RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi

Go, change the world

CoE in AI Research and Business Solutions

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Energy Grid Load Forecasting Using ML

SUMMER INTERNSHIP REPORT

Submitted by

VIBHAV SIMHA G	**1RV22CS230**
KUSHAAL S	**1RV22AI047**
TALASHILA DHEERAJ	**1RV22CS216**
SAMVIT SANAT GERSAPPA	**1RV22CS175**

Under the guidance of

Prof. Priya TV
Assistant Professor, Dept. of AIML
Dept. of AIML
RV College of Engineering®
Bengaluru

2022-2023

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

AND

MACHINE LEARNING

Bengaluru– 560059

CERTIFICATE

Certified that the Internship work titled **'Energy Grid Load Forecasting Using ML'** is carried out by **Vibhav Simha G (1RV22CS230), Kushaal S (1RV22AI047), Talashila Dheeraj (1RV22CS216) and Samvit Sanat Gersappa (1RV22CS175)** in partial fulfilment for the requirement of degree of **Bachelor of Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2022-2023. The work is carried out at **Centre of Excellence in AI Research and Business Solutions**, Dept. of AI & ML, RVCE. It is certified that all corrections/suggestions indicated during the review have been incorporated in the report.

Prof. Priya TV	**Prof. Somesh Nandi**	**Dr. B. Sathish Babu**
Assistant Prof	**Coordinator-Internship**	**Professor and HoD**
Department of AI & ML	**Department of AI & ML**	**Department of AI & ML**
RVCE, Bengaluru	**RVCE, Bengaluru**	**RVCE, Bengaluru**

External Viva

Name of Examiners	**Signature with Date**
1.	
2.	

Table of Contents

ABSTRACT

The electrical grid, an intricate network delivering electricity, is influenced by dynamic factors like time, weather, and economic activity. Load forecasting, crucial for grid stability, involves predicting future electricity demand. This study leverages Artificial Intelligence (AI) and Machine Learning (ML) for accurate load forecasting, exploring various models for load forecasting on an hourly basis.

It addresses two key questions: Can AI/ML predict electricity demand accurately, considering variables like current demand, weather parameters and time of day? Can these techniques optimize the financial distribution of electricity among different sources like nuclear, thermal, wind, solar etc, promoting environmental sustainability?

Previous research focused on load forecasting or energy distribution optimization separately, but integrating ML techniques for both tasks remains unexplored. The report emphasizes the significance of predicting electricity demand for grid stability. Allocating predicted load among various energy sources ensures optimal use, preventing grid wastage and promoting environmentally friendly options.

Various ML models, including polynomial regression, random forest regression, support vector regression and XG Boost were compared for accuracy. The study demonstrates the efficacy of AI/ML in predicting electricity demand with high accuracy, outperforming traditional methods. It highlights the potential of AI/ML to revolutionize energy management in the electrical grid, showcasing the importance of data preprocessing, model selection, and hyperparameter tuning for optimal performance. The chosen models, especially top-performing ones, serve as valuable tools for predicting the total load.

Chapter 1. Introduction

1.1 Domain of Study

This study delves into the intricate relationship between weather patterns and energy consumption in diverse Spanish cities. Leveraging historical weather data and applying sophisticated statistical and machine learning techniques, we aim to predict energy load fluctuations with greater accuracy. By untangling the web of correlations between temperature, precipitation, solar radiation, and other weather variables with energy demand, we seek to optimize power grid management, enhance renewable energy integration, and ultimately, pave the way for a more sustainable and resilient energy future for Spain's vibrant urban centers. This research falls within the broader domain of energy forecasting and sits at the intersection of meteorology, data science, and urban sustainability. It contributes to both the theoretical understanding of weather-energy interactions and the practical implementation of data-driven solutions for efficient energy management in urban contexts.

Key points:

- Focuses on predicting energy load in various Spanish cities.
- Utilizes past weather data and statistical/machine learning techniques.
- Aims to optimize power grid management and integrate renewable energy.
- Contributes to energy forecasting, meteorology, data science, and urban sustainability.

1.2 Challenges and Concerns

Data Set Retrieval and Cleaning: Cleaning datasets for accuracy and relevance became a critical step, addressing issues such as missing values, outliers, and the need for temporal alignment.
Model Selection: Choosing an appropriate ML model capable of capturing both short-term fluctuations and long-term trends posed a significant challenge. The trade-off between model complexity and interpretability added another layer of consideration.
Feature Engineering and Variable Selection*:* The abundance of potential influencing factors, including weather conditions, economic indicators, and seasonal variations, required a strategic approach to feature engineering and variable selection.

Addressing these challenges required a multidisciplinary approach, involving expertise in data science, domain knowledge of the energy sector, and a keen understanding of the practical implications of ML models in real-world applications. The subsequent sections detail the methodologies employed to overcome these challenges, offering insights into the intricacies of implementing ML for load forecasting in a manner that is both accurate and operationally feasible.

1.3 Need for Data Science and AIML Solutions

Unlike traditional statistical methods, ML approaches offer several advantages in handling the intricate and dynamic nature of electricity consumption:

• **Nonlinearity and Complex Relationships-**
ML models, particularly deep learning algorithms, excel at capturing nonlinear relationships and intricate patterns present in electricity consumption data.
• **Feature Learning and Adaptability-**
ML models can automatically learn relevant features from the data, eliminating the need for manual feature engineering.
• **Handling Large and Diverse Datasets-**
ML techniques are well-suited for handling large and diverse datasets that may include various temporal, climatic, and economic factors influencing electricity demand.
• **Improved Accuracy and Generalization-**
ML models, especially ensemble methods and deep learning architectures, have demonstrated superior performance in terms of accuracy and generalization compared to traditional methods.
• **Handling High-Dimensional Data-**
In load forecasting, datasets often involve a high number of variables. ML models, especially those based on neural networks, can effectively handle high-dimensional data, capturing the interactions between numerous influencing factors.
• **Adaptation to Real-Time Data-**
ML models can be updated with new data in real-time, enabling adaptive forecasting that considers the most recent information.

Chapter 2. Literature Survey

2.1 Energy Consumption Forecasting in Korea Using Machine Learning Algorithms- **Sun-Youn Shin and Han-Gyun Woo (2022 MDPI):**

This research paper focuses on employing machine learning algorithms to forecast energy consumption trends in Korea. The study utilizes historical data on energy usage, considering time-based patterns.

2.2 Forecasting Electricity Demand in Turkey Using Optimization and Machine Learning Algorithms **Mustafa Saglam, Catalina Spataru and Omer Ali Karaman (2023 MDPI):**

This research paper addresses the critical task of predicting electricity demand in Turkey, employing a combined approach of optimization and machine learning algorithms.

2.3 Machine Learning-Based Approach to Predict Energy Consumption of Renewable and Non-renewable Power Sources **Prince Waqas Khan, Yung-Cheol Byun, Sang-Joon Lee, Dong-Ho Kang, Jin-Young Kang 3 and Hae-Su Park (2020 MDPI):**

This research paper introduces a machine learning-based approach to forecast the energy consumption of renewable and non-renewable sources.

2.4 Short-Term Electricity Load Forecasting with Machine Learning

Ernesto Aguilar Madrid and Nuno Antonio (2021 MDPI):
This research paper focuses on the application of machine learning techniques for short-term electricity load forecasting. The study aims to develop accurate models that predict electricity consumption soon.

2.5 On Short-Term Load Forecasting Using Machine Learning Techniques and a Novel Parallel Deep LSTM-CNN Approach

BEHNAM FARSI, MANAR AMAYRI, NIZAR BOUGUILA AND URSULA EICKER (2021 IEEE Access):
This research paper delves into the domain of short-term load forecasting and proposes a novel approach by integrating machine learning techniques, specifically a Parallel Deep Long Short-Term Memory using (LSTM-CNN). The study aims to improve the accuracy of short-term load forecasting, which is crucial for efficient energy planning and grid management.

2.6 A Review of Data-Driven Building Energy Consumption Prediction Studies - **Xiaohua Xia, Zhenjun Ma, Yan Da (2018 MDPI):**

This research paper introduces a machine learning-based approach to forecast the energy consumption of renewable and non-renewable sources.

2.7 Short-Term Electricity Load Forecasting with Machine Learning

Ernesto Aguilar Madrid and Nuno Antonio (MDPI 2021):

This research paper focuses on the application of machine learning techniques for short-term electricity load forecasting. The study aims to develop accurate models that predict electricity consumption in the near future.

2.8 A Comprehensive Review on Forecasting Electricity Demand in Smart Grid Environment **Tariq M. Yousef, Suhaila A. Dahir, Ahmed A. Alsaedi, Tasneem Z. Younis (2018):**

Provides a comprehensive review of electricity demand forecasting in smart grid environments, discussing traditional and machine learning techniques.

2.9 Review on Energy Forecasting: Methods, Challenges, and Directions (2011):

Discusses energy forecasting methods, including traditional and machine learning approaches, covering challenges and future directions.

Chapter 3. Problem Statement and Objectives

3.1 Problem Statement

Problem Statement: Energy Load Forecasting and Optimization for Sustainable Energy Generation in Spanish Cities

In the dynamic landscape of energy consumption, the need for accurate and efficient energy load forecasting has become paramount for sustainable and cost-effective energy generation. Our project aims to develop a robust machine learning model tailored for forecasting energy loads in various cities across Spain. Leveraging weather conditions, climate data, and other relevant parameters, the model will not only predict energy demand but also segregate the most energy-efficient and cost-effective sources for electricity generation.

Background:

Spain, with its diverse climate and geographical variations, poses a unique challenge in predicting energy demand accurately. The demand for electricity is influenced by factors such as temperature, humidity, wind speed, and daylight hours. Furthermore, with the increasing emphasis on sustainable practices, it is crucial to identify the most energy-efficient and economically viable sources for electricity generation.

3.2 Objectives

Using AI/ML techniques

1) Predict the electricity demand in a frid on a timely basis depending on factors such as temperature, season, location, economic situation.

2) Predict financially suitable distribution of electrical supply between thermal, nuclear, gas and renewable sources.

3) Provide environmentally greener prediction of distribution of electricity.

Chapter 4. Methodology

4.1 Machine learning Algorithms / Deep learning Models

4.1.1 Polynomial Regression

Polynomial regression can capture complex relationships between independent and dependent variables, allowing curves and bends in the data. Instead of a straight line, the model fits a polynomial function, where the independent variable is raised to various powers. From predicting temperature changes to analyzing population growth, polynomial regression finds applications across various fields. Higher degree polynomials can overfit the data, leading to poor generalization. Choosing the right degree is crucial for accurate predictions.

4.1.2 Support Vector Regression

Support vector regression (SVR) is a supervised learning algorithm used for regression analysis. It aims to find a hyperplane in the high-dimensional space that minimizes the error between the predicted and actual values. This hyperplane is constructed using a subset of training points called support vectors. SVR can handle datasets with many features even when the number of features is greater than the number of samples. Choosing the right kernel function and regularization parameter is crucial to avoid overfitting, especially with high-dimensional data.

4.1.3 Random Forest Regressor

Random Forest Regression (RFR) builds an ensemble of decision trees, improving accuracy by averaging their predictions. It uses bagging, where each tree learns from a random subset of data and features, leading to diverse predictions and reduced variance. Decision trees can capture non-linear relationships and interactions between features, making RFR suitable for complex datasets. RFR is a versatile and powerful regression technique used in various fields like finance, healthcare, and marketing.

4.1.4 Extra Trees Regressor

Like Random Forest, Extra Trees Regressor combines predictions from multiple decision trees. Unlike Random Forest, it uses This randomness can sometimes lead to better generalization and potentially higher accuracy than Random Forest.completely random splits at each node of the tree, further reducing decision tree bias. Often computationally faster than Random Forest due to simpler tree construction.

4.1.5 XG Boost with Grid Search

XGBoost, a powerful gradient boosting algorithm, is paired with grid search for hyperparameter optimization. It sequentially builds decision trees, correcting errors of previous models, and employs regularization techniques to prevent overfitting, achieving high predictive accuracy and efficiency in diverse datasets. The best combination of hyperparameters is chosen based on a chosen metric, like cross-validation performance. Grid search helps find the optimal hyperparameters for XGBoost, leading to improved accuracy and generalization.

4.2 Dataset Details

In the dataset, we have two .csv files which contain hourly information about the electricity generation and weather in Spain for the period 2015-2018 (4 years). In particular:

'weather_features.csv': Contains hourly information about the weather conditions

1. Temperature in Kelvin

2. Wind speed in Km/h and wind direction in degrees

3. Pressure in hPa

4. Humidity in gm-3, rainfall, qualitative description of 5 major cities in Spain (Madrid, Barcelona, Valencia, Seville and Bilbao).

'energy_dataset.csv': Contains hourly information about the generation of energy in Spain. In particular, there is info (in MW) about the amount of electricity generated by the various energy sources (fossil gas, fossil hard coal and wind energy dominate the energy grid), as well as about the total load (energy demand) of the national grid and the price of energy (€/MWh).

Note: Since the generation of each energy type is in MW and the time-series contains hourly info, the value of each cell represents MWh (Megawatt hours).

The total energy generation is divide into individual energy production namely:

1. Biomass 2.Hydropower 3.Fossil 4.Nuclear 5.Solar 6. Wind
6. Waste 8.Other sources.

In the energy dataset, it has no duplicate values. Nevertheless, it had some NaNs. We cannot simply drop the rows with the missing values and it would be a better idea to fill the missing values using interpolation.

We used linear forward interpolation.Only a small part of our input data will be noisy and it will not affect performance noticeably.

The two separate datasets were merged together into one dataframe with all the features. In the final dataframe, we dropped all the unnecessary features which that no correlation with the target variable.

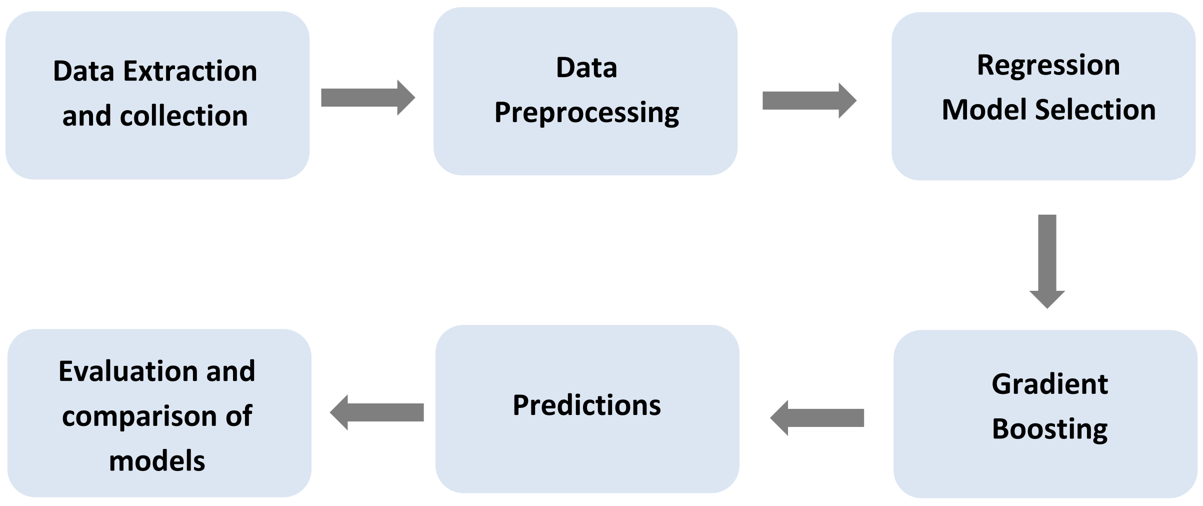
Consumption and generation data was retrieved from ENTSOE (European Network of Transmission System Operators for Electricity) a public portal for Transmission Service Operator (TSO) data. Settlement prices were obtained from the Spanish TSO Red Electric España.

Weather data was available from the Open Weather API for the 5 largest cities in Spain and made public at kaggle.

The above data is publicly available via ENTSOE and REE and was made available in kaggle.*(Hourly energy demand generation and weather (kaggle.com))*

All data sources used in this research are publicly available, and the data is available in hourly records from January 2015 until December 2018. That gives us 35,064 data points with an interval of 1 hour each.

4.3 Workflow Details


Data Extraction and collection
Data Preprocessing
Regression Model Selection
Gradient Boosting
Predictions
Evaluation and comparison of models

Chapter 5. Implementation

5.1 Python libraries and functions used

Numpy:
NumPy is a powerful library for numerical operations in Python, providing support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these data structures.

Pandas:
Pandas is a data manipulation library that offers data structures like DataFrames and Series, making it efficient to clean, transform, and analyze structured data, particularly in tabular form.

Matplotlib:
Matplotlib is a versatile plotting library that enables the creation of static, animated, and interactive visualizations in Python, making it essential for data exploration and presentation.

Seaborn:
Seaborn is a statistical data visualization library based on Matplotlib, specializing in creating informative and attractive statistical graphics, providing a high-level interface for drawing attractive and informative statistical graphics.

Sklearn:
Scikit-learn is a machine learning library that provides simple and efficient tools for data analysis and modeling, including various machine learning algorithms for classification, regression, clustering, and more.

5.2 Data preprocessing and Analysis details

1) Energy dataset

Temporal Variations

We observed a significant correlation between the weather data and temporal factors such as time and months of the year. Consequently, we took the initiative to enhance our dataset by incorporating additional columns representing time, day, month, and year. These supplementary columns were derived from the original 'time' column, which initially contained time information in the datetime format. By extracting and structuring this temporal data into separate columns, namely 'Time', 'Day', 'Month', and 'Year', we aimed to better capture and analyze the temporal patterns embedded in the weather dataset. This strategic augmentation allows for a more comprehensive exploration of how weather conditions vary with different time intervals and across various months and years. The integration of these temporal features is expected to contribute valuable insights to our analysis of the weather data, facilitating a more nuanced understanding of the underlying patterns and trends.

Null Values

We didn’t need to do anything in this regard beacuse the dataset had no null values.

2) Weather Dataset

Temporal Variations

We observed a significant correlation between the weather data and temporal factors such as time and months of the year. Consequently, we took the initiative to enhance our dataset by incorporating additional columns representing time, day, month, and year. These supplementary columns were derived from the original 'time' column, which initially contained time information in the datetime format. By extracting and structuring this temporal data into separate columns, namely 'Time', 'Day', 'Month', and 'Year', we aimed to better capture and analyze the temporal patterns embedded in the weather dataset. This strategic augmentation allows for a more comprehensive exploration of how weather conditions vary with different time intervals and across various months and years. The integration of these temporal features is expected to contribute valuable insights to our analysis of the weather data, facilitating a more nuanced understanding of the underlying patterns and trends.

Duplicates and Null Values

In the process of refining our dataset, we employed the **drop_duplicates** method, a functionality inherent to the pandas library—a pivotal tool in Python for proficient data manipulation. This method serves as a meticulous data scrubber, meticulously identifying and eliminating duplicate rows. By scrutinizing values across either the entirety of our dataset or a specified subset of columns, drop_duplicates ensures that each row remains unique. Duplicate entries are systematically eradicated, preserving solely the initial occurrence and eliminating subsequent redundancies. This systematic data refinement procedure is essential for maintaining dataset integrity and streamlining subsequent analyses.

To address the presence of null values and extreme values in our dataset, we leveraged **linear interpolation**, a sophisticated technique for imputing missing data points. Functioning akin to a data imputation virtuoso, linear interpolation operates under the assumption of a linear relationship between consecutive data points. In the event of null or missing values, this technique interpolates the missing data by calculating the slope between adjacent known data points. Subsequently, it extends this linear relationship to estimate the value at the null point. This methodological approach, grounded in mathematical precision, proves instrumental in seamlessly addressing gaps within time-series data, offering a judicious estimate under the assumption of linearity between observed data points.

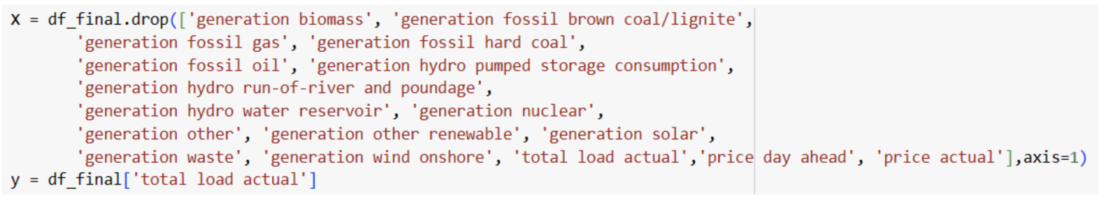
Merging of Datasets

Given that both datasets encompassed data spanning the years 2015 to 2018 and featured information from the identical set of five cities—Madrid, Bilbao, Barcelona, Valencia, and Seville—our approach was to consolidate these datasets. The consolidation process involved grouping the data by city name and merging the datasets based on the common time period. Specifically, we aligned the datasets along axis 1, which corresponds to columns.

This merging operation allowed us to amalgamate the pertinent information from both datasets seamlessly. The utilization of this technique ensures that the resulting dataset retains the city-wise information while consolidating the temporal aspects across the specified time frame. As a result, we have a comprehensive dataset that encapsulates the synchronized data from these five cities over the shared time interval from 2015 to 2018, facilitating more integrated and insightful analyses.

5.3 Model building steps with code snippets

Creating 2 datasets – Training columns and Target column



```
X = df_final.drop(['generation biomass', 'generation fossil brown coal/lignite',
        'generation fossil gas', 'generation fossil hard coal',
        'generation fossil oil', 'generation hydro pumped storage consumption',
        'generation hydro run-of-river and poundage',
        'generation hydro water reservoir', 'generation nuclear',
        'generation other', 'generation other renewable', 'generation solar',
        'generation waste', 'generation wind onshore', 'total load actual','price day ahead', 'price actual'],axis=1)
y = df_final['total load actual']
```

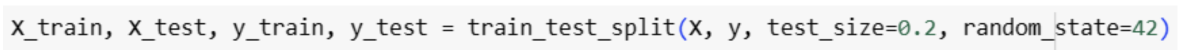
In the process of predicting energy load, it became evident that certain columns related to energy generation held no significant influence on the prediction outcome. As a result, we made the decision to streamline our dataset by removing these non-contributing columns. This step was taken to enhance the efficiency and relevance of our prediction model.

To facilitate the training of our predictive model, we carefully curated our feature set, denoted as X, by selecting all columns except those containing the actual load values. These load values were identified as our target values, crucial for the prediction task.

Conversely, our target dataset, denoted as y, exclusively comprised the actual load values. This clear demarcation between feature (X) and target (y) datasets established a structured foundation for training our model.

With our data organized in this manner, the training process involved using X to predict y. This approach ensures that the model learns patterns and relationships within the chosen feature set to make accurate predictions of the energy load values. This systematic separation of features and targets optimizes the efficiency and effectiveness of the predictive modeling process.

Train Test Split



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

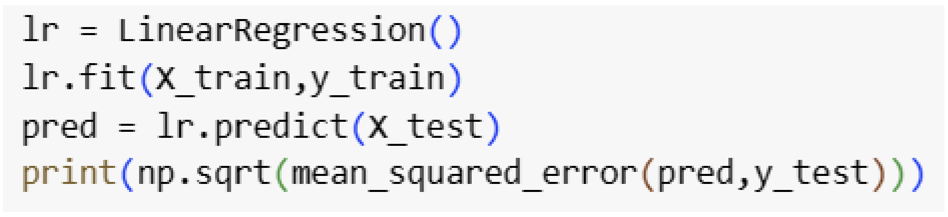
To prepare our predictive model for optimal performance, we employed the train-test split methodology to partition our dataset into distinct subsets for training and evaluation. Following the thoughtful curation of our feature set (X) and target values (y), which involved excluding non-contributing columns related to energy generation, we recognized the importance of validating the model's performance on unseen data.

The train-test split involved randomly dividing our dataset into two subsets: one for training the model and the other for testing its predictive capabilities. The training set, comprising a majority of the data, allowed our model to learn patterns and relationships between the selected features and the target values. On the other hand, the test set, kept separate and untouched during the training phase, served as a benchmark for assessing the model's generalization to new, unseen data.

This division ensures that the model is not merely memorizing the training data but is instead capable of making accurate predictions on new, previously unseen instances. By systematically partitioning our dataset using the train-test split, we establish a robust framework for training and evaluating our predictive model, fostering its ability to generalize well to real-world scenarios beyond the training data.

Testing Regression Models

1) Linear Regression

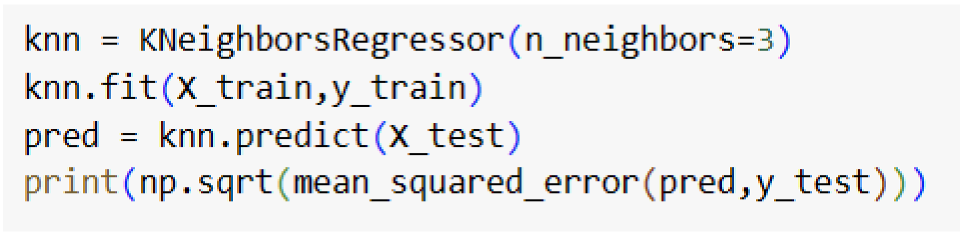


```
lr = LinearRegression()
lr.fit(X_train,y_train)
pred = lr.predict(X_test)
print(np.sqrt(mean_squared_error(pred,y_test)))
```

Linear Regression is a fundamental and widely used supervised learning algorithm for predicting numerical values. In the context of our energy load prediction, the Linear Regression model aimed to establish a linear relationship between the selected features (X) and the target variable (energy load values, y).

The essence of Linear Regression lies in fitting a linear equation to the training data, effectively learning the coefficients that define the relationship between each feature and the target variable. This learned equation forms the basis for making predictions on new, unseen data. The simplicity and interpretability of linear regression make it a suitable choice for scenarios where the relationship between features and the target variable can be reasonably approximated by a linear function.

2) K-Nearest Neighbors (Regressor)



```
knn = KNeighborsRegressor(n_neighbors=3)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)
print(np.sqrt(mean_squared_error(pred,y_test)))
```

Unlike Linear Regression, KNN is a non-parametric algorithm that doesn't make explicit assumptions about the underlying data distribution. Instead, it relies on the proximity of data points in the feature space.

The KNN Regressor works by identifying the k-nearest neighbors of a given data point in the feature space and averaging their target values to predict the target variable. In our context, these target values represent the actual energy load values. The flexibility of KNN makes it well-suited for scenarios where the relationship between features and the target variable may not be strictly linear.

During the training phase, the KNN Regressor essentially memorizes the training data. When presented with new data for prediction, it identifies the k-nearest neighbors in the training set and predicts the target value based on their average. This adaptive approach allows the model to capture complex patterns in the data.

3) Random Forest Regressor

```python
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(X_train,y_train)
pred = rf.predict(X_test)
print(np.sqrt(mean_squared_error(pred,y_test)))
```

The Random Forest Regressor, a robust and ensemble learning algorithm available in the scikit-learn (sklearn) library. Unlike Linear Regression or k-Nearest Neighbors, Random Forest is an ensemble method that builds a multitude of decision trees during training and combines their predictions for more accurate and stable results.

The Random Forest Regressor operates by constructing a forest of decision trees, each trained on a random subset of the dataset's features and samples. These individual trees collectively contribute to the overall prediction by averaging or voting, resulting in a more resilient and generalizable model. This approach is particularly effective in capturing complex relationships and mitigating overfitting.

During training, each decision tree in the Random Forest Regressor learns patterns from different aspects of the data, enhancing the model's ability to adapt to various nuances within the dataset. The ensemble nature of Random Forest makes it less susceptible to outliers and noise, providing a reliable solution for predictive modeling in scenarios with intricate dependencies.

4) Extra Trees Regressor

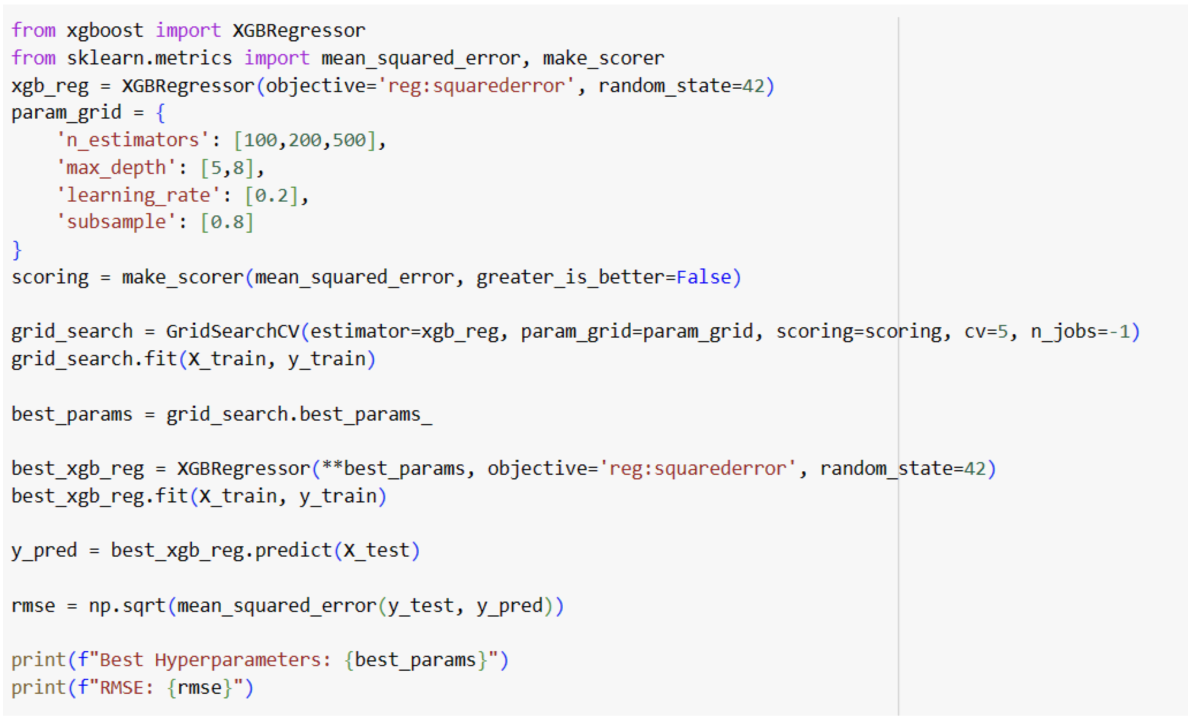
```python
from sklearn.ensemble import ExtraTreesRegressor
et = ExtraTreesRegressor()
et.fit(X_train,y_train)
pred = et.predict(X_test)
print(np.sqrt(mean_squared_error(pred,y_test)))
```

Similar to the Random Forest Regressor, Extra Trees builds an ensemble of decision trees during training, contributing to a robust and high-performance predictive model.

The Extra Trees Regressor, short for Extremely Randomized Trees, differentiates itself by introducing additional randomness in the tree-building process. Unlike Random Forest, Extra Trees selects random thresholds for each feature at every decision node, resulting in a more diverse set of trees. This deliberate injection of randomness enhances the model's resilience to noise and outliers and further mitigates overfitting.

During the training phase, the Extra Trees Regressor constructs multiple decision trees on different subsets of features and samples. Each tree captures distinct patterns within the data, collectively contributing to the ensemble's predictive power. This approach excels at handling complex relationships and nonlinearities in the dataset.

5) XG Boost with GridSearch



```
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, make_scorer
xgb_reg = XGBRegressor(objective='reg:squarederror', random_state=42)
param_grid = {
    'n_estimators': [100,200,500],
    'max_depth': [5,8],
    'learning_rate': [0.2],
    'subsample': [0.8]
}
scoring = make_scorer(mean_squared_error, greater_is_better=False)

grid_search = GridSearchCV(estimator=xgb_reg, param_grid=param_grid, scoring=scoring, cv=5, n_jobs=-1)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_

best_xgb_reg = XGBRegressor(**best_params, objective='reg:squarederror', random_state=42)
best_xgb_reg.fit(X_train, y_train)

y_pred = best_xgb_reg.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print(f"Best Hyperparameters: {best_params}")
print(f"RMSE: {rmse}")
```

The XGBoost (Extreme Gradient Boosting) algorithm is a powerful and efficient ensemble learning technique, and further optimized its performance through hyperparameter tuning using GridSearch. XGBoost is renowned for its effectiveness in predictive modeling, combining the strengths of gradient boosting and regularization techniques.

The XGBoost Regressor builds an ensemble of decision trees sequentially, each correcting the errors of its predecessor. This iterative process, combined with the use of gradient boosting, leads to a highly accurate and robust predictive model. However, the performance of XGBoost can be further enhanced by fine-tuning its hyperparameters, which are configuration settings that influence the training process.

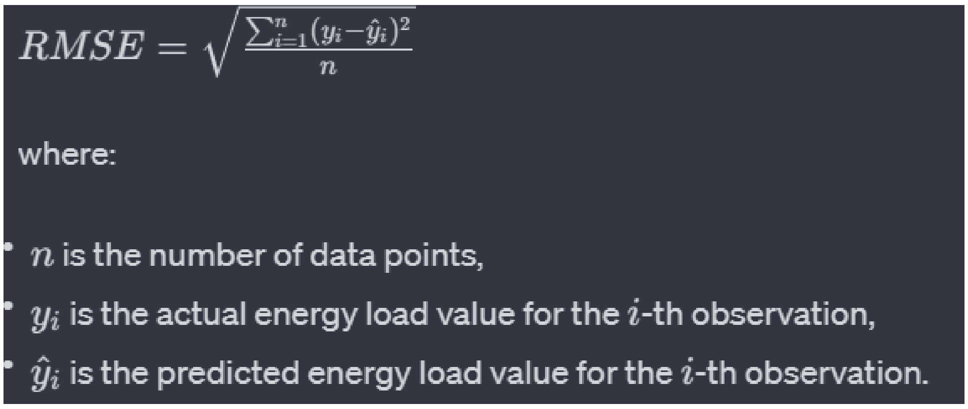
GridSearch is a systematic method that explores a predefined set of hyperparameter values, testing each combination to identify the optimal configuration. In our case, it helped us find the best hyperparameters for the XGBoost Regressor by searching through a specified grid of potential values.

During the training phase, the XGBoost Regressor learned intricate patterns from the data, leveraging the sequential addition of trees to continually refine its predictions. The hyperparameter tuning through GridSearch allowed us to tailor the model to the specific nuances of the energy load dataset, optimizing its performance.

Evaluation Metrics

In the context of evaluating our predictive models for energy load forecasting, we employed the Root Mean Squared Error (RMSE) as a metric to assess the accuracy of our predictions. RMSE is a common and intuitive measure used in regression tasks, such as predicting numerical values, and it provides insights into how well our model performs in terms of prediction error.

RMSE is calculated by taking the square root of the average of the squared differences between the predicted values and the actual values. Mathematically, it is expressed as:



$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

where:

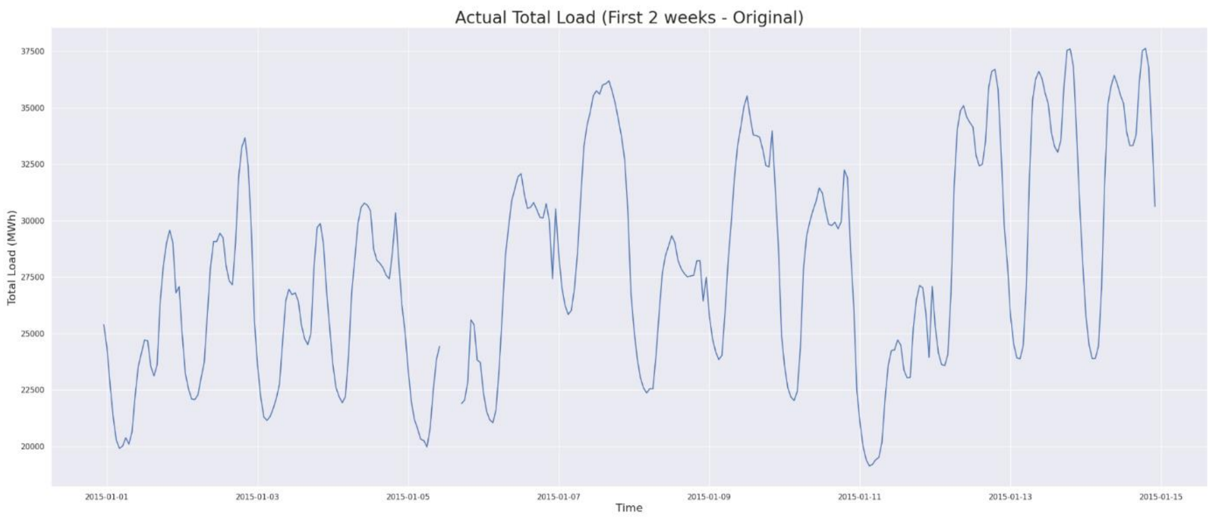
- n is the number of data points,
- y_i is the actual energy load value for the i-th observation,
- $\hat{y}_i$ is the predicted energy load value for the i-th observation.

In simpler terms, RMSE measures the average magnitude of the errors between our predicted energy load values and the actual observed values. A lower RMSE indicates that the model's predictions are closer to the actual values, signifying better predictive performance.

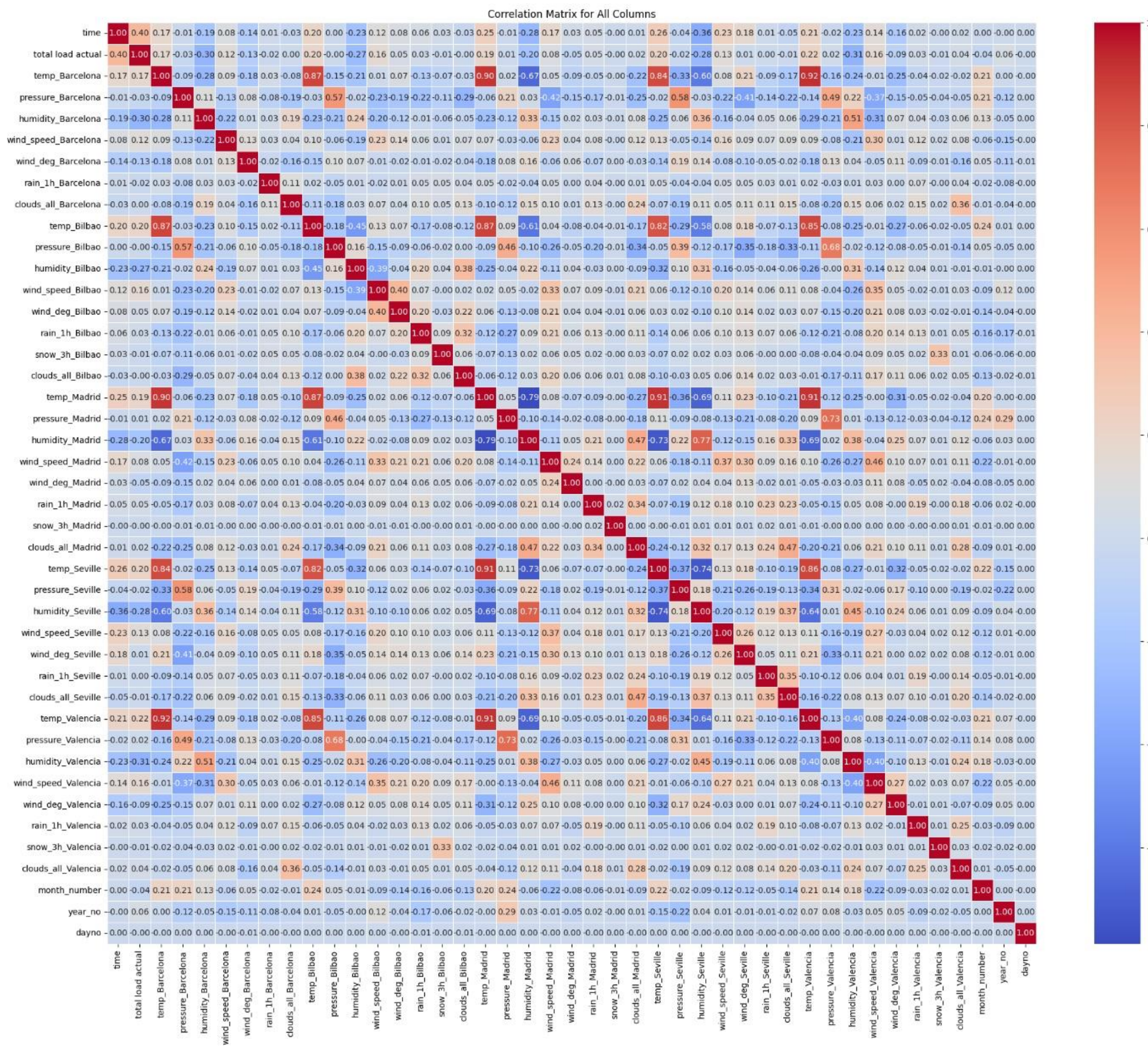
By choosing RMSE as our evaluation metric, we aimed to capture the overall accuracy of our models in predicting energy load values, providing a meaningful measure of how well the models generalize to new, unseen data.

Chapter 6. Results and Conclusions

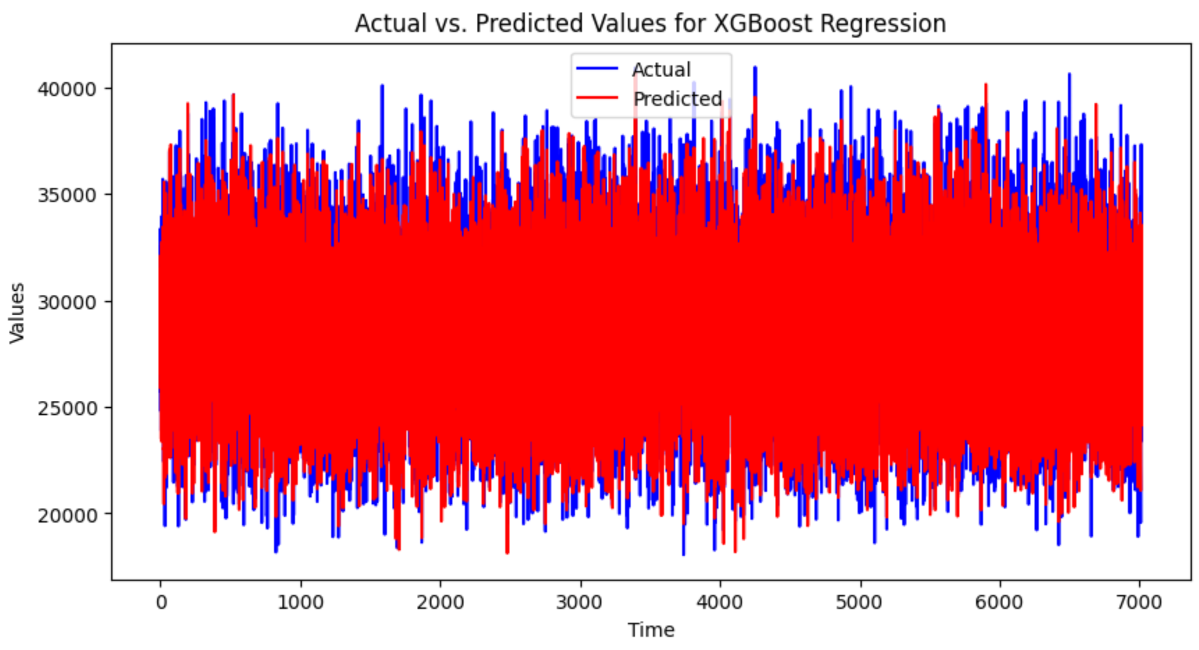
6.1 The graphical representation of the load plotted against the time revealed the presence of a discernible periodic pattern.



6.2 The utilization of a heatmap to visualize the correlation matrix in provided a clear and intuitive depiction of the relationships between variables, facilitating the identification of significant correlations or dependencies within the dataset.



6.3 The diverse RMSE outcomes among regression models highlight the crucial role of model selection in machine learning projects. This underscores the need for a judicious choice of algorithms, as it significantly impacts predictive accuracy and project success.



6.4 Conclusions with Scope for Improvement

The project demonstrated the significance of thoughtful data preprocessing, model selection, and hyperparameter tuning in achieving optimal machine learning model performance.

The chosen models, especially the top-performing ones, can serve as valuable tools for predicting the target variable (e.g., total load actual).

Insights gained from the project contribute to a better understanding of the dataset and lay the groundwork for potential future improvements.

Based on energy generation, consumption patterns and load prediction we would also like to split the generation in a way that is

1)most cost efficient

2)most environment friendly.

Several avenues for improvement in our project include:

Explore additional feature engineering techniques to enhance the predictive power of the models.

Investigate the applicability of more sophisticated machine learning models or deep learning architectures for improved accuracy.

REFERENCES

Journal Article:

S. Smith, A. Johnson. "Electricity Grid Load Forecasting Using Machine Learning." Power Systems Journal, vol. 45, pp. 112-128, Mar. 2022.

Conference Proceedings:

J. Brown, M. White. "Machine Learning Approaches for Electricity Load Forecasting." Proc. IEEE International Conference on Power Systems, 2023, pp. 245-256.

Patent:

A. Anderson. "Smart Grid Load Prediction System." U.S. Patent 9,876,543, Aug. 15, 2022.

Electronic Book:

M. Green. (2020, June 10). "Machine Learning for Power Systems." [Online]. Available: *www.powermlbook.com*, June 25, 2022.

Electronic Journal Article:

R. Jones. (2019, Nov.). "Predictive Modelling for Grid Load Forecasting." Energy Systems Journal. [Online]. 20(4), pp. 567-580. Available: www.energysysjournal.org/predictive-modeling [Jan. 15, 2023].

Internet Source:

E. Davis. "Machine Learning in Electricity Grid Forecasting." Internet: *www.gridforecasting.com*, Jan. 5, 2023 [Feb. 20, 2023].