SUMMER TRAINING REPORT

On

AI/ML using Python

Submitted in partial fulfilment of requirements for the award of the

Degree of

Bachelor of Technology

In

ECE

Submitted By

Triyan Gupta

12211502822

Under the guidance of

Mr. Achin Jain

Mr. A K Dubey

Mrs Neha Sharma

Mrs Neha Gupta



ECE Department

Bharati Vidyapeeth’s College of Engineering, New Delhi – 110063, INDIA

# 

# TABLE OF CONTENTS

Chapter 1- INTRODUCTION TO PROJECT ............................................................................................................1

Chapter 2- PROJECT WORK UNDERTAKEN

...........................................................................................................2

Chapter 3- RESULT AND DISCUSSION

...........................................................................................................7

Chapter 4- SUMMARY

...........................................................................................................8

Chapter 5- DATASHEET

...........................................................................................................9

Chapter 6- REFERENCES

...........................................................................................................11

# LIST OF FIGURES:-

FIG 1- SHOWCASING VARIOUS STEPS OF MACHINE LEARNING

FIG 2.1- SPAM vs HAM PIECHART

FIG 2.2 - SPAM vs HAM HISTOGRAM

FIG 2.3 - WEB APP INTERFACE

# 

# ABOUT INSTITUTE:-

To be an institute of excellence that provides quality technical education and research to create competent graduates for serving industry and society.

**INSTITUTE  MISSION:-**

* **M1:** To impart quality technical education through dynamic teaching-learning environment
* **M2:** To promote research and innovation activities which gives opportunities for life-long learning in context of academic and industry.
* **M3:** To build up links with industry-institute through partnerships and collaborative developmental works.
* **M4:** To inculcate work ethics and commitment in graduates for their future endeavors to serve the society.

# ACKNOWLEDGEMENT: -

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project.

First and foremost, I wish to thank my supervisor, Mr. Achin Jain and Mr. A K Dubey for their invaluable guidance, continuous support, and encouragement throughout this project. Their insights and feedback have been instrumental in shaping my research and enhancing my understanding of the subject.

I am also deeply grateful to Mrs Neha Sharma and Mrs Neha Gupta for their advice and for providing the resources necessary to carry out this work. Their expertise and constructive criticism were greatly appreciated.

Special thanks to my peers and classmates, especially Pradyumna and Priyanshu, for their collaboration, support, and for creating a stimulating environment for discussion and feedback. Your camaraderie and shared knowledge have been a tremendous help.

I would also like to acknowledge my college BVCOE New Delhi for providing the resources and facilities that were crucial for this project.

Lastly, I extend my heartfelt thanks to my family and friends for their unwavering support and encouragement. Their patience and understanding have been a constant source of motivation.

This project would not have been possible without the collective effort and support of all these individuals. Thank you.

# ABSTRACT: -

In an era where digital communication is ubiquitous, the challenge of managing and filtering unwanted spam messages has become increasingly critical. This project addresses this issue by developing a robust spam detection application using advanced artificial intelligence (AI) and machine learning (ML) techniques.

The primary objective of this project was to design and implement a spam detection system that effectively distinguishes between legitimate and spam messages. The application leverages state-of-the-art machine learning algorithms, including Natural Language Processing (NLP) and classification models, to analyze and categorize text messages.

The system was trained on a diverse dataset comprising a significant volume of both spam and non-spam messages.

Various preprocessing techniques, such as tokenization, stop-word removal, and stemming, were employed to enhance the quality of the data. The machine learning modelswere then evaluated for accuracy and performance.

The final application demonstrated a high accuracy rate in detecting spam messages, providing an effective solution to the problem of unsolicited digital content. The integration of this application into messaging platforms can significantly improve user experience by filtering out unwanted messages and reducing the risk of phishing and other malicious activities.

This project not only highlights the potential of AI/ML in solving real-world problems but also serves as a practical implementation of these technologies in the field of digital communication security.

# Chapter 1- Introduction to Project

Machine Learning-Based Spam Detection Using Python

## Introduction

Spam detection is a crucial application in the field of Natural Language Processing (NLP) and Machine Learning (ML). The goal of this project was to develop a spam detection application using basic ML techniques. The application classifies messages as either "spam" or "not spam" based on their content.

## Methodology

Libraries used in Implementation:

* The project employs Python libraries including:
* Pandas for data manipulation.
* NumPy for numerical operations.
* NLTK for text processing
* Matplotlib and Seaborn for data visualization.
* Scikit-learn for feature scaling and evaluation metrics.
* Streamlit for web interface.

## Objectives

* To develop a spam detection system using machine learning algorithms.
* To compare the performance of Multinomial, Bernoulli, and Gaussian Naive Bayes models.
* To evaluate the performance of the model in terms of accuracy, precision and confusion matrix.

To create a user-friendly interface for testing the spam detection functionality.

Page 1

# Chapter 2-PROJECT WORK UNDERTAKEN

## Data Collection

* **Dataset Used:** The dataset for training and evaluating the model was collected from [https://www.kaggle.com/datasets/uciml/smsspam-collection-dataset](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset).
* **Data Description:** The dataset contains text messages labelled as "spam" or "ham" (not spam). Each message is accompanied by a label indicating its classification.

## Data Pre-Processing

* **Cleaning:** Removed special characters, numbers, and excessive white spaces.
* **Normalization:** Converted all text to lowercase to maintain uniformity.
* **Tokenization:** Split messages into individual words.
* **Vectorization:** Converted text into numerical format using techniques such as Bag of Words (BoW) or TF-IDF.

## Model Selection and Training

•**Algorithms Evaluated:**

* **Multinomial Naive Bayes:** Suitable for text classification tasks where features are word counts or term frequencies.
* **Bernoulli Naive Bayes:** Assumes binary features (word presence or absence) and is effective when features are binary. o**Gaussian Naive Bayes:** Assumes that features follow a Gaussian distribution, less suited for text data.

Page 2

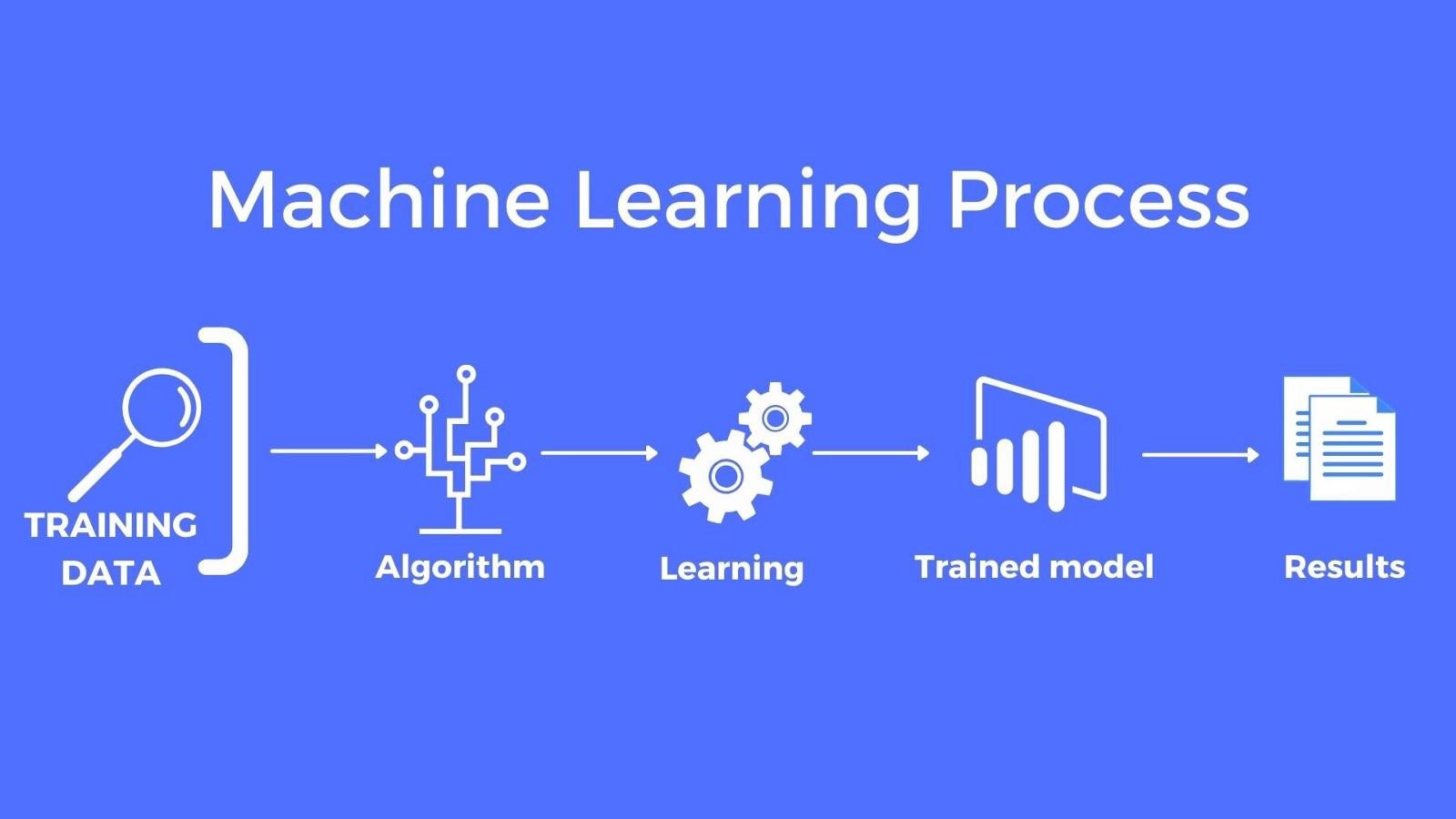
## Model Training

* **Train-Test Split:** The dataset was divided into training (80%) and testing (20%) sets.
* **Training Process:** Each Naive Bayes model was trained on the

training set using Scikit-learn’s implementations.

Fig 1: SHOWCASING VARIOUS STEPS OF MACHINE LEARNING

CODE



import numpy as np import pandas as pd

df = pd.read\_csv('spam.csv',encoding='ISO-8859-1') df.head()

df.shapedf.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed:

4'],inplace=True)

df.rename(columns={'v1':'target','v2':'text'},inplace=True) df.head()

from sklearn.preprocessing import LabelEncoder encoder = LabelEncoder()

df['target'] = encoder.fit\_transform(df['target']) df = df.drop\_duplicates(keep='first')

import matplotlib.pyplot as pltplt.pie(df['target'].value\_counts(), labels=['ham','spam'],autopct="%0.1f") plt.show()

Page 3

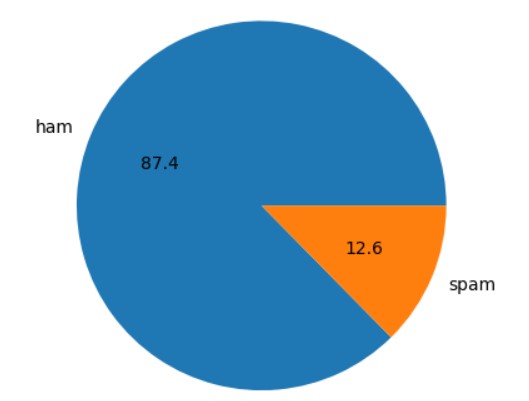
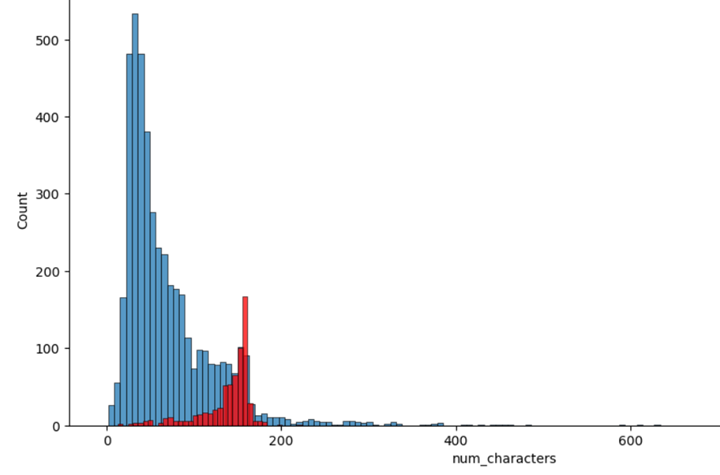


Fig 2.1 : SPAM vs HAM PIECHART

Fig 2.2 : SPAM vs HAM HISTOGRAM

Page 4

import nltk

df['num\_words'] = df['text'].apply(lambda x:len(nltk.word\_tokenize(x))) df['num\_sentences'] = df['text'].apply(lambda x:len(nltk.sent\_tokenize(x)))

df[df['target'] ==

0][['num\_characters','num\_words','num\_sentences']].describe()

plt.figure(figsize=(12,6))

sns.histplot(df[df['target'] == 0]['num\_characters']) sns.histplot(df[df['target'] == 1]['num\_characters'],color='red') plt.show()

from nltk.corpus import stopwords import string from nltk.stem import PorterStemmer from nltk.tokenize import word\_tokenizeps = PorterStemmer() # func to transform text def transform\_text(text): text = text.lower()

text = nltk.word\_tokenize(text)

# removing special chars y = [] for i in text:

# is aplhanumeric then append in list y

if i.isalnum():

y.append(i)

# copy y to text text = y[:] y.clear()

# removestopword and punctuation for i in text:

if i not in stopwords.words('english') and i not in string.punctuation: y.append(i)

text = y[:] y.clear()

# stemming for i in text:

y.append(ps.stem(i))

return " ".join(y)

df['transformed\_text'] = df['text'].apply(transform\_text)

from sklearn.feature\_extraction.text import CountVectorizer,TfidfVectorizer # bag of words:

Page 5

cv = CountVectorizer()

# tfidf

tfidf = TfidfVectorizer(max\_features=3000)

X = tfidf.fit\_transform(df['transformed\_text']).toarray() y = df['target'].values X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=2)

from sklearn.naive\_bayes import

GaussianNB,MultinomialNB,BernoulliNB from sklearn.metrics import accuracy\_score,confusion\_matrix,precision\_score

gnb = GaussianNB() mnb = MultinomialNB()

bnb = BernoulliNB()

gnb.fit(X\_train,y\_train)

y\_pred1 = gnb.predict(X\_test) print(accuracy\_score(y\_test,y\_pred1)) print(confusion\_matrix(y\_test,y\_pred1))

print(precision\_score(y\_test,y\_pred1))

mnb.fit(X\_train,y\_train) y\_pred2 = mnb.predict(X\_test) print(accuracy\_score(y\_test,y\_pred2)) print(confusion\_matrix(y\_test,y\_pred2))

print(precision\_score(y\_test,y\_pred2))

bnb.fit(X\_train,y\_train) y\_pred3 = bnb.predict(X\_test) print(accuracy\_score(y\_test,y\_pred3)) print(confusion\_matrix(y\_test,y\_pred3))

print(precision\_score(y\_test,y\_pred3))

import pickle pickle.dump(tfidf,open('vectorizer.pkl','wb')) pickle.dump(mnb,open('model.pkl','wb'))

Page 6

# 

# Chapter 3- RESULT AND DISCUSSION

## Model Performance

* **Multinomial Naive Bayes:**
  + **Accuracy:** 0.9709864603481625 o**Precision:** 1.0
* **Bernoulli Naive Bayes:**
  + **Accuracy:** 0.9835589941972921 o**Precision:** 0.991869918699187
* **Gaussian Naive Bayes:**
  + **Accuracy:** 0.874274661508704 o**Precision:** 0.5181818181818182

Comparative Analysis

* **Multinomial Naive Bayes** performed the best among the evaluated models, with the highest accuracy, precision and recall . This aligns with its suitability for text classification tasks where feature counts are relevant.

## Challenges

* **Data Quality:** The effectiveness of the models depends on the quality and size of the dataset.
* **Feature Representation:** The choice of feature representation (TF-IDF) influences model performance.
* **Model Limitations:** Naive Bayes models assume feature independence, which might not hold true in all text classification scenarios.

Page 7

# Chapter 4- SUMMARY:-

**Title:** Spam Detection Using Artificial Intelligence and Machine Learning

**Overview:**

The project focuses on developing an intelligent spam detection system leveraging artificial intelligence (AI) and machine learning (ML) techniques to enhance digital communication security. The primary goal was to create a robust application capable of accurately classifying messages as either spam or legitimate, thereby improving user experience and safeguarding against unwanted and potentially harmful content.

**Objective:**

The main objective was to design and implement a machine learning-based system that can effectively distinguish between spam and non-spam messages. This involved training a model on a dataset of text messages and employing various AI/ML techniques to achieve high accuracy in classification.

**Results:**

The developed spam detection application achieved a high level of accuracy, effectively distinguishing between spam and non-spam messages. The system demonstrated a robust performance across different metrics, indicating its reliability in real-world scenarios.

**Conclusion:**

This project successfully implemented a machine learning-based solution for spam detection, showcasing the potential of AI in enhancing digital communication security. The application provides a practical tool for filtering unwanted messages, thus contributing to a more secure and efficient messaging experience.

**Future Work:**

Future enhancements may include expanding the dataset, incorporating more advanced machine learning models, and integrating additional features to further improve detection accuracy and user experience.

Page 8

# DATA SHEET: -

DATASET LINK:-

* **Dataset Used:** The dataset for training and evaluating the model was collected from [https://www.kaggle.com/datasets/uciml/smsspam-collection-dataset](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset).

CODE:-

import streamlit as st

import pickle

import string

from nltk.corpus import stopwords

import nltk

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

def transform\_text(text):

text = text.lower()

text = nltk.word\_tokenize(text)

y = []

for i in text:

if i.isalnum():

y.append(i)

text = y[:]

y.clear()

for i in text:

if i not in stopwords.words('english') and i not in string.punctuation:

y.append(i)

text = y[:]

y.clear()

for i in text:

y.append(ps.stem(i))

return " ".join(y)

# Load vectorizer and model

Page 9

tfidf = pickle.load(open('vectorizer.pkl', 'rb'))

model = pickle.load(open('model.pkl', 'rb'))

st.title("Email/SMS Spam Classifier")

input\_sms = st.text\_area("Enter the message")

if st.button('Predict'):

# 1. preprocess

transformed\_sms = transform\_text(input\_sms)

# 2. vectorize

vector\_input = tfidf.transform([transformed\_sms])

# 3. predict

result = model.predict(vector\_input)[0]

# Debug: print the result value

st.write("Debug: Prediction result = {}".format(result))

# 4. Display

if result == 1:

st.header("Spam")

else:

st.header("Not Spam")

Fig 2.3:- WEB APP INTERFACE

Page 10

# REFERENCES: -

**Links:-**

 **Python Software Foundation**. Python Language Reference, version 3.9. Available at: <https://docs.python.org/3.9/>

 **Applying Multinomial Naive Bayes to NLP Problems**

**-** <https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/>

 **NumPy Developers**. NumPy Documentation, version 1.21. Available at: https://numpy.org/doc/stable/

 **Scikit-Learn Developers**. scikit-learn: Machine Learning in Python, version 0.24. Available at: https://scikit-learn.org/stable/

 **Streamlit Inc.**. Streamlit Documentation, version 0.88. Available at: https://docs.streamlit.io/en/stable/

 **Natural Language Processing Techniques and Examples**

- <https://www.revuze.it/blog/natural-language-processing-techniques/>

 **Spam Detection using Machine Learning Methods**

**-https://medium.com/@Coursesteach/spam-detection-using-machine-learning-methods-dd5dbc799b6b**

 **Brownlee, J.** (2019). Machine Learning Mastery With Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End. Machine Learning Mastery.

**Books:-**

 **Goodfellow, I., Bengio, Y., & Courville, A.** (2016). Deep Learning. MIT Press.  
This textbook provides a comprehensive introduction to deep learning, including foundational concepts that are applicable to machine learning projects like spam detection.

 **Alpaydin, E.** (2020). Introduction to Machine Learning (4th Edition). MIT Press.  
This book covers the principles and techniques of machine learning, offering insights that are directly relevant to your project.

Page 11

 **Murphy, K. P.** (2012). Machine Learning: A Probabilistic Perspective. MIT Press.  
This book provides a detailed exploration of machine learning from a probabilistic standpoint, useful for understanding the algorithms and models used in your spam detection project.

Page 12