National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Nayana Palathinkal Shibu
Student ID: x23174960

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nayana Palathinkal Shibu		
Student ID:	x23174960		
Programme:	Data Analytics	**Year:**	2024
Module:	MSc Research Project		
Lecturer:	Noel Cosgrave		
Submission Due Date:	12/12/2024		
Project Title:	Configuration Manual		
Word Count:	504	**Page Count:**	5

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: Nayana Palathinkal Shibu

Date: 11/12/2024

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

Nayana Palathinkal Shibu
Student ID: x23174960

1 Introduction

This configuration manual demonstrates the implementation of the project "Enhanced Customer Behaviour Analysis Using Stacking Classifiers for Churn Prediction". The main objective of this project is to predict customers' behaviour on churn using different machine learning algorithms like Logistic Regression, AdaBoost Classifier, XGBoost Classifier and Stacking Classifier. This manual includes the details on system configuration, including hardware and software specifications, and the libraries required and used to implement the project.

2 Hardware Specification

Operating System	Windows 11 Home
Processor	Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz
RAM	16.0 GB
System Type	64-bit operating system, x64-based processor

3 Software Specification

Programming Language	Python 3.11.5
Tool	Jupyter Notebook

4 Importing Required Libraries

Purpose	**Libraries**
Warnings Management	warnings
Data Manipulation	pandas
Visualization	plotly.express, matplotlib.pyplot, seaborn

Preprocessing	OneHotEncoder from sklearn.preprocessing, SMOTE from imblearn.over_sampling
Feature Importance	ensemble from sklearn, numpy
Dataset Splitting	train_test_split from sklearn.model_selection
Modelling	LogisticRegression from sklearn.linear_model, AdaBoost Classifier from sklearn.ensemble, XGBClassifier from xgboost, StackingClassifier from mlxtend.classifier
Cross-Validation and Hyperparameter Tuning	GridSearchCV and cross_val_score from sklearn.model_selection
Model Evaluation	confusion_matrix, classification_report and accuracy_score from sklearn.metrics

The listed libraries are imported as needed during the implementation to ensure clarity and avoid unnecessary imports, as seen in Figure 1.

Figure 1: Importing pandas for loading the dataset.

Similarly, other libraries are imported in the respective sections of the implementation.

5 Version Check of Libraries

Figure 2 given below shows the code for checking whether the listed libraries are installed and can be imported, along with their versions. Figure 3 is the output for the code in Figure 2 that verifies the installation and the version of the listed Python libraries. So, Figure 4 gives the code for installing packages based on the versions used in the implementation of this project and it is seen in the initial phase of code as a comment. This ensures that all the required libraries are installed and correctly imported for running the project without any errors and helps in reproducibility.

NOTE: If you do not wish to update the versions of the libraries, download the specified dependencies as a '.txt' file and create a virtual environment and use it to run the notebooks associated with this project. Or this file can be run in Google Colab without the need for installing or upgrading the versions of the packages used in this project.

In [5]:

```python
#libraries
libraries={
    'warnings': 'warnings',
    'pandas': 'pandas',
    'plotly.express': 'plotly.express',
    'matplotlib.pyplot': 'matplotlib.pyplot',
    'seaborn': 'seaborn',
    'sklearn.preprocessing.OneHotEncoder': 'sklearn.preprocessing',
    'imblearn.over_sampling.SMOTE': 'imblearn.over_sampling',
    'sklearn.ensemble': 'sklearn.ensemble',
    'numpy': 'numpy',
    'sklearn.model_selection.train_test_split': 'sklearn.model_selection',
    'sklearn.linear_model.LogisticRegression': 'sklearn.linear_model',
    'sklearn.ensemble.AdaBoostClassifier': 'sklearn.ensemble',
    'xgboost.XGBClassifier': 'xgboost',
    'mlxtend.classifier.StackingClassifier': 'mlxtend.classifier',
    'sklearn.model_selection.GridSearchCV': 'sklearn.model_selection',
    'sklearn.model_selection.cross_val_score': 'sklearn.model_selection',
    'sklearn.metrics.confusion_matrix': 'sklearn.metrics',
    'sklearn.metrics.classification_report': 'sklearn.metrics',
    'sklearn.metrics.accuracy_score': 'sklearn.metrics'
}

#checking installation, import libraries and display versions
for lib, import_name in libraries.items():
    try:
        base_module=import_name.split('.')[0]
        module= __import__(base_module)
        version=getattr(module, '__version__', 'Version: Not Available')
        print(f"{lib} from '{import_name}'. Version: {version}")
    except ModuleNotFoundError:
        print(f"{lib} is not installed.")
    except ImportError:
        print(f"{lib} could not be imported.")

```

Figure 2: Code for checking library installation and versions.

```
warnings from 'warnings'. Version: Version: Not Available
pandas from 'pandas'. Version: 2.0.3
plotly.express from 'plotly.express'. Version: 5.9.0
matplotlib.pyplot from 'matplotlib.pyplot'. Version: 3.7.2
seaborn from 'seaborn'. Version: 0.12.2
sklearn.preprocessing.OneHotEncoder from 'sklearn.preprocessing'. Version: 1.5.1
imblearn.over_sampling.SMOTE from 'imblearn.over_sampling'. Version: 0.12.3
sklearn.ensemble from 'sklearn.ensemble'. Version: 1.5.1
numpy from 'numpy'. Version: 1.26.4
sklearn.model_selection.train_test_split from 'sklearn.model_selection'. Version: 1.5.1
sklearn.linear_model.LogisticRegression from 'sklearn.linear_model'. Version: 1.5.1
sklearn.ensemble.AdaBoostClassifier from 'sklearn.ensemble'. Version: 1.5.1
xgboost.XGBClassifier from 'xgboost'. Version: 2.1.2
mlxtend.classifier.StackingClassifier from 'mlxtend.classifier'. Version: 0.23.1
sklearn.model_selection.GridSearchCV from 'sklearn.model_selection'. Version: 1.5.1
sklearn.model_selection.cross_val_score from 'sklearn.model_selection'. Version: 1.5.1
sklearn.metrics.confusion_matrix from 'sklearn.metrics'. Version: 1.5.1
sklearn.metrics.classification_report from 'sklearn.metrics'. Version: 1.5.1
sklearn.metrics.accuracy_score from 'sklearn.metrics'. Version: 1.5.1
```

Figure 3: Python libraries and versions.

In [59]:

```python
import subprocess
import pkg_resources

#list of required packages with their versions
required_packages={
    "pandas": "2.0.3",
    "plotly": "5.9.0",
    "matplotlib": "3.7.2",
    "seaborn": "0.12.2",
    "scikit-learn": "1.5.1",
    "imbalanced-learn": "0.12.3",
    "numpy": "1.26.4",
    "xgboost": "2.1.2",
    "mlxtend": "0.23.1",
}

#function to install or upgrade the package to the specified version
def install_package(package, version):
    subprocess.check_call(["pip", "install", f"{package}=={version}"])

#function to Check each package's version and install if not matching
def check_and_install():
    for package, required_version in required_packages.items():
        try:
            installed_version=pkg_resources.get_distribution(package).version
            if installed_version != required_version:
                print(f"{package} version mismatch")
                install_package(package, required_version)
            else:
                print(f"{package} is matching")
        except pkg_resources.DistributionNotFound:
            print(f"{package} not installed.")
            install_package(package, required_version)

if __name__ == "__main__":
    check_and_install()
```

Figure 4: Python Code to install and upgrade package version.

6 Project Code Artefacts

Figure 5 given below shows the project folder which consists of the dataset in CSV format, ‘Dataset’ and the project code, ‘Research Code’ of this project. It has one folder inside, ‘Comparison’, which has the dataset in XLSX format, ‘Comparison_Dataset’ and the code, ‘Code’ as seen in Figure 6 which is used for the comparative study to evaluate the performance of the proposed system by comparing it with another system using this dataset, ‘Comparison_Dataset’.

Figure 5: Project folder.

Figure 6: Comparison folder.