National College of Ireland

Configuration Manual

MSc Research Project
Practicum 2

Diego Marquez Salazar
Student ID: x23410914

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Diego Marquez Salazar
Student ID:	x23410914
Programme:	MSc in AI for Business
Programme:	MSc in AI for Business
Module:	Practicum 2
Supervisor:	Rejwanul Haque
Submission Due Date:	Monday 15th September 2025
Project Title:	Fine-Tuned DistilBERT and Interactive Dashboard for Early Detection of Suicidal risk in Social Media Posts
Word Count:	614 **Page Count** 5

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:

Date: Monday 15th September 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

Diego Marquez Salazar
Student ID: x23410914

1 Introduction

This manual provides a detailed, step-by-step guide to replicate the development of the suicide risk detection system, from environment setup to model fine-tuning and deployment through a Streamlit dashboard.

It covers technical prerequisites, dataset preparation, model training, saving/loading the fine-tuned model, and running the dashboard for both single-text analysis and batch CSV file analysis.

2 Technical Requirements

2.1 Hardware Requirements

Minimum specifications for running model fine-tuning and dashboard locally:

- Processor: Intel Core i7 (12th Gen) or equivalent.
- Memory (RAM): 16 GB or more.
- GPU: NVIDIA RTX 4070 with 8 GB VRAM (or higher recommended).
- Storage: At least 20 GB free space.
- Operating System: Windows 10/11, macOS (latest), or Linux (Ubuntu 20.04 or newer).

Note: Fine-tuning on CPU is possible but extremely slow. GPU acceleration via CUDA is highly recommended.

2.2 Software Requirements

- Python: 3.9+
- Pip: Latest version
- Git: Installed for version control (optional)
- CUDA Toolkit & cuDNN: Compatible with your PyTorch version (for GPU training)

3 Environment Setup

3.1 Create and Activate Virtual Environment

```
python -m venv venv
venv\Scripts\activate
```

3.2 Install Required Libraries

Ensure you have requirements.txt containing:

```
accelerate==0.30.1
aiohappyeyeballs==2.6.1
aiohttp==3.12.15
aiosignal==1.4.0
async-timeout==5.0.1
attrs==25.3.0
certifi==2025.7.14
charset-normalizer==3.4.2
click==8.2.1
colorama==0.4.6
contourpy==1.3.2
cycler==0.12.1
datasets==4.0.0
dill==0.3.8
filelock==3.13.1
fonttools==4.59.0
frozenlist==1.7.0
fsspec==2024.6.1
huggingface-hub==0.34.3
idna==3.10
Jinja2==3.1.4
joblib==1.5.1
kiwisolver==1.4.8
MarkupSafe==2.1.5
matplotlib==3.10.5
mpmath==1.3.0
multidict==6.6.3
multiprocess==0.70.16
networkx==3.3
nltk==3.9.1
numpy==2.1.2
packaging==25.0
pandas==2.3.1
pillow==11.0.0
propcache==0.3.2
psutil==7.0.0
pyarrow==21.0.0
pyparsing==3.2.3
python-dateutil==2.9.0.post0
pytz==2025.2
PyYAML==6.0.2
regex==2025.7.34
requests==2.32.4
safetensors==0.5.3
scikit-learn==1.7.1
scipy==1.15.3
six==1.17.0
sympy==1.13.1
threadpoolctl==3.6.0
tokenizers==0.19.1
torch==2.5.1+cu121
torchaudio==2.5.1+cu121
```

```
torchvision==0.20.1+cu121
tqdm==4.67.1
transformers==4.40.2
typing_extensions==4.12.2
tzdata==2025.2
urllib3==2.5.0
xxhash==3.5.0
yarl==1.20.1

```

Install dependencies:

```
pip install -r requirements.txt
```

4 Data preparation

4.1 Download dataset

The dataset is available from Mendeley Data:

URL: https://data.mendeley.com/datasets/z8s6w86tr3/1

Download and save the CSV file into your project folder.

4.2 Understanding Dataset Structure

- **text:** The Reddit post content.
- **label:** Binary classification label (0 = no suicidal ideation, 1 = suicidal ideation).

4.3 Preprocessing Steps Implemented

- Removal of URLs, special characters, and excessive whitespace.
- Lowercasing text.
- Tokenization using the DistilBertTokenizerFast from HuggingFace Transformers.
- Truncation and padding to a max length of 128 tokens.

5 Fine-Tuning the Model

5.1 Load Pretrained Model and Tokenizer

```
from transformers import DistilBertTokenizerFast, DistilBertForSequenceClassification

tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-base-uncased')
model = DistilBertForSequenceClassification.from_pretrained ('distilbert-base-uncased',
num_labels=2)
```

5.2 Train-Test Split and Cross-Validation

- Used K-Fold Cross-Validation (k=5) to ensure robust evaluation.

- Balanced dataset ensured equal distribution of both classes across folds.

5.3 Training Process

- Optimizer: AdamW
- Learning rate: 5e-5
- Batch size: 16
- Epochs: 3-5 depending on GPU capacity.
- Loss function: CrossEntropyLoss
- Metrics: Accuracy, F1-score, ROC-AUC.

5.4 Saving the Model and Tokenizer

After training:

```
1. model.save_pretrained("model")
2. tokenizer.save_pretrained("model")
3.
```

This creates a /model folder containing all necessary files for later use.

6 Streamlit Dashboard Setup

The dashboard contains two main tabs:

- Single Text Analysis:
  - User inputs a single text in a text box.
  - The model outputs risk label and probability score.

- Batch CSV Analysis:
  - User uploads a CSV with a text column.
  - The dashboard processes all rows and displays:
    - Class distribution bar chart.
    - Histogram of predicted probabilities.
    - Frequent word cloud for high-risk texts.
    - Table with top high-risk entries.

6.1 Running the Dashboard

```
1. streamlit run app.py
```

7 Troubleshooting

CUDA Out of Memory Error:

- Reduce batch size (`batch_size=8` or `4`).
- Enable mixed precision training with `torch.cuda.amp`.

Tokenizer Import Error:

- Update Transformers library with:

```
1. pip install --upgrade transformers
```

References

No references available.