SMS Spam Detection System - Quick Start Guide

Introduction

Welcome to the SMS Spam Detection System! This guide will walk you through the steps to install, set up, and use the system to detect spam text messages. The system uses machine learning to identify potentially unwanted messages with high accuracy.

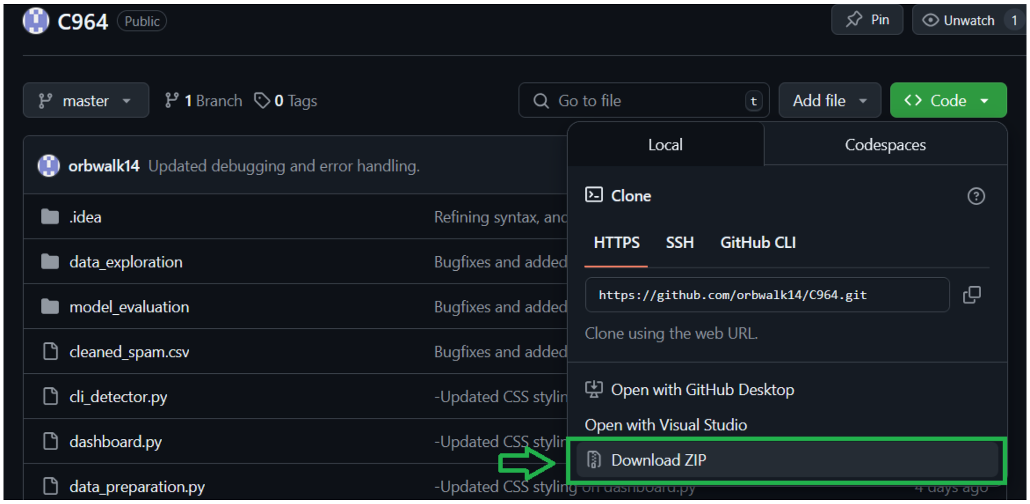
Requirements

- Minimum 4GB RAM
- 500MB of disk space
- Operating System: Windows, macOS, or Linux
- Python: https://www.python.org/downloads/
- Pip: https://packaging.python.org/en/latest/tutorials/installing-packages/
- Project ZIP file: C964

Installation:

To start you will need the program files, download the attached zip file titled 'C964' to a directory of your choosing.

The ZIP file can also be accessed via GitHub at the address: https://github.com/orbwalk14/C964, find and click the green button labeled 'code' and select 'Download ZIP'.



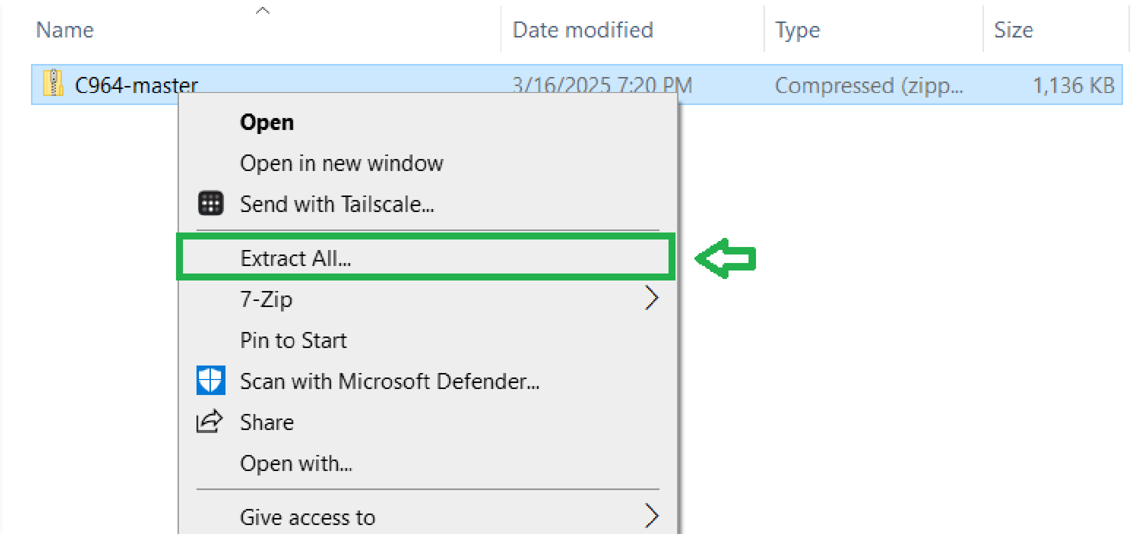
orbwalk14 Updated debugging and error handling.

.idea Refining syntax, anc
data_exploration Bugfixes and added
model_evaluation Bugfixes and added
cleaned_spam.csv Bugfixes and added
cli_detector.py -Updated CSS stylin
dashboard.py -Updated CSS stylin
data_preparation.py -Updated CSS styling on dashboard.py 4 days ago

https://github.com/orbwalk14/C964.git

Clone using the web URL.

Once you have downloaded your zip file, open your file explorer and find where you have downloaded your file. Right click the file then select 'Extract All…' and follow the prompt to extract the file to your desired location (IMPORTANT: Please make a note of the path to your files).

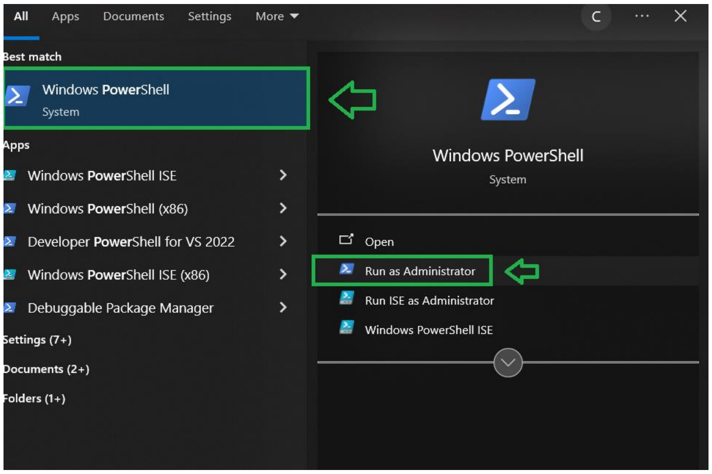


Name	Date modified	Type	Size
C964-master	3/16/2025 7:20 PM	Compressed (zipp...	1,136 KB

You now have the required files to showcase the technology! For the next section we will be running the program, for this guide I will be using the Windows operating system but will include notes on how the instructions differ from other operating systems.

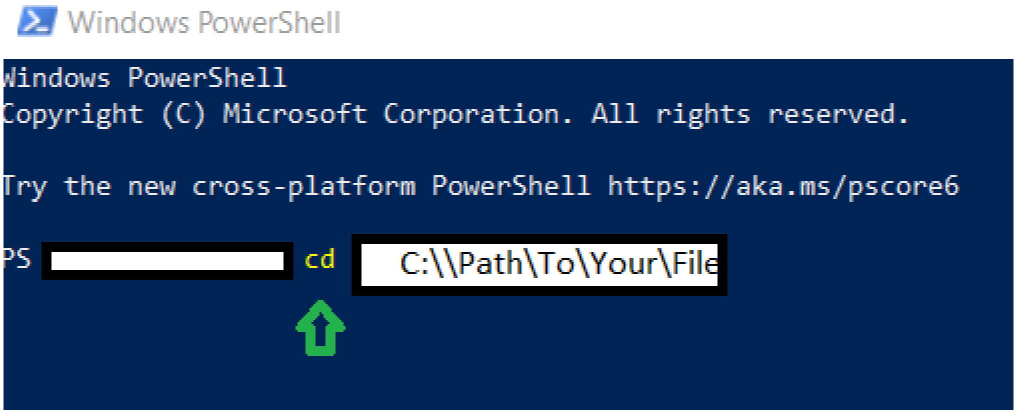
Preparing your environment:

To begin the process of preparing your environment, open a PowerShell window with administrator privileges.



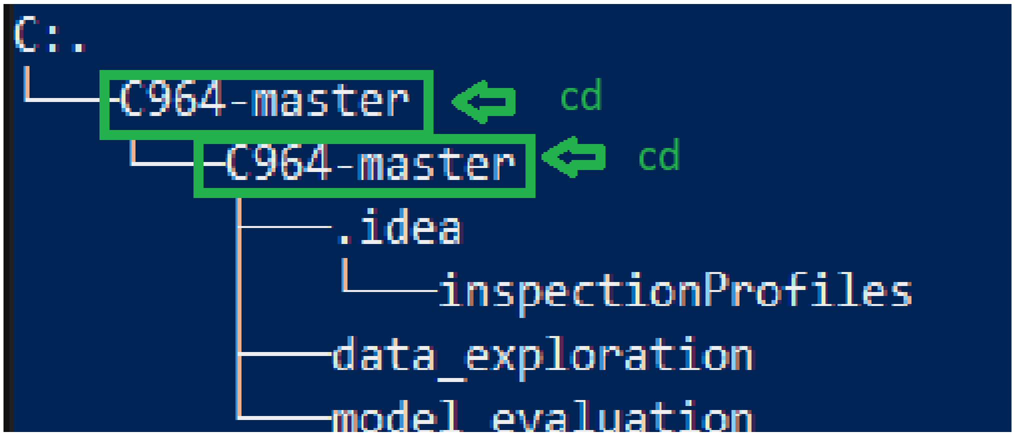
(Open the Terminal application in Linux or MacOS, ensure you are logged in as a user with administrative privileges).

Use the command ‘cd’ (change directory), to navigate to the path of where you unzipped your project files.

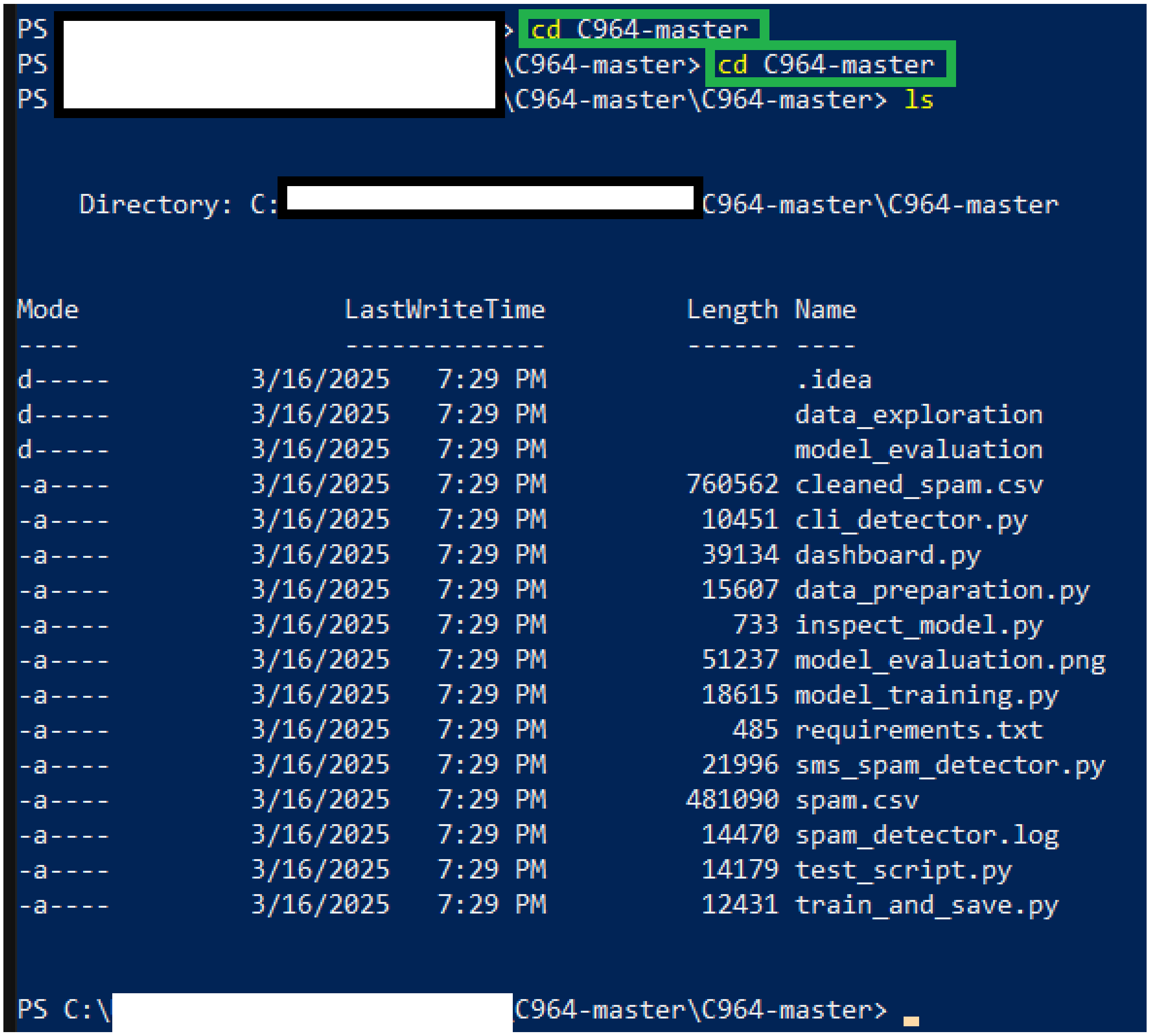


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS [redacted] cd C:\\Path\To\Your\File
```

You are now at the top-level directory of the program files; to prepare the environment to run the program, we must go deeper into the files. Use the ‘cd’ command again to navigate down two levels to get to where the python files are.

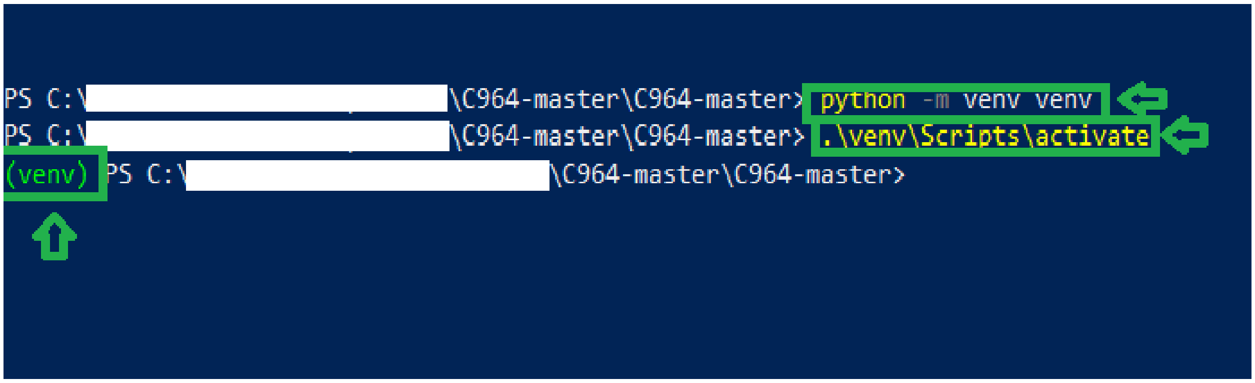


```
C:.
└───C964-master
    └───C964-master
        ├───.idea
        │   └───inspectionProfiles
        ├───data_exploration
        └───model_evaluation
```



```
PS [redacted]> cd C964-master
PS \C964-master> cd C964-master
PS \C964-master\C964-master> ls


    Directory: C:[redacted]C964-master\C964-master


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         3/16/2025   7:29 PM                .idea
d-----         3/16/2025   7:29 PM                data_exploration
d-----         3/16/2025   7:29 PM                model_evaluation
-a----         3/16/2025   7:29 PM         760562 cleaned_spam.csv
-a----         3/16/2025   7:29 PM          10451 cli_detector.py
-a----         3/16/2025   7:29 PM          39134 dashboard.py
-a----         3/16/2025   7:29 PM          15607 data_preparation.py
-a----         3/16/2025   7:29 PM            733 inspect_model.py
-a----         3/16/2025   7:29 PM          51237 model_evaluation.png
-a----         3/16/2025   7:29 PM          18615 model_training.py
-a----         3/16/2025   7:29 PM            485 requirements.txt
-a----         3/16/2025   7:29 PM          21996 sms_spam_detector.py
-a----         3/16/2025   7:29 PM         481090 spam.csv
-a----         3/16/2025   7:29 PM          14470 spam_detector.log
-a----         3/16/2025   7:29 PM          14179 test_script.py
-a----         3/16/2025   7:29 PM          12431 train_and_save.py


PS C:\[redacted]C964-master\C964-master>
```

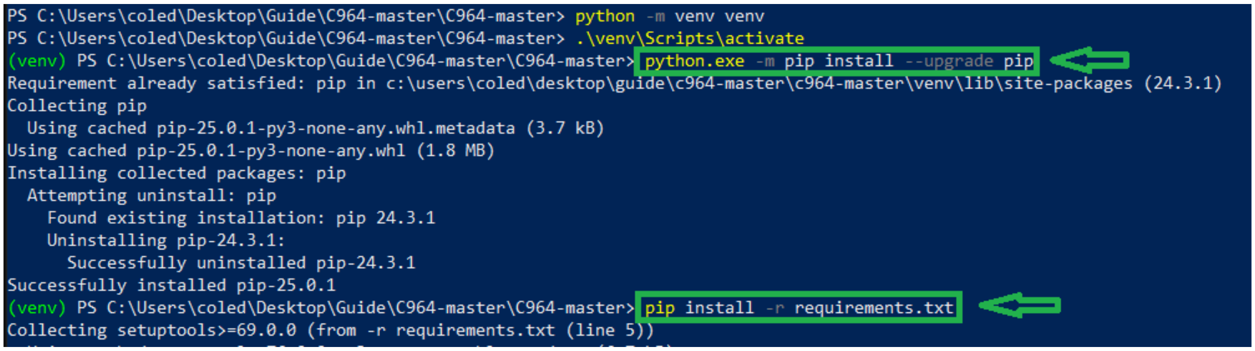
Next, we must create and activate a virtual environment for our program to run in, think of this like preparing a stage for our program to perform on.

1. python –m venv venv
2. .\venv\Scripts\activate
3. On macOS/Linux: source venv/bin/activate
4. Ensure you see the green '(venv)' tag on your terminal, this signifies you have correctly setup your virtual environment.
5. 

```
PS C:\[redacted]\C964-master\C964-master> python -m venv venv
PS C:\[redacted]\C964-master\C964-master> .\venv\Scripts\activate
(venv) PS C:\[redacted]\C964-master\C964-master>
```

The last step for preparing the environment is making sure the program has all packages it requires downloaded; this requires you to run two commands. Be sure to type the commands exactly as they are presented here.

1. Ensure Pip is up to date: python.exe -m pip install --upgrade pip
2. Download requirements.txt: pip install -r requirements.txt



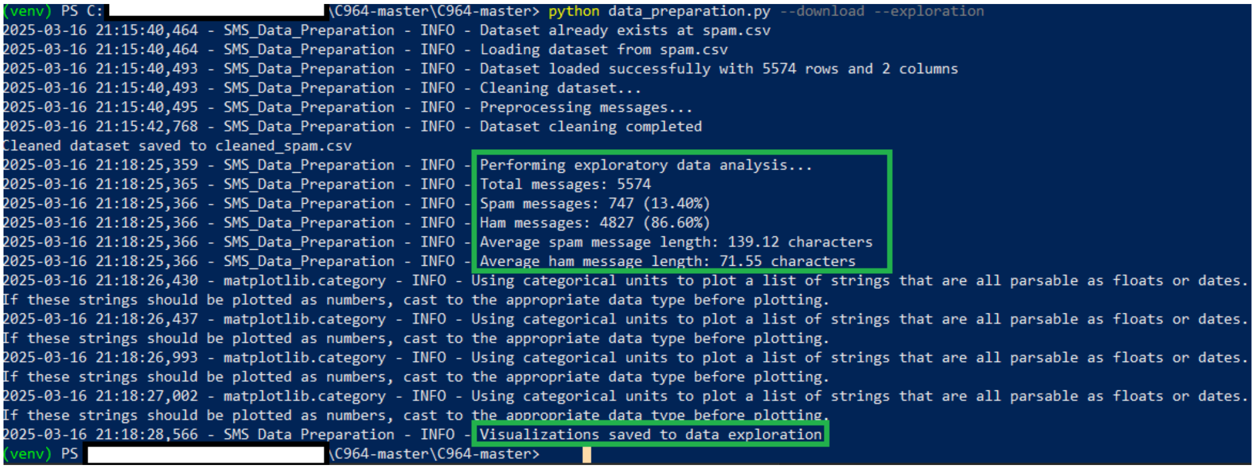
```
PS C:\Users\coled\Desktop\Guide\C964-master\C964-master> python -m venv venv
PS C:\Users\coled\Desktop\Guide\C964-master\C964-master> .\venv\Scripts\activate
(venv) PS C:\Users\coled\Desktop\Guide\C964-master\C964-master> python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\coled\desktop\guide\c964-master\c964-master\venv\lib\site-packages (24.3.1)
Collecting pip
  Using cached pip-25.0.1-py3-none-any.whl.metadata (3.7 kB)
Using cached pip-25.0.1-py3-none-any.whl (1.8 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.3.1
    Uninstalling pip-24.3.1:
      Successfully uninstalled pip-24.3.1
Successfully installed pip-25.0.1
(venv) PS C:\Users\coled\Desktop\Guide\C964-master\C964-master> pip install -r requirements.txt
Collecting setuptools>=69.0.0 (from -r requirements.txt (line 5))
```

You have now successfully set the stage for the program to run efficiently! In the next section make sure to follow the commands exactly to ensure that your program runs correctly.

Running the Program:

The first step of running the program entails preparing and analyzing the data to be fed into the model, to accomplish this run the following command in your terminal window:

1. Download and process the dataset:
   python data_preparation.py --download –exploration
2. You will be provided with terminal notifications explaining to you what is currently happening in the program.



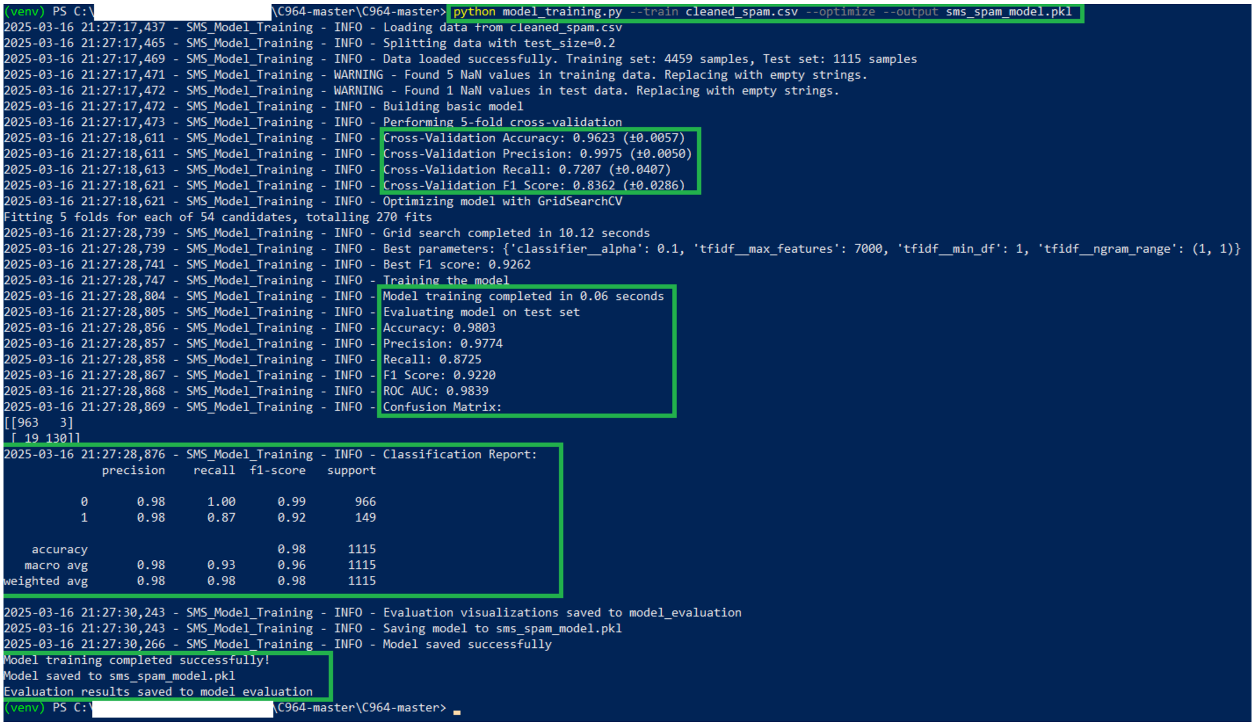
```
(venv) PS C:[redacted]\C964-master\C964-master> python data_preparation.py --download --exploration
2025-03-16 21:15:40,464 - SMS_Data_Preparation - INFO - Dataset already exists at spam.csv
2025-03-16 21:15:40,464 - SMS_Data_Preparation - INFO - Loading dataset from spam.csv
2025-03-16 21:15:40,493 - SMS_Data_Preparation - INFO - Dataset loaded successfully with 5574 rows and 2 columns
2025-03-16 21:15:40,493 - SMS_Data_Preparation - INFO - Cleaning dataset...
2025-03-16 21:15:40,495 - SMS_Data_Preparation - INFO - Preprocessing messages...
2025-03-16 21:15:42,768 - SMS_Data_Preparation - INFO - Dataset cleaning completed
Cleaned dataset saved to cleaned_spam.csv
2025-03-16 21:18:25,359 - SMS_Data_Preparation - INFO - Performing exploratory data analysis...
2025-03-16 21:18:25,365 - SMS_Data_Preparation - INFO - Total messages: 5574
2025-03-16 21:18:25,366 - SMS_Data_Preparation - INFO - Spam messages: 747 (13.40%)
2025-03-16 21:18:25,366 - SMS_Data_Preparation - INFO - Ham messages: 4827 (86.60%)
2025-03-16 21:18:25,366 - SMS_Data_Preparation - INFO - Average spam message length: 139.12 characters
2025-03-16 21:18:25,366 - SMS_Data_Preparation - INFO - Average ham message length: 71.55 characters
2025-03-16 21:18:26,430 - matplotlib.category - INFO - Using categorical units to plot a list of strings that are all parsable as floats or dates.
If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
2025-03-16 21:18:26,437 - matplotlib.category - INFO - Using categorical units to plot a list of strings that are all parsable as floats or dates.
If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
2025-03-16 21:18:26,993 - matplotlib.category - INFO - Using categorical units to plot a list of strings that are all parsable as floats or dates.
If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
2025-03-16 21:18:27,002 - matplotlib.category - INFO - Using categorical units to plot a list of strings that are all parsable as floats or dates.
If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
2025-03-16 21:18:28,566 - SMS_Data_Preparation - INFO - Visualizations saved to data_exploration
(venv) PS [redacted]\C964-master\C964-master>
```

This command downloads the SMS Spam Collection dataset if it doesn't already exist on your machine and cleans and processes the data, as well as generates visualizations about the dataset as you will see later.

Now it is time to train the machine learning model on the processed dataset:

python model_training.py --train cleaned_spam.csv --optimize

This will load the cleaned dataset into the program, split it into training and testing sets. Then train and optimize a Naive Bayes classifier with TF-IDF vectorization then evaluate the model's performance. Note that the trained model will be saved as 'sms_spam_model.pkl', the evaluation results will be stored to model_evaluation.



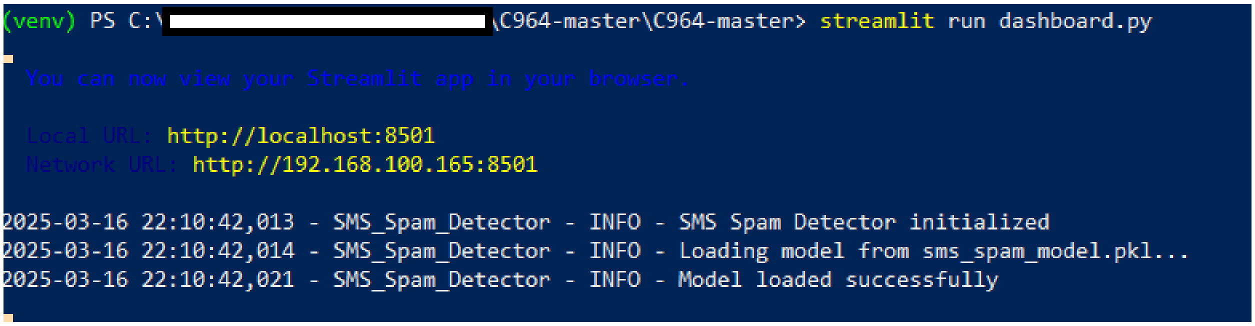
```
(venv) PS C:\[redacted]\C964-master\C964-master> python model_training.py --train cleaned_spam.csv --optimize --output sms_spam_model.pkl
2025-03-16 21:27:17,437 - SMS_Model_Training - INFO - Loading data from cleaned_spam.csv
2025-03-16 21:27:17,465 - SMS_Model_Training - INFO - Splitting data with test_size=0.2
2025-03-16 21:27:17,469 - SMS_Model_Training - INFO - Data loaded successfully. Training set: 4459 samples, Test set: 1115 samples
2025-03-16 21:27:17,471 - SMS_Model_Training - WARNING - Found 5 NaN values in training data. Replacing with empty strings.
2025-03-16 21:27:17,472 - SMS_Model_Training - WARNING - Found 1 NaN values in test data. Replacing with empty strings.
2025-03-16 21:27:17,472 - SMS_Model_Training - INFO - Building basic model
2025-03-16 21:27:17,473 - SMS_Model_Training - INFO - Performing 5-fold cross-validation
2025-03-16 21:27:18,611 - SMS_Model_Training - INFO - Cross-Validation Accuracy: 0.9623 (±0.0057)
2025-03-16 21:27:18,611 - SMS_Model_Training - INFO - Cross-Validation Precision: 0.9975 (±0.0050)
2025-03-16 21:27:18,613 - SMS_Model_Training - INFO - Cross-Validation Recall: 0.7207 (±0.0407)
2025-03-16 21:27:18,621 - SMS_Model_Training - INFO - Cross-Validation F1 Score: 0.8362 (±0.0286)
2025-03-16 21:27:18,621 - SMS_Model_Training - INFO - Optimizing model with GridSearchCV
Fitting 5 folds for each of 54 candidates, totalling 270 fits
2025-03-16 21:27:28,739 - SMS_Model_Training - INFO - Grid search completed in 10.12 seconds
2025-03-16 21:27:28,739 - SMS_Model_Training - INFO - Best parameters: {'classifier__alpha': 0.1, 'tfidf__max_features': 7000, 'tfidf__min_df': 1, 'tfidf__ngram_range': (1, 1)}
2025-03-16 21:27:28,741 - SMS_Model_Training - INFO - Best F1 score: 0.9262
2025-03-16 21:27:28,747 - SMS_Model_Training - INFO - Training the model
2025-03-16 21:27:28,804 - SMS_Model_Training - INFO - Model training completed in 0.06 seconds
2025-03-16 21:27:28,805 - SMS_Model_Training - INFO - Evaluating model on test set
2025-03-16 21:27:28,856 - SMS_Model_Training - INFO - Accuracy: 0.9803
2025-03-16 21:27:28,857 - SMS_Model_Training - INFO - Precision: 0.9774
2025-03-16 21:27:28,858 - SMS_Model_Training - INFO - Recall: 0.8725
2025-03-16 21:27:28,867 - SMS_Model_Training - INFO - F1 Score: 0.9220
2025-03-16 21:27:28,868 - SMS_Model_Training - INFO - ROC AUC: 0.9839
2025-03-16 21:27:28,869 - SMS_Model_Training - INFO - Confusion Matrix:
[[963   3]
 [ 19 130]]
2025-03-16 21:27:28,876 - SMS_Model_Training - INFO - Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       966
           1       0.98      0.87      0.92       149

    accuracy                           0.98      1115
   macro avg       0.98      0.93      0.96      1115
weighted avg       0.98      0.98      0.98      1115

2025-03-16 21:27:30,243 - SMS_Model_Training - INFO - Evaluation visualizations saved to model_evaluation
2025-03-16 21:27:30,243 - SMS_Model_Training - INFO - Saving model to sms_spam_model.pkl
2025-03-16 21:27:30,266 - SMS_Model_Training - INFO - Model saved successfully
Model training completed successfully!
Model saved to sms_spam_model.pkl
Evaluation results saved to model_evaluation
(venv) PS C:\[redacted]\C964-master\C964-master>
```

The program has now been trained and evaluated; to view the dashboard, you have two options. Option 1 is through the Streamlit dashboard which launches in your web browser and provides an overview of the system, visualizations of the model's performance, and a capability to upload a CSV file to be ran through the model. Option 2 is to use the CLI application where you can choose to analyze a single message, or a CSV file.

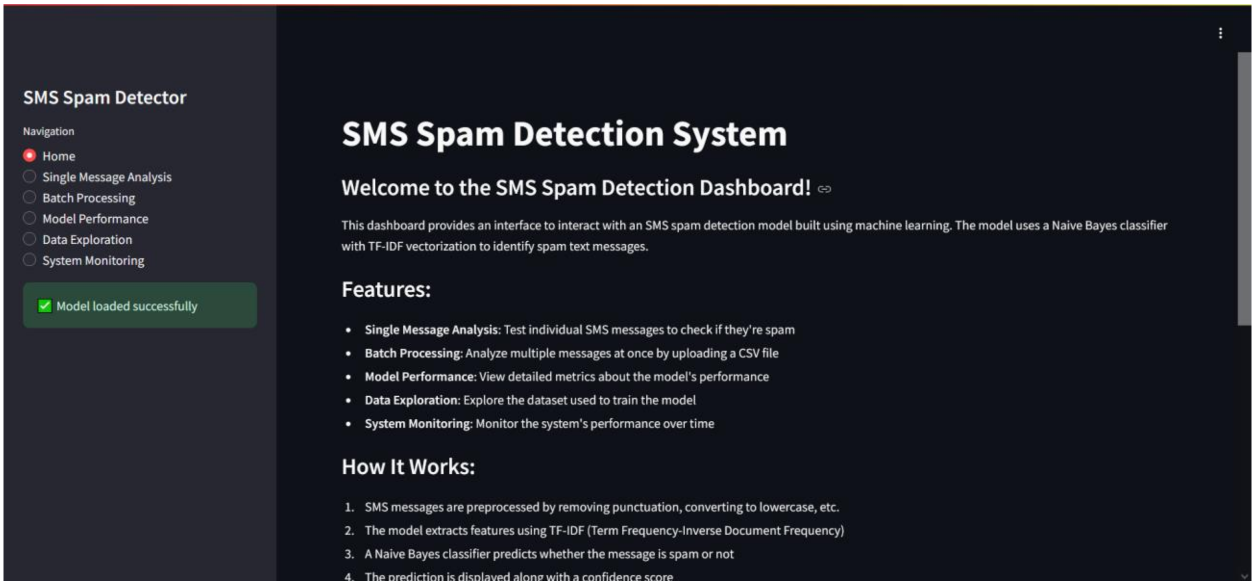
For your Streamlit dashboard run the following command:

```
streamlit run dashboard.py
```

This will launch the dashboard in your browser at 'http://localhost:8501/' in your default browser.



```
(venv) PS C:\[redacted]\C964-master\C964-master> streamlit run dashboard.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.100.165:8501

2025-03-16 22:10:42,013 - SMS_Spam_Detector - INFO - SMS Spam Detector initialized
2025-03-16 22:10:42,014 - SMS_Spam_Detector - INFO - Loading model from sms_spam_model.pkl...
2025-03-16 22:10:42,021 - SMS_Spam_Detector - INFO - Model loaded successfully
```



SMS Spam Detection System

Welcome to the SMS Spam Detection Dashboard!

This dashboard provides an interface to interact with an SMS spam detection model built using machine learning. The model uses a Naive Bayes classifier with TF-IDF vectorization to identify spam text messages.

Features:

- **Single Message Analysis:** Test individual SMS messages to check if they're spam
- **Batch Processing:** Analyze multiple messages at once by uploading a CSV file
- **Model Performance:** View detailed metrics about the model's performance
- **Data Exploration:** Explore the dataset used to train the model
- **System Monitoring:** Monitor the system's performance over time

How It Works:

1. SMS messages are preprocessed by removing punctuation, converting to lowercase, etc.
2. The model extracts features using TF-IDF (Term Frequency-Inverse Document Frequency)
3. A Naive Bayes classifier predicts whether the message is spam or not
4. The prediction is displayed along with a confidence score

Using the detailed information on the page you can navigate the webpage and explore all of its features.

To stop the dashboard, return to your terminal window and press ‘ctrl’ + ‘c’.

Troubleshooting:

If you are having trouble, please refer to the log files that each program generates after it is ran. Also ensure that you didn't encounter any problems when downloading requirements.txt. Please do not hesitate to reach out if any undocumented problems occur.