National College of Ireland

# Configuration Manual

MSc Research Project
Programme Name

Rajas Abhijit Zende
Student ID: x23352779

School of Computing
National College of Ireland

Supervisor: Dr Muslim Jameel Sayed

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** Rajas Abhijit Zende

**Student ID:** x23352779

**Programme:** MSc in artificial intelligence for business **Year:** 2024-2025

**Module:** …… MSC research project

**Lecturer:** ……Dr Muslim Jameel Sayeed

**Submission Due Date:** 15/09/25

**Project Title:** Evaluating the Impact of remote work on Employee Productivity and Satisfaction using Machine Learning Approaches

**Word Count:** 258 words **Page Count:** ……3 pages

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:** Rajas Abhijit Zende

**Date:** 11/08/25

**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

Rajas Abhijit Zende
Student ID: x23352779

## 1 Overview.

This configuration manual offers specific instructions on how to set up the python development environment and configurations to process the employee productivity and performance dataset.

This study was carried out on windows 11, on a intel Processor 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz, 2688 Mhz, 6 Cores.

Minimum system requirements to run the code are as follows

**Operating system Requirements**. Windows 10/11, macOS 10.14

**RAM requirements** minimum 8gb of required

**Storage requirements :** at least 2gb of free storage space is required

**Processor requirements :** Recommended intel i7 ( intel i5 can work as well)

**Python version 3.12.4 or higher.**

**Libraries required for implementing the code.**

1. Pandas (1,5.3) : Required for Data analysis and manipulation.
2. Numpy (1.24.3) Required for numerical calculations.
3. Scikit-learn (1.3.0) required for importing machine learning algorithms.
4. Matplotlib (3.7.1) required for plotting and visualizatons.
5. Seaborn (0.12.2) required for data visualization.
6. Jupyter notebook : development environment.

```
# Importing dataset ( is in the SAME folder as this Python/Jupyter file)
df =pd.read_csv("D:/NCI Study/thesis folder/Weekly Reports/Extended_Employee_Performance_and_Productivity_Data.csv")
```

## Data Exploration

Replace for the dataset path with your own dataset path which is uploaded in the zip file import the dataset and run the code

Run the below code for importing the necessary libraries

```
import pandas as pd
import numpy as np
import sklearn
import lightgbm as lgb
import matplotlib.pyplot as plt
import seaborn as sns

print("Pandas version:", pd.__version__)
print("NumPy version:", np.__version__)
print("Scikit-learn version:", sklearn.__version__)
print("LightGBM version:", lgb.__version__)
print("Matplotlib version:", plt.matplotlib.__version__)
print("All libraries imported successfully!")
```

```
Pandas version: 2.2.2
NumPy version: 1.26.4
Scikit-learn version: 1.4.2
LightGBM version: 4.6.0
Matplotlib version: 3.8.4
All libraries imported successfully!
```

**Import the necessary libraries for implementing random forest model**

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error


# Importing Random forest model

X_train, X_test, y_train, y_test = train_test_split(X_final, y_final, test_size=0.2, random_state=42)

rf_model = RandomForestRegressor(n_estimators=400, max_depth=8, random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)

print(" Baseline Random Forest:")
print(f"R²: {r2_score(y_test, rf_pred):.4f}")
print(f"MAE: {mean_absolute_error(y_test, rf_pred):.2f}")
print(f"MSE: {mean_squared_error(y_test, rf_pred):.2f}")
```

```
 Baseline Random Forest:
R²: 0.3103
MAE: 0.71
MSE: 0.76
```

**Parameter confiuaraton of  Hyper tunedRandom forest models.**

Use the below paramerts to use the hyper tuned random forest model. And also import grid search CV test combinations of the hypertuned random forest model.

```
# Hypertuning the random forest Regressor for best results
from sklearn.model_selection import GridSearchCV


param_grid = {
    'n_estimators': [250,],
    'max_depth': [None,],
    'min_samples_split': [2,5,],
    'min_samples_leaf': [1,2,],
    'max_features': ['sqrt', 'log2']
}

# Setting up the grid search

rf = RandomForestRegressor(random_state=42)
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid,
                           cv=3, n_jobs=-1, scoring='r2', verbose=2)

grid_search.fit(X_train, y_train)
best_rf = grid_search.best_estimator_
y_pred_rf = best_rf.predict(X_test)
print("Test R² score:", r2_score(y_test, y_pred_rf))
```

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
Test R² score: 0.5174596885903501
```

For importing the light GBM model import LGBM regressor from lightgbm.

```
#implementing Light gradient boosting machine (LGBM) Regressor
from lightgbm import LGBMRegressor


lgbm_model = LGBMRegressor(
    boosting_type='gbdt',
    n_estimators= 250,
    learning_rate=0.6,
    max_depth=7,
    subsample=0.5,
    colsample_bytree=0.5,
    random_state=42,
     reg_alpha=0.8,
    reg_lambda=0.8,
    force_row_wise = True,
)


lgbm_model.fit(X_train, y_train)
lgbm_pred = lgbm_model.predict(X_test)



print(f"R²: {r2_score(y_test, lgbm_pred):.4f}")
print(f"MAE: {mean_absolute_error(y_test, lgbm_pred):.2f}")
print(f"MSE: {mean_squared_error(y_test, lgbm_pred):.2f}")
```

```
[LightGBM] [Info] Total Bins 879
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 7
[LightGBM] [Info] Start training from score 2.999730
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
R²: 0.8099
MAE: 0.36
MSE: 0.21
```

# 2 Evaluation libraries.

For evaluating the performance of models and splitting the data into testing and training import the below libraries called train_test_split, r2_score, mean_absolute_error, mean_squared_error.

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error


# Importing Random forest model

X_train, X_test, y_train, y_test = train_test_split(X_final, y_final, test_size=0.2, random_state=42)
```

Make sure that the XlsxWriter library is installed on your local machine to enable saving results to a new Excel file. If not already installed, use pip install XlsxWriter.

```
pip install XlsxWriter
```

```
Requirement already satisfied: XlsxWriter in c:\users\raj\anaconda3\lib\site-packages (3.2.5)
Note: you may need to restart the kernel to use updated packages.
```

```
import xlsxwriter

# Creating a new excel file
workbook = xlsxwriter.Workbook('Efficiency_Predictions.xlsx')
worksheet = workbook.add_worksheet('Results')

# Write headers of file
worksheet.write('A1', 'Actual Efficiency Score')
worksheet.write('B1', 'Predicted Efficiency Score')
```