National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Bhupinder Singh
Student ID: x22197982

School of Computing
National College of Ireland

Supervisor: Vikas Tomer

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Bhupinder Singh
Student ID:	x22197982
Programme:	Masters In Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Vikas Tomer
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	2558
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:	Bhupinder Singh
Date:	12/08/2024

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

Bhupinder Singh
x22197982

1 Introduction:

The main objective of this research paper is predicting the wine quality with high accuracy using different machine learning models like Random Forest, Gradient Boosting, XGBoost, SVM and Stacking Ensemble. This configuration manual basically provides detailed set of instructions for setting up the environment, then understanding the project structure in detail, then how to run the code and finally how to do trouble shooting if any issues are faced.

1.1 What is the purpose of this configuration manual:

The main Purpose of this configuration manual is to make sure that, the users are well known with setting up the necessary environment required to implement this project. Also, this manual helps to understand the structure of this project, also helps to understand how the code is running and what all are the minor requirements in the code which must be taken care to make sure the code runs. Basically, this Configuration manual is set of instructions which can be followed to replicate this work or build upon this work done in this project. Also, this configuration manual shows transparency in the working of this research Project, which make sure the Reproducibility of this research Project as well. By using this manual peer review and validation could be facilitated. And because of the transparency and reproducibility this manual helps others to replicate this research and build upon this search as well.

1.2 System Requirements:

These are the Minimum requirements which are necessary to have for the system to run and execute the project code successfully. There are two types of system requirements, Hardware and software requirements which have been discussed below in detail.

1. Hardware Requirements- Processor must be at least '13th Gen Intel Core i5-13500H' and above, then Memory must be '16GB RAM (15.6GB useable)' and above, and then the storage must be '500GB SSD' and above.

2. Software Requirements- Operating System must be Windows 11 Home Single Language Version 23HS and above, then Programming Language must be Python (Version - 3.12.4), and the development environment must be Jupyter Notebook.

1.3 What are the Python Dependencies:

There are many Frameworks and libraries which have been used in this project and these must be used within the code to make sure that all the required work is done.

1. Scikit-learn (version 0.24.1) must be used for used for Data Processing, Model Training and for Evaluation.
2. Pandas (version 1.2.3) must be used for Data analysis and for Data Manipulation.
3. Numpy (version 1.19.5) must be used for the Different Numerical computations.
4. Matplotlib (3.3.4) and Seaborn (0.11.1) must be used for different Data Visualization, mainly the ones done in Exploratory Data analysis section under methodology.
5. Xgboost (1.3.3) must be used to implement advance machine learning model like XGBoost.
6. Imbalanced-learn (version 0.8.0) must be used for SMOTE, to make sure that dataset is balanced.

1.4 Environmental setup:

These are the different steps which make sure that everything is ready to implement the code for the different machine learning models.

1.4.1 Installing the Python Programming language:

1. Installing python is the first step in this Environmental setup, it must be downloaded and installed using this site (Download Python for Windows).
2. During the installation of the python, it is important to make sure that "Add python to Path" is selected.
3. Then after this step, to make sure that installation has been successfully, 'python –version' command must be used in command prompt.

1.4.2 Creating a Virtual Environment:

1. This step optional but it is recommended, as this make sure that an isolated workspace is created for the project where different libraries could be installed specifically for that project.
2. This could be done using 'venv', using the command 'python -m venv myenv' to create this environment and then using the command '`myenv\Scripts\activate`' to activate this environment.
3. Also 'conda' can used for this purpose, using the command 'conda create –name myenv python =3.8' to create the environment and then using the command 'conda activate myenv' to activate this environment.
4. If the virtual environment is not used, the required libraries can be installed directly, this approach is also fine.

1.4.3 Installing the different libraries of python:

The required libraries must be installed manually in the starting itself to make sure that the later implementation of the code is smooth.

1. 'Pip install scikit – learn == 0.24.1'.
2. 'Pip install pandas == 1.2.3'.
3. 'Pip install numpy == 1.19.5'.
4. 'Pip install seaborn == 0.11.1'.
5. 'Pip install xgboost == 1.3.3'.
6. 'Pip install matplotlib == 3.3.4'.
7. 'Pip install imbalanced- learn == 0.8.0'.

1.4.4 Setting up the Jupyter Notebook:

Once the above libraries are installed successfully, then install the Jupyter Notebook with the help of the command 'pip install notebook'. Then after this installation the Jupyter notebook must be launched using command 'jupyter notebook'.

1.4.5 Verification of the installed libraries:

This is important because these libraries must be installed properly with correct version so that different Data analysis steps within the code are implemented successfully.Below are the command which must be used to check whether the installation is completed or not.

```
import sklearn
import pandas as pd
import numpy as np
import seaborn as sns
import xgboost as xgb

print("Scikit-learn:", sklearn.__version__)
print("Pandas:", pd.__version__)
print("NumPy:", np.__version__)
print("Seaborn:", sns.__version__)
print("XGBoost:", xgb.__version__)
```

Figure 1: Code for verification of installed libraries

2 Code Structure

The code of this research project is organized into different cells, each cell is having relevant comments within it. These comments make sure that anyone can understand, what exactly is happening in that cell code. Below are the different sections of the code.

1. Importing important Libraries- all the important libraries have been installed in the starting of each cell.

2. Different steps have been performed in the code, starting with combing of the 4 different datasets. This has been achieved using pandas, and this step of combing the 4 different datasets make sure that all the relevant data points are included for the further analysis.

3. The next step in the code is Featuring engineering, in this step a new variable is created as per the project requirements with help of other features and different heuristics. This step helps to optimize the dataset for model training in next steps ,Yao and Jia (2023); Mor et al. (2022); Di and Yang (2022); Pascua et al. (2023).

4. The combined dataset which additional feature in it, is reloaded and inspected to understand the structure of the data within the dataset. Also step helps to understand the distribution of datapoints within the dataset and for finding any anomalies if present.

5. Then of any outlier is detected, it has been removed with the help of Inter Quartile method (IQR). This step makes sure that these outliers doesn't have any impact on the different model performance in later stages when different models would be implemented on this dataset.

6. The different categorical features of the dataset are then encoded using One- Hot encoding, as this is the basic requirement of machine learning models.

7. Then the dataset is balanced using Synthetic Minority Over Sampling Technique (SMOTE). This step makes sure that there is no class imbalance with the target variable of the dataset.

8. Exploratory Data Analysis (EDA) is performed on the balanced dataset, steps like statistical Summary of numerical features, distribution plots of Numerical features, box plots of Numerical features vs target variable, count plots of categorical features vs target variable and correlation matrix have been used to understand the relationship between the different features of the dataset.

9. Then the dataset is split into training and test subsets, where training sets is 80% and testing set is of 20%.

10. Different machine learning models like Random Forest, Gradient Boosting, XG-Boost and SVM are trained on the dataset with full features sets and with reduced features sets.

11. Each model is hyperparameter tuned, to obtain the maximum accuracy. All the best performing models are combined with help of stacking ensemble model to obtain the maximum accuracy in wine prediction.

Design specifications:

2.1 Workflow:

This workflow basically talks about different steps which have been performed from starting of the Project till last to get the desired wine quality prediction.

1. Data Collection – Datasets were collected from different online sources for different features of the wine.

2. Data Processing – Different Datasets was combined to get a common dataset. Pre-Processing steps like cleaning, Normalization and handling missing values and outliers were performed on the combined Dataset.

3. Base Learners models like Random Forest, Gradient Boosting, XGBoost and SVM were trained on the pre-processed dataset using Gross Validation techniques.

4. Hyperparameter Tunning of each model was done using GridSearchCV to get optimal results.

5. Predictions from the above base Learners were used to train the Logistic Regression meta-Learner.

6. Model's performances were evaluated using different metrics like Accuracy, recall, Precision and F1-score.

2.2 Model Parameters:

1. Base Learners - Random Forest model was trained with (100, 200, 300, 400, 500) as estimators and with (None, 10, 20, 30, 40, 50) as maximum depth.

2. Gradient Boosting model was trained with (100, 200, 300, 400, 500) as estimators, with different learning rate of (0.01, 0.05, 0.1, 0.2) and with (3, 4, 5, 6, 7) as maximum depth.

3. XGBoost model was trained with (100, 200, 500) as estimators, with different learning rate of (0.01, 0.05, 0.1) and maximum depth as (3, 5, 7, 10).

4. Cross Validation technique with 3 Folds were used for the hyperparameter Tunning.

5. Stacking Ensemble model was trained with 100 estimators and random state 42 for random forest , then for Gradient Boosting, with 500 estimators and learning rate of 0.05 and maximum depth of 7 and random state of 42 , then for XGBosst with 500 estimators and learning rate of 0.1 and maximum depth of 10 and random state of 42.

6. Meta Learner – Logistic regression model was used as meta learner, using the prediction of above base learners model combined to get the final prediction.

7. Evaluation Metrics- Different Metrics like accuracy, Precision, recall and F1 score were used to evaluate the performance of different models used.

3 Running of the code

3.1 Execution of the code:

Now open the terminal and then activate the virtual environment if required, after this go to the Project Directory and start running the Jupyter notebook with command 'Jupyter notebook'. Once this is done open the notebook which is having the code for the research Project and the cells one by one sequentially.

3.2 Output:

Observe the output from each cell as this would help to understand the flow of the project and if any changes are required within the code of the cell, changes can be made by seeing the output. Then the performance evaluation metrics of each model could be observed form these outputs (Accuracy, Precision, Recall, F1-score).

3.3 Customization of the code:

The code can be customized to use it for different dataset with number of features and to do hyperparameter tunning of the existing models to get better results.

3.4 Extension of this project:

1. **Adding new Models:** New models cloud be used within this code structure as this is achievable by integrating the new models in the existing pipelines of this code. New models could be defined within the relevant section of the report, this could be achieved by adding new cell within the existing code and these new models could be included in the ensemble later.

2. **Feature Engineering:** Newer feature could be added in the dataset by using knowledge from existing research within this topic or by using domain specific knowledge gained by some expert. As this feature engineering would enhance the overall performance of machine learning models and more accuracy could be achieved in wine quality assessment.

3. **Advance machine learning techniques:** These advance machine learning techniques like different parameters for hyperparameter tunning, different cross validation techniques, using deep learning models could be explored to improve overall performance.

3.5 Troubleshooting:

These are some challenges which might come while executing the code.

1. **Library Version conflicts:** This could happen if the version of the required libraries is different from what has been used and discussed above. To mitigate this correct version of each library must be installed prior to the execution of the code.

2. **Memory Error:** If the execution time is more and the memory error occur during the execution of different models, the dataset size must be reduced, or the batch processing approach must be used. As this would help to execute the code in limited time.

3.6 Conclusion:

So, by following all the above steps in sequence as discussed above, the environment could be set up to execute the project code successfully. The results could be replicated using the Jupyter Notebook of this research Project, and customization of this project can be done by using different datasets, models and parameters.

References

Di, S. and Yang, Y. (2022). Prediction of red wine quality using one-dimensional convolutional neural networks, *arXiv preprint arXiv:2208.14008* .

Mor, N. S., Asras, T., Gal, E., Demasia, T., Tarab, E., Ezekiel, N., Nikapros, O., Semimufar, O., Gladky, E., Karpenko, M. et al. (2022). Wine quality and type prediction from physicochemical properties using neural networks for machine learning: a free software for winemakers and customers., *AgriRxiv* (2022): 20220051475.

Pascua, K. B., Lagura, H. D., Lumacad, G. S., Pensona, A. K. N. and Jalop, M. J. I. (2023). Combined synthetic minority oversampling technique and deep neural network for red wine quality prediction, *2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*, IEEE, pp. 609–614.

Yao, R. and Jia, G. (2023). Machine learning aided inverse design for new wine, *2023 4th International Conference on Industrial Engineering and Artificial Intelligence (IEAI)*, IEEE, pp. 51–61.