Configuration Manual

MSc Research Project
MSc in Data Analytics

Rohan Sunil Mohite
Student ID: x23118865

School of Computing
National College of Ireland

Supervisor: Furqan Rustam

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rohan Sunil Mohite
Student ID:	x23118865
Programme:	MSc in Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Furqan Rustam
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	XXX
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:	Rohan Sunil Mohite
Date:	16th September 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

Rohan Sunil Mohite
x23118865

1 Introduction

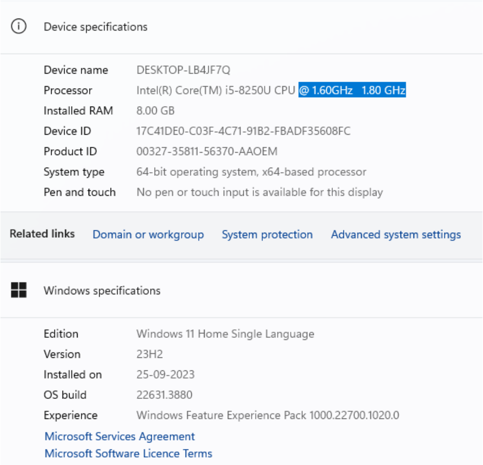
This manual provides a brief on how to conduct the prediction of extramarital affairs utilizing deep learning techniques. This including the procedures on how to set up the environment, implement as well as evaluate the models.

2 System Requirements

The following configurations was applied for the project's implementation:

2.1 Local Machine

Windows 11 with 5th Gen Intel(R) Core(TM) i5-8250U @ 1.60GHz 1.80 GHz with 8GB Ram and 64 Bit operating system.



Device specifications

Device name	DESKTOP-LB4JF7Q
Processor	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Installed RAM	8.00 GB
Device ID	17C41DE0-C03F-4C71-91B2-FBADF35608FC
Product ID	00327-35811-56370-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Windows specifications

Edition	Windows 11 Home Single Language
Version	23H2
Installed on	25-09-2023
OS build	22631.3880
Experience	Windows Feature Experience Pack 1000.22700.1020.0

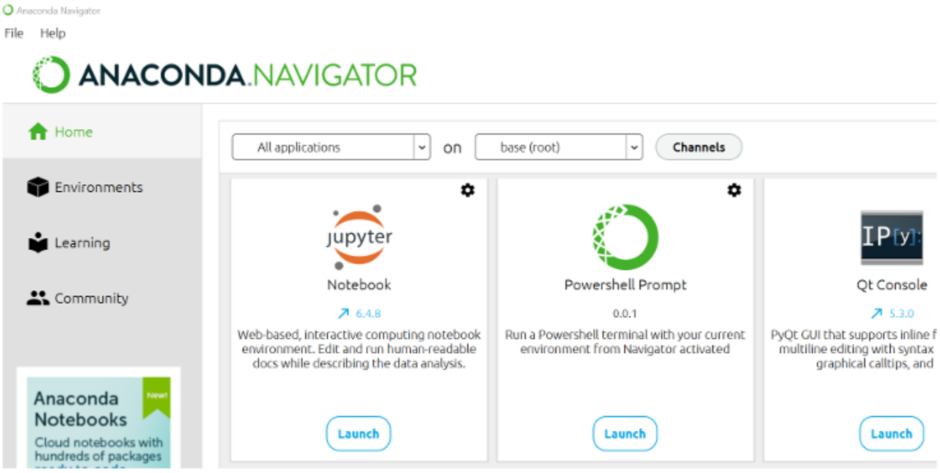
Microsoft Services Agreement
Microsoft Software Licence Terms

Figure 1: Hardware Configuration

3 Software Requirements

For this project, the coding processes were done with the help of Jupyter Notebook in Anaconda navigator which is a python environment allowing for combining code, visualizations, and descriptive documents. This environment helped in calls to run the various

Python scripts and in performing the data pre-processing and cleaning, model building, and model performance evaluation activities.



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Run a Powershell terminal with your current environment from Navigator activated

PyQt GUI that supports inline f multiline editing with syntax graphical calltips, and

Figure 2: Anaconda Navigator

4 Package Requirements

All the requirement packages in the Python environment were installed via pip and conda in Jupyter notebook. Below is the list of the packages that has been installed.

- **Pandas**
- **Scipy**
- **Scikit-learn**
- **Tensorflow**
- **Numpy**
- **Matplotlib**
- **Seaborn**

5 Dataset Description

The data set used in this project was downloaded from Kaggle and contains 6, 367 entries with 10 variables, and it includes demographic and social aspects that contributes to infidelity. This data is useful for to calibrate and assess the models for predicting the human behavior, which is used in this research.

https://www.kaggle.com/datasets/gargmanas/affairsdata

5.1 Variable Descriptions of dataset

The values present in the dataset follows a particular scale which is showned in table

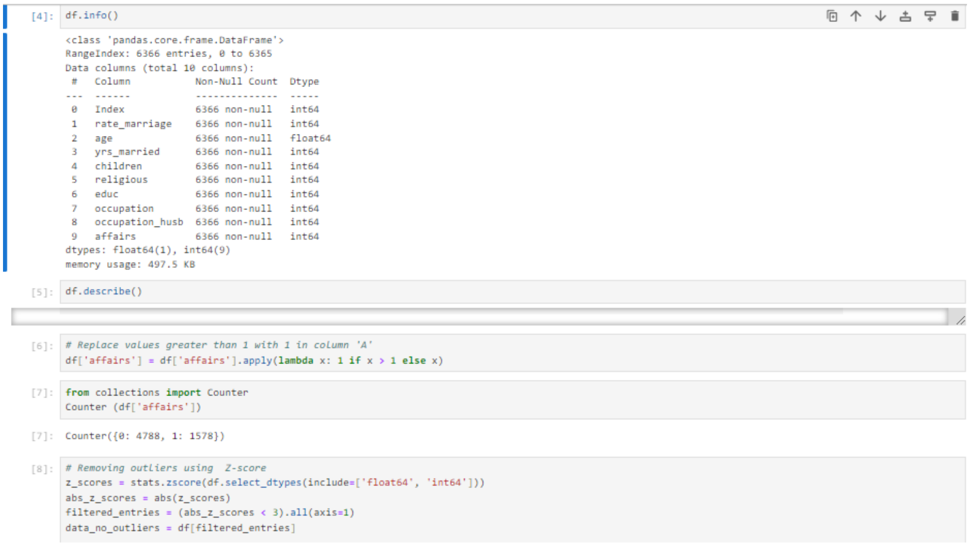
Number of observations:	6366
Number of variables:	9
Variable name definitions:	
rate_marriage	How rate marriage, 1 = very poor, 2 = poor, 3 = fair, 4 = good, 5 = very good
age	Age
yrs_married	No. years married. Interval approximations. See original paper for detailed explanation.
children	Children
religious	How religious, 1 = not, 2 = mildly, 3 = fairly, 4 = strongly
educ	Level of education, 9 = grade school, 12 = high school, 14 = some college, 16 = college graduate, 17 = some graduate school, 20 = advanced degree
occupation	1 = student, 2 = farming, agriculture; semi-skilled, or unskilled worker; 3 = white-collar; 4 = teacher counselor social worker, nurse; artist, writers; technician, skilled worker, 5 = managerial, administrative, business, 6 = professional with advanced degree
occupation_husb	Husband's occupation. Same as occupation.
affairs	Measure of time spent in extramarital affairs

6 Model Preparation

The BinaryCLass.ipynb and MultiClass.ipnb can be found in the artefacts zip file and the file describes the whole installation of necessary libraries, loading models, and the processing of data. It describes the steps needed to train the models, the process through which their performances are assessed, and how their results like the models' predictions and performances, are stored. The difference between Multi class and Binary class implementation is, in Binary class the target variable value greater than 1 is considered as 1 and the value less than 1 is considered as 0.

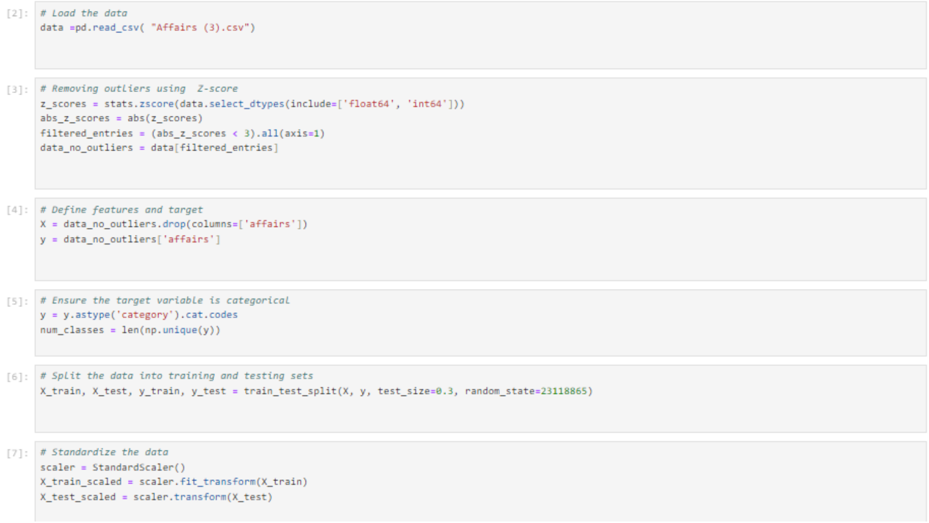
6.1 Model Implementation

The first step in the implementation is the data preprocessing, in which I treated missing values in an adequate manner and transformed categorical variables into a format that can be used in machine learning model. I normalized features where necessary because normalization is in most cases very important especially with models such as SVM and KNN. Distinguishing outliers and removing them from the data or using techniques such as Z-score filtering. In model building, I used Logistic Regression, which is easy to interpret and Random Forest is used due to complexity and to prevent issue of overfitting. I also applied long short-term memory and Convolutional neural network to analyze sequential data for its patterns. The architectures of these models were developed with much consideration in order to achieve the best results; LSTM was used due to its ability to work with time-dependent data while CNN ,because of its ability to work on features.Each of the used models was hyperparameter tuned and applied K-fold cross validation to select the best settings. In order to examine the models, evaluation measures such as accuracy, precision, recall, and F1-score were adopted to get the best picture of the accuracy levels of the models.Following all these steps, it is possible to make a very solid and easily scalable solution to replicated by other people or teams.



```
[4]: df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 6366 entries, 0 to 6365
     Data columns (total 10 columns):
      #   Column           Non-Null Count  Dtype  
     ---  ------           --------------  -----  
      0   Index            6366 non-null   int64  
      1   rate_marriage    6366 non-null   int64  
      2   age              6366 non-null   float64
      3   yrs_married      6366 non-null   int64  
      4   children         6366 non-null   int64  
      5   religious        6366 non-null   int64  
      6   educ             6366 non-null   int64  
      7   occupation       6366 non-null   int64  
      8   occupation_husb  6366 non-null   int64  
      9   affairs          6366 non-null   int64  
     dtypes: float64(1), int64(9)
     memory usage: 497.5 KB

[5]: df.describe()

[6]: # Replace values greater than 1 with 1 in column 'A'
     df['affairs'] = df['affairs'].apply(lambda x: 1 if x > 1 else x)

[7]: from collections import Counter
     Counter (df['affairs'])

[7]: Counter({0: 4788, 1: 1578})

[8]: # Removing outliers using  Z-score
     z_scores = stats.zscore(df.select_dtypes(include=['float64', 'int64']))
     abs_z_scores = abs(z_scores)
     filtered_entries = (abs_z_scores < 3).all(axis=1)
     data_no_outliers = df[filtered_entries]
```

Figure 3: Binary Class Implementation



```
[2]: # Load the data
     data =pd.read_csv( "Affairs (3).csv")

[3]: # Removing outliers using  Z-score
     z_scores = stats.zscore(data.select_dtypes(include=['float64', 'int64']))
     abs_z_scores = abs(z_scores)
     filtered_entries = (abs_z_scores < 3).all(axis=1)
     data_no_outliers = data[filtered_entries]

[4]: # Define features and target
     X = data_no_outliers.drop(columns=['affairs'])
     y = data_no_outliers['affairs']

[5]: # Ensure the target variable is categorical
     y = y.astype('category').cat.codes
     num_classes = len(np.unique(y))

[6]: # Split the data into training and testing sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=23118865)

[7]: # Standardize the data
     scaler = StandardScaler()
     X_train_scaled = scaler.fit_transform(X_train)
     X_test_scaled = scaler.transform(X_test)
```

Figure 4: Multi Class Implementation