Configuration Manual

MSc Research Project
Master of Science in Data Analytics

Deepak Singh Kirola
Student ID: x22141855

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Deepak Singh Kirols
Student ID:	x22141855
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Arghir Nicolae Moldovan
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	6

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:	Deepak Singh Kirola
Date:	30th January 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

Deepak Singh Kirola
x22141855

1 Introduction

This document aims to provide clear instructions for recreating the deep learning framework used to retrieve memories from lifelogging data

1.1 Target Audience

This manual is for people and companies in the construction business looking to improve workplace safety using advanced computer techniques Mohammed and Mahmud (2020). It's made for those who handle the computer system, analyze data, and take care of the special safety management system we talk about here

2 Hardware Requirements

The following setups were used to carry out the project:

2.1 Local Machine

- **Operating System:** macOS Sonoma, Version 14.0
- **Chip:** Apple M2
- **Memory:** 8 GB
- **Display:** Built-in Liquid Retina Display, 13.6-inch (2560 × 1664)
- **Storage:** Macintosh HD - 94.05 GB available of 245.11 GB

Figure 1: Hardware Configuration on Local Machine

3 Software Requirement

3.1 Tableau:

Purpose: Insightful Visualization
Description: Tableau Desktop version 2022.4 Figure 2 is utilized for insightful visualization, creating clear representations that enhance the interpretability of incident data, aiding informed decision-making.

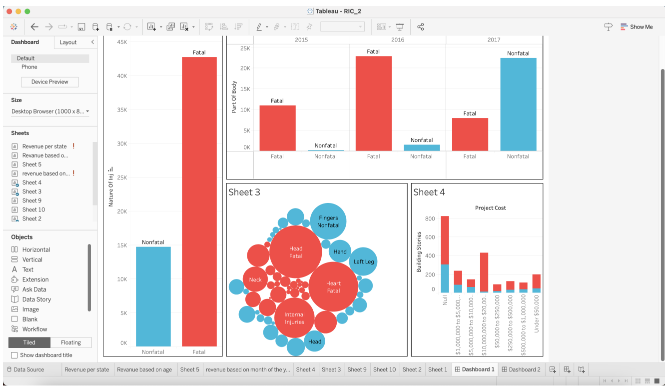

Figure 2: Configuration on Tableau

3.2 Python:

- **Purpose:** Foundation for Machine Learning Implementation
- **Description:** Python 3.9.13 Figure 3 serves as the core programming language, supporting diverse tasks such as modeling and visualization within our machine learning system.

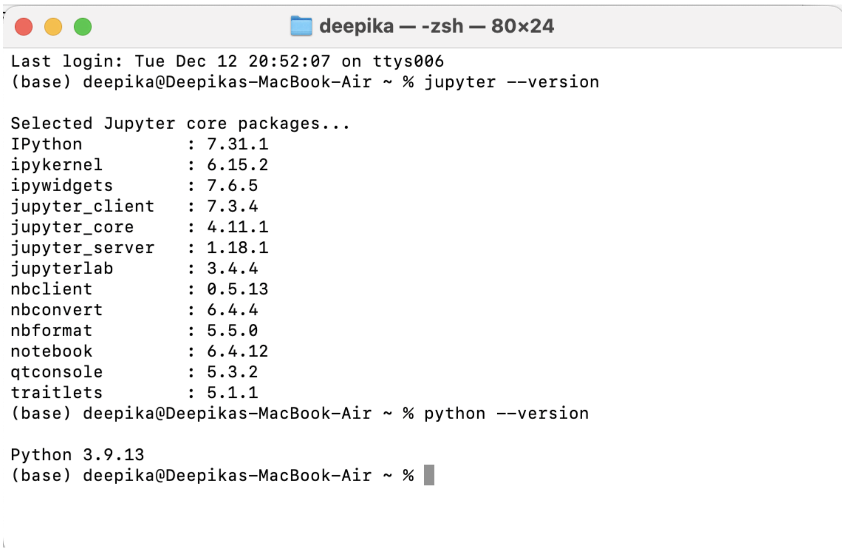

Figure 3: Configuration on Python

Using Anaconda Navigator, the retrieval framework was set up on the home server using a Jupyter Notebook Figure 2. The navigator makes opening a Jupyter Notebook, running Python code, and obtaining pictures easier.

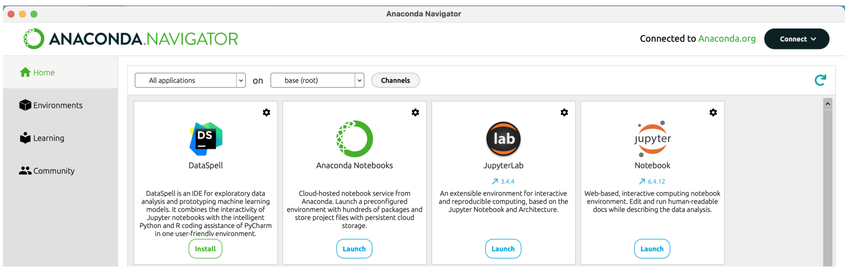

Figure 4: Configuration on Anaconda Navigator

4 Libraries Requirements

The imported libraries are listed below Figure 5:

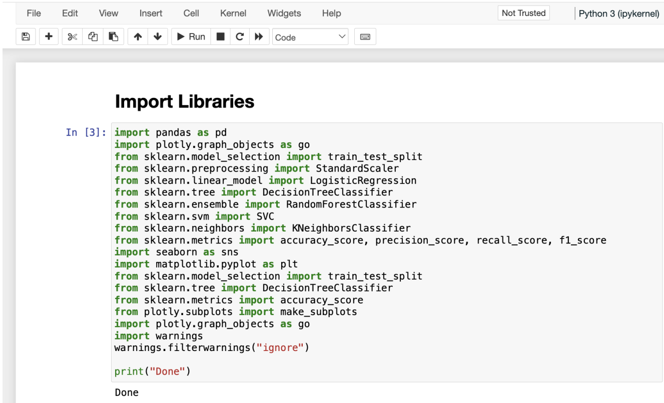

Figure 5: Configuration on Anaconda Navigator

5 Dataset Description

- **Name:** OSHA Accident and Injury Data `https://www.kaggle.com/datasets/ruqaiyaship/osha-accident-and-injury-data-1517`
- **Description:** This dataset includes summaries that summarize construction worker incidents and injuries from 2015 to 2017.
- **Rows:** 4848
- **Columns:** 29

6 Model Preparation

In the model preparation phase, various critical steps are taken to configure the machine learning model for optimal utilization. The following points outline the configuration details:

- **Data Preprocessing:** Prepare the dataset by managing missing values Figure 6, scaling features Figure 7, and encoding categorical variables.
- **Feature Selection:** Identify and choose pertinent features that significantly contribute to the model's effectiveness.
- **Model Training:** Train the machine learning model using the processed data Figure 8.
- **Model Selection:** Opt for the suitable machine learning model based on the problem's nature (classification, regression) and the data's characteristics.
- **Hyperparameter Tuning:** Adjust the model's hyperparameters precisely to enhance its performance.

Data Preprocessing

```
In [9]: # Print columns with missing values
missing_value_cols = []
for column in data_injury.columns:
    miss_cnt = data_injury[column].isnull().sum()
    if miss_cnt > 0:
        missing_value_cols.extend([column])
print(f"The columns with missing values:", missing_value_cols)

The columns with missing values: ['Nature of Injury', 'Part of Body', 'Event type', 'Enviro
nmental Factor', 'Human Factor']
```

```
In [10]: # Print columns with missing values and their respective counts
missing_value_info = []
for column in data_injury.columns:
    miss_count = data_injury[column].isnull().sum()
    if miss_count > 0:
        missing_value_info.append({'Column': column, 'Missing Values': miss_count})

# Print the results
print("Columns with missing values:")
for info in missing_value_info:
    print(f"{info['Column']}: {info['Missing Values']} missing values")

Columns with missing values:
Nature of Injury: 2 missing values
Part of Body: 2 missing values
Event type: 2 missing values
Environmental Factor: 7 missing values
Human Factor: 7 missing values
```

```
In [11]: data_injury.dropna(inplace=True)
```

```
In [12]: # Display the shape of the dataset
print("Shape of the dataset :-", data_injury.shape)

Shape of the dataset :- (4838, 29)
```

Figure 6: Data Preprocessing

```
In [35]: from sklearn.preprocessing import MinMaxScaler


numeric_columns = ["Construction End Use", "Building Stories", "Project Cost", "Project Type"
                   'nature_of_inj','part_of_body','event_type','Environmental Factor','hum_fac
scaler = MinMaxScaler()
data_injury[numeric_columns] = scaler.fit_transform(data_injury[numeric_columns])

print(data_injury.head())

   Construction End Use  Building Stories  Project Cost  Project Type  \
0              0.000000          0.000000      0.000000           0.0
1              0.470588          0.041667      0.000000           0.2
2              0.000000          0.000000      0.000000           0.0
3              0.117647          0.041667      0.857143           0.2
4              0.000000          0.000000      0.000000           0.0

   Degree of Injury  nature_of_inj  part_of_body  event_type  \
0                 1       0.000000      0.300000    0.076923
1                 1       0.380952      0.300000    0.076923
2                 1       0.190476      0.366667    1.000000
3                 0       0.952381      0.400000    0.307692
4                 1       0.095238      0.633333    0.000000

   Environmental Factor  hum_factor  task_assigned
0                0.0000    0.684211              0
1                0.4375    0.473684              0
2                0.4375    0.157895              1
3                1.0000    0.684211              0
4                0.5000    0.736842              0
```

Figure 7: scaling features

Split Data

```
In [39]: from sklearn.model_selection import train_test_split

# Define your features (X) and the target variable (y)
X = data_injury.drop('Degree of Injury', axis=1)  # Features
y = data_injury['Degree of Injury']  # Target variable

X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.25, random_state=42)
```

Figure 8: Model Training

References

Mohammed, J. and Mahmud, M. J. (2020). Selection of a machine learning algorithm for osha fatalities, *2020 IEEE Technology & Engineering Management Conference (TEMSCON)*, IEEE, pp. 1–5.