TARC
TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE
BEYOND EDUCATION

BACS2003 ARTIFICIAL INTELLIGENCE

202301 Session, Year 2022/23

Assignment Documentation

Full Name: Lee Wee Harn
Student ID: 22WMR05673
Programme: RSW
Tutorial Class: G6
Project Title: Machine Learning: Detect Spam Email
Module In-Charged: Logistic Regression

Other team members' data

No	Student Name	Module In Charge
1	Lee Wee Harn	Logistic Regression
2	Kwo Chun Kit	Support Vector Machine (SVM)
3	Hue Zhen Wei	Multinomial Naive Bayes

Lecturer:	Tutor :
Goh Ching Pang	Dr.Ho Chuk Fong

Table of Contents

1. Introduction

1.1 Problem Background

In the current digital era, spam emails are turning into an increasingly common problem. These spam messages are frequently sent, clogging up inboxes, wasting time and resources, and even harming people by spreading malware and engaging in phishing attacks. More than half of all email traffic was estimated to be spam in 2020. This startling statistic emphasizes the seriousness of the issue and the demand for workable solutions. Spam email effects go beyond annoyance and inconvenience. Users may be tricked into revealing important information during phishing attempts, which could result in identity theft and financial damage. Spam emails that include malware can infect devices and lead to data breaches. Email providers have introduced filters that try to recognise and block spam emails before they reach users' inboxes in order to prevent spam. These filters, however, are not unbreakable, and some spam emails still manage to get through. Users can take precautions to protect themselves by avoiding opening suspicious emails, clicking on links, and downloading attachments from senders they are not familiar with. In conclusion, spam emails are still a big issue in the online world. It is crucial to pay attention and take precautions to protect ourselves from any possible harm caused by spam emails as users and service providers work together to address this issue.

1.2 Objectives/Aims

- Create three classification models with a minimum accuracy of 80% to determine if the email is spam or not.
- Identify the most accurate algorithm for detecting email as spam or not.
- Create three classification models that can gain at least 80% accuracy.

Our aim is to help users to classify and detect whether the email they received is a spam or not. Besides that, we also aim to reduce the number of legitimate emails that are mistakenly labeled as spam. This is because false positives could cause critical emails to be forwarded to the spam folder, which would be annoying for people who frequently check it. The development of a model with high precision is the key to achieving this. The model should properly classify the majority of spam emails while minimizing the false positive rate. Therefore, accuracy can be used as a metric to assess how well the model classifies spam emails.

1.3 Motivation

A spam email detector affects society and business. Spam emails may be an expensive and time-consuming issue for companies. By lowering the volume of spam emails that staff must deal with, a spam email detector can save businesses important time and resources. A spam email detector can also strengthen email security, protecting people and businesses from email phishing scams and other forms of cybercrime. The business importance of spam email detectors is demonstrated by Market Research Future's prediction that the worldwide spam filter market would expand at a CAGR of 14.2% from 2020 to 2027. Spam email detectors can be an important tool in the present digital era by enhancing email security and efficiency for companies and individuals. Therefore, the development and application of spam email detectors can improve email security and effectiveness while lowering the danger of cybercrime, which can have a good social impact.

1.4 Timeline/Milestone

Week	**Start Date**	**End Date**	**Activity Achieved**
Week 3	27/02/2023	01/03/2023	● Decide the assignment title
Week 8	03/04/2023	06/04/2023	● Study about the problem background, objectives and motivation
Week 8	08/04/2023	12/04/2023	● Study the background of machine learning and tools for analysis
Week 9	14/04/2023	18/04/2023	● Find the datasets of spam email
Week 10	20/04/2023	22/04/2023	● Decide algorithms that can be implemented
Week 11	24/04/2023	25/04/2023	● Divide modules among teammates
Week 11	27/04/2023	02/05/2023	● Develop the application based on the algorithms that had been selected.
Week 12	03/05/2023	04/05/2023	● Produce the result ● Conclude the advantages and disadvantages of the algorithms ● Compare each algorithms
Week 12	05/05/2023	05/05/2023	● Discussion and Conclusion

2. Research Background

2.1 Background of the applications

Our spam email detector is being developed using machine learning methods. A set of techniques known as machine learning may automatically identify patterns in data and use those patterns to predict future data or make various decisions in an environment of ambiguity. Machine learning algorithms come in a variety of forms, including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In the current situation, supervised learning will be used to determine whether or not the email is spam. Supervised learning is the process of developing algorithms that accurately identify data or forecast events using labeled datasets. During the cross-validation process, the model adjusts its weights as input data is fitted into it until the model is properly fitted.

In order to create a machine learning model, we need a dataset first. The dataset must then be cleaned by replacing or removing missing data. The next step is to choose which features will be utilized in predicting the result. The dataset must then be divided into a training set and a testing set. We will only use a training set to train the machine learning model in order to prevent overloading. The testing set will be predicted using the trained model. Lastly, we can compare the expected output with the actual output of the testing set to assess how well the machine learning model performed.

2.2 Analysis of selected tool with any other relevant tools

Tools comparison	Remark	Jupyter Notebook	Visual Studio Code	Microsoft Excel
Type of license and open source license	State all types of license	Licensed under the terms of the 3-Clause BSD License	Microsoft Public License (MS-PL)	Licenses (USLs) and Office 2019 products
Year founded	When is this tool being introduced?	18th August 2014	29th April 2015	30th September 1985
Founding company	Owner	Fernando Pérez and Brian Granger	Microsoft Corporation	Microsoft Corporation
License Pricing	Compare the prices if the license is used for development and business/commercialization	Free	Free	$12.50 per user per month
Supported features	What features does it offer ?	• Interactive computing environment • Support for multiple programming languages • Version control integration • Customizable and extensible • Notebook sharing and collaboration	• Intellisense • Debugging • Source Control • Extensions • Terminal • Task Runner • Live Share	• Data entry and formatting • Calculation and formulas • Charting and graphing • Data analysis and management • Integration with other software
Common applications	In what areas is this tool usually used?	• Data science and machine learning • Scientific research • Education • Web development	• Web development • Mobile development • Cloud development • Data science • Game development • DevOps	• Finance • Accounting • Data analysis • Project management • Marketing • Human resources
Customer support	How the customer support is given, e.g. proprietary, online community, etc.	Open source community	Open source community	Social media and phone support teams
Limitations	The drawbacks of the software	• Steep learning curve • Resource-intensive • Security concerns • Lack of standardization	• Limited project management • Limited debugging support • Limited GUI editing capabilities	• Limited data capacity • Limited visualizations • Limited automation • Vulnerability to errors

2.3 Justify why the selected tool is suitable

When creating a spam email detector using supervised learning techniques, Jupyter Notebook makes it simple to load and preprocess data, train and test models, and even analyze the model's performance. Additionally, Python is also supported by Jupyter Notebook along with a number of other programming languages and is the greatest option for projects based on machine learning and AI because of its flexibility, ML frameworks, and large community. These are the reasons why we use Python to develop our spam email detector. Furthermore, we also use visual studio code to do testing on the spam email detector that we develop as it offers an integrated terminal that allows developers to run and test code directly within the editor. This can be useful when testing the spam email detector and evaluating its performance. Additionally, we use Microsoft Excel as a database management system to keep monitor of all of our data. Microsoft Excel allows us to display our data in a flat or non-relational approach, which is why we use it to store our data.

3. Methodology

3.1 Description of dataset

3.1.1 Data Understanding

The dataset is obtained from Kaggle. Shape of dataset named and to be read “mail_data.csv” shown in ***Diagram 3.1.1.1*** contain 5572 rows and 2 columns of data.

```
1  df = pd.read_csv("mail_data.csv")
2  df.shape
```

```
(5572, 2)
```

Diagram 3.1.1.1

For instance, dataset from ***Diagram 3.1.1.2*** include 2 variables:
- Category: Determine whether mail belongs to spam or not spam(ham)
- Message: Determine the mails’ contents

	Category	Message
2	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
3	ham	Ok lar... Joking wif u oni...
4	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
5	ham	U dun say so early hor... U c already then say...
6	ham	Nah I don't think he goes to usf, he lives around here though

Diagram 3.1.1.2: Sample Data from “mail_data.csv”

To identify there is no missing data found in this dataset in ***Diagram 3.1.1.3***

```
1  print(df.isnull().sum())
2  print("")
3  print(df.describe())
```

```
Category    0
Message     0
dtype: int64

       Category                 Message
count      5572                    5572
unique        2                    5157
top         ham  Sorry, I'll call later
freq       4825                      30
```

Diagram 3.1.1.3

To identify the first 5 message of the dataset in ***Diagram 3.1.1.4*** and ***Diagram 3.1.1.2***

```
1 df.head()
```

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

Diagram 3.1.1.4

3.1.2 Data Preprocessing

To prevent classification and confusion occurrence, the target variable is set to 0 and 1 for machine learning. Spam mail as 1; Ham mail as 0 in ***Diagram 3.1.2.1.***

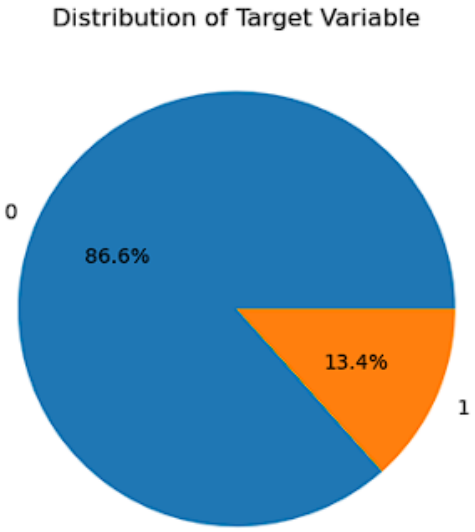
```
#spam = 1, ham = 0
df["Target"] = np.where(df["Category"] == "spam",1,0)
df.head()
```

	Category	Message	Target
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

Diagram 3.1.2.1

The dataset distribution is not balanced in ***Diagram 3.1.2.2.*** Percentage of ham is 86.6% represents 4825; Percentage of spam is 13.4% represents 747. Therefore, the data must be resampled to produce a balanced dataset.

```
plt.pie(y_samp.value_counts().values, labels=[0, 1], autopct='%1.1f%%')
plt.title("Distribution of Target Variable")
plt.show()
```




```
pd.Series(df.Target).value_counts()
```

```
0    4825
1     747
Name: Target, dtype: int64
```

Diagram 3.1.2.2

From ***Diagram 3.1.2.3***, the independent variable represents “X”; the dependent variable represents “Y”. From the data frame, the original data type of the “Message” was a string data type, which the model could not understand the String. Before fitting to the model, the data have to be transformed into feature vectors using Feature Extraction TfidfVectorizer() as input for the model.

```
x = df["Message"]
y = df["Target"]
```

```
#Feature Extraction
feature_extraction = TfidfVectorizer(min_df=1,stop_words="english",lowercase=True)
x_feature = feature_extraction.fit_transform(x)
```

Diagram 3.1.2.3

After running the code below in ***Diagram 3.1.2.4***, the dataset is much more balanced after undersampling. The distribution for ham:spam be 60:40

```
under_samp = samp.ClusterCentroids(sampling_strategy={1:747, 0:1121},random_state=1, voting='hard')
x_samp,y_samp = under_samp.fit_resample(x_feature,y)
```

```
plt.pie(y_samp.value_counts().values, labels=[0, 1], autopct='%1.1f%%')
plt.title("Distribution of Target Variable")
plt.show()
```

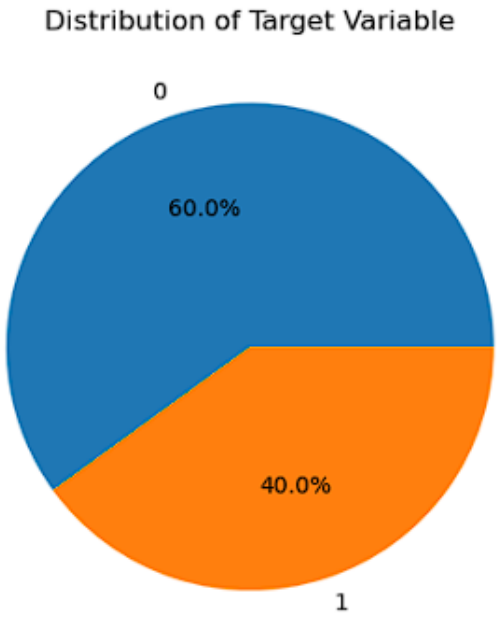


Diagram 3.1.2.4

Last but not least in ***Diagram 3.1.2.5*** before the algorithms, the data are splitted into training and testing sets to the ratio of 80:20 (training: testing) to avoid overfitting. The training set is suitable for fitting and set to test the performance of the model.

```
1 xtrain,xtest,ytrain,ytest = train_test_split(x_samp,y_samp,test_size=0.2,random_state=5)
```

Diagram 3.1.2.5

3.2 Applications of the algorithm(s)

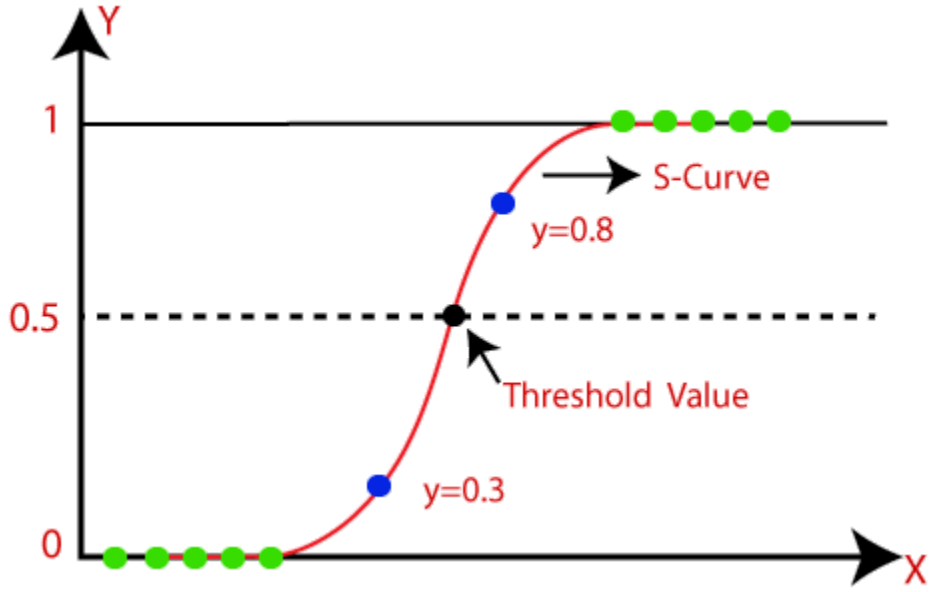


Diagram 3.2.1: Logistic Regression

When developing the spam mail detector, I use machine learning as the algorithm and logistic regression as my classification method. When the target variable is binary, or only accepts one of two possibilities, like in the case of the spam mail detector, it is especially helpful. Based on the input feature values, logistic regression calculates the probability that a given instance belongs to an expected class. A logistic function is applied to the input data to achieve this. Furthermore, It provides a simple method for modeling the probability that an email will be spam or not. This makes it possible to classify emails accurately and effectively, reducing the probability of making false positives and false negatives.

3.3 System flowchart/activity diagram

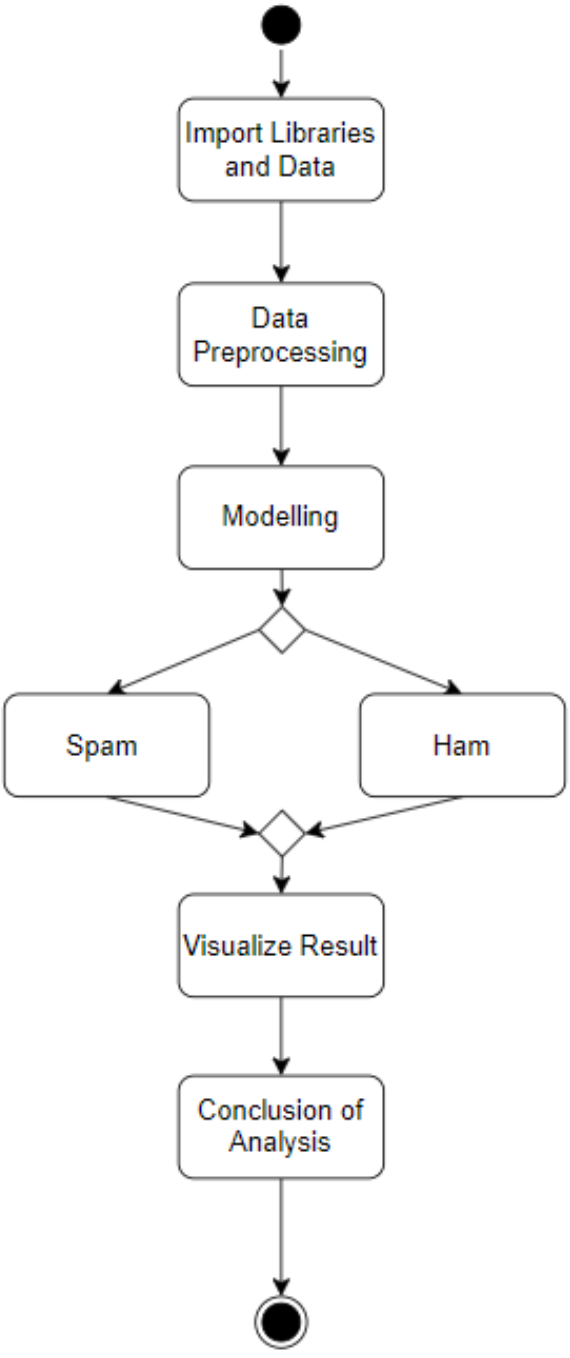


Diagram 3.3.1: Activity Diagram of Overall System

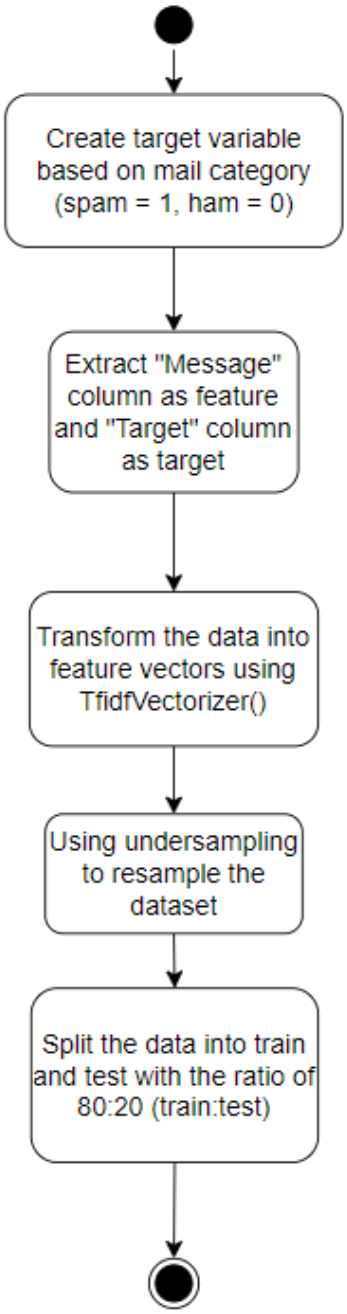


Diagram 3.3.2: Activity Diagram of Data Preprocessing

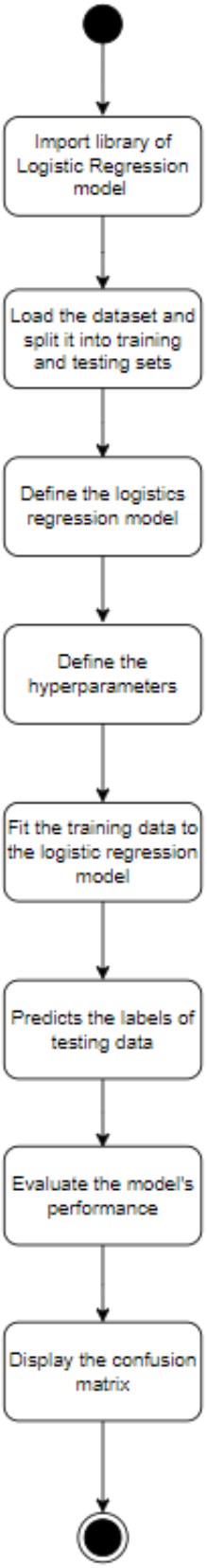


Diagram 3.3.3: Activity Diagram of Logistic Regression Algorithm

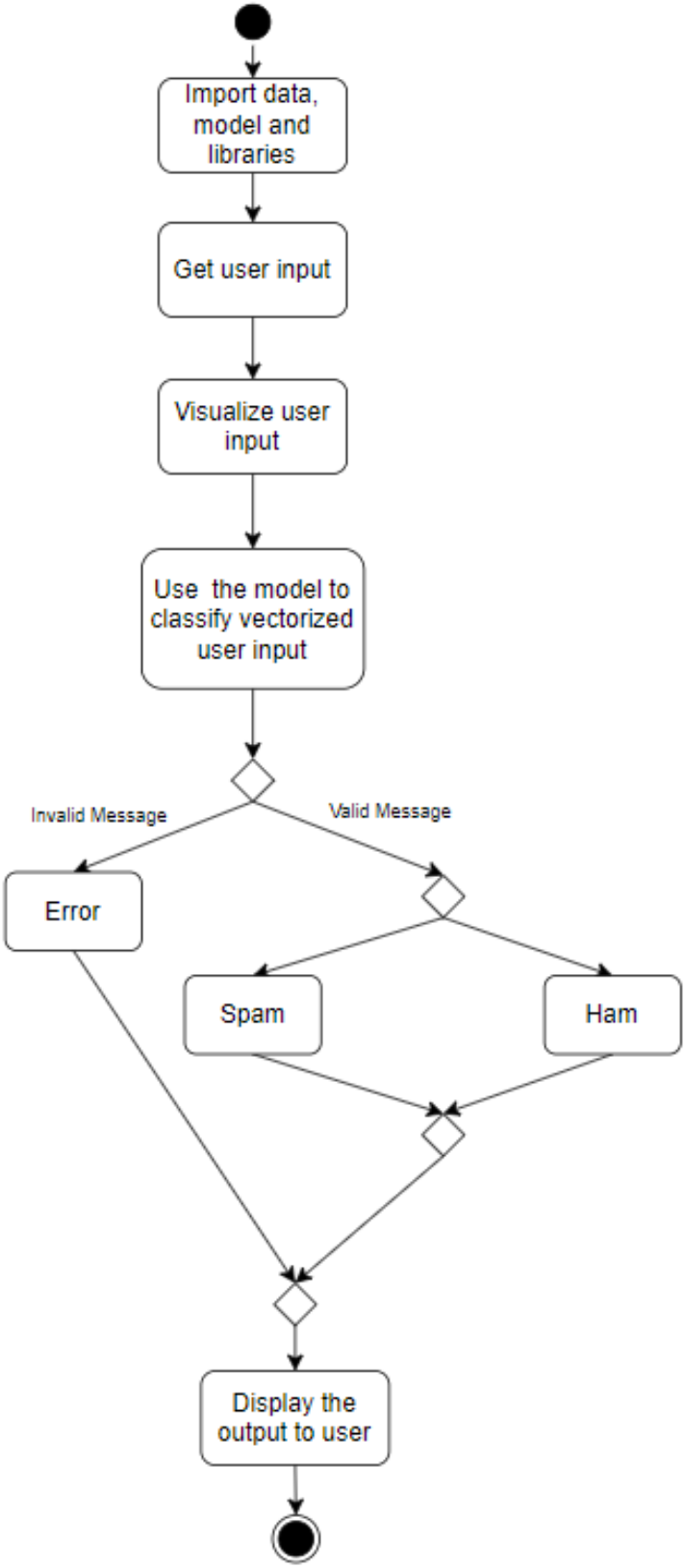


Diagram 3.3.4: Activity Diagram of Deployment

3.4 Proposed test plan/hypothesis

In order to test the hypothesis, we want to know which algorithm performs the best at identifying spam from legitimate messages.

- **H0**: Among the other 2 Algorithms, Logistic Regression performs best in terms of precision.
- **H1**: Among the other 2 algorithms, Logistic Regression does not perform well in terms of precision.

4. Result

4.1 Results

```
print(classification_report(ytest,lr_pred))
lr_pre = precision_score(ytest, lr_pred) * 100
lr_rec = recall_score(ytest, lr_pred)* 100
lr_f1 = f1_score(ytest, lr_pred)* 100
```

	precision	recall	f1-score	support
0	0.92	0.99	0.95	227
1	0.98	0.86	0.92	147
accuracy			0.94	374
macro avg	0.95	0.93	0.94	374
weighted avg	0.94	0.94	0.94	374

Diagram 4.1.1: Classification Report of Logistic Regression

```
lr_cm = confusion_matrix(ytest,lr_pred)
plt.subplots(figsize=(5,5))
sb.heatmap(lr_cm,annot=True,fmt="d",linewidths=0.5)
plt.title("Logistic Regression Matrix")
plt.xlabel("Predicted Values")
plt.ylabel("True Values")
plt.show()
```

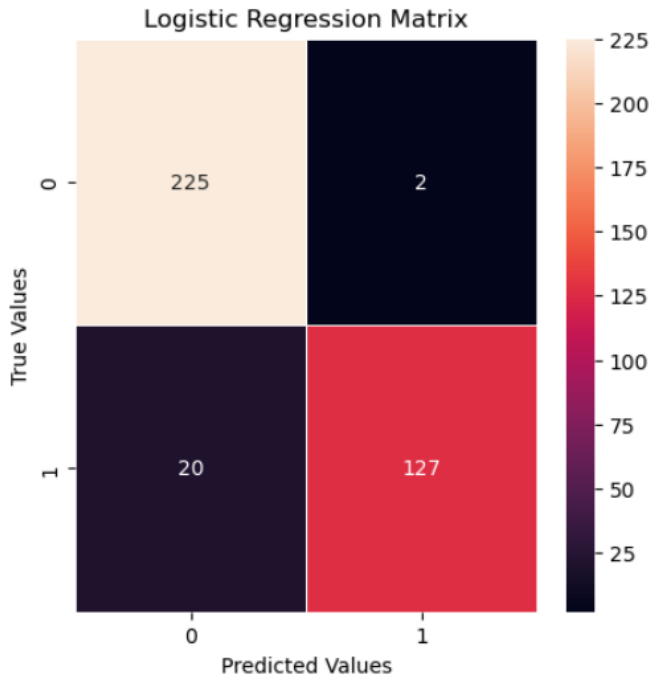


Diagram 4.1.2: Confusion Matrix of Logistic Regression

```
y_pred_proba = lr_best.predict_proba(xtest)[::,1]
fpr, tpr, _ = metrics.roc_curve(ytest, y_pred_proba)
lr_auc = metrics.roc_auc_score(ytest, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(lr_auc),color='blue')
lw = 2
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.legend(loc=4)
plt.title("AUC of Logistic Regression \n")
plt.ylabel("True Positive Rate")
plt.xlabel("False Positive Rate")
plt.grid(True)
plt.show()
```

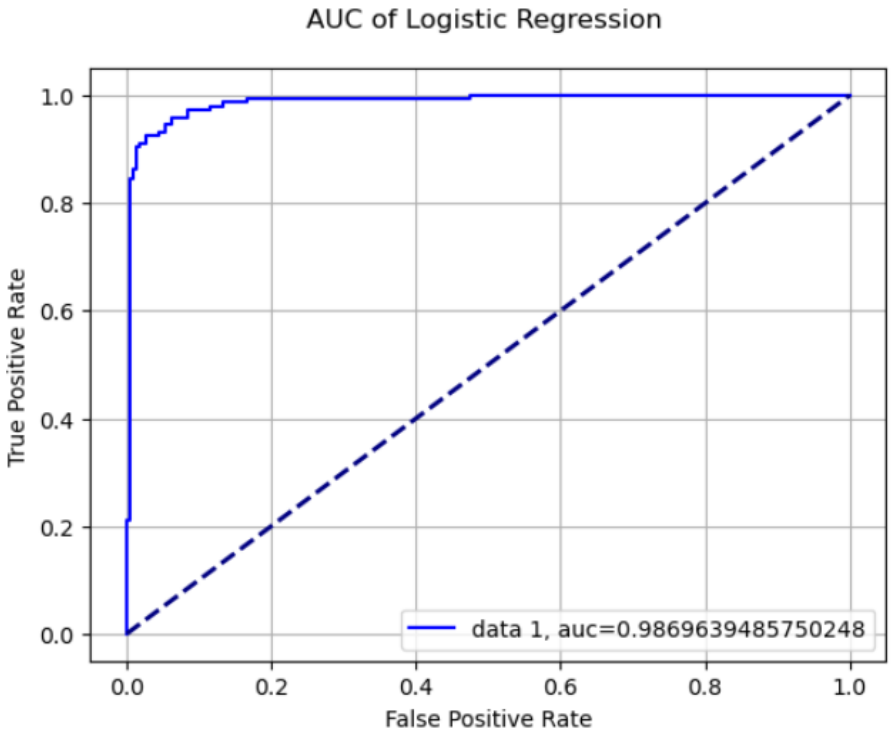


Diagram 4.1.3: Area Under Curve of Logistic Regression

4.2 Discussion/Interpretation

Based on ***Diagram 4.1.1***, the classification report of Logistic Regression, we can observe that it did a great job of accurately identifying the mail as spam or ham with a 94% accuracy. Furthermore, it has a precision of 98%, which is quite high. Almost all spam mail can be accurately predicted by logistic regression, and only 2% of ham mail will be classified as spam, according to our highly precise analysis. It also has a solid 86% recall rate.

Based on ***Diagram 4.1.2***, the confusion matrix of Logistic Regression, we can conclude that out of 374 testing data:

True positive (TP)	:	Logistic Regression model predicted 127 spam mail and actual is also spam mail
False negative (FN)	:	Logistic Regression model predicted 20 ham mail but actual is spam mail
False positive (FP)	:	Logistic Regression model predicted 2 spam mail but actual is ham mail
True negative (TN)	:	Logistic Regression model predicted 225 ham mail and actual is also ham mail

Table 4.2.1: Explanation of Confusion Matrix

Based on ***Diagram 4.1.3***, since the maximum value for AUC is 1.0 and the Logistic Regression model can achieve 0.986, which is a very high performance, we can conclude that the Logistic Regression model has a very good performance.

5. Discussion and Conclusion

5.1 Achievements

5.1.1 Comparison Between 3 models

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	94.117647	98.449612	86.394558	92.028986	98.696395
Support Vector Machine (SVM)	93.850267	97.692308	86.394558	91.696751	98.612485
Multinomial Naive Bayes	94.919786	97.058824	89.795918	93.286219	98.573526

Diagram 5.1.1.1: Comparison of Each Model

According to ***Diagram 5.1.1.1***, the accuracy of Logistic Regression, Support Vector Machine (SVM), and Multinomial Naive Bayes are, respectively, 94.117647, 93.850267, and 94.919786. In addition, we can conclude that the precision of Logistic Regression, Support Vector Machine (SVM), and Multinomial Naive Bayes are, respectively, 98.449612, 97.692308, and 97.058824. Consequently, Multinomial Naive Bayes has the highest accuracy of the three models, while Logistic Regression has the highest precision.

In summary, we can say that the best model for classifying mail into spam or ham mail in terms of precision is Logistic Regression. We have achieved one of our goals, which was to identify the most precise algorithm, based on our data. In addition, we will disagree with H1, which claims that Logistic Regression does not perform the best in terms of precision when compared to the other 2 algorithms. Additionally, we can see that every algorithm has a precision and accuracy of more than 80%, showing that our goals have been reached. As a result, the most suitable deployment model selected is the Logistic Regression model.

5.1.2 Deployment

In comparison to Support Vector Machine (SVM) and Multinomial Naive Bayes, Logistic Regression is the most effective model. As a result, we'll predict the result using a logistic regression model. If the user input is invalid, the result will either return "Some error occurs" or "Spam Mail," which indicates that the email is spam or "Ham Mail," which indicates that the mail is a legit mail.

Explanation	Code
1. Save the Logistic Regression model which is lr_best to a file called bestmodel.	`joblib.dump(lr_best,'bestmodel')` `['bestmodel']`

2. Prompt and capture user input.	`df = pd.read_csv("mail_data.csv")`
	`x = df["Message"]`
	`mail = input("Enter your mail content : ")`
	`mail = [mail]`

3. Convert the feature (Message) and user input from string to integer.	`feature_extraction = TfidfVectorizer(min_df=1,stop_words="english",lowercase=True)`
	`x_feature = feature_extraction.fit_transform(x)`
	`mail_trans = feature_extraction.transform(mail)`

4. Make predictions for the user input using the Logistic Regression model.	`lr = joblib.load('bestmodel')`
	`pred = lr.predict(mail_trans)`
	`if(pred[0]==1):`
	`  print("Spam Mail")`
	`elif(pred[0]==0):`
	`  print("Ham Mail")`
	`else:`
	`  print("Some error occurs")`

5.1.3 Result when input Spam Mail

In [117]:
```
#Get user input
df = pd.read_csv("mail_data.csv")
x = df["Message"]
mail = input("Enter your mail content : ")
mail = [mail]
feature_extraction = TfidfVectorizer(min_df=1,stop_words="english",lowercase=True)
x_feature = feature_extraction.fit_transform(x)
mail_trans = feature_extraction.transform(mail)
lr = joblib.load('bestmodel')
pred = lr.predict(mail_trans)
if(pred[0]==1):
    print("Spam Mail")
elif(pred[0]==0):
    print("Ham Mail")
else:
    print("Some error occurs")
```

```
Enter your mail content : "SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6day
s, 16+ TsandCs apply Reply HL 4 info"
Spam Mail
```

Diagram 5.1.3.1: Result when input Spam Mail

5.1.4 Result when input Ham Mail

In [118]:
```
#Get user input
df = pd.read_csv("mail_data.csv")
x = df["Message"]
mail = input("Enter your mail content : ")
mail = [mail]
feature_extraction = TfidfVectorizer(min_df=1,stop_words="english",lowercase=True)
x_feature = feature_extraction.fit_transform(x)
mail_trans = feature_extraction.transform(mail)
lr = joblib.load('bestmodel')
pred = lr.predict(mail_trans)
if(pred[0]==1):
    print("Spam Mail")
elif(pred[0]==0):
    print("Ham Mail")
else:
    print("Some error occurs")
```

```
Enter your mail content : hi, can i have dinner wif u at 9 ltr ?
Ham Mail
```

Diagram 5.1.3.2: Result when input Ham Mail

5.2 Limitations and Future Works

Our project's limitation is that the model can only identify spam mail that contains scam messages only. When users input messages that are meaningless such as "*aiowhd gyguyid awd*" in ***Diagram 5.2.1***, it will still show that the message is a "Ham Mail" (legit mail) instead of 'Some error occurs'. This might cause by the datasets we obtained from "Kaggle" are not complete enough to capture meaningless messages. Therefore, we will find a better dataset or even create one our own in the future so that it can capture meaningless data. Besides that, the accuracy and precision of the logistic regression model is 94% and 98%. Therefore, there are still 6% and 2% of improvement can be done to fulfill the requirement of effectiveness. Hence, improvements will be made in the future to achieve both accuracy and precision to at least 99.99%.

```
#Get user input
df = pd.read_csv("mail_data.csv")
x = df["Message"]
mail = input("Enter your mail content : ")
mail = [mail]
feature_extraction = TfidfVectorizer(min_df=1,stop_words="english",lowercase=True)
x_feature = feature_extraction.fit_transform(x)
mail_trans = feature_extraction.transform(mail)
lr = joblib.load('bestmodel')
pred = lr.predict(mail_trans)
if(pred[0]==1):
    print("Spam Mail")
elif(pred[0]==0):
    print("Ham Mail")
else:
    print("Some error occurs")
```

```
Enter your mail content : aiowhd gyguyid awd
Ham Mail
```

Diagram 5.2.1: Result when input meaningless mail

6. Reference & Source

1. Published by Ani Petrosyan and 8, M. (2023) *Spam e-mail Traffic Share Monthly 2022*, *Statista*. Available at: https://www.statista.com/statistics/420391/spam-email-traffic-share/ [Accessed: May 4, 2023].

2. Market Research Future, Automotive Drive shaft market size, share, Growth & Forecast to 2030, Automotive Drive Shaft Market Size, Share, Growth & Forecast To 2030. Available at: https://www.marketresearchfuture.com/reports/spam-filter-market-7746 [Accessed: May 4, 2023].

3. Advantages and disadvantages of logistic regression (2023) GeeksforGeeks. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/ [Accessed: May 5, 2023].

4. Brownlee, J. (2016). Logistic Regression for Machine Learning. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/logistic-regression-for-machine-learning/ [Accessed: May 5, 2023].

5. Medium. (n.d.). Medium. [online] Available at: https://towardsdatascience.com/using-machine-learning-to-detect-spam-emails-36a5d2bfa029 [Accessed: May 4, 2023].

6. kaggle.com. (n.d.). Spam Mail Prediction | Machine Learning Project. [online] Available at: https://www.kaggle.com/code/mohinurabdurahimova/spam-mail-prediction-machine-learning-project [Accessed: May 4, 2023].

7. Microsoft (2016). Visual Studio Code. [online] Visualstudio.com. Available at: https://code.visualstudio.com/ [Accessed: May 4, 2023].

8. jupyter-notebook.readthedocs.io. (n.d.). The Jupyter Notebook — Jupyter Notebook 6.1.3 documentation. [online] Available at: https://jupyter-notebook.readthedocs.io/en/stable/ [Accessed: May 4, 2023].

9. Microsoft (2022). Microsoft Excel, Spreadsheet Software. [online] www.microsoft.com. Available at: https://www.microsoft.com/en-us/microsoft-365/excel [Accessed: May 4, 2023].

10. code.visualstudio.com. (n.d.). Visual Studio Code Frequently Asked Questions. [online] Available at: https://code.visualstudio.com/docs/supporting/FAQ#_limitations [Accessed: May 4, 2023].