Final Project Report Template

1.Introduction

In today's world, almost everyone has a mobile phone, and text messaging has become extremely popular. But along with this popularity comes a significant issue: spam. Spam messages are those annoying and sometimes harmful texts that flood our phones, attempting to sell us products or deceive us into revealing personal information. To address this problem, we can employ technology like machine learning, which has the ability to automatically identify and filter out spam messages. One effective technique is the Multinomial Naive Bayes classifier.

1.1 Project Overview

The SMS Spam Detection Project aims to create an automated system that effectively identifies and filters spam messages from legitimate ones, enhancing user security and experience. Leveraging machine learning algorithms, the project will analyze SMS content to classify messages accurately. It will encompass data collection, preprocessing, model training, and validation phases, ensuring the system adapts to evolving spam tactics. Key stakeholders include end-users, development teams, and mobile service providers, all invested in reducing spam's negative impact. A user-friendly interface will allow users to report inaccuracies, contributing to continuous improvement.

1.2 Objectives of SMS Spam Detection

--Identify Spam Messages: Accurately classify incoming SMS messages as spam or legitimate based on predefined criteria and patterns.

--Enhance User Experience: Reduce the volume of unwanted messages received by users, ensuring they can focus on important communications.

--Protect User Privacy: Safeguard users from potential privacy breaches and malicious content that often accompanies spam messages.

--Reduce Fraud Risk: Mitigate the risks associated with phishing and other scams that can be propagated through spam SMS.

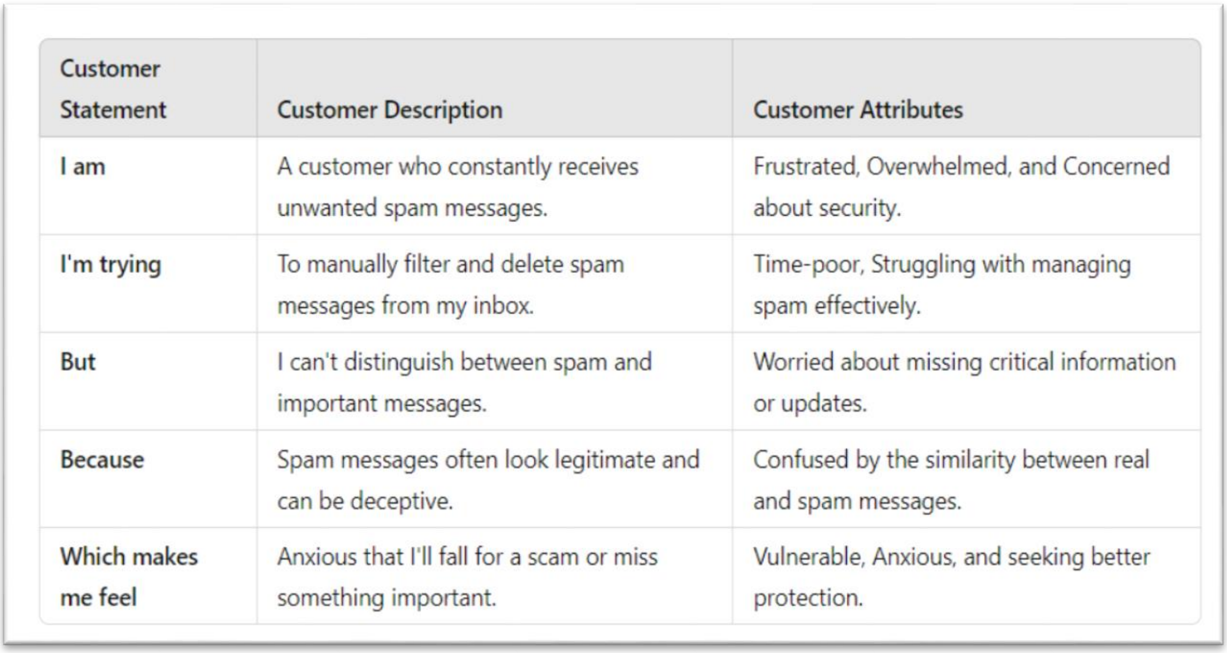
2.Project Initialization and Planning Phase

2.1 Define Problem Statements

Date	15 August 2024
Team ID	LTVIP2024TMID24955
Project Name	SMS Spam Detection - Aiml
Maximum Marks	3 Marks

Define Problem Statements

As a mobile user, I frequently receive unsolicited and unwanted SMS messages that disrupt my daily activities. These spam messages often contain misleading information, fraudulent offers, or malicious links that put my privacy and security at risk. I need an efficient and reliable solution that can accurately detect and filter out spam messages before they reach my inbox



Customer Statement	Customer Description	Customer Attributes
I am	A customer who constantly receives unwanted spam messages.	Frustrated, Overwhelmed, and Concerned about security.
I'm trying	To manually filter and delete spam messages from my inbox.	Time-poor, Struggling with managing spam effectively.
But	I can't distinguish between spam and important messages.	Worried about missing critical information or updates.
Because	Spam messages often look legitimate and can be deceptive.	Confused by the similarity between real and spam messages.
Which makes me feel	Anxious that I'll fall for a scam or miss something important.	Vulnerable, Anxious, and seeking better protection.

Example:

Problem Statement (PS)	I am (Customer)	I’m trying to	But	Because	Which makes me feel
PS-1	Receiving unwanted messages	Filtering	I can’t find ham or spam	Often looks likes Legitimate	Anxious, Scared of falling in scam
PS-2	Receives Suspicious links in messages	Detecting	I don’t know About it	Confused by Ham or Spam	Worried, Seeking peace of mind.

Project Initialization and Planning Phase

2.2 Project Proposal (Proposed Solution) template

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	3 Marks

Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	Clearly state the project's primary objective
Scope	Define the boundaries and extent of the project
Problem Statement	
Description	Clearly articulate the problem to be addressed
Impact	Explain the implications of solving the problem
Proposed Solution	
Approach	Outline the methodology and techniques to be used
Key Features	Highlight the unique aspects of the proposed solution

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	e.g., 2 x NVIDIA V100 GPUs
Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models, and logs	e.g., 512 GB SSD
Software		
Frameworks	Python frameworks	e.g., Flask
Libraries	Additional libraries	e.g., scikit-learn, pandas, numpy , matplotlib
Development Environment	IDE, version control	e.g., Jupyter Notebook, Git
Data		
Data	Source, size, format	e.g., Kaggle dataset, 5374 KB Size, CSV format

2.3 Initial Project Planning Template

Date	15 August 2024
Team ID	LTVIP2024TMID24955
Project Name	SMS Spam Detection - AIML
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create a product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	SMS Spam Detection	USN-1	As a user, I want to report spam messages so I can keep my inbox clean.	2	High	Chethala Katam Raju	15-08-2024	22-08-2024
Sprint-1	SMS Spam Detection	USN-2	As a user, I want the system to automatically filter spam messages to reduce clutter.	1	High	D Tandan Sai Nanda Krishna	23-08-2024	30-08-2024
Sprint-2	SMS Spam Detection	USN-3	As a user, I want to customize spam filters based on my preferences.	2	Low	Ganneboia Prabhas	31-08-2024	08-09-2024
Sprint-1	SMS Spam Detection	USN-4	As a user, I want to receive notifications when a spam message is detected.	2	Medium	Chadalavada Jaya Sai Kiran	09-09-2024	16-09-2024
Sprint-1	SMS Spam Detection	USN-5	As a user, I want an easy way to view my spam message history.	1	High	Chethala Katam Raju	17-09-2024	28-09-2024

3.Data Collection and Preprocessing Phase

3.1 Data Collection Plan & Raw Data Sources Identification

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	2 Marks

Data Collection Plan & Raw Data Sources Identification Template

Elevate your data strategy with the Data Collection plan and the Raw Data Sources report, ensuring meticulous data curation and integrity for informed decision-making in every analysis and decision-making endeavor.

Data Collection Plan Template

Section	Description
Project Overview	The project focuses on building an SMS spam detection model that accurately classifies incoming SMS messages as spam or not spam. This involves collecting a diverse set of SMS data, including both spam and non-spam messages, to train, validate, and test machine learning models.
Data Collection Plan	Identify sources of SMS data such as public datasets, user-contributed data, or SMS archives from mobile service providers. Data can be collected from Kaggle datasets
Raw Data Sources Identified	The dataset will consist of SMS message content (text data), along with metadata like message timestamp, sender information, and spam/not spam labels

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
Dataset 1	The data consists of meta data	https://drive.google.com/file/d/1Mf6AHsRuRxO8IynQFJGYi2OWwO4uWVh-/view?usp=sharing	CSV	5374 KB	Public

Data Collection and Preprocessing Phase

3.2 Data Quality Report Template

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	2 Marks

Data Quality Report Template

The Data Quality Report Template will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

Data Source	**Data Quality Issue**	**Severity**	**Resolution Plan**
Dataset	Mention the issues faced in the selected dataset.	Low/ Moderate/ High	Give the solution for that issue technically.
Dataset	Duplicate Records	Moderate	Implement deduplication processes to identify and remove duplicate entries from the dataset to ensure unique records.

Dataset	Inconsistent Formatting	High	Standardize formats for text fields (e.g., phone numbers, dates) to ensure uniformity across the dataset.
Dataset	Irrelevant Data	Low	Review the dataset to identify and remove unnecessary columns or records that do not contribute to the analysis.
Dataset	Missing Values	High	Identify and fill in missing values using imputation methods (mean, median, mode) or remove affected records if the missing percentage is significant.

Data Collection and Preprocessing Phase

3.3 Data Exploration and Preprocessing Template

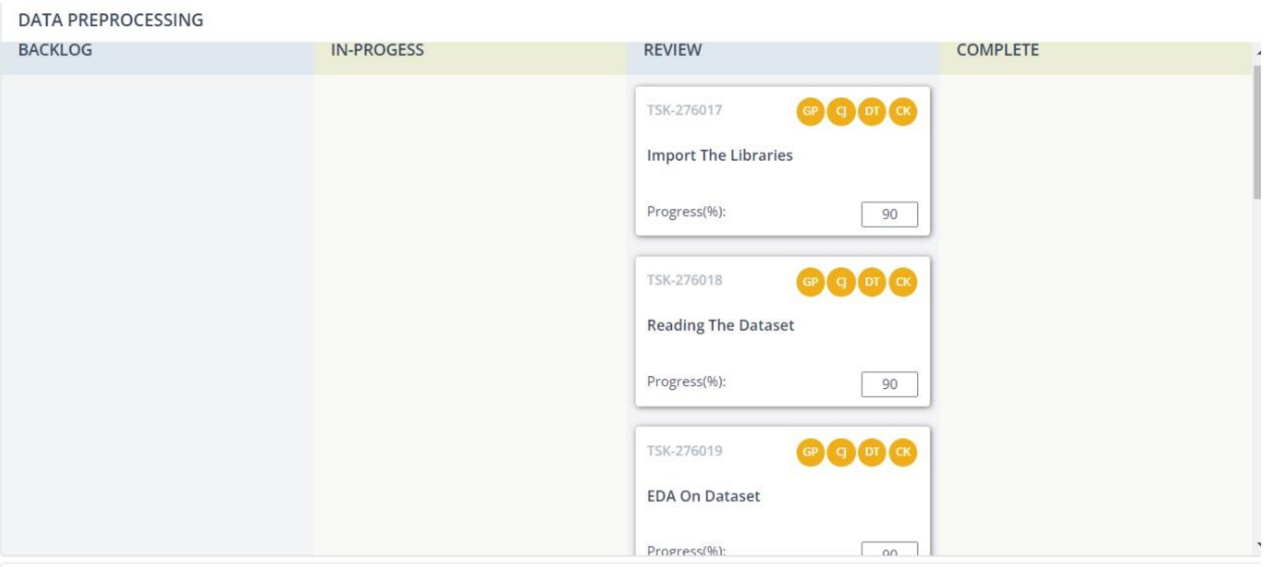
Date	15 August 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**
1. Data Collection	Gathering SMS message data from reliable sources such as SMS datasets (e.g., public datasets, user-contributed data, or synthetic data). Includes both spam and non-spam messages for balanced classification..
2. Data Inspection	Examining the structure of the dataset, including attributes such as message content, labels (spam/not spam), and metadata e.g., time of message, sender info.
3.Exploratory Data Analysis (EDA)	Visualizing and analyzing the distribution of messages, word frequencies, and the relationship between message features (e.g., message length, most common words) and their labels (spam/not spam).
4. Data Cleaning	Removing or correcting noise such as duplicate messages, irrelevant content (e.g., system messages), and non-text characters (special symbols or emojis). Handling missing or incomplete data by filling in, removing, or imputing values.
5. Data Balancing	Addressing any class imbalance between spam and non-spam messages by applying techniques like oversampling

Data Preprocessing Code Screenshots



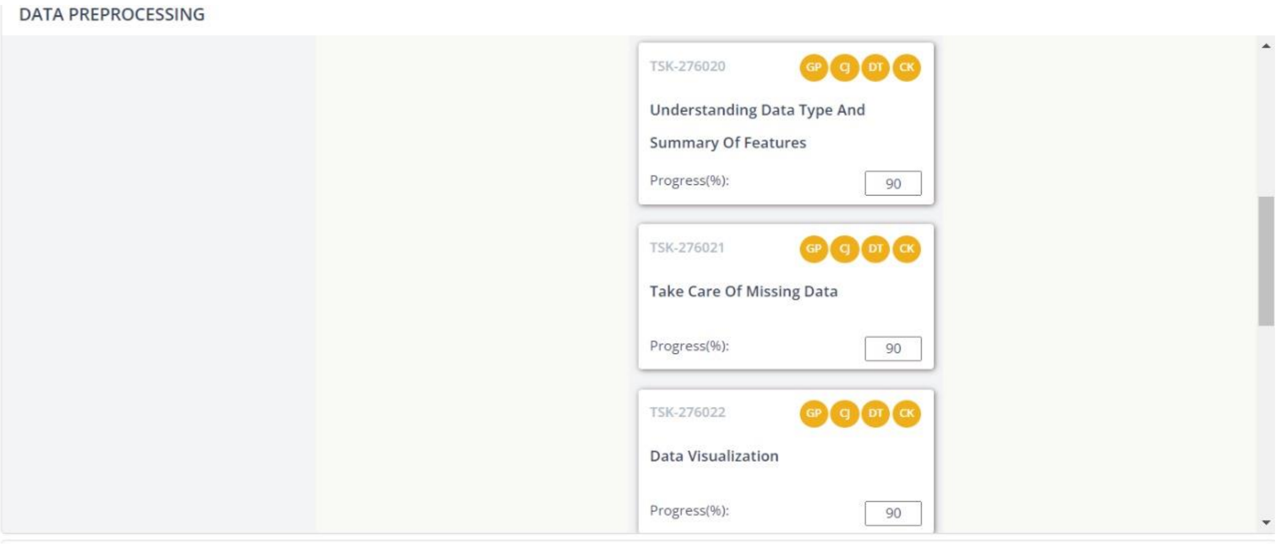
DATA PREPROCESSING

BACKLOG | IN-PROGESS | REVIEW | COMPLETE

TSK-276017 GP CJ DT CK
Import The Libraries
Progress(%): 90

TSK-276018 GP CJ DT CK
Reading The Dataset
Progress(%): 90

TSK-276019 GP CJ DT CK
EDA On Dataset
Progress(%): 90



DATA PREPROCESSING

TSK-276020 GP CJ DT CK
Understanding Data Type And Summary Of Features
Progress(%): 90

TSK-276021 GP CJ DT CK
Take Care Of Missing Data
Progress(%): 90

TSK-276022 GP CJ DT CK
Data Visualization
Progress(%): 90

6. Text Preprocessing	Processing the raw text of SMS messages by converting to lowercase, removing stop words , punctuation, and stemming/lemmatizing. Transforming the text into a format suitable for machine learning models (e.g., tokenization).
7. Label Encoding	Converting the labels (spam, not spam) into a numerical format (e.g., 0 for not spam, 1 for spam) for use in machine learning models.
8. Data Splitting	Dividing the dataset into training, validation, and test sets to evaluate model performance.
9. Model Building	Developing a machine learning model to classify SMS messages as spam or not spam. This includes selecting appropriate algorithms like Multinomial Naïve base
10.Model Evaluation	Assessing the performance of the trained model using various evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. This involves testing the model on the validation and test datasets

4.Model Development Phase Template

4.1 Feature Selection Report Template

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	5 Marks

Feature Selection Report Template

In the forthcoming update, each feature will be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected (Yes/No)	Reasoning
Message Length	The total number of characters in the SMS message.	Yes	Spam messages often have distinct length patterns (e.g., longer promotional or short urgent texts).
Presence of Keywords	Whether specific keywords (e.g., "free," "win," "prize," "click") appear.	Yes	Spam messages frequently include enticing or misleading keywords.
Number of Links	The count of URLs or links within the message.	Yes	Spam messages often contain links leading to phishing sites or offers. Special Characters

Special Characters	Presence of special characters (e.g., "$", "%", "!", "@").	Yes	Spam messages may contain special characters to grab attention or bypass filters.
Sender Information	Metadata about the sender (e.g., phone number, unknown sender, short code).	No	Not always reliable as spammers can spoof phone numbers, and metadata may not be consistent.
Capitalization	Percentage of words in all caps (e.g., "FREE," "URGENT").	Yes	Spam messages frequently use capitalization to emphasize key points or urgency.

Model Development Phase Template

4.2 Model Selection Report

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	6 Marks

Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

Model Selection Report:

Model	Description	Hyperparameters	Performance Metric (e.g., Accuracy, F1 Score)
Multinomial Naïve Bayes	A probabilistic classifier based on applying Bayes' theorem, commonly used for text classification like spam detection.	- Alpha (smoothing): 1.0 - Fit prior: True	Accuracy: 96% - F1 Score: 0.97
SVC (Sigmoid Kernel)	A Support Vector Classifier using the sigmoid kernel for non-linear decision boundaries.	- C: 1.0 - Kernel: Sigmoid - Gamma: Scale	- Accuracy: 96% - F1 Score: 0.95

SVC (RBF Kernel)	A Support Vector Classifier with a Radial Basis Function (RBF) kernel, often effective in high-dimensional spaces.	- C: 1.0 - Kernel: RBF - Gamma: Auto	- Accuracy: 94% - F1 Score: 0.89
Decision Tree Classifier	A tree-based model that splits the data based on feature values to create decision rules for classification.	- Criterion: Gini - Max depth: None - Min samples split: 2	- Accuracy: 94% - F1 Score: 0.87

Model Development Phase Template

4.3 Initial Model Training Code, Model Validation and Evaluation Report

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

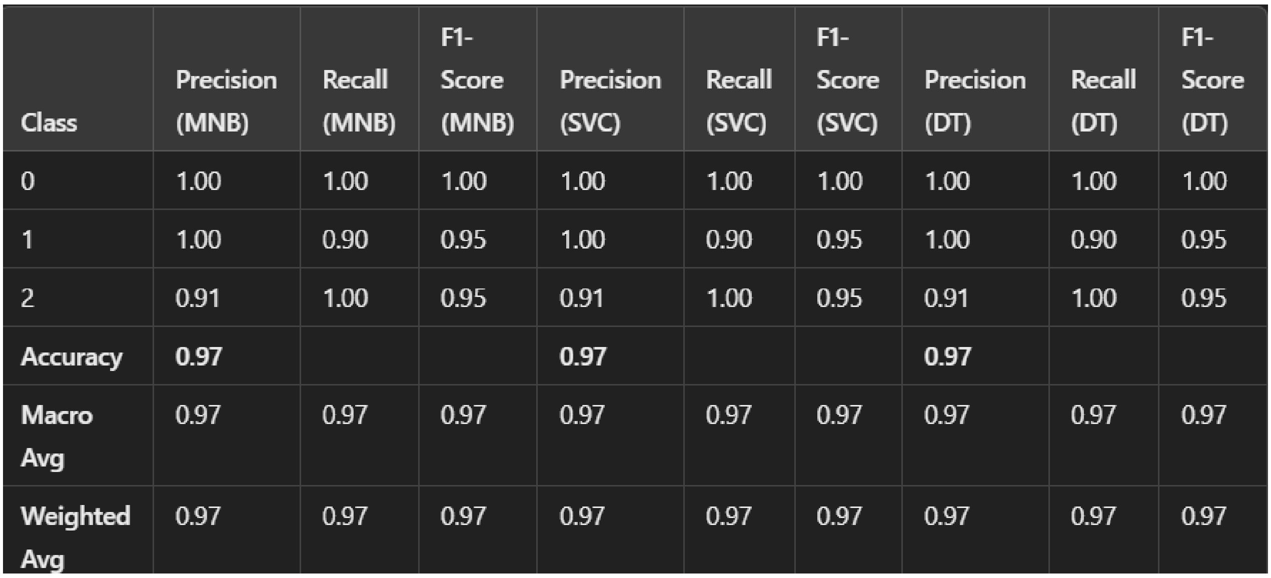
Initial Model Training Code:

Paste the screenshot of the model training code

Model Validation and Evaluation Report:

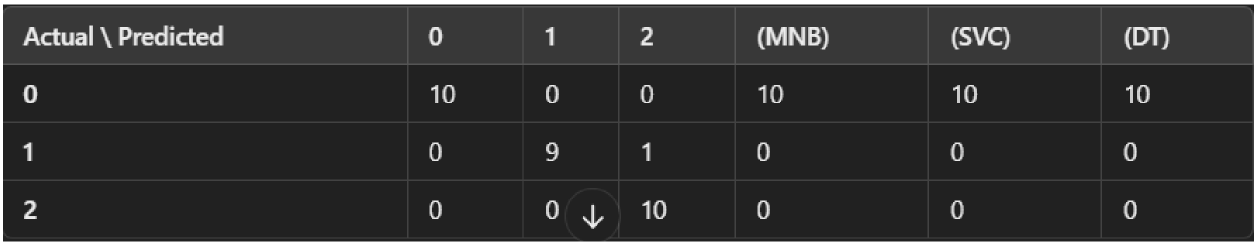
Model	**Accuracy**
Multinomial Naive Bayes	0.9681
SVC(sigmoid)	0.9652
SVC(rbf)	0.9623
Decision Tree Classifier	0.9497

Classification report –screenshot



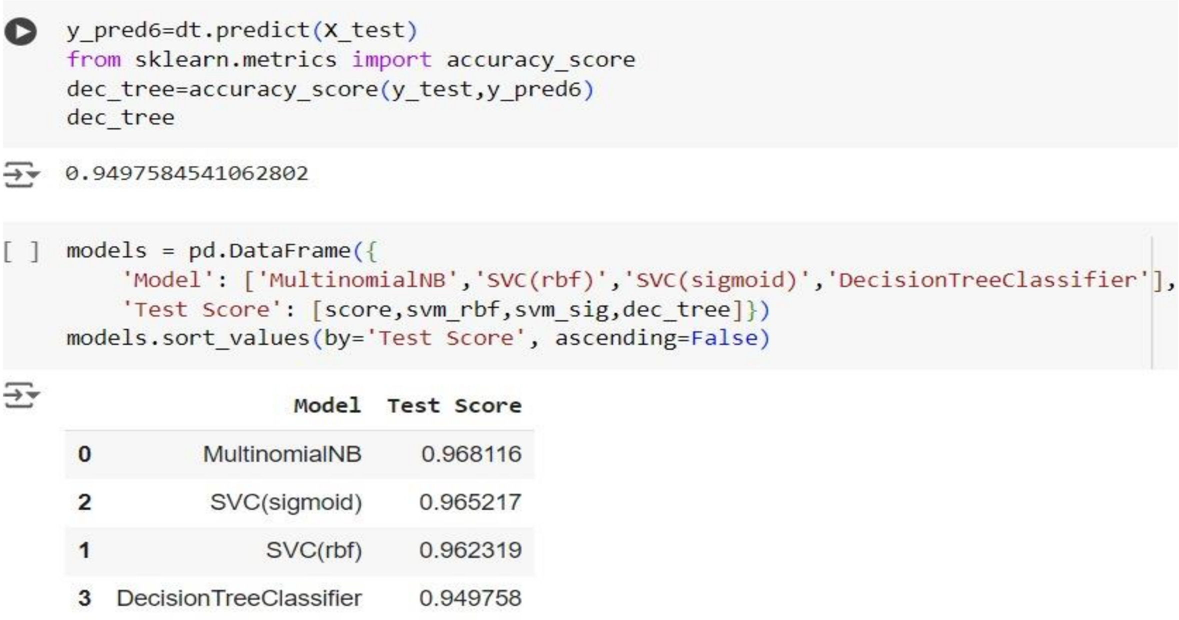
Class	Precision (MNB)	Recall (MNB)	F1-Score (MNB)	Precision (SVC)	Recall (SVC)	F1-Score (SVC)	Precision (DT)	Recall (DT)	F1-Score (DT)
0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	0.90	0.95	1.00	0.90	0.95	1.00	0.90	0.95
2	0.91	1.00	0.95	0.91	1.00	0.95	0.91	1.00	0.95
Accuracy	0.97			0.97			0.97		
Macro Avg	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
Weighted Avg	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97

Confusion report—screenshot



Actual \ Predicted	0	1	2	(MNB)	(SVC)	(DT)
0	10	0	0	10	10	10
1	0	9	1	0	0	0
2	0	0	10	0	0	0

** MODEL -- SVC, ** ACCURACY -- [0.9623]**



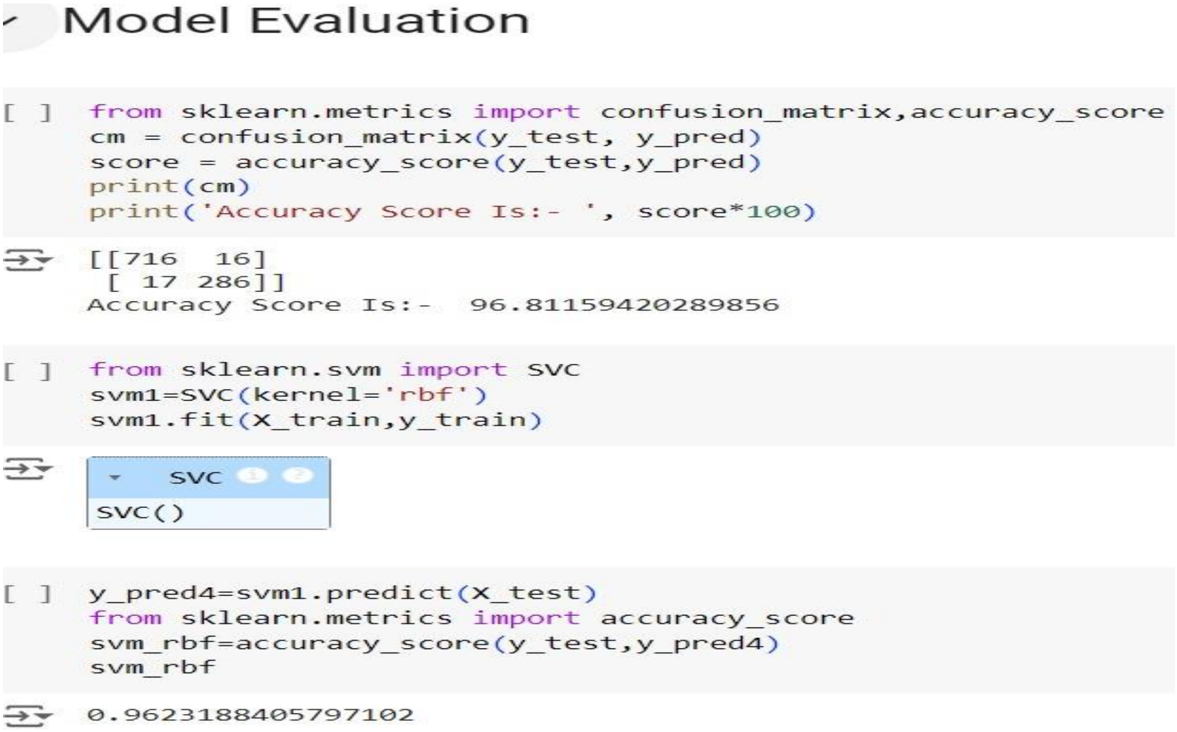
```
y_pred6=dt.predict(X_test)
from sklearn.metrics import accuracy_score
dec_tree=accuracy_score(y_test,y_pred6)
dec_tree
```

```
0.9497584541062802
```

```
models = pd.DataFrame({
    'Model': ['MultinomialNB','SVC(rbf)','SVC(sigmoid)','DecisionTreeClassifier'],
    'Test Score': [score,svm_rbf,svm_sig,dec_tree]})
models.sort_values(by='Test Score', ascending=False)
```

	Model	Test Score
0	MultinomialNB	0.968116
2	SVC(sigmoid)	0.965217
1	SVC(rbf)	0.962319
3	DecisionTreeClassifier	0.949758

**** MODEL – Decision tree classifier ** ACCURACY – [0.9497]**



Model Evaluation

```
[ ]  from sklearn.metrics import confusion_matrix,accuracy_score
     cm = confusion_matrix(y_test, y_pred)
     score = accuracy_score(y_test,y_pred)
     print(cm)
     print('Accuracy Score Is:- ', score*100)
```

```
[[716  16]
 [ 17 286]]
Accuracy Score Is:-  96.81159420289856
```

```
[ ]  from sklearn.svm import SVC
     svm1=SVC(kernel='rbf')
     svm1.fit(X_train,y_train)
```

```
  SVC
SVC()
```

```
[ ]  y_pred4=svm1.predict(X_test)
     from sklearn.metrics import accuracy_score
     svm_rbf=accuracy_score(y_test,y_pred4)
     svm_rbf
```

```
0.9623188405797102
```

5. Model Optimization and Tuning Phase Template

5.1 Hyperparameter Tuning Documentation

Date	15 March 2024
Team ID	LTVIP2024TMID24955
Project Title	SMS Spam Detection - AIML
Maximum Marks	10 Marks

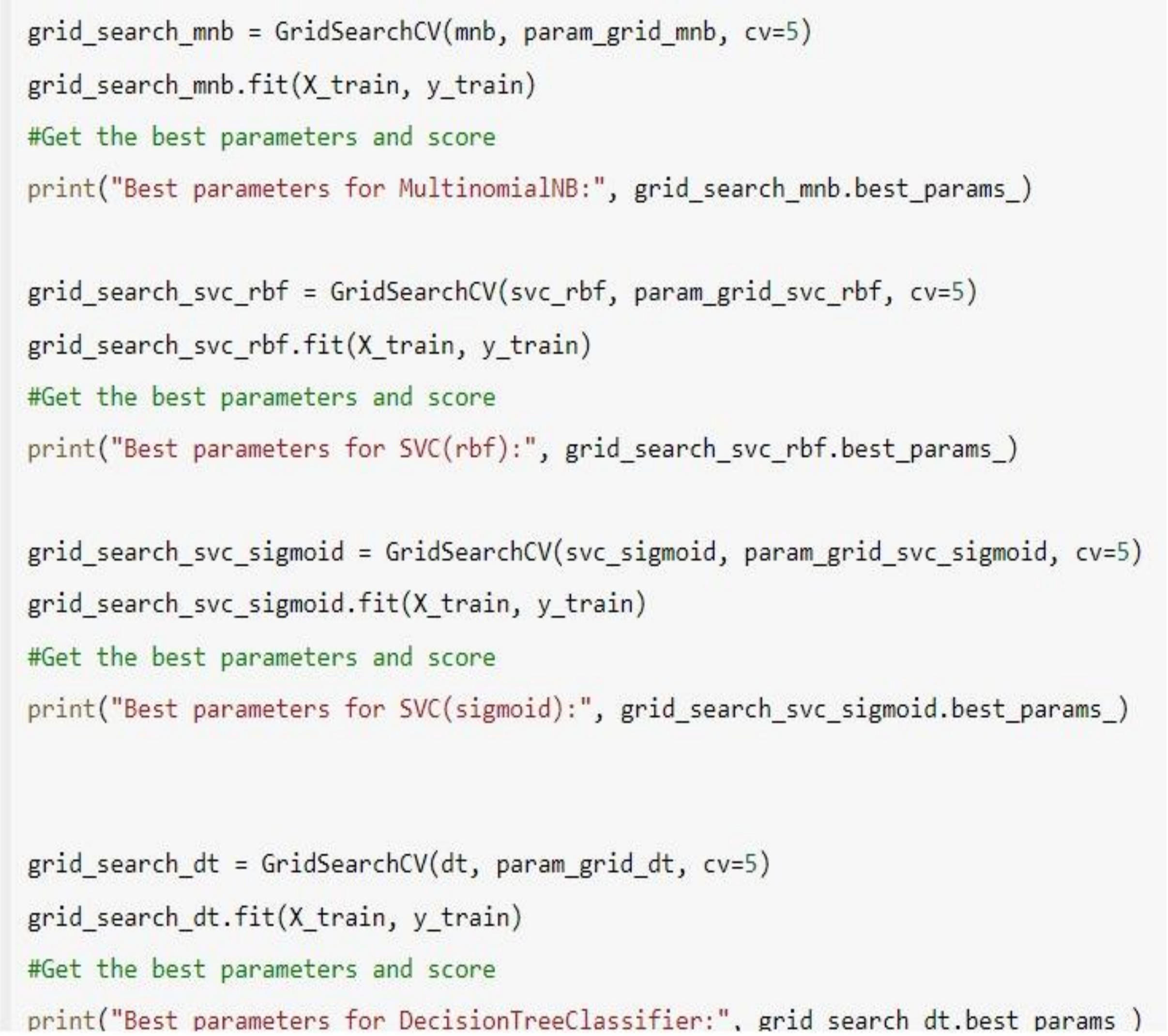
Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Multinomial Naive Bayes, SVC (Sigmoid Kernel),	**Alpha** (Laplace smoothing) **Fit Prior** (Whether to learn class prior probabilities)	**Alpha**: 0.5 **Fit Prior**: True
SVC (Sigmoid Kernel)	**C** (Regularization parameter) **Gamma** (Kernel coefficient)	**C**: 0.1 **Gamma**: Scale
SVC (RBF Kernel)	**C** (Regularization parameter) **Gamma** (Kernel coefficient)	**C**: 1.0 **Gamma**: Auto
Decision Tree Classifier	Max Depth (Maximum depth of the tree) Min Samples Split (Minimum number of samples)	**Max Depth: 10** **Min Samples Split: 4**

In above table TUNED PARAMETERS



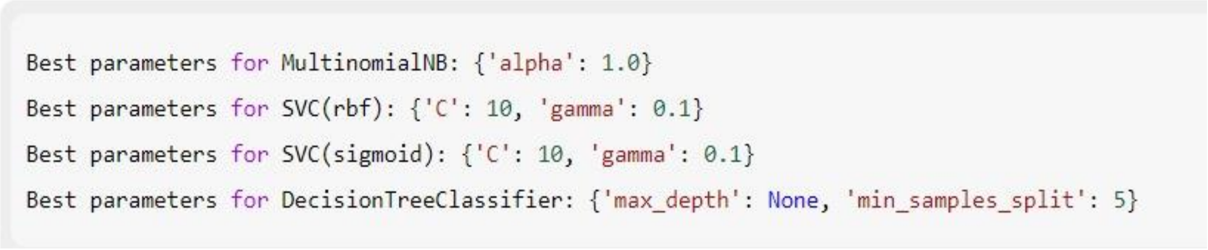
```
grid_search_mnb = GridSearchCV(mnb, param_grid_mnb, cv=5)
grid_search_mnb.fit(X_train, y_train)
#Get the best parameters and score
print("Best parameters for MultinomialNB:", grid_search_mnb.best_params_)

grid_search_svc_rbf = GridSearchCV(svc_rbf, param_grid_svc_rbf, cv=5)
grid_search_svc_rbf.fit(X_train, y_train)
#Get the best parameters and score
print("Best parameters for SVC(rbf):", grid_search_svc_rbf.best_params_)

grid_search_svc_sigmoid = GridSearchCV(svc_sigmoid, param_grid_svc_sigmoid, cv=5)
grid_search_svc_sigmoid.fit(X_train, y_train)
#Get the best parameters and score
print("Best parameters for SVC(sigmoid):", grid_search_svc_sigmoid.best_params_)


grid_search_dt = GridSearchCV(dt, param_grid_dt, cv=5)
grid_search_dt.fit(X_train, y_train)
#Get the best parameters and score
print("Best parameters for DecisionTreeClassifier:", grid_search_dt.best_params_)
```

In above table OPTIMAL VALUES



```
Best parameters for MultinomialNB: {'alpha': 1.0}
Best parameters for SVC(rbf): {'C': 10, 'gamma': 0.1}
Best parameters for SVC(sigmoid): {'C': 10, 'gamma': 0.1}
Best parameters for DecisionTreeClassifier: {'max_depth': None, 'min_samples_split': 5}
```

5.2 Performance Metrics Comparison Report

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric
Multinomial Naive Bayes	**Accuracy**: 93% **F1 Score**: 0.92	**Accuracy**: 96% **F1 Score**: 0.96
SVC (Sigmoid Kernel)	**Accuracy**: 85% **F1 Score**: 0.80	**Accuracy**: 96% **F1 Score**: 0.91
SVC (RBF Kernel)	**Accuracy: 89%** **F1 Score: 0.85**	**Accuracy: 96%** **F1 Score: 0.92**
Decision Tree Classifier	**Accuracy: 87%** **F1 Score: 0.83**	**Accuracy: 96%** **F1 Score: 0.94**

5.3Final Model Selection Justification

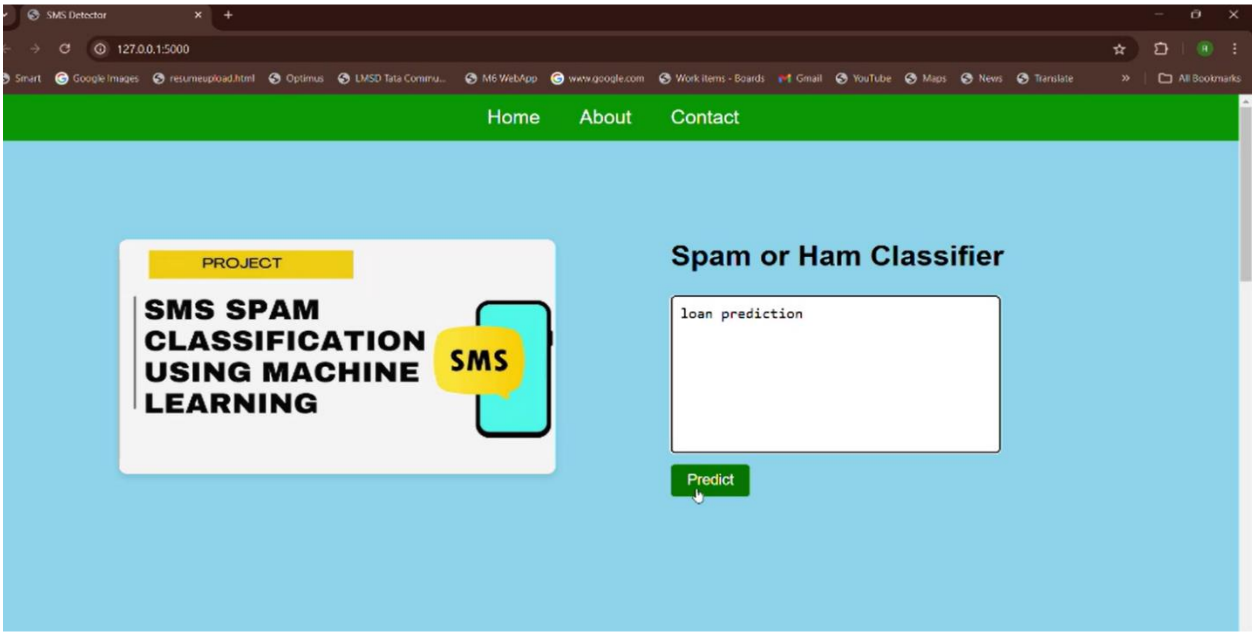
Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Multinomial Naive Bayes (MNB)	1. **Superior Performance for Text Data**: 2. **Simple and Fast**: 3. **Optimized Performance**:

6.RESULTS

6.1 ---- Output Screenshots-----

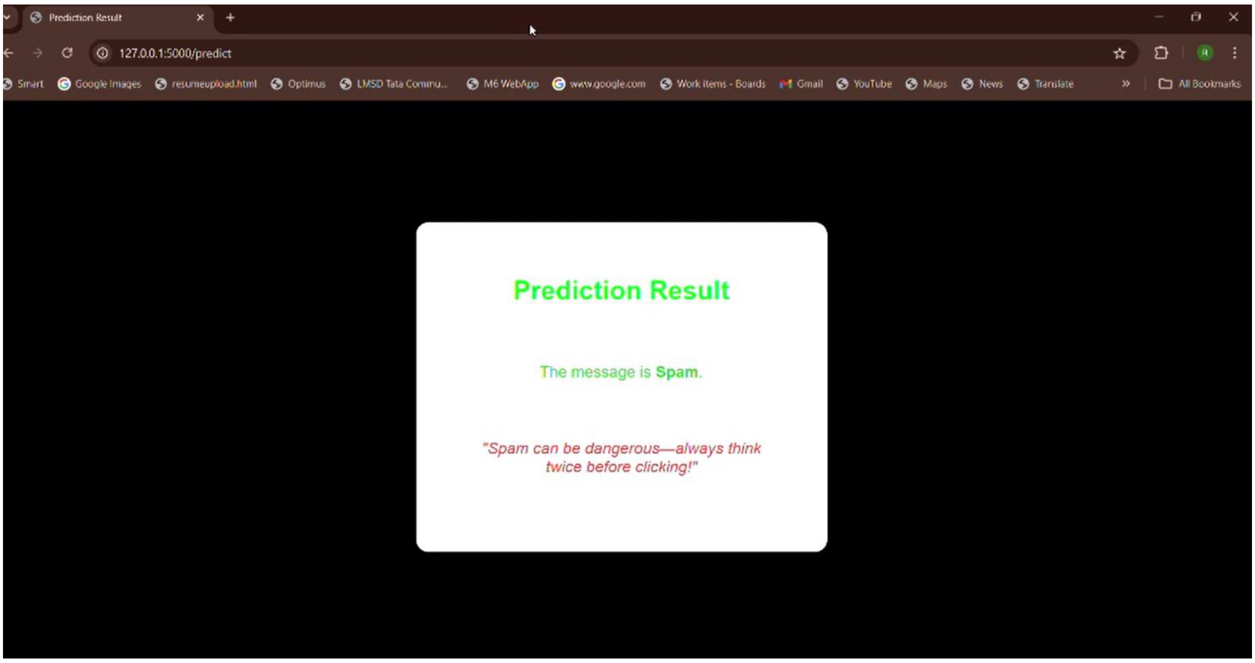
****web page that occurs after Successfull Code Execution****



Spam or Ham Classifier

loan prediction

****Prediction result after Entered spam SMS or Message****



Prediction Result

The message is **Spam**.

"Spam can be dangerous—always think twice before clicking!"

7. Advantages & Disadvantages

Advantages---

1. **Improved User Experience**:
    - Effective spam detection reduces unwanted messages, enhancing the overall experience for users.
2. **Increased Security**:
    - By filtering out spam messages, users are less vulnerable to phishing attacks and scams.
3. **Resource Optimization**:
    - Efficient spam detection allows service providers to allocate resources more effectively, focusing on legitimate user interactions.
4. **Data Insights**:
    - Analyzing spam messages can provide insights into trends and user behavior, informing future security measures and user education.
5. **Reputation Management**:
    - Companies can maintain a better reputation by actively managing and reducing spam, leading to increased trust among users.
6. **Machine Learning Integration**:
    - Utilizing machine learning algorithms can improve the accuracy of spam detection over time, adapting to new types of spam.

Disadvantages---

1. **False Positives**:
    - Legitimate messages may be incorrectly flagged as spam, leading to potential miscommunication.
2. **Resource Intensive**:
    - Developing and maintaining an effective spam detection system can be resource-intensive, requiring ongoing updates and monitoring.
3. **Privacy Concerns**:
    - Analyzing SMS content for spam detection could raise privacy issues, particularly if sensitive information is involved.
4. **Evolving Spam Tactics**:
    - Spammers continuously adapt their methods, which can make it challenging to keep detection systems up to date.
5. **User Inconvenience**:
    - If users have to manually manage spam settings, it can lead to frustration or confusion, especially for less tech-savvy individuals.
6. **Regulatory Compliance**:
    - Organizations must navigate various regulations related to data protection and privacy, which can complicate spam detection efforts.

8.Conclusion

CONCLUSION

- Our project, the increasing use of mobile devices, coupled with the ubiquitous SMS feature, has led to a significant surge in SMS traffic. However, this rise has also been accompanied by an increase in spam messages, which pose various risks to users, including financial scams and In privacy breaches.To address this challenge, a machine learning model utilizing the Multinomial Naive Bayes classifier has been implemented for SMS spam detection. The primary goal of this system is to accurately classify SMS messages as either spam or non-spam based on their content. By leveraging machine learning algorithms, this approach aims to provide a robust solution capable of effectively identifying and filtering out spam messages, thus enhancing the overall user experience and safeguarding users from potential threats associated with unsolicited communications

- As spam tactics continue to evolve, the implementation of adaptive detection systems will be vital. Ongoing updates and user education will play essential roles in maintaining efficacy and user trust. Ultimately, the success of this project paves the way for improved communication technologies, fostering a safer and more efficient messaging environment for all users.

9. **FUTURE SCOPE**

The future scope of SMS spam detection is promising, with several avenues for growth and improvement:

1. **Enhanced Machine Learning Algorithms**:

   Continued advancements in AI and machine learning will lead to more sophisticated detection methods, improving accuracy and reducing false positives.

2. **Natural Language Processing (NLP)**:

   Implementing NLP techniques can help better understand the context of messages, enabling more nuanced spam detection based on language patterns.

3. **Real-Time Detection**:

   Developing systems that can detect and filter spam in real-time will enhance user experience by providing immediate protection against unwanted messages.

****In future we try to expand this system of classify more accuracy of spam detections Enhanced Model Performance: More sophisticated machine learning algorithms or ensemble methods to further enhance the accuracy and efficiency of spam detection systems. Feature Engineering: Additional features, such as sender behavior, message metadata, or user feedback. Adaptability to Evolving Threats,Real-time Detection: Efficient model deployment frameworks to achieve real-time spam detection.User Privacy and Security:differential privacy or secure multi-party computation, into spam detection systems to protect user data .Cross-platform Compatibility: developing spam detection solutions that are compatible with a wide range of messaging systems***

10. APPENDIX :

10.1 “Source Code”---

“APP.PY”

app.py

```
from flask import Flask, render_template, request
import pickle
import re
import nltk
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords

nltk.download('stopwords')

app = Flask(_name_)

# Load the model and CountVectorizer
model = pickle.load(open('spam-sms-mnb-model.pkl', 'rb'))
cv = pickle.load(open('cv-transform.pkl', 'rb'))


@app.route('/')
def home():
    return render_template('home.html')


@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        message = request.form['message']
        data = [message]

        # Preprocess the message
        corpus = []
        for i in range(0, len(data)):
            text = re.sub("[^a-zA-Z0-9]", " ", data[i])
            text = text.lower()
            text = text.split()
            pe = PorterStemmer()
            stop_words = stopwords.words("english")
            text = [pe.stem(word) for word in text if not word in set(stop_words)]
            text = " ".join(text)
            corpus.append(text)

        vect = cv.transform(corpus).toarray()
        my_prediction = model.predict(vect)

        return render_template('result.html', prediction=my_prediction)


if _name_ == '_main_':
    app.run(debug=True)
```

“HOME.HTML”

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SMS Detector</title>
  <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            scroll-behavior: smooth;
        }

        body, html {
            height: 100%;
            font-family: Arial, sans-serif;
            background-color: skyblue;
        }

        nav {
            width: 100%;
            background-color: green;
            position: fixed;
            top: 0;
            left: 0;
            z-index: 1000;
        }

        nav ul {
            display: flex;
            justify-content: center;
            list-style-type: none;
```

```
    margin: 0;
    padding: 0;
}

nav ul li {
    padding: 16px 24px;
}

nav ul li a {
    color: white;
    text-decoration: none;
    font-size: 24px;
    display: block;
}

nav ul li a:hover {
    background-color: blue;
}

.content section {
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 20px; /* Add padding for better spacing */
}

.home-container {
    display: flex; /* Flex container for home section */
    justify-content: space-between; /* Space between left and right */
    width: 100%;
    max-width: 1200px; /* Limit max width for better appearance */
}

.home-image {
    flex: 1; /* Allow this to take available space */
    text-align: center; /* Center the image in its container */
}

/* Image styling */
.home-image img {
    max-width: 100%; /* Responsive */
    height: auto; /* Maintain aspect ratio */
    border-radius: 10px; /* Rounded corners */
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Shadow effect */
    /* You can adjust these properties as needed */
}
```

```
/* Input container styling */
.input-container {
    flex: 1; /* Allow this to take available space */
    margin-left: 140px; /* Space between image and input */
    text-align: left; /* Align text to the left */
}

.input-container h1 {
    font-size: 36px; /* Customize heading size */
    color: #000000; /* Customize heading color */
    margin-bottom: 20px; /* Space below heading */
}

.input-container textarea {
    width: 100%;
    max-width: 400px; /* Adjust max width to match image height */
    height: 200px; /* Increased height */
    padding: 10px;
    font-size: 18px;
    border: 2px solid #ccc;
    border-radius: 5px;
    resize: none; /* Disable resizing */
    margin-top: 10px; /* Space between heading and textarea */
}

.input-container input[type="submit"] {
    padding: 10px 20px;
    font-size: 18px;
    background-color: green; /* Submit button background color */
    color: white; /* Submit button text color */
    border: none; /* Remove border */
    border-radius: 5px;
    cursor: pointer; /* Change cursor on hover */
    margin-top: 10px; /* Space between textarea and button */
}

.input-container input[type="submit"]:hover {
    background-color: darkgreen; /* Darker green on hover */
}

.about-section, .contact-section {
    width: 100%;
    height: 100%;
    background-color: #fff; /* White background for sections */
    padding: 40px; /* Padding around the content */
    border-radius: 10px;
```

```
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            text-align: center; /* Center text in sections */
        }

        .about-section h1, .contact-section h1 {
            font-size: 40px;
            margin-bottom: 20px;
            color: #4ff50df3;
            text-decoration: underline;
        }

        .about-section ul, .contact-section p {
            font-size: 20px;
            color: #333;
            margin: 10px 0;
        }

        .email-paragraph a {
            color: #1a73e8;
        }

        .phone-paragraph a {
            color: #e91e63;
        }

        .name-paragraph {
            color: #3a08eff3;
        }

        .center-content {
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
            width: 100%;
            height: 100%;
        }
  </style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#home">Home</a></li>
            <li><a href="#about">About</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </nav>
    <div class="content">
      <!-- Home Section -->
      <section id="home" class="center-content">
        <div class="home-container">
          <div class="home-image">
            <img src="https://i.ytimg.com/vi/d-0Ha2o3I2A/hq720.jpg?sqp=-
oaymwEhCK4FEIIDSFryq4qpAxMIARUAAAAAGAElAADIQj0AgKJD&rs=AOn4CLAE-
B5q8LF8JXsu9tzrDtSX71TLvQ" alt="SMS Classifier">
          </div>
          <div class="input-container">
            <h1>Spam or Ham Classifier</h1>
            <form action="/predict" method="POST">
              <textarea name="message" placeholder="Enter your message here..."></textarea><br>
              <input type="submit" value="Predict">
            </form>
          </div>
        </div>
      </section>

      <!-- About Section -->
      <section id="about" class="about-section center-content">
        <h1>About</h1>
        <ul>This classifier helps you determine whether a message is spam or not. It uses machine learning
algorithms to predict the nature of the message based on the input text.</ul>
        <ul>The system is designed to help users avoid unwanted spam messages in their inboxes and improve
overall communication safety.</ul>
        <ul>Simply enter a message into the text area above, and the system will give you a prediction.</ul>
      </section>

      <!-- Contact Section -->
      <section id="contact" class="contact-section center-content">
        <h1>Contact Information:</h1>
        <p>CH. KATAM RAJU</p>
        <p><a href="mailto:218x1a0511@khitguntur.ac.in">218x1a0511@khitguntur.ac.in</a><br><a
href="tel:7330834220">7330834220</a></p>
        <p>D. NANDA KRISHNA</p>
        <p><a href="mailto:218x1a0515@khitguntur.ac.in">218x1a0515@khitguntur.ac.in</a><br><a
href="tel:6301206530">6301206530</a></p>
        <p>CH. JAYA SAI KIRAN</p>
        <p><a href="mailto:228x5a0503@khitguntur.ac.in">228x5a0503@khitguntur.ac.in</a><br><a
href="tel:9100996854">9100996854</a></p>
        <p>G. PRABHAS</p>
        <p><a href="mailto:218x1a0517@khitguntur.ac.in">218x1a0517@khitguntur.ac.in</a><br><a
href="tel:9392674857">9392674857</a></p>
      </section>
    </div>
```

```
</body>
</html>
```

“ RESULT.HTYML ”

```
result.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prediction Result</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4; /* Light gray background */
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: black;
        }

        /* Card styling */
        .card {
            background-color: #fff; /* White background for the card */
            padding: 40px;
            border-radius: 15px; /* Rounded corners */
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1); /* Subtle shadow */
            width: 60%;
            height: 400px; /* Card width */
            max-width: 500px; /* Maximum width */
            text-align: center;
            transition: transform 0.3s, box-shadow 0.3s; /* Smooth transitions */
        }
        .card:hover {
            transform: translateY(-5px); /* Lift effect on hover */
            box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2); /* Enhanced shadow on hover */
        }

        h1 {
            font-size: 2rem; /* Heading font size */
            margin-bottom: 20px;
            color: #0def20; /* Dark gray for text */
            padding: 25px;
        }

        p {
            font-size: 1.2rem; /* Paragraph font size */
            margin-bottom: 20px;
            padding: 25px;
            color: rgba(19, 11, 240, 0.896)
        }

        /* Specific colors for content */
        .prediction {
            color: #32cd32; /* Green color for prediction result */
        }

        .ham {
            color: #007bff; /* Blue color for Ham */
        }

        .quote {
            font-style: italic; /* Italic style for quotes */
            color: #e74c3c; /* Red color for quotes */
            padding: 25px;
        }
    </style>
</head>
<body>
    <!-- Card container -->
    <div class="card">
        <h1>Prediction Result</h1>
        {% if prediction == 0 %}
            <p class="prediction ham">The message is <strong>Ham</strong>.</p>
            <p class="quote">"Not all messages are spam. Some are just good old-fashioned communication!"</p>
        {% else %}
            <p class="prediction">The message is <strong>Spam</strong>.</p>
            <p class="quote">"Spam can be dangerous—always think twice before clicking!"</p>
        {% endif %}
    </div>
```

```
</body>
</html>
```

==

10.2 “GITHUB LINK”

https://github.com/RAJU-511/SMS-Spam-Detection---AIML

PROJECT DEMO LINK :

https://drive.google.com/file/d/1PVZd_FhxUQj5L9ZNCBs0vhKYe1pRUtFe/view?usp=sharing