National College of Ireland

Configuration Manual

MSc Research Project
MSc. Data Analytics

Samiksha Tripathi
Student ID: x23102055

School of Computing
National College of Ireland

Supervisor: Teerath Kumar Menghwar

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Samiksha Tripathi
Student ID:	x23102055
Programme:	MSc. Data Analytics
Year:	2023-2024
Module:	MSc Research Project
Supervisor:	Teerath Kumar Menghwar
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	358
Page Count:	10

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:	Samiksha Tripathi
Date:	12th August 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

Samiksha Tripathi
x23102055

1 Introduction

The main use of this manual is to log the technical procedure of the research. It contains information about the tools, environment and code configurations utilised for the study of this project. The manual also contains a few snippets of the code artefact that include certain configurations specific to the project. The aim of the manual is to provide steps for reproducing similar work and extend the scope of the research. Section 2 discusses the environment used for executing the project. Section 3 contains information about the data collection process. Section 4 contains details of initial analysis and model implementation. Section 5 discusses the experiments performed in the project and their results.

2 Environment

The hardware used for the study of this project is shown in Figure 1. It has Processor of 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz with an Installed RAM 20.0 GB (19.8 GB usable) and operates on a System type 64-bit operating system, x64-based processor.

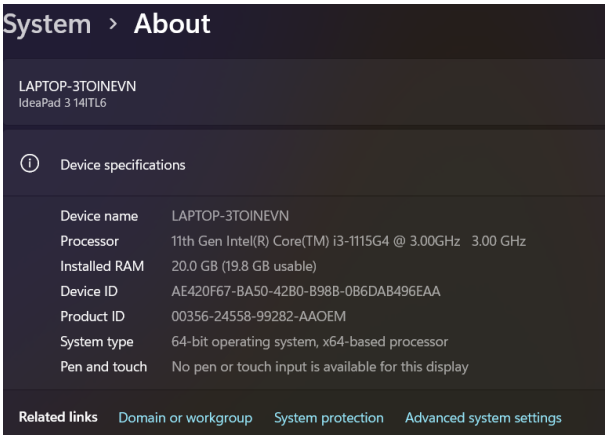


Figure 1: System Configuration

The integrated development environment used for coding was Visual Studio. Jupyter Notebook as editor was used for legible log of codes. To aid the system computation and increase the speed of execution, use of Google Colab[1] was done.

[1]Google Colab.

3 Data Collection

The Building Stock data for Ireland published by Sustainable Energy Authority of Ireland (SEAI) [2] is available for download along with the User Information guide that contains the data description of the dataset as shown in Figure 2

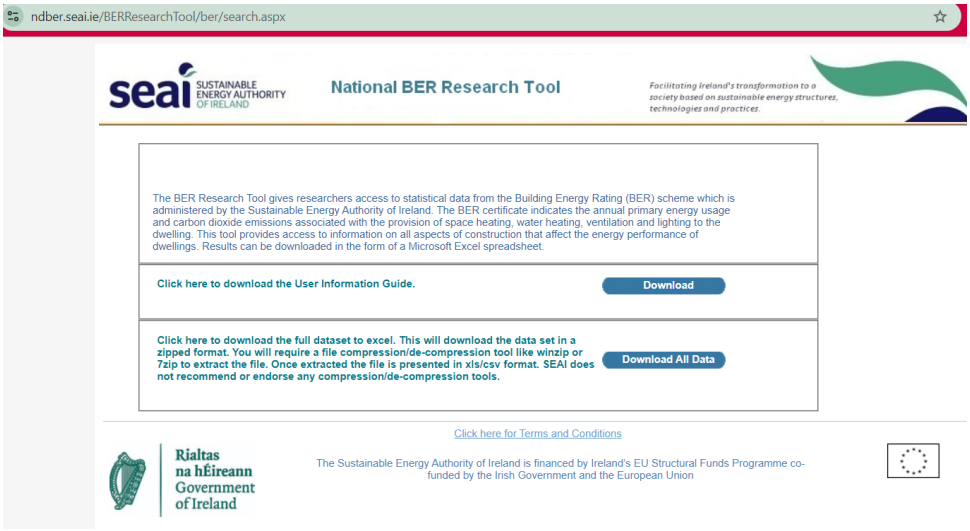


Figure 2: Dataset Location

4 Library and Package installation

Figure 3, shows the two packages to be installed before running the code.

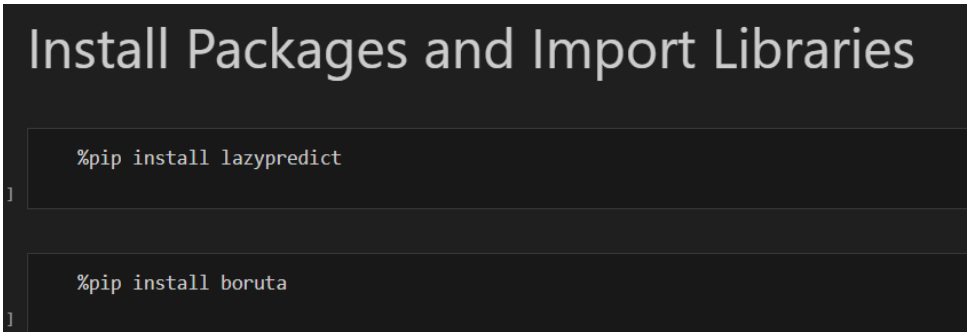


Figure 3: Installed Packages

Figure 4, shows the libraries to be imported for various methods in the code.

The datset has 211 columns Sustainable Energy Authority of Ireland (SEAI) (2024), along with user guide. After loading the data, 50 columns are dropped from the dataset based on domain knowledge. Figure 9

[2]Sustainable Energy Authority of Ireland .

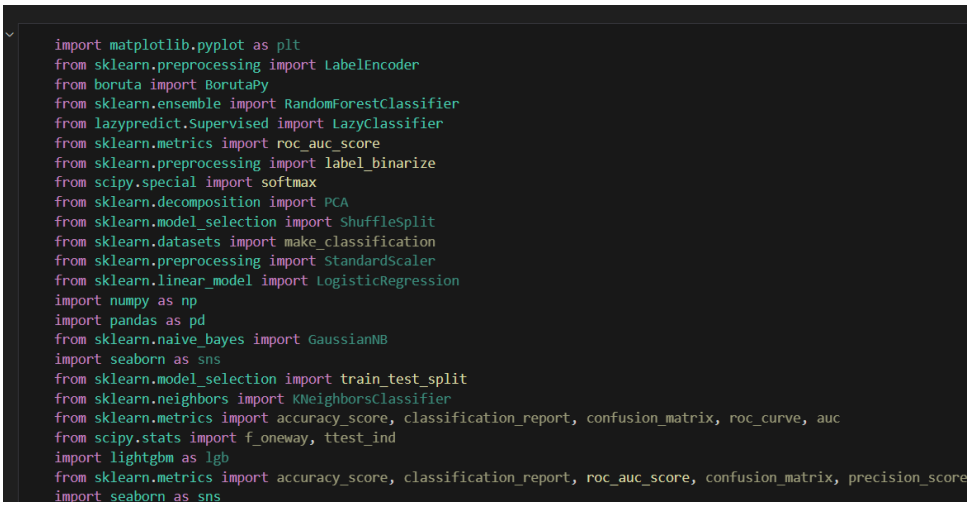


Figure 4: Imported Libraries

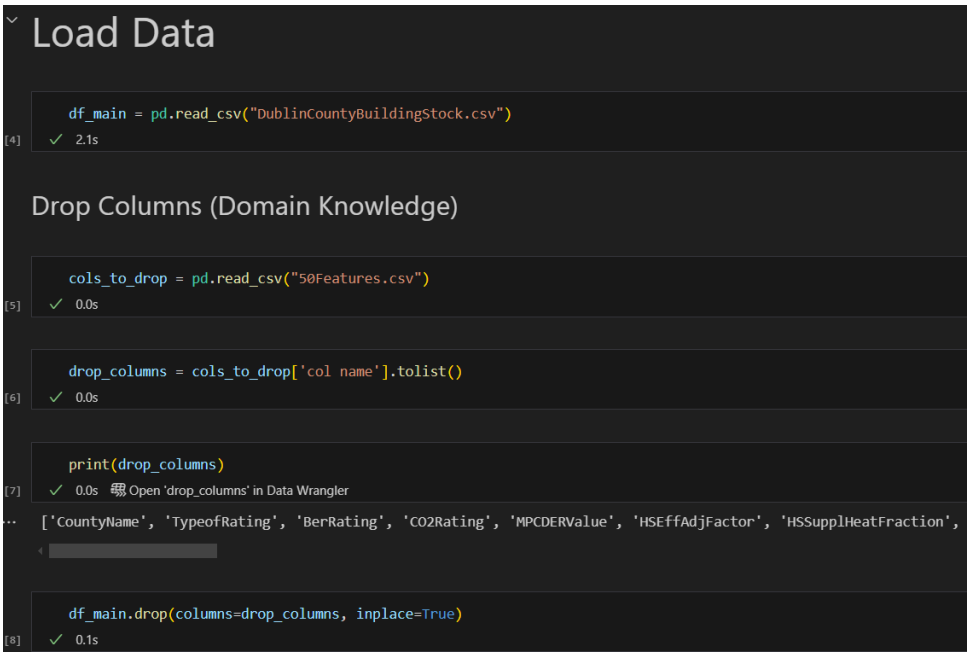
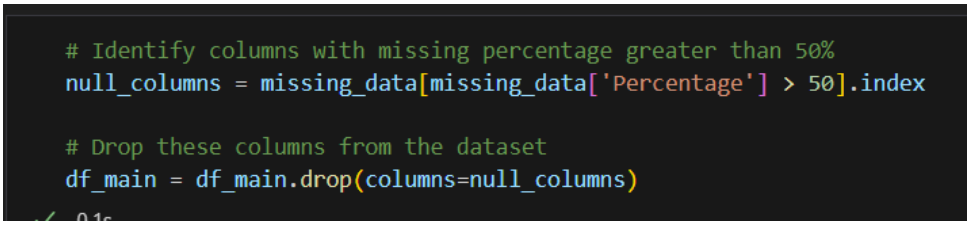


Figure 5: Domain Feature Selection

5 Data Cleaning and Processing

Figure 6, shows the handling of missing values by dropping features that are 50 percent empty.



```
# Identify columns with missing percentage greater than 50%
null_columns = missing_data[missing_data['Percentage'] > 50].index

# Drop these columns from the dataset
df_main = df_main.drop(columns=null_columns)
```

Figure 6: Missing value handling

Figure 7, shows the method used for filling up the remaining empty columns.

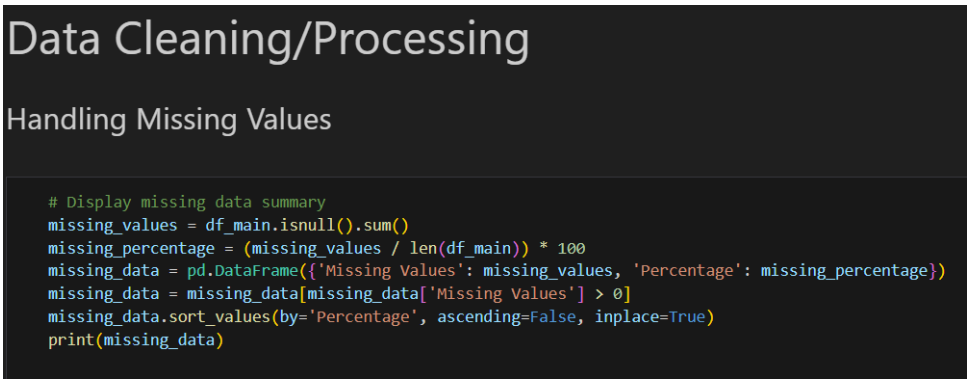


Figure 7: Substituting Missing Values

Figure 8, shows the method used for transforming the data using Label Encoding.
Figure **??**, shows the boruta method used for features selection.

6 Modelling

Figure 10, shows the method used for splitting the dataset into train and test for ML modelling.

Figure 11, shows the implementation of 25 ML models using the package LazyPredict.
Figure 12, shows the model implementation for LGBM ML model.

7 Experiments

Figure 13, Figure 14, Figure 15, Figure 16, show the experiments performed in the Retrofit analysis part of the research.

8 Statistical Inference

Figure 17, shows the statistical inference of the performed experiments.

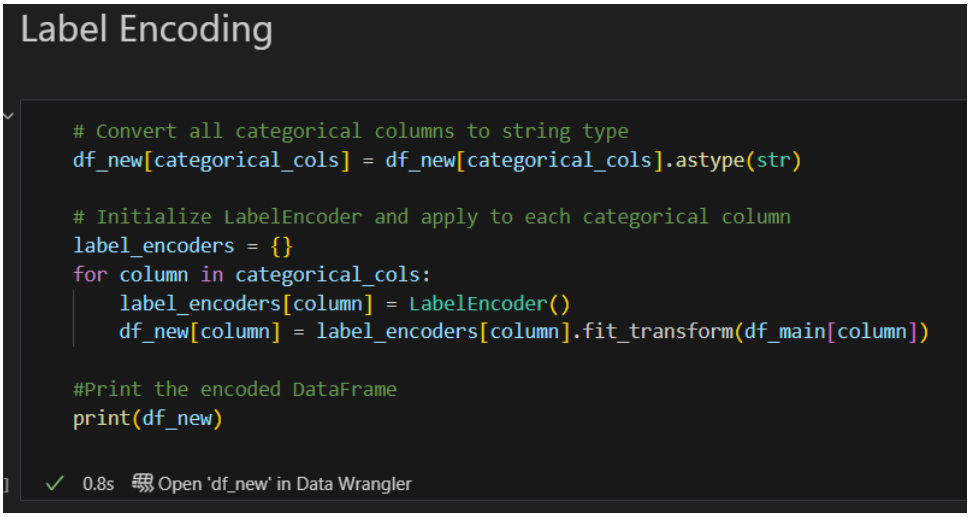


Figure 8: Data transformation

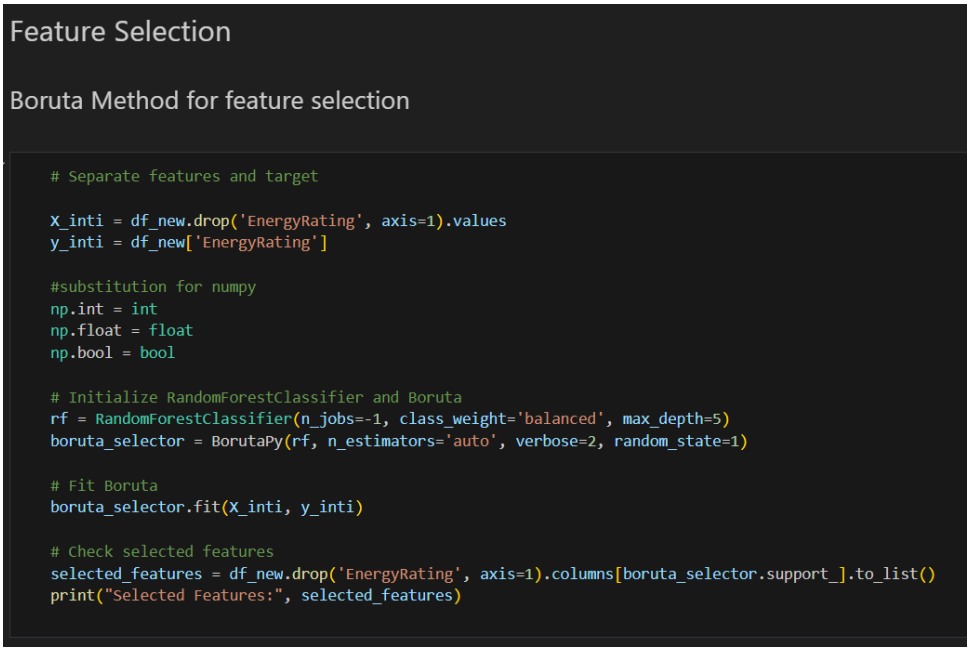


Figure 9: Boruta Feature Selection

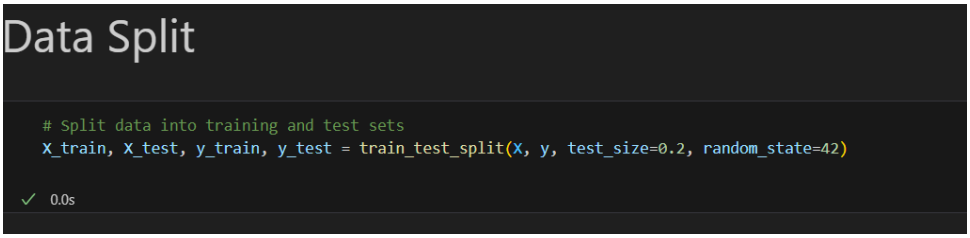


Figure 10: Data split configuration

Applying ML Models

Lazy Predict

```
# Apply LazyClassifier
clf = LazyClassifier(verbose=0, ignore_warnings=True, custom_metric=None)
models, predictions = clf.fit(X_train, X_test, y_train, y_test)

# Custom function to calculate multi-class ROC AUC
def compute_multiclass_roc_auc(y_true, y_pred_proba, average='macro'):
    y_true_binarized = label_binarize(y_true, classes=np.unique(y_true))
    return roc_auc_score(y_true_binarized, y_pred_proba, average=average)

# Custom function to get probabilities or decision scores
def get_proba_or_decision_function(model, X_test):
    if hasattr(model, 'predict_proba'):
        return model.predict_proba(X_test)
    elif hasattr(model, 'decision_function'):
        decision_scores = model.decision_function(X_test)
        if decision_scores.ndim == 1:  # Binary classification case
            decision_scores = np.vstack([1 - decision_scores, decision_scores]).T
        return softmax(decision_scores, axis=1)
    else:
        raise AttributeError(f"Model {model} does not have predict_proba or decision_function.")
```

Figure 11: Lazy Predict

LGBMClassifier

```
# Initialize LightGBM model
model = lgb.LGBMClassifier()

# Train the model
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)
y_pred_proba = model.predict_proba(X_test)[:, 1]  # Probability estimates for the positive class

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Classification Report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Figure 12: LGBM implementation

Figure 13: Experiment1

Figure 14: Experiment2

Figure 15: Experiment3

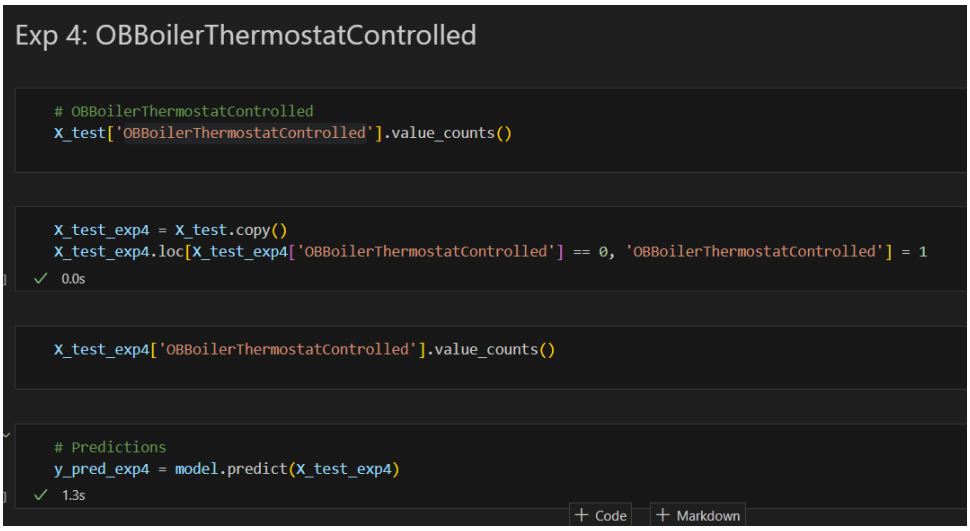


Figure 16: Experiment4

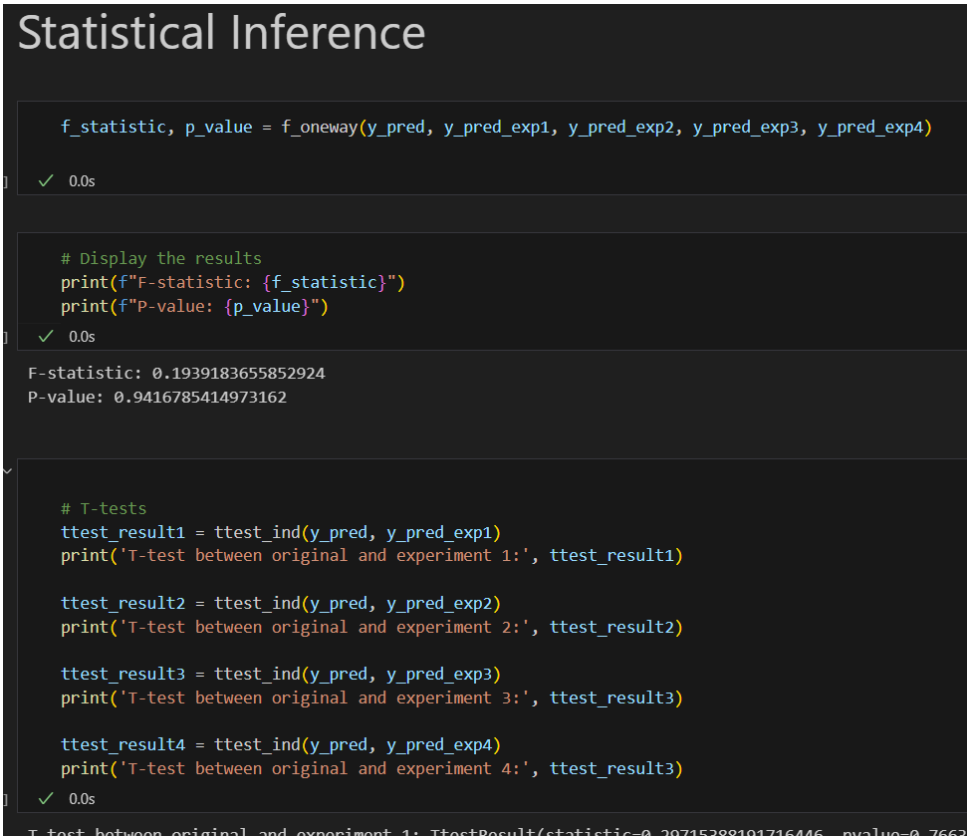


Figure 17: Statistical Inference

References

Sustainable Energy Authority of Ireland (SEAI) (2024). Ber research tool, https://ndber.seai.ie/BERResearchTool/ber/search.aspx. Accessed: 2024-08-12.