OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

CT5165 Principles of Machine Learning
Assignment 3 – Time Series Forecasting

Zhe Jing Chin - 22221970 - 1CSD1

Question 1

In this assignment, the open source package - *scikit learn* is chosen as the machine learning package. It provides simple and efficient tools for predictive data analysis and is accessible by the public. The reason is because it has a wide range of supervised regression models that are suitable for regression tasks. Besides, it comes with other features including preporcessing and model selection functions that are useful when building the hypothesis. For this task, I have chosen *LinearRegression* and *ElasticNet* to predict the electricity consumption.

LinearRegression is an ordinary least square linear regression model. It is the most common algorithm to estimate coefficients that describe the linear relationships between one or more independent variables (features) and the dependent variable (target). The objective of this algorithm is to minimize the sum of squared residuals. (Sharma, 2022)[2]

ElasticNet is an advanced version of LinearRegression which penalized the regression model. It includes both L1 and L2 as regularizer. However, we could simply assign the weight of L1 and L2 penalties with the hyperparameter – “alpha”. By assigning “alpha” to 0, the algorithm becomes an ordinary least square. (Brownlee, 2020)[1]

Question 2

Before we train the models, data cleaning process is necessary. The provided dataset has a highly right skewed distributed records on the target, thus normalization is applied to the dependent variable to increase the performance of model as it could help algorithms to learn the pattern better. *Box-cox transformation* is used in this case as they provided a fairly normal distribution on the data.

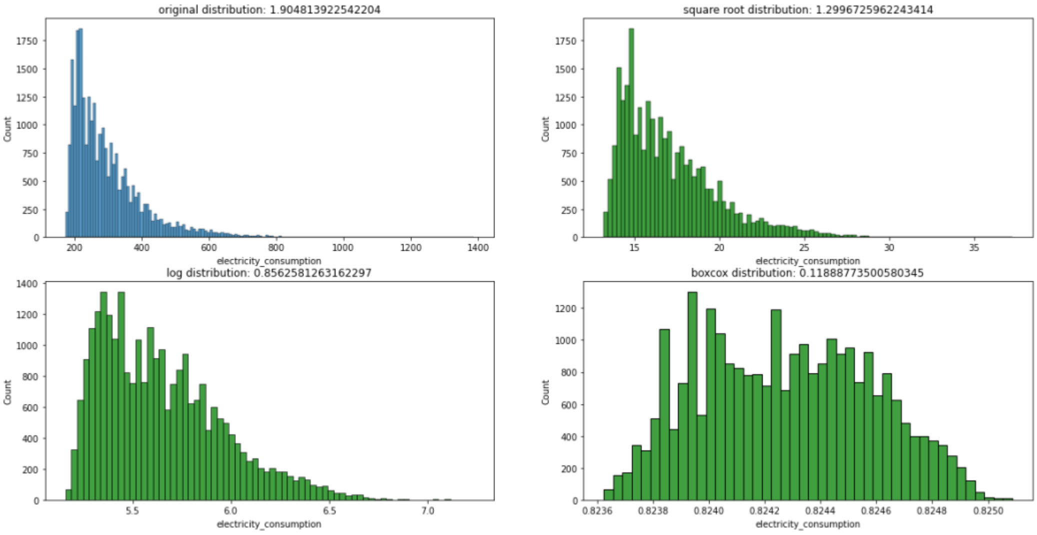
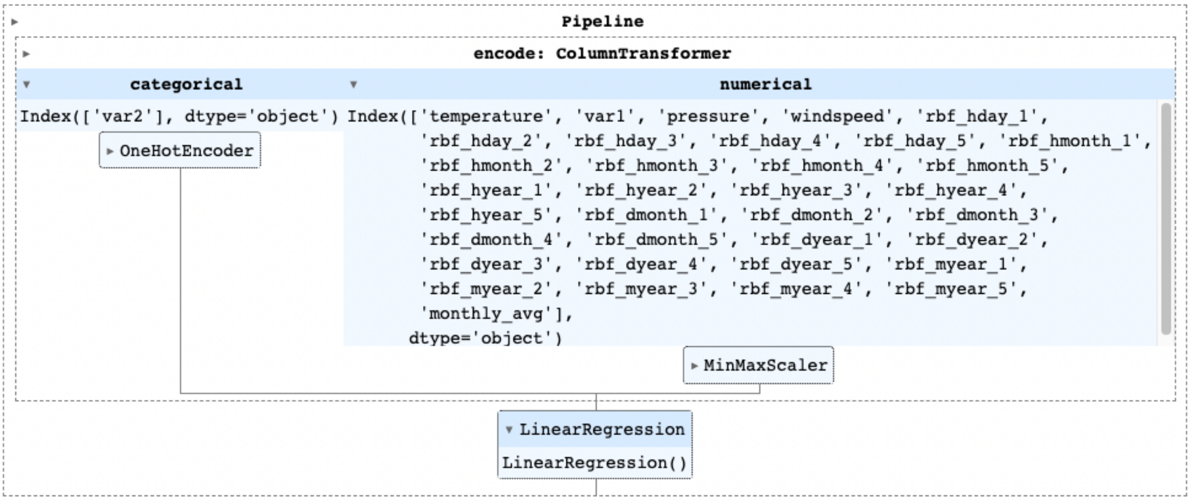


Figure 1. Shows the distribution of dependent variable

Furthermore, several feature engineering methods are applied on the independent variables. Firstly, we extract cyclical features from the "datetime" column. As the "datetime" is up until hour, hence we extract the *hour of day / month / year, day of month / year and month of year* from the "datetime" column. We apply *RepeatingBasisFunction* on the features extracted earlier. It is an radial basis function which encodes the input with commonly the Euclidean distance of some fixed point, such that the circular nature of January 2017 and December 2016 is recorded. In addition, the monthly average of energy consumption is added as another feature.

Scikit Learn Pipeline is used to construct the models for easier training, testing and predicting. For *LinearRegression,* We use *OneHotEncoder* and *MinMaxScaler* to encode the categorical feature and scale the numerical features respectively. OneHotEncoder is used as the categorical column does not have ordering.



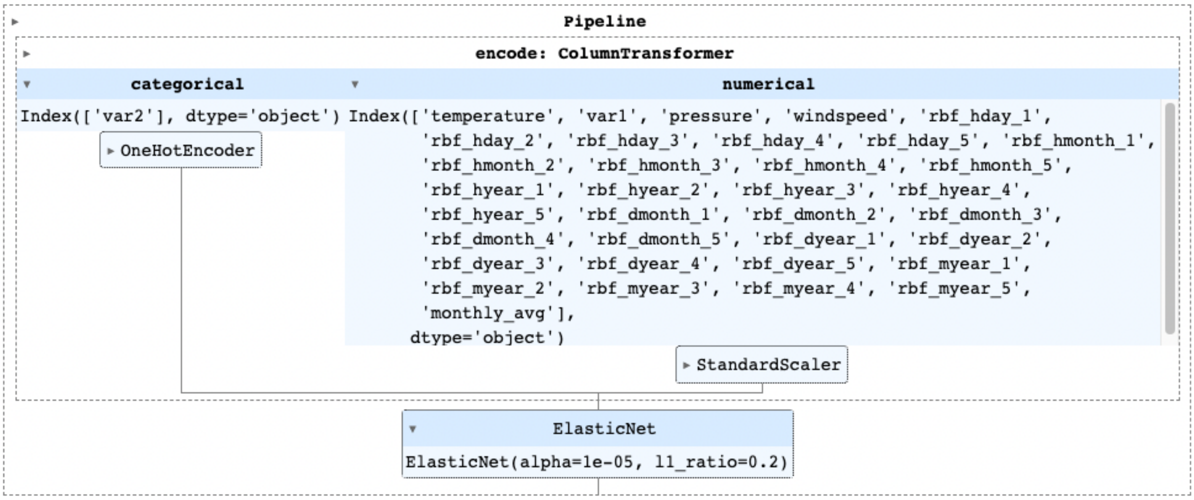
```
Index(['var2'], dtype='object')
```

```
Index(['temperature', 'var1', 'pressure', 'windspeed', 'rbf_hday_1',
       'rbf_hday_2', 'rbf_hday_3', 'rbf_hday_4', 'rbf_hday_5', 'rbf_hmonth_1',
       'rbf_hmonth_2', 'rbf_hmonth_3', 'rbf_hmonth_4', 'rbf_hmonth_5',
       'rbf_hyear_1', 'rbf_hyear_2', 'rbf_hyear_3', 'rbf_hyear_4',
       'rbf_hyear_5', 'rbf_dmonth_1', 'rbf_dmonth_2', 'rbf_dmonth_3',
       'rbf_dmonth_4', 'rbf_dmonth_5', 'rbf_dyear_1', 'rbf_dyear_2',
       'rbf_dyear_3', 'rbf_dyear_4', 'rbf_dyear_5', 'rbf_myear_1',
       'rbf_myear_2', 'rbf_myear_3', 'rbf_myear_4', 'rbf_myear_5',
       'monthly_avg'],
      dtype='object')
```

```
LinearRegression()
```

Figure 2. Structure of *LinearRegression*

For *ElasticNet,* We use *OneHotEncoder* and *StandardScaler* to encode the categorical feature and scale the numerical features respectively. *StandardScaler* is used as the L1 and L2 regularizers are assuming all features to be centered around 0. However *ElasticNet* has some hyperparameters that are available for tuning, hence GridSearchCV is applied to train the model with a set of defined range on the "alpha" and "l1_ratio". The range for "alpha" is [1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02] while "l1_ratio" has range defined as [0, 0.2, 0.4, 0.6, 0.8, 1]. Based on the search, we are able to get the best performed hyperparameters setting of **{'alpha': 1e-05, 'l1_ratio': 0.2}** with best R^2 score of 0.4529. In short, we have 0.00001 as constant that multiplies the penalty terms with 0.2 for L1 penalty and 0.8 for L2 penalty.

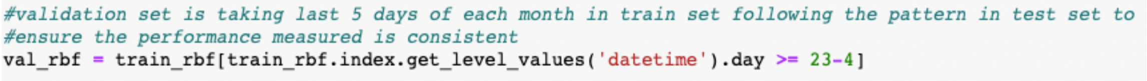


```
Pipeline
encode: ColumnTransformer
categorical                                numerical
Index(['var2'], dtype='object') Index(['temperature', 'var1', 'pressure', 'windspeed', 'rbf_hday_1',
                                        'rbf_hday_2', 'rbf_hday_3', 'rbf_hday_4', 'rbf_hday_5', 'rbf_hmonth_1',
                                        'rbf_hmonth_2', 'rbf_hmonth_3', 'rbf_hmonth_4', 'rbf_hmonth_5',
                                        'rbf_hyear_1', 'rbf_hyear_2', 'rbf_hyear_3', 'rbf_hyear_4',
                                        'rbf_hyear_5', 'rbf_dmonth_1', 'rbf_dmonth_2', 'rbf_dmonth_3',
                                        'rbf_dmonth_4', 'rbf_dmonth_5', 'rbf_dyear_1', 'rbf_dyear_2',
                                        'rbf_dyear_3', 'rbf_dyear_4', 'rbf_dyear_5', 'rbf_myear_1',
                                        'rbf_myear_2', 'rbf_myear_3', 'rbf_myear_4', 'rbf_myear_5',
                                        'monthly_avg'],
                                       dtype='object')
OneHotEncoder                                                StandardScaler
ElasticNet
ElasticNet(alpha=1e-05, l1_ratio=0.2)
```

Figure 3. Structure of *ElasticNet*

Question 3

The given dataset is separated into two parts, train and test. However, the test set consists only unseen data, thus it is necessary to extract a portion of data from the training set to validate the performance of the built models. To simulate the scenario for unseen test set, we extract five days at the end of each month as our validation set. The days included are from 19 until 23. The target column is extracted from features to avoid sending them as an input features when training the model.



```
#validation set is taking last 5 days of each month in train set following the pattern in test set to
#ensure the performance measured is consistent
val_rbf = train_rbf[train_rbf.index.get_level_values('datetime').day >= 23-4]
```

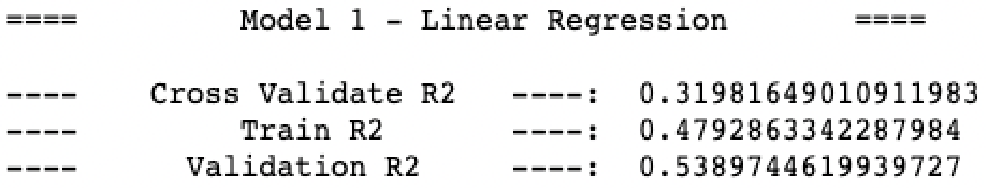
Figure 4. Train-Validation split

We use the validation set to measure the performance of the trained models. If it achieved a lower score than the train set, we could conclude that the model is overfitting while the low score on the train set is indicating that the model is underfitting.

Cross validation is applicable in this task but it is different from the usual split. We apply time series cross validation instead. Moreover, no shuffling is applied as the samples are split in sequence of the time and the successive training sets are supersets of those that come before them. The test samples are always from a further timeline after the train samples.

Question 4

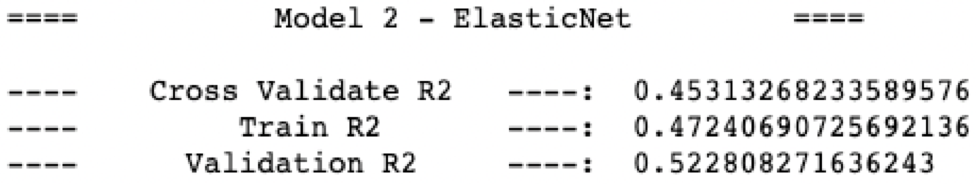
Both the regression models are evaluated with R^2 metric. It is the coefficient of determination which is to measure the proportion of variance of a dependent variable from the independent variables by finding the best fit line.



```
====          Model 1 - Linear Regression        ====

----     Cross Validate R2    ----:  0.31981649010911983
----          Train R2        ----:  0.4792863342287984
----       Validation R2      ----:  0.5389744619939727
```

Figure 5. *LinearRegression* R^2 performance



```
====             Model 2 - ElasticNet          ====

----     Cross Validate R2    ----:  0.45313268233589576
----          Train R2        ----:  0.47240690725692136
----       Validation R2      ----:  0.522808271636243
```

Figure 6. *ElasticNet* R^2 performance

From Figure 5 and 6, we are able to see that the performance of both the regression models are identical. Both models did not overfit and achieved an R^2 of about 0.48 in training and about 0.53 at validation. The reason might be both of them are linear models, although *ElasticNet* applied a penalty on both Lasso and Ridge regularizers, it does not help us to enhance the performance to be better than the ordinary least square.

Appendix:

1. Brownlee, J. (2020, June 11). *How to develop elastic net regression models in Python*. Machine Learning Mastery. Retrieved November 13, 2022, from https://machinelearningmastery.com/elastic-net-regression-in-python/
2. Sharma, G. (2022, July 21). *Regression algorithms: 5 regression algorithms you should know*. Analytics Vidhya. Retrieved November 13, 2022, from https://www.analyticsvidhya.com/blog/2021/05/5-regression-algorithms-you-should-know-introductory-guide/#:~:text=Linear%20Regression%20is%20an%20ML,the%20other%20given%20independent%20variables.