TARC
TUNKU ABDUL RAHMAN
UNIVERSITY COLLEGE

BEYOND EDUCATION

BACS3074 ARTIFICIAL INTELLIGENCE

202201 Session, Year 2021/22

Assignment Documentation

Full Name: Wong Sai Seng
Student ID: 21WMR05340
Programme: RST2
Tutorial Class: G2
Project Title: Natural Language Processing
Module In-Charged: Data Crawling, Naive Bayes algorithm, WordCloud

Other team members' data

No	Student Name	Module In Charge
1	Wong Rong Kai	Data preprocessing, K-Nearest Neighbour algorithm

Lecturer: Dr. Goh Ching Pang	**Tutor : Puan Noor Aida** **Dr. Goh Ching Pang**

1. Introduction

1.1. Problem Background

In this high-risk period of covid-19 in our country in Malaysia. This makes people stay at home and reduce their time for going out to the shopping mall or market during their leisure time. For this situation, online shopping has become very popular in the last few years because people are lazy and scared to go out. So that as a seller, we need to analyse the customer feeling about the product that sells on the amazon website. Therefore, sentiment analysis allows us to extract, score, categorise, and visualise the emotions and opinions expressed by our own consumers in their evaluations. Are they expressing good, negative, or neutral thoughts or feelings? Brands may utilise consumer sentiments and views as a roadmap to improve the products and services. In this case, we are going to do a sentimental analysis for the customers that buy the face mask product on the amazon website.

1.2. Objectives/Aims

Develop a sentimental analysis using 2 different algorithm

1. Naive Bayes algorithm
2. K-Nearest Neighbour algorithm

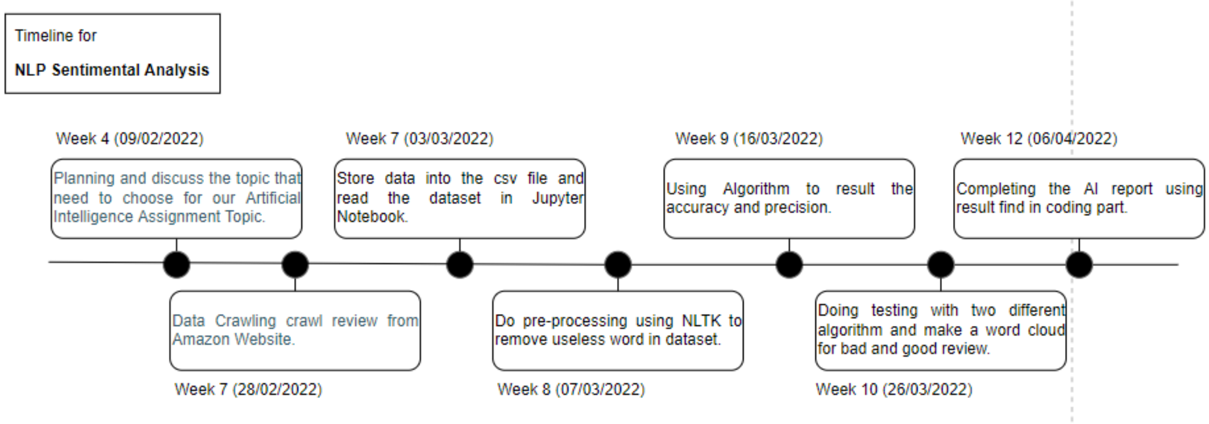
To compare the advantages and disadvantages between two algorithms.

1.3. Motivation

Since sentiment analysis is used in analysing customer feedback, survey responses, and product reviews nowadays. As a beginner to study sentiment analysis using python, we aim to investigate which algorithms will get better results.

The motivation for Sentiment Analysis, according to (Ramteke et al., 2012), is twofold. Consumers and manufacturers alike place a high value on "customer feedback" on products and services. As a result, both industry and academics have put out significant effort in the field of sentiment analysis. Researchers discovered that internet reviews had a considerable impact on customers' purchasing decisions (C. Ware, 2014). As a result, product evaluations and comments are a key indicator of client happiness, which has an impact on the company's performance. Therefore, we need to find out if the customers will continue to shop to the product by analysing the reviews in the dataset.

1.4. Timeline/Milestone


Timeline for
NLP Sentimental Analysis
Week 4 (09/02/2022)
Planning and discuss the topic that need to choose for our Artificial Intelligence Assignment Topic.
Week 7 (03/03/2022)
Store data into the csv file and read the dataset in Jupyter Notebook.
Week 9 (16/03/2022)
Using Algorithm to result the accuracy and precision.
Week 12 (06/04/2022)
Completing the AI report using result find in coding part.
Data Crawling crawl review from Amazon Website.
Week 7 (28/02/2022)
Do pre-processing using NLTK to remove useless word in dataset.
Week 8 (07/03/2022)
Doing testing with two different algorithm and make a word cloud for bad and good review.
Week 10 (26/03/2022)

2. Research Background

2.1. Background of the applications

Sentiment Analysis is a prominent NLP approach that identifies whether data is positive, negative, or neutral based on a piece of text, for example a remark, review, or document. It has a wide range of applications in healthcare, customer service, finance, and other industries (Frank Andrade, 2021).

We can examine text at varying degrees of depth, depending on our objectives.For example, We might use the average emotional tone of a bunch of reviews to figure out what proportion of people enjoyed our new apparel line. If we want to discover why visitors like or dislike a certain garment, or whether they compare it to comparable goods from other companies, we will need to examine each review phrase for specific elements and keyword usage.

Sentiment analysis has a wide range of applications and is quite powerful. The capacity to derive insights from social data is becoming increasingly popular among businesses throughout the world (Kristian Bannister, 2018).

Sentiment Analysis Use Cases

1. Segment Buyer Groups Based on Opinion

The ability to track consumer sentiment helps a company to understand which customers are more opinionated than others. For example, many people estimate that 20% of customers cause 80% of customer problems. If this statistic is correct, you will be able to segregate the traits of that group and either remedy frequent problems or avoid them altogether. Of course, eliminating purchasers would imply that, depending on the level/type of opinions of that group, there is little to no ROI.

2. Plan Product/Service Improvements:

   Customer feedback is a gold mine of information, especially when it comes to what you sell. Customer feedback might inspire you to update software, improve the design of tangible things, or improve your services. This information can sometimes lead to new products or services for your company to provide.

3. Plan Process Improvements

   Customer satisfaction isn't always high. Negative feedback, on the other hand, isn't always untrue. These viewpoints may require systematic categorization, implying an improvement in your entire customer service (or other) procedure.

4. Continuously Track Sentiment Over Time

   Sentiment is a measure that should be monitored on a regular basis. Opinions will shift as you enhance your procedures and goods. Seeing these shifts allows you to better navigate the emotional seas.

2.2. Analysis of selected tool with any other relevant tools

Tools comparison	**Remark**	Jupyter Notebook	Docker Desktop	Google Docs	Draw IO
Type of licence and open source license	State all types of license	BSD licence (Open Source)	Open source	Open source	Open Source
Year founded	When is this tool being introduced?	2014	2008	2006	2000
Founding company	Owner	Project Jupyter (Fernando Pérez)	Scott Johnston	Google LLC	JGraph Ltd
License Pricing	Compare the prices if the license is used for development and business/commercialization	Free	Persona - Free Pro - $5 / month Team - $7/user/month Start with a minimum of 5 users - $25. Business - &21/user/month	Free	Free
Supported features	What features that it offers?	to visualise charts generated by running code cells To enable this feature, the IPython kernel is intended to work in	Networking that is defined by software. The Docker CLI and Engine enable operators to establish	Google Docs is a web-based word processor that allows users to create notebooks, diagrams, and	Flowcharts, wireframes, UML diagrams, organisational charts, network diagrams, and other

		tandem with the matplotlib charting library. The kernel has support for specific charting libraries.	segregated networks for containers without touching a single router. Operators and developers create systems with sophisticated network topologies and specify them in configuration files.	highlight markdown. It's best for collaboration and document creation utilising pre-made templates.	diagrams are all possible.
Common applications	In what areas this tool is usually used?	Data science Scientific computing Computational journalism Machine learning	Standardised executable components that combine application source code with the OS libraries and dependencies needed to run that code in any environment.	Create documentation	Used to collaborate with other users in real-time while creating and editing diagrams, flowchart, barchart etc..
Customer support	How the customer support is given, e.g. proprietary, online community, etc.	Stack Overflow	Docker community	Feedback to google	Email
Limitations	The drawbacks of the software	Non-linear workflow Hard to test long asynchronous tasks. Long asynchronous jobs are difficult to test. There is less security. It causes the cells to run out of sequence. There is no IDE integration, no linting, and no code-style correction in the Jupyter notebook.	Containers aren't appropriate for all applications.	The picture you insert cannot be larger than 50MB and must be in one of the following formats:.jpg,.png, or.gif. There are constraints in terms of size. Regardless of the amount of pages or font size, up to 1.02 million characters are possible. A text document converted to Google Docs format can be up to 50 MB in size. To edit documents, you must be connected to the internet; there is no offline mode.	When you open an existing diagram, the view is occasionally in an odd place, and managing the Z order of shapes can be difficult.

2.3. Justify why the selected tool is suitable

First of all, Docker Desktop is a simple-to-use programme for building and sharing containerized apps and microservices on your Mac or Windows computer. To crawl the data to the csv file, we have to use the docker desktop to access the amazon website to crawl the product review.

For the Jupyter Notebook, I used this software to do my coding assignment. In the Jupyter Notebook, I had been using it to read the dataset that I grab by using the docker desktop, doing the Naive Bayes algorithm and wordCloud. Beside that, the dataset pre-processing and the K-Nearest Neighbour algorithm have also been done using the Jupyter Notebook.

Beside that, the draw io is a software to let us draw some graphs or charts. In this report, I used the draw io to make some charts like timeline diagrams, flowchart, and etc. Furthermore, the google document is used to write the report after we finish the coding part and get the result in the Jupyter Notebook.

3. Methodology

3.1. Description of dataset

Data collection

The dataset used in this assignment is a set of product reviews that collected from the amazon.com website. In the limited time within these 14 weeks for the semester, there are a total of 3270 rows of data that contain in the dataset. The dataset in our project is using web crawling to grab customers' reviews from amazon website. In the dataset, the following features are shown below:

- Product (Product name for the review, eg. 100Pcs Disposable Face Masks)
- Title (Review title/ summary)
- Rating (Rating for the review)
- Content (Review text from the customers)

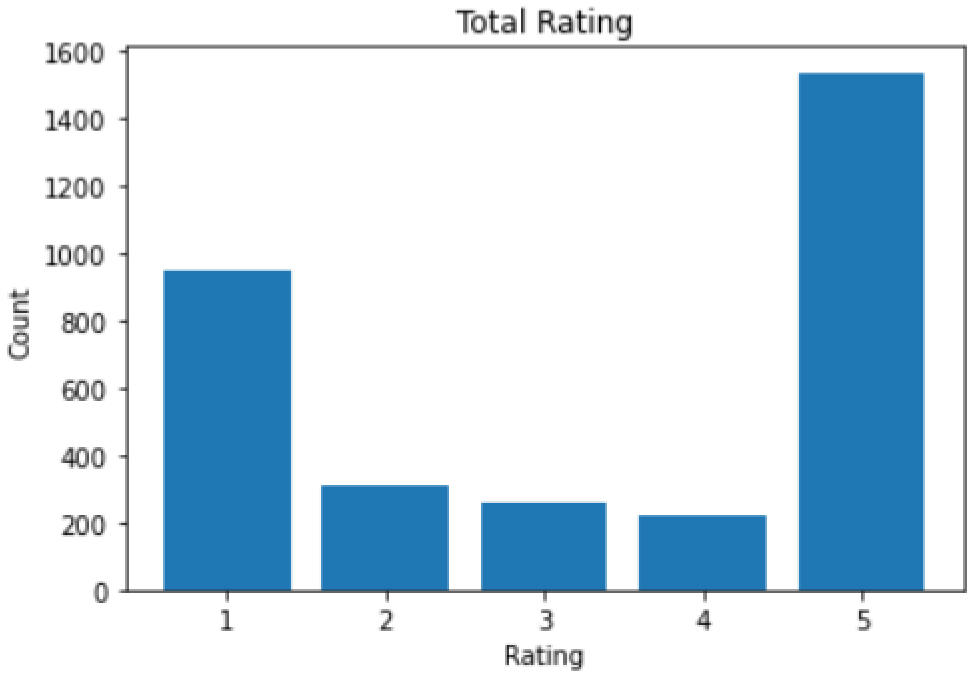
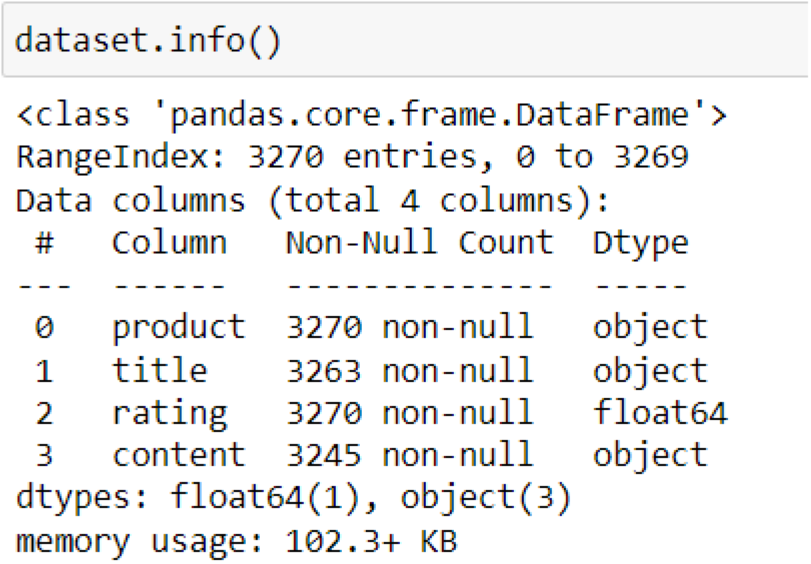


Figure 3.1 Total Rating in the dataset

In the figure 3.1 shown above is the total number of ratings in the dataset crawl on the product on amazon website. Besides that, there are ratings from 1 to 5 that are rated by the buyers that buy the product in amazon.



```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3270 entries, 0 to 3269
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   product  3270 non-null   object 
 1   title    3263 non-null   object 
 2   rating   3270 non-null   float64
 3   content  3245 non-null   object 
dtypes: float64(1), object(3)
memory usage: 102.3+ KB
```

Figure 3.2 Data type for each Column in data set

In the figure 3.2 shown above, there are 4 columns included in the dataset which are product, title, rating and content. For those columns, there are different data types applied to each item. As figure 3.2 shown, the data types for product, title and content is object and rating is float data type.

3.2. Applications of the algorithm(s)

Naive Bayes Classifier

For a huge amount of data, Naive Bayes is the simplest and fastest categorization algorithm. The Naive Bayes classifier has been effectively employed in a variety of applications, including spam filtering, text classification, sentiment analysis, and recommendation systems (Hardikkumar Dhaduk, 2021). For unknown class prediction, it employs the Bayes probability theorem. Assume that we want to categorise user reviews as good or bad in a dataset. Therefore, data scientists are frequently called upon to undertake sentiment analysis.

The Naive Bayes algorithm is a supervised learning technique for addressing classification issues that is based on the Bayes theorem. It is mostly utilised in text classification tasks that need a large training dataset. The Naive Bayes Classifier is a simple and effective classification method that aids in the development of rapid machine learning models capable of making quick predictions. It's a probabilistic classifier, which means it makes predictions based on an object's likelihood.

Naive :

- It implies that the appearance of one feature is unrelated to the appearance of other characteristics. If the colour, shape, and flavour of the fruit are used to identify it, a red, spherical, and sweet fruit is identified as an apple. As a result, each aspect helps to identify that it is an apple without relying on the others.

Bayes :

- Since it's based on the Bayes' Theorem concept.

$$\mathrm{P(A|B)}=\frac{P(B|A)P(A)}{P(B)}$$

Figure 3.2.1 Bayes' theorem formula

Naive Bayes Algorithm .

1. Multinomial Naïve Bayes
   i. When we have discrete data, we use it for example in movie ratings ranging 1 and 5 as each rating will have a certain frequency to represent. The count of each word is used in text learning to predict the class or label.

2. Gaussian Naïve Bayes
    i. When all of our characteristics are continuous, Gaussian Naive Bayes are employed because of the normal distribution assumption. Sepal width, petal width, sepal length, and petal length, for example, are characteristics in the Iris dataset. As a result, the breadth and length of its characteristics may vary throughout the data set. We can't represent characteristics in terms of how frequently they occur. This indicates that the data is updated on a regular basis. As a result, we apply Gaussian Naive Bayes in this case.
3. Bernoulli Naïve Bayes
    i. It assumes that all of our characteristics are binary, with only two possible values. 0s denote "word does not appear in the document" and 1s denote "word appears in the document."

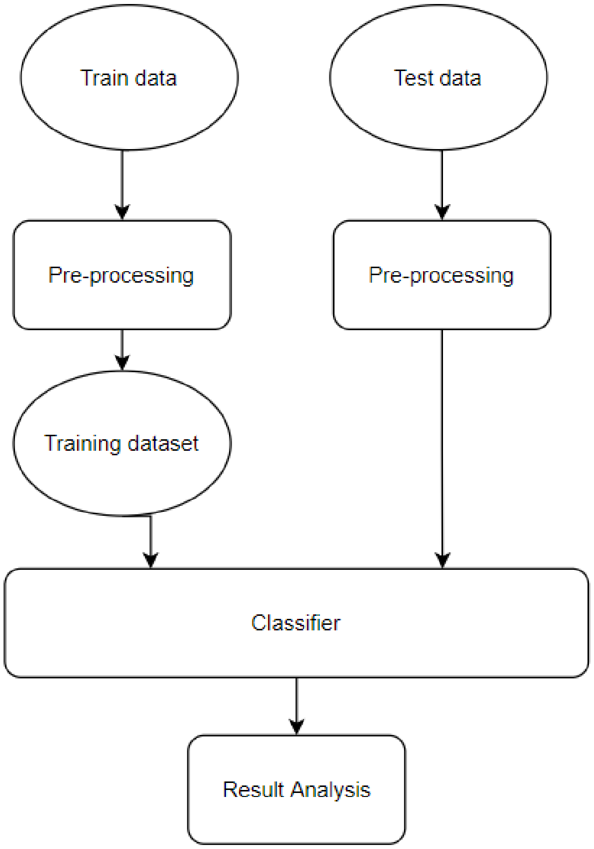


Figure 3.2.2 Sentiment Analysis Methodology

In the figure 3.2.2, it shows the flow for the sentiment analysis methodology. In the figure shown there are two types of data which are train data and test data. Those data should go through the process which is pre-processing. The adjustments we apply to our data before feeding it to the algorithm are referred to as pre-processing. So, Data preprocessing is a method for converting unclean data into a clean data collection. In other words, anytime data is received from various sources, it is collected in raw format, which makes analysis impossible. After that, the train data should go through another process called training dataset. It is to increase AI models' decision-making abilities. Therefore, it will classify the text and it is to classify the text into a set of words Text classification uses natural language processing (NLP) to evaluate text and assign a set of predetermined tags or categories depending on its context. Lastly, is finally to do the result analysis after training and testing the data.

3.3. System flowchart/activity diagram

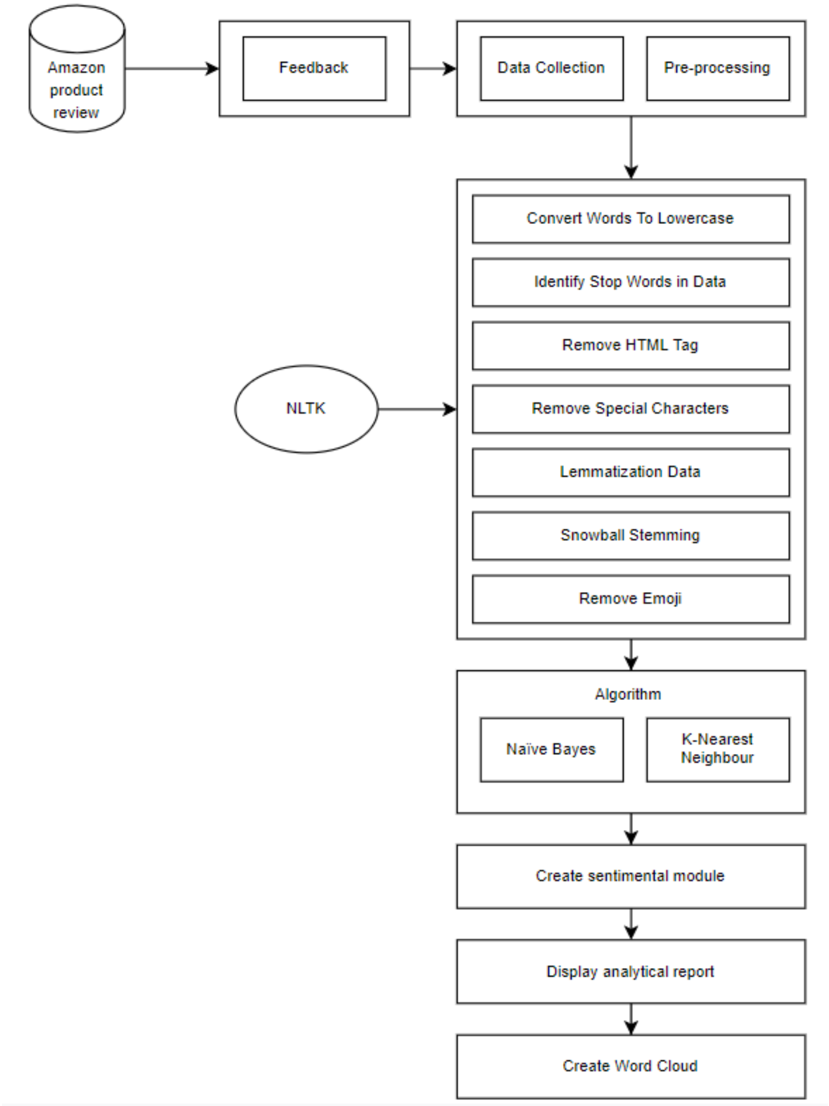


Figure 3.3.1 System Flowchart/Activity Diagram

`Read Dataset

- Pandas is a python library and it can be used to analyse data. We will Import pandas as pd to read the dataset content. Since our dataset is in csv format, therefore we have to use read_csv to read the csv data.

Remove NA/ null value

- Dropna function is used to drop the null value in the dataset. After reading the dataset content, we will drop the row with null value or empty to increase the accuracy score while we do the sentimental analysis.

Replace the rating to 0 and 1

- After dropping the null value row, we have replaced the rating to become 0 and 1. For the rating that is 1 and 2 in the dataset, we will use the loc function to replace them into 0. In this means, the rating for the 1 and 2 we will assume that they are negative reviews. Besides that, we will replace rating with 4 and 5 to become 1 as positive reviews. After that, the rating for 3 will be removed from the dataset when divided into train and test dataset.

Divide the dataset into train and test

- In this part, a total 75% of the dataset will be used to train and another last 25% will be used for testing. The reason why training data is larger than testing data is because the test data is used to evaluate the trained model's performance on an unknown dataset, whereas the training data is used to learn the model. The task of learning the model is more difficult than judging its performance. If the training data set is small, the model is prone to overfitting and may struggle to generalise to new data patterns not observed in the training data. As a result, model performance often improves as the amount of the training dataset grows.

WordCloud

- In the wordCloud section, the magnitude of each word represents its frequency or relevance in a word cloud, which is a data visualisation tool for visualising text data. A word cloud can be used to emphasise important textual data points. Data from social networking websites is frequently analysed using word clouds. Besides that, I have shown the word for all ratings from 1 to 5, negative review (1 to 2), and positive review (4-5) by using word Cloud.

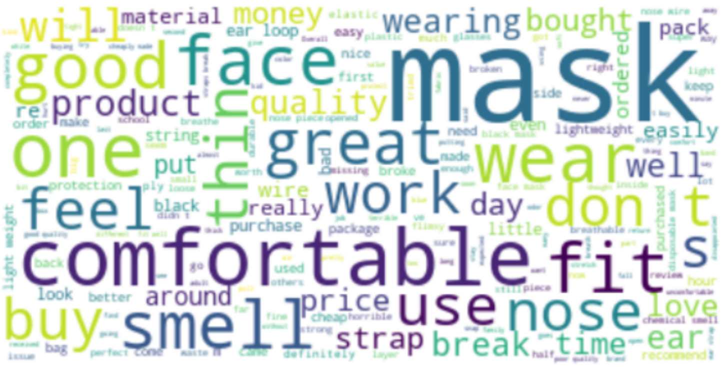


Figure 3.3.2 All Review Content

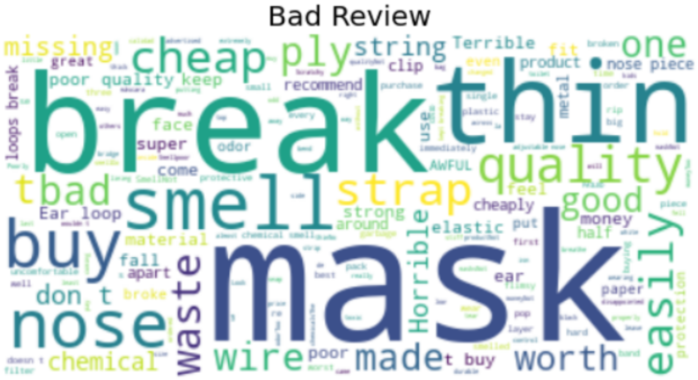


Figure 3.3.3 Bad Review Content

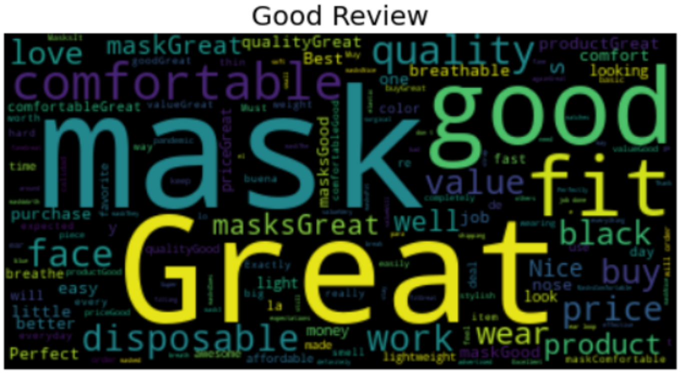


Figure 3.3.4 Good Review Content

3.4. Proposed test plan/hypothesis

Train data : 75% in the dataset of data used to train the AI to do the sentiment analysis.
Test data : 25% in the dataset of data used to test the result that the AI can predict the sentiment analysis or not.

To test the accuracy score for :

1. Multinomial Naive Bayes
2. Gaussian Naive Bayes
3. Bernoulli Naive Bayes

To calculate the Precision, recall, f1-score and support for

1. Multinomial Naive Bayes
2. Gaussian Naive Bayes
3. Bernoulli Naive Bayes

Calculate AUC for :

1. Multinomial Naive Bayes
2. Gaussian Naive Bayes
3. Bernoulli Naive Bayes

Display Result ROC for :

1. Multinomial Naive Bayes
2. Gaussian Naive Bayes
3. Bernoulli Naive Bayes

Display Confusion Matrix for :

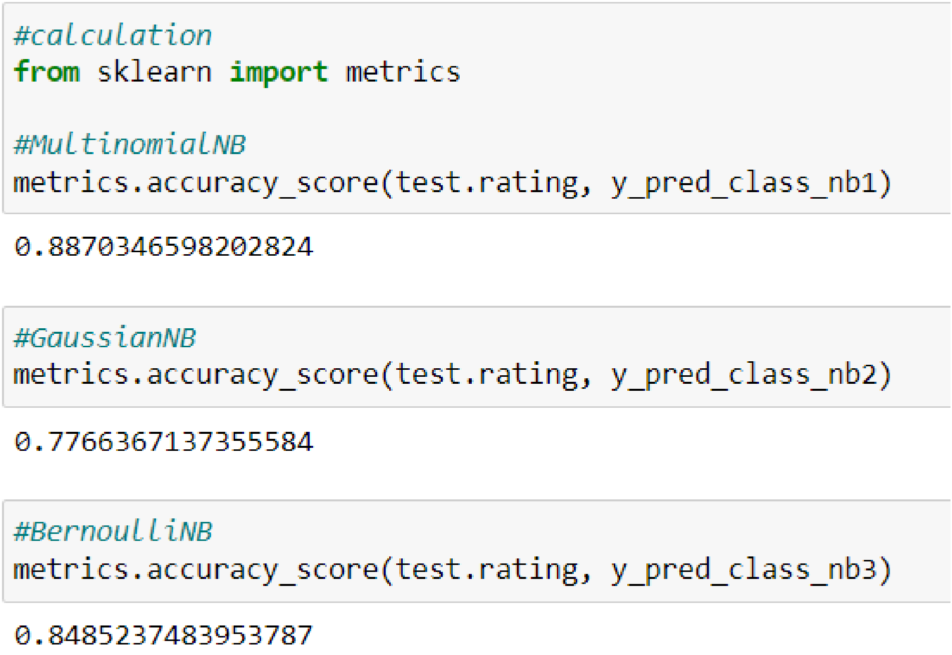
1. Multinomial Naive Bayes
2. Gaussian Naive Bayes
3. Bernoulli Naive Bayes

Test the prediction after using the naive bayes algorithm and the knn-algorithm.
The input need to test with both algorithm :

1. "I feel disappointed to this mask "
2. "This mask is perfect, I like it."

4. Result

4.1. Results



```
#calculation
from sklearn import metrics

#MultinomialNB
metrics.accuracy_score(test.rating, y_pred_class_nb1)
```

```
0.8870346598202824
```

```
#GaussianNB
metrics.accuracy_score(test.rating, y_pred_class_nb2)
```

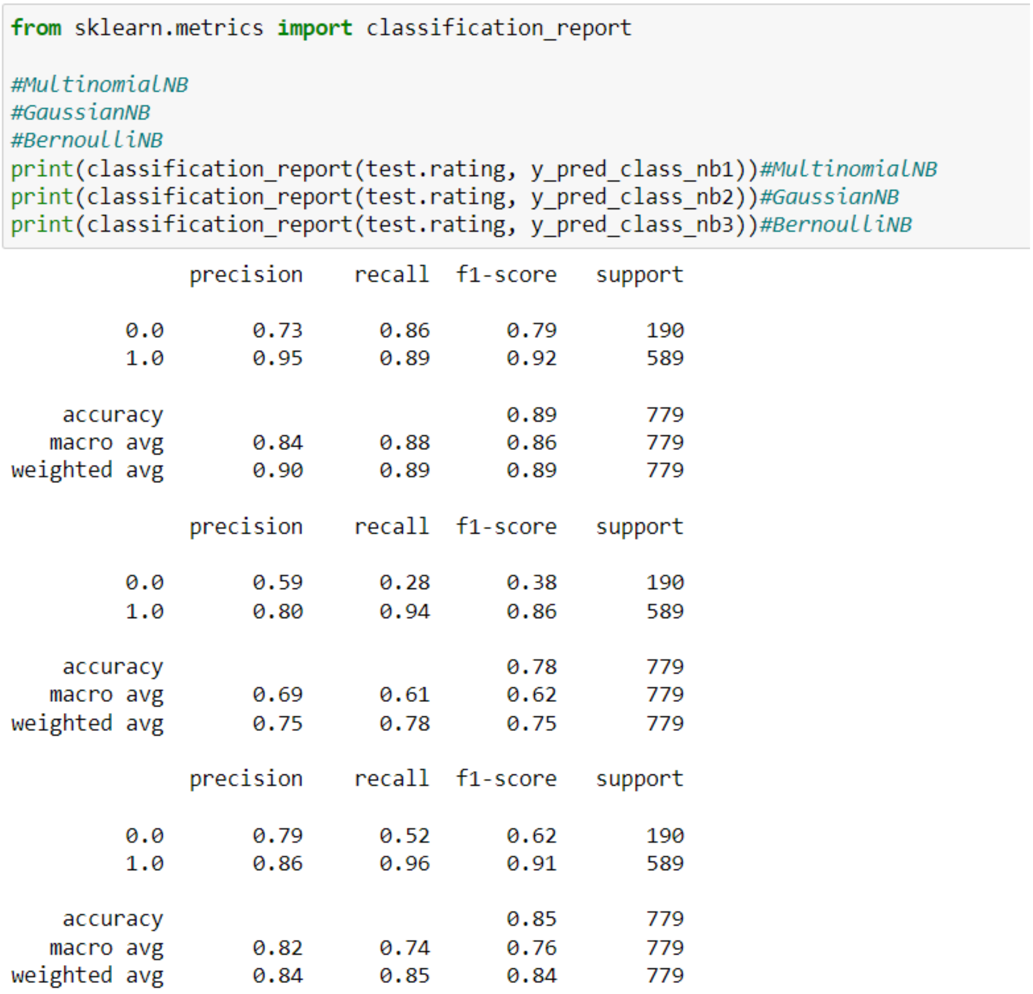
```
0.7766367137355584
```

```
#BernoulliNB
metrics.accuracy_score(test.rating, y_pred_class_nb3)
```

```
0.8485237483953787
```

Figure 4.1.1 Accuracy Score for Multinomial, Gaussian and Bernoulli Naive Bayes

Figure 4.1.1 above shows the accuracy score for Multinomial, Gaussian and Bernoulli Naive Bayes. In Python Scikit learn, the accuracy score method is used to calculate the accuracy of either the fraction or count of accurate predictions. It reflects the ratio of the total of true positives and true negatives among all the forecasts mathematically.



```
from sklearn.metrics import classification_report

#MultinomialNB
#GaussianNB
#BernoulliNB
print(classification_report(test.rating, y_pred_class_nb1))#MultinomialNB
print(classification_report(test.rating, y_pred_class_nb2))#GaussianNB
print(classification_report(test.rating, y_pred_class_nb3))#BernoulliNB
```

```
              precision    recall  f1-score   support

         0.0       0.73      0.86      0.79       190
         1.0       0.95      0.89      0.92       589

    accuracy                           0.89       779
   macro avg       0.84      0.88      0.86       779
weighted avg       0.90      0.89      0.89       779

              precision    recall  f1-score   support

         0.0       0.59      0.28      0.38       190
         1.0       0.80      0.94      0.86       589

    accuracy                           0.78       779
   macro avg       0.69      0.61      0.62       779
weighted avg       0.75      0.78      0.75       779

              precision    recall  f1-score   support

         0.0       0.79      0.52      0.62       190
         1.0       0.86      0.96      0.91       589

    accuracy                           0.85       779
   macro avg       0.82      0.74      0.76       779
weighted avg       0.84      0.85      0.84       779
```

Figure 4.1.2 Precision, recall, f1-score and support for Multinomial, Gaussian and Bernoulli Naive Bayes

Figure 4.1.2 shown above is the precision, recall, f1-score and support for Multinomial, Gaussian and Bernoulli Naive Bayes. The accuracy score is calculated by dividing the number of properly identified data instances (true positives) by the total number of data instances as the figure 4.1.3 shown below.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

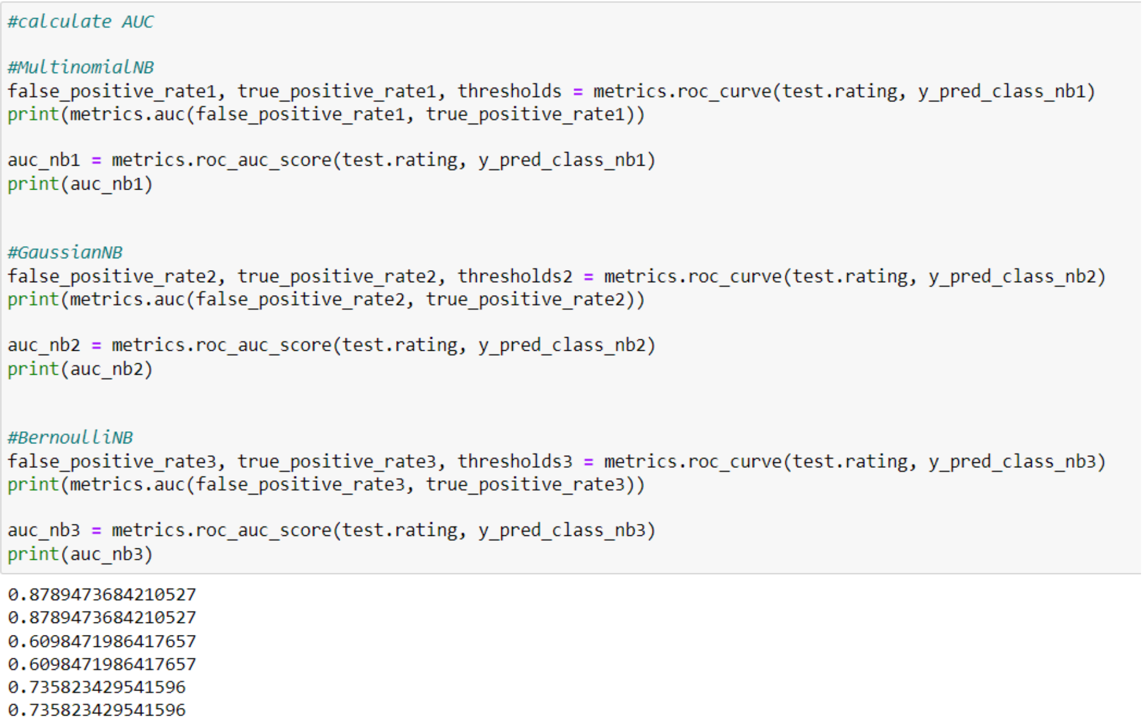
Figure 4.1.3 Precision calculation

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

Figure 4.1.4 Recall calculation

$$\text{F1 score} = \frac{\text{Precision * Recall}}{\text{Precision + Recall}} \text{ *2}$$

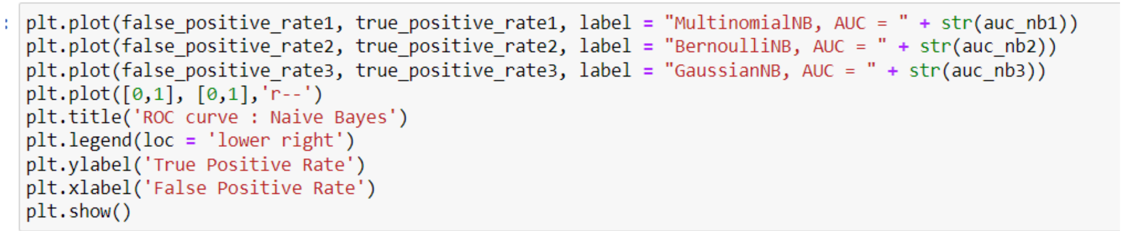
Figure 4.1.5 F1 score calculation



```
#calculate AUC

#MultinomialNB
false_positive_rate1, true_positive_rate1, thresholds = metrics.roc_curve(test.rating, y_pred_class_nb1)
print(metrics.auc(false_positive_rate1, true_positive_rate1))

auc_nb1 = metrics.roc_auc_score(test.rating, y_pred_class_nb1)
print(auc_nb1)


#GaussianNB
false_positive_rate2, true_positive_rate2, thresholds2 = metrics.roc_curve(test.rating, y_pred_class_nb2)
print(metrics.auc(false_positive_rate2, true_positive_rate2))

auc_nb2 = metrics.roc_auc_score(test.rating, y_pred_class_nb2)
print(auc_nb2)


#BernoulliNB
false_positive_rate3, true_positive_rate3, thresholds3 = metrics.roc_curve(test.rating, y_pred_class_nb3)
print(metrics.auc(false_positive_rate3, true_positive_rate3))

auc_nb3 = metrics.roc_auc_score(test.rating, y_pred_class_nb3)
print(auc_nb3)
```

```
0.8789473684210527
0.8789473684210527
0.6098471986417657
0.6098471986417657
0.735823429541596
0.735823429541596
```

Figure 4.1.6 Calculate AUC for Multinomial, Gaussian and Bernoulli Naive Bayes



```
plt.plot(false_positive_rate1, true_positive_rate1, label = "MultinomialNB, AUC = " + str(auc_nb1))
plt.plot(false_positive_rate2, true_positive_rate2, label = "BernoulliNB, AUC = " + str(auc_nb2))
plt.plot(false_positive_rate3, true_positive_rate3, label = "GaussianNB, AUC = " + str(auc_nb3))
plt.plot([0,1], [0,1],'r--')
plt.title('ROC curve : Naive Bayes')
plt.legend(loc = 'lower right')
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

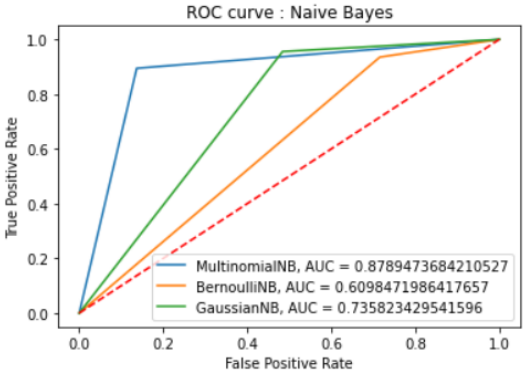


Figure 4.1.7 Display Result ROC for Multinomial, Gaussian and Bernoulli Naive Bayes

"Area under the ROC Curve" is the abbreviation for "Area under the ROC Curve." AUC, in other words, assesses the full two-dimensional area beneath the entire ROC curve (think integral calculus) from (0,0) to (1,0). (1,1).

A receiver operating characteristic curve (ROC curve) is a graph that shows how well a classification model performs across all categorization levels. Two parameters are shown on this curve is the true positive rate and the false positive rate. The true positive rate (TPR) vs. false positive rate (FPR) at various categorization criteria is plotted on a ROC curve. As the classification threshold is lowered, more items are classified as positive, resulting in an increase in both False Positives and True Positives. A typical ROC curve is seen in the diagram below.

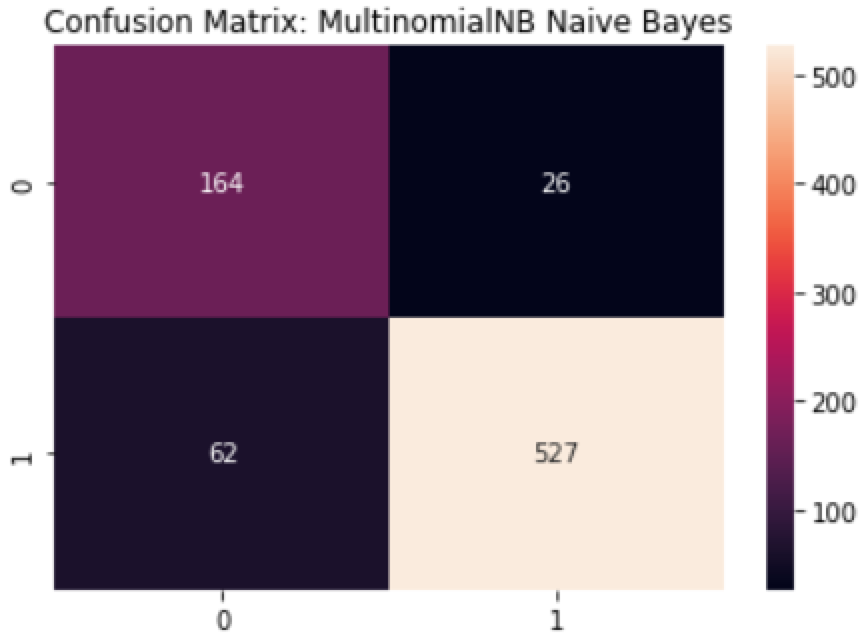


Figure 4.1.8 Confusion Matrix for Multinomial Naive Bayes

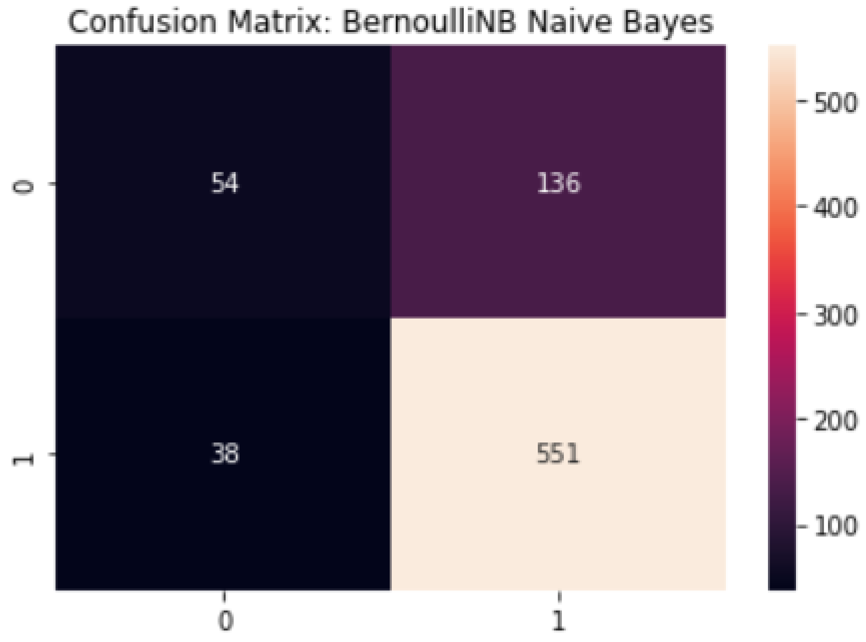


Figure 4.1.9 Confusion Matrix for Bernoulli Naive Bayes

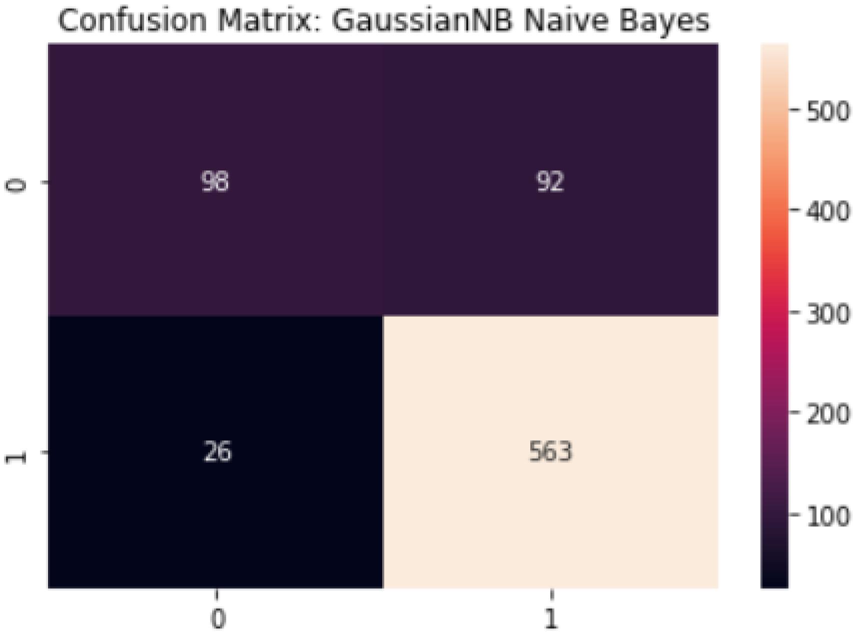
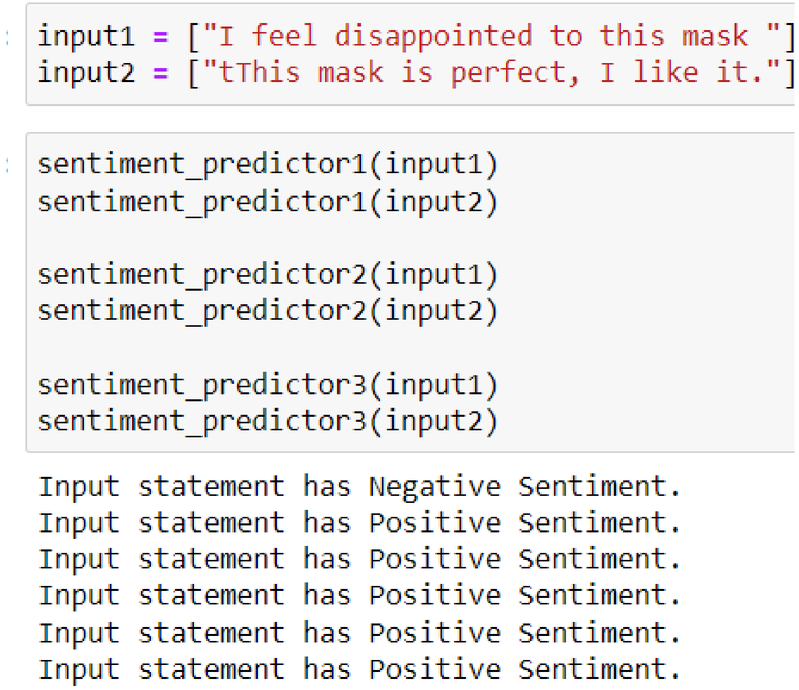


Figure 4.1.10 Confusion Matrix for Gaussian Naive Bayes

A confusion matrix is a table that shows how many accurate and erroneous guesses a classifier made. It may be used to calculate performance measures like accuracy, precision, recall, and F1-score to evaluate the performance of a classification model. In the figure below shown is the confusion matrix table for Multinomial, Bernoulli and Gaussian Naive Bayes. So in the confusion matrix, the row represents the predicted value for the AI predict, and the column represents the actual value for the negative or positive value.

4.2. Discussion/Interpretation

As the accuracy score shown in figure 4.1.1, we can see the accuracy score for multinomial naive bayes is 0.887 which is 88.7%. Besides that, the accuracy score of 0.777 which is 77.7% is using the gaussian naives bayes algorithm to calculate it out. Then for the bernoulli naive bayes accuracy score is 0.848 which is 84.8%. In figure 4.1.1, we can know that multinomial naive bayes have the highest accuracy score from these three algorithms.



```
input1 = ["I feel disappointed to this mask "]
input2 = ["tThis mask is perfect, I like it."]
```

```
sentiment_predictor1(input1)
sentiment_predictor1(input2)

sentiment_predictor2(input1)
sentiment_predictor2(input2)

sentiment_predictor3(input1)
sentiment_predictor3(input2)
```

```
Input statement has Negative Sentiment.
Input statement has Positive Sentiment.
Input statement has Positive Sentiment.
Input statement has Positive Sentiment.
Input statement has Positive Sentiment.
Input statement has Positive Sentiment.
```

Figure 4.2.1 Result for sentiment predictor

For the result shown in the figure 4.2.1, we can find out that only the multinomial naive bayes can predict the input string correctly which is the negative sentiment.

5. Discussion and Conclusion

5.1. Achievements

The achievements in this assignment is that sentiment analysis is successfully worked from reading the dataset, pre-processing data to train and test the data to let the AI determine the positive and negative sentences after using those algorithms. Because it can predict whether a review will be good or negative, the prototype has achieved the assignment's aim.

The suggested technique will categorise reviews into two categories which are positive and negative, as well as evaluate the success of the algorithm using various indicators. This would make it much easier for the user to determine the polarity of the review, saving the user time.

Machine learning will be utilised as the classification technique, as it is a commonly used methodology with several advantages over other approaches. Based on the dataset, this is used to forecast the polarity of feelings.

In conclusion, I found that the Multinomial Naive Bayes algorithm has a highest accuracy score depending on Bernoulli and Gaussian Naive Bayes.

5.2. Limitations and Future Works

The limitations of the project are that the amount of data in the dataset that we crawl on the amazon website is not enough to do the analysis for the sentimental analysis. Besides that, the algorithm to do the prediction is not that much in this project. For future works, I will try out more types of algorithms out of supervised learning and also learn about the unsupervised learning algorithm.

Reference & Source

1. Vinita Sharma, 2014, SentimentAnalysis. Available at: <https://www.cfilt.iitb.ac.in/resources/surveys/SentimentAnalysis-Vinita.pdf>
2. Hanan Alasmari, 2020, Sentimental Visualization: Semantic Analysis of Online Product Reviews Using Python and Tableau. Available at: <https://ieeexplore-ieee-org.tarcez.tarc.edu.my/document/9391769>
3. Charu Nanda; Mohit Dua; Garima Nanda, 2018, Sentiment Analysis of Movie Reviews in Hindi Language Using Machine Learning. Available at: <https://ieeexplore-ieee-org.tarcez.tarc.edu.my/document/8524223>
4. Md. Rakibul Haque; Salma Akter Lima; Sadia Zaman Mishu, 2019, Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews. Available at: <https://ieeexplore-ieee-org.tarcez.tarc.edu.my/document/9303573>
5. Sentiment Analysis: Types, Tools, and Use Cases. Available at: <https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/>
6. Kristian Bannister, 2018, Understanding Sentiment Analysis: What It Is & Why It's Used. Available at: <https://www.brandwatch.com/blog/understanding-sentiment-analysis/>
7. The Team at CallMiner, 2019, What is Sentiment Analysis? Examples, Best Practices, & More. Available at: <https://callminer.com/blog/sentiment-analysis-examples-best-practices>
8. Hardikkumar, 2021, Performing Sentiment Analysis With Naive Bayes Classifier! Available at: <https://www.analyticsvidhya.com/blog/2021/07/performing-sentiment-analysis-with-naive-bayes-classifier/>
9. Manish Sharma, 2020, Sentiment Analysis: An Introduction to Naive Bayes Algorithm. Available at: <https://towardsdatascience.com/sentiment-analysis-introduction-to-naive-bayes-algorithm-96831d77ac91>
10. Apoorva Aryaa, Vishal Shuklab, Arvind Negic, Kapil Guptad, A Review: Sentiment Analysis and Opinion Mining. Available at: <https://deliverypdf.ssrn.com/delivery.php?ID=607074089082021070022114094006065100019000031052064056118109013085029077069119126076045022127062105116060077102120003077080077105018000043039098123108098029112007123003044033106024078106023085029110019112098107118120009107024088071022015125073028096001&EXT=pdf&INDEX=TRUE>

11. Great Learning Team, 2020, Multinomial Naive Bayes Explained. Available at: <https://www.mygreatlearning.com/blog/multinomial-naive-bayes-explained/>
12. Classification: ROC Curve and AUC. Available at: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>