# Configuration Manual

MSc Research Project
Programme Name

Ciarán Jones
Student ID: 21135231

School of Computing
National College of Ireland

Supervisor: Anh Duong Trinh

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** ……. Ciarán Jones ……………………………………………………………………………………

**Student ID:** …………21135231…………………………………………………………………………………..……

**Programme:** … MSc in Data Analytics……………………………**Year:** ……2023……………..

**Module:** ……… MSc Research Project……………………………………………………….………

**Lecturer:** ……… Anh Duong Trinh ……………………………………….………

**Submission Due Date:** ………14/08/2023……………………………………………………………………….………

**Project Title:** …… Enhancing Ambulance Resource Management through Machine Learning-based Demand Prediction in Dublin, Ireland …………….………

**Word Count:** ……662………………………… **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:** ………14/08/2023……………………………………………………………………………

**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

Ciarán Jones
Student ID: 21135231

## 1 Repository Overview

The Ambulance-Demand-Forecast repository is diligently structured to streamline data processing, modelling, and analysis processes. Below provides an insight into the directory layout:

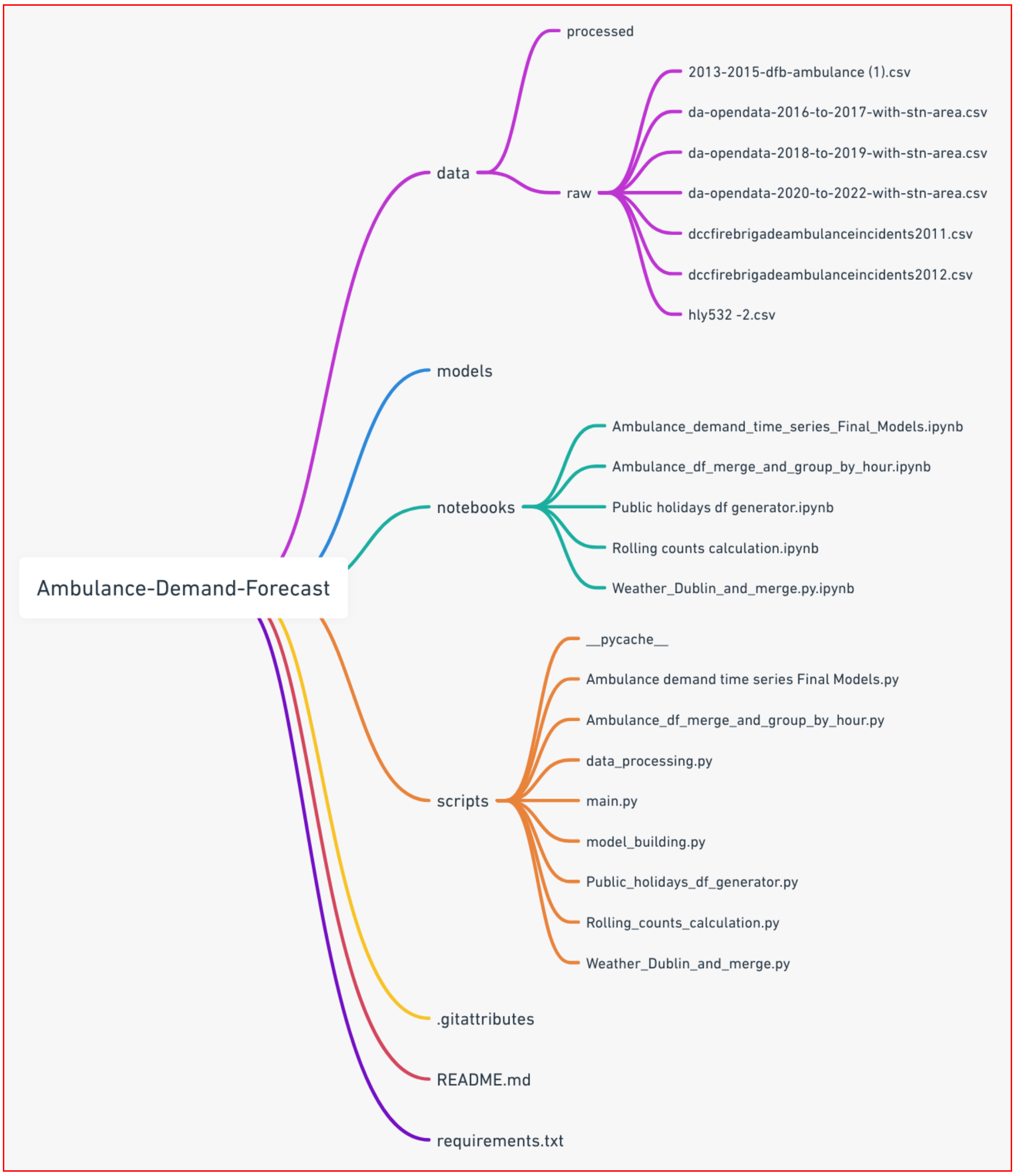
processed
2013-2015-dfb-ambulance (1).csv
da-opendata-2016-to-2017-with-stn-area.csv
da-opendata-2018-to-2019-with-stn-area.csv
da-opendata-2020-to-2022-with-stn-area.csv
dccfirebrigadeambulanceincidents2011.csv
dccfirebrigadeambulanceincidents2012.csv
hly532 -2.csv
data
raw
models
Ambulance_demand_time_series_Final_Models.ipynb
Ambulance_df_merge_and_group_by_hour.ipynb
Public holidays df generator.ipynb
Rolling counts calculation.ipynb
Weather_Dublin_and_merge.py.ipynb
notebooks
Ambulance-Demand-Forecast
__pycache__
Ambulance demand time series Final Models.py
Ambulance_df_merge_and_group_by_hour.py
data_processing.py
main.py
model_building.py
Public_holidays_df_generator.py
Rolling_counts_calculation.py
Weather_Dublin_and_merge.py
scripts
.gitattributes
README.md
requirements.txt

**Figure 1. Git Project Repository Structure**

**Description:**

- **data:**
  - processed: Contains datasets refined for modeling. Is populated by running data_processing.py.
  - raw: Initially untouched datasets, including ambulance logs, weather data, and holiday lists.
- **models:** Will contain trained forecasting models post running model_building.py script.
- **notebooks:** Contains Jupyter notebooks intended for exploratory data analysis, initial experimentations, and model evaluations.
- **scripts:** Houses Refactored Python scripts responsible for various stages of the pipeline, ranging from data amalgamation to model training.

# 2 Environment Setup

Before diving into the project, set up the appropriate environment. This section guides through the preparatory steps.

## 2.1 Python Installation:

Ensure Python 3.10.9 is installed. If missing, download and install.

## 2.2 Virtual Environment (Recommended):

- Firstly, install virtualenv by executing command in terminal:

  ‘pip install virtualenv’

- Move to the project directory:

  ‘cd path/to/Ambulance-Demand-Forecast’

- Craft a virtual environment:
  ‘python3.10 -m venv venv’

- Engage the virtual environment:
  - On Windows:

    ‘.\venv\Scripts\activate’
  - On macOS/Linux:

    ‘source venv/bin/activate’

For a comprehensive understanding of virtual environments, refer to the Python official documentation (Python Software Foundation, 2022).

### 2.3 Installing Dependencies:

After setting up Python and (optionally) a virtual environment, you can install the necessary libraries and dependencies using the requirements.txt file provided in the repository:

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

This command ensures that all required libraries, including their specific versions, are installed.

## 3 Data Management and Execution

For impeccable model outputs, data handling is pivotal. Here's a guide:

### 3.1 Data Directory Structure:

- **raw:** Hosts untouched data - ambulance logs, weather data, and holiday listings.
- **processed:** Post data refinement, datasets reside here.

### 3.2 Script Execution:

#### 3.2.1 Data Processing (data_processing.py):

Tasks covered:

- Data amalgamation.
- Generating holiday data.
- Weather data incorporation.
- Computing rolling ambulance demand.

Kick-start with running in terminal:

```
‘python scripts/data_processing.py’
```

#### 3.2.2 Model Building (model_building.py):

Commences the model training encompassing:

MLP Neural Network algorithm.
XGBoost algorithm.
TensorFlow Decision Forests algorithm.

Run via command:

```
‘python scripts/model_building.py’
```

## 3.3 Model Repository:

Once cultivated, models will populate in the ‘models’ directory.

For a deeper understanding of TensorFlow's Decision Forests, refer to TensorFlow's official documentation (TensorFlow Authors, 2022).

# 4 Google Colab Execution for Model Building

Google Colab is a cloud-based platform that provides free access to high-performance GPUs, including the T4, making it an excellent choice for intensive model training sessions. To execute the model building process using the provided Jupyter notebook on Google Colab:

## 4.1 Setting Up Google Colab:

1. Visit Google Colab. Link in references (Google Colab ,2023).
2. Sign in with your Google account.
3. Click on **Upload** and select the
   ‘**Ambulance_demand_time_series_Final_Models.ipynb**’
   file from the notebooks directory.
4. Also upload the **‘Final_full_ambulance_df.csv’** file from the data/processed folder.

## 4.2 GPU Configuration:

1. Once the notebook is uploaded, click on **Runtime** in the top menu.
2. Select **Change runtime type**.
3. In the pop-up window, set the **Hardware accelerator** dropdown to **GPU**.

## 4.3 Executing the Notebook:

1. Navigate to the data cell and adjust the paths to point to the appropriate location in your Google Drive or upload data directly to Colab.
2. Execute the cells sequentially.

## 4.4 Leveraging the T4 GPU:

By default, the T4 GPU provided by Google Colab should be utilized by TensorFlow or any other deep learning library in use. Ensure your models are set to run on GPU by checking the device placement or configuring the necessary GPU settings in your training code.

**Note**: Always remember to save your trained models and any other outputs to Google Drive or download them to your local machine before ending the Colab session. Colab's VM is ephemeral, and you'll lose any data saved locally once the session ends.

# References

Google Colab (2023). Google Colaboratory [Online]. Available at: https://colab.research.google.com/ (Accessed: 10 August 2023).

Python Software Foundation (2022). Virtual Environments and Packages. Python.org. Available at: https://docs.python.org/3/tutorial/venv.html [Accessed 10 August 2023].

TensorFlow Authors (2022). TensorFlow Decision Forests. TensorFlow.org. Available at: https://www.tensorflow.org/decision_forests [Accessed 10 August 2023].