Delaware Hackathon: Methodology Report

Group 8: Patrick Dundon, Ajay
Parihar, Manjunagaraj Rudrappa, Shivam Sarin

May 7th, 2019

1. Our goal

Our mission is to prove that developing a predictive model using external data can ultimately improve trading perofrmance. By analyzing the French electricity market as well as external variables such as weather patterns, we want to be able to build an accurate model to predict the change in price difference between the IntraDay and DayAhead markets to know what action to take on a daily basis (buy DayAhead or buy IntraDay).

As we are predicting future price difference (increase or decrease), this is therefore a classification problem. The bulk of the data processing and modelling to be discussed in this report was undertaken in R programming language.

2. Our data (collecting, processing and basetable)

In review of our previous exploratory work and reports, we will take an extensive look at the data involved in our hackathon project as far as how we gathered it, the steps we took in data processing, as well as the final basetable that was developed.

The data

The bulk of our data comes from both the EPEXSPOT French electricity market website as well as various weather metrics from Paris, France, which of course compose many of the main variables affecting electricity consumption. In the end, we focused on and collected data for a 4-year period (beginning January 2015 to end December 2018).

The processing

By focusing on a 4-year period for hourly data (24 observations per day), we knew immediately what size of data we would be dealing with. The nature of our data (prices, dates, times, temperature) meant we had to make sure all observations had values that made sense. The option to download electricity market data from the EPXSPOT website gave us a solid base of clean data to start with. From there, we gathered relevant weather data (temperature patterns, precipitation) and aggregated them to the financial data once the number of observations matched.

From there we focused on building additional variables from the existing data such as predicted temperature for each specific hour of the day, temperature change day over day, accounting for holidays and weekends, and of course, our target variable. The target variable was taken as 1 if the PriceDIFF variable value (price difference between IntraDay and DayAhead) was greater than one standard deviation of Price DIFF, otherwise it was taken as 0. Additionally, we made sure to keep consistency throughout the data by following the same %Y-%m-%d format and changing all the temperature-related variables to Celsius. Some duplicate values from date and time were also identified which were dealt with by selecting only distinct values.

From this point, our data processing shifted towards dealing with outliers and missing values. For temperature-related variables, we did not consider any of the data to be outliers worth changing, as weather is often unpredictable in nature and can vary quite highly over the course of a certain time period. We did however come across several missing data values for the following variables, which were dealt with in the following way:

- precipitation, temperature (minimum, maximum, average, variance, predicted): replaced with the mean value for that variable
- Temperature difference (day over day): replaced with 0 for simplicity, as it was very close to the mean of 0.005594

The basetable

Our basetable is over 35,000 observations in length (one observation per day from January 6th, 2014 until December 31st, 2018). It contains several variables including those concerning financial electricity market metrics, weather patterns, key calendar

dates, energy generation and energy consumption. This basetable was explained in more detail in our basetable report. Since the report, we have added variables for lag (such as 24, 48, 72 and 96 hour lag variables for DayAhead and IntraDay markets as well as priceDIFF). Below is a partial snapshot of the basetable:

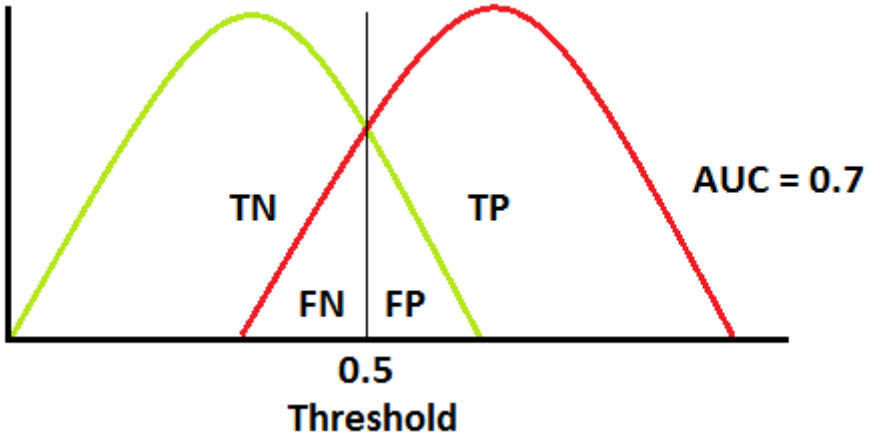
fromdate	fromtime	IntradayPrice	PRCP	TMAX	TMIN	TVAR	TAVG	TDIFF	Significant	TPRED
2015-01-07	0	40.10	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00
2015-01-07	1	30.00	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00
2015-01-07	2	65.00	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00
2015-01-07	3	40.00	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00
2015-01-07	4	40.00	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00
2015-01-07	5	35.25	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00
2015-01-07	6	52.00	0.00	8.33	0.00	8.33	2.78	1.67	0	0.00

3. Methods

Modelling

We are planning to create two models based on standard deviation of price variation between intraday and day ahead to minimize risk:

Model 1= std(IntraDay-DayAhead)
Model 2 = std(DayAhead-Intraday)



To this point, we have tested three main models:

1. **Logistic regression**

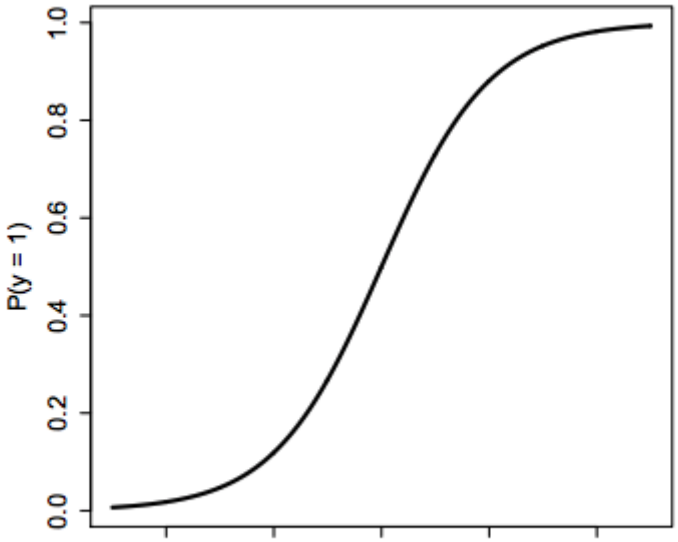
Target variable: We calculate the target variable on Intraday and Dayahead:

```
basetable1$PriceDIFF>sd(basetable1$PriceDIFF)+mean(basetable1$PriceDIFF),1,0
```

Description: As the name already indicates, logistic regression is a regression analysis technique. Regression analysis is a set of statistical processes that we can use to estimate the relationships among variables. More specifically, we use this set of techniques to model and analyze the relationship between a dependent variable and one or more independent variables. Regression analysis helps you to understand how the typical value of the dependent variable changes when one of the independent variables is adjusted and others are held fixed.

Logistic regression is an instance of classification technique that you can use to predict a qualitative response. More specifically, logistic regression models the probability that gender belongs to a category.

That means that, if you are trying to do gender classification, where the response gender falls into one of the two categories, male or female.



$$\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$$

$$P(y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k)}}$$

Why we chose it: It is the easiest model to implement and acts as a baseline model to test all the future models.

2. **XGBoost (binary:logistic)**

Description: As the boosting ensemble methodology combines weak classifiers and turns them into strong classifiers, we plan to use this algorithm to improve our call predictions. The type of boosting, we will use, is XGBoost, namely, implementation of gradient boosted decision trees. By weighing the model outcomes based on the previous results, we will get better response probabilities for each time slot. XGBoost is ideal because it tends to perform fast and well on large datasets.

Also, it has a wide variety of tuning parameters for cross-validation and regularization to optimize the results. However, it may not be easy to understand predictions, and there is a potential possibility of overfitting; a small change in the feature set can create radical changes in the model. Building more generalized trees using both learning rate and depth of tree may be necessary.

Why we chose it: XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost stands for eXtreme Gradient Boosting. The library is laser focused on computational speed and model performance, as such there are few frills. Nevertheless, it does offer a number of advanced features.

3. **Random Forest**

Description: In addition to two logit-based models, random forest model will be used for classification where we get the time slot for each customer as a class output. In

order to gain more insights, we will also investigate probabilities for each class (the total number of trees voting for a specific time slot) and the default threshold. By adjusting the default threshold, we are going to change our focus from majority voting to weight voting.

As our success rate for each customer is not high and the data is highly imbalanced, changing cut off value from 0.5 (majority voting) to 0.3 and giving more weight to 0.3 than 0.2 will lead to more accurate outcome. *Cut off point will be decided when the sensitivity and specificity are at their maximum.

Since this model takes the outcome of each decision tree into consideration, the overall performance of ensemble decision tree methods tends to be better than a simple decision tree, and it is fast to train. Besides, the random forest algorithm can effectively capture non-linear relationships with the target variable, avoids overfitting and deals with multicollinearity problems. However, due to the nature of this model, the model can still get very large and complex, resulting in difficulties to interpret the predictive variables.

Why we chose it: Random forest algorithm can be used for both classifications and regression task, thus, it provides higher accuracy. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data. If there are more trees, it won't allow overfitting trees in the model. It has the power to handle a large data set with higher dimensionality

Parameters

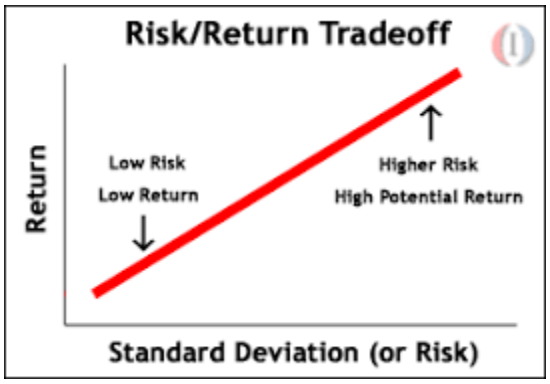
In order to find the best parameters to train our different models, we will use a grid search cross validation method allowing to find the optimal parameters for our models and prevent over-fitting.

The grid search algorithm is the traditional way of performing hyper parameter optimization. It is an exhaustive searching through a manually specified subset of the hyper parameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, in our case: a cross-validation.

We will divide the data into training, validation and testing set. We first train the model on the training set and test it on the validation set. After we test all the models, we find the best model and re-run the model on the final training set comprising of training and validation set. We finally test it on the final test set.

Risk Management

In the financial world, risk management is the process of identification, analysis and acceptance or mitigation of uncertainty in investment decisions. Essentially, risk management occurs when an investor or fund manager analyzes and attempts to quantify the potential for losses in an investment and then takes the appropriate action (or inaction) given his investment objectives and risk tolerance.



Based on a (hopefully large) number of individuals for which the score and condition is known, researchers may use ROC curve analysis to determine the ability of the score to classify or predict the condition. The analysis may also be used to determine the optimal cutoff value (optimal decision threshold). For a given cutoff value, a positive or negative diagnosis is made for each unit by comparing the measurement to the cutoff value. If the measurement is less (or greater, as the case may be) than the cutoff, the predicted condition is negative. Otherwise, the predicted condition is positive.

We will use sensitivity and F1 score for trading and to minimize the risk.

Classification Table

		Predicted Condition: Positive	Predicted Condition: Negative
True Condition	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Long-Term Gain

For our trading strategy we will compare monthly gain with vs without model minus trading cost:

Gain = Gain(with_model)-Gain(without_model) - trading fees

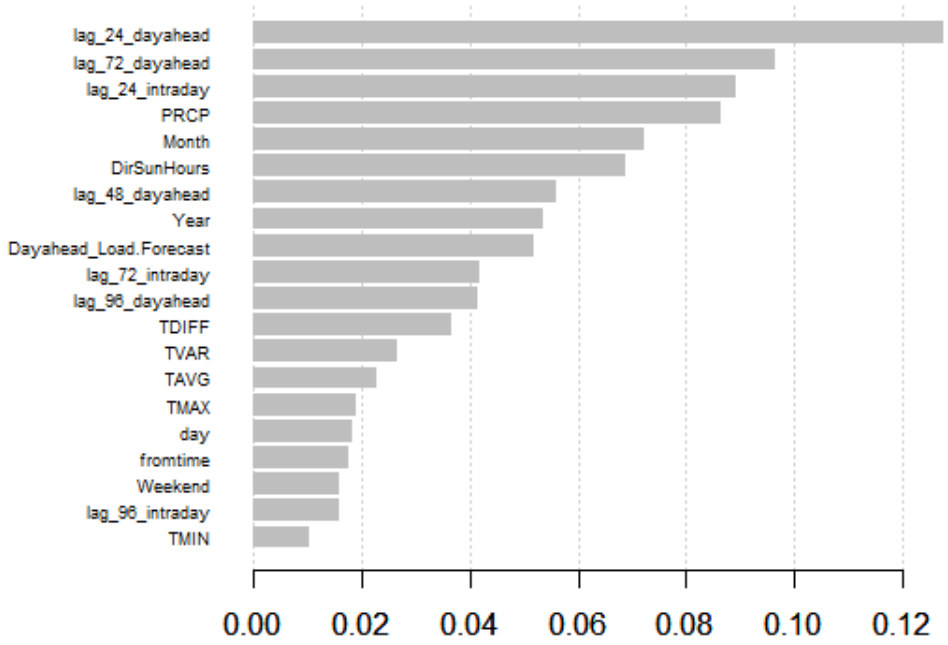
4. Experimental setup

From our base table of 35,000+ hourly observations, we decided to follow a training/validation/testing split of 75%/12.5%/12.5% for the modelling process. For our observed 4-year period that therefore equates to a 3-year train set size, 6-month validation set size and 6-month test set size. All the variables in the basetable were used still meaning none were removed or discounted in the process of creating the training, validation and test set splits.

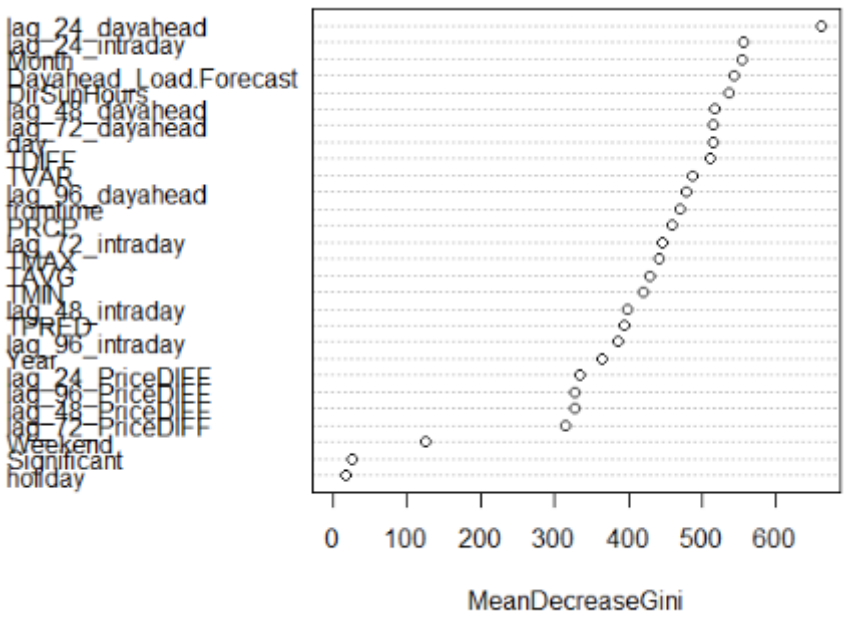
As we consider this to be a classification problem, the chosen metric to evaluate performance of the several models tested was ROC/AUC score. This is one of the most common machine learning performance evaluation metrics and thus we felt the most beneficial way to determining the best model among several. We also use F1 scores to quantify the accuracy of our binary classification models. To balance the data, we made use of oversampling and SMOTE sampling method (depending on the model).

5. Intermediate results

Important Variables from XG-Boost with SMOTE sampling:



Important Variables from Random Forest with SMOTE sampling:



From the above Results, we can see that lag_24_dayahead and lag_24_intraday are the most important variables.

Evaluation Results:

Models	Thresholds for Target variable	Sampling methods	AUC	F1 Score	Accuracy
Logistic Regression	Price Diff > Standard Deviation(Price Diff) = 13.201	Over Sampling	0.5238	0.5906	0.5694
XG Boost	Price Diff > Standard Deviation(Price Diff) = 13.201	Over Sampling	0.5257	0.1534	0.6051
Logistic Regression	Price Diff > Standard Deviation(Price Diff) = 13.201	SMOTE	**0.5373**	**0.3653**	0.7537
XG Boost	Price Diff > Standard Deviation(Price Diff) = 13.201	SMOTE	0.5355	0.1525	0.8696
Random Forest	Price Diff > Standard Deviation(Price Diff) = 13.201	SMOTE	0.5337	0.1366	0.8969

Logistic Regression with SMOTE sampling gives best Results to date.

6. Looking further

Limitations

One of the main limitations we faced over the course of our project was indeed the data sourcing process. While there are several factors that could theoretically have an impact on electricity price, we found that many of them either did not have reliable data online or would come at a considerable cost (for example, in-depth historical wind data costing in excess of €900). This of course has an impact on the potential scope of how well our final model results represent a truthful outcome.

Also, simply considering the volatile nature of a financial market, another limitation we come across is not being able to account for unforeseen fluctuations based on economic conditions. We must make assumptions that external factors for which we did not or could not quantify as variables will remain consistent with how they have been in the past 4 years (beginning 2015- end 2018).

Further improvements

While we still have time and therefore will continue to search for new features to implement, we agree that our prediction process could be improved with more feature engineering to test a larger variety of variables on our models and evaluate if that improves performance.

We felt in the brainstorming phase of our project that a 4-5-year observational period would be enough. However, as there exists a longer pool of historical data on the EPEXSPOT website, we perhaps could have considered a 5-10-year observational period to take into consideration more data for our modelling process.

Based on current modelling results, as we still trying to improve performance, it could also be useful to explore some additional models. However, in the interest of time, we prefer to focus on improving the current ones explained above.

Next steps

Once satisfied with final model results, we will begin to apply what we have developed in a business context to show how the model can help decision making in the electricity trading market. We plan to have two approaches (one model where priceDIFF is IntraDay – DayAhead and one where priceDIFF is DayAhead – Intraday).

Being able to reasonably predict accurate targets (again, target = 1 if priceDIFF > 1 standard deviation of priceDIFF) will enable us to decide when to act, either by buying IntraDay or busying DayAhead. Of course, other factors such as trading fees will need to be considered when computing potential Profit/Loss values for using this model.

7. Data sources

http://www.epexspot.com/en/
http://www.meteofrance.com/climat/france
https://www.ncdc.noaa.gov/cdo-web/
https://www.investopedia.com/terms/r/riskmanagement.asp
http://manishbarnwal.com/blog/2017/05/18/choosing_probability_cut-off_in_classification/