

Configuration Manual

MSc Research Project
Data Analytics

Daphne Shekinah Tennison Daniel

Student ID: X23190027

School of Computing
National College of Ireland

Supervisor: Vikas Tomer

**National College of Ireland
Project Submission Sheet
School of Computing**



Student Name:	Daphne Shekinah Tennison Daniel
Student ID:	X23190027
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Vikas Tomer
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	493
Page Count:	7

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:	Daphne Shekinah Tennison Daniel
Date:	29th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

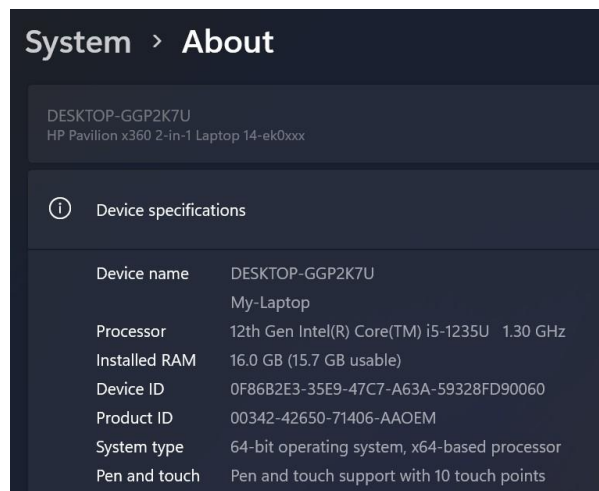
Daphne Shekinah Tennison Daniel
X23190027

1 Introduction

This is the configuration manual made for the research done on the topic, "Predictive Modeling of Readmission in Patients with Schizophrenia Using Machine Learning Models". The purpose of this manual is to enable anyone, who's willing to create a research that's similar to this, the steps to proceed. This manual has the code snippets for the key parts. Section 2 shows the hardware and software used to run the code. Section 3 and Section 5 has the specifics of the dataset and pre-processing done on the said dataset, respectively. The libraries that are used for data processing and modelling of the dataset is briefed in Section 4. Section 6 and Section ?? contains the details of the prediction models and the evaluation metrics, respectively.

2 Environment

The code for this project is run in a with the specifications shown in Figure 1. The code was run in Google Colab.



The image shows a screenshot of the 'System > About' page in Google Colab. It displays the following information:

DESKTOP-GGP2K7U HP Pavilion x360 2-in-1 Laptop 14-ek0xxx	
Device specifications	
Device name	DESKTOP-GGP2K7U My-Laptop
Processor	12th Gen Intel(R) Core(TM) i5-1235U 1.30 GHz
Installed RAM	16.0 GB (15.7 GB usable)
Device ID	0F86B2E3-35E9-47C7-A63A-59328FD90060
Product ID	00342-42650-71406-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	Pen and touch support with 10 touch points

Figure 1: System Configuration

3 Dataset

The dataset used in this research project was taken from the Central Statistics Office website. It was published by the National Psychiatric Inpatient Reporting System (NPIRS)

under the Health Research Board of Ireland. It was last downloaded on October 31, 2024, before it was updated on November 7, 2024. ¹

4 Libraries Used

Figure 2 shows the list of all the libraries used in the code for data visualizations, model implementation and the metrics used for the evaluation of the models.

```
Libraries

[2] import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, explained_variance_score
import numpy as np
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
import xgboost as xgb

from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

Figure 2: Libraries Used

5 Data Cleaning

This section shows all the steps taken to clean the dataset in order to implement the machine learning models. Figure 3 shows the dataset before it was cleaned.

```
#dataset preview
data.info(), data.head()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4284 entries, 0 to 4283
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Statistic Label                       4284 non-null   object
1   Year                                 4284 non-null   int64
2   Type of Admission                    4284 non-null   object
3   Sex                                  4284 non-null   object
4   ICD 10 Diagnostic Group              4284 non-null   object
5   UNIT                                4284 non-null   object
6   VALUE                                2786 non-null   float64
dtypes: float64(1), int64(1), object(5)
memory usage: 234.4+ KB
```

Figure 3: Dataset

¹<https://data.cso.ie/table/HRA05>

Figure 4 shows the dataset cleaning steps taken.

```

v Data Cleaning

[ ] #Filter the dataset for Schizophrenia
schizophrenia_keywords = ["Schizophrenia", "schizotypal", "delusional"]
schizophrenia_data = data[data['ICD 10 Diagnostic Group'].str.contains(
    "|".join(schizophrenia_keywords), case=False, na=False
)]

[ ] schizophrenia_data.loc[:, 'VALUE'] = schizophrenia_data['VALUE'].fillna(0)

[ ] #grouping data for admission
schizophrenia_trends = schizophrenia_data.groupby(
    ['Year', 'Sex', 'ICD 10 Diagnostic Group']
)['VALUE'].sum().reset_index()

```

Figure 4: Dataset Cleaning

The cleaned dataset is shown in Figure 5.

```

#cleaned dataset
schizophrenia_trends.head(), total_trend.head()

```

	Year	Sex	ICD 10 Diagnostic Group	VALUE
0	2006	Both sexes	Schizophrenia	0.0
1	2006	Both sexes	Schizophrenia, schizotypal and delusional diso...	4805.8
2	2006	Female	Schizophrenia	0.0
3	2006	Female	Schizophrenia, schizotypal and delusional diso...	1807.5
4	2006	Male	Schizophrenia	0.0,

	Year	Total Admissions	YoY Change
0	2006	9722.3	NaN
1	2007	9552.4	-169.9
2	2008	9562.6	10.2
3	2009	10018.3	455.7
4	2010	9467.5	-550.8)

Figure 5: Dataset after cleaning

Further feature engineering was performed on the dataset and it is shown in Figure 6.

```
#Rolling Average admissions for 3 years
total_trend['Rolling Avg (3Y)'] = total_trend['Total Admissions'].rolling(window=3).mean()

#Contribution of each sex in total admissions percentage
sex_trend = schizophrenia_trends.groupby(['Year', 'Sex'])['VALUE'].sum().reset_index()
total_by_year = sex_trend.groupby('Year')['VALUE'].sum().reset_index()
sex_trend = sex_trend.merge(total_by_year, on='Year', suffixes=('', '_Total'))
sex_trend['Percentage Contribution'] = (sex_trend['VALUE'] / sex_trend['VALUE_Total']) * 100

total_trend.tail(), sex_trend.head()
```

	Year	Total Admissions	YoY Change	Rolling Avg (3Y)
12	2018	8877.6	202.3	8790.833333
13	2019	9143.6	266.0	8898.833333
14	2020	8726.9	-416.7	8916.033333
15	2021	8774.3	47.4	8881.600000
16	2022	9251.0	476.7	8917.400000

	Year	Sex	VALUE	VALUE_Total	Percentage Contribution
0	2006	Both sexes	4805.8	9722.3	49.430690
1	2006	Female	1807.5	9722.3	18.591280
2	2006	Male	3109.0	9722.3	31.978030
3	2007	Both sexes	4721.8	9552.4	49.430510
4	2007	Female	1850.4	9552.4	19.371048

Figure 6: Feature Engineering

6 Modeling & Evaluation

This section shows the models used for predicting both total admissions and readmissions. The machine learning models used for the prediction of total admissions is shown in Figure 7.

```
model_data = total_trend.dropna()
X = model_data[['Year', 'YoY Change', 'Rolling Avg (3Y)']]
y = model_data['Total Admissions']

#Split data
#Training 2006 - 2020
#Testing 2021 - 2022

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=2, shuffle=False)

models = {
    "Random Forest": RandomForestRegressor(random_state=42, n_estimators=100),
    "Linear Regression": LinearRegression(),
    "XGBoost": xgb.XGBRegressor(random_state=42, objective="reg:squarederror", n_estimators=100),
}
```

Figure 7: Models for Total Admissions

The evaluation metrics used for the total admissions model is given in Figure 8.

```

results = {}
for model_name, model in models.items():
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    evs = explained_variance_score(y_test, y_pred)

    n = len(y_test)
    p = X_test.shape[1]
    adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)

    results[model_name] = {
        "Model": model,
        "RMSE": rmse,
        "MAE": mae,
        "R²": r2,
        "Adjusted R²": adj_r2,
        "Explained Variance": evs,
        "Predictions": y_pred,
    }

```

Figure 8: Evaluation for Total Admissions

The steps taken to prepare the dataset for the prediction of readmissions is shown in Figure 9.

```

Readmissions

from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
import xgboost as xgb

data['Readmitted'] = data['Type of Admission'].apply(lambda x: 1 if x == 'All admissions' else 0)

X = data[['Year', 'Sex', 'ICD 10 Diagnostic Group', 'VALUE']]
y = data['Readmitted']

X.loc[:, 'VALUE'] = X['VALUE'].fillna(X['VALUE'].median())
X = X.fillna('Unknown')

X = pd.get_dummies(X, drop_first=True)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

```

Figure 9: Data Prep for Readmissions

Implementation of Logistic Regression for the prediction of readmissions is given in

Figure 10.

```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

#Scaling

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = LogisticRegression(solver='liblinear', max_iter=1000, C=1.0)

model.fit(X_train_scaled, y_train)

LogisticRegression
LogisticRegression(max_iter=1000, solver='liblinear')

y_pred = model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

Figure 10: Logistic Regression for Readmissions

Implementation of Random Forest for readmission prediction is given below.

```
model_data = total_trend.dropna()
X = model_data[['Year', 'YoY Change', 'Rolling Avg (3Y)']]
y = model_data['Total Admissions']

#Split data
#Training 2006 - 2020
#Testing 2021 - 2022

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=2, shuffle=False)

models = {
    "Random Forest": RandomForestRegressor(random_state=42, n_estimators=100),
    "Linear Regression": LinearRegression(),
    "XGBoost": xgb.XGBRegressor(random_state=42, objective="reg:squarederror", n_estimators=100),
}
```

Figure 11: Random Forest for Readmissions

XGBoost is implemented for the prediction of hospital readmissions and it is presented in Figure 12.


```

model_data = total_trend.dropna()
X = model_data[['Year', 'YoY Change', 'Rolling Avg (3Y)']]
y = model_data['Total Admissions']

#Split data
#Training 2006 - 2020
#Testing 2021 - 2022

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=2, shuffle=False)

models = {
    "Random Forest": RandomForestRegressor(random_state=42, n_estimators=100),
    "Linear Regression": LinearRegression(),
    "XGBoost": xgb.XGBRegressor(random_state=42, objective="reg:squarederror", n_estimators=100),
}

```

Figure 12: XGBoost for Readmissions