Application of Large Language Models for Spam Detection

MSc Research Project
Cybersecurity

Jose Merida
Student ID: 23271621

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Jose Fernando Merida Ramos		
Student ID:	23271621		
Programme:	Cybersecurity	**Year:**	2024
Module:	MSc Research Project		
Lecturer:	Michael Prior		
Submission Due Date:	12/12/2024		
Project Title:	Application of Large Language Models for Spam Detection		
Word Count:	256 **Page Count: 4**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jose Fernando Merida
Date:	12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jose Fernando Merida Ramos
Student ID: 23271621

1 Installing Dependencies

- Create Virtual Environment
    - python -m venv venvSpamDet

```
C:\Python3.8>python -m venv venvSpamDet
```

Figure 1 Creation of Virtual Environment

- Activate virtual environment
    - Go to your virtual environment, /scripts
    - Write activate

The image below shows the steps:

```
C:\Python3.8>cd venvSpamDet

C:\Python3.8\venvSpamDet>cd scripts

C:\Python3.8\venvSpamDet\Scripts>activate

(venvSpamDet) C:\Python3.8\venvSpamDet\Scripts>
```

Figure 2 Activate Virtual Environment

- Installing dependencies

- Move where the Python scripts are, in our case, the scripts are D:\2024\NCI\Semester 3\Practicum 2\GitHub\Final Project\Final-Project. Run the next snippet:
- pip install -r requirements.txt

```
(venvSpamDet) D:\2024\NCI\Semester 3\Practicum 2\GitHub\Final Project\Final-Project>pip install -r requirements.txt
Requirement already satisfied: fastapi in c:\python3.8\venvspamdet\lib\site-packages (from -r requirements.txt (line 1)) (0.115.5)
Requirement already satisfied: uvicorn in c:\python3.8\venvspamdet\lib\site-packages (from -r requirements.txt (line 2)) (0.32.1)
Requirement already satisfied: pydantic in c:\python3.8\venvspamdet\lib\site-packages (from -r requirements.txt (line 3)) (2.10.2)
Collecting transformers==4.21.1 (from -r requirements.txt (line 4))
```

Figure 3 Install Dependencies

2 Running the API

By this snippet you can start the API

- uvicorn app:app –reload

```
(venvSpamDet) D:\2024\NCI\Semester 3\Practicum 2\GitHub\Final Project\Final-Project>uvicorn app:app --reload
INFO:     Will watch for changes in these directories: ['D:\\2024\\NCI\\Semester 3\\Practicum 2\\GitHub\\Final Project\\Final-Projec
t']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [63048] using StatReload
c:\python3.8\venvspamdet\lib\site-packages\sklearn\base.py:348: InconsistentVersionWarning: Trying to unpickle estimator SVC from ve
rsion 0.24.2 when using version 1.3.2. This might lead to breaking code or invalid results. Use at your own risk. For more info plea
```

Figure 4 Running API

3 Testing the API

- Go to your browser and open the URL http://127.0.0.1:8000/docs#/default/classify_message_classify_post

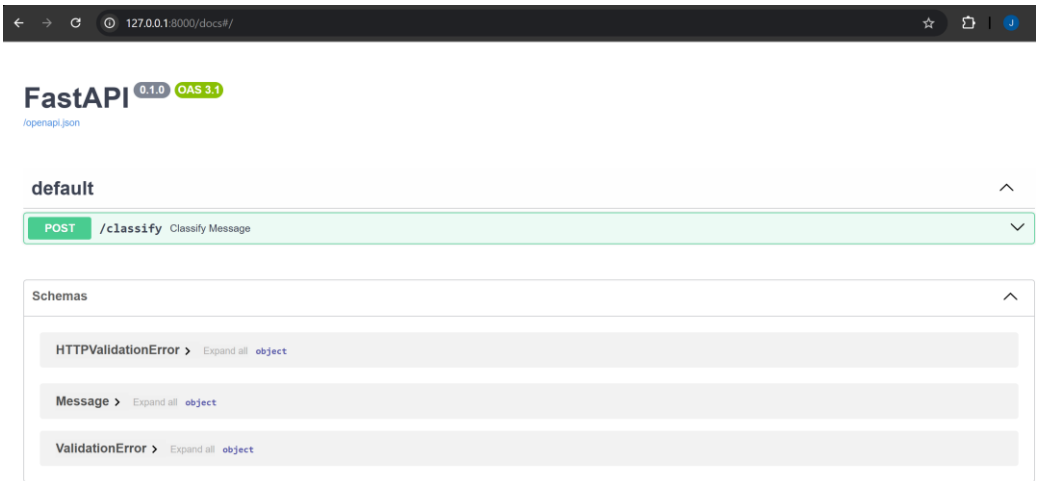

Figure 5 API

- Clic in /classify right bottom call “Try out”

Figure 6 Testing API

Now you can write a message to be detected as Spam or Ham, in our case the message is “Congratulations! You’ve been selected for a FREE gift card worth $1,000! Click here to claim”. After writing your message you can click on execute and see the response.

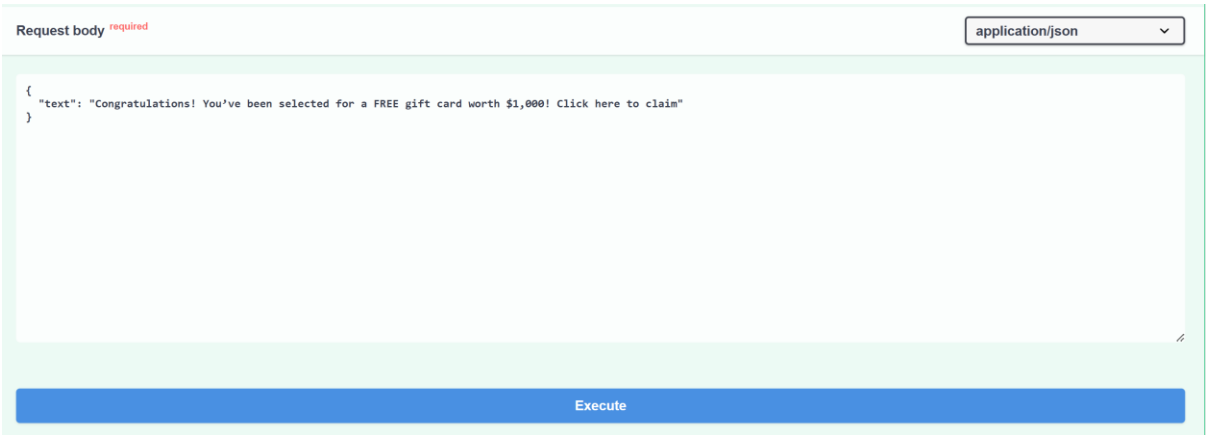

Figure 7 Executing API

The response has been classify as “Spam”:

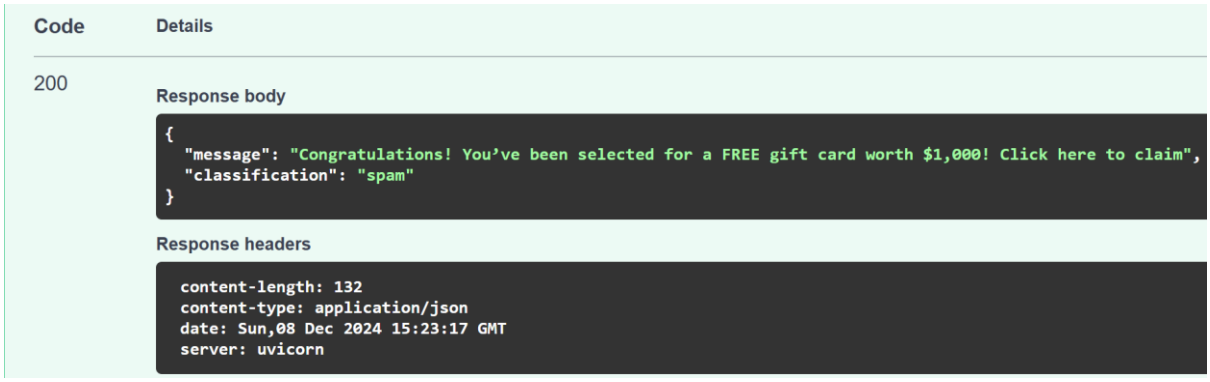

Figure 8 Response

4 Running the Jupyter Notebook Script

Open up the Jupyter Notebook, and open the file name BertSentimental.ipynb, the file contains:

- Installation of dependencies
- Script to take the Dataset and converting to embeddings using BERT
- SVM Training and Testing model
- The Propose Model's Matrix (precision, recall, f1-score, support and accuracy)
- Confusion Matrix
- Traditional Technique
- Traditional Technique's Matrix (precision, recall, f1-score, support and accuracy)

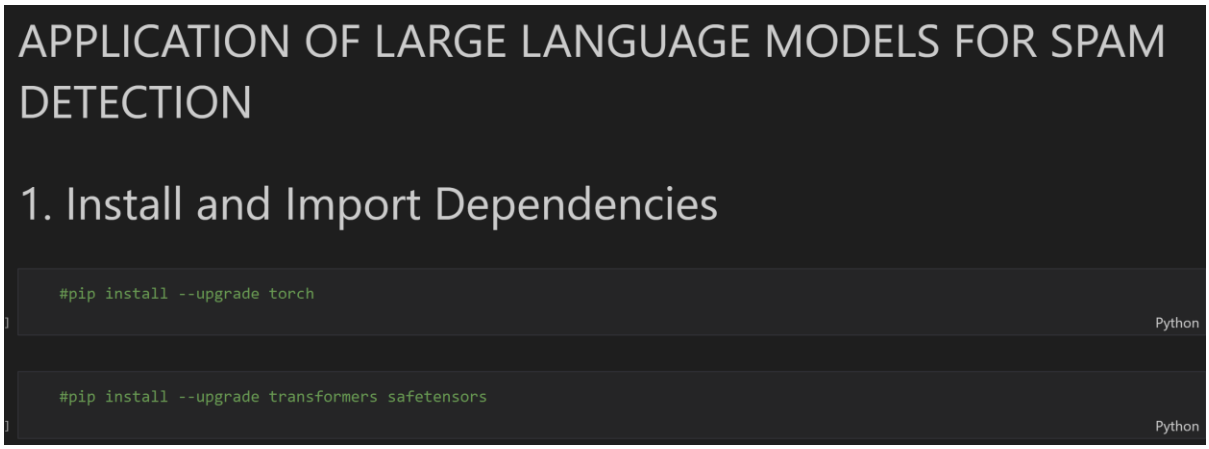

Figure 9 Jupyter Notebook Scripts