**Wai Ying Wong**

**Python Code Link :** [**https://www.kaggle.com/code/winniewg05/classification-on-load-type**](https://www.kaggle.com/code/winniewg05/classification-on-load-type)

# **Classification model on Steel Industry Energy Consumption Load Type**

**Introduction**

In this report, we are using a cleaned steel industry dataset stored on UCI Machine Learning Repository and the Korea Electric Power Corporation website that contains information on thirty thousand steel energy consumption. This model is based on energy usage patterns and predicts the load type using classification models. The purpose of this model is helping to enhance the steel energy consumption system by automating load type categorization and improving the efficiency in the energy manufacturers.

**Dataset Preparation**

This dataset is obtained from UCI Machine Learning Repository and is collected from a small-scale steel industry called DAEWOO Steel Co. Ltd in Gwangyang in South Korea. The steel energy data includes the carbon dioxide emissions, lagging and leading current reactive power, the lagging and leading current power factor, and load types. Below is the original data file, it does not have any missing values, and the dataset's attributes are outlined.

**Table

Description automatically generated**

**Summary of the data features**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Description** | **Type** |
| date | Date & Time | Categorical |
| Usage\_kWh | Industry Energy Consumption Continuous kWh | Numerical |
| Lagging\_Current\_Reactive.Power\_kVarh | Lagging Current reactive power Continuous kVarh | Numerical |
| Leading\_Current\_Reactive\_Power\_kVarh | Leading Current reactive power Continuous kVarh | Numerical |
| CO2(tCO2) | tCO2(CO2) Continuous ppm | Numerical |
| Lagging\_Current\_Power\_Factor | Lagging Current power factor Continuous % | Numerical |
| Leading\_Current\_Power\_Factor | Leading Current Power factor Continuous % | Numerical |
| NSM | Number of Seconds from midnight Continuous S | Numerical |
| WeekStatus | Week status (Weekend (0) or a Weekday (1)) | Binary |
| Day\_of\_week | Day of week (Sunday to Saturday) | Categorical |
| Load\_Type | Load Type (Light Load, Medium Load, Maximum Load) | Categorical |

**EDA**

This is the summary statistics, and the dataset has total 35040 rows.

Table

Description automatically generated

The bar and pie charts are to understand the target distribution, Load Type. Light Load has the highest proportion, with 52% of the total, especially on weekends and Sundays. Medium Load and Maximum Load have similar distributions of 28% and 21%, respectively.

Chart

Description automatically generated

Chart, bar chart, waterfall chart

Description automatically generated

More than 15000 Light Load and half of Medium Load at 0.00 CO2, while more than half of Maximum Load emitted 0.02 and 0.03 CO2. The right chart shows a right-skewed boxplot and a few outliers above 0.05 CO2.

Graphical user interface, diagram

Description automatically generated

Most energy consumption usages are below 4.5kWh, and more than 15000 records are between 2.5 and 4.5kWh. Usages over 20kWh are mainly from Maximum Load and Medium Load. The maximum usage is at around 120kWh. Lagging Reactive Power is mainly below 10kVarh, and most of the Medium Load are located at 0 kVarh. The majority of Lagging Reactive Power are at 0kVarh and the maximum is at 5kVarh. These three features also displayed as left-skewed boxplots and they have many outliers.

Graphical user interface, diagram

Description automatically generated

Lagging Power Factor mainly distributes between 40 to 100 and more than 8000 counts at 100. The Medium and Maximum Load gather at the range of 80 and 100. Besides, the Leading Power Factor only locate at 100, with a few outliers at below 100 and above 20. They have both left-skewed boxplots.

Diagram

Description automatically generated with low confidence

From the NSM chart below, we can see that Light Load has 35000 NSM or below. Between 35000 to 60000 NSM are contributed by Maximum Load, and from 60000 to 80000 are mainly from Medium Load. Over 82000 NMS are only from light load type. The NMS boxplot is the only chart that shows a normal distribution, the lower quartile is slightly higher than 20000, and the upper quartile is around 64000. The minimum starts from 0, and the maximum is over 80000. The median is close to 42000, and no outlier.

Chart, histogram

Description automatically generated

The scatterplot at the left shows a moderately strong and positive relationship. As Lagging Reactive Power increases, Usage also tends to increase. While the scatterplot at the right shows a negative association between Leading Reactive Power and Usage, as Leading Reactive Power decreases, the Usage increases.

Chart, scatter chart

Description automatically generated

The scatterplots of usage and Lagging Power Factor or Usage and Leading Power Factor do not show very strong positive relationships but we can still see that the Lagging or Leading Power Factor increase, the Usage also high as well.

Chart

Description automatically generated

The pairplot plot a pairwise relationship of the numeric values. The values are coloured based on load type. This helps us to discover more correlations. There are other positive correlations such as Lagging Reactive Power and CO2, CO2 and Usage. Also, there is negative correlations such as Leading Reactive Power and Leading Power Factor.

Diagram

Description automatically generated with medium confidence

**Data Pre-Processing**

For the modelling dataset, column names have been edited and the value of the categorical columns (Day of week and Load Type) are changed to numeric with the label encoder by assigning each category to a number during the data transformation. Label encoding is chosen instead of one-hot encoding because weekdays are ordinal, and label encoding does not add any new columns and lower computations for the modelling. Load Type 0 is Light Load; 2 is Medium Load; 1 is Maximum Load. Apart from that, the ‘date column’ has been removed because it does not show any correlation with other data. This is the cleaned version of the dataset for the modelling.

**Table

Description automatically generated**

This is the heatmap after data processing, we can see that Load Type has correlation with NMS, lagging power factor usage and CO2.

Table

Description automatically generated

**Feature Observation & Hypothesis**

The charts above show that features such as NMS, Lagging Power Factor, Usage, CO2 and Week Status have strong positive relations with the Load Type. Therefore, the hypothesis is that the features mentioned before will impact the prediction the Load Type.

**Model Planning**

The dataset has been split into training and testing: the test size is 25%, and the random state is 42. The numerical features and categorical columns (Day of week and Week Status) are separated for the preprocessor because numerical features will perform feature scaling, and categorical columns will be excluded with a small range of data. We use standardization for feature scaling, subtracting the mean value and dividing by the standard deviation so that the resulting distribution has a unit variance. Also, standardization is much less affected by outliers.

We will train with various Supervised Machine Learning models. Logistic regression, KNN, SVM Linear, and SVM RBF are capable on binary classifiers. Decision Tree, Naive Bayes and Random Forest can handle multiple classes natively. AdaBoost is a boosting algorithm with decision trees with one level which means with Decision trees with only one split. Furthermore, XGBoost and CatBoost are gradient boosting ensemble algorithms that fit boosted decision trees by minimizing an error gradient. Given that the dataset has many samples (>35,000 rows) with a limited number of features which a more complex models, such as tree-based and boosting algorithms models, would have better accuracy because they can lead to higher predictive performance and have better handling of a mix of numerical and categorical features.

**Model Implementation**

We use the pipeline to chain the preprocessing and model fitting. ColumnTransformer is used to create and apply separate transformers for numerical and categorical features in the preprocessor. Then we use a loop to fit piped processed data in all the selected models and the results are appended to a table. After that, we choose the best for modelling based on accuracy and the result table is listed below ranked by accuracy.

Table

Description automatically generated with medium confidence

Given that the performance of XGBoost is the best, it was chosen for hyper-parameter tuning by Grid Search with Cross Validation. Cross Validation is a method to generalize the machine learning model and to avoid overfitting our machine learning implementation. Grid Search CV will use cross-validation to evaluate all possible combinations of hyperparameter values. We can follow the codes and search for this model's best combination of hyperparameter values.

It can be observed that the best parameters are: 'learning\_rate': 0.05, 'max\_depth': 9, 'n\_estimators': 140.

**Results Interpretation and Implications**

After finding out the best parameters by optimizing the accuracy score, we tried the model on the test data set.

**Confusion Matrix**

A study of the confusion matrix will provide an understanding into the model’s predictive power from the figure. By looking at the confusion matrix, it can be observed that the model is predicting very well on 0 (Light Load) which as 4398 predictions match the actual. While predicting 1(Maximum Load) and 2 (Medium Load) has similar performance, most of their prediction can match the actual, and they are less likely to predict to be 0 (Light Load). However, there are some predictions which actual should be 1(Maximum Load) but 259 of them predict to 2 (Medium Load), while there are 366 actual (Medium Load) and predict as 1 (Maximum Load).

**Chart

Description automatically generated**

**Classification Report**

From the classification report, we can see the precision, recall, and f1-score for each class. Class 0 (Light Load) is observed to have a relatively high positive rate which are all above 0.97. Even though classes 1(Maximum Load) and 2 (Medium Load) have a relatively lower rate which is 0.80 and 0.85 respectively, they still have a high positive rate. These cumulative scores can be found in the macro avg and weighted avg in the table. The high f1-score in class 0 led to the overall weighted avg to be 0.90.

**Table, calendar

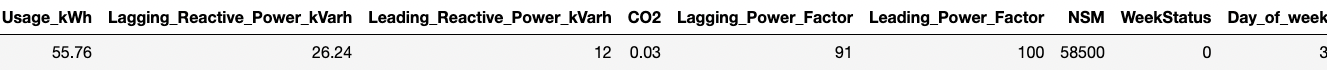
Description automatically generated**

**Analysis**

Overall, the classification shows high accuracy, especially on class 0 (Light Load), even the 1 and 2 are not as high as class 0, but it is still over 80%. Class 1 and 2's accuracy are lower because the Light Load contributes more than half of the dataset, and Medium or Maximum Load only has around 20% of the data so that an imbalanced dataset may occur. Therefore, to improve the model, I suggest balancing the data by obtaining more datasets, especially for Medium and Maximum Load, or proportioning the data set using the ratio. Besides, we can use Synthetic Minority Oversampling Technique (SMOTE) which generates synthetic data for the minority classes by randomly picking a point from the minority class and computing the k-nearest neighbours for this point.

**Out Of Sample Prediction**

These are randomly selected the figures for out of sample predictions, and using Grid\_CBC, which is the final model for prediction, will generate the result for the Load Type class. Hence, in the real world, users can fill in the required data into the columns, or they can have another sheet to fill in the required columns. After fitting the data into the model, the system can automatically predict the load type data.



Graphical user interface, text, application

Description automatically generated

**Conclusion**

This report is about building a classification model to help predict load type based on the steel energy consumption data. The target has three classes defining the level of the load type (Light, Medium, and Maximum Load). In the modelling process, ten classification models are trained, and we selected XGBoost because the model's overall accuracy is ~ 90%. However, the accuracy is weighted with the high accuracy on class 0 (Light Load), and the model still has a few false predictions for classes 1 and 2 (Medium and Maximum Load). Therefore, there is scope for improvement before the model is deployed in a real-world scenario.

**References**

Steel Industry Energy Consumption Dataset Data Set. (2021). UCI Machine Learning Repository.

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. (2019). Aurélien Géron

Practical Machine Learning with Python. (2018). Dipanjan Sarkar, Raghav Bali and Tushar Sharma

10 Techniques to deal with Imbalanced Classes in Machine Learning. (2018). Benai Kumar URL <https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/>

# Seaborn Pairplot. (2021). [Kalsoom Bibi](https://linuxhint.com/author/kalsoombibi/) https://linuxhint.com/seaborn-pairplot/

AdaBoost Classifier Example In Python. (2019). Cory Maklin. URL https://towardsdatascience.com/machine-learning-part-17-boosting-algorithms-adaboost-in-python-d00faac6c464

# AdaBoost Algorithm – A Complete Guide for Beginners. (2021). [Anshul Saini](https://www.analyticsvidhya.com/blog/author/anshul508/)  <https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/>

# How to Use the ColumnTransformer for Data Preparation. (2019). [Jason Brownlee](https://machinelearningmastery.com/author/jasonb/) https://machinelearningmastery.com/columntransformer-for-numerical-and-categorical-data/