Configuration Manual for Energy Load Prediction Across Multi-European Countries

MSc Research Project
Data Analytics

Kranthi Boyini
Student ID: 22245219

School of Computing
National College of Ireland

Supervisor: Jaswinder Singh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Kranthi Boyini
Student ID:	22245219
Programme:	Masters in Data Analytics **Year:** 2024
Module:	MSc Research Project
Lecturer:	Jaswinder Singh
Submission Due Date:	07-11-2024
Project Title:	Energy Load Prediction Across Multi-European Countries
Word Count:	847 **Page Count:** 6

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: Kranthi Boyini

Date: 07-11-2024

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

Kranthi Boyini
Student ID: 22245219

1. INTRODUCTION

1.1 **Overview of the Energy Demand Forecasting Model**

This configuration manual describes a high-performance energy demand forecasting model designed to support power grid stability and optimize energy resource allocation. By integrating machine learning, advanced time-series analysis techniques, and explainability methods, the model accurately predicts future energy demands, enabling stakeholders to anticipate fluctuations and make informed decisions for energy distribution.

Key features of the model include:

- Utilization of various machine learning algorithms such as Linear Regression, Decision Tree, Random Forest, and XGBoost for precise demand prediction.
- Custom preprocessing steps, including resampling and normalization, to handle large-scale time-series data effectively.
- Feature engineering that captures essential patterns in energy consumption and generation data.
- Model explainability through SHAP (SHapley Additive exPlanations) values to interpret the impact of each feature on model predictions, enhancing transparency and trust in the model's outputs.

The model architecture consists of:

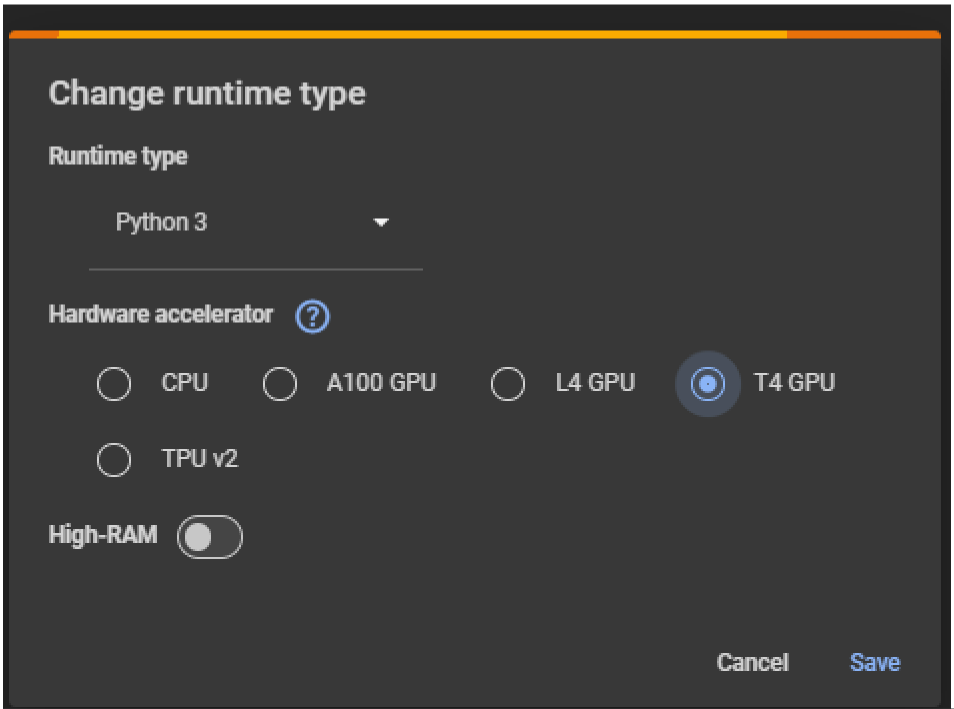
1. Preprocessing Pipeline: It takes care of missing values, normalization of data, and resampling of time-series data for proper training of models.
2. Feature Engineering Layer: This generates features like renewable power usage percentage and traditional power usage percentage that describe the trend of energy.
3. Multi-Model Training: Various models have been trained and evaluated separately, for which hyperparameter tuning has been applied to extract the best out of each model.
4. SHAP Analysis Layer: With SHAP values, feature importance can be learned, enabling the stakeholders to understand which factors most dictate the demand predictions.
5. Model Evaluation Layer: It uses quantification of the MAE and MSE accuracy metrics, which stands for mean absolute error and mean square error, respectively.
6. Prediction and Visualization: Final model predictions are visualized side by side with actual demand, and SHAP summaries emphasize the most important features for interpretability.

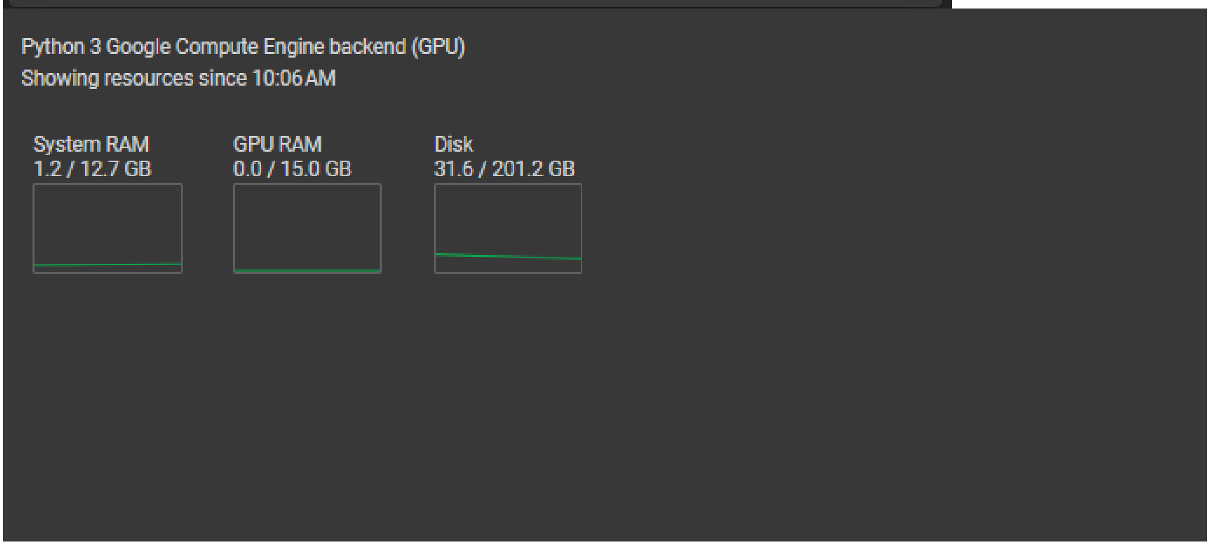
This forecasting model will serve to enhance the energy management strategy by allowing for correct demand forecast and proactive measures that can be taken in order to assure grid reliability. The SHAP values incorporated into the model provide better interpretability through which the stakeholders can draw much valuable inferences from the drivers of energy demand and aid in making data-driven decisions.

1 Connecting to a Colab Session and Drive setup

1.1 Colab session setup

1. Go to https://colab.research.google.com/
2. If prompted, sign in with your Google account
3. Click on "New Notebook" to start a new session
4. Ensure you're using a GPU runtime:
   - Go to "Runtime" > "Change runtime type"
   - Select "GPU" from the Hardware accelerator dropdown menu
   - Click "Save"



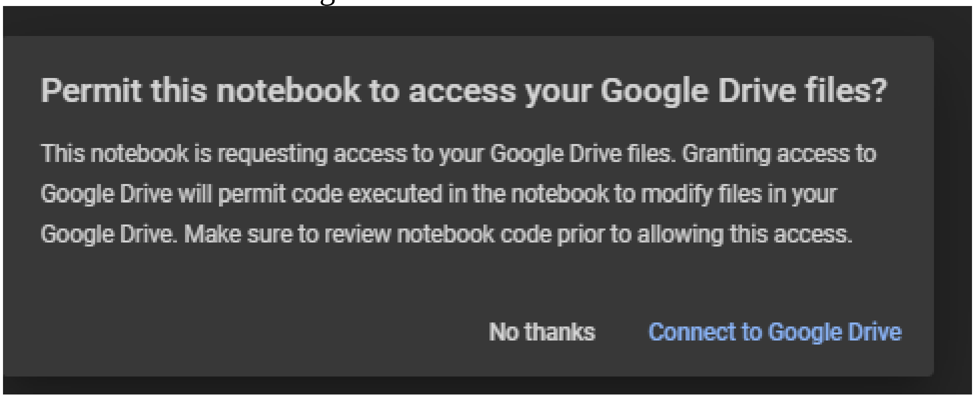


1.2 Drive connection

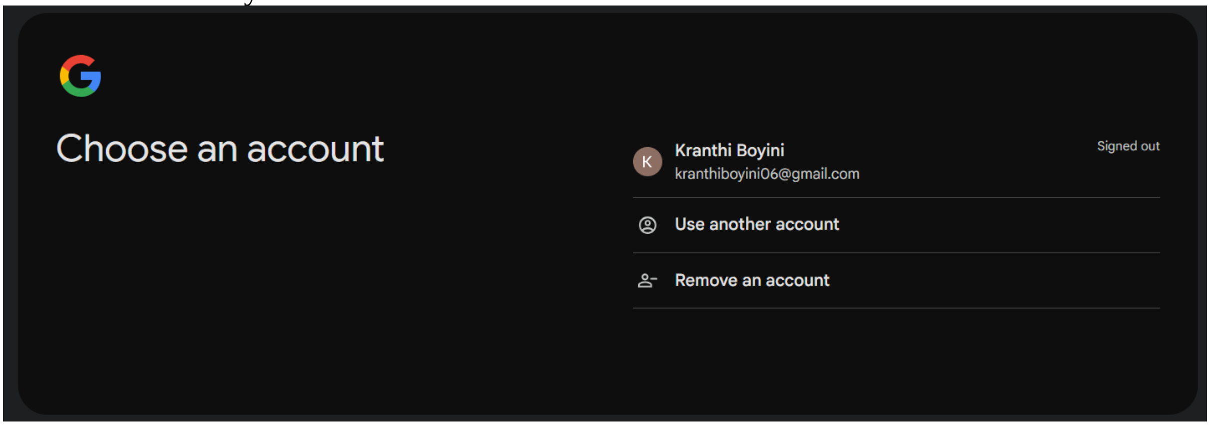
1. Run the following code to mount your Google Drive:

from google.colab import drive
drive.mount('/content/drive')

2. Follow the authentication prompts to give Colab access to your Google Drive as follow and select Connect to Google Drive



3. Select the account you want to connect to as follows



2 Uploading Files to Google Drive

Before you start with Google Colab, you need to upload the provided files to your Google Drive:

1. Go to https://drive.google.com/ and sign in with your Google account.
2. In your Google Drive, create a new folder called "EnergyPrediction".
3. Inside "EnergyPrediction", create the following subfolders:
   - Models
   - Datasets
4. Upload the files as follows:
   - Upload "weather_data.csv", "time_series_60min_singleindex.csv", "ninja_pv_wind_profiles_singleindex.csv","final_data.csv", "modelling_data.csv" to Datasets Folder.

   - Upload "decision_tree_model.pkl", "linear_regression_model.pkl", "decision_tree_model.pkl", "xgboost_model.pkl". to "Model" folder.

3 Changing the Paths

When working with the Energy Prediction model in Google Colab, it's crucial to set up the correct paths to your model and dataset. These paths will point to the locations in your Google Drive where you've stored the necessary files.

Now change the paths as in the following as per the above directory structure

3.1 Below is for the model file:

```
[16] loaded_results = {}
     for name in models.keys():
         print(f"Loading model: {name}")
         loaded_model = joblib.load(f"drive/MyDrive/EnergyPrediction/Models/{name.replace(' ', '_').lower()}_model.pkl")

         # Perform inference using the loaded model
         y_new_test_pred = loaded_model.predict(X_test_scaled)
         loaded_results[name] = {
             'model': loaded_model,
             'y_new_test_pred': y_new_test_pred
         }
         print(f"Predictions for {name} loaded model completed.\n")
```

3.2 Change dataset in the dataset_path variable

```
[3] modelling_data = pd.read_csv("drive/MyDrive/EnergyPrediction/Datasets/modelling_data.csv")
    modelling_data = modelling_data.drop(columns=["Unnamed: 0"],axis=1)
    modelling_data.head()
```

4 Running the Energy Prediction Models

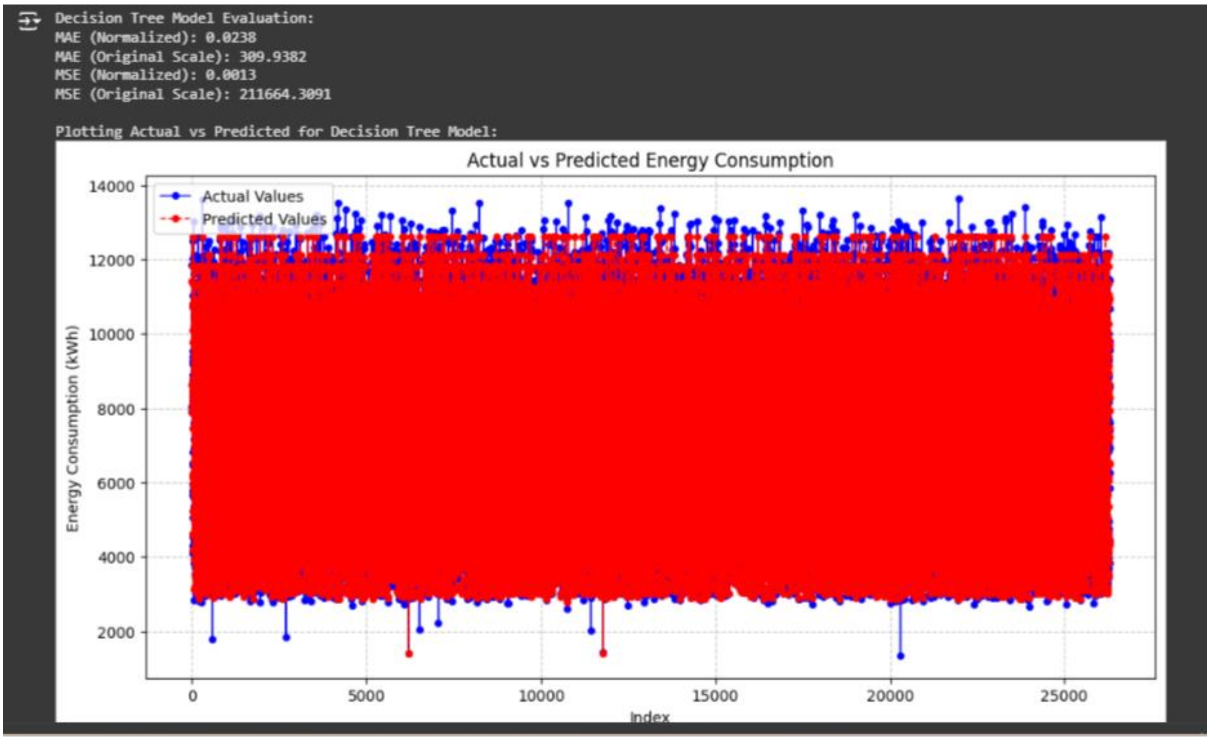
After you've successfully uploaded the files and configured the paths, you can execute the code as follows.

You can execute cells in several ways:

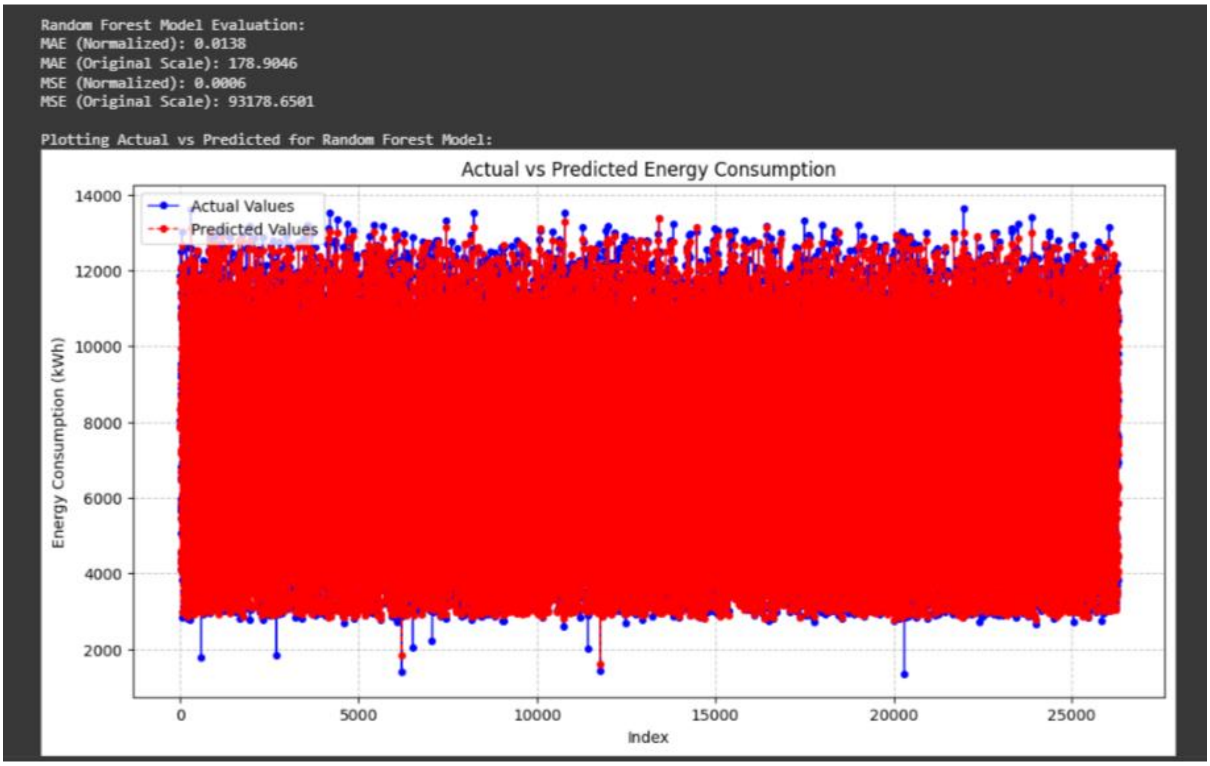
1. **Run a Single Cell**: Click on the cell you want to execute and either:
   - Press **Shift + Enter** on your keyboard.
   - Click the **Play button** (▶) on the left side of the cell.
2. **Run All Cells**: To run all cells in the notebook from top to bottom:
   - Go to **Runtime** in the top menu and select **Run all**.
3. **Run from a Specific Cell to the End**: If you want to run all cells starting from a specific cell to the last cell in the notebook:
   - Click on the cell where you want to start.
   - Go to **Runtime** > **Run after**.
4. **Run a Cell Again**: If you've made changes to the code or need to re-execute a specific cell:
   - Just press **Shift + Enter** or click the **Play button** again.
5. **Run Cells by Keyboard Shortcut**: You can also use additional shortcuts for running cells:
   - **Ctrl + Enter**: Runs the selected cell without moving to the next cell.
   - **Alt + Enter**: Runs the selected cell and inserts a new cell below.

5 Results :

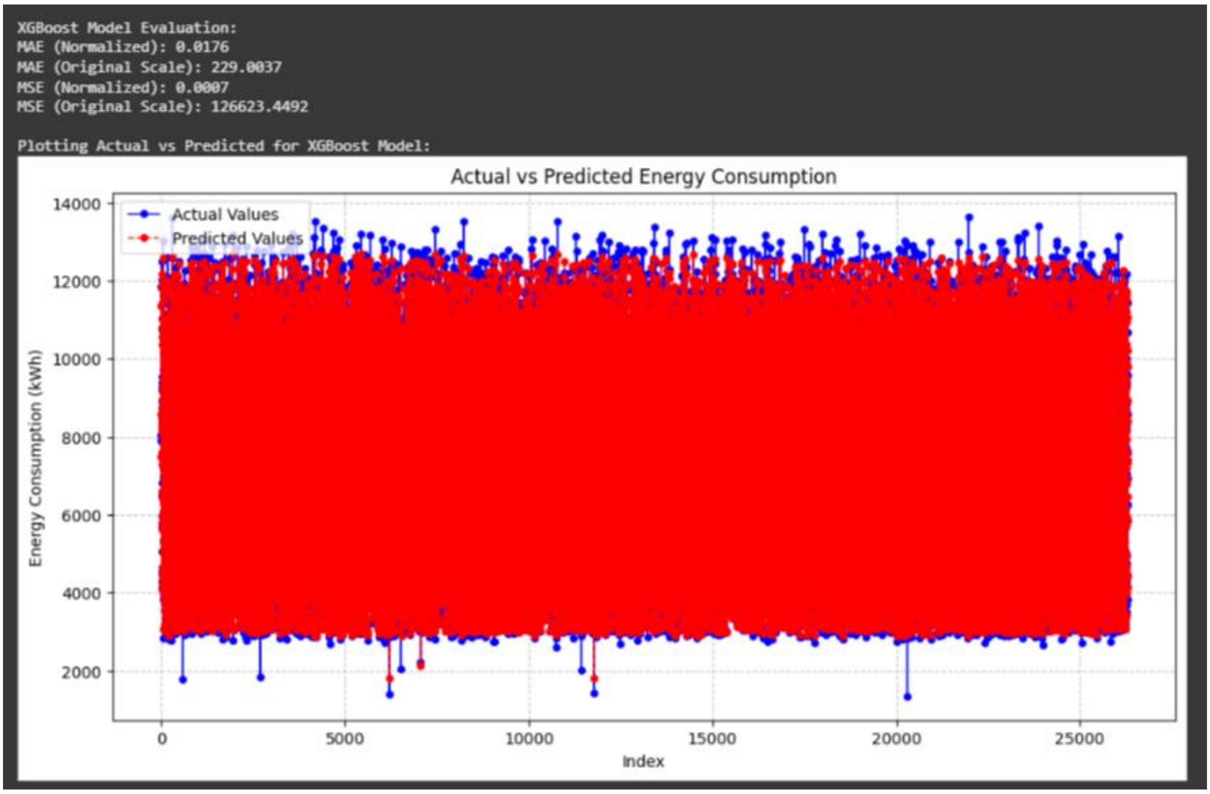
5.1 Decision Tree Regressor



5.2 Random Forest Regressor



5.3 XG Boost Regressor



References

Google Colab. (n.d.). https://colab.research.google.com/ .