

---

# SMS SPAM CLASSIFIER

---

## Project Report

SUPERVISOR: Mr. Mahesh Kumar

### SUBMITTED BY:

KUSUM: 22/25002

ADITYA KUMAR: 22/25012

DIVYANSHI: 22/25021

TUSHAR RANA: 22/25064



(2024-2025)

Department of Computer Science

ATMA RAM SANATAN DHARMA COLLEGE

(University of Delhi)

# ACKNOWLEDGEMENT

This Project was jointly undertaken by Kusum, Aditya Kumar, Divyanshi, and Tushar Rana as their Sem V Data Mining for Knowledge Discovery Project, with the guidance and supervision of Mr. Mahesh Kumar. Our primary thanks go to him, who poured over every inch of the project with thorough attention and helped us throughout the working of the project. It is our privilege to acknowledge our deepest gratitude to him for his inspiration which helped us immensely. We are extremely grateful to him for his unstilted support and encouragement in the preparation of this project.

Kusum  
(22/25002)

Aditya Kumar  
(22/25012)

Divyanshi  
(22/25021)

Tushar Rana  
(22/25064)

**ATMA RAM SANATAN DHARMA**  
**(University Of Delhi)**

**CERTIFICATE**

This is to certify that the project entitled “**SMS Spam Classifier**” has been successfully completed by **Kusum, Aditya Kumar, Divyanshi and Tushar Rana** of B.Sc. Physical Sciences with C.S. during Semester-V from Atma Ram Sanatan Dharma College, University of Delhi under the supervision of **Mr. Mahesh Kumar**.

**Mr. Mahesh Kumar**  
(Dept. of Computer Science)  
Atma Ram Sanatan Dharma College  
University Of Delhi

# CONTENTS

PROBLEM STATEMENT .....	6
DATA MINING TECHNIQUES .....	7
2.1. Data Mining Techniques.....	7
2.1.1. Classification .....	8
2.1.2. Association .....	8
2.1.3. Clustering .....	8
2.1.4. Regression .....	8
2.2 Classification .....	8
2.2.1. K-Nearest Neighbour.....	8
2.2.2. Naive Bayes .....	9
2.2.3. Decision Tree .....	9
2.3. Why Classification? .....	9
DATASET DESCRIPTION .....	10
3.1 Number of Records .....	10
3.2 Number of Attributes .....	10
3.3 Types of Attributes .....	10
3.4 Missing Values or Nulls .....	10
3.5 Attribute Description.....	11
3.6 Distribution/Histograms.....	11
3.7 Detecting Outliers .....	13
DATA PREPROCESSING .....	15
4.1 Handling Null Values .....	15
4.2 Handling Duplicate Values.....	16
4.3 Feature Extraction.....	16
4.4 Feature Scaling .....	17
4.4.1. Normalization.....	17
4.4.2. Standardization .....	18
4.5 Conversion .....	18
4.6 Data Sampling and Subsetting .....	19
4.6.1. Train-Test Split .....	19
4.4.2 Types of splitting .....	20
BUILDING MODELS.....	21
5.1 Model 1: KNN .....	21

5.2 Model 2: Naive Bayes.....	25
5.3 Model 3: Decision Tree .....	29
MODEL EVALUATION AND RESULTS .....	34
6.1 METRICS .....	34
6.1.1 ACCURACY .....	34
6.1.2 PRECISION .....	34
6.1.3 RECALL.....	34
6.1.4 F1-SCORE.....	35
6.2. CONFUSION MATRIX .....	35
6.3. Experimental Results and Comparison .....	37
INFERENCES AND CONCLUSION.....	38
REFERENCES.....	39

## **Chapter – 1**

### **PROBLEM STATEMENT**

In today's digital age, mobile communication has become an essential part of everyday life. However, the prevalence of unsolicited and potentially harmful spam messages poses a significant challenge to users' privacy and security. Spam messages can range from annoying advertisements to malicious links that can compromise personal information and data security. Therefore, it is crucial to develop an effective solution to automatically identify and filter out spam messages from legitimate ones.

The objective of this project is to build an SMS Spam Classifier that can accurately distinguish between spam and non-spam (ham) messages. The classifier will leverage machine learning techniques to analyse the content of SMS messages and classify them accordingly. The solution aims to minimize false positives (ham messages incorrectly classified as spam) and false negatives (spam messages incorrectly classified as ham) to ensure that users receive only relevant and safe communication.

## Chapter - 2

### DATA MINING TECHNIQUES

Data mining is the process of extracting useful information from large data repositories. The results of the extraction process are useful hidden patterns and information from large data repositories for categorizing it into useful data. Data mining techniques are used in almost every area that deals with large data. The finding of the process can be used by various organizations to predict their targeted customers, for marketing, helping in decision making, and other data requirements to eventually cost-cutting and generating revenue.

#### 2.1. Data Mining Techniques

Data mining techniques use different mathematical algorithms, machine learning techniques, and models for analysis and prediction. Data mining techniques are further classified into supervised learning and unsupervised learning.

**Supervised Learning:** Supervised learning has the presence of a supervisor as a teacher. Basically, supervised learning is when we teach or train the machine using data that labelled. Which means some data is already tagged with the correct is well answer. After that, the machine is provided with a new set of data so that the supervised learning algorithm analyses the training data and produces a correct outcome from labelled data.

**Unsupervised Learning:** Unsupervised learning is the training of a machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data. Unlike supervised learning, no teacher is provided which means no training will be given to the machine. Therefore, the machine is restricted to finding the hidden structure in unlabelled data by itself.

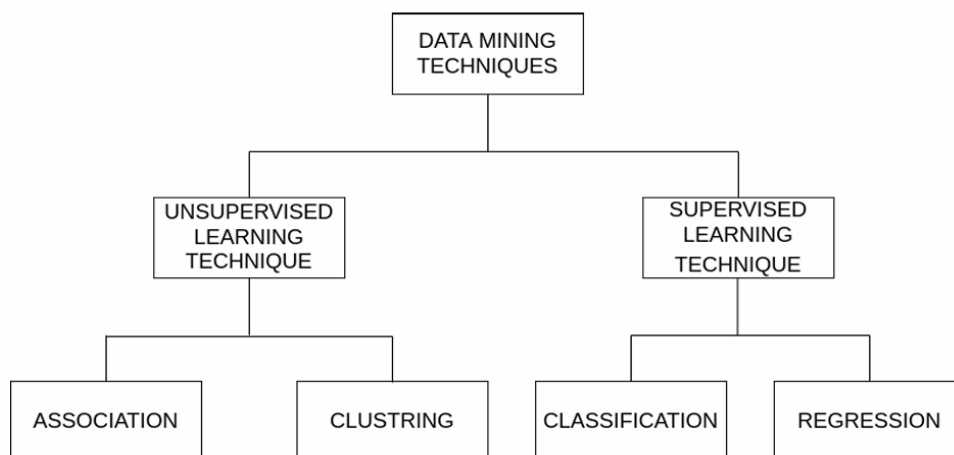


Figure2.1- Data Mining Techniques

Types of data mining techniques are as follows:

### 2.1.1. Classification

Classification as the name suggests is a data mining technique that helps to classify data in different classes. It analyses the different attributes which are associated with different data types and once it identifies the main purpose of these data types, it categorizes the data.

### 2.1.2. Association

Association is a statistics-related data mining technique. found in data are linked to other It indicates that certain data or events data or data-driven events. Association rules are if-then statements that support showing the probability of interactions between data items within large data sets in different types of databases.

### 2.1.3. Clustering

Clustering analysis is a data mining technique to identify similar data. differences and similarities between the data. It helps to identify the Graph approaches are ideal for using cluster analytics. It involves grouping chunks of data together based on their similarities.

### 2.1.4. Regression

Regression refers to a data mining technique that is used to predict the numeric values in each data set. For example, regression might be used to predict the product or service cost or other variables. It is also used in various industries for business and marketing behaviour, trend analysis, and financial forecast.

## 2.2 Classification

Classification is the technique that assigns the object one of the predefined classes. It is used to categorize data into different categories. Classification is the task of learning a target function  $f$  that maps each attribute set to one of the predefined class labels. The target function is also known as informally as a classification.

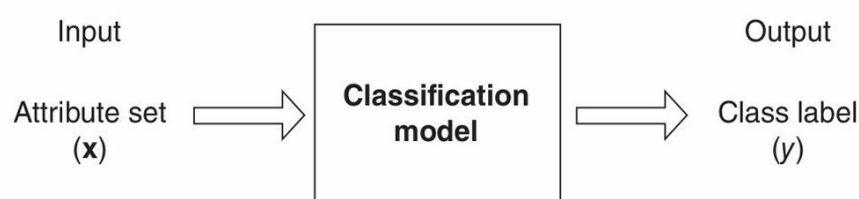


Figure 2.2- A schematic illustration of a classification task.

Some well-known classification methods are Decision Tree, Nearest-Neighbour, Naive Bayes Classifier, etc.

### 2.2.1. K-Nearest Neighbour

K-Nearest Neighbour commonly known as KNN classifier stores all the available data and puts the new cases based on similarity with attributes of available data. The nearest neighbour classifier represents each example as a data point in a  $d$ -dimensional space, where  $d$  is the number of attributes. The  $k$ -nearest neighbours of a given test instance  $z$  refer to the training



examples that are closest to  $z$ . It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

### **2.2.2. Naive Bayes**

Naive Bayes is a probabilistic classification model since it uses probability theory to represent the relationship between attributes and class labels. It is based on Bayes Theorem, “the Bayes theorem is a mathematical formula used to determine the conditional probability of events. Essentially, the Bayes theorem describes the probability of an event based on prior knowledge of the conditions that might be relevant to the event”. The formula of Bayes Theorem:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}.$$

It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Hence each feature individually contributes to identifying that it is an apple without depending on each other. And it is called Bayes because it depends on Bayes theorem.

### **2.2.3. Decision Tree**

A decision tree is a structure that includes a root node, branches, and leaf nodes. A class label is associated with every leaf node of the decision tree. The non-terminal nodes, which include the root and internal nodes, contain attribute test conditions. Starting from the root node, we apply the attribute test condition and follow the appropriate branch based on the outcome of the test. This will lead us either to another internal node, for which a new attribute test condition is applied, or to a leaf node. Once a leaf node is reached, we assign the class label associated with the node to the test instance.

## **2.3. Why Classification?**

Classification is crucial in spam detection as it helps identify and filter out spam emails efficiently, which would be impossible to do manually. By using machine learning algorithms, these classifiers can improve their accuracy over time by learning from past data, reducing the number of false positives and negatives. This not only enhances the user experience by keeping their inboxes clean but also increases security by preventing phishing attempts, scams, and malware from reaching users. Ultimately, a good spam detection system saves time and protects users from potential threats.

## Chapter - 3

### DATASET DESCRIPTION

This section includes a detailed description of data from the number of instances to the missing values, outliers, types of data attributes, and many more. This description helps us to understand our data more specifically so that probable mining techniques/models can be applied like in this case the dataset is of categorical type so, here by Classification models like Decision Tree, naive Bayes and K -Nearest Neighbour are used. For this study the data has been collected from Kaggle.

#### 3.1 Number of Records

The raw dataset consists of 5572 (five thousand five hundred seventy-two) records, for the study and prediction whole data is taken into action.

#### 3.2 Number of Attributes

There are in total 5 (five) attributes named as:

1. V1
2. V2
3. Unnamed :2
4. Unnamed: 3
5. Unnamed: 4

#### 3.3 Types of Attributes

There are many types of data such as Nominal, binary, Ordinal, Interval, and the ratio given below is the description of each attribute with its type:

1. V1 - Nominal
2. V2 - Nominal
3. Unnamed :2 - Nominal
4. Unnamed: 3 - Nominal
5. Unnamed: 4 – Nominal

#### 3.4 Missing Values or Nulls

We have missing values in Unnamed: 2, Unnamed: 3, Unnamed: 4. All of them will be dropped because they contain more than 50% missing values and rest will be filled during preprocessing.

### 3.5 Attribute Description

1. v1                      object
2. v2                      object
3. Unnamed: 2      object
4. Unnamed: 3      object
5. Unnamed: 4      object

### 3.6 Distribution/Histograms

A histogram is a graph showing frequency distributions.

It is a graph showing the number of observations within each given interval.

Here, column distribution contains x-axis that shows the unique data values of each attribute and y-axis shows the frequency or count in each attribute

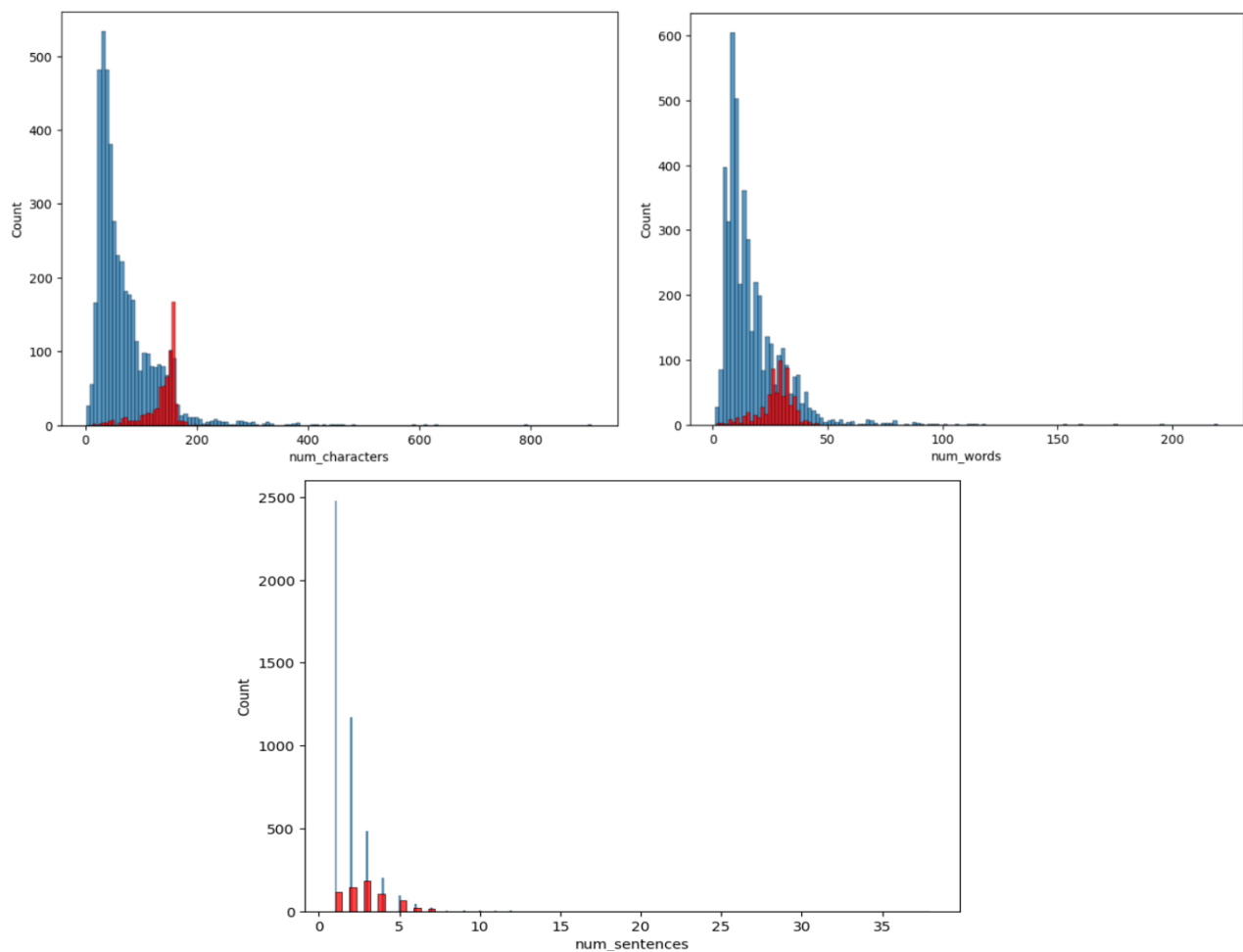


Figure 3.1- Column Distribution Graph

Correlation is used to represent the statistical measure of the linear relationship between two variables. It can also be said to measure the dependence between two different variables. It is used for data distribution and data understanding of multiple variables and the goal is to find a correlation between all these variables and store them using the appropriate data structure, the matrix data structure is used. Such a matrix is called a correlation matrix.

The value of the correlation coefficient can take any values from -1 to 1.

- If the value is 1, it is said to be a positive correlation between the two variables. This means that when one variable increases, the other variable also increases.
- If the value is -1, it is said to be a negative correlation between the two variables. This means that when one variable increases, the other variable decreases.
- If the value is 0, there is no correlation between the two variables. This means that the variables change in a random manner with respect to each other

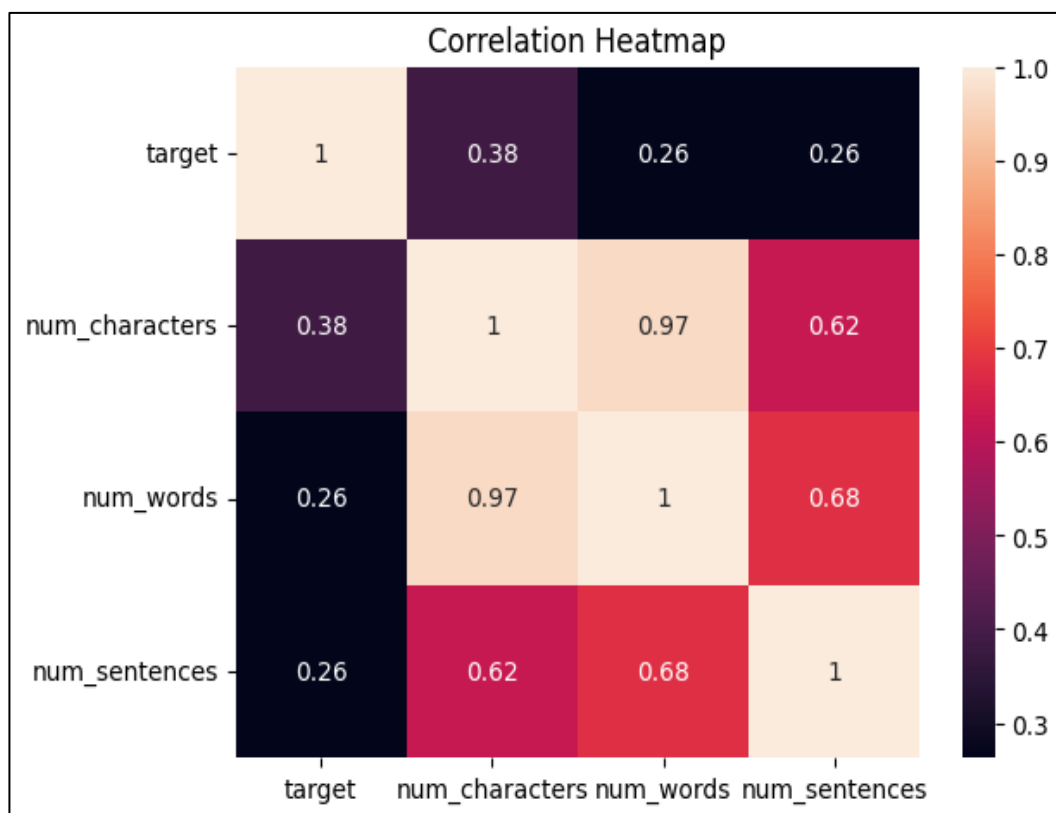


Figure 3.2- Correlation Heatmap

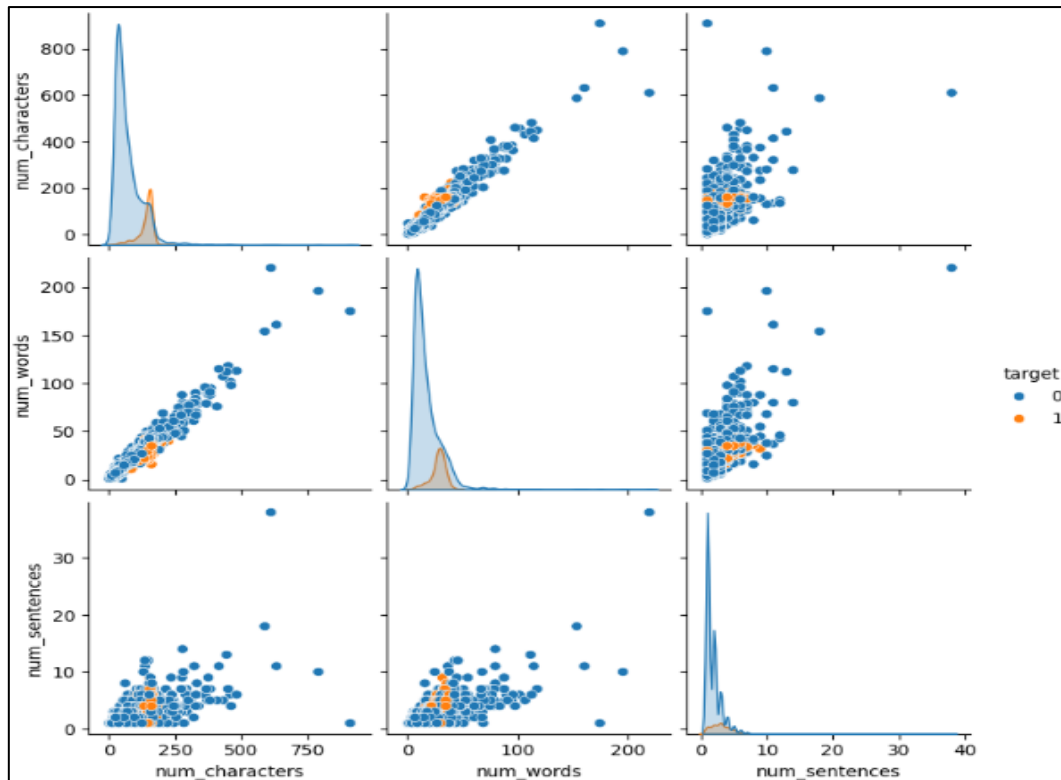


Figure 3.3- Correlation represented using the Pairplot

### 3.7 Detecting Outliers

An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution errors. Major outliers are the noise that interferes with data analysis. To detect the outliers, we can simply check out using the box plot or scatter plot and then can remove or modify them with some condition. The analysis for outlier detection is referred to as outlier mining. There are many ways to detect the outliers, and the removal process is the data frame same as removing a data item from the panda's data frame or by replacing it with mean, median or mode according to your choice and data frame

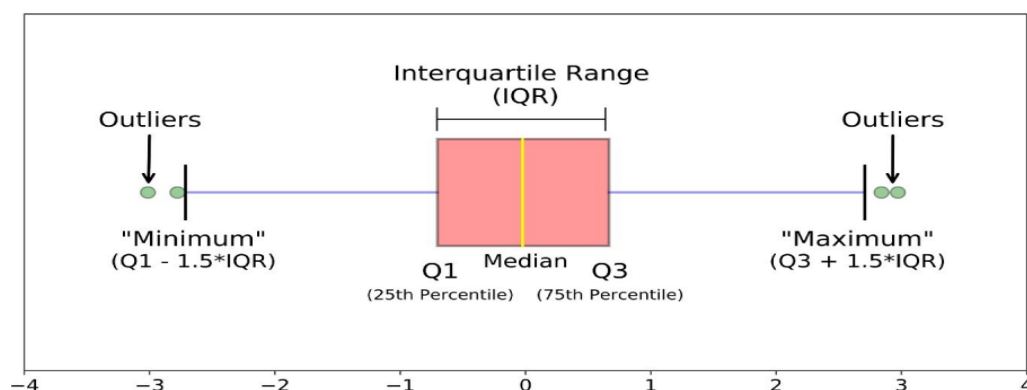


Figure 3.4- Description of Box Plot

Source: Box Plot

To detect the outliers Box plot is the most common way of representation. Boxplots are a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”).

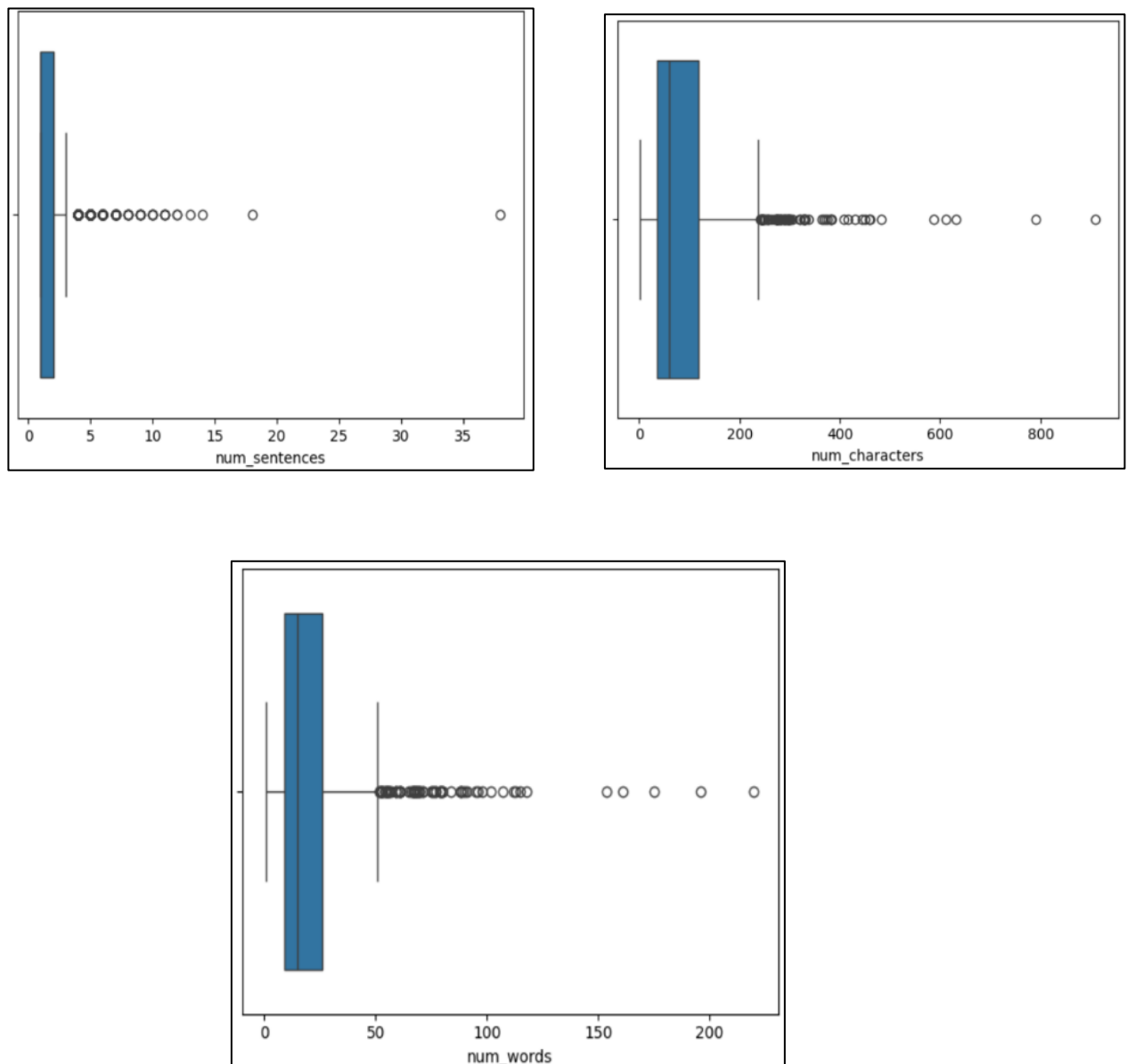


Figure 3.5- Shows the outliers from the number of characters, words and sentences attribute respectively (The black deep dots are represented as outliers).

How will outliers ' presence will impact the project?

Outliers can sometimes skew the results of your model, leading to poor performance. In such cases, removing them might improve accuracy. Sometimes outliers represent new types of spam or unusual legitimate messages. Removing them could mean losing valuable information. So, we have decided not to remove these outliers from our dataset.

## Chapter – 4

### DATA PREPROCESSING

Data preprocessing is a data mining process of transforming the raw data into an understandable format. The quality of data must be checked before applying data mining techniques. Data preprocessing involves three major steps data cleaning, data transformation, data reduction.

#### 4.1 Handling Null Values

In the raw data, there can be some values missing which may give trouble if not handled. So, to obviate this issue we handle these missing values using various ways:

**Ignore the tuple:** This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

**Fill the missing values:** There are various ways to do this task. we can choose to fill the missing values by some default values, by attribute mean, or the most probable value.

In this dataset all the null values are created explicitly and then replaced by the mode of that column.

Checking the number of null values for each column

```
print(df.isnull().sum())  
v1      0  
v2      0  
Unnamed: 2    5522  
Unnamed: 3    5560  
Unnamed: 4    5566  
dtype: int64
```

Unnamed: 2, Unnamed: 3, Unnamed: 4 have more than 50% values so we will drop them

```
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)
```

Final count of null values after handling them

```
print(df.isnull().sum())  
v1      0  
v2      0  
dtype: int64
```

## 4.2 Handling Duplicate Values

Handling duplicate values in your data is an important step in data cleaning and preprocessing. Duplicate values can skew your analysis and lead to inaccurate model predictions.

Checking the duplicate values in the data

```
df.duplicated().sum()
np.int64(403)
```

Removing the duplicate values from the data

```
df.drop_duplicates(inplace=True)
df.duplicated().sum()
np.int64(0)
df.shape
(5169, 2)
```

## 4.3 Feature Extraction

It's a process used in data preprocessing and involves creating new features (or columns) from the existing data. Feature extraction transforms raw data into informative features. This can enhance the predictive power of machine learning models by providing additional relevant information.

Creating a new column “num\_characters” which consist of the number of characters of the text SMSs.

```
df['num_characters'] = df['text'].apply(len) #creating a new coulmn with the number of characters of the text sms for each row
```

Creating another column “num\_words” which consist of the number of the words used in each text SMSs.

```
df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x))) #creating a new coulmn with the number of words in the text sms
```



Creating another column “num\_sentence” which consist of the number of sentences used in each text SMSs.

```
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x))) # creating a column by counting the number of sentences in the text sms
df.head()
```

Creating a column “transformed\_text” by applying a function on Text column which convert the text into **Lowercase**, helps in **Tokenization**, Removing special characters, removing stop words and punctuation, **Stemming**.

Function used for transforming the text:

```
# creating a function for converting text into lower case, Tokenization,Removing special characters,Removing stop words and punctuation,Stemming
def transform_text(text):
    text = text.lower() # convert all the sms into lowercase
    text = nltk.word_tokenize(text) # breaking down of text into smaller units called tokens.

    y = [] #taking an empty list
    for i in text: #applying for loop on the text to remove special chacters
        if i.isalnum():
            y.append(i) #it will append all the tokenized text which are only alphanumeric

    text = y[:] #copy the y to the variable text
    y.clear() # after copying all the data of y into text variable clearing the y

    for i in text: # again applying another for loop for removing stop words(a,an,is,or,if) and punctuation
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    text = y[:] #copy the y to the variable text
    y.clear() # after copying all the data of y into text variable clearing the y

    for i in text: #again using another for loop on text for removing stemming(reduce words to their root or base form)
        y.append(ps.stem(i)) # reduce words to their root or base form and append them in y

    return " ".join(y) # returnig y as a string
```

Using this function and creating a new column “transformed\_text”

```
df['transformed_text'] = df['text'].apply(transform_text)
df.head()
```

## 4.4 Feature Scaling

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

### 4.4.1. Normalization

It is a technique to scale the data values with a distribution range of 0-1

$$X_{\text{new}} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

Here,  $X_i$  = Original Value

$\min(X)$  = Minimum of Column X

$\max(X)$  = Maximum of Column X

#### 4.4.2. Standardization

It is a technique that re-scales a feature value so that it has a distribution with 0 mean value and variance equals 1.

$$X_{\text{new}} = \frac{X_i - X_{\text{mean}}}{\text{Standard Deviation}}$$

Here,  $X_i$  = Original value

$X_{\text{mean}}$  = Mean of Column X

Standard Deviation = Standard Deviation of column X

#### 4.5 Conversion

Conversion is a common method to convert categorical data to numerical. Since the machine is always good at dealing with numerical data therefore it is important to convert categorical data to numeric form. In this given dataset Target is converted from categorical to numerical.

The data set before converting categorical data into numerical form:

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

The data is converted in the following manner:

**Target:** 0 refers to ham and 1 refers to spam

Code to convert categorical data into numerical form:

```
from sklearn.preprocessing import LabelEncoder #LabelEncoder converts categorical labels into numerical values
encoder = LabelEncoder()

df['target'] = encoder.fit_transform(df['target']) # converting target column into numerical and fitting it into 0 and 1.
```

The data set after converting categorical data into numerical form:

```
df.head() # 0 - ham and 1 - spam
```

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

## 4.6 Data Sampling and Subsetting

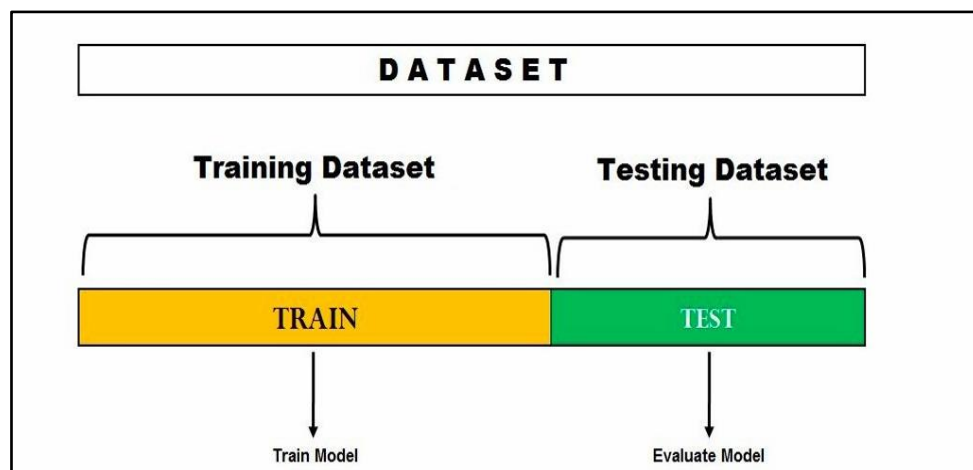
Sampling is a commonly used approach for selecting a subset of the data objects to be analysed. Data Sampling is used because it is too computationally expensive in terms of the memory or time required to process all the data.

### 4.6.1. Train-Test Split

The train-test split is a technique for evaluating the performance of a Data Mining algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

- Train Dataset: Used to fit the model.
- Test Dataset: Used to evaluate the fit model



The objective is to estimate the performance of the model on new data Which is not used to train the model.

The procedure has one main configuration parameter, which is the size of the train and test sets. This is most expressed as a percentage between 0 and 1 for either the train or test datasets. For example, a training set with the size of 0.67 (67 percent) means that the remaining percentage of 0.33 (33 percent) is assigned to the test set.

Nevertheless, common split percentages include:

Train: 80%, Test: 20%

Train: 75%, Test: 25%

Train: 67%, Test: 33%

#### **4.4.2 Types of splitting**

Three types of splitting have been used in concern to the heart disease prediction.

##### **1. Hold-Out Method**

The holdout method is the simplest sort of method to evaluate a classifier. In this method, the data set is separated into two sets, called the Training set and Test set. A classifier performs the function of assigning data items in each collection to a target category or class. In such a method data set is partitioned, such that – maximum data belongs to the training set and the remaining data belongs to the test set.

##### **2. Random Subsampling**

Random sub-sampling, which is also known as Monte Carlo cross-validation, as multiple holdouts or as repeated evaluation set, is based on randomly splitting the data into subsets, whereby the size of the subsets is defined by the user. The random partitioning of the data can be repeated arbitrarily often. In contrast to a full cross-validation procedure, random subsampling has been shown to be asymptotically consistent resulting in more pessimistic predictions of the test data compared with cross-validation. The predictions of the test data give a realistic estimation of the predictions of external validation data.

##### **3. Cross-Validation**

The most popular subsampling technique is cross-validation. For an n-fold cross-validation, the data are partitioned into n equal parts. The first part is used as a test data set; the rest is used as the calibration data set. Then, the second part is used for the test data and the rest is used for a new calibration. This procedure is repeated n times and the predictions of the n test data are averaged. It is essential that no knowledge of the models is transferred from the fold to the fold. There exist no clear rules on how many folds to use for the cross-validation, whereby the simplest and clearest way of performing cross-validation is to leave one sample out at a time. This special variant of cross-validation is also called full cross-validation, leave-one-out, or jack knifing and gives a unique and therefore reproducible result. Yet, it has been shown that increasing the number of cross validation groups results in lower root mean square errors of predictions giving overly optimistic estimations of predictivity.

## Chapter – 5

### BUILDING MODELS

There are many possible classification models with varying levels of model complexity that can be used to capture patterns in the training data. Among these possibilities, we want to select the model that shows the lowest generalization error rate.

#### 5.1 Model 1: KNN

K-Nearest Neighbour commonly known as the KNN classifier stores all the available data and puts the new cases based on similarity with attributes of available data. [3] The nearest neighbour classifier represents each example as a data point in a d-dimensional space, where d is the number of attributes. For computing the distance of each record the Euclidean distance has been found,

$$\begin{aligned}d(\mathbf{p}, \mathbf{q}) &= d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.\end{aligned}$$

Example: Suppose we have an image of a creature that looks like a cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dog's images and based on the most similar features it will put it in either cat or dog category.

## SPLIT TYPE: HOLD-OUT METHOD

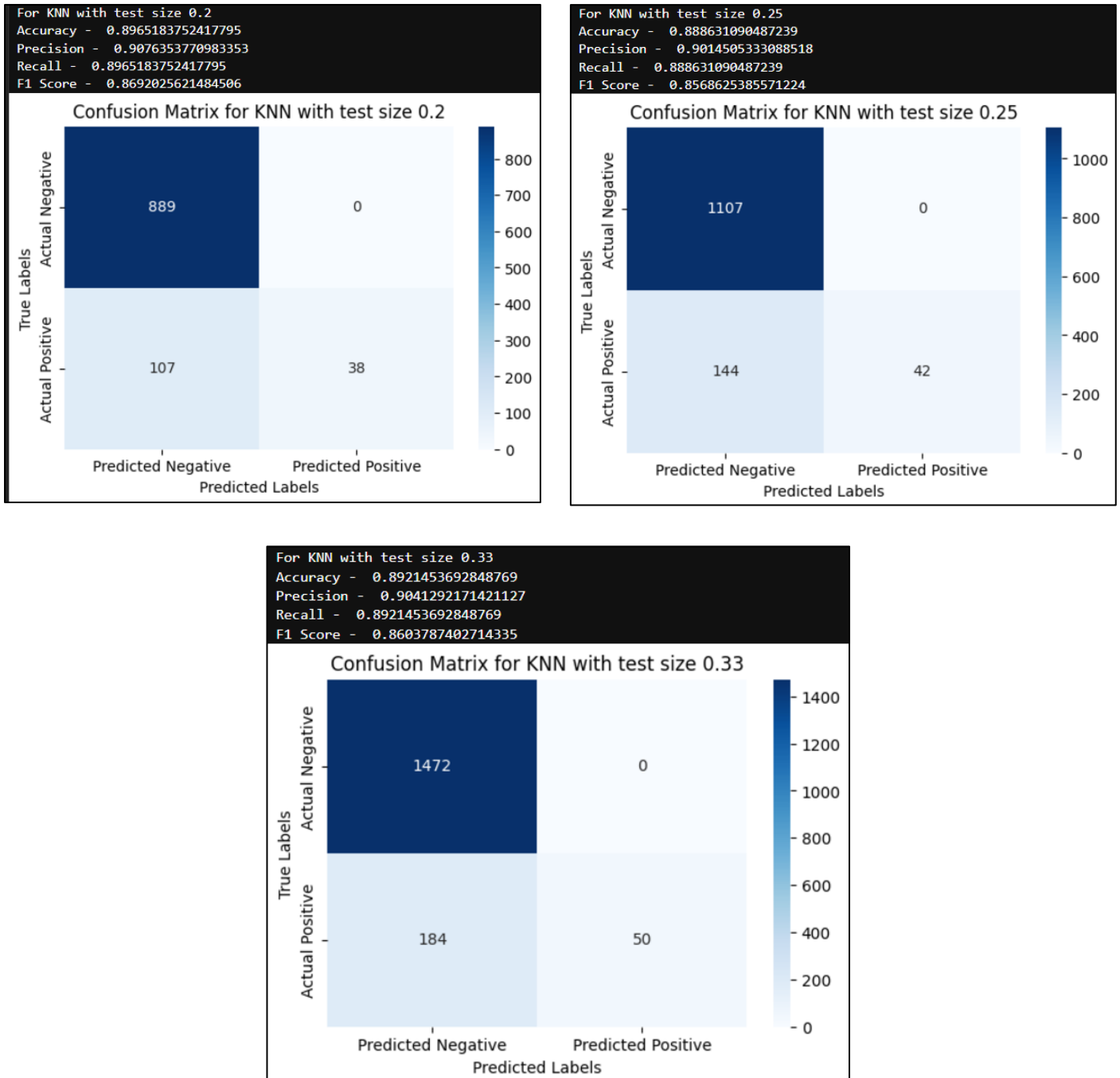
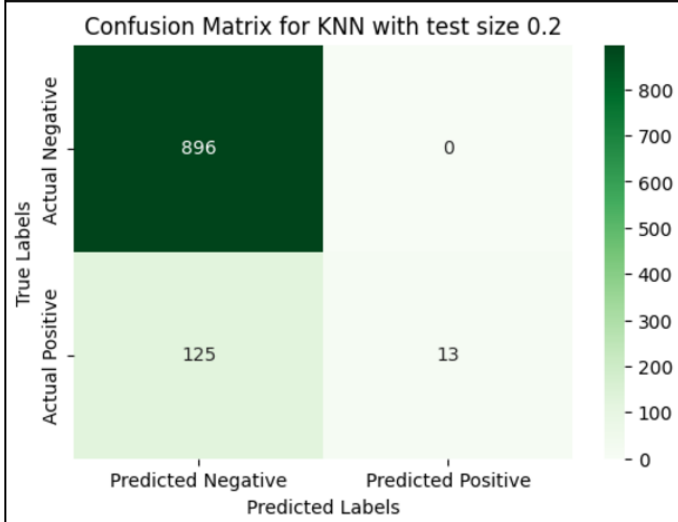


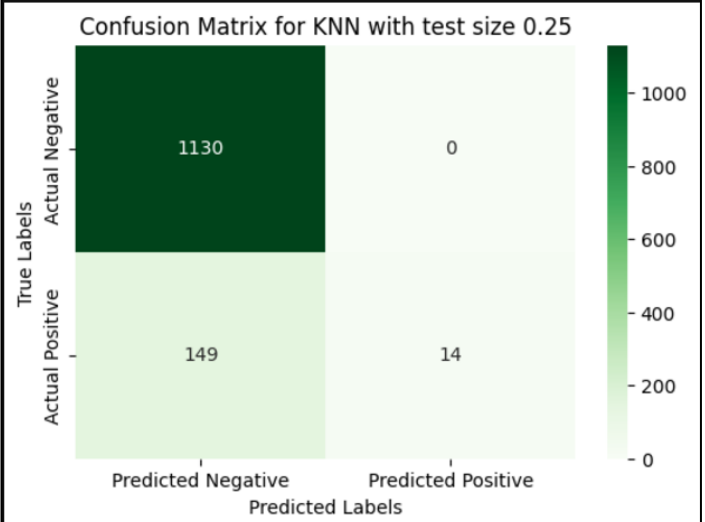
Figure 5.1 -Confusion Matrix for split type hold out and with three split ratios as 0.20,0.25 and 0.33 respectively using the K-Nearest Neighbour classifier.

## SPLIT METHOD – RANDOM SUBSAMPLING

For KNN with test size 0.2  
 Accuracy - 0.879110251450677  
 Precision - 0.8939106614101926  
 Recall - 0.879110251450677  
 F1 Score - 0.8330144754343037



For KNN with test size 0.25  
 Accuracy - 0.8847641144624904  
 Precision - 0.898188779783123  
 Recall - 0.8847641144624904  
 F1 Score - 0.8398246243894374



For KNN with test size 0.33  
 Accuracy - 0.8851113716295428  
 Precision - 0.8984515204213731  
 Recall - 0.8851113716295428  
 F1 Score - 0.8401214232849069

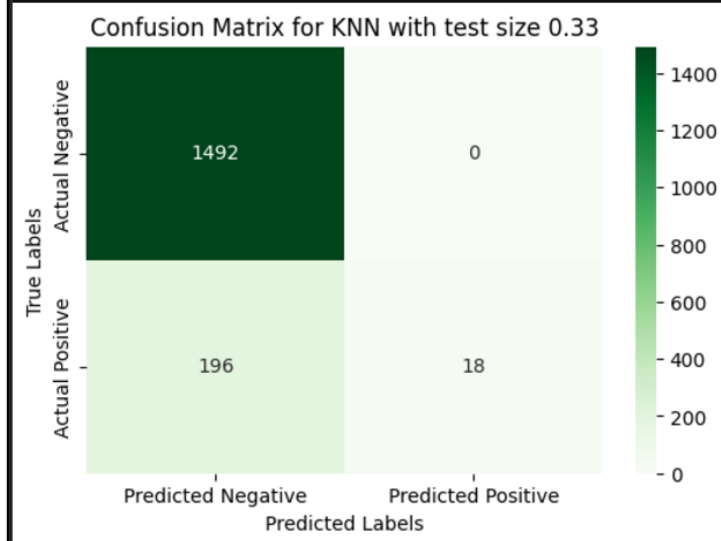


Figure 5.2 - Confusion Matrix for split type Random Subsampling and with three spilled ratios as 0.20,0.25 and 0.33 respectively using the K-Nearest Neighbour classifier.

## SPLIT METHOD - CROSS-VELIDAITON

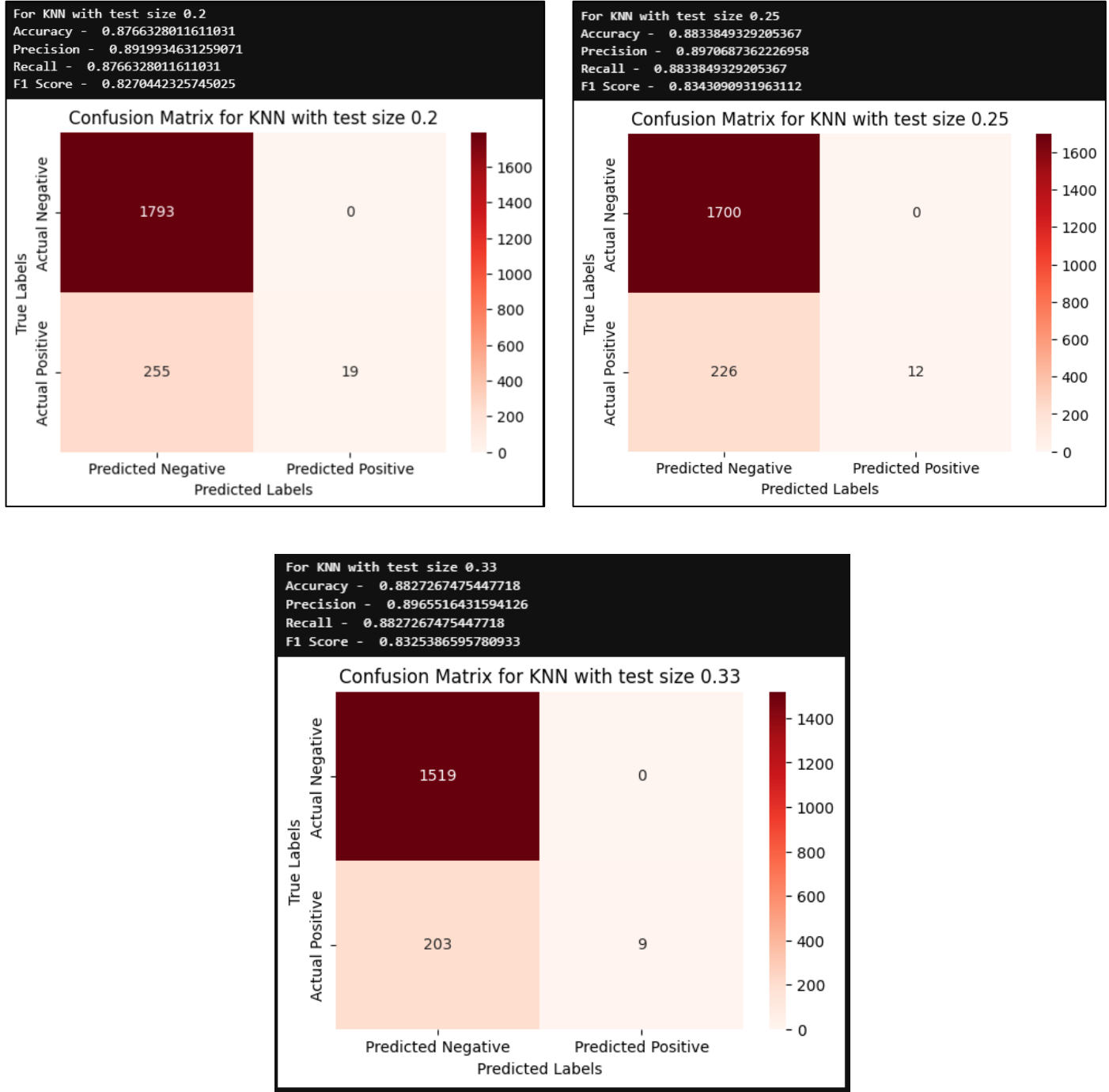


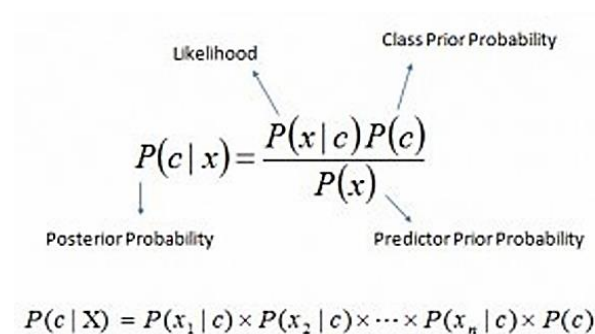
Figure 5.3: Confusion Matrix for split type Cross Validation and with three spilled ratios as 0.20,0.25 and 0.33 respectively using the K-Nearest Neighbour classifier.



## 5.2 Model 2: Naive Bayes

Naive Bayes is a probabilistic classification model since it uses probability theory to represent the relationship between attributes and class labels. It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features based on the Bayes Theorem.

Bayes theorem provides a way of calculating posterior probability  $P(c|x)$  from  $P(c)$ ,  $P(x|c)$ . Look at the equation below:



The diagram shows the equation  $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$  with four labels and arrows: 'Likelihood' points to  $P(x|c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c|x)$ , and 'Predictor Prior Probability' points to  $P(x)$ .

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Above,

- $P(c|x)$  is the posterior probability of class (c, target) given predictor (x, attributes).
- $P(c)$  is the prior probability of class.
- $P(x|c)$  is the likelihood which is the probability of the predictor given class.
- $P(x)$  is the prior probability of the predictor

## SPLIT METHOD : HOLD OUT

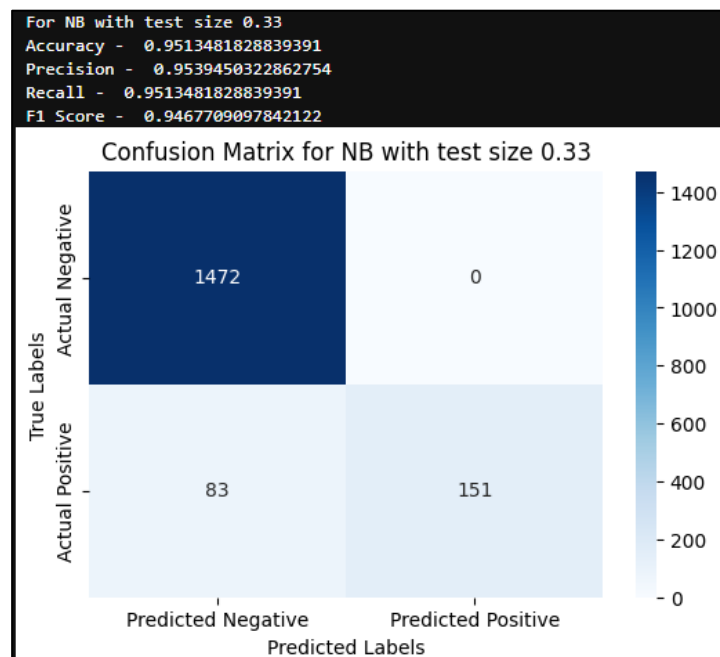
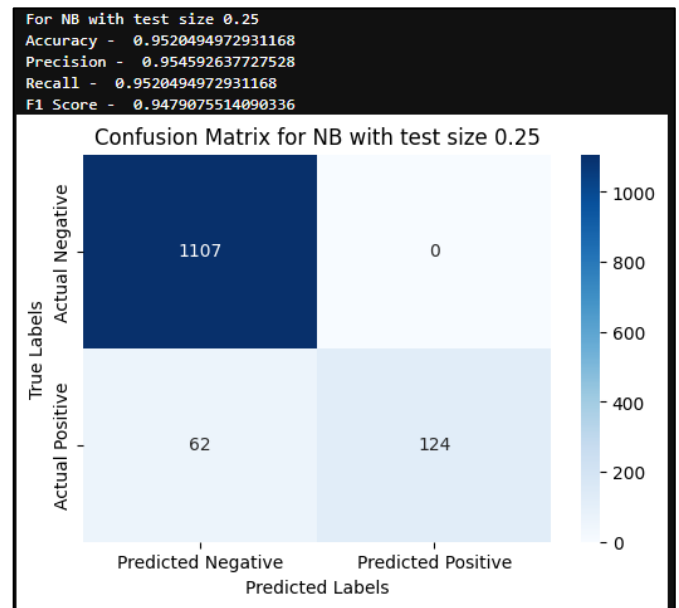
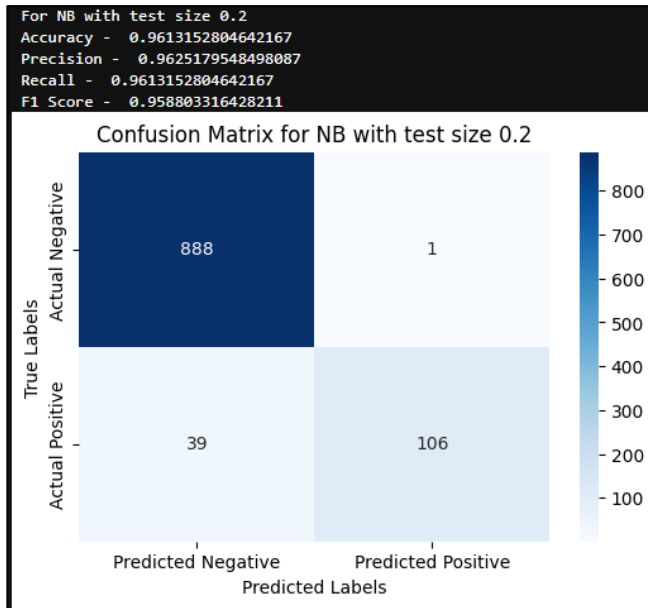


Figure 5.4: Confusion Matrix for split type hold out and with three spilled ratios as 0.20,0.25 and 0.33 respectively using the navies Bayes classifier.

## SPLIT METHOD – RANDOM SUBSAMPLING

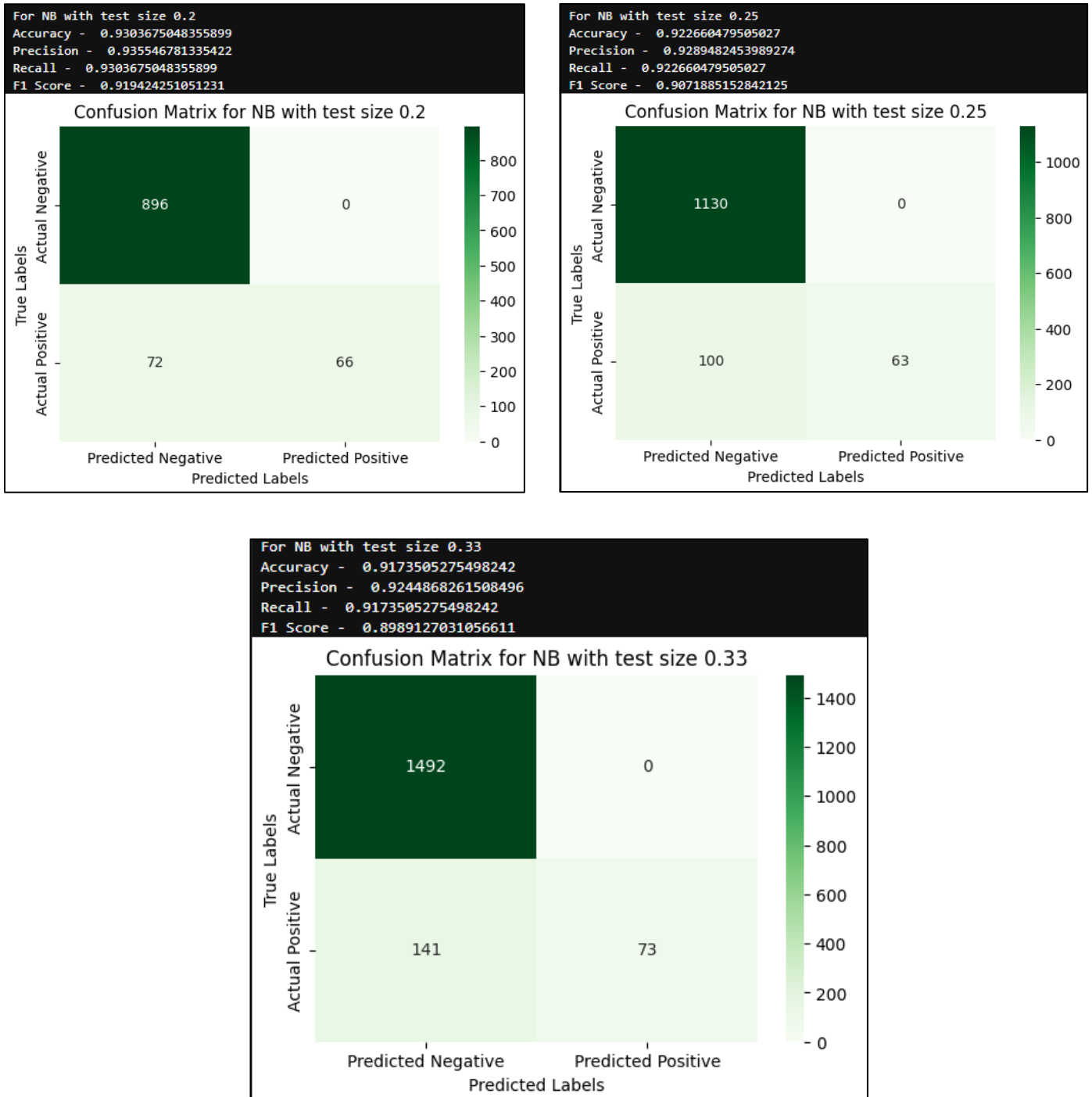


Figure 5.5 : Confusion Matrix for split type Random Subsampling and with three spilled ratios as 0.20,0.25 and 0.33 respectively using the navies Bayes classifier.

## SPLIT METHOD – CROSS VELIDAITON

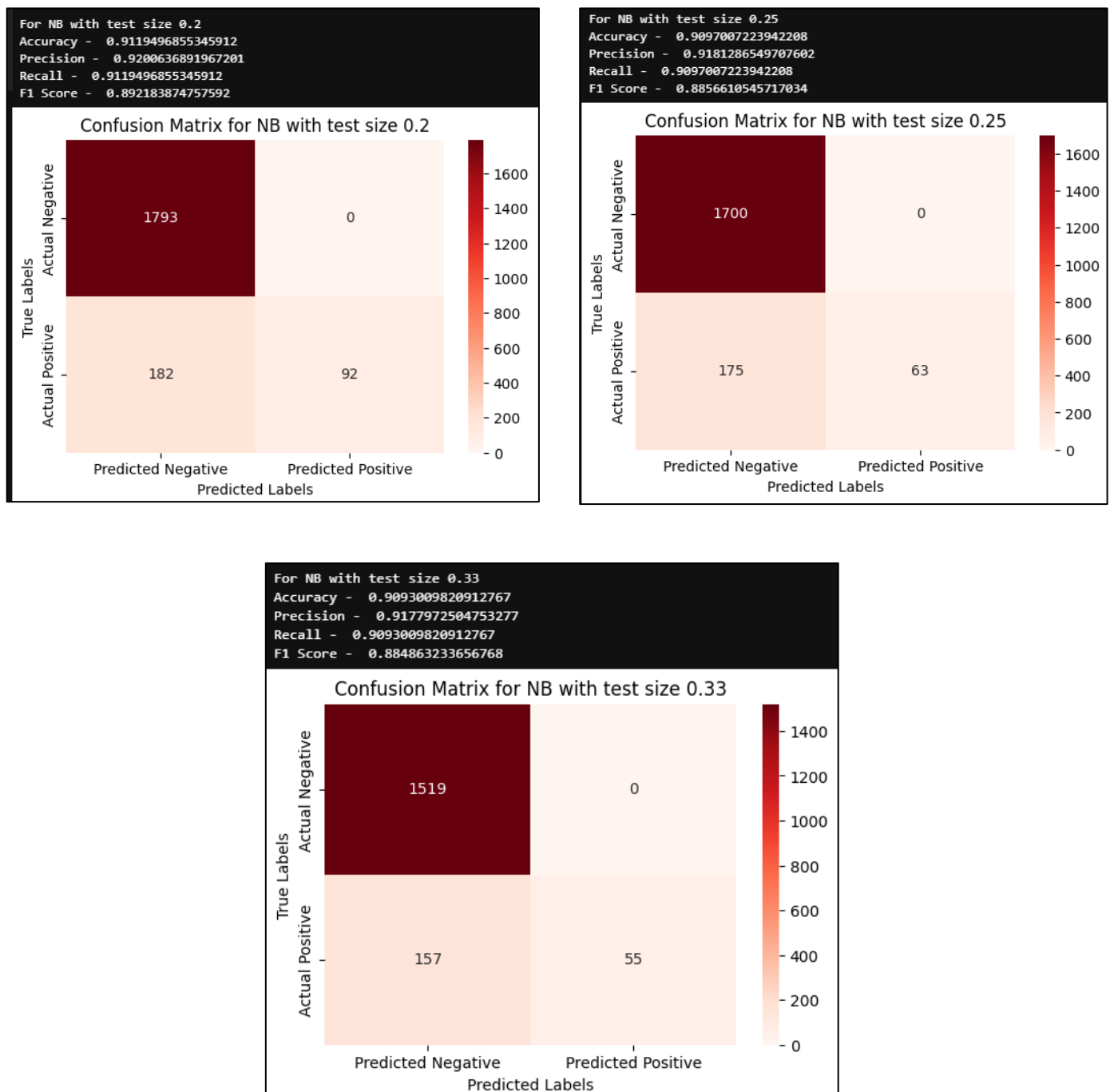


Figure 5.6 : Confusion Matrix for split type Cross Validation and with three spilled ratios as 0.20,0.25 and 0.33 respectively using the navies Bayes classifier.

### 5.3 Model 3: Decision Tree

**Decision Tree** A decision tree is a structure that includes a root node, branches, and leaf nodes. A class label is associated with every leaf node of the decision tree. The non-terminal nodes, which include the root and internal nodes, contain attribute test conditions. Each possible outcome of the attribute test condition is associated with exactly one child of this node. Starting from the root node, we apply the attribute test condition and follow the appropriate branch based on the outcome of the test. This will lead us either to another internal node, for which a new attribute test condition is applied or to a leaf node. Once a leaf node is reached, we assign the class label associated with the node to the test instance.

Important Terminologies:

1. **Root Node:** This attribute is used for dividing the data into two or more sets. The feature attribute in this node is selected based on Attribute Selection Techniques.
2. **Branch or Sub-Tree:** A part of the entire decision tree is called a branch or sub-tree.
3. **Splitting:** Dividing a node into two or more sub-nodes based on if-else conditions.
4. **Decision Node:** After splitting the sub-nodes into further sub-nodes, then it is called the decision node.
5. **Leaf or Terminal Node:** This is the end of the decision tree where it cannot be split into further sub-nodes.
6. **Pruning:** Removing a sub-node from the tree is called pruning

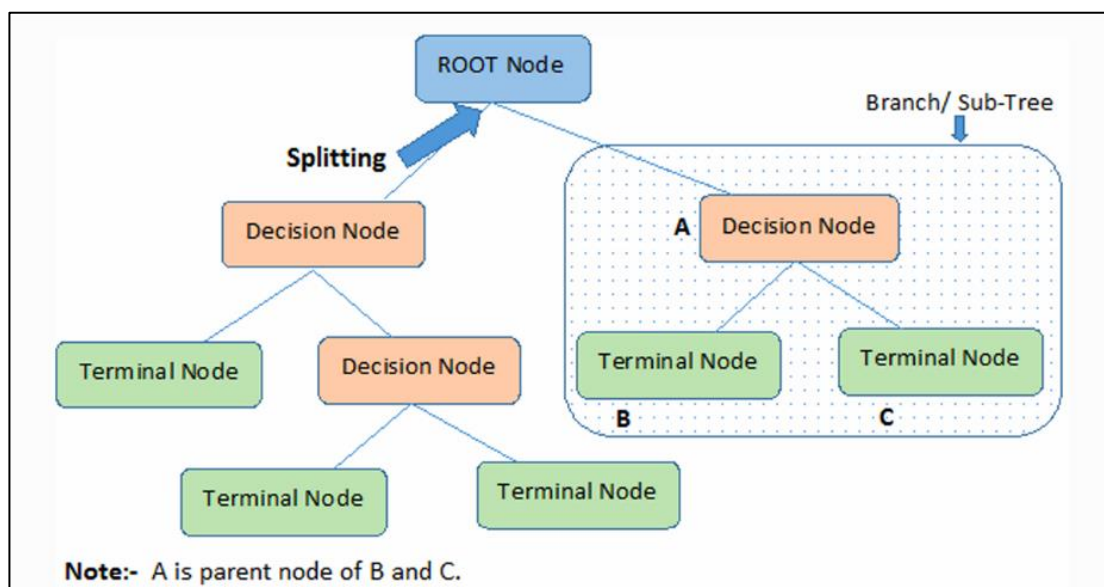


Figure 5.7- Decision Tree  
Source: [Decision Tree](#)

## Working of Decision Tree

1. The root node feature is selected based on the results from the Attribute Selection Measure(ASM).
2. The ASM is repeated until a leaf node, or a terminal node cannot be split into sub-nodes.

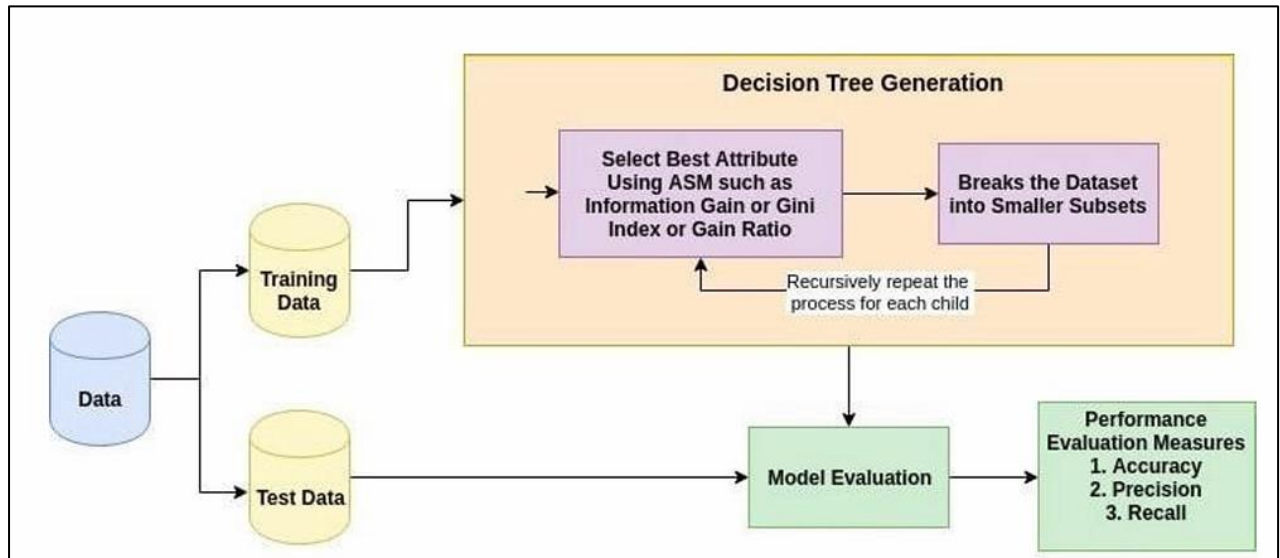


Figure 5.8- Working of Decision Tree

Source: [working of Decision Tree](#)

## SPLIT METHOD: HOLD OUT

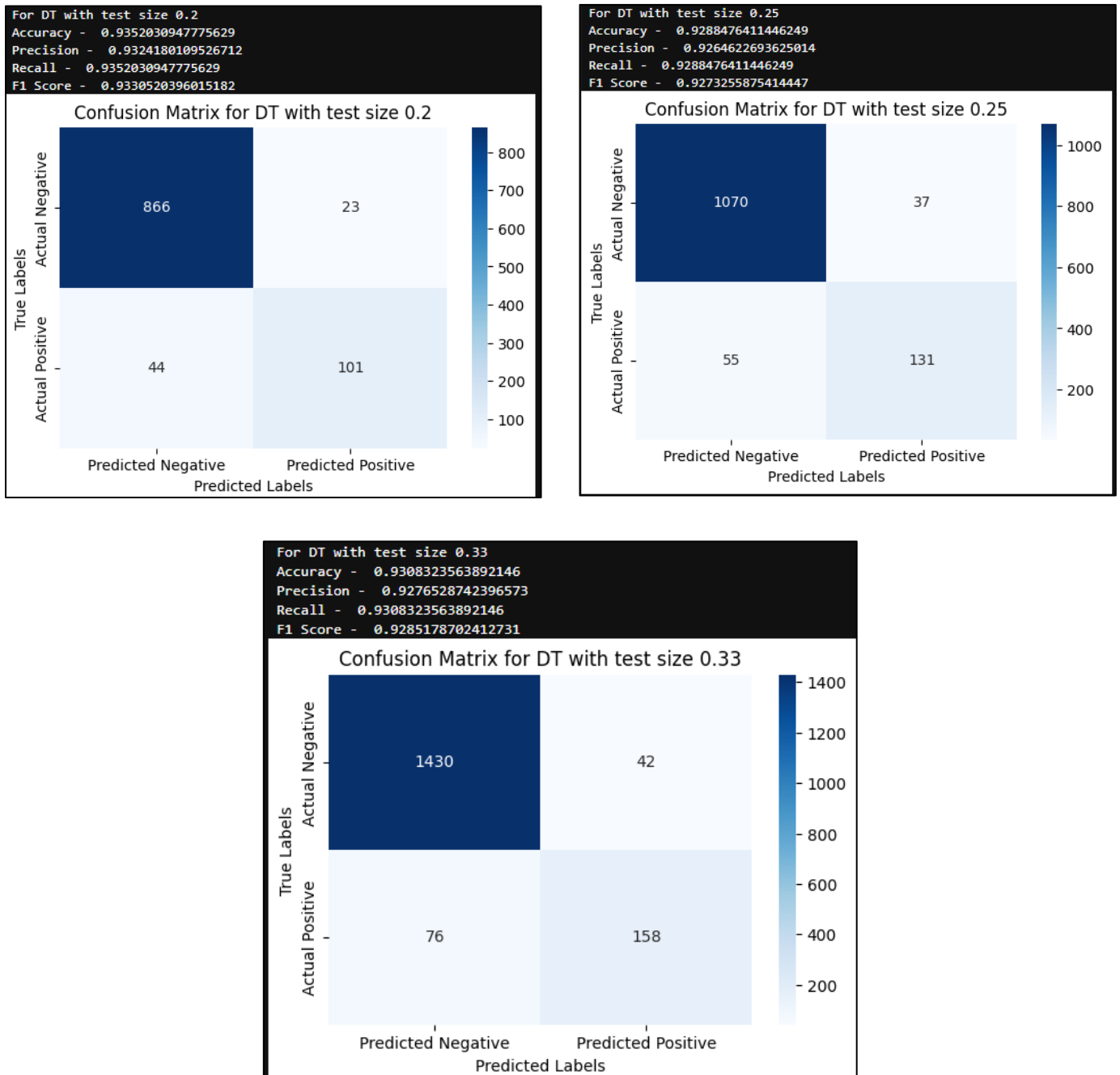


Figure 5.9: Confusion Matrix for split type hold out and with three split ratio as 0.20, 0.25 and 0.33 respectively using the decision tree classifier.

## SPLIT METHOD – RANDOM SUBSAMPLING

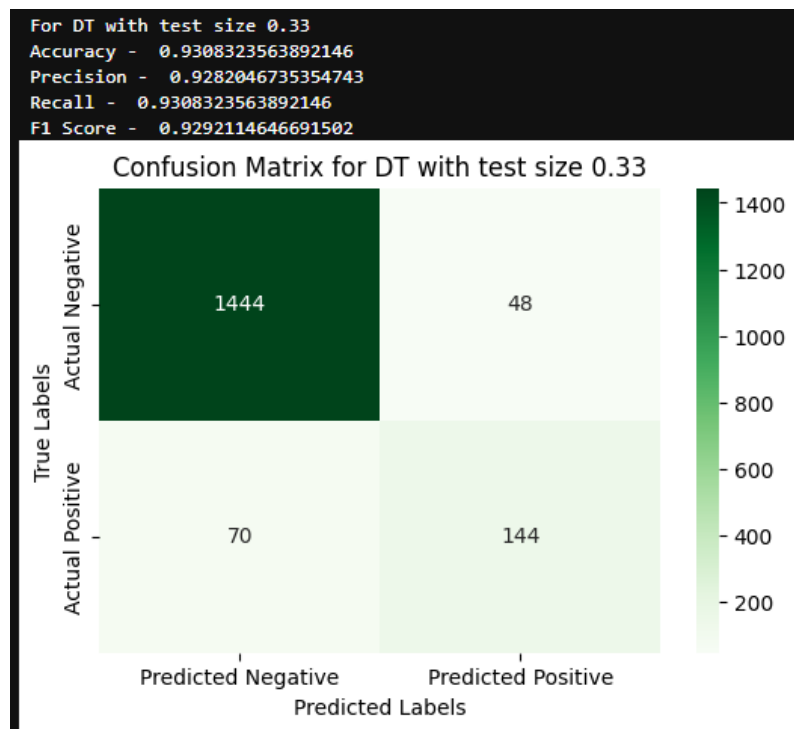
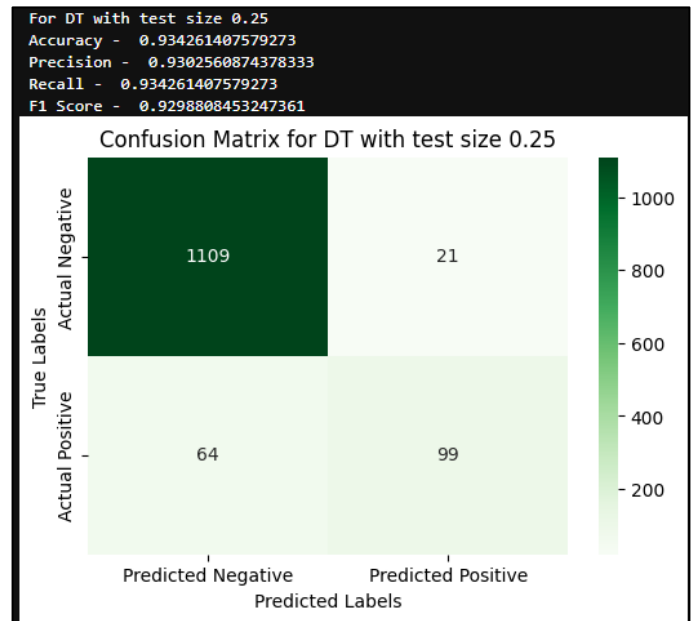
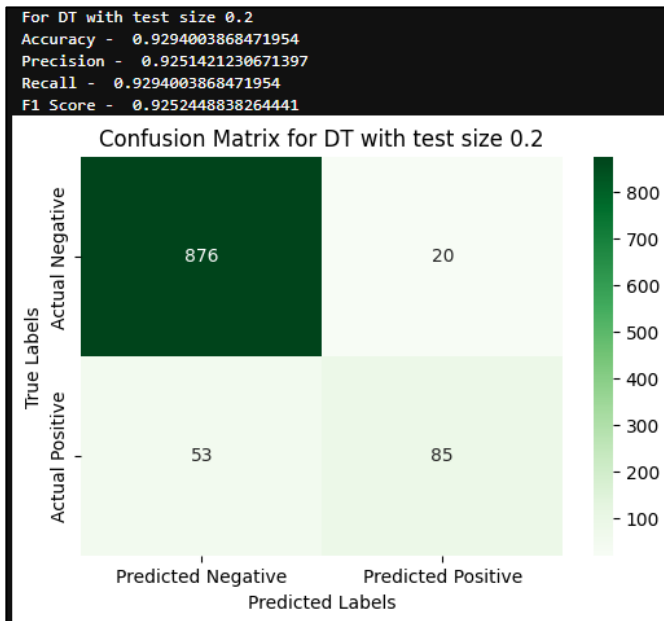


Figure 5.10: Confusion Matrix for split type Random Subsampling and with three split ratio as 0.20, 0.25 and 0.33 respectively using the decision tree classifier.



## SPLIT METHOD: CROSS VALIDATION

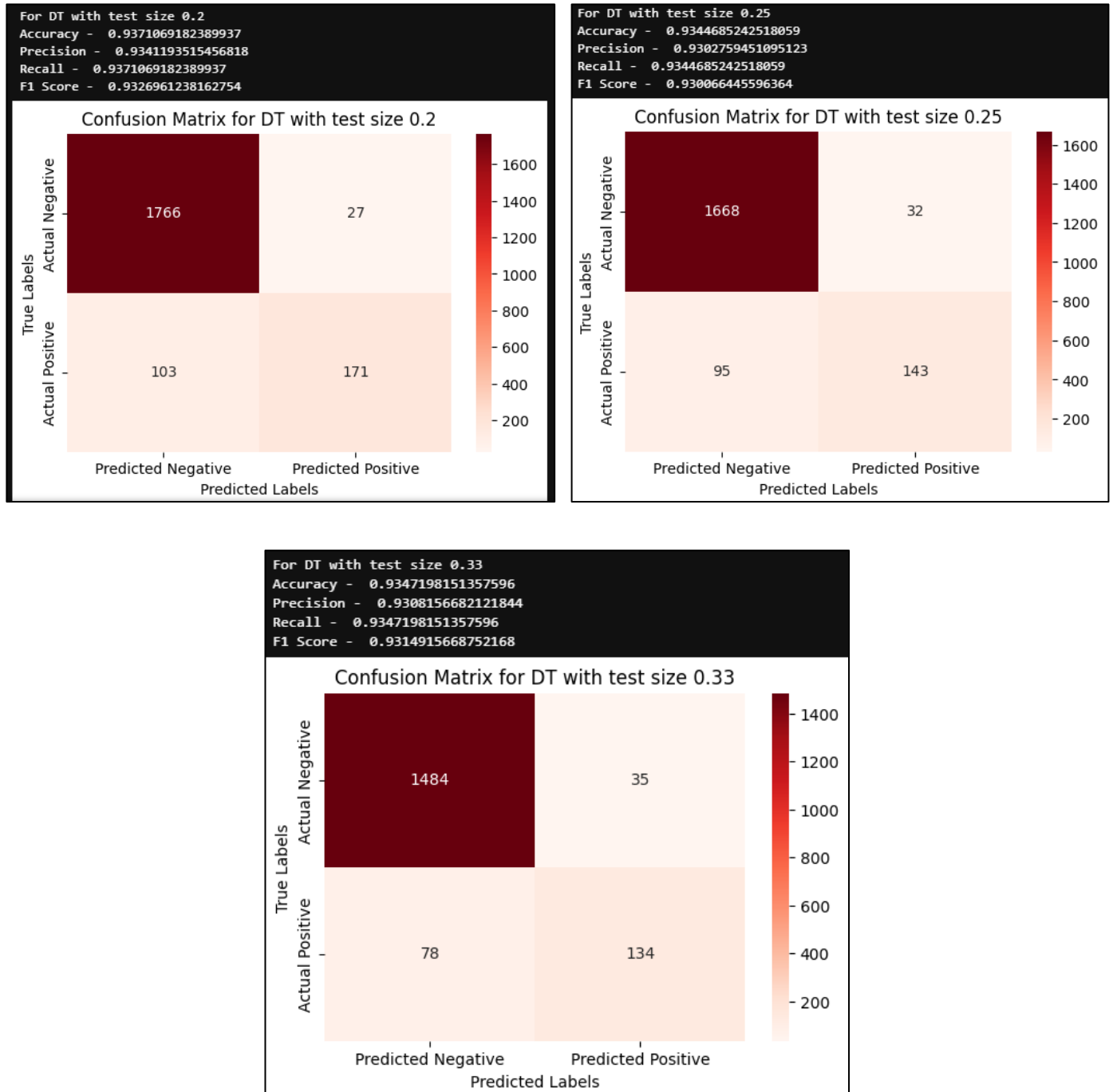


Figure 5.11: Confusion Matrix for split type Cross Validation and with three split ratios as 0.20, 0.25 and 0.33 respectively using the decision tree classifier.

## Chapter – 6

### MODEL EVALUATION AND RESULTS

#### 6.1 METRICS

##### 6.1.1 ACCURACY

Model accuracy is a machine learning model performance metric that is defined as the ratio of true positives and true negatives to all positive and negative observations. In other words, accuracy tells us how often we can expect our machine learning model will correctly predict an outcome out of the total number of times it made predictions.

Mathematically, it represents the ratio of the sum of true positives and true negatives out of all the predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

##### 6.1.2 PRECISION

The model precision score represents the model's ability to correctly predict the positives out of all the positive predictions it made. The precision score is a useful measure of the success of prediction when the classes are very imbalanced. Mathematically, it represents the ratio of true positive to the sum of positive. true positive and false

$$Precision = \frac{TP}{TP + FP}$$

##### 6.1.3 RECALL

Recall score is a useful measure of success of prediction when the classes are very imbalanced. This is unlike precision which measures how many predictions made by models are positive out of all positive predictions made. The higher the recall score, the better the model is at identifying both positive and negative examples. Mathematically, it represents the ratio of true positive to the sum of negative.

$$Recall = \frac{TP}{TP + FN}$$

### 6.1.4 F1-SCORE

True positive and false Model F1 score represents the model score as a function of precision and recall score. F-score is a model performance metric that gives equal weight to both the Precision and Recall for measuring its performance in terms of accuracy, making it an alternative to Accuracy metrics (it doesn't require us to know the total number of observations). It's often used as a single value that provides high-level information about the model's output quality. This is a useful measure of the model in the scenarios where one tries to optimize either of precision or recall score and as a result, the model performance suffers.

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

### 6.2. CONFUSION MATRIX

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values.

with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:

- The target variable has two values: Positive or Negative
- The columns represent the actual values of the target variable
- The rows represent the predicted values of the target variable

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Here,

**True Positive (TP)**

- The predicted value matches the actual value
- The actual value was positive, and the model predicted a positive value

**True Negative (TN)**

- The predicted value matches the actual value
- The actual value was negative, and the model predicted a negative value

**False Positive (FP)**

- The predicted value was falsely predicted
- The actual value was negative, but the model predicted a positive value

**False Negative (FN)**

- The predicted value was falsely predicted
- The actual value was positive, but the model predicted a negative value

### 6.3. Experimental Results and Comparison

Model	Split Type	Test Size	Accuracy	Precision	Recall	F1 Score
KNN	Hold Out	0.20	0.89651	0.90763	0.89651	0.86920
		0.25	0.88863	0.90145	0.88863	0.85686
		0.33	0.89214	0.90412	0.89214	0.86037
	Random Sampling	0.20	0.87911	0.89391	0.87911	0.83301
		0.25	0.88476	0.89818	0.884764	0.83982
		0.33	0.88511	0.89845	0.88511	0.84012
	Cross Validation	0.20	0.87663	0.89199	0.89199	0.82704
		0.25	0.88338	0.89706	0.88338	0.83430
		0.33	0.88272	0.89655	0.88272	0.83253
Navies Bayes	Hold Out	0.20	0.96131	0.96251	0.96131	0.95880
		0.25	0.95204	0.95459	0.95204	0.94790
		0.33	0.95134	0.95394	0.95134	0.94677
	Random Sampling	0.20	0.93036	0.93554	0.93036	0.94942
		0.25	0.92266	0.92894	0.92266	0.90718
		0.33	0.91735	0.92448	0.91735	0.89891
	Cross Validation	0.20	0.91194	0.92006	0.91194	0.89218
		0.25	0.90970	0.91812	0.90970	0.88566
		0.33	0.90930	0.91779	0.90930	0.88486
Decision Tree	Hold out	0.20	0.93520	0.93241	0.93520	0.93305
		.25	0.92884	0.92646	0.92884	0.92732
		.33	0.93083	0.92765	0.93083	0.92851
	Random Sampling	.20	0.92940	0.92514	0.92940	0.92524
		.25	0.93426	0.93025	0.93426	0.92988
		.33	0.93083	0.92820	0.93083	0.92921
	Cross Validation	.20	0.93710	0.93411	0.93710	0.93269
		.25	0.93446	0.93027	0.93446	0.93006
		.33	0.93471	0.93081	0.93471	0.93149

## Chapter – 7

### INFERENCES AND CONCLUSION

In the Context of different model evaluation firstly, K- nearest neighbour has shown a good performance with maximum accuracy in holdout splitting with test size 0.20 whereas same can be seen in the Navies bayes and decision tree. While the increase in the test size shows a slight decline in the performance of K-nearest neighbour except in random subsampling in that particular model, In Navies bayes same pattern can be seen with same exception with random subsampling in which the test size with 0.25 have high accuracy in same model and split type whereas decision tree is all along the declined accuracy for every split in accordance to the test size.

The Highest Accuracy obtained in the project is 0.96131 which is of Navies bayes model with hold out split and 0.20 test size and lowest accuracy can be seen is 0.87663 which is of KNN with cross validation split of test size 0.25.

In this context and the dataset, the conclusion can be made on two bases the first is the sampling that has been chosen the best and secondly is which metric is better for best conclusion.

In concern to the sampling, Evaluation based on the Hold-out set can have a high variance because it depends heavily on which data points end up in the training set and which in test data and as the data is sparse data hold out method is not well suited, further random subsampling and cross validation can play the major role in this prediction which assures the less chance of high variance.

Moving further to the metric from the classification report the best metric according to this project will be recall because Recall is important in medical cases where it doesn't matter whether we raise a false alarm, but the actual positive cases should not go undetected.

So, according to the understanding classification report for all the model conclusions can be made that Navies Bayes Model is the best fit approach to this case study with hold out split type with test size 0.20 (80% - train data, 20% - test data) which gives the maximum recall of 0.9588.

## Chapter 8

### REFERENCES

1. GeeksforGeeks:  
<https://www.geeksforgeeks.org/introduction-of-holdout-method/>
2. Data Mining Techniques, Retrieved from Talend:  
<https://www.talend.com/resources/data-mining-techniques/>
3. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 2016.
4. [http://www.frank-dieterle.com/phd/2\\_4\\_3.html](http://www.frank-dieterle.com/phd/2_4_3.html)
5. <https://towardsai.net/p/programming/decision-trees-explained-with-a-practical-example-fe47872d3b53#:~:text=A%20decision%20tree%20is%20one,it%20according%20to%20the%20condit%20ions.>
6. <https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model>
7. Dataset Link: [SMS Spam Collection Dataset](#)