Steel Industry Energy Consumption: A Classification Model Analysis

Robert Iannuzzo

3/17/2025

The Data

Features

- **Usage_kWh** – Total energy consumption in kilowatt-hours.
- **Lagging_Current_Reactive_Power_kVarh** – Reactive power when current lags the voltage.
- **Leading_Current_Reactive_Power_kVarh** – Reactive power when current leads the voltage.
- **CO_2 (tCO_2)** – Carbon dioxide emissions in metric tons.
- **Lagging_Current_Power_Factor** – Efficiency measure when current lags the voltage.
- **Leading_Current_Power_Factor** – Efficiency measure when current leads the voltage.
- **NSM (Number of Seconds from Midnight)** – Time of day in seconds.
- **WeekStatus** – Encoded as 0 (Weekday) or 1 (Weekend).
- **Day_of_week** – Encoded as an integer (1–7) representing the day of the week.

Targets

- Light_Load
- Medium_Load
- Maximum_Load

Exploratory Data Analysis

- We have 9 columns, 3 targets and 35040 data points
- Since we our targets is a set of 3 we have a classification problem and will need to use classification methods

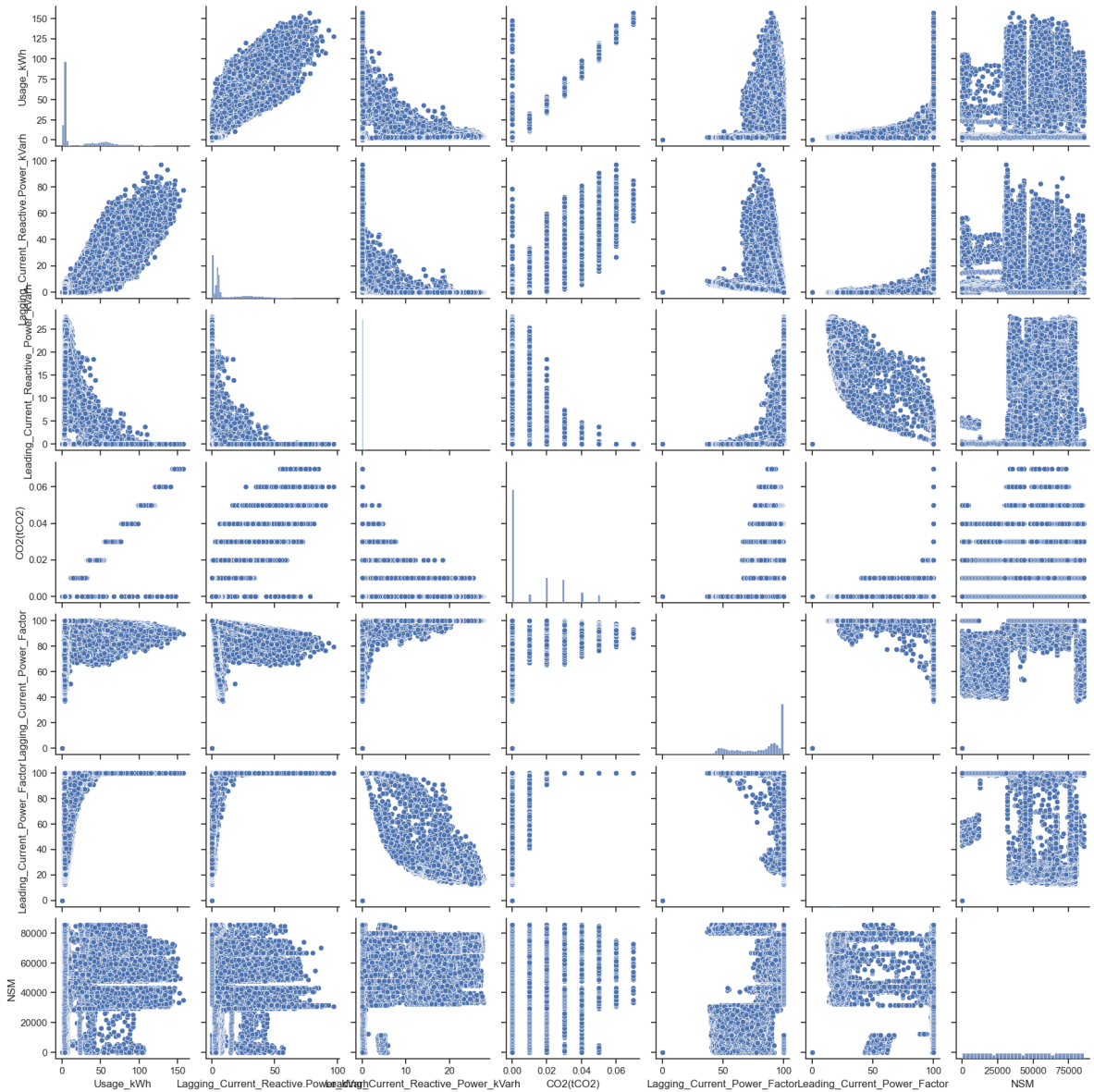
Regression Models

- Regression models assume continuous features and try to fit linear relationships to the data
- We clearly do not have continuous features here (there are only 3 possible outputs)
- MSE scores and R^2 scores for regression models

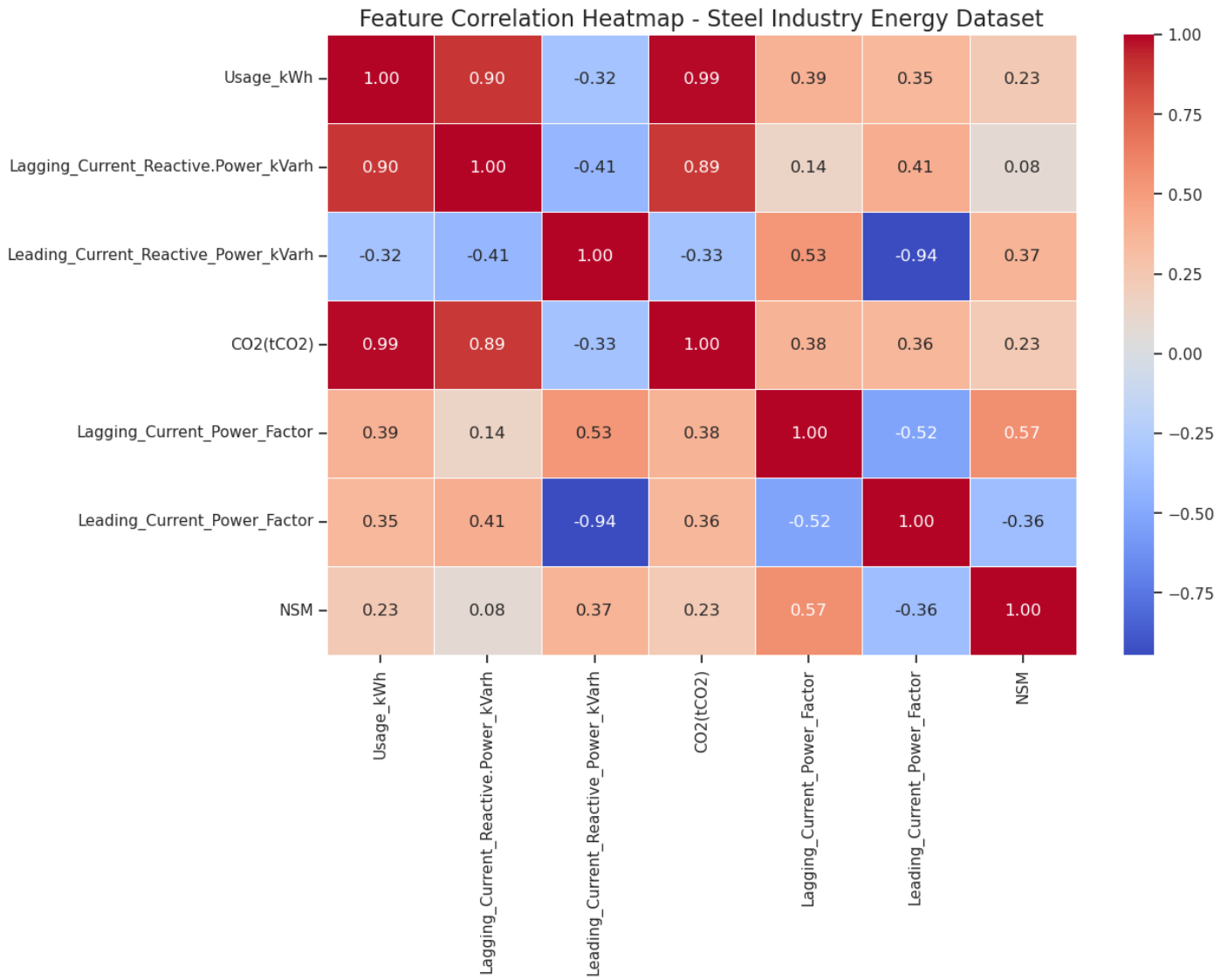
	Model	MSE	R2 Score
0	Ridge	0.264211	0.575153
1	Lasso	0.314652	0.494045
2	PCR	0.286111	0.539937
3	PLS	0.271472	0.563477

- It is clear these are not performing well

Scatterplot of Key Features



Heatmap of Feature Correlations



Logistic Regression

- We started with logistic regression and moved on from there
- Decent, but not excellent, accuracy

```
Logistic Regression Accuracy: 0.7591
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.90      0.90      3592
           1       0.63      0.60      0.61      1984
           2       0.58      0.63      0.60      1432

    accuracy                           0.76      7008
   macro avg       0.71      0.71      0.71      7008
weighted avg       0.76      0.76      0.76      7008
```

Linear Discriminant Analysis

- This model performed worse than the logistic regression model

```
LDA Accuracy: 0.7476
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90      3592
           1       0.62      0.54      0.58      1984
           2       0.55      0.71      0.62      1432

    accuracy                           0.75      7008
   macro avg       0.70      0.71      0.70      7008
weighted avg       0.76      0.75      0.75      7008
```

Quadratic Discriminant Analysis

- This model performs even worse than the linear discriminant analysis

```
QDA Accuracy: 0.7156
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.87      0.88      3572
           1       0.67      0.30      0.41      1937
           2       0.50      0.89      0.64      1499

    accuracy                           0.72      7008
   macro avg       0.69      0.68      0.64      7008
weighted avg       0.75      0.72      0.70      7008
```

K-Nearest Neighbors

- We had a good initial accuracy with this model

```
KNN Accuracy: 0.8770
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      3572
           1       0.79      0.79      0.79      1937
           2       0.79      0.79      0.79      1499

    accuracy                           0.88      7008
   macro avg       0.85      0.85      0.85      7008
weighted avg       0.88      0.88      0.88      7008
```

K-Nearest Neighbors cont.

- I also tuned the model to find the best K
- We ended up finding that K=7 was the best

```
Best K: {'knn__n_neighbors': 7}
Best Accuracy: 0.8827
```

- Here is the full report

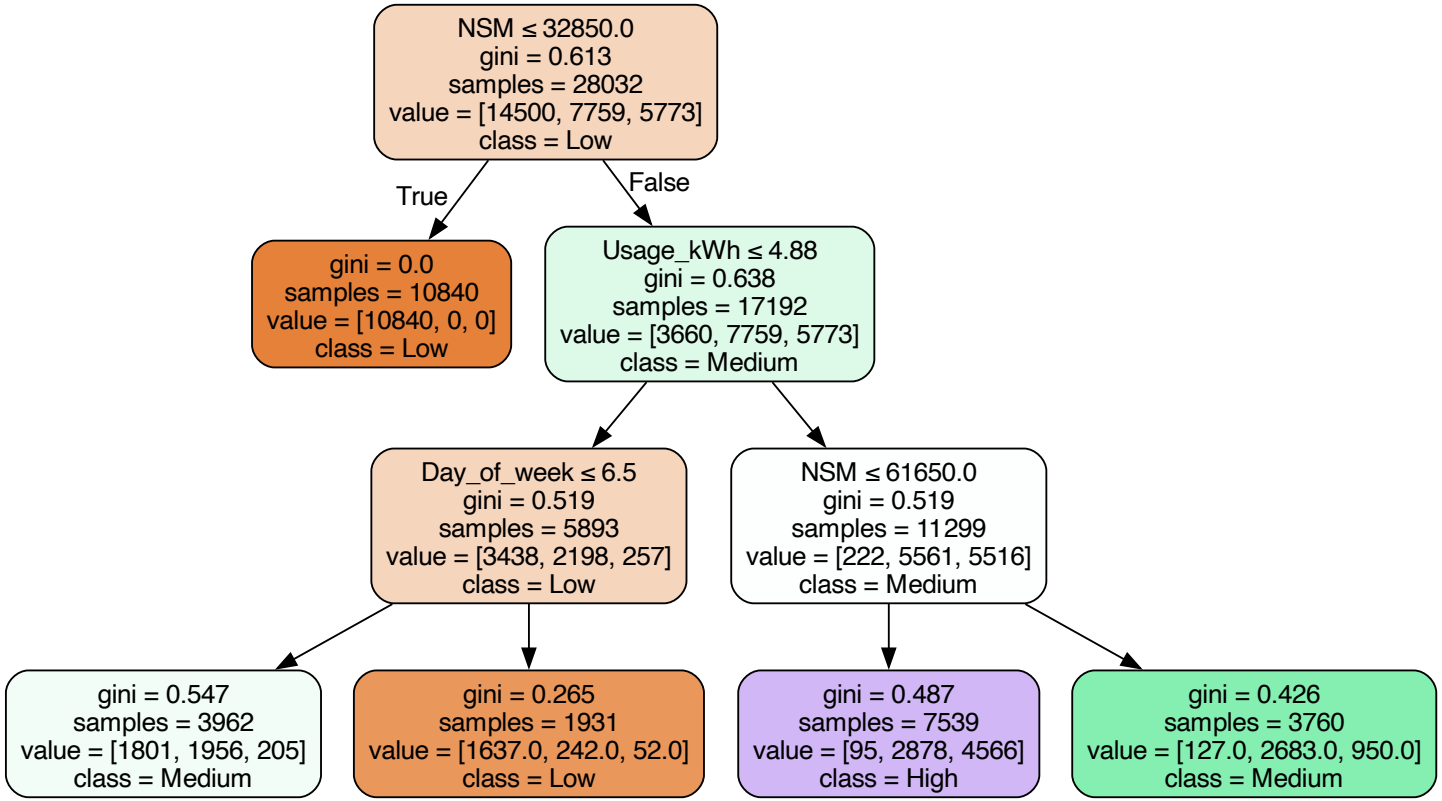
```
KNN (k=7) Accuracy: 0.8809
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      3572
           1       0.80      0.80      0.80      1937
           2       0.80      0.80      0.80      1499

    accuracy                           0.88      7008
   macro avg       0.85      0.85      0.85      7008
weighted avg       0.88      0.88      0.88      7008
```

Decision Tree Model

- Our first tree model did not work as well as KNN (however, it predicts light load very well)

```
Decision Tree Accuracy: 0.7687

Classification Report (Decision Tree):
              precision    recall  f1-score   support

           0       0.98      0.85      0.91      3592
           1       0.60      0.61      0.60      1984
           2       0.60      0.78      0.68      1432

    accuracy                           0.77      7008
   macro avg       0.73      0.75      0.73      7008
weighted avg       0.79      0.77      0.78      7008
```

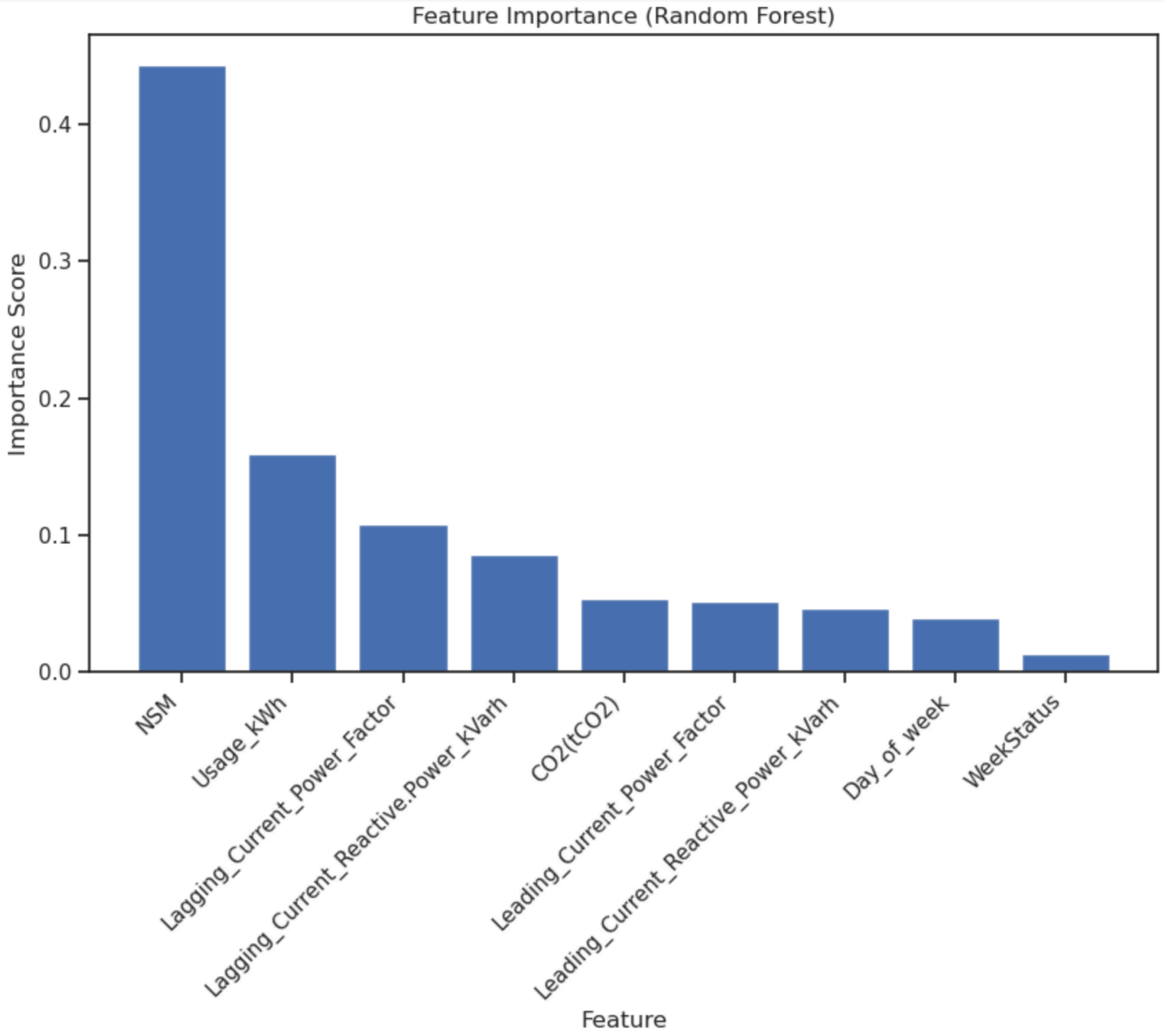
Decision Tree Visualization



Random Forest

- This performed better than any model so far

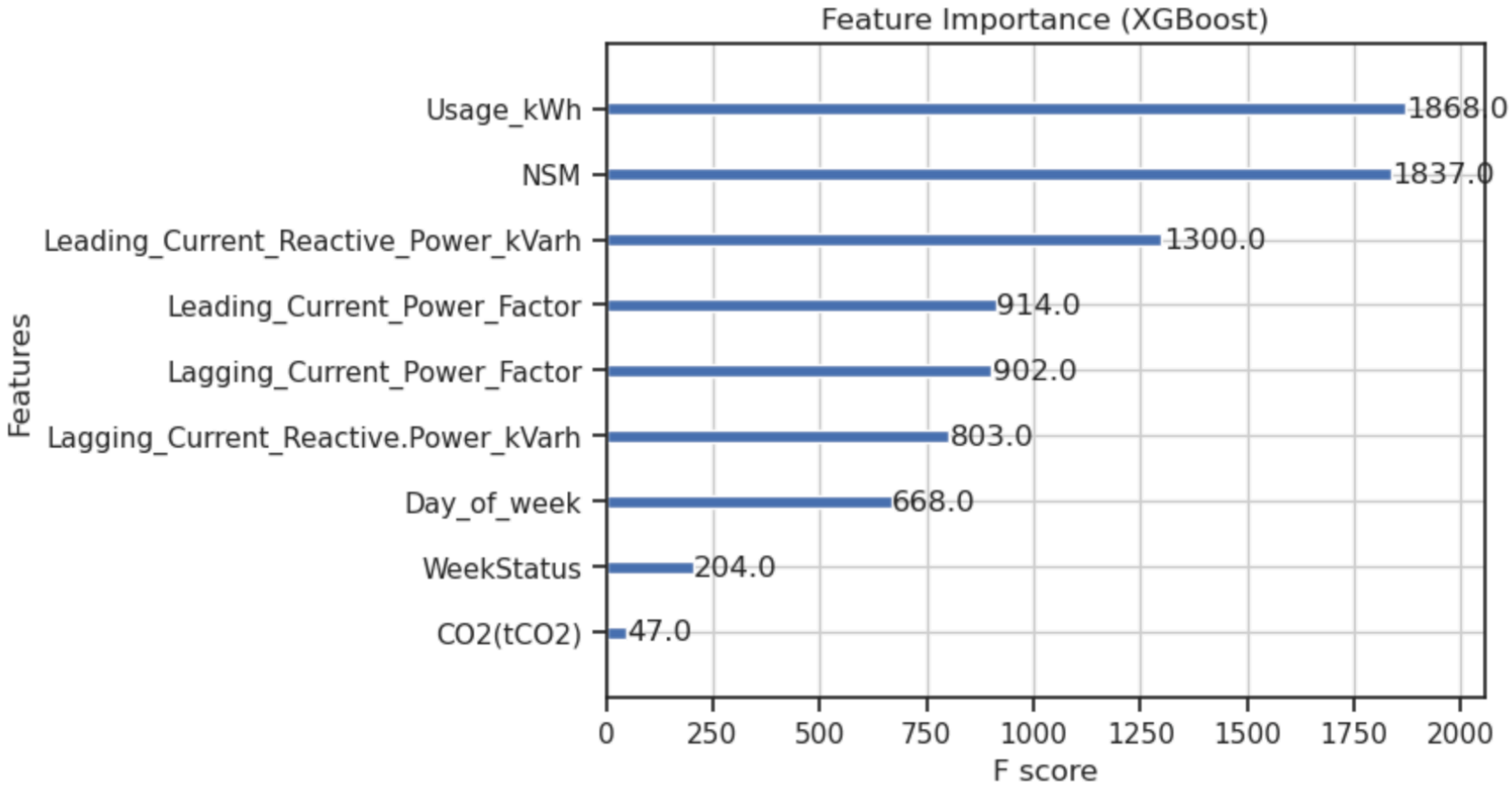
```
Random Forest Accuracy: 0.9092
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      3592
           1       0.86      0.84      0.85      1984
           2       0.82      0.84      0.83      1432

    accuracy                           0.91      7008
   macro avg       0.88      0.88      0.88      7008
weighted avg       0.91      0.91      0.91      7008
```


Feature Importance (Random Forest)
Importance Score
0.0
0.1
0.2
0.3
0.4
NSM
Usage_kWh
Lagging_Current_Power_Factor
Lagging_Current_Reactive.Power_kVarh
CO2(tCO2)
Leading_Current_Power_Factor
Leading_Current_Reactive_Power_kVarh
Day_of_week
WeekStatus
Feature

XGBoost

- XGBoost also ran very well

```
XGBoost Accuracy: 0.9054
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      3572
           1       0.85      0.82      0.83      1937
           2       0.81      0.85      0.83      1499

    accuracy                           0.91      7008
   macro avg       0.88      0.88      0.88      7008
weighted avg       0.91      0.91      0.91      7008
```


Feature Importance (XGBoost)
Usage_kWh
NSM
Leading_Current_Reactive_Power_kVarh
Leading_Current_Power_Factor
Lagging_Current_Power_Factor
Lagging_Current_Reactive.Power_kVarh
Day_of_week
WeekStatus
CO2(tCO2)
1868.0
1837.0
1300.0
914.0
902.0
803.0
668.0
204.0
47.0
Features
0
250
500
750
1000
1250
1500
1750
2000
F score

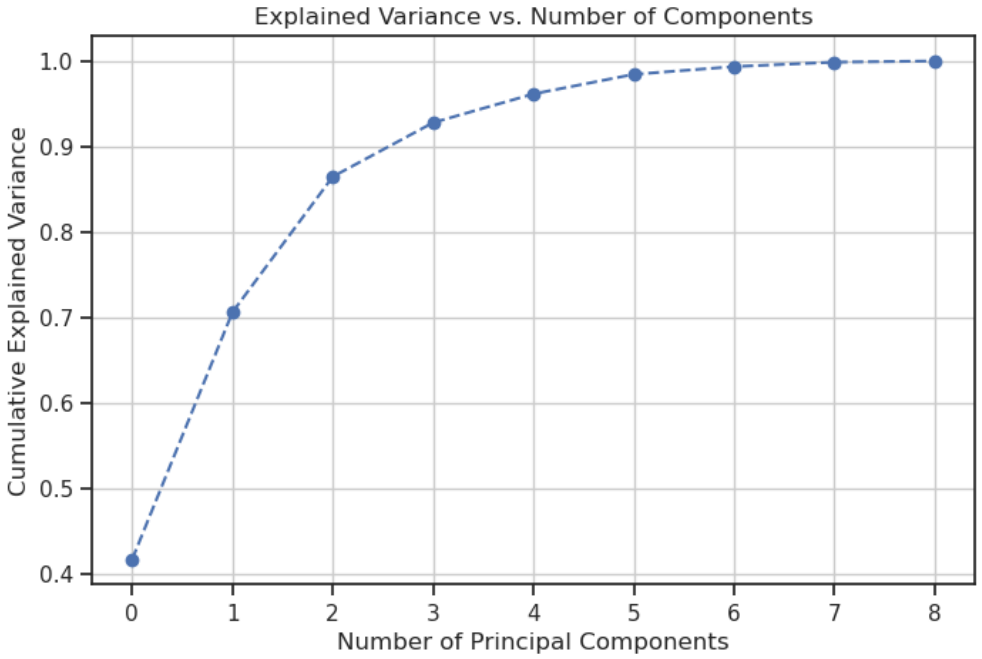
Support Vector Machine

- I also waned to look at the SVM model despite Random Forest and XGBoost working out well
- SVM did not perform nearly as well as the aforementioned models

```
SVM Accuracy: 0.8209

Classification Report (SVM):
              precision    recall  f1-score   support

           0       0.91      0.95      0.93      3592
           1       0.75      0.62      0.68      1984
           2       0.68      0.77      0.72      1432

    accuracy                           0.82      7008
   macro avg       0.78      0.78      0.78      7008
weighted avg       0.82      0.82      0.82      7008
```

Principal Component Analysis

- We also explored whether or no reducing the number of components would help our models
- Number of component determination:



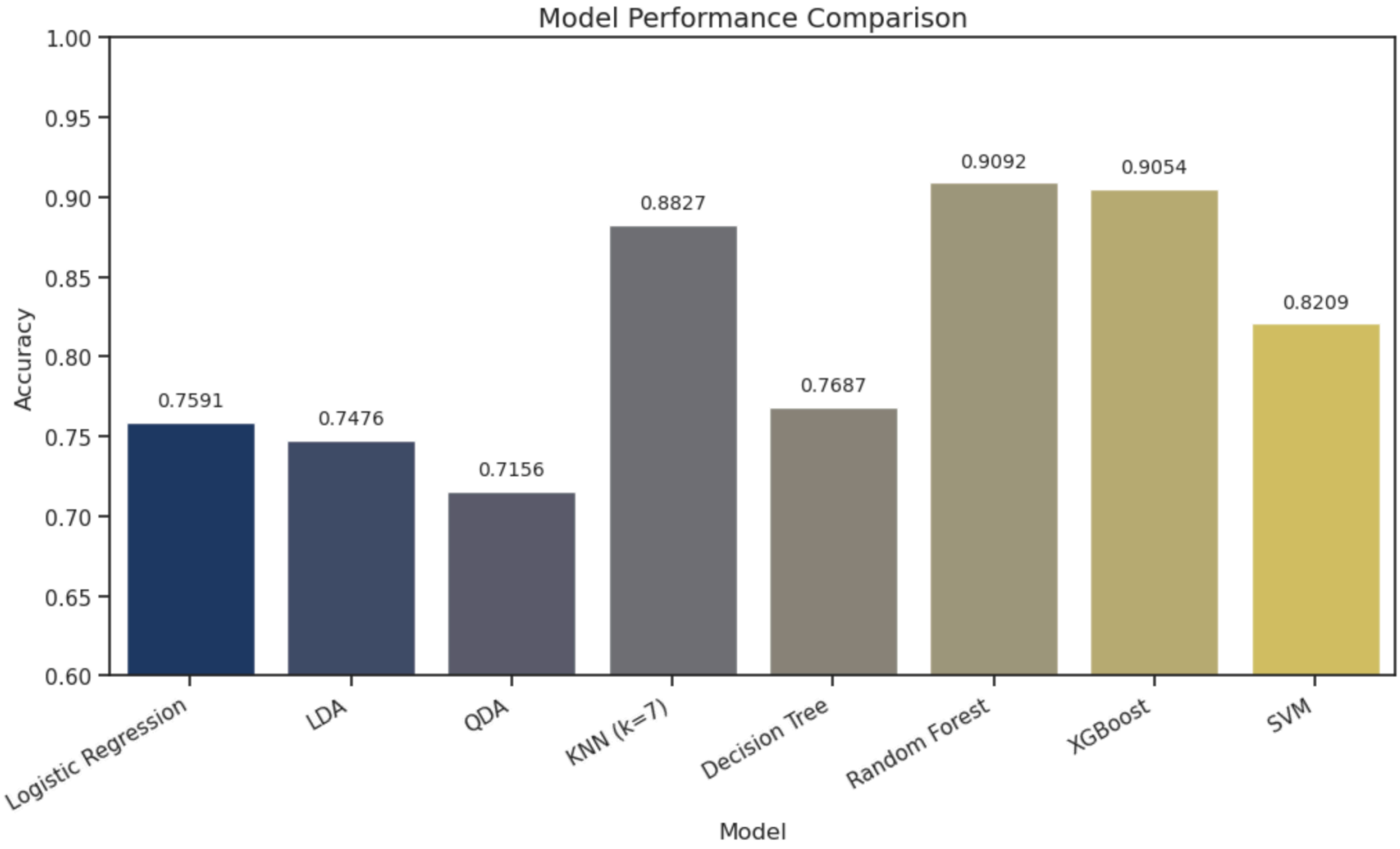
PCA continued

- We ran a few models with PCA components set to 3

```
SVM Accuracy (Original Data): 0.8126

SVM Accuracy (After PCA, 3 Components): 0.7800

Logistic Regression Accuracy (Original Data): 0.7561
Logistic Regression Accuracy (After PCA, 3 Components): 0.7205
```

- It is clear that PCA is not helping our model accuracy so I decided not run it for every model explored so far

Final Model Performance Comparison



Conclusion & Key Takeaways

- Random Forest & XGBoost performed best
- PCA didn’t help due to loss of critical features
- Decision tree & feature importance aid interpretability
- Next steps: Optimize hyperparameters, explore deep learning

Thank You! Questions?

References

- V E, S., Shin, C., & Cho, Y. (2021). Steel Industry Energy Consumption [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C52G8C.