National College of Ireland

A Machine Learning approach for Predicting Corporate ESG Ratings and role of Country ESG data on Prediction.

Configuration Manual

MSc Research Project
Data Analysis

Gurpreet Kaur Bhuie
Student ID: x21231061

School of Computing
National College of Ireland

Supervisor:Athanasios Staikopoulos

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Gurpreet Kaur Bhuie……

Student ID: ……x21231061…….……

Programme: ……Data Analysis…… **Year:** …2023-2024

Module: ……Research Project…….……

Lecturer: …… Athanasios Staikopoulos…….……

Submission Due Date: ……14/12/2023…….……

Project Title: A Machine Learning approach for Predicting Corporate ESG Ratings and role of Country ESG data on prediction

Word Count: …… **Page Count:** ……..……

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: …Gurpreet Kaur Bhuie……

Date: ……14/12/2023……

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

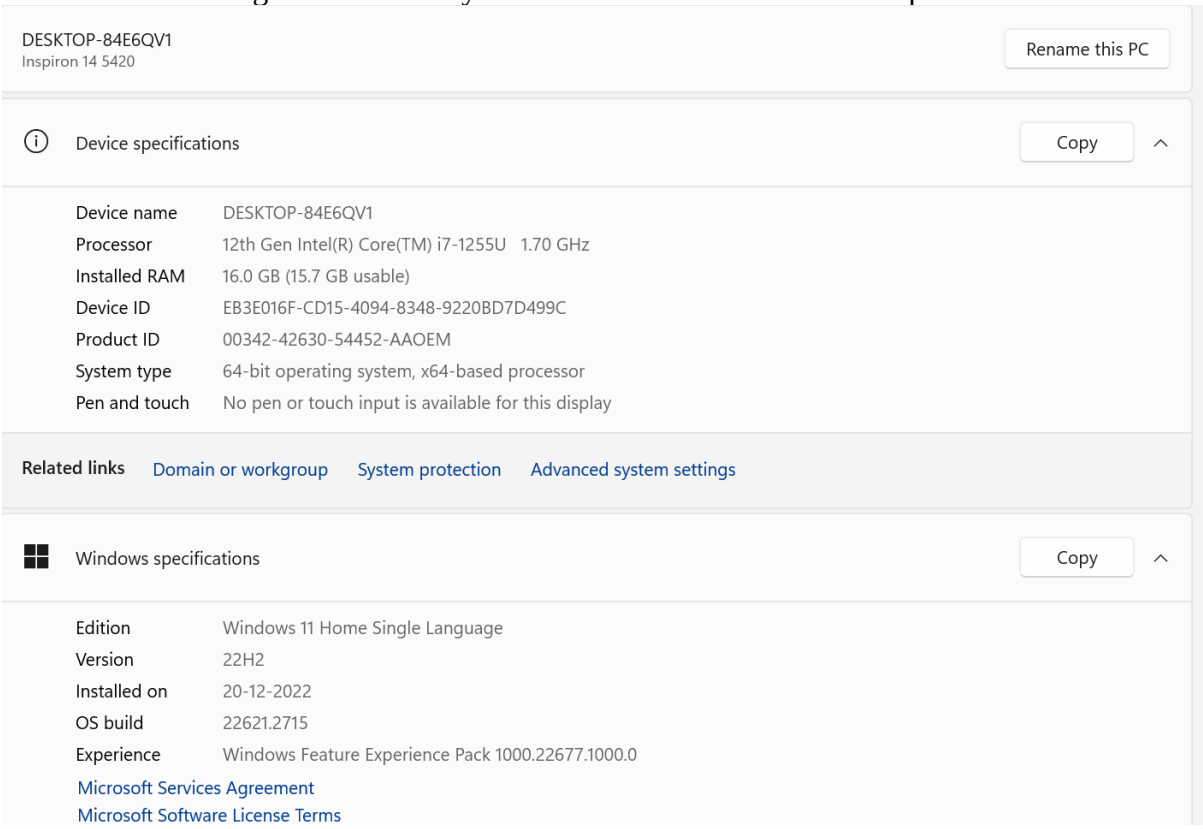
Gurpreet Kaur Bhuie
Student ID: x21231061

1 Introduction

This document contains the detailed instruction on how to replicate the experiment. The configuration manual discusses the machine requirement needed to build and run this model. The steps to all the installation required are mentioned in this document.
This experiment requires postgresql for uploading the data and joining it. However this step can be avoided as transformed spreadsheets is also attached in the code artifact.

2 Hardware configuration

The hardware configuration of the system used to build and run this experiment is as follows:



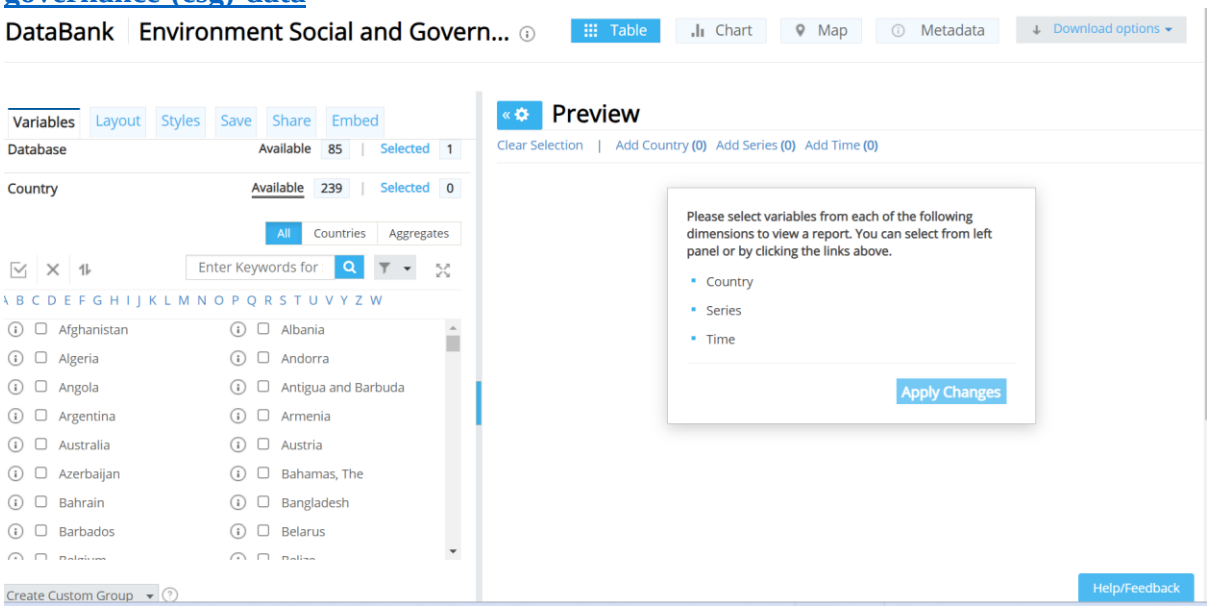
3 Project Files

This section describes the project files needed to replicate the experiment.

Pre-requisite:

Postgresql should be up and running. Make sure the database is connected based on the connection string or modify the connection string as needed.

Dataset:
Country ESG Data: https://databank.worldbank.org/source/environment-social-and-governance-(esg)-data



Company historical ESG dataset:
https://www.bloomberg.com/professional/product/esg-data/?utm_medium=Adwords_SEM&utm_source=pdsrch&utm_content=APAC_ESGdata_2023&utm_campaign=728003&tactic=728003&gad_source=1&gclid=Cj0KCQiAyeWrBhDDARIsAGP1mWRT63GRKgkB_g9A2sLUUuN82xbnKHNXgC9v4wFeWdXNXAOk_1ZhReaAoctEALw_wcB
https://github.com/MinghanWang1995/ESG-Rating-and-Green-Revenue-Analysis/tree/master/Excel%20Files

Code:
Zip file that contains below files to be executed in same order:

1. ESG_data.ipynb
2. Country_ESG_Data.ipynb
3. Country_ESG_toDB.ipynb
4. Random Forest Regression on ESG data – Final modelling and evaluation

4 Software used:

- Microsoft Excel for maintain initial dataset.
- Jupyter Notebook for coding the model and evaluation.
- Postgresql for maintain data in table and joining using sql.

5 Replicating the experiment:

- Import the libraries and read data

In [1]:
```
import pandas as pd
import numpy as np
from statsmodels.tsa.vector_ar.var_model import VAR
from sklearn.preprocessing import LabelEncoder
import pandas as pd
import seaborn as sns
import psycopg2
from psycopg2 import sql
from statsmodels.tsa.arima.model import ARIMA
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error, r2_score
```

In [2]:
```
file_path = 'ESG Historical Score.xlsx'


df = pd.read_excel(file_path)

# Display DataFrame
print(df)
```

- Create table in postgresql and push data into tables
  3 Tables were created and data was uploaded in each of them
  Country_ESG_Data_new contains all the ESG datapoints related to country
  ESG_DATA contains all the ESG datapoints related to company
  Country_info table contains country name, iso code and country code, this table will help join the other 2 table as mentioned in the query below.

```
import pandas as pd
import psycopg2

# Connect to the PostgreSQL database
conn = psycopg2.connect(
    host="localhost",
    database="postgres",
    user="dap",
    password="dap"
)


table_name = 'Country_ESG_DATA_NEW'

# Read Excel file into a pandas DataFrame
excel_file_path = 'transposed_data.xlsx'
df = pd.read_excel(excel_file_path)

# Create a cursor object to execute PostgreSQL commands
cur = conn.cursor()

# Create table in PostgreSQL if it doesn't exist
cur.execute('''
    CREATE TABLE IF NOT EXISTS {} (
        "Country_Name" VARCHAR,
        "Country_Code" VARCHAR,
        "Year" INTEGER,
        "Access to_clean_fuels and technologies for cooking (% of population)" FLOAT,
        "Access to electricity (% of population)" FLOAT,
        "Annualized average growth rate in per capita real survey mean consumption or income, total population (%)" FLOAT,
        "Cause of death, by communicable diseases and maternal, prenatal and nutrition conditions (% of total)" FLOAT,
        "Children in employment, total (% of children ages 7-14)" FLOAT,
        "Fertility rate, total (births per woman)" FLOAT,
```

- Join the table and pull data back into excel for replication – this step was introduced so it is easy to replicate the experiment even if database step is skipped.

```python
import pandas as pd
import psycopg2
from sqlalchemy import create_engine

# Connect to the PostgreSQL database
conn = psycopg2.connect(
    host="localhost",
    database="postgres",
    user="dap",
    password="dap"
)

# Create a database connection
engine = create_engine("postgresql+psycopg2://dap:dap@localhost/postgres")

# SQL Query
sql_query = """
    SELECT *
    FROM esg_data e
    LEFT JOIN country_info ci ON e."ISO Code" = ci."iso_code"
    LEFT JOIN country_esg_data_new ce ON ci."country_name" = ce."Country_Name"
                                    AND EXTRACT(YEAR FROM e.months) = ce."Year"
    WHERE ce."Year" IS NOT NULL
"""

# Execute the query and fetch the results into a pandas DataFrame
df = pd.read_sql(sql_query, engine)

# Save the DataFrame to a CSV file
df.to_csv('output_data.csv', index=False)

# Close the database connection
conn.close()
```

- Modeling and evaluation:
  First the target variables are extracted into a variable called target_variable. Target variable contains e,s,g which we are trying to predict.
  Next the loop to train and evaluate the model in iterated through target_variable and data is split into test and train to achive this.
  PCA is applied from dimensionality reduction
  The importance of the paramaters is also assessed.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.decomposition import PCA

# Loop over target variables
for target_variable in target_variables:
    # Convert target variable to numeric, replacing non-numeric values with NaN
    df[target_variable] = pd.to_numeric(df[target_variable], errors='coerce')

    # Dropping rows with NaN values in the target variable
    df = df.dropna(subset=[target_variable])

    # Selecting only numeric columns for PCA
    numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
    X_numeric = df[numeric_columns]

    # One-hot encoding for categorical columns
    categorical_columns = df.select_dtypes(include=['object']).columns
    df_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)

    # Combine numeric and encoded categorical columns
    X = pd.concat([X_numeric, df_encoded], axis=1)

    # Splitting the data into train and test sets after one-hot encoding
    X_train, X_test, y_train, y_test = train_test_split(X, df[target_variable], test_size=0.2, random_state=42)

    # Determine the optimal number of components using the elbow method
    explained_variance = []
    for n_components in range(1, min(X_train.shape[0], X_train.shape[1])):
```