National College of Ireland

Configuration Manual

MSc Research Project
MSc in Data Analytics

Antony Yesudas
Student ID: x20243405

School of Computing
National College of Ireland

Supervisor: Pramod Pathak,Paul Stynes, Musfira Jilani

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Antony Yesudas
Student ID:	x20243405
Programme:	MSc in Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Pramod Pathak,Paul Stynes, Musfira Jilani
Submission Due Date:	15/08/2022
Project Title:	Configuration Manual
Word Count:	852
Page Count:	8

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:	17th September 2022

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

Antony Yesudas
x20243405

1 Introduction

This document aims to inform about the procedures required in putting the research project "A Machine Learning Framework to Predict Depression, Anxiety, and Stress" into practice. The configuration handbook outlines the precise steps taken to complete the research in detail. The study aims to ascertain whether machine learning techniques can reliably forecast mental diseases like depression, anxiety, and stress. We used three machine learning methods to detect DAS and compared their effectiveness. The configuration manual's organizational structure, which details the project's implementation phases, is given below:

- Section:2: System requirements: The system configuration tools and technologies used in the research will be described in this part.
- Section:3: Data acquisition: This part will discuss how and where we gathered the data for this research.
- Section:4: Data Preprocessing :This section will cover the implementation of several machine learning models, including data preparation and transformation.
- Section:5: Implementation:The steps taken to implement the machine learning algorithm will be covered in this section.
- Section:6: Conclusion:This section will cover the Configuration Manual's conclusion.

2 System requirements

The figure:1 illustrates the specific system setup that was used in the study.

Operating System:	Windows 10 Home
Installed Memory (RAM):	16.0 GB
Processor:	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz

Figure 1: System configuration

Python programming language has been used for the research's implementation, with Jupyter Notebook as the IDE. The following are the packages and libraries employed in the study: Python, Jupyter Notebook, Pandas, Numpy, Seaborn, Scikitlearn, Tensorflow, Keras, re,matplotlib.

3 Data acquisition

The steps taken to collect the research's data will be covered in this part. Our study uses data from an online poll created by the author (Lovibond Lovibond, 1995). Data was collected from online surveys between 2017 and 2019. DASS42 consists of 42 questions in total. The data came from the outcomes of an online survey. The dataset consists of 39,775 records and 172 features, including 84 variables related to the time it took to complete each question, the location of the item in the survey, and 42 columns for survey questions. Ten personality questions were included, as well as 16 columns for the word checklist and 3 for introelapse, testelapse, and surveyelapse. There are 17 additional columns for attributes, including those for race, education, orientation, nation, age, gender, and religion etc.Fig:2 shows dataset feature names.

```
dataset.columns.values
```

```
array(['"Q1A', 'Q1I', 'Q1E', 'Q2A', 'Q2I', 'Q2E', 'Q3A', 'Q3I', 'Q3E',
       'Q4A', 'Q4I', 'Q4E', 'Q5A', 'Q5I', 'Q5E', 'Q6A', 'Q6I', 'Q6E',
       'Q7A', 'Q7I', 'Q7E', 'Q8A', 'Q8I', 'Q8E', 'Q9A', 'Q9I', 'Q9E',
       'Q10A', 'Q10I', 'Q10E', 'Q11A', 'Q11I', 'Q11E', 'Q12A', 'Q12I',
       'Q12E', 'Q13A', 'Q13I', 'Q13E', 'Q14A', 'Q14I', 'Q14E', 'Q15A',
       'Q15I', 'Q15E', 'Q16A', 'Q16I', 'Q16E', 'Q17A', 'Q17I', 'Q17E',
       'Q18A', 'Q18I', 'Q18E', 'Q19A', 'Q19I', 'Q19E', 'Q20A', 'Q20I',
       'Q20E', 'Q21A', 'Q21I', 'Q21E', 'Q22A', 'Q22I', 'Q22E', 'Q23A',
       'Q23I', 'Q23E', 'Q24A', 'Q24I', 'Q24E', 'Q25A', 'Q25I', 'Q25E',
       'Q26A', 'Q26I', 'Q26E', 'Q27A', 'Q27I', 'Q27E', 'Q28A', 'Q28I',
       'Q28E', 'Q29A', 'Q29I', 'Q29E', 'Q30A', 'Q30I', 'Q30E', 'Q31A',
       'Q31I', 'Q31E', 'Q32A', 'Q32I', 'Q32E', 'Q33A', 'Q33I', 'Q33E',
       'Q34A', 'Q34I', 'Q34E', 'Q35A', 'Q35I', 'Q35E', 'Q36A', 'Q36I',
       'Q36E', 'Q37A', 'Q37I', 'Q37E', 'Q38A', 'Q38I', 'Q38E', 'Q39A',
       'Q39I', 'Q39E', 'Q40A', 'Q40I', 'Q40E', 'Q41A', 'Q41I', 'Q41E',
       'Q42A', 'Q42I', 'Q42E', 'country', 'source', 'introelapse',
       'testelapse', 'surveyelapse', 'TIPI1', 'TIPI2', 'TIPI3', 'TIPI4',
       'TIPI5', 'TIPI6', 'TIPI7', 'TIPI8', 'TIPI9', 'TIPI10', 'VCL1',
       'VCL2', 'VCL3', 'VCL4', 'VCL5', 'VCL6', 'VCL7', 'VCL8', 'VCL9',
       'VCL10', 'VCL11', 'VCL12', 'VCL13', 'VCL14', 'VCL15', 'VCL16',
       'education', 'urban', 'gender', 'engnat', 'age', 'screensize',
       'uniquenetworklocation', 'hand', 'religion', 'orientation', 'race',
       'voted', 'married', 'familysize', 'major",,,,'], dtype=object)
```

Figure 2: Dataset feature names

4 Data Preprocessing

In this section, we'll go over the pre-processing data for the study step by step.

4.1 Data Preparation

The data(fig:3), was collected from the online open-psychometrics data repository [1] and stored in local system.

[1] DataSource:`http://openpsychometrics.org/_rawdata/`

```
dataset = pd.read_csv("data.csv", sep=r'\t', engine='python')
```

```
dataset.head(10)
```

	"Q1A	Q1I	Q1E	Q2A	Q2I	Q2E	Q3A	Q3I	Q3E	Q4A	...	screensize	uniquenetworklocation	hand	religion	orientation	race	voted	married	familysize
0	"4	28	3890	4	25	2122	2	16	1944	4	...	1	1	1	12	1	10	2	1	2
1	"4	2	8118	1	36	2890	2	35	4777	3	...	2	1	2	7	0	70	2	1	4
2	"3	7	5784	1	33	4373	4	41	3242	1	...	2	1	1	4	3	60	1	1	3
3	"2	23	5081	3	11	6837	2	37	5521	1	...	2	1	2	4	5	70	2	1	5
4	"2	36	3215	2	13	7731	3	5	4156	4	...	2	2	3	10	1	10	2	1	4
5	"1	18	6116	1	28	3193	2	2	12542	1	...	2	1	1	4	1	70	2	1	4
6	"1	20	4325	1	34	4009	2	38	3604	3	...	2	1	1	7	2	60	2	1	4
7	"1	34	4796	1	9	2618	1	39	5823	1	...	2	1	1	2	2	60	1	1	2
8	"4	4	3470	4	14	2139	3	1	11043	4	...	1	1	1	12	2	70	2	1	4
9	"3	38	5187	2	28	2600	4	9	2015	1	...	2	1	1	2	2	60	2	1	3

10 rows × 172 columns

Figure 3: DAS Dataset

4.2 Transformation of Data

This part will go over how the primary data were transformed and how the Depression, Anxiety, and Stress dataset was extracted. Figure:4, Two Features contain unwanted punctuations, so we removed the punctuations with the help of 're' package as shown in the figure.

```
dataset['Q1A']
```

```
0         "4
1         "4
2         "3
3         "2
4         "2
         ..
39770     "2
39771     "3
39772     "2
39773     "3
39774     "2
Name: Q1A, Length: 39775, dtype: object
```

```
import string
def rem_punct(txt):
    txt_nopunt = "".join([c for c in txt if c not in string.punctuation])
    return txt_nopunt
```

```
dataset['Q1A'] = dataset['Q1A'].apply(lambda x: rem_punct(x))
```

Figure 4: Removing punctuations

Figure:5,'major' columns contains void values so replaced it with 'No Degree'.

```
dataset['major']=dataset['major'].replace('','No Degree')
```

```
dataset['major']
```

```
0                 No Degree
1                 No Degree
2                 No Degree
3                   biology
4                Psychology
```

Figure 5: Replacing ' ' with 'No Degree'

Later we removed position and time features of the questionnaire(Fig:6).

```
time = [i for i in data.iloc[:,0:126] if  'E' in i]
position = [i for i in data.iloc[:,0:126] if  'I' in i]
```

```
data=data.drop(position,axis=1)
data=data.drop(time,axis=1)
```

Figure 6: Dropped time and position features

We renamed ten personality questions and also the major degrees of the participants(Fig:7).

```
data_1=data_1.rename(columns={'TIPI1':'Extraverted-enthusiastic','TIPI2':'Critical-quarrelsome',
                        'TIPI3':'Dependable-self_disciplined','TIPI4':'Anxious-easily upset',
                        'TIPI5':'Open to new experiences-complex','TIPI6':'Reserved-quiet',
                        'TIPI7':'Sympathetic-warm','TIPI8':'Disorganized-careless','TIPI9':'Calm-emotionally_stable',
                        'TIPI10':'Conventional-uncreative'})
print('Shape:',data_1.shape)
print('Attributes:',data_1.columns)
```

Figure 7: Renamed ten personality questions

From the primary dataset(Fig:8), we extracted our Depression, Anxiety and Stress Datasets.

Depression, Anxiety and Stress : Datasets

```
def sub(data_2):
    return data_2.subtract(1,axis=1)
data_2=sub(data_2)
DASS_keys = {'Depression': [3, 5, 10, 13, 16, 17, 21, 24, 26, 31, 34, 37, 38, 42],
             'Anxiety': [2, 4, 7, 9, 15, 19, 20, 23, 25, 28, 30, 36, 40, 41],
             'Stress': [1, 6, 8, 11, 12, 14, 18, 22, 27, 29, 32, 33, 35, 39]}
Dep = []
for i in DASS_keys["Depression"]:
    Dep.append('Q'+str(i)+'A')
Stress = []
for i in DASS_keys["Stress"]:
    Stress.append('Q'+str(i)+'A')
Anx = []
for i in DASS_keys["Anxiety"]:
    Anx.append('Q'+str(i)+'A')
depression= data_2.filter(Dep)
stress = data_2.filter(Stress)
anxiety = data_2.filter(Anx)
```

Figure 8: DAS dataset creation

Next we created three Four news features: Age groups, Total count, Condition, Severity. Total count is created by the total row wise sum of questionnaire features(Fig:9).

```
def condition(x):
    if x<=9:
        return 'Normal'
    if  10<=x<=13:
        return 'Mild'
    if 14<=x<=20:
        return 'Moderate'
    if 21<=x<=27:
        return 'Severe'
    if x>28:
        return 'Extremely Severe'

Depression['Condition']=Depression['Total_Count'].apply(condition)
Depression.head()
```

Figure 9: Condition Feature.

Next we checked the correlation heatmap for each dataset (Fig:10).

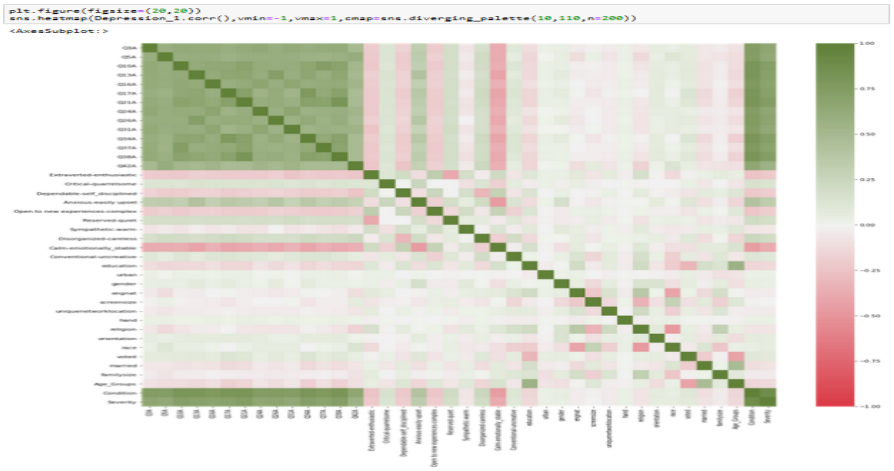

Figure 10: Depression : Correlation

Before the model creation, we scaled our dataset with MinMaxScaler and then applied PCA to the data(Fig:11), we find the number of components through the plot.

```
pca = PCA().fit(X_scaled)

%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (12,6)

fig, ax = plt.subplots()
xi = np.arange(1, 39, step=1)
y = np.cumsum(pca.explained_variance_ratio_)

plt.ylim(0.0,1.1)
plt.plot(xi, y, marker='o', linestyle='--', color='b')

plt.xlabel('Number of Components')
plt.xticks(np.arange(0, 39, step=1)) #change from 0-based array index to 1-based human-readable label
plt.ylabel('Cumulative variance (%)')
plt.title('The number of components needed to explain variance')

plt.axhline(y=0.95, color='r', linestyle='-')
plt.text(0.5, 0.85, '95% cut-off threshold', color = 'red', fontsize=16)

ax.grid(axis='x')
plt.show()
```

```
pca = PCA(n_components = 32)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
```

Figure 11: PCA

5 Implementation

The research is conducted with three machine learning algorithms, and we applied each algorithm to all three datasets. The Algorithms we implemented for our project are Random Forest, GaussianNB and Neural networks. For each dataset, we consider two cases, Case 1 with five severity levels and case 2 with a binary outcome.

5.1 Random Forest

On all three datasets, the Random Forest machine learning technique was used. The picture depicts the Random forest for Depression dataset's implementation(fig:12) and confusion matrix(fig:13).

```
RanFor=RandomForestClassifier(n_estimators=190,min_samples_split=3,min_samples_leaf=1,max_depth=160,max_features='auto').fit(X_tr
Acc_ran_1=round(accuracy_score(y_test,RanFor.predict(X_test_scaled)),3)
f1_ran=round(f1_score(y_test,RanFor.predict(X_test_scaled),average='weighted'),3)
recall_ran=round(recall_score(y_test,RanFor.predict(X_test_scaled),average='weighted'),3)
precision_ran=round(precision_score(y_test,RanFor.predict(X_test_scaled),average='weighted'),3)
print('Accuracy:',Acc_ran_1)
print('F1_Score:',f1_ran)
print('Recall_Score:',recall_ran)
print('Precision_Score:',precision_ran)
print('Cross Validation Score:',round(np.mean(cross_val_score(RanFor, X_train_scaled, y_train, cv = 6)),3))
classification=classification_report(
    digits=4,
    y_true=y_test,
    y_pred=RanFor.predict(X_test_scaled))
print(classification)
fig, ax = plt.subplots(figsize=(10, 10))
plot_confusion_matrix(RanFor,X_test_scaled,y_test,ax=ax,cmap = 'Greens')
```

Figure 12: Random Forest Implementation

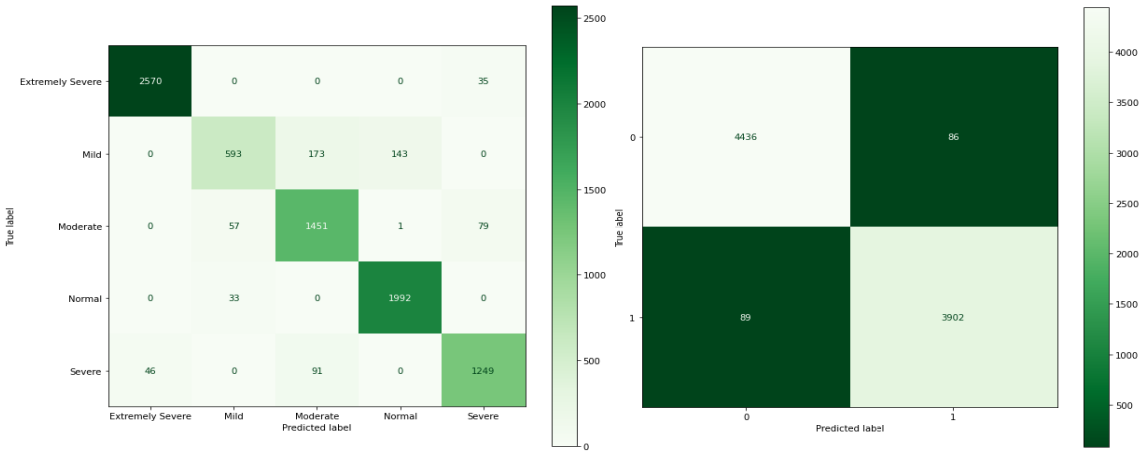

Figure 13: Random Forest : Confusion Matrix for Depression Dataset

5.2 Gaussian Naive Bayes

The machine learning method GaussianNB was applied to the three datasets. The figure depicts the GaussianNB for Stress dataset's implementation(fig:14) and confusion matrix(fig:15).

```
gb=GaussianNB().fit(X_train_scaled,y_train)
Acc_gb_1=round(accuracy_score(y_test,gb.predict(X_test_scaled)),3)
f1_gb=round(f1_score(y_test,gb.predict(X_test_scaled),average='weighted'),3)
recall_gb=round(recall_score(y_test,gb.predict(X_test_scaled),average='weighted'),3)
precision_gb=round(precision_score(y_test,gb.predict(X_test_scaled),average='weighted'),3)
print('Accuracy:',Acc_gb_1)
print('F1_Score:',f1_gb)
print('Recall_Score:',recall_gb)
print('Precision_Score:',precision_gb)
print('Cross Validation Score:',round(np.mean(cross_val_score(gb, X_train_scaled, y_train, cv = 6)),3))
classification=classification_report(
    digits=4,
    y_true=y_test,
    y_pred=gb.predict(X_test_scaled))
print(classification)
fig, ax = plt.subplots(figsize=(10, 10))
plot_confusion_matrix(gb,X_test_scaled,y_test,ax=ax,cmap = 'Blues')
```

Figure 14: Gaussian Naive Bayes Implementation

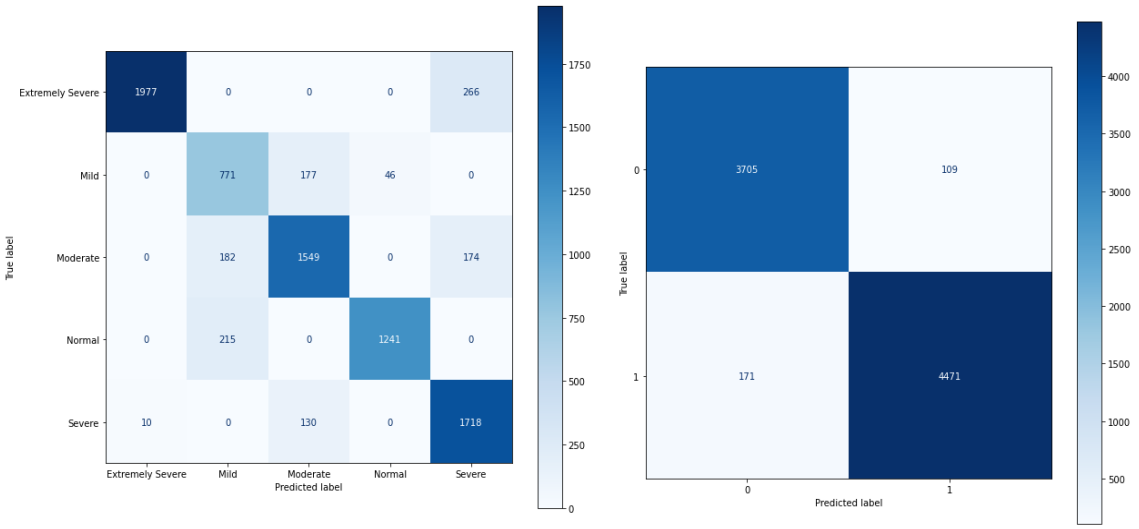

Figure 15: GaussianNB : Confusion Matrix for Stress Dataset

5.3 Neural Networks

On all three datasets, the neural networks machine learning technique was used. Figure displays the neural networks for anxiety implementation(fig:16) and confusion matrix(fig:17).

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(32,)),
    keras.layers.Dense(16, activation=tf.nn.relu),
    keras.layers.Dense(16, activation=tf.nn.relu),
    keras.layers.Dense(5, activation=tf.nn.softmax),
])
```

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['sparse_categorical_accuracy'])
```

Figure 16: Neural Networks Implementation

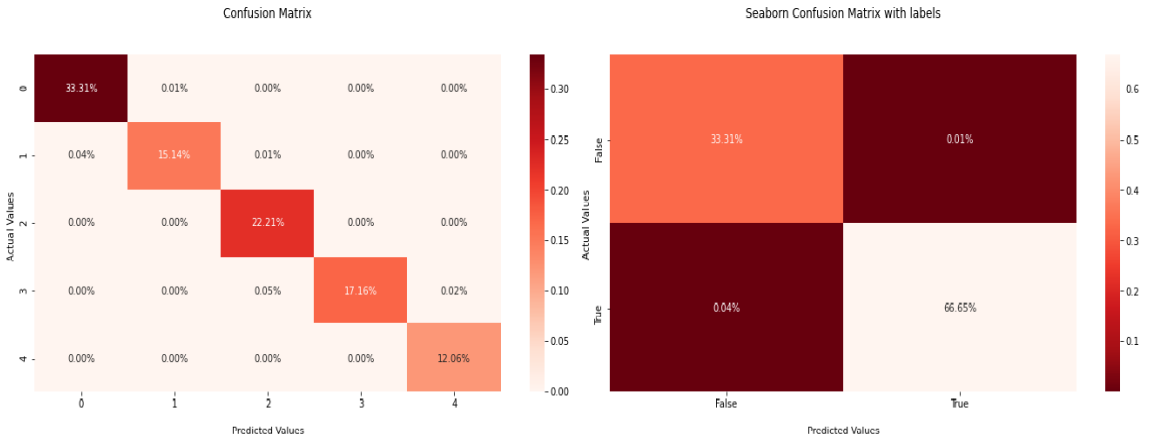

Figure 17: Neural Networks: Confusion Matrix for Anxiety Dataset

6 Conclusion

In conclusion, the data in this report shows how the research was applied fully and methodically. The report is divided into sections, each of which is thoroughly and methodically discussed.