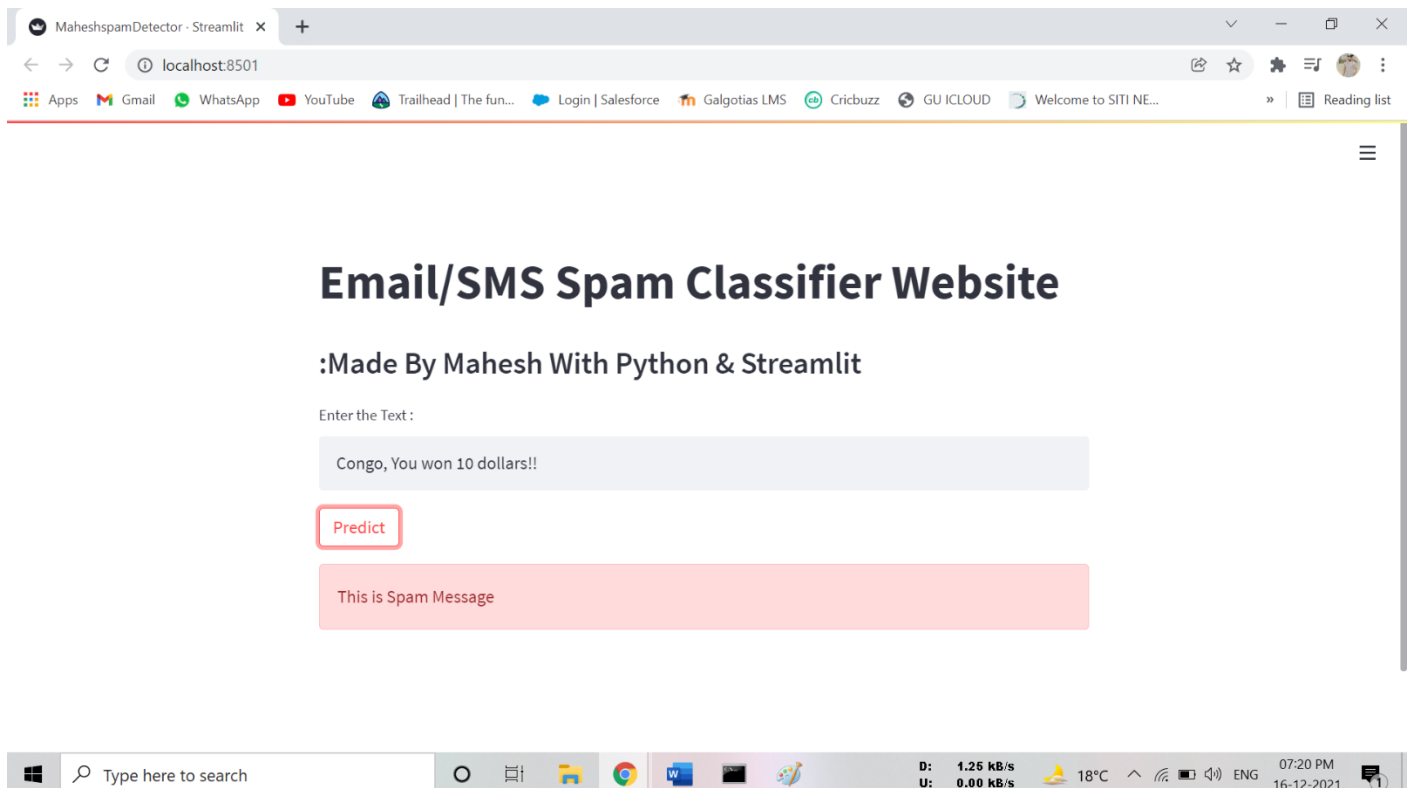


Fig: Successful prediction of Spam Message(Shown in red colour!!)