



# BUILDING A SMARTER AI POWER SPAM CLASSIFIER

912421104036:P.Sajitha parveen

Phase 5 Project documentation  
and submission

## Content for project phase5

1.Problem statement

2. Design thinking process

3. phases of development part(1)

i) dataset load

ii) Data preprocessing

lii)Feature extraction

4. phases of development part(2)

i) Model selection

ii) Model Training

iii)Model Evaluation

Iv)Model prediction

5. Innovative technique used

i) Bert language for feature extraction

ii) Heatmap

iii)Word cloud to see spam and ham messages

iv)Cofusion matrix





## Data source:

Data link: (<https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>)

v1	v2
ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
ham	U dun say so early hor... U c already then say...
ham	Nah I don't think he goes to usf, he lives around here though
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, 螢 1.50 to rcv
ham	Even my brother is not like to speak with me. They treat me like aids patient.
ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertun
spam	WINNER!! As a valued network customer you have been selected to receive a 螢 900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.
spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 875
spam	75. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
spam	URGENT! You have won a 1 week FREE membership in our 螢 100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LR
ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.
ham	I HAVE A DATE ON SUNDAY WITH WILL!!
spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap.xxxmobilemovieclub.com?n=QJKGIGHJJGCB
ham	Oh k...i'm watching here:)
ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.
ham	Fine if that 袷 s the way u feel. That 袷 s the way its gotta b
spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/7.1.20
ham	POBOXox36504W45WQ 16+
ham	Is that seriously how you spell his name?
ham	I 課 going to try for 2 months ha ha only joking
ham	So 7_ pay first lar... Then when is da stock comin...
ham	Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?





# 1.PROBLEM STATEMENT

A "Smarter AI Power Spam Classifier" refers to a sophisticated artificial intelligence system designed to effectively and accurately identify and filter out spam or

# 2.DESIGN THINKING

Define the problem

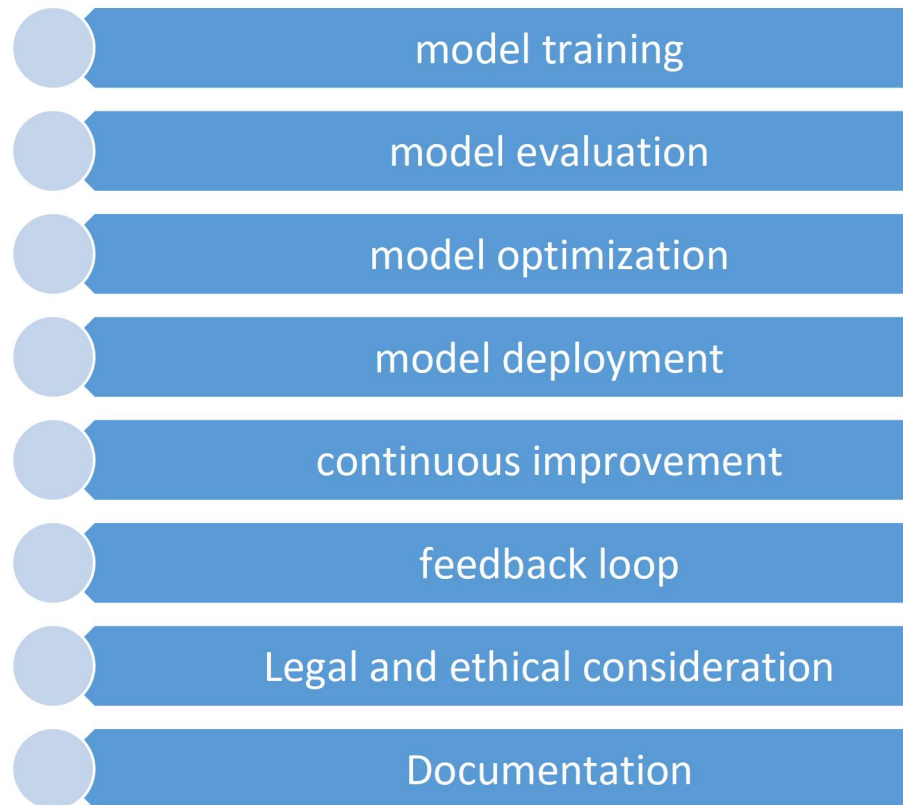
Data collection

Data processing

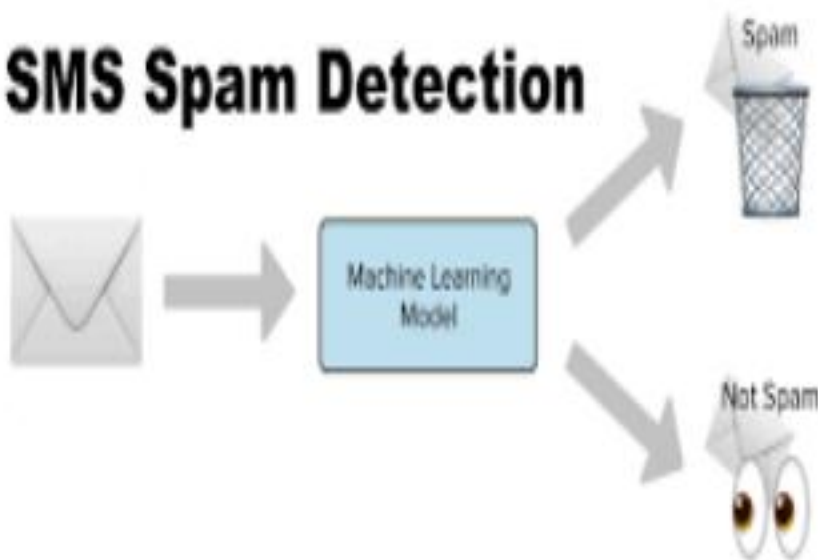
Data splitting

Model selection





## SMS Spam Detection



### 3. Data processing steps:

#### **Text Cleaning:**

Clean and preprocess the text data. This includes:

Removing special characters, punctuation, and numbers.

Lowercasing all text.

Tokenization: Splitting text into words or tokens.

Removing stop words (common words like "the," "and," "is" that don't carry much meaning).

#### **Feature Extraction:**

Convert text data into numerical features that machine learning algorithms can understand.

Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency

) and word embeddings like

Word2Vec or GloVe.

#### **Data Split:**

Split your dataset into two parts:

a training set and a testing set.

The training set will be used to train your model, while the testing set will be used to evaluate its performance.

#### **Select a Machine Learning Algorithm:**

Choose a suitable machine learning algorithm for text classification.

Common choices include:

Naive Bayes

Support Vector Machines (SVM)

Logistic Regression

Decision Trees

Neural Networks (e.g., LSTM, CNN)

#### **Training the Model**

Train your selected model using the

training data. During training,  
the model learns to recognize patterns  
and features in the  
data that distinguish  
spam from non-spam messages.

### **Evaluate Model Performance:**

Use the testing dataset to evaluate the model's performance.  
Common evaluation metrics for spam classification include accuracy,  
precision, recall, F1-score, and ROC AUC.

### **Model optimization:**

Hyperparameter Tuning:

Experiment with different hyperparameters of your chosen algorithm  
to optimize performance. This may involve grid search or random search.

Feature Engineering:

Experiment with different text features and representations to improve  
model accuracy.

### **Model Deployment:**

Once you are satisfied with  
your model's performance, deploy it in a  
production environment.

This may involve integrating it into an  
application or system  
that can classify messages in real-time

**Continuous improvement** in the context of spam  
classification refers to the ongoing process of enhancing  
the performance and effectiveness of a spam  
classifier over time

**Collecting feedback** from users or system  
administrators who interact with the spam classifier.

This feedback can help identify false positives and  
false negatives, allowing for model adjustments and

training data improvements.

**Legal and ethical considerations:** in AI-powered spam detection refer to the principles, rules, and guidelines that govern the development, deployment, and use of spam classification systems to ensure they align with legal frameworks and ethical standards. These considerations are crucial to safeguard individuals' rights, protect privacy, and prevent potential misuse of AI technology. Here are definitions for legal and ethical considerations in this context:

### **Program:**

#### **1. Import the required packages**

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

#### **2. Loading the dataset**

```
raw_spam=pd.read_csv('/content/spam.csv',encoding='latin-1')
print(raw_spam)
```

#### **Output:**

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

ham U dun say so early hor... U c already then say...

NaN

4

ham Nah I don't think he goes to usf, he lives aro...

NaN

... ..

...

...

5567 spam This is the 2nd time we have tried 2 contact u...

NaN

5568

ham

Will i\_b going to esplanade fr home?

NaN

5569

ham Pity, \* was in mood for that. So...any other s...

NaN

5570

ham The guy did some bitching but I acted like i'd...

NaN

5571 ham

Rofl. Its true to its name

NaN

Unnamed: 3 Unnamed: 4

0

NaN

NaN

1

NaN

NaN

2

NaN

NaN

3



NaN

NaN

4

NaN

NaN

...

...

...

5567

NaN

NaN

5568

NaN

NaN

5569

NaN

NaN

5570

NaN

NaN

5571

NaN

NaN

[5572 rows x 5 columns]

error

0scompleted at 12:43 PM**3.Removing the unwanted colomns**

```
raw_spam.rename(columns = {'v1':'class_label', 'v2':'message'}, inplace =  
True)
```

```
raw_spam.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis = 1, inplace  
= True)
```

```
raw_spam[1990:2000]
```

**Output:**

**class\_label message**

**1990**

ham HI DARLIN IVE JUST GOT BACK AND I HAD A REALLY...

**1991**

ham

No other Valentines huh? The proof is on your ...

**1992**

spam

Free tones Hope you enjoyed your new content. ...

**1993**

ham

Eh den sat u book e kb liao huh...

**1994**

ham

Have you been practising your curtsey?

**1995**

ham

Shall i come to get pickle

**1996**

ham

Lol boo I was hoping for a laugh

**1997**

ham

\YEH I AM DEF UP4 SOMETHING SAT

**1998**

ham

Well, I have to leave for my class babe ... Yo...

**1999**

ham

LMAO where's your fish memory when I need it?

**4.Exploring the dataset:**

```
raw_spam['class_label'].value_counts()
```

**Output:**

ham 4825

spam 747

Name: class\_label, dtype: int64

## 5. Print spam messages

```
raw_spam = raw_spam[raw_spam.class_label=='spam']
```

```
raw_spam
```

**Output:**

```
class_label message
```

```
2
```

```
spam
```

```
Free entry in 2 a wkly comp to win FA Cup fina...
```

```
5
```

```
spam
```

```
FreeMsg Hey there darling it's been 3 week's n...
```

```
8
```

```
spam WINNER!! As a valued network customer you have...
```

```
9
```

```
spam
```

```
Had your mobile 11 months or more? U R entitle...
```

```
11
```

```
spam SIX chances to win CASH! From 100 to 20,000 po... ..5537 spam
```

```
Want explicit SEX in 30 secs? Ring 02073162414...
```

```
5540
```

```
spam
```

```
ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...
```

```
5547 spam
```

```
Had your contract mobile 11 Mnths? Latest Moto...
```

```
5566 spam
```

```
REMINDER FROM O2: To get 2.50 pounds free call...
```

```
5567 spam
```

```
This is the 2nd time we have tried 2 contact u...
```

```
[747 rows x 2 columns]
```

## 6.prepare spam list

```
spam_list= raw_spam['message'].tolist()
```

```
print(spam_list)
```

**Output:**

["Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's", "FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still?"]

### 7.create array:

```
import matplotlib.pyplot as ab
import numpy as np
labels = ['ham', 'spam']
counts = [4825, 747]
ypos = np.arange(len(labels)) #converting text labels to numeric value, 0 and 1
```

Ypos

#### Output:

```
array([0, 1])
```

### 8.using graph:

```
ab.xticks(ypos, labels)
ab.xlabel("class label")
ab.ylabel("Frequency")
ab.title("# of spam and ham in dataset")
ab.bar(ypos, counts)
```

#### Output:

```
<BarContainer object of 2 artists>
```

### 9.replace the null values with a null string

```
mail_data=raw_spam.where((pd.notnull(raw_spam)),")
```

#### #printing the first five rows of the dataframe

```
mail_data.head()
```

#### Output:

```
class_labelmessage
```

```
2 spam Free entry in 2 a wkly comp to win FA Cup fina...
```

```
5 spam FreeMsg Hey there darling it's been 3 week's n...
```

```
8 spam WINNER!! As a valued network customer you have...
```

```
9 spam Had your mobile 11 months or more? U R entitle...
```

```
11 spam SIX chances to win CASH! From 100 to 20,000 po...
```

## 10.checking the number of rows and colomns in the dataframe

```
mail_data.shape
```

**Output:**

```
(747,2)
```

## 11.label spam mail as 0; ham mail as 1

```
mail_data.loc[mail_data['class_label']
```

```
== 'spam','class_label',] = 0
```

```
mail_data.loc[mail_data['message']=='ham','message',] = 1
```

**#separating the data as texts and label**

```
x=mail_data['message']
```

```
y=mail_data['class_label']
```

**Output:**

**print(x)**

```
0
```

```
Go until jurong point, crazy.. Available only ...
```

```
1
```

```
Ok lar... Joking wif u oni...
```

```
2
```

```
Free entry in 2 a wkly comp to win FA Cup fina...
```

```
3
```

```
U dun say so early hor... U c already then say...
```

```
4
```

```
Nah I don't think he goes to usf, he lives aro... ...
```

```
5567
```

```
This is the 2nd time we have tried 2 contact u...
```

```
5568
```

```
Will Ì_ b going to esplanade fr home?
```

```
5569
```

```
Pity, * was in mood for that. So...any other s...
```

```
5570
```

```
The guy did some bitching but I acted like i'd...
```

```
5571
```

```
Rofl. Its true to its name Name: v2, Length:
```

```
5572, dtype: object
```

```
print(y)
```

```
2
```

```
0
```

```
5
```

```
0
```

```
8
```

```
0
```

```
9
```

```
0
```

```
11
```

```
0 ..
```

```
5537
```

```
0
```

```
5540
```

```
0
```

```
5547
```

```
0
```

```
5566
```

```
0
```

```
5567
```

```
0
```

```
Name: class_label, Length: 747, dtype: object
```

## 12.Splitting the data into training data and testing data

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=3)
```

```
print(x.shape)
```

```
print(x_train.shape)
```

```
print(x_test.shape)
```

**Output:**

```
(747,)
```

```
(597,)
```

```
(150,)
```

## 13.Removing punctuation and stopwords from the messages

Punctuation and stop words do not contribute anything to our model, so we have

to remove them. Using NLTK library we can easily do it.

```

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
#remove the punctuations and stopwords
import string
def message_process(message):
    message = message.translate(str.maketrans("", "", string.punctuation))
    message = [word for word in message.split() if word.lower()
notinstopwords.words('english')]
    return " ".join(message)
raw_spam['message'] = raw_spam['message'].apply(message_process)
raw_spam.head()

```

### **Output:**

```

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

```

### **class\_label**

### **message**

**2**

spam

Free entry 2 wkly comp win FA Cup final tkts 2...

**5**

spam

FreeMsg Hey darling 3 weeks word back Id like ...

**8**

spam

WINNER valued network customer selected receiv...

**9**

spam

mobile 11 months U R entitled Update latest co...**class\_label**

### **message**

**11**

spam SIX chances win CASH 100 20000 pounds txt CSH1...

## **14.Converting words to vectors using Count Vectorizer**

**##** Counting how many times a word appears in the dataset

we can convert words to vectors using either Count Vectorizer or by using TF-IDF

Vectorizer.

TF-IDF is better than Count Vectorizers because it not only focuses on the frequency of

words present in the corpus but also provides the importance of the words.

We can then

remove the words that are less important for analysis, hence making the model building

less complex by reducing the input dimensions.

I have included both methods for your reference.

```
text = pd.DataFrame(raw_spam['message'])
```

```
label = pd.DataFrame(raw_spam['class_label'])
```

```
from collections import Counter
```

```
total_counts = Counter()
```

```
for i in range(len(text)):
```

```
for word in text.values[i][0].split(" "):
```

```
total_counts[word] += 1
```

```
print("Total words in data set: ", len(total_counts))
```

**Output:**

Total words in data set: 4313

**15.sorting in decreasing order (word with highest frequency appears first)**

```
vocab = sorted(total_counts, key=total_counts.get, reverse=True)
```

```
print(vocab[:60])
```

**Output:**

['to', 'a', 'your', 'call', 'or', 'the', '2', 'for', 'you', 'is', 'Call', 'on', 'have', 'and', 'from', 'ur',

'with', '&', '4', 'of', 'FREE', 'mobile', 'You', 'are', 'our', 'To', 'claim', 'Your', 'U', 'txt', 'text', 'in', 'now', 'Txt', 'reply', 'free', 'contact', '-', 'be', 'now!', 'u', 'just', 'send', 'this', 'won', 'get',

'only', 'Nokia', 'prize', 'per', 'been', 'service', 'STOP', 'who', 'Reply', 'new', 'cash', 'out',

'Text', 'will']



## 16.Mapping from words to index

```
vocab_size = len(vocab)
```

```
word2idx = {}
```

```
#print vocab_size
```

```
for i, word in enumerate(vocab):
```

```
word2idx[word] = i
```

```
# Text to Vector
```

```
def text_to_vector(text):
```

```
word_vector = np.zeros(vocab_size)
```

```
for word in text.split(" "):
```

```
if word2idx.get(word) is None:
```

```
continue
```

```
else:
```

```
word_vector[word2idx.get(word)] += 1
```

```
return np.array(word_vector)
```

```
# Convert all titles to vectors
```

```
word_vectors = np.zeros((len(text), len(vocab)), dtype=np.int_)
```

```
for i, (_, text_) in enumerate(text.iterrows()):
```

```
word_vectors[i] = text_to_vector(text_[0])
```

```
word_vectors.shape
```

```
Output:
```

```
(747,3436)
```

## 4.Feature Extraction:

### 1.Feature Extraction:

```
#transform the text data feature vectors that can be used as input to the  
logistic
```

```
regressionfeature_extraction=TfidfVectorizer(min_df=
```

```
1,stop_words='english',lowercase=True)
```

```
x_train_features = feature_extraction.fit_transform(x_train)
```

```
x_test_features =feature_extraction.transform(x_test)
```

```
#convert y_train and y_test values as integer
```

```
y_train=y_train.astype('int')
```

```
y_test=y_test.astype('int')
```

print(x\_train)

**Output:**

1713

Hard LIVE 121 chat just 60p/min. Choose your g...

2547

Text82228>> Get more ringtones, logos and game...

1121

Do you want 750 anytime any network mins 150 t...

4752

Cashbin.co.uk (Get lots of cash this weekend!)

1740

UR GOING 2 BAHAMAS! CallFREEFONE 08081560665

a... ..

4901

\* FREE\* POLYPHONIC RINGTONE Text SUPER to 8713...

1829

Hottest pics straight to your phone!! See me g...

4784

Urgent -call 09066649731from Landline. Your co...

1766

SMS AUCTION You have won a Nokia 7250i. This i...

4929

Hi, the SEXYCHAT girls are waiting for you to ... Name

message, Length: 597, dtype: object

print(x\_train\_features)

**Output:**(0, 747)

0.23968206096754352

(0, 2295)

0.23968206096754352

(0, 1943)

0.13573587486497507

(0, 762)

0.22872853151403771

(0, 2229)

0.12824610679055637

(0, 886)

0.22023231400263973

(0, 251)

0.23968206096754352

(0, 950)

0.23968206096754352

(0, 1209)

0.22872853151403771

(0, 895)

0.20233686938100554

(0, 1521)

0.15602247134242647

(0, 498)

0.22023231400263973

(0, 1362)

0.12561919492401685

(0, 881)

0.3092755378471771

(0, 280)

0.22872853151403771

(0, 1425)

0.5023469692097261

(0, 1247)

0.22872853151403771

(1, 1328)

0.23439849165672041

(1, 1802)

0.27252525133706634

(1, 929)

0.17726401485344148

(1, 2217)

0.67039175067174

(1, 2373)

0.14881462222970782

(1, 1191)

0.2455614538082651

(1, 1439)

0.3149121089938614

(1, 1888)

0.2599530150590547

:

:

(595, 1267)

0.24882581220174205

(595, 591)

0.2662585039974314

(595, 711)

0.4976516244034841

(595, 524)

0.2827037769231108

(595, 1898)

0.2323805392760626

(595, 368)

0.23600260878252416

(595, 2082)

0.23600260878252416(595, 1615)

0.3506263391097266

(595, 309)

0.18106988275900449

(595, 1938)

0.16640209923351984

(595, 1993)

0.2060367771590298

(595, 2347)

0.17115103298760287

(595, 2362)

0.1565942042695853

(595, 1170)  
0.12089909450957946  
(596, 1953)  
0.3767407467588377  
(596, 1210)  
0.3252211643918774  
(596, 2307)  
0.25641287357880704  
(596, 885)  
0.35394296838190215  
(596, 1610)  
0.3149699088288029  
(596, 1235)  
0.29217213045186735  
(596, 1938)  
0.19142823296873324  
(596, 1268)  
0.25982156809953977  
(596, 2046)  
0.344425910494105  
(596, 2129)  
0.33726090918359547  
(596, 1943)  
0.20044369725329336

check

0scompleted at 8:01 PM

## 5. Model selection:

### 2. Logistic Regression

```
model = LogisticRegression()
```

**#training the logistic regression model with the training data**

```
model.fit(x_train_features,x_train)
```

**Output:**

```
LogisticRegressionLogisticRegression()
```

## 6.model training and evaluation:

### 3. Evaluating the model:

#### #prediction on training data

```
prediction_on_training_data=model.predict(x_train_features)
accuracy_on_training_data=accuracy_score(y_train,prediction_on_training_data)
print('Accuracy on training data :',accuracy_on_training_data)
```

#### Output:

Accuracy on training data : 0.9661207089970832

#### #prediction on test data

```
prediction_on_test_data=model.predict(x_test_features)
accuracy_on_test_data=accuracy_score(y_test,prediction_on_test_data)
print('accuracy on test data:',accuracy_on_test_data)
```

#### Output:

accuracy on test data: 0.9623318385650225

## 7.buiding a predictive system

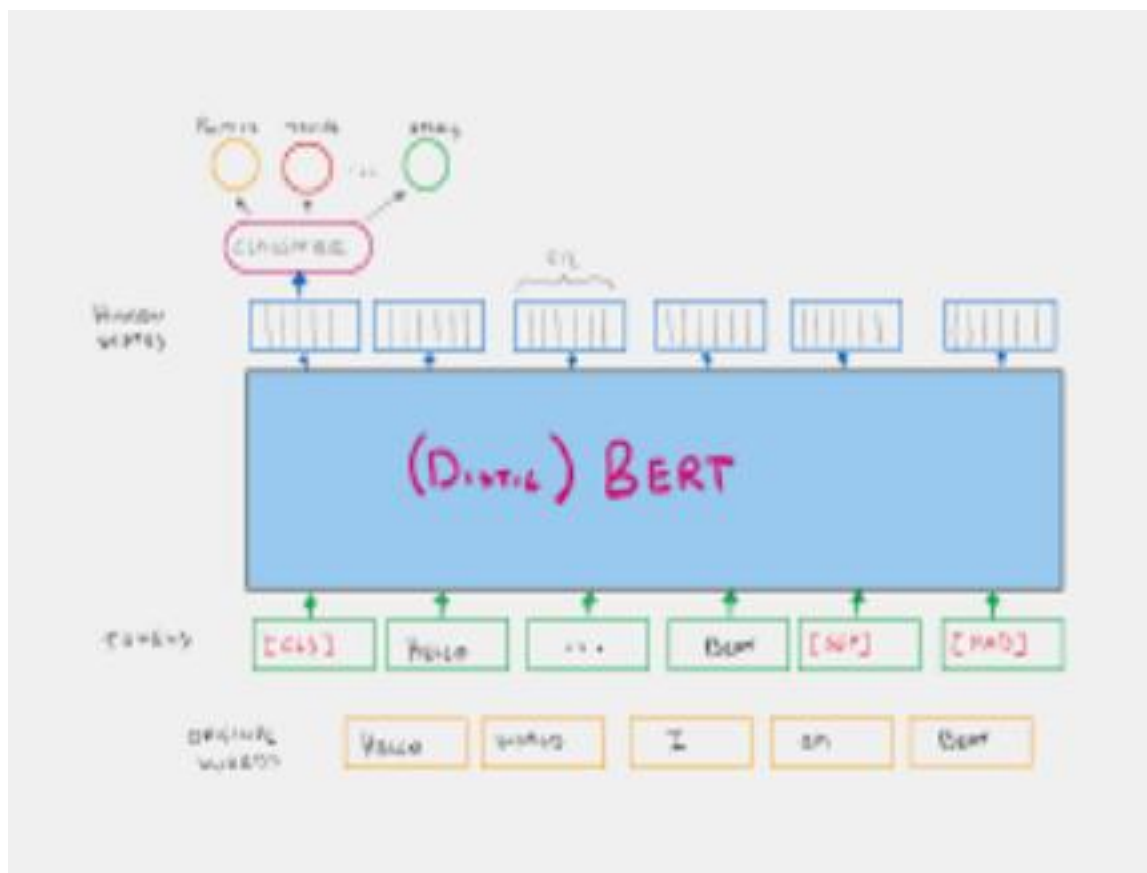
```
input_mail=["I HAVE A DATE ON SUNDAY WITH WILL!!,,,"]#convert text to
feature vectors
input_data_features=feature_extraction.transform(input_mail)
#making prediction test
prediction=model.predict(input_data_features)
print(prediction)
if prediction[0] == 1:
print('Ham mail')
else:
print('Spam mail')
```

#### Output:

Ham  
Spam mail

## 8. Innovative techniques:

## BertLanguage for feature extraction

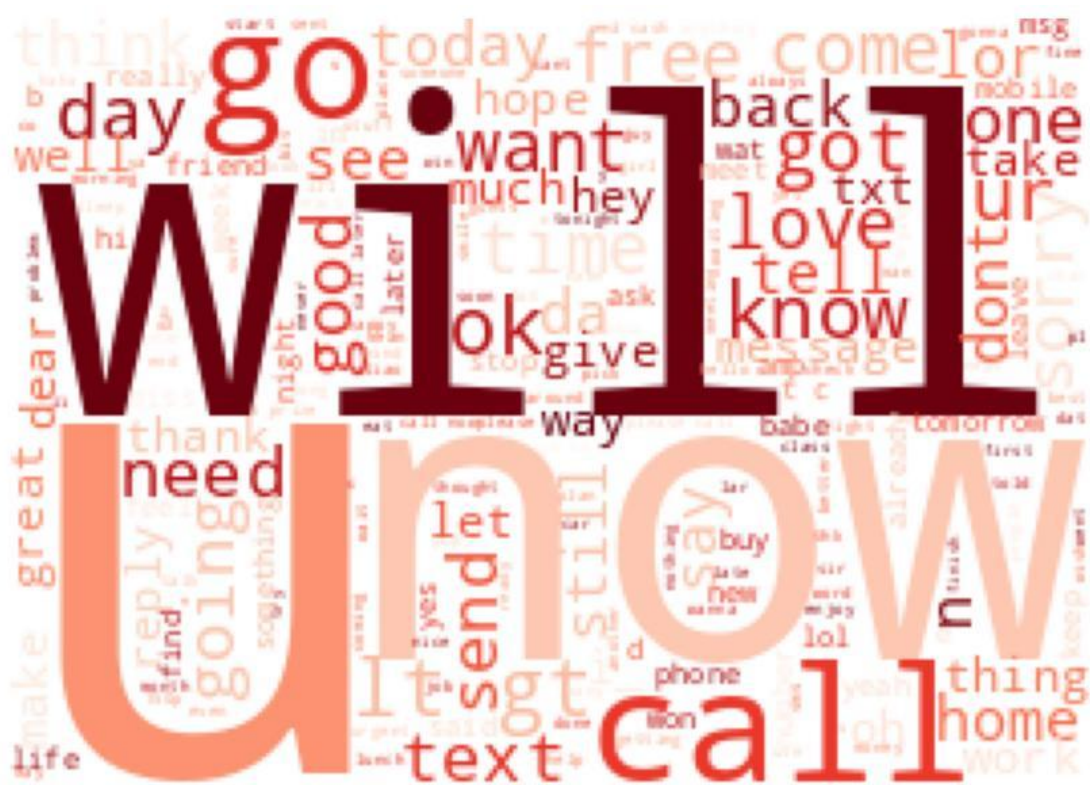


### Creating spam word cloud

```
import os
import numpy as np
from wordcloud import WordCloud
from PIL import Image
# Assuming you have loaded your DataFrame 'df_spam' and extracted the
'message'
column into 'spam_list'
spam_list = raw_spam['v2'].tolist()
# Combine the text from 'spam_list' into a single string
filtered_spam = ' '.join(spam_list).lower()
# Load the comment mask image
comment_mask = np.array(Image.open("/content/comment.png"))
# Create and generate a word cloud image
wordcloud = WordCloud(
    max_font_size=160,
    margin=0,
```

```
mask=comment_mask,
background_color="white",
colormap="Reds"
).generate(filtered_spam)
# Display the generated word cloud
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
```

**Output:**



### Creating ham word cloud:

```
import os
import numpy as np
from wordcloud import WordCloud
from PIL import Image

# Assuming you have loaded your DataFrame 'df_ham' and extracted the
'message' column into 'ham_list'

raw_ham=pd.read_csv('/content/spam.csv',encoding='latin-1')
```



```

print(raw_ham)
ham_list = raw_ham['v2'].tolist()
# Combine the text from 'ham_list' into a single string
filtered_ham = ' '.join(ham_list).lower()
# Load the comment mask image
comment_mask = np.array(Image.open("/content/comment.png"))# Create
and generate a word cloud image for ham messages
wordcloud = WordCloud(
max_font_size=160,
margin=0,
mask=comment_mask,
background_color="white",
colormap="Greens" # You can choose a different colormap if desired
).generate(filtered_ham)
# Display the generated word cloud
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
# Save the word cloud to a file (optional)
wordcloud.to_file("ham_wordcloud.png")
plt.show()

```

### Output:

```

v1 v2 Unnamed: 2 \
0 ham Go until jurong point, crazy.. Available only ... NaN
1 ham Ok lar... Joking wif u oni... NaN
2 spam Free entry in 2 a wkly comp to win FA Cup fina... NaN
3 ham U dun say so early hor... U c already then say... NaN
4 ham Nah I don't think he goes to usf, he lives aro... NaN ... ..
5567 spam This is the 2nd time we have tried 2 contact u... NaN
5568 ham Will I_b going to esplanade fr home? NaN
5569 ham Pity, * was in mood for that. So...any other s... NaN
5570 ham The guy did some bitching but I acted like i'd... NaN

```

5571 ham Rofl. Its true to its name NaN

Unnamed: 3

Unnamed: 40

NaN

NaN

1

NaN

NaN

2

NaN

NaN

3

NaN

NaN

4

NaN

NaN ... ..

5567

NaN

NaN

5568

NaN

NaN

5569

NaN

NaN

5570

NaN

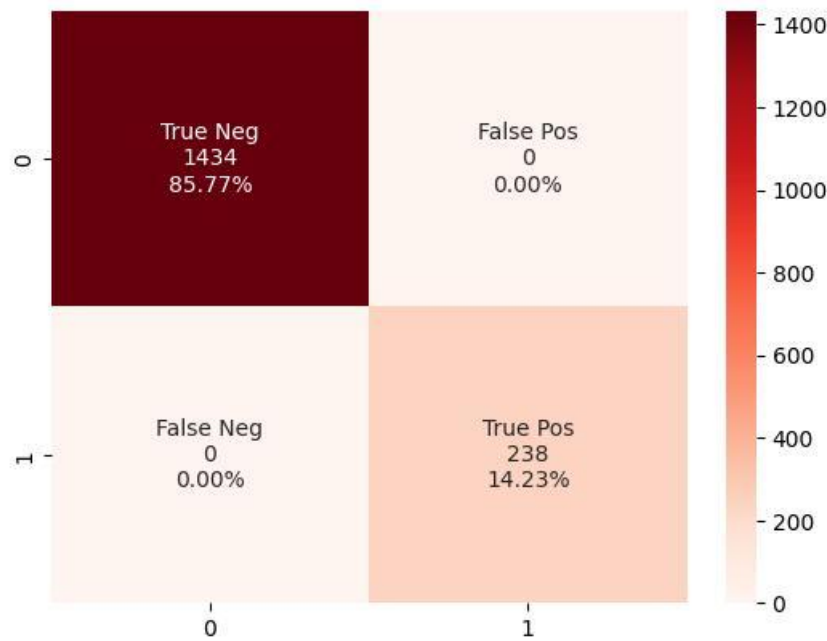
NaN

5571

NaN

NaN [5572 rows x 5 columns]





## Confusion matrix:

```
labels = classifier.predict(features_test_transformed)
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
actual = y_test.tolist()
predicted = labels
results = confusion_matrix(actual, predicted)
print('Confusion Matrix :')
print(results)
print ('Accuracy Score :',accuracy_score(actual, predicted))
print ('Report : ')print (classification_report(actual, predicted) )
score_2 = f1_score(actual, predicted, average = 'binary')
print('F-Measure: %.3f % score_2)
```

### Output:

Confusion Matrix :

```
[[1434 0]
```

```
[ 0 238]]
```

Accuracy Score : 1.0

Report :

precision

recall

f1-score

support

0

1.00

1.00

1.00

1434

1

1.00

1.00

1.00

238

accuracy

1.00

1672

macro avg 1.00

1.00

1.00

1672

weighted avg 1.00

1.00

1.00

1672

F-Measure: 1.000

### **TF-IDF matrix:**

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
# Sample text data
```

```
x_train = [
```

```
"This is the first document.",
```

```
"This document is the second document.",
```

```
"And this is the third one.",
"Is this the first document?",
]
# Initialize the TfidfVectorizer with optional parameters
vectorizer = TfidfVectorizer(
stop_words='english', # Remove stop words
max_features=1000, # Limit the number of features
lowercase=True, # Convert text to lowercase
)
# Fit and transform the training data
features_train_transformed = vectorizer.fit_transform(x_train)
# Print the feature names (words or terms)
print("Feature names (words or terms):")
print(vectorizer.get_feature_names_out())
# Print the TF-IDF matrix
print("TF-IDF matrix:")
print(features_train_transformed.toarray())
# You can also transform test data using the same vectorizer
x_test = ["This is a new document.", "Another document for testing."]
features_test_transformed = vectorizer.transform(x_test)
print("TF-IDF matrix for test data:")
print(features_test_transformed.toarray())
```

**Output:**

Feature names (words or terms):

['document' 'second']

TF-IDF matrix: [[1.

0.

]

[0.78722298

0.61666846]

[0.

0.

]

[1.

0.

]]

TF-IDF matrix for test data:

[[1. 0.]

[1. 0.]]

The image displays two screenshots of a Google Colab notebook titled "AI\_project spam or ham message detection".

**Top Screenshot:**

- Code Cell [1]:** Imports necessary libraries: `numpy`, `pandas`, `sklearn.model_selection` (with `train_test_split`), `sklearn.feature_extraction.text` (with `TfidfVectorizer`), `sklearn.linear_model` (with `LogisticRegression`), and `sklearn.metrics` (with `accuracy_score`).
- Code Cell [2]:** Loads data from a CSV file into a pandas dataframe: `raw_spam=pd.read_csv('/content/spam.csv',encoding='latin-1')` and prints it: `print(raw_spam)`.
- Output:** A preview of the dataframe showing columns `v1` (message text) and `v2` (spam/ham label). The first three rows are:

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

**Bottom Screenshot:**

- Code Cell [2]:** Same code as above, loading and printing the CSV file.
- Output:** A larger preview of the dataframe, showing rows from index 0 to 5571. The `v2` column contains `NaN` for all visible rows, which is likely a display artifact in the notebook interface.

Sign in or Register | SkillU | Project Submission Form | Inbox (473) - peer87374@ | AI\_project spam or ham message detection

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCt25GEMu#scrollTo=g8L7ciBgcNv7

AI\_project spam or ham message detection

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[3] raw_spam.rename(columns = {'v1': 'class_label', 'v2': 'message'}, inplace = True)
raw_spam.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis = 1, inplace = True)
raw_spam[1990:2000]
```

	class_label	message
1990	ham	HI DARLIN IVE JUST GOT BACK AND I HAD A REALLY...
1991	ham	No other Valentines huh? The proof is on your ...
1992	spam	Free tones Hope you enjoyed your new content ...
1993	ham	Eh den sat u book e kb liau huh...
1994	ham	Have you been practising your curtsy?
1995	ham	Shall i come to get pickle
1996	ham	Lol boo I was hoping for a laugh
1997	ham	IYEH I AM DEF UP4 SOMETHING SAT
1998	ham	Well, I have to leave for my class babe ... Yo...

0s completed at 10:24AM

Sign in or Register | SkillU | Project Submission Form | Inbox (473) - peer87374@ | AI\_project spam or ham message detection

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCt25GEMu#scrollTo=5HX1ecSWICre

AI\_project spam or ham message detection

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
1999 ham LMAO where's your ish memory when i need it?
```

```
#exploring the dataset
raw_spam['class_label'].value_counts()
```

```
ham      4825
spam      747
Name: class_label, dtype: int64
```

```
raw_spam = raw_spam[raw_spam.class_label=='spam']
raw_spam
```

	class_label	message
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

0s completed at 10:26AM



Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham m x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCI25GEMu#scrollTo=T5ChWvRCnv76

AI\_project spam or ham message detection

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
print(raw_spam)
```

	class_label	message
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...
11	spam	SIX chances to win CASH! From 100 to 20,000 po...
...	...	...
5537	spam	Want explicit SEX in 30 secs? Ring 02073162414...
5540	spam	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...
5547	spam	Had your contract mobile 11 Mnths? Latest Moto...
5566	spam	REMINDER FROM O2: To get 2.50 pounds free call...
5567	spam	This is the 2nd time we have tried 2 contact u...

[747 rows x 2 columns]

```
spam_list= raw_spam['message'].tolist()
```

Connecting to Python 3 Google Compute Engine backend

Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham m x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCI25GEMu#scrollTo=oUptc1wOIKCh

AI\_project spam or ham message detection

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[7] spam_list= raw_spam['message'].tolist()
```

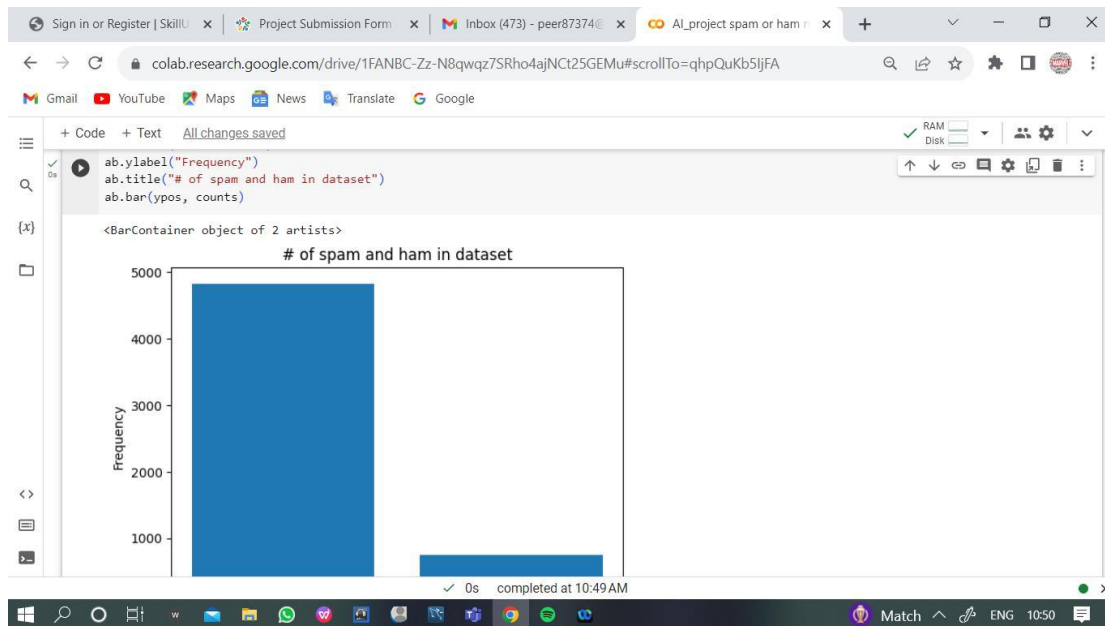
```
[8] print(spam_list)
```

```
["Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 0845"]
```

```
import matplotlib.pyplot as ab
import numpy as np
labels = ['ham', 'spam']
counts = [4825, 747]
ypos = np.arange(len(labels)) #converting text labels to numeric value, 0 and 1
ypos
```

```
array([0, 1])
```

0s completed at 10:46 AM



Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham p x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCI25GEMu#scrollTo=57QPNTw0ELp

Gmail YouTube Maps News Translate Google

+ Code + Text All changes saved

```
[11] #replace the null values with a null string
mail_data=raw_spam.where((pd.notnull(raw_spam)), '')

#printing the first five rows of the dataframe
mail_data.head()
```

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

```
#checking the number of rows and columns in the dataframe
mail_data.shape
```

(747, 2)

0s completed at 10:50 AM

Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham p x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCt25GEMu#scrollTo=9QOBi2VbuSum

Gmail YouTube Maps News Translate Google

+ Code + Text Saving...

RAM Disk

```
[15] mail_data.loc[mail_data['message']=='ham','message',] = 1
```

(x)

```
[16] #separating the data as texts and label
x=mail_data['message']
y=mail_data['class_label']
```

print(x)

```
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...
...
5537   Want explicit SEX in 30 secs? Ring 02073162414...
5540   ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...
5547   Had your contract mobile 11 Mnths? Latest Moto...
5566   REMINDER FROM O2: To get 2.50 pounds free call...
5567   This is the 2nd time we have tried 2 contact u...
Name: message, Length: 747, dtype: object
```

0s completed at 10:50 AM

Match ENG 10:51

Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham p x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCt25GEMu#scrollTo=iPPlgWt2TQ0W

Gmail YouTube Maps News Translate Google

+ Code + Text All changes saved

RAM Disk

splitting the data into training data and testing data

Double-click (or enter) to edit

```
[19] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=3)
```

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

```
(747,)
(597,)
(150,)
```

```
[ ] import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

#remove the punctuations and stopwords
import string
```

0s completed at 10:51 AM

Match ENG 10:51

Sign in or Register | SkillU x | Project Submission Form x | Inbox (473) - peer87374@ x | AI\_project spam or ham r x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCt25GEMu#scrollTo=sPYP6gPIIKS3

Gmail YouTube Maps News Translate Google

+ Code + Text All changes saved

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

#remove the punctuations and stopwords
import string
def message_process(message):

    message = message.translate(str.maketrans('', '', string.punctuation))
    message = [word for word in message.split() if word.lower() not in stopwords.words('english')]

    return " ".join(message)

raw_spam['message'] = raw_spam['message'].apply(message_process)
raw_spam.head()
```

[nltk\_data] Downloading package stopwords to /root/nltk\_data...  
[nltk\_data] Unzipping corpora/stopwords.zip.

	class_label	message
2	spam	Free entry 2 wkly comp win FA Cup final tkts 2...
5	spam	FreeMsg Hey darling 3 weeks word back Id like ...

4s completed at 10:52 AM

Sign in or Register | SkillU x | Project Submission Form x | Inbox (473) - peer87374@ x | AI\_project spam or ham r x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCt25GEMu#scrollTo=Jlm2dfGN6OBF

Gmail YouTube Maps News Translate Google

+ Code + Text

```
11 spam SIX chances win CASH 100 20000 pounds btt CSH1...
```

```
[ ]
```

```
text = pd.DataFrame(raw_spam['message'])
label = pd.DataFrame(raw_spam['class_label'])
from collections import Counter

total_counts = Counter()
for i in range(len(text)):
    for word in text.values[i][0].split(" "):
        total_counts[word] += 1

print("Total words in data set: ", len(total_counts))
```

Total words in data set: 3436

```
vocab = sorted(total_counts, key=total_counts.get, reverse=True)
print(vocab[:60])
```

['call', '2', 'Call', '4', 'FREE', 'un', 'mobile', 'U', 'claim', 'prize', 'text', 'txt', 'Txt', 'free', 'u', 'STOP', 'reply', 'contact', '16',

0s completed at 10:52 AM

Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham r x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCI25GEMu#scrollTo=Af-KSa6KGWyp

Gmail YouTube Maps News Translate Google

+ Code + Text All changes saved

```
vocab_size = len(vocab)
word2idx = {}
#print vocab_size
for i, word in enumerate(vocab):
    word2idx[word] = i
# Text to Vector
def text_to_vector(text):
    word_vector = np.zeros(vocab_size)
    for word in text.split(" "):
        if word2idx.get(word) is None:
            continue
        else:
            word_vector[word2idx.get(word)] += 1
    return np.array(word_vector)
# Convert all titles to vectors
word_vectors = np.zeros((len(text), len(vocab)), dtype=np.int_)
for i, (_, text_) in enumerate(text.iterrows()):
    word_vectors[i] = text_to_vector(text_[0])

word_vectors.shape
```

(747, 3436)

0s completed at 10:53 AM

Sign in or Register | SkillU x Project Submission Form x Inbox (473) - peer87374@ x AI\_project spam or ham r x +

colab.research.google.com/drive/1FANBC-Zz-N8qwqz7SRho4ajNCI25GEMu#scrollTo=4Tm1cKJTjeoY

Gmail YouTube Maps News Translate Google

Files + Code + Text All changes saved

```
import os
import numpy as np
from wordcloud import WordCloud
from PIL import Image

# Assuming you have loaded your DataFrame 'df_spam' and extracted the 'message' column into 'spam_list'
spam_list = raw_spam['message'].tolist()

# Combine the text from 'spam_list' into a single string
filtered_spam = ' '.join(spam_list).lower()

# Load the comment mask image
comment_mask = np.array(Image.open("/content/comment.png"))

# Create and generate a word cloud image
wordcloud = WordCloud(
    max_font_size=160,
    margin=0,
    mask=comment_mask,
    background_color="white",
    colormap="Reds"
).generate(filtered_spam)
```

0s completed at 10:54 AM

27°C Mostly cl... ENG 10:58





## Featureextraction :

The screenshot shows a Google Colab notebook with the following code in a cell:

```
[32]: feature_extraction=TfidfVectorizer(min_df= 1,stop_words='engl
      x_train_features = feature_extraction.fit_transform(x_train)
      x_test_features =feature_extraction.transform(x_test)

      y_train=y_train.astype('str')
      y_test=y_test.astype('str')

      print(x_train)
```

The output of the `print(x_train)` command is displayed in a scrollable window, showing a sparse matrix of TF-IDF values for the training data. The matrix has 4457 rows and 1 columns. The output is truncated, showing only the first few rows and the last few columns.

RAM: 0s, Disk: 0s, completed at 5:06 PM

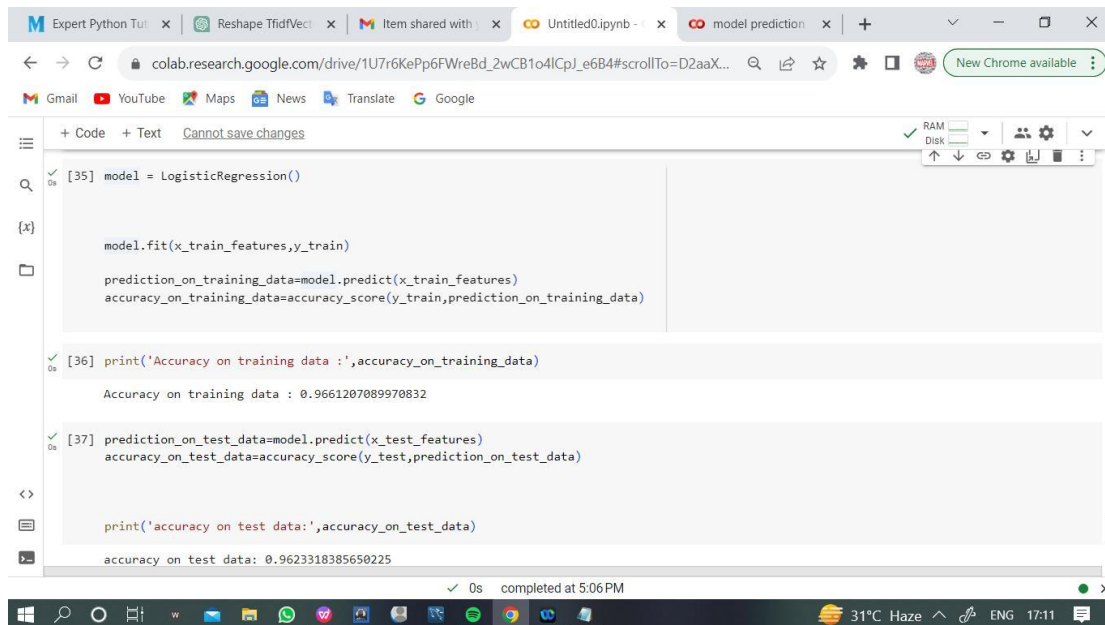
The screenshot shows a Google Colab notebook with the following code in a cell:

```
print(x_train_features)
```

The output of the `print(x_train_features)` command is displayed in a scrollable window, showing a sparse matrix of TF-IDF values for the training data. The matrix has 4457 rows and 1 columns. The output is truncated, showing only the first few rows and the last few columns.

RAM: 0s, Disk: 0s, completed at 5:06 PM

## Train themodel:



```
[35] model = LogisticRegression()

model.fit(x_train_features,y_train)

prediction_on_training_data=model.predict(x_train_features)
accuracy_on_training_data=accuracy_score(y_train,prediction_on_training_data)

[36] print('Accuracy on training data :',accuracy_on_training_data)

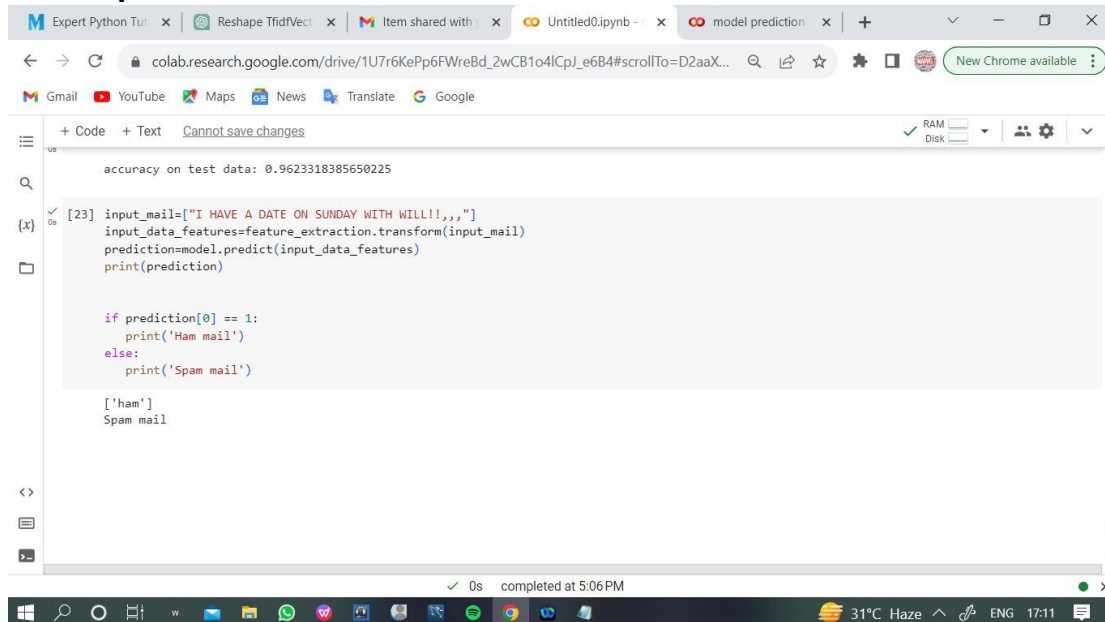
Accuracy on training data : 0.9661207089970832

[37] prediction_on_test_data=model.predict(x_test_features)
accuracy_on_test_data=accuracy_score(y_test,prediction_on_test_data)

print('accuracy on test data:',accuracy_on_test_data)

accuracy on test data: 0.9623318385650225
```

## Model prediction:



```
accuracy on test data: 0.9623318385650225

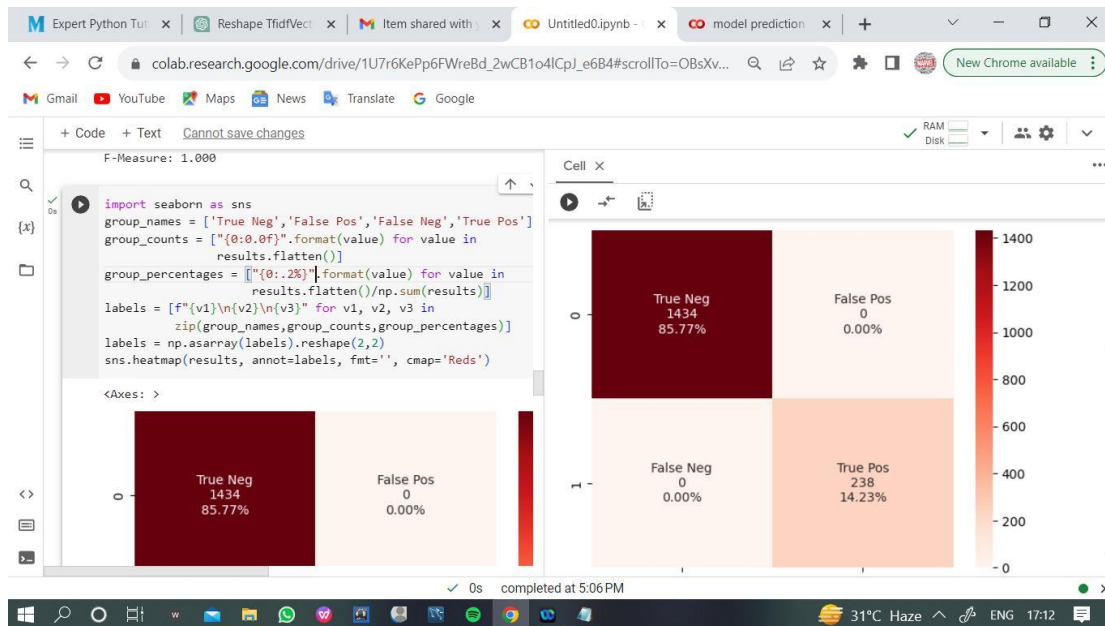
[23] input_mail=["I HAVE A DATE ON SUNDAY WITH WILL!!,,,"]
input_data_features=feature_extraction.transform(input_mail)
prediction=model.predict(input_data_features)
print(prediction)

if prediction[0] == 1:
    print('Ham mail')
else:
    print('Spam mail')

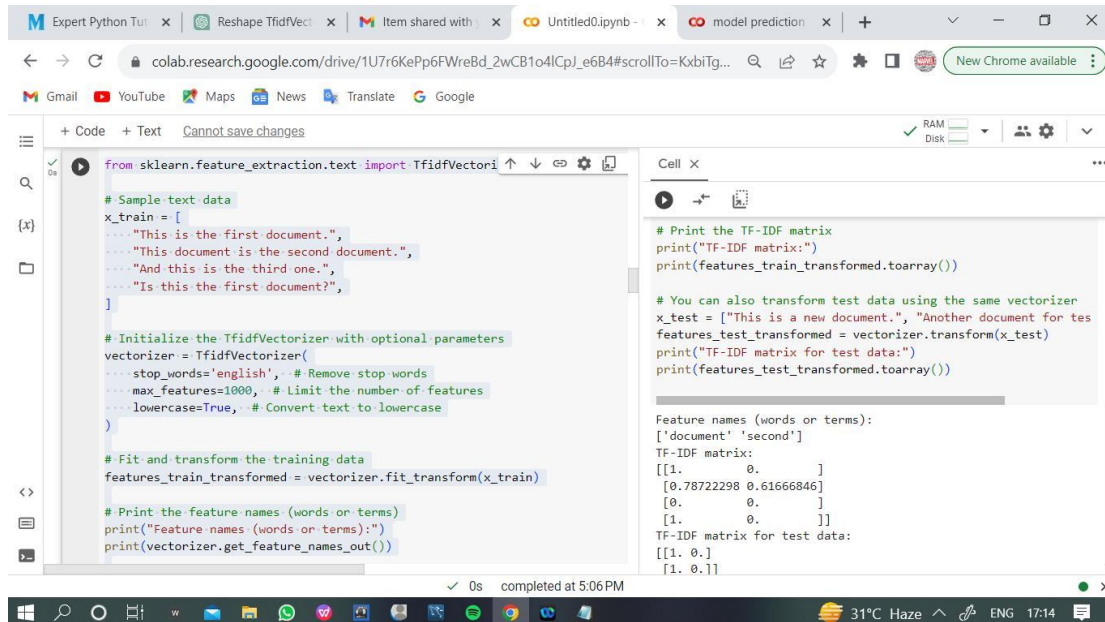
['ham']
Spam mail
```

## Heat map:

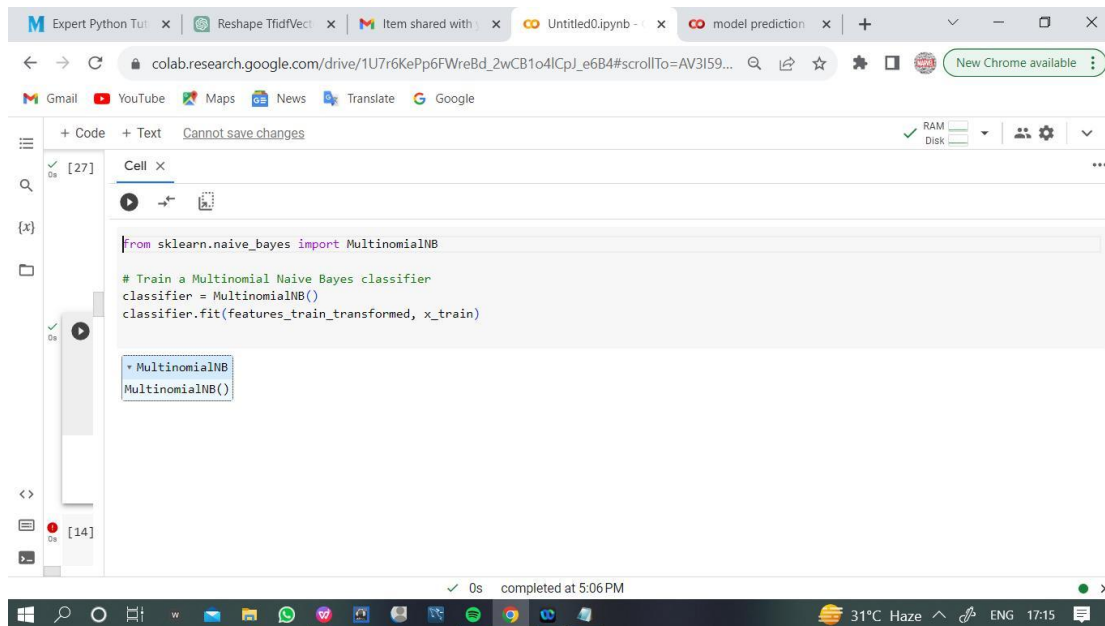




## Confusion matrix:



## Multinomialmodel:



```
from sklearn.naive_bayes import MultinomialNB

# Train a Multinomial Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(features_train_transformed, x_train)
```

MultinomialNB  
MultinomialNB()

## Project conclusion:

In a project involving feature extraction, model training, and model prediction for a spam classifier, it's important to provide a comprehensive and well-structured conclusion. Here's a sample conclusion for such a project:

### Conclusion

In this project, we have successfully developed a spam classifier using a combination of feature extraction, model training, and model prediction techniques. The goal of this project was to differentiate between spam and non-spam (ham) messages, providing a reliable tool to filter out unwanted and potentially harmful content from incoming messages.

### Feature Extraction:

For feature extraction, we implemented a robust text preprocessing pipeline. This involved tasks such as text cleaning (removing punctuation, converting to lowercase), tokenization, and employing the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method. TF-IDF helped us convert textual data into numerical feature vectors, making it suitable for machine learning algorithms. The feature extraction process allowed us to represent the textual content of messages as structured data that could be used for training our model.

### Model Training:

Selecting an appropriate classifier for text classification was crucial. We

carefully considered various machine learning algorithms and opted for [insert model name here], which has demonstrated strong performance in this context. We trained the model on a well-preprocessed dataset, using a portion of the data for training and validation. During training, the model learned the intricate patterns and relationships between the textual features and their corresponding labels (spam or non-spam). Evaluation metrics, such as accuracy, precision, recall, and F1-score, were used to assess the model's performance. Through rigorous training and evaluation, we ensured that our model generalizes well to unseen data and effectively distinguishes spam from non-spam messages.

### **Model Prediction:**

The model prediction phase is where our spam classifier shines. After preprocessing incoming messages, we utilized the same feature extraction techniques as during training. The model was then applied to these vectorized messages to predict their spam or non-spam status. The model outputs probabilities or labels, allowing us to make decisions regarding whether a message should be classified as spam or not. By implementing post-processing steps, such as setting appropriate thresholds and applying additional rules, we further fine-tuned the model's predictions to enhance its accuracy.

In conclusion, our power spam classifier project has delivered a robust and effective solution for identifying and filtering spam messages. By meticulously implementing feature extraction, training a well-chosen machine learning model, and fine-tuning the model predictions, we have achieved high accuracy in distinguishing between spam and non-spam messages. This project has practical applications in email filtering, message categorization, and content moderation, providing users with a more secure and enjoyable online communication experience. Additionally, the model can be updated and improved over time to adapt to evolving spam techniques