National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Shresta Sanjeeva Shetty
Student ID: x23204745

School of Computing
National College of Ireland

Supervisor: Eamon Nolan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shresta Sanjeeva Shetty
Student ID:	X23204745
Programme:	MSc in Data Analytics **Year:** 2024-2025
Module:	Research in Computing
Lecturer:	Eamon Nolan
Submission Due Date:	29th January 2025
Project Title:	Analyzing Customer Sentiment on Social Media for Brand Reputation and Feedback Insights
Word Count:	332 **Page Count:** 3

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: Shresta Sanjeeva Shetty

Date: 27th January 2025

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

Shresta Sanjeeva Shetty
Student ID: x23204745

1. SYSTEM REQUIREMENTS

Before running the application, ensure the system meets the following requirements:

- **Python version**: 3.11
- **Libraries**:
  - pandas
  - re
  - matplotlib
  - torch
  - transformers
  - sklearn
  - tqdm
- **Additional Tools**:
  - Google Colab for GPU/TPU support (recommended)
  - Internet connection for downloading pre-trained models from Hugging Face

2. SETTING UP THE ENVIRONMENT

1. **Install Python and Pip:** Download and install the latest version of Python from python.org.
2. **Install Dependencies:** Use the following command to install all required Python libraries:

pip install pandas matplotlib torch transformers scikit-learn tqdm

3. DATASET CONFIGURATION

Link to the dataset: https://www.kaggle.com/datasets/prkhrawsthi/twitter-sentiment-dataset-3-million-labelled-rows

- Primary Dataset: twitter_dataset.csv

  The dataset must be in the root directory or provide the correct file path.

- Dataset Columns: Ensure the dataset contains the following columns:

  tweet: Textual data for sentiment analysis.

- sentiment: Labels indicating sentiment (e.g., 0 for negative, 1 for neutral, 2 for positive).

4. CODE SECTIONS OVERVIEW

The application consists of several stages. Each stage requires specific configurations:

1. **Data Loading**:
   Modify file_path to point to the dataset location:

```
file_path = 'path_to_your_dataset/twitter_dataset.csv'
```

2. **Chunk Processing**
   Adjust chunk_size based on memory availability. The default is 100,000.

3. **Text Cleaning**:
   This process removes unwanted characters, lowercase text, and strips excess spaces.

4. **Class Balancing**

   The upsampling factor for minority classes can be adjusted in the following code:

5. **Model Training Configuration**

   The application supports Random Forest, Decision Tree, and BERT models.

   BERT model settings (e.g., max_length) can be modified in the tokenizer initialization:

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

6. **Batch Processing**

   Adjust batch sizes in the DataLoader initialization

```
train_dataloader = DataLoader(train_data, batch_size=16, shuffle=True)
val_dataloader = DataLoader(val_data, batch_size=64)
```

7. **Model Evaluation**

   Use the evaluate function to adjust metrics like loss and accuracy.

5. MODEL SAVING AND LOADING

1. **Saving the Model**

   The trained BERT model is saved locally
   model.save_pretrained('./bert_model')

2. **To compress and upload the model to Google Drive**

   !zip -r bert_model.zip ./bert_model

3. **Loading the Model**

   Ensure the saved model is accessible

```
model = BertModel.from_pretrained('/content/bert_model/bert_model')
```