Electricity Demand Forecasting

Energy Analytics

Bryan Cheong

Paul Drianno

Louis Perdrix

Christopher Sasanuma

Joaquin Umaschi

MSc Business Analytics

Imperial College London

1 Introduction

Accurately forecasting electricity demand is essential for effective energy planning and grid reliability. In this project, we participated in a forecasting competition where we developed and refined multiple models to predict daily electricity demand based on weather and calendar features. This report walks through the evolution of our modeling approach over the course of the competition, from our initial baseline model to more advanced machine learning (ML) techniques. In the end, we present our final recommended model along with key takeaways and suggestions for future improvement.

2 Exploratory Data Analysis (EDA)

2.1 Data Cleaning

The dataset consists of three separate files:

- Daily overall demand data for electricity in the UK since 2020.
- Daily sunshine duration records for London, Leeds, and Bristol, including forecasts for the last 60 days. Note that historical values are in minutes while forecasts are in hours.
- Temperature measurements for London, Leeds, and Bristol at four time points (0h, 6h, 12h, and 18h), along with forecasts for the last 60 days.

The dataset spans approximately 2000 days of observation. Before analysis, we cleaned and merged these files, addressing two key data quality issues to ensure forecast accuracy.

2.1.1 Duplicate Records

We identified a duplicate timestamp in the temperature dataset for January 7, 2025. To maintain data integrity, we retained only the first instance. This approach eliminated redundancy that could potentially bias our features and models.

2.1.2 Missing Data Imputation

A gap in temperature data was identified from December 22-23, 2023. Rather than removing these records, we implemented an hour-matched imputation method that maintained temporal integrity. For each missing value, we averaged temperatures from the same hour on surrounding days (up to two days before and after), preserving both daily cycles and seasonal patterns:

```
2023-12-23 06:00: Filled using [12-22, 12-24, 12-21, 12-25] of hour [6]: 8.6964°C
```

This method was chosen for two key reasons:

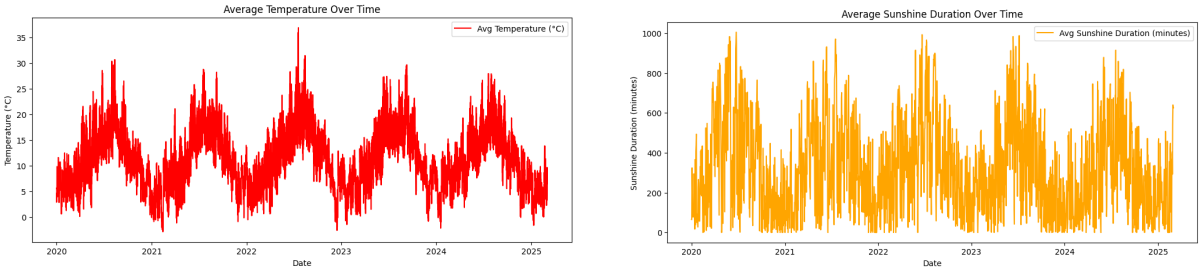
1. Temporal continuity: Maintaining the correct sequence preserved the time series structure
2. Lag demand integrity: Lag demand must remain accurate by maintaining all dates.

After imputation, we verified complete coverage with no missing values and completed our data cleaning process. More feature engineering will be done for each of our three models in Section 3.

2.2 Correlation Analysis

To understand how weather influences electricity demand, we examined the correlations between total demand, average temperature, and sunshine duration. Visually, demand shows strong seasonality, with higher values in winter, while temperature peaks in summer, indicating a negative relationship.

As expected, we found a negative correlation between temperature and demand, consistent with heating needs during colder periods. Sunshine duration was also negatively correlated with demand, though the effect was weaker. These patterns guided our feature selection, particularly the use of temperature-derived metrics such as HDD and CDD.



(a) Avg Temperature (b) Avg Sunshine Duration

Figure 1: Temperature and Sunshine Duration Over Time

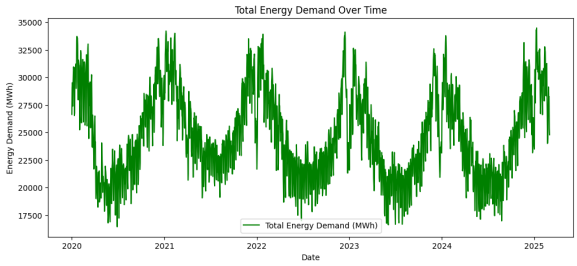
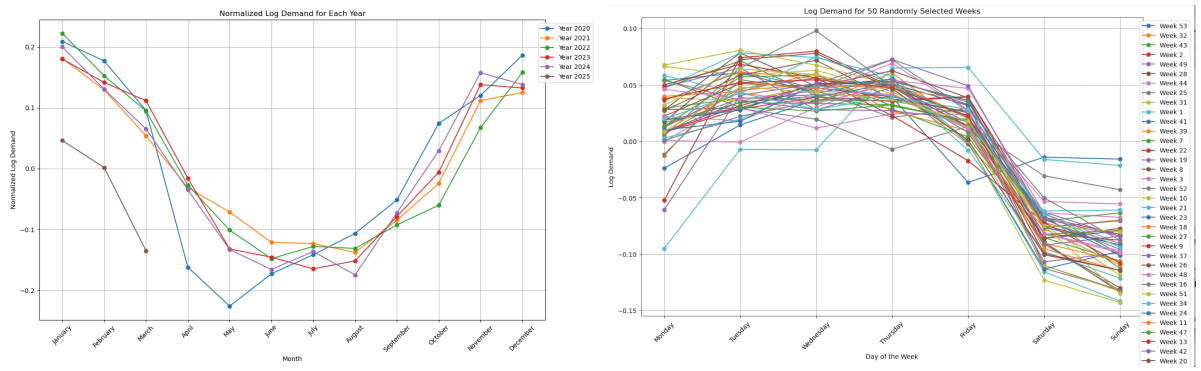


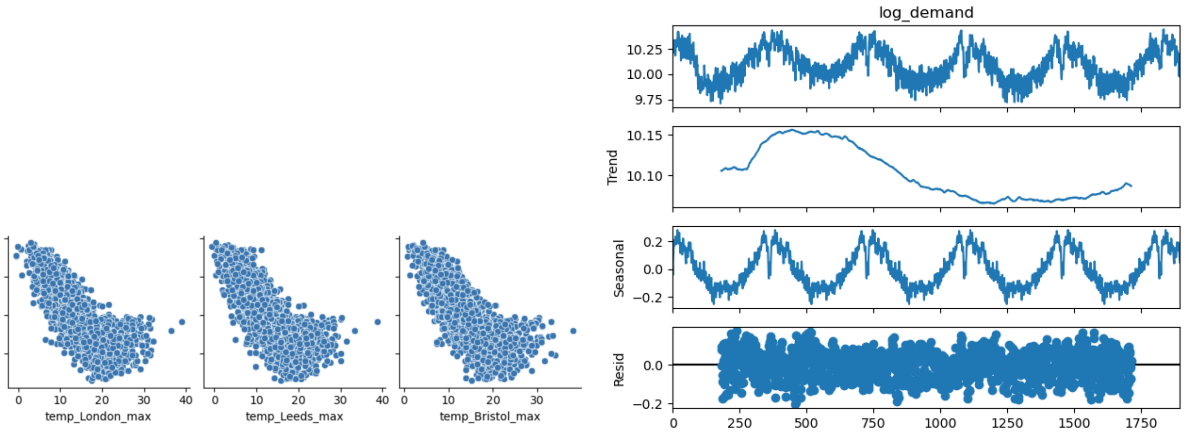
Figure 2: Total Energy Demand Over Time



(a) Log demand by month (b) Log demand by weekday

Figure 3: Monthly and weekly patterns in electricity demand

Electricity demand shows a consistent seasonal trend, with significantly lower values during summer months and higher consumption during winter. This likely reflects increased heating use in colder months. On a weekly level, demand tends to peak midweek and drop over the weekend, which is consistent with reduced industrial and commercial activity.



(a) Max temperature vs log demand (b) STL decomposition of log demand

Figure 4: Temperature and seasonal demand patterns

The scatter plot highlights a nonlinear relationship between temperature and electricity demand. Demand is highest at low temperatures due to heating requirements, decreases around 15–20°C, and rises again at higher temperatures, reflecting cooling needs. The STL decomposition further illustrates seasonal peaks in winter and drops around holidays like Christmas, along with a gradual long-term trend.

3 Model Selection and Forecast Timeline

3.1 Model 1 – Baseline Model

The baseline model includes the following features:

- log(demand)
- Weekend dummy
- Average daily temperature
- Month dummy
- 10 lags of $log(demand)$
- Performance: $R^2 = 91.42$, RMSE $= 0.048$

So the linear regression equation is :

$$\log(\text{demand}) = \beta_0 + \beta_1 \text{average\_temp\_london} + \beta_2 \text{average\_temp\_leeds} + \beta_3 \text{average\_temp\_bristol} + \beta_4 \text{is\_weekend} + \sum_{j=1}^{11} \gamma_j \text{month}_j + \sum_{i=1}^{10} \delta_i \text{lag}_i + \epsilon \quad (1)$$

3.1.1 Feature Selection

Building on exploratory data analysis, we identify a strong correlation between temperature and *log(demand)*. Additionally, we observe significant seasonality on both weekly and monthly scales. For instance, the analysis in Section 2.2 revealed consistent weekly cycles, with demand typically peaking midweek and significantly dropping on weekends. So we created a dummy variable `is_weekend` to account for this effect.

The correlation with temperature suggests that using regression is preferable to a pure time series model like ARIMA or Holt-Winters, as it allows us to capture relationships beyond just trend and seasonality.

As a first approach, we aim for simplicity to prevent overfitting while maintaining high interpretability. We use the average daily temperature, which is significantly correlated with *log(demand)*. To account for seasonality, we include dummies for both weekly and yearly trends. We also tested a Covid dummy variable to capture the drop in demand from April to June 2020, but this did not yield significant improvements. Additionally, we incorporate 10 lags to utilize information from previous days, which can be useful during holidays for instance. This adjustment results in a significant performance improvement. However, to maintain model simplicity, we do not include sunshine duration, as it shows a low correlation with *log(demand)*.

3.1.2 Computation of the Standard Error

The main sources of error in our model arise from the model itself and the weather forecast used for predictions. To account for both sources, we calculate the standard error as follows:

- The model's RMSE is 0.04779.
- The standard error of the weather forecast error distribution, computed as (actual value − forecast value), is 0.006.
- We aggregate these errors using the Euclidean norm:

$$\sigma_{\text{total}} = \sqrt{(\text{RMSE})^2 + (\text{Weather Error})^2} = \sqrt{(0.04779)^2 + (0.006)^2} = 0.048. \quad (2)$$

- We use this standard error value to assess the uncertainty of our model.

3.1.3 Strengths and Limitations

Despite its simplicity, the baseline model performs well, achieving an RMSE of 0.04779.

However, the model has certain limitations, particularly in the naive selection of features, such as the number of lags. Future improvements will explore new features and transformations to enhance model accuracy and robustness.

3.2 Model 2 – Regularized Regression and Decision Trees + More Features

3.2.1 Candidate Models

To enhance predictive performance, we expanded our modelling approach to include regularized linear models and one tree-based algorithms. This allowed us to effectively utilise our expanded feature set while managing the risk of overfitting. Our model selection included:

1. **Linear Regression:** Baseline to assess the value of regularization and non-linear models.
2. **Regularized Linear Models**:
   - **Lasso Regression**: Applies L_1 regularization, shrinking some coefficients to zero
   - **Ridge Regression**: Employs L_2 regularization, reducing coefficients proportionally to address multicollinearity without eliminating features
3. **Gradient Boosting - XGBoost**: Used to capture non-linear relationships and complex interactions. Chosen as a benchmark against linear models while maintaining interpretability, more ML techniques are explored in Section 3.3.

Each model underwent hyperparameter tuning through grid search cross-validation to optimise performance while preventing overfitting. Then, performance metrics on the test data guided our final selection of Model 2.

3.2.2 Capturing Daily Temperature Variability

We evaluated using 6-hourly temperature measurements but identified key challenges: this would require forward-filling daily demand across four 6-hour periods, creating artificial step patterns that could mislead the model. Instead, we captured temperature variability with daily periods by calculating four key statistics for each city: mean, minimum, maximum, and standard deviation. This approach preserved temporal alignment with demand data while capturing the essential intra-day temperature variability that influence energy consumption.

3.2.3 Heating Degree Days and Cooling Degree Days

Heating Degree Days (HDD) and Cooling Degree Days (CDD) were added to our initial model to explicitly account for the nonlinear relationship between temperature and energy demand. HDD captures heating needs when the temperature falls below a comfort threshold, calculated as:

$$\text{HDD} = \max(\text{Threshold Temperature} - \text{Actual Temperature}, 0)$$

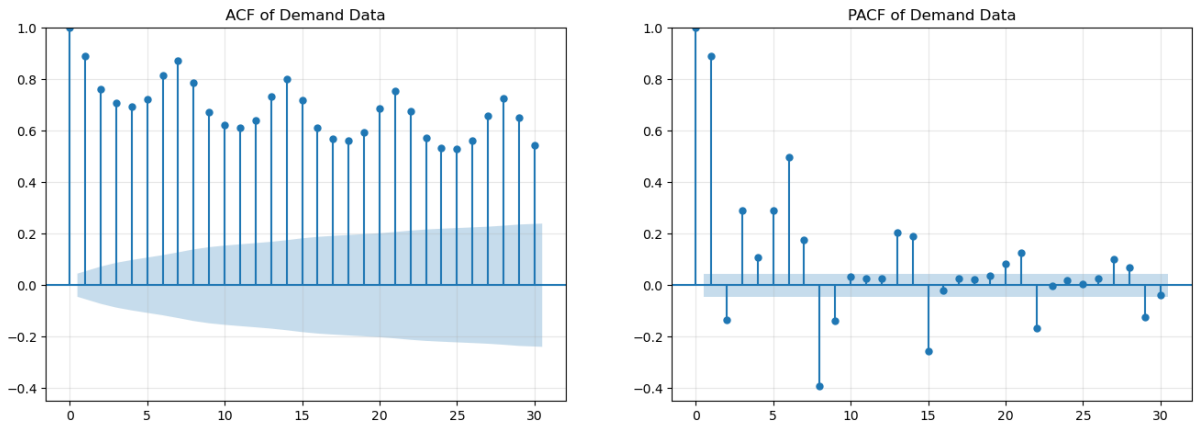
Conversely, CDD quantifies cooling demand when temperatures exceed a higher threshold:

$$\text{CDD} = \max(\text{Actual Temperature} - \text{Threshold Temperature}, 0)$$

We selected thresholds of $15°C$ for HDD and $22°C$ for CDD, consistent with standards widely recognized in the literature for residential comfort (Valor et al., 2001; Sailor and Muñoz, 1997; Hong and Wang, 2014; Hor et al., 2005). Incorporating HDD and CDD improved the model's ability to capture seasonal variations and temperature-driven peaks in energy consumption, thus enhancing predictive accuracy and practical interpretability.

3.2.4 Exploring Autocorrelation with ACF and PACF

This section applies a data-driven approach to overcome the naive lag selection limitations noted in Section 3.1.3 by identifying which historical time points significantly influence future values.



(a) ACF Plot of the *log*(*demand*) Data (b) PACF Plot of the *log*(*demand*) Data

Figure 5: ACF and PACF Plot of the *log*(*demand*) Data

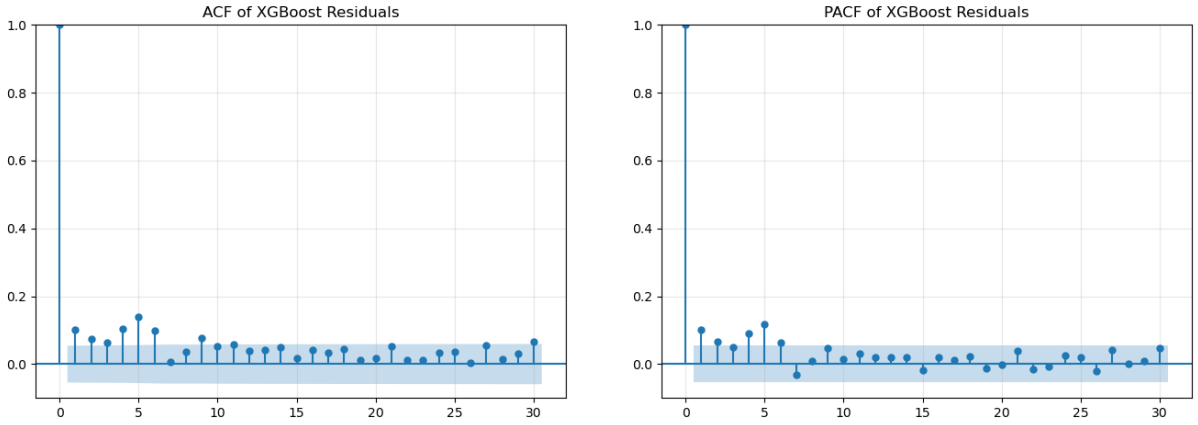
The autocorrelation function (ACF) and partial ACF (PACF) plots in Figure 5 revealed strong temporal dependencies in the demand, providing statistical guidance for optimal lag selection.

The ACF plot showed positive autocorrelation extending 30 lags, with stronger correlations at lags 1-7, 14, 21 and 28, indicating strong day-to-day and weekly patterns. This slow decay in the ACF suggests the presence of seasonal components and long-memory processes in demand.

The PACF revealed significant direct dependencies after controlling for intervening lags: strong short-term persistence (lags 1-6), weekly cyclical patterns (lags 7-9), bi-weekly effects (lags 13-15), and monthly patterns (lags 20-22, 27-29). This indicated complex temporal dynamics in energy consumption beyond simple day-to-day persistence.

Based on this analysis, we implemented the following lag features in our model to capture temporal patterns in energy consumption demand:

- **Lags 1-3:** Looks at short-term autoregressive effects, essential for next-day forecasting
- **Lags 6-8:** Captures weekly patterns in energy consumption
- **Lags 14:** Accounts for two-week cyclical patterns
- **Lags 21, 28:** Addresses monthly seasonality and longer-term effects



(a) ACF Residuals of the XGBoost Model (b) PACF Residuals of the XGBoost Model

Figure 6: ACF and PACF of the XGBoost Model

The ACF and PACF residual analysis showed substantially reduced autocorrelation after adding lag features. The Durbin-Watson score rose from ≈ 1 to ≈ 1.5 (ideal value is 2). There may be benefits from including lags at 4 and 5 which showed statistical significance, but the current setup captured the key lags while keeping the feature space manageable.

3.2.5 Public Holiday

Public holidays exhibited distinct demand patterns, showing significant drops that often exceeded weekend reductions. We implemented an `is_public_holiday` indicator using the `workalendar` package's `UnitedKingdom` calendar class, comprehensively capturing UK holidays (2020-2026) without maintaining a manual list. This feature allowed our model to anticipate demand reductions on holidays, improving forecast accuracy during these periods.

3.2.6 Cross-Validation

We implemented cross-validation using sklearn's `TimeSeriesSplit` that preserves chronological order of observations. For each model, we conducted an extensive grid search with results: Lasso ($\alpha = 0.0001$), Ridge ($\alpha = 8.0$), and XGBoost (depth=2, learning_rate=0.1, estimators=300).

All models showed comparable cross-validation RMSE values in the range of 0.032-0.033, with XGBoost requiring more tuning iterations but has the best validation set RMSE of **0.0321**.

3.2.7 Model Performance and Residual Analysis

Model performance was evaluated on a held-out test set (15% of most recent data) as shown:

Table 1: Model Performance on Test Set

Metric	Linear Regression	Lasso	Ridge	XGBoost
RMSE	0.0320	0.0321	**0.0319**	0.0324
R^2	0.9639	0.9637	**0.9642**	0.9630
Std. Dev.	0.0318	0.0319	**0.0317**	0.0321

Note: Bold values indicate best performance. RMSE and standard deviation are computed on log-transformed demand.

The standard deviation (shown in Table 1 as Std. Dev.) uses the calculations in Section 3.1.2

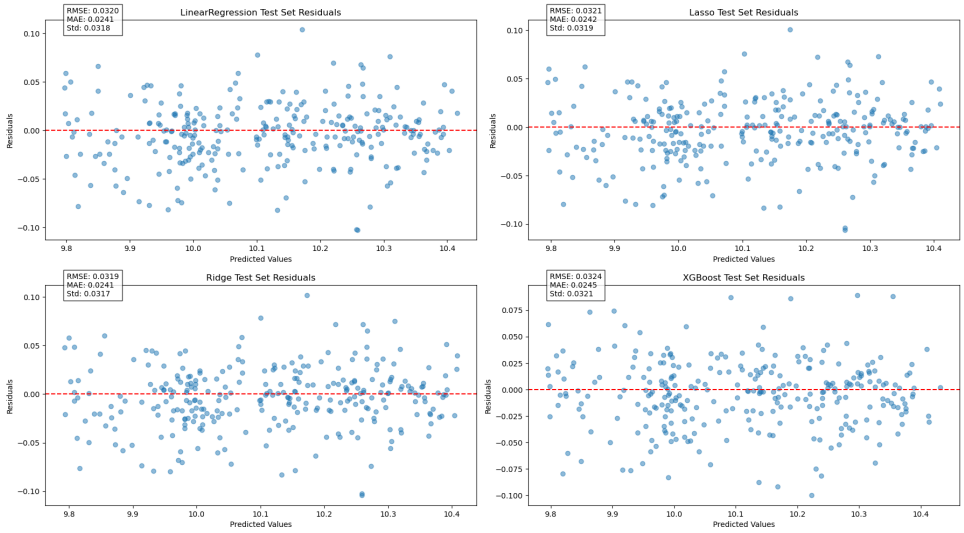


Figure 7: Residual Plots of All Model 2 Candidates

For each model, we calculated residuals on the log-transformed test data as follows:

$$\text{residuals} = y_{\text{test}} - y_{\text{predicted}} \tag{3}$$

In Figure 7, all models exhibit homoscedastic residuals scattered around zero with no clear pattern, indicating a good fit with the test set's demand.

3.3 Model 3 – Machine Learning Models

3.3.1 Justification for ML models selected

- **Tree-based ensemble regressors**: Random Forest Regressor, Gradient Boosting, XGBoost

  We used these tree-based ensemble regressors as they can handle nonlinear relationships, noisy and high dimensional data well, and they are also less sensitive to outlier and irrelevant features.

Furthermore, since tree based models split on features (that most reduces the error of the model), we thought we could also use it for feature selection purposes.

- **Kernel based**: Support Vector Regression

  We also decided to use a kernel based model (Support Vector Regression). Kernel based models are useful in capturing non-linear relationships as well as they can conduct feature transformation. We tested the model using the main kernel methods including the Linear Kernel, Radial Basis Function, and Polynomial Kernel.

- **Neural Network**:

  Finally, we also tested with the neural network. While we didn't want to place too much emphasis on the NN model as it has poor interpretability, we thought it would be useful as a point of comparison.

3.3.2 Feature Selection

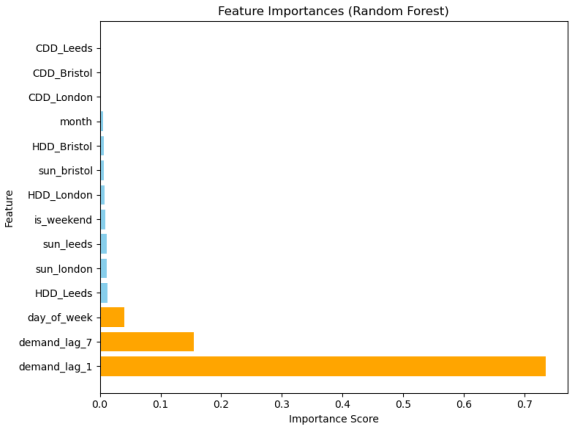
We used the the tree-based ensemble models both for prediction and feature selection. Since the feature importance score measures the proportion of contribution to the decision making in the model, this gave us valuable information to further narrow down the list of key features for the subsequent models.

Looking at Figure 8, both the Random Forest classifier and Gradient Boosting rated **Log demand of yesterday (lag 1)** , **Log demand of last week (last 7)**, and **Day of the week** as the most important features. Furthermore, while it was expected for the log demand lag features to perform well, it was interesting to see that the HDD features were far more valuable than the CDD measures. This reflects the UK's climate where heating dominates energy usage due to cool temperatures and limited air conditioning infrastructure.

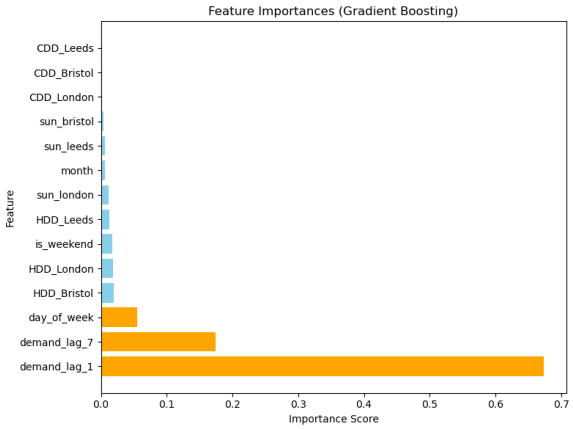
One caveat to note is that while the feature importance scores do show the decision making power of certain features, it doesn't account for issues of multi-collinearity. There is a chance that, some features explained the similar noise in the data due to similar confounding factors such as the temperature, so a low feature importance score must be not fully discounted in its importance in the model.

The inability of the Feature Importance Scores to fully capture the noise can be seen in the Scree Plot which plots the percentage of explain variance from each of the principal components. When comparing the Figure 9a with 9b, we can see that principal component with high weighting of lag demand 1, lag demand 7, and the HDD measures captured 35% of the variance in the linear regression model; however, the second principal component containing a high weighting of the CDD measures also explained around 18% of the variance in the regression model. Subsequently, the third and fourth principal components with the weekend and day of week variables as well as sunshine levels respectively had a explained variance value of 12-13%.

To briefly mention, we also see similar feature loadings (the weights) for different categories in each of the principal components (e.g. HDD values have similar loading factors as demand lag 1 and 7). Without diving into the math of PCA, this suggests some issues with multicollinearity as this indicates that the features with similar loading factors are moving together in relation to the variance of the data. Said another way, these features are contributing similarly to the same dimensions of variance.

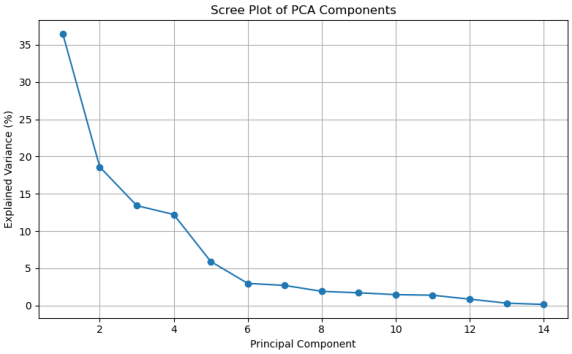


(a) Feature Importance from Random Forest Regressor

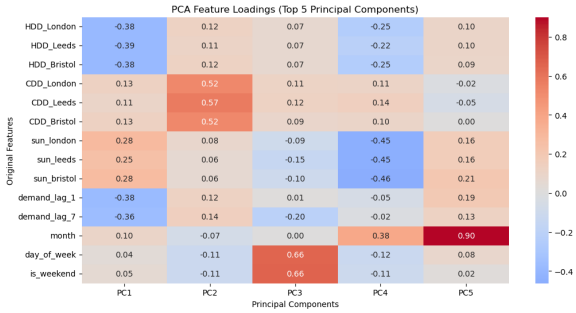


(b) Feature Importance from Gradient Boosting Regressor

Figure 8: Comparison of Feature Importance Scores from Random Forest and Gradient Boosting Regressors



(a) Cumulative Explain Variance with Principal Components in Linear Regression Model



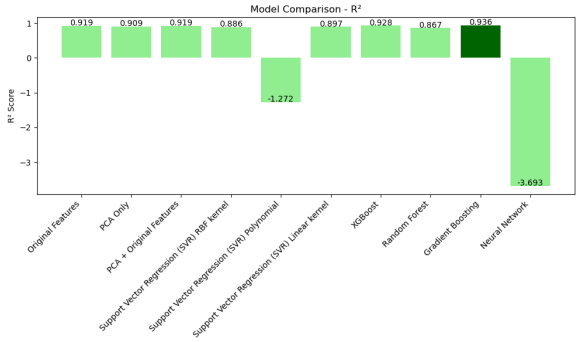
(b) RMSE values for Linear Regression vs ML Models

Figure 9: PCA Analysis of Features in Linear Regression Model

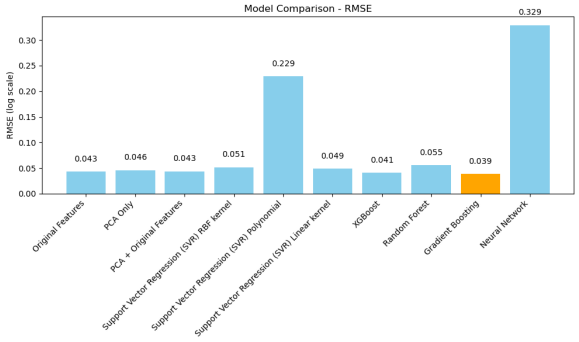
3.3.3 Model Performance

The various models tested using machine learning techniques (PCA) and models revealed that the ML models were not significantly better than our linear regression model, with the exception of the Gradient Boosting model and XGBoost model. Figure 10 show the comparisons of the R^2 values and root mean squared errors for each of the models. While slightly simplified measures of the performance of these models, the Support Vector Machine with the polynomial kernel parameter and the ReLu neural network (when tested with various values for the hidden layers), performed poorly.

Thus, for the prediction made using the ML models, we took the average log demand prediction for the best ML models (SVM rbf, XGBoost, Random Forest, and Gradient Boosting).



(a) R^2 values for Linear Regression vs ML Models



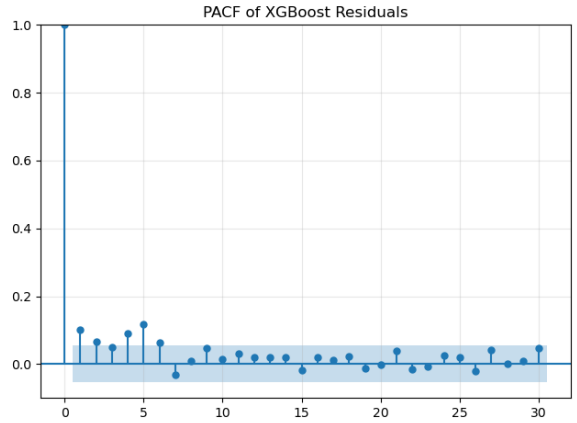
(b) RMSE values for Linear Regression vs ML Models

Figure 10: R^2 and RMSE For Linear Regression vs ML Models

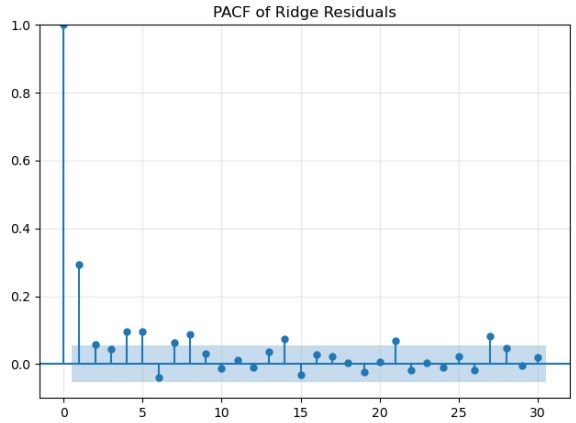
4 Final Forecasting Model

After extensive evaluation of multiple modelling approaches, we selected XGBoost as our final forecasting model, despite Ridge regression showing marginally better test error metrics. Our decision was based on several key considerations that extend beyond simple error metrics.

While Ridge regression achieved a slightly lower test RMSE (0.0319 vs 0.0324), XGBoost demonstrated superior residual properties. Analysis of residual autocorrelation revealed XGBoost captured temporal dependencies more effectively, with significantly reduced ACF and PACF patterns compared to Ridge regression (Figure 11). This advantage is crucial for energy forecasting, where properly accounting for temporal structure ensures more reliable predictions.



(a) PACF Residuals of XGBoost Model



(b) PACF Residuals of Ridge Regression Model

Figure 11: PACF Comparison: XGBoost Residuals Show Fewer Significant Lags Than Ridge, Indicates Better Temporal Structure Captured (ACF Not Shown to Save Space)

Then, the tree-based architecture of XGBoost naturally handles the high multicollinearity in our feature set (8 features with Variance Inflation Factor > 10), particularly among demand lags. This provides greater stability when correlation structures change in new data.

Our XGBoost model was optimised through cross-validation with the following parameters:

- Tree depth: 2 (prevents overfitting)
- Learning rate: 0.1 (balancing training speed and generalisation)

- Number of estimators: 300 (captures model complexity)
- Regularization: $\lambda = 0.5$, $\alpha = 0$ (L2 regularization shrinking coefficients like Ridge)

Our final XGBoost model incorporates these key features identified through our analysis:

- Demand lags(1-3, 6-8, 14, 21, 28): Capturing daily, weekly, and monthly temporal patterns
- HDD: Quantifying heating requirements that dominate UK energy consumption
- Calendar indicators: Weekend and public holiday dummies capture demand drops
- Temperature metrics: Correlated with $log(demand)$ to capture consumption patterns

We excluded CDD and sunshine duration after testing showed minimal contribution to model performance, reflecting the UK's limited cooling infrastructure and the weak correlation between sunshine and log-transformed demand.

For the standard deviation of our probabilistic forecast, we used the method from Section 3.1.2, which accounts for both demand and temperature forecast errors. This approach provides more comprehensive prediction intervals than using residual standard deviation alone, enabling more reliable risk assessment after incorporating temperature forecasts.

The model achieved an R^2 of 0.963 on the test set, explaining 96.3% of the variance in $log(demand)$ and outperforming both our baseline model (Section 3.1) and ML alternatives (Section 3.3). The Durbin-Watson statistic of 1.754 indicates minimal residual autocorrelation, while normally distributed errors (Jarque-Bera p-value > 0.05) validate our model assumptions.

In conclusion, XGBoost's exceptional handling of time-series patterns, effective management of multicollinearity, and strong performance justify its selection as our final forecasting model.

5 Conclusions and Reflections

5.1 What Additional Information Would Have Helped

When forecasting, there is a constant need for having more information to discover that unknown function that maps the inputs to the target variable (energy demand). Having access to real-time electricity prices would have enhanced the model to reflect demand responses in this dynamic market, particularly during high-price events. Additionally, disaggregated demand data, such as residential, commercial, or regional consumption, could have improved the accuracy of our models. This level of granularity would help uncover segment specific patterns and produce a more targeted forecasting.

5.2 Key Takeaways

This project highlighted the importance of combining domain knowledge with empirical model testing. We learned that even relatively simple models can achieve strong performance when built on well-selected features, such as lagged demand and temperature indicators. Among all variables explored, Heating Degree Days (HDD), demand lags, and day of the week, proved to be the most influential, which reinforces the dominant role of heating needs in UK electricity consumption.

Forecasting success depended not just on choosing the most complex or high performing algorithms, but most importantly on understanding temporal structures, preventing overfitting, and constructing appropriate validation procedures. Our iterative model development, from linear regression to XGBoost, emphasised that interpretability, residual behaviour, and robustness can outweigh (marginal) gains in RMSE.

5.3 Next Steps and Improvements

Moving forward, there are several ways our approach could be improved. First, to better quantify uncertainty, we propose implementing bootstrap methods and incorporating forecast uncertainty from weather predictors into the standard deviation of our probabilistic forecasts.

Additionally, while our final model reduced autocorrelation, the PACF plot showed that some dependencies remained unaccounted for. Incorporating (S)ARIMA errors in a hybrid model could help capture these patterns.

We also addressed multicollinearity using Principal Component Analysis (PCA), which helped reduce redundancy in our feature space. However, more exploration of advanced dimensionality reduction techniques or hybrid models could offer additional robustness, especially when having larger datasets or more complex weather features.

Finally, ensemble methods that integrate linear and non-linear components, or stacking models across forecasting horizons, present a path for improving accuracy and generalisability.

References

Hong, T. and Wang, Z. (2014). Weather impact on residential building energy consumption in the us: A statistical regression analysis. *Energy and Buildings*, 69:188–194.

Hor, C.-L., Watson, S. J., and Majithia, S. (2005). Analyzing the impact of weather variables on monthly electricity demand. *IEEE Transactions on Power Systems*, 20(4):2078–2085.

Sailor, D. J. and Muñoz, J. (1997). Sensitivity of electricity and natural gas consumption to climate in the usa—methodology and results for eight states. *Energy*, 22(10):987–998.

Valor, E., Meneu, V., and Caselles, V. (2001). Daily air temperature and electricity load in spain. *Journal of Applied Meteorology*, 40(8):1413–1421.