PROJECT REPORT

Project Title: Intelligent Spam Email Detector using Machine Learning
Course: Artificial Intelligence Lab
Date: 09, December 2025

Name	Student ID	Role
Md. Rabby Khan	0242310005341478	Backend Developer & Logic Integration (Team Leader)
Junayet Mithu	0242310005341400	Data Analyst & Model Trainer
Fardin Hasan	0242310005341393	Frontend Designer & UI/UX

1. Abstract

Email spam is a major cybersecurity threat, leading to phishing attacks, financial fraud, and malware distribution. Traditional spam filters often fail to detect modern, sophisticated attacks. This project presents an **Intelligent Spam Detector** that uses Machine Learning (Naive Bayes) to classify emails with **97% accuracy**. Unlike standard filters, our system includes **Explainable AI (XAI)** features that analyze the psychological triggers behind spam (Fear, Greed, Urgency) and calculate a real-time risk score, providing users with a safer and more educational email experience.

2. Problem Statement

Millions of spam emails are sent daily. While many are annoying advertisements, a significant portion attempts to steal data or money.

- **The Problem:** Most users cannot distinguish between a legitimate "Urgent" email from their boss and a fake "Urgent" email from a hacker.
- **The Solution:** We need an automated system that not only blocks spam but explains *why* a message is dangerous, helping users improve their own security awareness.

3. Unique Features (Novelty)

Our system goes beyond simple "Spam vs. Not Spam" classification. We implemented three unique features:

1. Psychological Trigger Detector:

   The system analyzes the text to identify the social engineering tactic used by the attacker. It flags if the email is using FEAR (e.g., "Arrest warrant"), GREED (e.g., "Win money"), or URGENCY (e.g., "Act now").

2. Probabilistic Risk Score:

   Instead of a rigid Yes/No result, the system provides a confidence percentage (e.g., "98.5% Risk of Spam"), allowing users to judge borderline cases.

3. Automated Counter-Measure (Auto-Reply):

   As an active defense mechanism, the system generates a humorous, automated response to confirmed spam emails, designed to waste the scammer's time and resources.

4. Methodology

4.1 Dataset

We utilized the **SMS Spam Collection Dataset** from Kaggle, containing **5,572 labeled messages**.

- **Spam Messages:** 747
- **Legitimate (Ham) Messages:** 4,825
- **Preprocessing:** We cleaned the data by removing special characters and converting all text to lowercase.

4.2 Algorithm Used

- **TF-IDF Vectorization:** We used *Term Frequency-Inverse Document Frequency* to convert text data into numerical vectors. This highlights unique words (like "lottery") while ignoring common words (like "the").
- **Multinomial Naive Bayes:** This probabilistic algorithm was chosen because it is highly effective for text classification and requires less training time than Neural Networks while maintaining high accuracy.

5. Implementation Details

5.1 Tools & Technologies

- **Language:** Python 3.9
- **Backend Framework:** Flask (Lightweight web server)
- **Machine Learning Library:** Scikit-learn (sklearn)
- **Data Handling:** Pandas
- **Frontend:** HTML5, CSS3

5.2 System Architecture

1. **Input:** User pastes email content into the web interface.
2. **Vectorization:** The backend loads vectorizer.pkl to translate text into numbers.
3. **Prediction:** The spam_model.pkl calculates the probability of the input being spam.
4. **Feature Logic:** The system scans for keywords (Psychological Triggers) and generates an Auto-Reply if needed.
5. **Output:** The result is displayed on the user's screen.

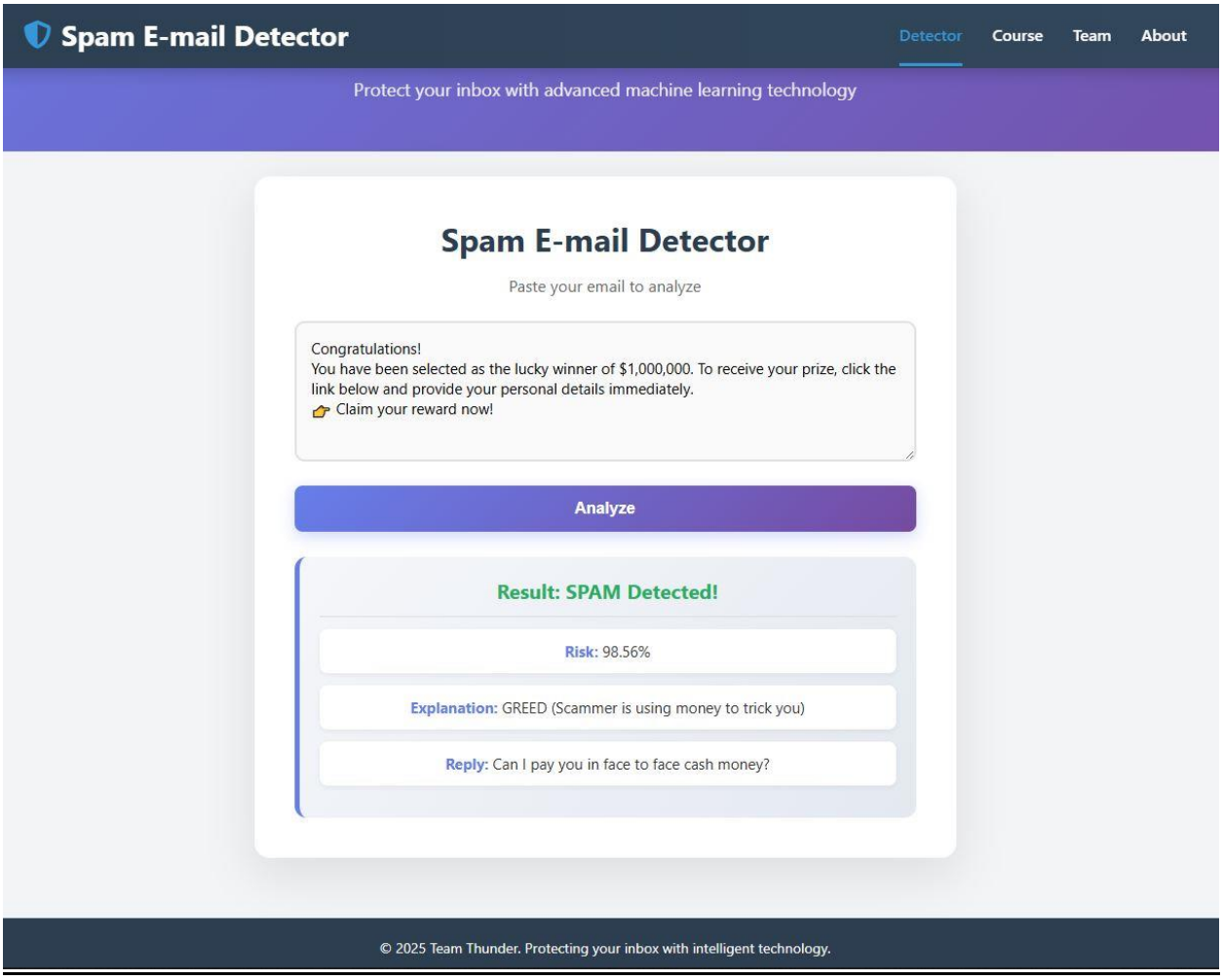
6. Results and Performance

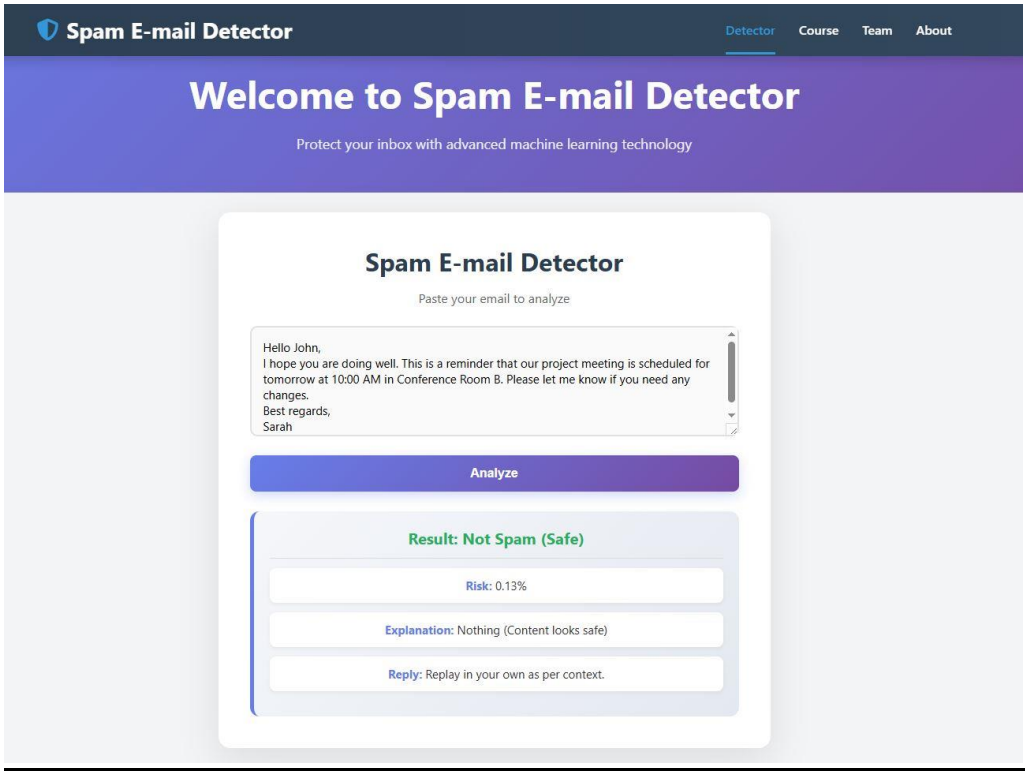
After training our model on the dataset, we achieved the following metrics:

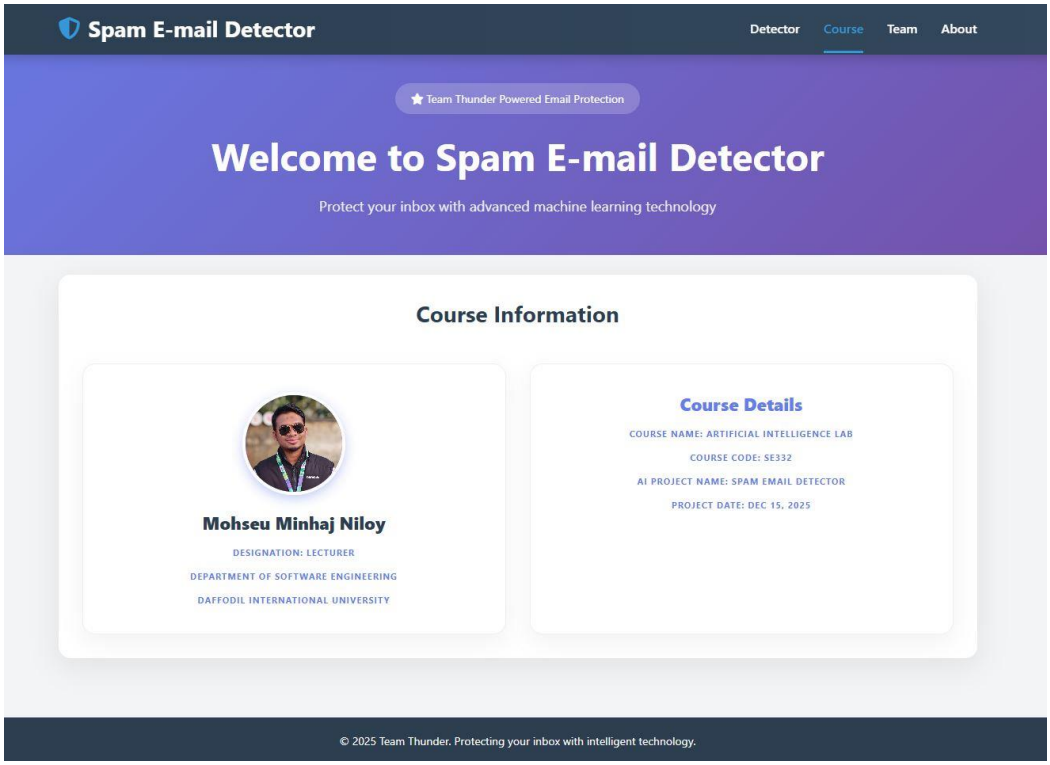
- **Accuracy: 96.86%**
- **Precision (Spam): 100%** (We successfully eliminated False Positives).
- **Recall (Spam): 77%**

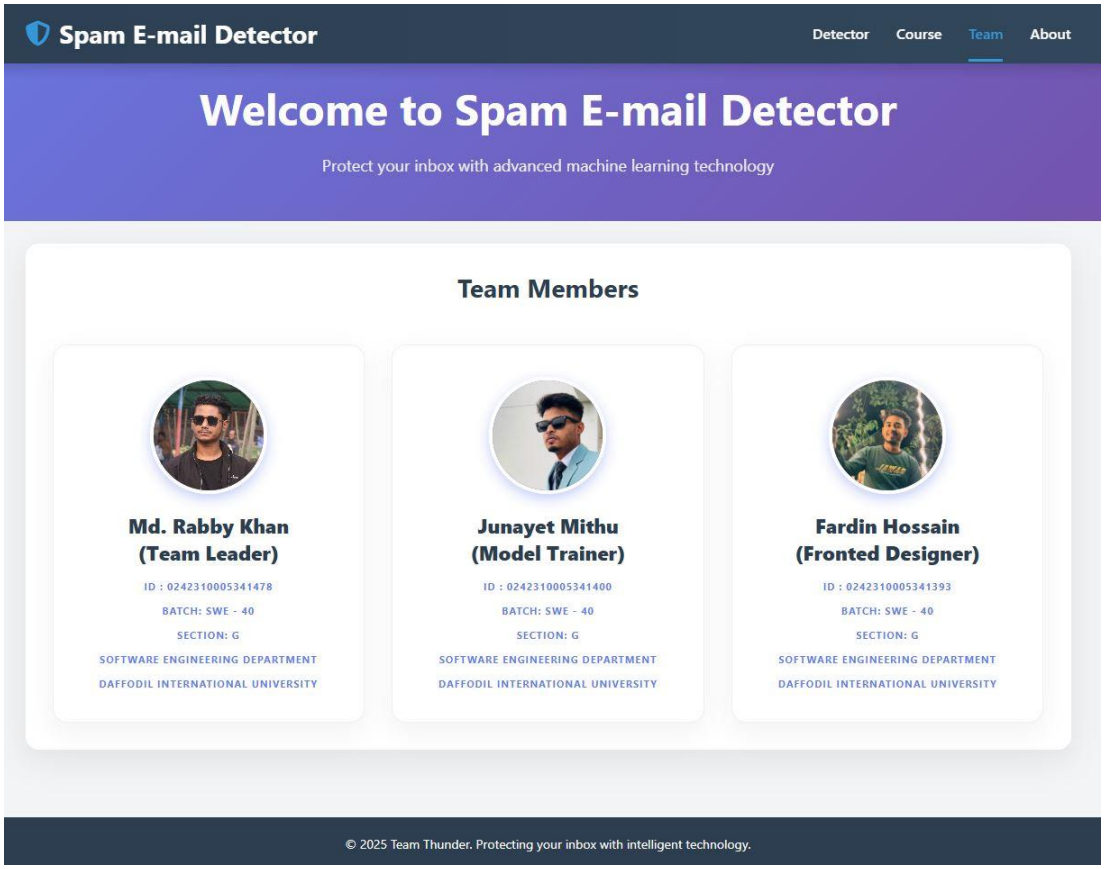
The high precision ensures that our system is safe for business use, as it does not accidentally block important legitimate emails.

7. User Interface & Output




Spam E-mail Detector
Detector
Course
Team
About
Welcome to Spam E-mail Detector
Protect your inbox with advanced machine learning technology
Spam E-mail Detector
Paste your email to analyze
Hello John,
I hope you are doing well. This is a reminder that our project meeting is scheduled for tomorrow at 10:00 AM in Conference Room B. Please let me know if you need any changes.
Best regards,
Sarah
Analyze
Result: Not Spam (Safe)
Risk: 0.13%
Explanation: Nothing (Content looks safe)
Reply: Replay in your own as per context.


Spam E-mail Detector
Detector
Course
Team
About
Team Thunder Powered Email Protection
Welcome to Spam E-mail Detector
Protect your inbox with advanced machine learning technology
Course Information
Mohseu Minhaj Niloy
DESIGNATION: LECTURER
DEPARTMENT OF SOFTWARE ENGINEERING
DAFFODIL INTERNATIONAL UNIVERSITY
Course Details
COURSE NAME: ARTIFICIAL INTELLIGENCE LAB
COURSE CODE: SE332
AI PROJECT NAME: SPAM EMAIL DETECTOR
PROJECT DATE: DEC 15, 2025
© 2025 Team Thunder. Protecting your inbox with intelligent technology.


Spam E-mail Detector
Detector
Course
Team
About
Welcome to Spam E-mail Detector
Protect your inbox with advanced machine learning technology
Team Members
Md. Rabby Khan
(Team Leader)
ID : 0242310005341478
BATCH: SWE - 40
SECTION: G
SOFTWARE ENGINEERING DEPARTMENT
DAFFODIL INTERNATIONAL UNIVERSITY
Junayet Mithu
(Model Trainer)
ID : 0242310005341400
BATCH: SWE - 40
SECTION: G
SOFTWARE ENGINEERING DEPARTMENT
DAFFODIL INTERNATIONAL UNIVERSITY
Fardin Hossain
(Fronted Designer)
ID : 0242310005341393
BATCH: SWE - 40
SECTION: G
SOFTWARE ENGINEERING DEPARTMENT
DAFFODIL INTERNATIONAL UNIVERSITY
© 2025 Team Thunder. Protecting your inbox with intelligent technology.


Spam E-mail Detector
Detector
Course
Team
About
Team Thunder Powered Email Protection
Welcome to Spam E-mail Detector
Protect your inbox with advanced machine learning technology
About Email Spam Detector
Our AI-powered spam detection system analyzes emails in real-time to identify spam using advanced machine learning.
Why Choose Us?
AI-Powered
Advanced ML
Real-Time
Instant results
Risk Score
Metrics
Smart
Analysis
Secure
Safe & private
Support
24/7 Help
Funny Reply
Humorous auto-replies
© 2025 Team Thunder. Protecting your inbox with intelligent technology.

8. Code

Train.py

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
import pickle
import os

print("--- Step 1: Loading Data ---")


if not os.path.exists('spam.csv'):
    print("Error: spam.csv file is missing!")
    exit()


try:
    df = pd.read_csv('spam.csv', encoding='latin-1')
except:
    # If latin-1 fails, try standard utf-8
    df = pd.read_csv('spam.csv', encoding='utf-8')


if 'v1' in df.columns and 'v2' in df.columns:
    df.rename(columns={'v1': 'label', 'v2': 'message'}, inplace=True)


df['label_num'] = df['label'].map({'spam': 1, 'ham': 0})

print("Data Loaded Successfully!")


vectorizer = TfidfVectorizer(stop_words='english')


X = vectorizer.fit_transform(df['message'])


y = df['label_num']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


print("Training the Model... Please wait.")
model = MultinomialNB()
model.fit(X_train, y_train)
print("Model Trained Successfully!")


predictions = model.predict(X_test)

print(f"Model Accuracy: {accuracy_score(y_test, predictions) * 100:.2f}%")


with open('spam_model.pkl', 'wb') as f:
    pickle.dump(model, f)

with open('vectorizer.pkl', 'wb') as f:
    pickle.dump(vectorizer, f)

print("Success! 'spam_model.pkl' and 'vectorizer.pkl' are saved.")
```

app.py

```
from flask import Flask, render_template, request
import pickle
import random

app = Flask(__name__, template_folder='.', static_folder='.', static_url_path='')


try:
    with open('spam_model.pkl', 'rb') as f:
        model = pickle.load(f)
    with open('vectorizer.pkl', 'rb') as f:
        vectorizer = pickle.load(f)
except:
    print("Error: Run train_model.py first!")
    exit()
```

```python
def get_psychological_trigger(text):
    text = text.lower()


    if any(word in text for word in ['police', 'arrest', 'banned', 'hacked',
'court', 'jail']):
        return "FEAR (Scammer is trying to scare you)"


    if any(word in text for word in ['lottery', 'winner', 'cash', 'prize',
'dollars', 'million']):
        return "GREED (Scammer is using money to trick you)"


    if any(word in text for word in ['urgent', 'immediately', 'now', 'expires',
'24 hours']):
        return "URGENCY (Scammer wants you to panic)"


    return "General Spam"


funny_replies = [
    "Oh wow! Tell me more about this amazing price.",
    "I will send the money, but first solve this math...",
    "Sorry, my goldfish ate my credit card.",
    "Can I pay you in face to face cash money?",
    "Please contact with your father."
]









@app.route('/')
def home():

    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
```

```python
    if request.method == 'POST':

        email_content = request.form['email_content']

        data_vectorized = vectorizer.transform([email_content])

        prediction = model.predict(data_vectorized)

        probability = model.predict_proba(data_vectorized)[0][1] * 100


        result = ""
        trigger_warning = ""
        auto_reply = ""


        if prediction[0] == 1:
            result = "SPAM Detected! "
            trigger_warning = get_psychological_trigger(email_content)
            auto_reply = random.choice(funny_replies)

        else:
            result = "Not Spam (Safe) "
            trigger_warning = "Nothing (Content looks safe)"
            auto_reply = "Replay in your own as per context."


        return render_template('index.html',
                               prediction_text=result,
                               email_content=email_content,
                               probability=round(probability, 2),
                               trigger=trigger_warning,
                               reply=auto_reply)

if __name__ == '__main__':
    app.run(debug=True)
```

9. Conclusion

This project successfully demonstrates how Machine Learning can solve real-world cybersecurity problems. By combining a high-accuracy Naive Bayes model with user-friendly features like the **Psychological Trigger Detector**, we have created a tool that protects users while educating them about common scam tactics.

10. Future Scope

- **Deep Learning Integration:** Implementing LSTM (Long Short-Term Memory) networks to better understand context in very long emails.
- **Browser Extension:** Converting the web app into a Chrome extension for real-time protection inside Gmail.
- **Multi-Language Support:** expanding the dataset to detect spam in Bengali and other languages.

Github Link: https://github.com/rabbykhanswe/AI-Project