Using weather data and sklearn packages to predict energy price for next hour

Chia Li
Electrical and Computer Engineering
University of Massachusetts Amherst
Amherst, U.S.
chiali@umass.edu

Jianyun Miao
Electrical and Computer Engineering
University of Massachusetts Amherst
Amherst, U.S.
jmiao@umass.edu

Chieh Min Chung
Computer Science
University of Massachusetts Amherst
Amherst, U.S.
chiehminchun@umass.edu

ABSTRACT

LMP price is influenced by numerous factors including weather conditions, demand, and the environment. The weather would be a direct factor in humans making decisions on their behavior. For example, a sunny day would increase the desire to go outside. In contrast, severe weather like heavy snowing in winter causes people to stay at home and set higher temperatures on heating.

With this observation, we'd like to predict the price of day-Ahead LMP price with weather data. Our contributions are (a) gathering 1-year weather data and day-Ahead LMP prices from websites. (b) compare four different regressors provided by sklearn packages on Google colab. (c) explore the impact of each weather feature.

KEYWORDS

Day-ahead LMP price, price prediction, Machine learning, sklearn, weather data,

1 Introduction

This work presents an approach based on machine learning to determine the degree to which these weather features influence the components that make up the locational marginal price (LMP) of electricity. LMP price forecasting is very valuable in competitive electricity markets because participants rely on it for generation, scheduling assets, and formulating effective bidding plans.[1][2] Linear regressors are widely used on price prediction tasks. Many available resources on the network provide us with materials for preparing our data and preprocessing it. There are also unsupervised learning methods, e.g., Long short-term memory (LSTM) neural network, despite the supervised learning. We did not include those unsupervised learning in this semester's project.

We divide our project into two parts:
(a) the same data frame with four different regression methods, which we are interested in the performance with different methods (MSE, RMSE, and MAPE)
(b) we explore the characteristics of each feature, and we add the features sequentially and train with Ridge CV. We expect the impact on MAPE and RMSE in the second part would help us in explaining its characteristics.
We like to have more discussion on explaining the impact of each feature, not just send the data Frame into the function and give an arbitrary number to grade the model we built.

2 Background

We start with gathering data from an online website with day-Ahead LMP prices from ISO-new England and weather data from wunderground.

2.1 Weather data

There are many choices. For example, we tried to access the data from the airport, they have their observations uploaded online. However, the format is not friendly for us since we manually copy-paste it to the Google sheets. We also find some Government departments are also collecting the historic weather data. However, the formats are not friendly to us on this project to work with. In the end, we find wunderground, recording hourly weather data. We expect to have more advanced approaches to spider the data from this website.

2.2 Day-Ahead LMP price

ISO-New England collects their historic LMP price (we only use this price in our project as Ground-Truth) and releases it on their website. However, there are thousands of nodes with hourly data. We calculate the mean price of each hour. Next problem is that a regular user can only download 15-day data for each request. This is inefficient in gathering data for us, too. Lastly, we also need to develop another approach to extract the mean price data without iterating each day from downloaded files.

2.3 Pre-processing

Sklean provides many libraries for preprocessing the data frame, and pandas also have many functions that help developers to process their data frame for training. For our project, we simply insert the mean value for the numeric numbers cell and the most frequent string for string type cell. However, as David Irwin mentioned in our presentation that it might not be a reasonable way for some type of missing data to insert the mean value. For example, the average highest temperature in Massachusetts is 23℉. However, if you insert this number in missing cells in summer. This might cause an exception for training and cause the model to be unreliable in the end. For this specific question, we have not Implemented any update on it. In version two, we insert the average temperature for missing cells.

Figure 1 shows the correlation with thirteen features, the inverted order is how we explore in the second part: features importance exploration.

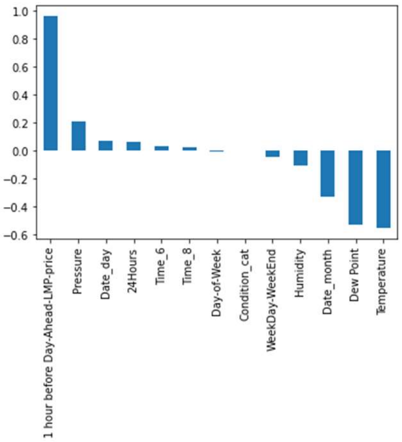

Figure 1: A bar chart of the features' correlations.

2.4 Evaluation Matrices

We use provided matrices by Sklearn: root mean square (RMSE) and mean absolute percentage error (MAPE) for evaluating our model to understand the performance of each regression method and the characteristic of each feature. Second, for our project, it is important to stress that this project is a time-series project. That is, splitting the training data set and testing data set can't be randomized by using the split_train_test () function provided by sklearn. Third, to minimize the external factors in our project. We disregard the data from Jan. 2022, Feb.2022, and Mar.2022. That is because the conflict in Eastern Europe might affect the energy market. In the result part, we will demonstrate the impact of external factors on our model.

In the result part, we find out that after adding specific features into our training we can improve the performance of the model. We will try to explain it in the result part.

2.5 Difference in the training function

Sklearn provide numerous methods for regression tasks, each method has its own mathematical equation, and this led to different result compared to each other.

2.5.1 linear regression This estimation procedure is simple and, most importantly, we consider this model as our persistent model in which computation time is controllable in our understanding.

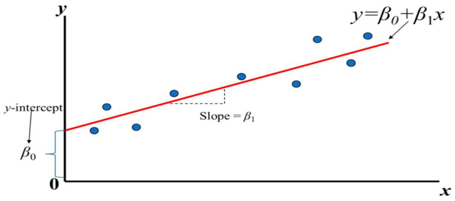

Figure 2: Concept of Linear regression

2.5.2 SVR Support Vector regression is a type of Support vector machine that supports linear and non-linear regression. As it seems in the below graph, the mission is to fit as many instances as possible between the lines while limiting the marginal violations. In general, SVR tries to minimize the error rate.

The advantage of SVR is that the decision model can be easily updated. Besides, it has excellent generalization capability, with high prediction accuracy.

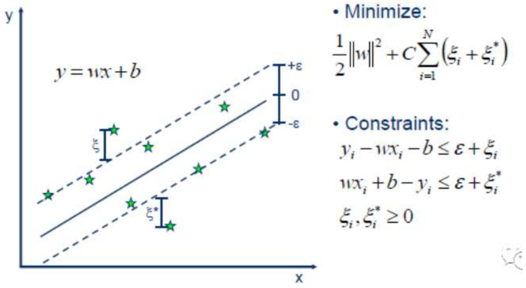
y
$y = wx + b$
+ε
0
-ε
ξ
ξ*
x
• Minimize:
$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$
• Constraints:
$y_i - wx_i - b \le \varepsilon + \xi_i$
$wx_i + b - y_i \le \varepsilon + \xi_i^*$
$\xi_i, \xi_i^* \ge 0$

Figure 3: Concept of SVR

2.5.3 Ridge CV If we want to find the important parameters in the model, we can use the Ridge model to identify which parameters are important, because the regression coefficients of unimportant parameters will approach 0 in the model, but because they will not really disappear, we can use the regression model.

$$minimize\left\{SSE + \lambda \sum_{j=1}^{p} \quad \beta_j^2\right\} \tag{1}$$

2.5.4 Lasso In terms of functionality, when the user's model has too many parameters and wants to automatically remove unimportant variables, the Lasso model should be selected

$$minimize\left\{SSE + \lambda \sum_{j=1}^{p} \quad \left|\beta_j\right|\right\} \tag{2}$$

3 Method

We use sklearn, pandas, and other Python packages on Google colab, which all wildly use libraries and platforms without configuring your own environment on the desktop.

3.1 Data preprocessing

We load the data as a data frame, a data type provided by pandas that can perform row/column operations on it. We first examine the missing data in each feature, as we discussed in the last part, we fill all missing data with the most frequent string for string type cell and the mean value for the numerical cell. The missing data rows are roughly 230 rows out of 8921 rows in our data frame.

We consider the mean value approach would not impact the accuracy in this level of the data frame. Before calling the regression functions and training with the data, we drop some irrelevant features(columns) from the original data frame. For example, the corr() function shows that the humidity's correlation is irrelevant to the LMP price. We drop some features that have lower correlation or are conceptually irrelevant to the LMP price.

Despite the irrelevant columns, there are also some features that are extracted or transformed into other forms. For example, after some tests, we discarded the "AM/PM". Instead, we have time groups every six hours a group and every eight hours a group. After such a transformation, the original columns of "TIME" and "AM/PM" have no use for us.

We also notice that the training time increased due to the added feature "1-hour previous LMP price". The float type feature increases the computation time on every regression task. One way we are considering is using the encoder method to scale the price between zero and one, which might reduce the computation time.

In our project, there are many features that need to be categorized again to have a small set of labels for clear training. One specific column is" Condition" and its paired column "Condition_cat". The data type sent into training is an integer type, whose original type is a string type describing the weather conditions. In our first published version of the code, we simply turn the type into a category and encode it with the built-in function into the integer type for the next step of training. We assume that the recording string is from a limited set, which means the description of the weather should repeat. However, we did not check the entire column. This might cause noise to the model since this feature will be the feature affecting the performance of our result. We are considering having other feature extraction and printing the distribution in the future for better analysis.

3.2 Splitting training set and testing set

As we mentioned above, the conflict in Europe causes an external impact on the international market in which the price and the weather data during that time are disregarded temporarily.

Approximately, we use 75% data (8921(hours) - 90(days)*24(hours)) for both explorations. Since the project is time series related, that means cannot randomize the set for training and testing. We split the first 70% rows(hours)for training and the rest 30% rows(hours) for testing.

3.3 Impact of functions

Sklearn provides numerous functions for linear regression tasks. For example, in our project, we use Ridge CV, linear regressor, SVR (with kernel trick), and Lasso CV. The reason for different algorithms is that we want to understand more than the mathematical definition of each function. As shown in Table 1, even though they all can fulfill the linear regression tasks, the results of prediction are different after all. Although we mentioned the importance of time series in our project, the cross validation in Ridge and Lasso folds the data set into many sets. Instead of the cross validation provided by the library, we should try the time series cross validation trick in the next version of publishing. Straight forward, the regressor with cross validation is only slower than other regressors on Ridge CV since the Lasso CV drops some parameters which reduce the training time for it. The longer and slower here is a relative concept. The fastest training is done in a few seconds and the slowest is roughly twenty seconds.

	Linear regressor	SVR	Lasso CV	Ridge CV
	Testing	Testing	Testing	Testing
MSE	71.5096	385.8708	74.2069	72.1047
RMSE	8.4563	19.6436	8.6143	8.4914
MAPE	0.0891	0.1641	0.0877	0.0890

Table 1 Numerical comparison of the different models for prediction of LMP

3.4 Impact of features

Since we only have 12 features, we add the features sequentially to see the impact on our weather-price model. We want to show the cross impact of features on our model, and we found out that the seventh feature "Month" (feature name: "Date_Month"). During the presentation and poster, we originally thought that it is the weather condition that is affecting the accuracy of the price prediction. After some updates on our not-friendly code, we find out that we lost the first result, and this causes the wrong conclusion.

Our new conclusion is that: the reason for the month matter is because the demand for heating in buildings is correlated with weather and the price, which also fits the assumption we made at the beginning.

Despite the weather features in the original data frame, the added feature "1-hour previous price" increase the accuracy dramatically and increase the computation time trade off at the same time. We presume that the time series property act like short term memory which provide some hint on external factors on the energy price.

Last, to improve the accuracy in this part, we think having more data (3-year data) will help. One way to think is the repeated days and hours are highly correlated to the demand that the people are using. Next, since we disregard the data in 2022(Jan., Feb., Mar.), there are no repeated days for us, even if we include those data, the high variant price might cause a bad model.

4 Result

4.1 First part: function difference of exploration

We demonstrate the four methods we used. As mentioned, the Lasso CV has the worst case with high RMSE and MAPE on both the training set and test set. However, the whole process time is fast as well as the linear regression. In some scales of data, Lasso CV could be a persistent model to propose in first training and improve the accuracy with other methods that need more training time.

On the weather feature only model, we notice the very last segments of prediction are highly error compared with other time segments. We think this is caused by the scale of the data set since most of the training data are some winter data and most of the time is relatively warmer days during 2021. We predict the price with the last 30% of data from the 75% of original data, and the last 30% of data is relatively unstable compared to the training data.

For this problem, we solve it by adding a new feature "1-hour previous price" to our original data frame. The result is perfectly following the actual price in the final fix.

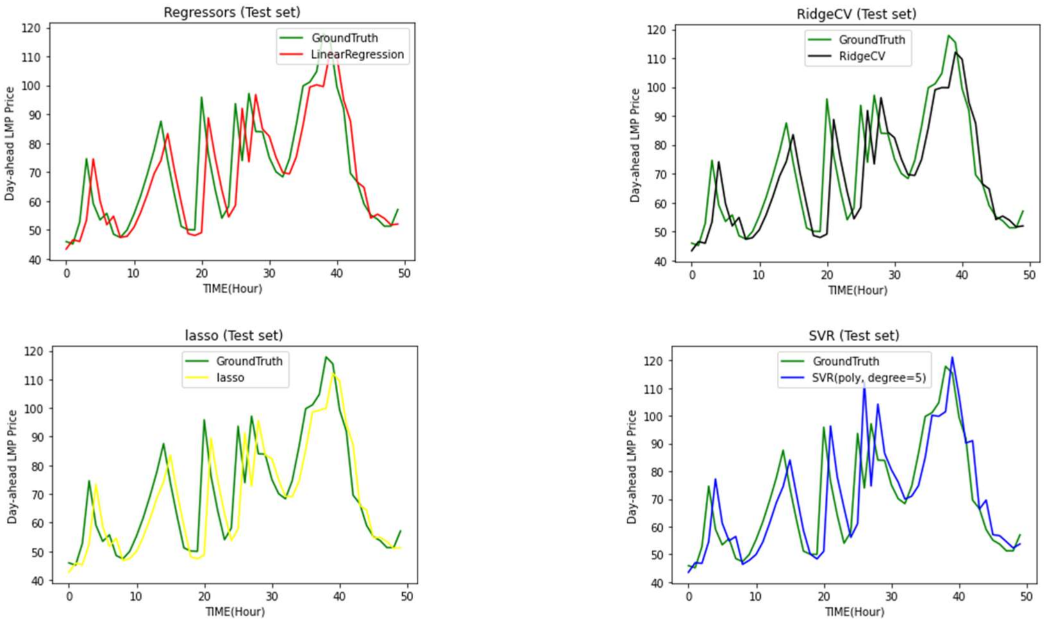

Figure 4: Predicted price versus ground truth; Evaluated mode: Ridge CV, Linear Regression, Lasso CV, SVR (ploy, deg=5)

4.2 Second part features the importance of exploration

We demonstrate twelve features' impact (Figure 5) on the weather feature-only model and give the reason why we think the month is the reason that impacts our model and improves the performance. If we have a longer period, the role of time (group of 8 and group of 6 ...etc.) will be more significant than the model we have right now.

The accuracy with twelve features is bad. After we have the thirteenth feature "1-hour previous price", the result is shown in Figure 4 above and is much better than we have in Figure 6.

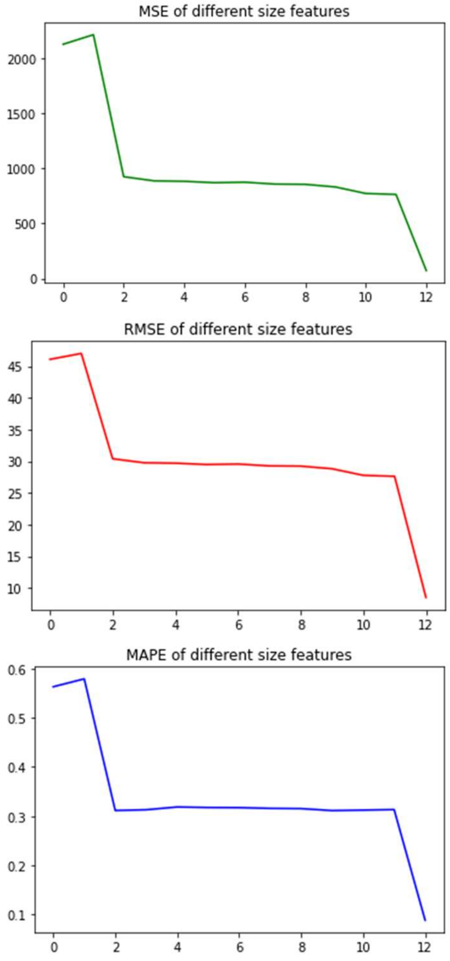

Figure 5. Impact of the sixth feature on weather feature only

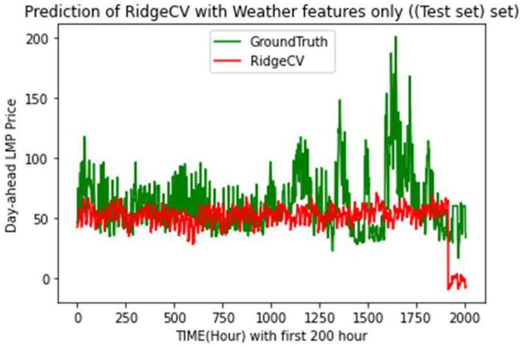

Figure 6. Prediction versus ground truth on weather only model, the MAPE is ~0.3135 compared to the model with "1-hour previous price" is ~0.0890. Also, the thirteen features provide a better adaptation to external factors.

5 Future work

5.1 Scale of data

To minimize the error in section 5.1, the solution we came up with is to include a longer period in our data. To fulfill this requirement, we need to develop an efficient way to gather the data from the target website (for the first publication, our data comes from the wunderground).

5.2 Quality of data

One mentioned problem is that the weather conditions' encoding process is not good enough. If we label the condition with fewer labels on that feature, the performance of the model might be improved. In the last section, we mentioned the new method for our data can also improve the quality of data. That is because the method we have right now is collecting the data manually which took much time to correct and sort the integrity of data.

5.3 External factor

Although some research says that the conflict in Eastern Europe is not the main reason why the Oil price goes up, then the energy price goes up. However, we still can take this factor to affect some portion of the price. We are looking forward to examining the external factors, e.g., change in the dollar, inflation in the US, global oil, natural gas, and coal price. With this information, we think our model can reflect and adapt the price with higher accuracy.

As mentioned, the time series property act like short-term memory on this project. However, what if we have more previous hourly average prices to act like more layers of memory on our project. We are looking forward to having more grouping features of the previous price like we have on-time features.

6 Conclusion

We demonstrate the potential of predicting LMP prices with weather data. The weather forecast is more stable and less error-prone with advanced technology. Our contribution to this paper is in two parts: (a) explore the difference of numerous linear regression functions provided by sklearn (we examine the linear regression, SVR with kernel poly and degree as five, Ridge CV, and Lasso CV with the same data frame). (b) Next, we explore the potential of each processed feature, and we give our opinion on why this feature can reduce the MAPE and RMSE in our model.

Before we include the "1-hour previous price" to our data frame, we consider having more columns for external columns. However, in the end, the time series property on the thirteenth feature can reflect some level of external factors' impact.

In future work, we list the works that would improve our model performance and accuracy.

REFERENCES

[1] Wunder ground website: https://www.wunderground.com/

[2] Price website of ISO-New England: https://www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/lmp-by-node

[3] Sklearn entry website: https://scikit-learn.org/stable/index.html

[4] Ridge CV sklearn website: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html

[5] Lasso CV sklearn website: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html

[6] Linear regression sklearn website: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[7] SVR sklearn website: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[8] Our project on GitHub: https://github.com/BOTnreLI/LinearRegressor_withsklearn