

TSWR DEGREE-COLLEGE[W] JAGATHGIRIGUTTA



A Project Report On

SPAM DETECTION ON MOBILE

Submitted in partial fulfillment of the requirements for award of the
degree of

BACHELOR OF SCIENCE

IN

DATA SCIENCE

By

B. Lavanya (281220539012)

Under the guidance of

Mrs. Sangeetha Madam

(Department of Data Science)

TSWRDC-W Jagathgirigutta

TSWRDC(W)JAGATHGIRIGUTTA
DEPARTMENT OF DATA SCIENCE

ACKNOWLEDGEMENT

I would like to take this chance to express my heartfelt gratitude to everyone who assisted, encouraged, motivated, and cooperated in various ways during my project work. It gives me great pleasure to express gratitude to everyone who contributed to the successful conclusion of project.

I would like to thank and express my admiration and respect to project guide **Sangeetha Madam** for the guidance, cooperation, and encouragement throughout the development of this project. Last but not least, I would like to appreciate all of the respondents for their assistance in all circumstances.

BY

B. Lavanya (281220539012)

TSWRDC(W)JAGATHGIRIGUTTA
DEPARTMENT OF DATA SCIENCE



DECLARATION BY CANDIDATE

I attest that I have submitted the project titled “**SPAM DETECTION ON MOBILE**” as part of the requirement for the award of the Bachelor of Technology degree in Data Science. I confirm that this dissertation is my original work and has not been used as a basis for any other academic qualifications, awards, or publications, nor has any part of it been previously published or submitted for publication.

BY

B. Lavanya (281220539012)

HEAD OF THE DEPARTMENT

(DATASCIENCE)

MRS.NUTHANA

DATASCIENCE FACULTY

MRS.SANGEETHA

PRINCIPAL

DR.ISHRATH

EXTERNAL GUIDE

TSWRDC(W)JAGATHGIRIGUTTA
DEPARTMENT OF DATA SCIENCE



CERTIFICATE

This is to certify that the project report entitled **“SPAM DETECTION ON MOBILE”**, being submitted by **BADDAM LAVANYA (281220539012)** in partial fulfilment of the requirement for the award of the degree of Bachelor of Science in

MATHEMATICS, STATISTICS AND DATA SCIENCE is a record of bonafide work carried out by them. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

HEAD OF THE DEPARTMENT

(DATASCIENCE)

MRS.NUTHANA

DATASCIENCE FACULTY

MRS.SANGEETHA

PRINCIPAL

DR.ISHRATH

EXTERNAL GUIDE

TSWRDC(W)JAGATHGIRIGUTTA
DEPARTMENT OF DATA SCIENCE



CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that the project report entitled “**SPAM DETECTION ON MOBILE**”, being submitted by **BADDAM LAVANYA (281220539012)** in partial fulfilment of the requirement for the award of the degree of Bachelor of Science in **MATHEMATICS, STATISTICS AND DATA SCIENCE** is a record of bonafide work carried out by them.

HEAD OF THE DEPARTMENT

(DATASCIENCE)

MRS.NUTHANA

DATASCIENCE FACULTY

MRS.SANGEETHA

TABLE OF CONTENTS

1.Introduction

1.1 Background and Motivation

1.2 Problem Statement

1.3 Objectives

1.4 Scope and Limitations

2.Literature Review

2.1 Overview of Spam Detection Techniques

2.2 Existing System

2.3 Proposed System

3.Data Collection and Preprocessing

3.1 Data Collection

3.2 Data Preprocessing

4.Feature Extraction

4.1 F-P Growth

5.Software Requirements

6.Hardware Requirements

7.Design

7.1 Introduction

7.2 Architecture Diagram

7.3 Unified Modelling Learning

7.4 Building Blocks of UML Diagrams

7.5 UML Diagrams for Spam Detection on Mobile

8.Evaluation and Results

9.Conclusion and Future Work

ABSTRACT

SPAM DETECTION ON MOBILE

SMS (Short Message Service) is still the primary choice as a communication medium even though nowadays mobile phones are growing with a variety of communication media messenger applications. However, nowadays along with the SMS tariff reduction leads to the increase of SMS spam, as used by some people as an alternative to advertise and fraud. Therefore, it becomes an important issue as it can bug and harm the users and one of its solutions is with automatic SMS spam filtering. One of the most challenging in SMS spam filtering is its accuracy. In this research we proposed to enhance SMS spam filtering performance by combining two of data mining task association and classification. FP-growth in association is utilized for mining frequent pattern on SMS and Naive Bayes Classifier is used to classify whether SMS is spam or ham. Training data was using SMS spam collection from previous research. The result of using Collaboration of Naive Bayes and FP-Growth performs the highest average accuracy of 90%. FP-Growth for dataset SMS Spam Collection and improves the precision score; thus, the classification result is more accurate. In theory, spam detection can be implemented at any location and multiple stages of process can occur at the same time

1. Introduction

1.1. BACKGROUND AND MOTIVATION

In every organization, every product that needs to be tested before making it available to the society. So, we have to maintain the details of test process. In the present project “Spam Detection on Mobile” we are able to keep track all the testing details of bugs occurred in different projects. In the standalone system, it is some more difficult to maintain different databases on every system. So, in every branch we have to install the software and maintain a database individually for tracking the bugs occurred in that branch.

1.2 PROBLEM STATEMENT

- i. SMS is a text-based communication media that allows mobile phone users to share a short text. Along with the widespread use and popularity as the most important communications media, there are plenty of those who use it for commercial purposes such as advertising media and even fraud.
- ii. The reduced SMS rate is one of the causes of increasing SMS spam. When we receive any SMS/Mails it may be either HAM (Important messages) or SPAM (least important messages). But sometimes SPAM messages divert our mind. So, we need such system which can either block the SPAM message as either it moves the SPAM message into different folder without disturbing USER.

1.3. OBJECTIVES

- i. There are four objectives that need to be achieved in this project:
- ii. To study on how to use machine learning techniques for spam detection.
- iii. To modify machine learning algorithm in computer system settings.
- iv. To leverage modified machine learning algorithm in knowledge analysis software.
- v. To test the machine learning algorithm real data from machine learning data repository.

1.4. SCOPE AND LIMITATIONS

Scope –

Spam detection on mobile devices refers to the identification and filtering of unsolicited messages, usually sent through SMS or messaging apps. The scope of spam detection on mobile devices includes identifying and blocking spam messages, preventing fraudulent and malicious messages, and reducing the overall number of unwanted messages received by mobile device users.

Limitations -

- i. Limited processing power
- ii. Limited data storage
- iii. Limited network connectivity
- iv. Diverse messaging platforms
- v. Privacy concerns
- vi. False positives

2.LITERATURE REVIEW

2.1. OVERVIEW OF SPAM DETECTION TECHNIQUES

Spam detection techniques are used to identify and filter unsolicited messages, emails, or other forms of unwanted communication. Here is an overview of some common spam detection techniques:

- i. **Rule-Based Filtering:** Uses predefined rules to identify spam based on criteria such as sender's address, keywords, or specific phrases.
- ii. **Content-Based Filtering:** Analyzes message content to identify common characteristics of spam using machine learning algorithms.
- iii. **Blacklisting/Whitelisting:** Blocks messages from known sources of spam (blacklisting) or allows messages from trusted sources to pass through (whitelisting).
- iv. **Heuristics-Based Filtering:** Uses machine learning algorithms to analyze message content and identify patterns and characteristics commonly found in spam messages.
- v. **Bayesian Filtering:** Uses probability-based algorithms to classify messages based on their similarity to previously identified spam and non-spam messages.
- vi. **Sender Authentication:** Verifies the authenticity of the sender's domain and prevents spoofing of email addresses using techniques such as DKIM and SPF.

2.2. EXISTING SYSTEM

Existing systems for spam detection on mobile devices, including Google's Spam Protection for Messages, True caller, SMS Spam Filter, Apple's Built-in Spam Detection, and SMS Organizer. These systems use various techniques such as rule-based filtering, content-based filtering, machine learning algorithms, blacklisting/whitelisting, and sender authentication to identify and filter spam messages on mobile devices. Users can choose from a range of options based on their specific needs and preferences, including custom filters and blocklists, and some systems also allow users to report spam messages to improve their effectiveness over time.

2.3. PROPOSED SYSTEM

We are using Machine Learning algorithm (Naïve Bayes Algorithm) to eradicate such problem. In this algorithm model will train the machine by its 70% and 30% of dataset. Through this 70% data our machine will be trained enough to decide which is the SPAM message or which is the HAM message.

3. DATA-COLLECTION AND DATA-PREPROCESSING

3.1. DATA-COLLECTION

In this project, existing machine learning algorithm is used and modified to fit the need of project. The reasons are because machine learning algorithm is adept at reviewing large volume of data. It is typically improving over time because of the ever-increasing data that are processed. It gives the algorithm more experience and be used to make better predictions.

Machine learning allows for instantaneous adaption without human intervention. It identifies new threats and trends and implements the appropriate measures. It is also saving time as it is its automated nature.

- i. Data collection involves gathering message data and metadata from various sources, such as messaging apps or email clients.
- ii. The collected data may include message content, sender information, receiver information, message timestamps, and other relevant metadata.
- iii. The data collection process may be automated or user-initiated, depending on the specific system and user preferences.
- iv. Some spam detection systems may also collect data from external sources, such as global spam databases or user feedback mechanisms.
- v. The collected data is then preprocessed, transformed, and analyzed using various techniques such as rule-based filtering, content-based filtering, machine learning algorithms, and sender authentication to identify and filter spam messages.

3.2. DATA-PREPROCESSING

- i. Data preprocessing is a critical step in preparing collected message data and metadata for analysis to identify and filter spam messages.
- ii. Common preprocessing steps include cleaning and transforming the data by removing unwanted characters or symbols, converting text to lowercase, and removing stop words.
- iii. The data may also be normalized or scaled to ensure that all features have the same range of values and to improve the accuracy of machine learning models.
- iv. Feature selection or dimensionality reduction techniques may be used to reduce the number of features used in analysis, which can improve computational efficiency and prevent overfitting.
- v. Once the data has been preprocessed, it is ready for analysis using various techniques such as rule-based filtering, content-based filtering, or machine learning algorithms to identify and filter spam messages.

4. FEATURE EXTRACTION

4.1. F-P GROWTH

Frequent pattern growth (F-P Growth) is a data mining algorithm used for feature extraction in spam detection on mobile devices. It constructs a frequent pattern tree from preprocessed message data, and extracts the most frequent patterns in the data, such as specific words or phrases commonly found in spam messages. F-P Growth can identify complex patterns that are difficult to detect using other techniques but may require significant processing power. Extracted features are used as input for spam filtering techniques, such as rule-based filtering or machine learning algorithms, to identify and filter spam messages on mobile devices.

5. SOFTWARE REQUIREMENTS

- Operating System - Windows 11
- Database - CSV File, TSV File
- Programming Language - Python
- IDE - Jupiter, Python 3.7

6. HARDWARE REQUIREMENTS

- System : Intel CORE i5
- Hard Disk : 2 TB
- Screen : 15 VGA Color.
- Ram : 16GB.

7.DESIGN

7.1. INTRODUCTION

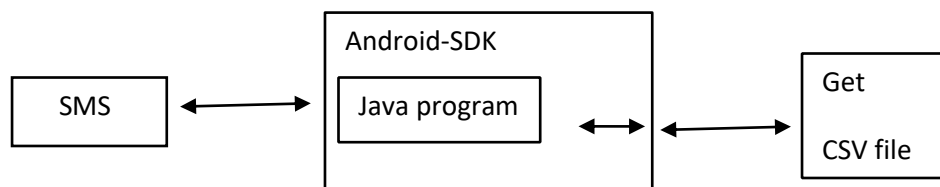
Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming, as in a stylized software engineering process." Software design usually involves problem solving and planning a software solution. This includes both a low-level component design and a high-level, architecture design.

7.2. ARCHITECTURE DIAGRAM

Architecture diagram is a diagram of a system, in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.

A SMS user for who the application looks like a user interface actually consists of a database called as SQLite that comes along with Android SDK and need no other installation. This is the database that is used to store and retrieve information. This is an application that is developed in java and hence all its features apply here as well such as platform independence, data hiding.

Fig 6.2 Architecture



7.3. UNIFIED MODELING LANGUAGE (UML)

The unified modeling is a standard language for specifying, visualizing, constructing and documenting the system and its components is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure and control information about the systems.

Depending on the development culture, some of these artifacts are treated more or less formally than others. Such artifacts are not only the deliverables of a project; they are also critical in controlling, measuring, and communicating about a system during its development and after its deployment.

The UML addresses the documentation of a system's architecture and all of its details. The UML also provides a language for expressing requirements and for tests. Finally, the UML provides a language for modeling the activities of project planning and release management.

7.4. BUILDING BLOCKS OF UML

The vocabulary of the UML encompasses three kinds of building blocks:

- i. Things
- ii. Relationships
- iii. Diagrams

Things are the abstractions that are first-class citizens in a model, relationships tie these things together and diagrams group interesting collections of things.

I. THINGS IN THE UML

There are four kinds of things in the UML:

1. Structural things
2. Behavioral things
3. Grouping things
4. A Notational things

1.STRUCTURAL THINGS are the nouns of UML models. The structural things used in the project design are:

First, a **class** is a description of a set of objects that share the same attributes, operations, relationships and semantics.

| |
|------------|
| Window |
| Origin |
| Size |
| open () |
| close () |
| move () |
| display () |
| |

Fig: Classes

Second, a **use case** is a description of set of sequence of actions that a system performs that yields an observable result of value to particular actor.

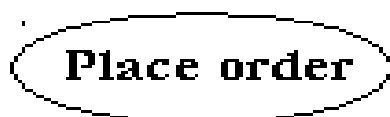


Fig: Use Cases

Third, a node is a physical element that exists at runtime and represents a computational resource, generally having at least some memory and often processing capability.

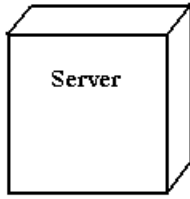


Fig: Nodes

2. BEHAVIOURAL THINGS are the dynamic parts of UML models. The behavioral thing used is:

INTERACTION: An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose. An interaction involves a number of other elements, including messages, action sequences (the behavior invoked by a message, and links (the connection between objects).

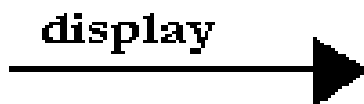


Fig: Messages

3.RELATIONAL IN THE UML

There are four kinds of relationships in the UML:

- i. Dependency
 - ii. Association
 - iii. Generalization
 - iv. Realization
- i. A **dependency** is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing (the dependent thing).



Fig: Dependencies

- ii. An **association** is a structural relationship that describes a set links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts.

Fig: Association

- iii. A **generalization** is a specialization/ generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).

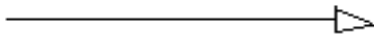


Fig: Generalization

- iv. A **realization** is a semantic relationship between classifiers, where in one classifier specifies a contract that another classifier guarantees to carry out.

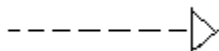


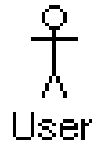
Fig: Realization

SEQUENCE DIAGRAMS

UML sequence diagrams are used to represent the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram.

Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

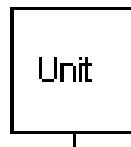
ACTOR Represents an external person or entity that interacts with the system



OBJECT Represents an object in the system or one of its components



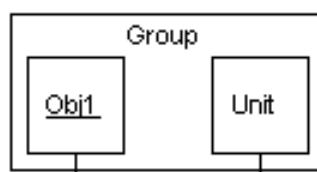
UNIT Represents a subsystem, component, unit, or other logical entity in the system (may or may not be implemented by objects)



SEPERATOR Represents an interface or boundary between subsystems, components or units (e.g., air interface, Internet, network)



GROUP Groups related header elements into subsystems or components

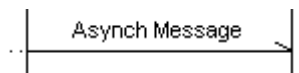


---SEQUENCE DIAGRAM BODY ELEMENTS---

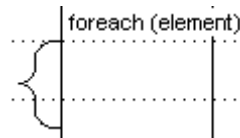
ACTION Represents an action taken by an actor, object or unit



ASYNCHRONOUS MESSAGE An asynchronous message between header elements



BLOCK A block representing a loop or conditional for a particular header element



CALL MESSAGE A call (procedure) message between header elements



CREATE MESSAGE A "create" message that creates a header element (represented by lifeline going from dashed to solid pattern)

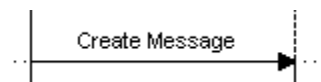
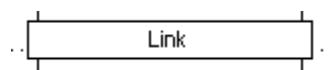
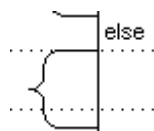


DIAGRAM LINK Represents a portion of a diagram treated as a functional block. Similar to a procedure or function call that abstracts functionality or details not shown at this level and can be an optional link to another diagram for elaboration.



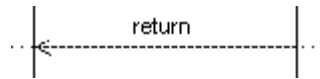
Else Block Represents an "else" block portion of a diagram block



MESSAGE A simple message between header elements

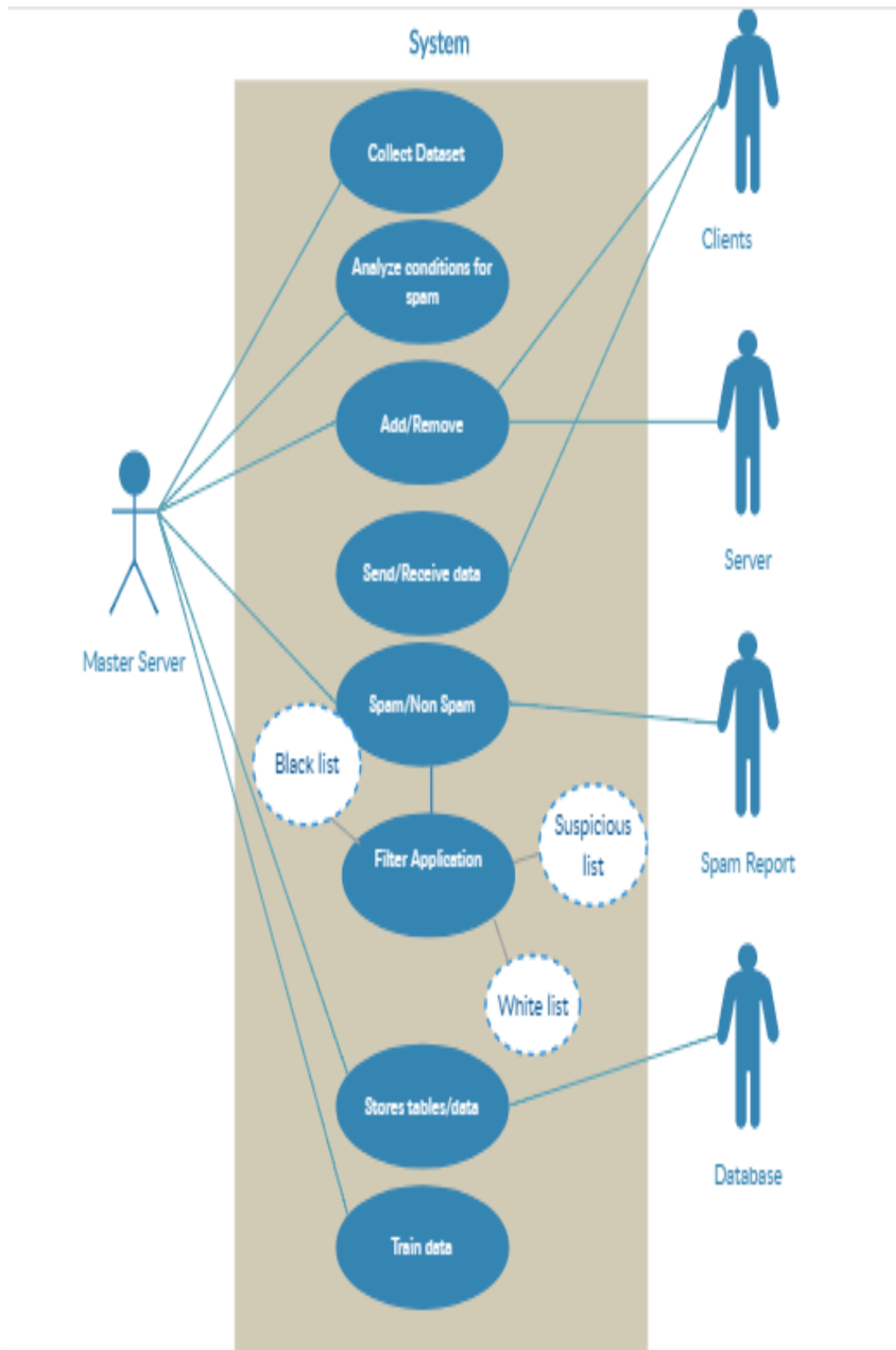


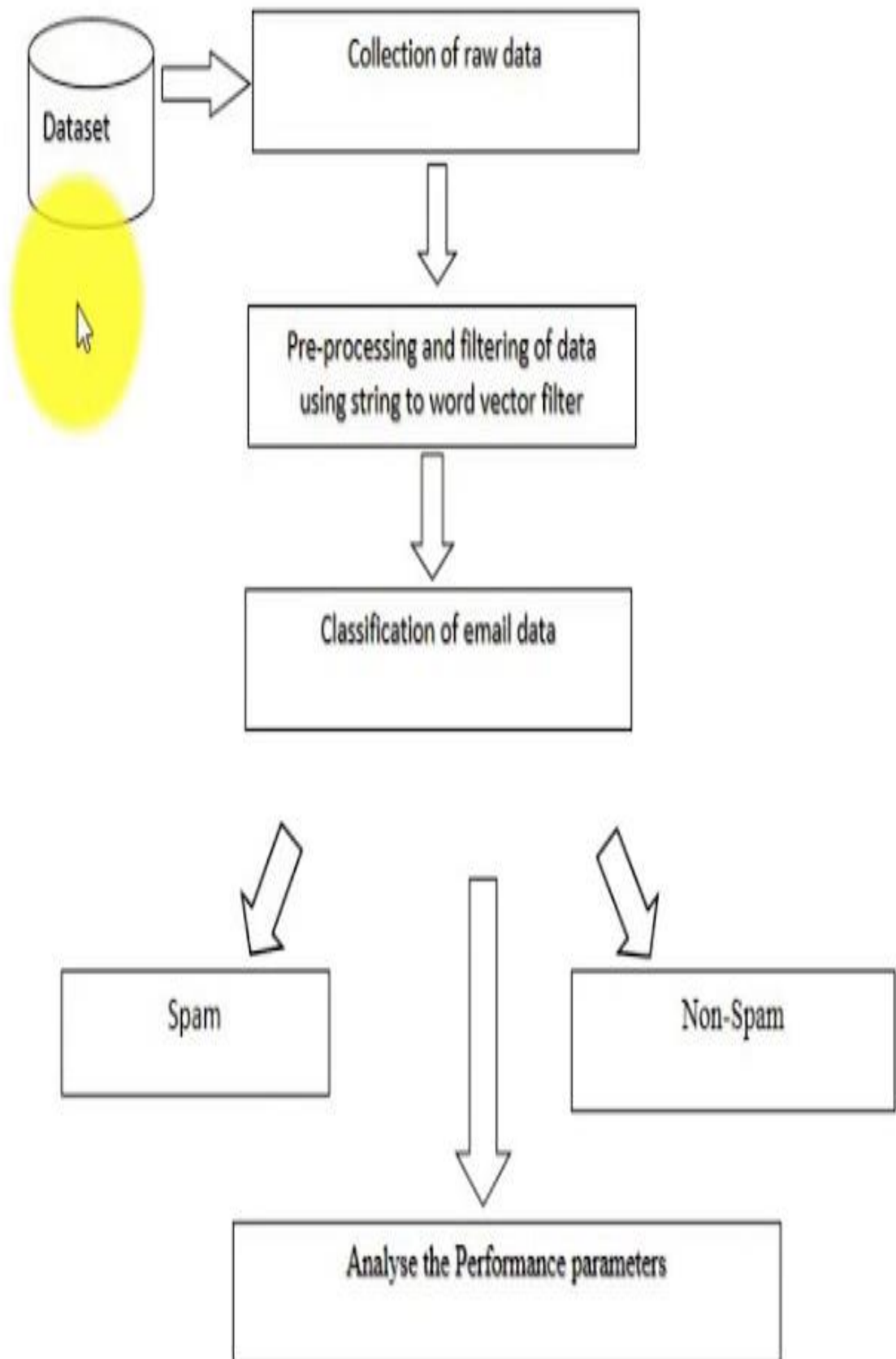
RETURN MESSAGE A return message between header elements




7.5.UML DIAGRAMS FOR SPAM DETECTION ON MOBILE

---USE CASE DIAGRAM---





8.EVALUATION AND RESULTS

Jupyter SMS Spam Detection (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [1]:


```
# importing modules
import pandas as pd      # its used for dataframe
import numpy as np       # its used for n dimensional array
import matplotlib.pyplot as plt  # its used for plotting the graph
```

In [2]:

```
#read csv file
data=pd.read_csv("spam.csv",encoding="latin_1")
data.head()
```

Out[2]:

| | 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 |
| 3 | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

Jupyter SMS Spam Detection Last Checkpoint: Last Thursday at 5:57 PM (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [7]:

```
#column names recharge
data = data.rename(columns={"v1":"label", "v2":"text"})
data.head()
```

Out[7]:

| | label | text |
|---|-------|---|
| 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... |

In [8]:

```
#Count observations in each Label
data.label.value_counts()
```

Out[8]:

```
ham    4826
spam    747
Name: label, dtype: int64
```

In [9]:

```
# convert Label to a numerical variable
```



```
In [9]: # convert label to a numerical variable
data['label_num'] = data.label.map({'ham':0, 'spam':1})
```

```
In [10]: data.head()
```

Out[10]:

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



```
In [11]: data['length'] = data['text'].apply(len)
data.head()
```

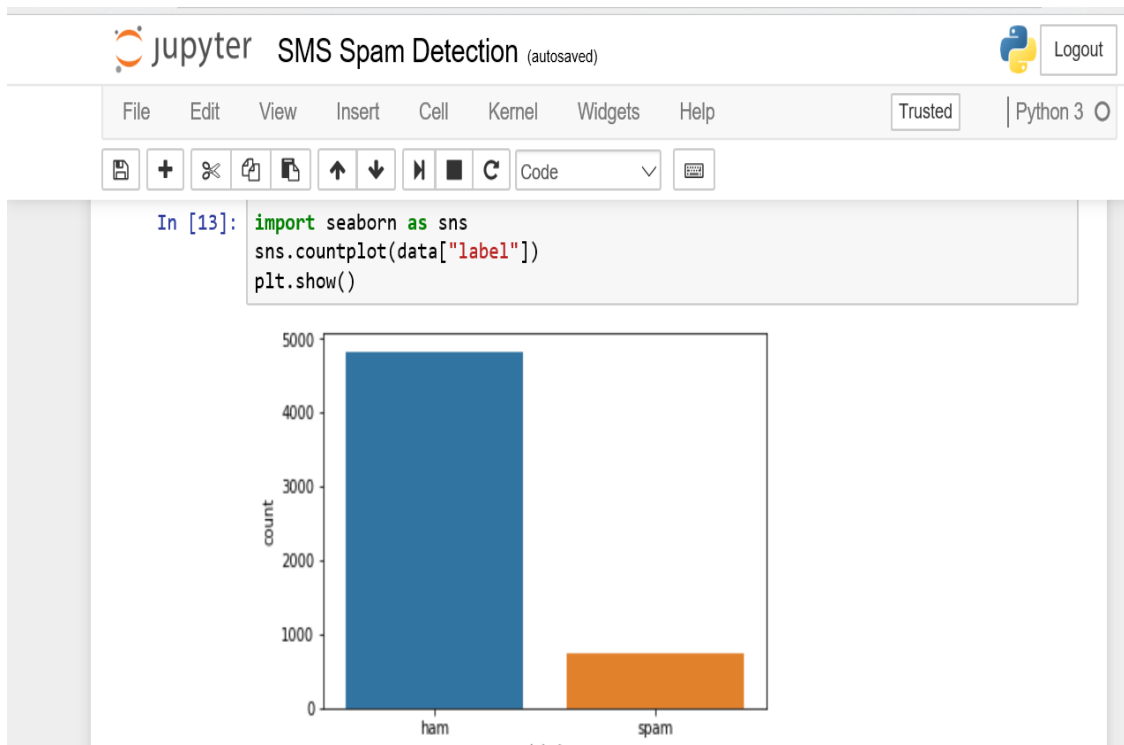
Out[11]:

| | label | text | label_num | length |
|---|-------|---|-----------|--------|
| 0 | ham | Go until jurong point, crazy.. Available only ... | 0 | 111 |
| 1 | ham | Ok lar... Joking wif u oni... | 0 | 29 |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... | 1 | 155 |
| 3 | ham | U dun say so early hor... U c already then say... | 0 | 49 |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | 0 | 61 |

```
In [11]: x=np.array(data['text'])
x
data.shape
```

Out[11]: (5573, 4)

```
In [12]: y = np.array(data['label_num'])
y
```



Jupyter SMS Spam Detection (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

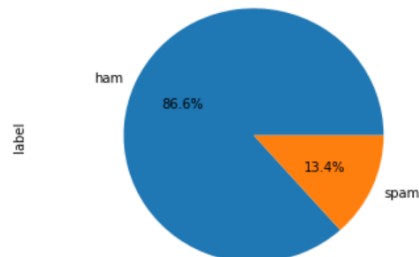
```
In [15]: spam1=data.loc[data['label']=='spam']
spam1["text"].head()
```

Out[15]: 2 Free entry in 2 a wkly comp to win FA Cup fina...
5 FreeMsg Hey there darling it's been 3 week's n...
8 WINNER!! As a valued network customer you have...
9 Had your mobile 11 months or more? U R entitle...
11 SIX chances to win CASH! From 100 to 20,000 po...
Name: text, dtype: object

```
In [16]: ham1=data.loc[data['label']=='ham']
ham1["text"].head()
```

Out[16]: 0 Go until jurong point, crazy.. Available only ...
1 Ok lar... Joking wif u oni...
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...
6 Even my brother is not like to speak with me. ...

```
In [14]: data["label"].value_counts().plot(kind="pie", autopct="%1.1f%%")
plt.axis("equal")
plt.show()
```



```
In [23]: x_train=np.array(data.iloc[0:500,1])
x_train.shape
```

```
Out[23]: (500,)
```

```
In [24]: y_train=np.array(data.iloc[0:500,0])
y_train[0:5]
```

```
Out[24]: array(['ham', 'ham', 'spam', 'ham', 'ham'], dtype=object)
```

```
In [19]: from sklearn.model_selection import train_test_split
```

```
In [20]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

```
In [21]: from sklearn.feature_extraction.text import CountVectorizer
count_vector = CountVectorizer()
print(count_vector)
```

```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
lowercase=True, max_df=1.0, max_features=None, min_df=1,
ngram_range=(1, 1), preprocessor=None, stop_words=None,
strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
tokenizer=None, vocabulary=None)
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3



```
In [22]: train_data = count_vector.fit_transform(x_train)
        test_data = count_vector.transform(x_test)
```

```
In [23]: from sklearn.naive_bayes import MultinomialNB
```

```
In [24]: model=MultinomialNB()
        model.fit(train_data,y_train)
```

```
Out[24]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
In [25]: pred=model.predict(test_data)
        pred
```

```
Out[25]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3



```
In [26]: model.score(test_data,y_test)
```

```
Out[26]: 0.98385167464114831
```

```
In [27]: from sklearn.metrics import classification_report
        nbreport=classification_report(y_test, pred)
        print(nbreport)
```

| | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0 | 0.99 | 0.99 | 0.99 | 1450 |
| 1 | 0.95 | 0.92 | 0.94 | 222 |
| avg / total | 0.98 | 0.98 | 0.98 | 1672 |



```
In [28]: from sklearn.metrics import f1_score
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
```

```
In [29]: y=[f1_score(y_test,pred),recall_score(y_test,pred),precision_score(y_test,pred)]
```

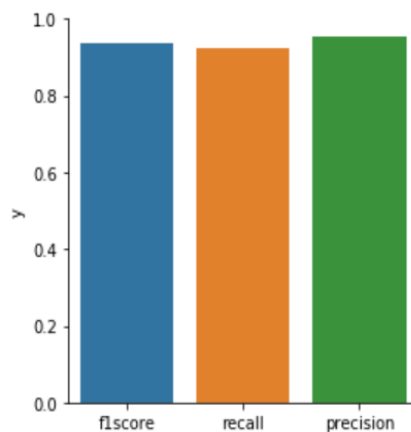
```
In [30]: x=["f1score", "recall", "precision"]
y=[f1_score(y_test,pred),recall_score(y_test,pred),precision_score(y_test,pred)]
df = pd.DataFrame(dict(x=x, y=y))
df
```

Out[30]:

| | x | y |
|---|-----------|----------|
| 0 | f1score | 0.938215 |
| 1 | recall | 0.923423 |
| 2 | precision | 0.953488 |

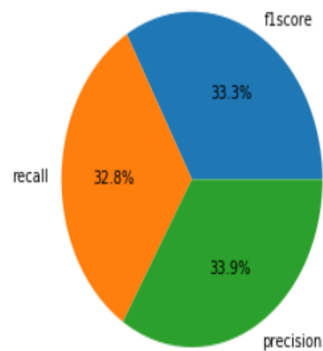


```
In [31]: sns.factorplot("x", "y", data=df, kind="bar")
plt.show()
```





```
In [33]: plt.pie(y, labels=x, autopct='%1.1f%%')
plt.axis("equal")
plt.show()
```



```
In [34]: #creating testing data
x_test=[ "hi how are you",
         "Free entry in 2 a wkly comp to win FA Cup fina...",
         "when will you go to home",
         "i will call you back",
         "are you busy now"]
```

```
In [35]: x_test.append("goodmoring")
x_test.append("WINNER!! As a valued network customer you have...")
```

```
In [36]: x_test
```

```
Out[36]: ['hi how are you',
          'Free entry in 2 a wkly comp to win FA Cup fina...',
          'when will you go to home',
          'i will call you back',
          'are you busy now',
          'goodmoring',
          'WINNER!! As a valued network customer you have...']
```

Jupyter SMS Spam Detection Last Checkpoint: Last Thursday at 5:57 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [37]: x_test1=np.array(x_test)
x_test1

Out[37]: array(['hi how are you',
               'Free entry in 2 a wkly comp to win FA Cup fina...',
               'when will you go to home', 'i will call you back',
               'are you busy now', 'goodmoring',
               'WINNER!! As a valued network customer you have...'],
              dtype='<U49')
```

```
In [38]: X_train=data.iloc[0:200,1]
X_train[0:6]

Out[38]: 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...
5    FreeMsg Hey there darling it's been 3 week's n...
Name: text, dtype: object
```

```
In [39]: Y_train=data.iloc[0:200,0]
Y_train[0:5]

Out[39]: 0    ham
1    ham
2    spam
3    ham
4    ham
Name: label, dtype: object
```

Jupyter SMS Spam Detection Last Checkpoint: Last Thursday at 5:57 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [40]: from sklearn.feature_extraction.text import CountVectorizer
count_vector = CountVectorizer()
print(count_vector)

CountVectorizer(analyzer='word', binary=False, decode_error='strict',
               dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
               lowercase=True, max_df=1.0, max_features=None, min_df=1,
               ngram_range=(1, 1), preprocessor=None, stop_words=None,
               strip_accents=None, token_pattern='(?u)\\b\\w+\\b',
               tokenizer=None, vocabulary=None)
```

```
In [41]: train_data = count_vector.fit_transform(X_train)
test_data = count_vector.transform(x_test1)
```

```
In [42]: train_data.shape

Out[42]: (200, 1159)
```

```
In [43]: test_data.shape

Out[43]: (7, 1159)
```


Jupyter SMS Spam Detection Last Checkpoint: Last Thursday at 5:57 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [45]: `from sklearn.naive_bayes import MultinomialNB`

In [46]: `model=MultinomialNB()
model.fit(train_data,Y_train)`

Out[46]: `MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)`

In [47]: `pred=model.predict(test_data)
pred`

Out[47]: `array(['ham', 'spam', 'ham', 'ham', 'ham', 'ham', 'spam'],
 dtype='<U4')`

In [48]: `y1=model.predict(test_data)`

In [49]: `y1`

Out[49]: `array(['ham', 'spam', 'ham', 'ham', 'ham', 'ham', 'spam'],
 dtype='<U4')`

Jupyter SMS Spam Detection (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [50]: `df = pd.DataFrame(dict(INPUT=x_test1, OUTPUT=y1))
df`

Out[50]:

| | INPUT | OUTPUT |
|---|---|--------|
| 0 | hi how are you | ham |
| 1 | Free entry in 2 a wkly comp to win FA Cup fina... | spam |
| 2 | when will you go to home | ham |
| 3 | i will call you back | ham |
| 4 | are you busy now | ham |
| 5 | goodmoring | ham |
| 6 | WINNER!! As a valued network customer you have... | spam |

In [51]: `df.iloc[1:2]`

Out[51]:

| | INPUT | OUTPUT |
|---|---|--------|
| 1 | Free entry in 2 a wkly comp to win FA Cup fina... | spam |

In [52]: `df.iloc[0:3]`

Out[52]:

| | INPUT | OUTPUT |
|---|---|--------|
| 0 | hi how are you | ham |
| 1 | Free entry in 2 a wkly comp to win FA Cup fina... | spam |
| 2 | when will you go to home | ham |

9.CONCLUSION AND FUTURE WORK

9.1. CONCLUSION

Based on the analysis of the tests performed in this research, it can be concluded that:

Both methods used in this research, the performances of both methods are equally well for SMS classification with average of the accuracy above 90%. The use of collaboration methods, Naive Bayes and FP-Growth, is superior to the average accuracy for each dataset.

The Accuracy best average is obtained when the SMS Spam Collection v.1 dataset with the 9% minimum support is used and the implementation of the FP-Growth has accuracy up to 98.506%.

The use of datasets with varied training data is agreeable to be applied by using the FP-Growth. By implementing the FP-Growth for feature extraction, it can elevate the score of precision. Thus, the system becomes more precise in providing the information requested by the users in response to the SMS classification

9.2. FUTURE WORK REFERENCES

- i. Herbert Schildt.2008,” Java Complete Reference”, Tata McGraw-Hill, 7th Edition, pp. 177-180.
- ii. Grady Brooch, James Rambaugh.1998, “Unified Modeling Language User Guide” Addison Wesley Publishing, chapter 8-31.
- iii. www.android.com
- iv. <http://developer.android.com/index.html>
- v. www.google.com
- vi. <http://en.wikipedia.org/wiki/SQLite>