Contents

I. Definition

Project Overview

The project for this capstone is to predict the electricity consumption in France, to do that RTE the organism that is managing the French electricity network provides an API to access different datasets related to the network. For the study, the data related to the consumption will be used, these data are provided since January 2008 with a timestep of 30 minutes. For this project, the idea is to create a tool that can predict the daily and the half hourly consumption at a national scale, this tool can be very useful for the energy provider like EDF that has to manage the power sources in France.

To make this analysis, it's important to know that the electricity is the main energy vector used in France. The electricity consumption will be impacted by the weather condition, there will be a seasonal effect link to the use of the heating systems in the French households. Another aspect to think in the electricity consumption is that the needs will be different in function of the day of the week and the hour of the day.

To complete this project, the following datasets will be used:

- The consumption data of RTE[1]
- The weather data form 63 airport stations from Weather Underground[2]
- The GEOFLA dataset from IGN that provides information on the different cities in France[3]

The last dataset will be used to have a better idea on the repartition of the population in France.

The code related to this project can be found in the repository in the footer.

Problem Statement

The goal of this project is to create an algorithm that will forecast the electricity consumption in France at a daily scale and half hourly scale. To complete this project, I will divide the process to find the right algorithm in different tasks:

- **Daily consumption forecast**
    - Test different algorithms only based on electricity consumption and weather data from the centre of France (just based on the outdoor temperature and the daily consumption) and compare them to the benchmark model
    - Test the impact of the training set size on the model
    - Test the models with different inputs (more weather parameters and time information)
    - Test a second weather dataset based on a combination of all the weather data and the IGN data
- **Half hourly consumption forecast**
    - Try a similar approach than previously

With this find the right model, the different test models will be compared to a benchmark model and different metrics are going to be used to make the comparison.

[1] https://rte-opendata.opendatasoft.com/explore/dataset/cdc_conso/?disjunctive.qualite
[2] https://www.wunderground.com/
[3] http://professionnels.ign.fr/geofla

Metrics

This project can be classified in the regressor problem category so to assess the quality of the model, the following metrics will be used:

- The r^2 score
- The time of execution

II. Analysis

Data Exploration

About the data, for the RTE dataset the data are clean but there are two categories the definitive data and the intermediary data, for the project the data used will be the one that have the definitive status. This dataset represents 140254 points. In the following table there is a sample of the data provided by RTE.

	tstp	consumption	quality
0	2008-07-06 10:00:00	44486	Definitive
1	2008-07-06 11:00:00	46270	Definitive
2	2008-07-06 11:30:00	44801	Definitive
3	2008-07-06 13:00:00	41919	Definitive
4	2008-07-06 14:30:00	40135	Definitive

Table 1: Extract of the RTE data

For the weather data, there was some kind of issue in term of data quality, in the weather underground 73 stations seem to be useable for this project but the quality of the data was very various so it has been decided to choose the data that have enough points between the timestamp defined by the RTE data in the definitive status with at least one point per hour. In the case there was some missing points a linear regression has been done to fill the gap. To have a 30 minutes timestamp a resampling has been done. This dataset represents 140254 points. The IGN dataset, was a very clean dataset to use so now specific issue to use it. In the following table there is an extract of the weather from the centre of France (Clermont-Ferrand station) that will be used on the first part of the project.

	tstp	outdoor_temperature	outdoor_humidity	wind_speed
0	2006-01-01 00:00:00	6.000000	54.0	22.200000
1	2006-01-01 00:30:00	5.833333	58.5	19.116667
2	2006-01-01 01:00:00	5.666667	63.0	16.033333
3	2006-01-01 01:30:00	5.500000	67.5	12.950000
4	2006-01-01 02:00:00	5.333333	72.0	9.866667

Table 2: Extract of the weather data from the centre of France

To create the second weather dataset that is a combination of all the weather stations collected form weather underground, the following process has been used:

- Associated for each department (state) the weather station closest to the main city in the department
- Gave a weight for each department based on the ratio that the local population represent on the total population
- Made a linear combination of all the weather stations associated with their population weight

In the following table, there is an extract of the second weather dataset that represented the weighted weather dataset.

tstp	outdoor_humidity	outdoor_temperature	wind_speed
2007-06-06 14:30:00	62.555688	22.965797	17.073516
2007-06-06 15:00:00	63.815123	22.869189	17.166643
2007-06-06 15:30:00	64.946597	22.470817	17.159937
2007-06-06 16:00:00	67.336973	22.186515	16.841663
2007-06-06 16:30:00	67.237105	22.166204	16.926711

Table 3: Extract the of the weighted weather data

Exploratory Visualization

For the study, like mention before there is seasonal effect on the energy consumption in France, in the following figure there is an illustration of this seasonal effect.

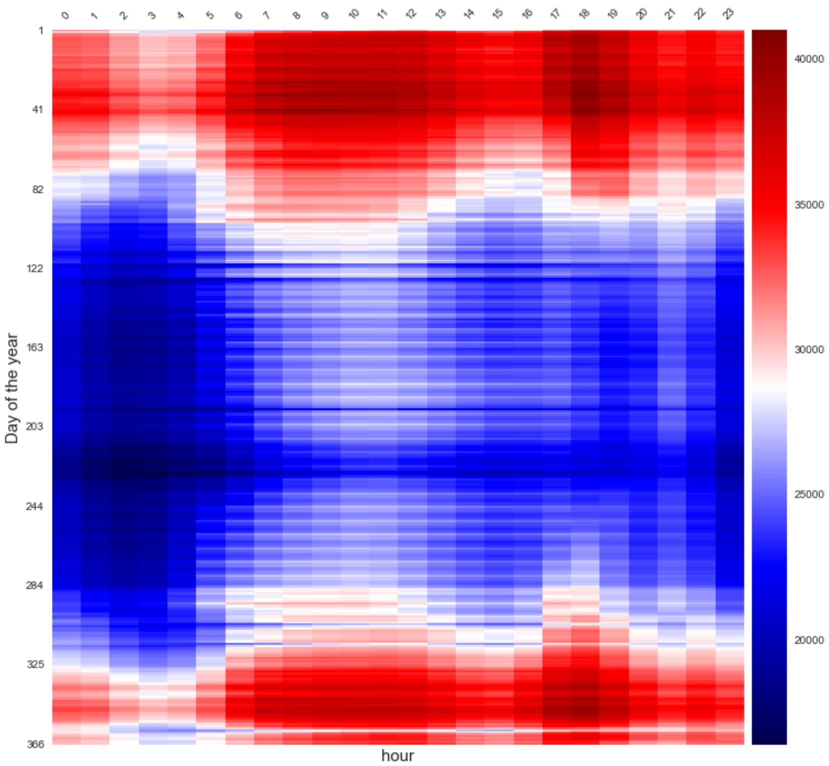

Figure 1: Seasonal effect of the electricity consumption in France in function of the day of the year and the hour of the day

This figure is a good illustration of the impact of the moment of the year in the electricity consumption, it is less important during the summer (blue area in the centre of the heatmap) than during the winter. This heatmap is also an illustration of the variation of the electricity consumption in function of the moment of the day, the consumption seems less important between 0:00 and 07:00 than 07:00 and 12:00, and there is a “peak” between 17:00 and 20:00, this is the illustration of the rhythm of life of French people with the moment where they are active in their household during the “peak of the demand” in the heatmap. Another point to notice is the presence of some lack of consumption during some day, like during the Christmas period it can be noticed that the consumption seems less important than the day of the same month it can be explained by the fact that the industry and the people are in holidays so the consumption will be less important for this moment.

The final point on the seasonal effect analysis illustrates that in function of the day the consumption can be different. In the following figure there is an illustration of the consumption in France in function of the day of the week and the month of the year.

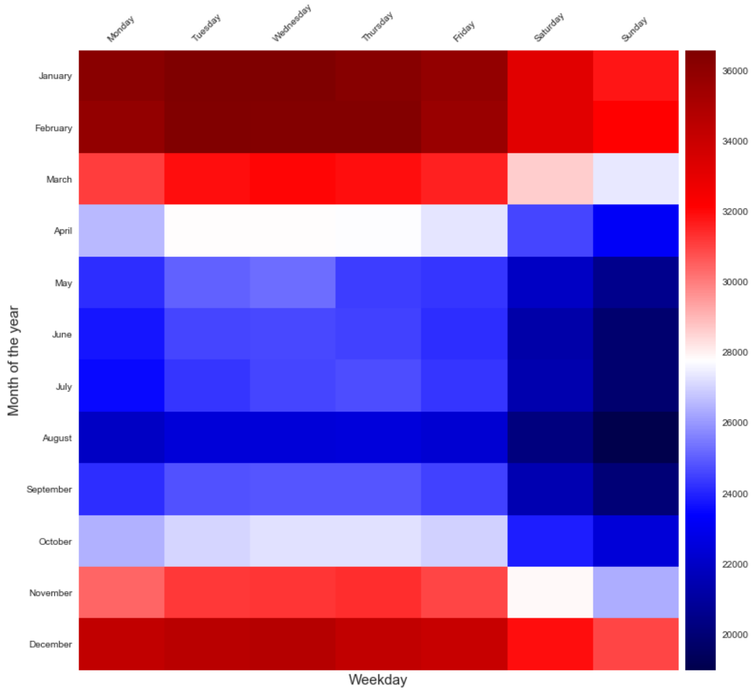

Figure 2: Heatmap of the average consumption in function of the day of the week and the month of the year

This heatmap illustrates the difference between the day of the week for the same kind of month, there is consumption difference between the working days (Monday to Friday) and the day off (Saturday Sunday).

Algorithms and Techniques

For the analytics, the work will be focus on the regressor approach to estimate the consumption. The library Scikit learn[4] is going to be used to make the project.

It will be test the following models:

- Polynomial regressor[5]
- Random forest regressor[6]
- Decision tree regressor[7]
- K-nearest neighbours[8]
- Neural network MLP regressor[9]

For each model some parameters are going to be modified to find the better combination that fit for our problem. There is a list of the parameters that are going to be tune during the study:

- Polynomial regressor: degree

---

[4] http://scikit-learn.org/stable/
[5] http://scikit-learn.org/stable/auto_examples/linear_model/plot_polynomial_interpolation.html
[6] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
[7] http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
[8] http://scikit-learn.org/stable/modules/neighbors.html
[9] http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

- Random forest: max depth
- Decision tree: max depth
- K-Nearest neighbours: number of neighbours and weights technics
- Neural network: number of hidden layers, the activation function and the kind of solver

To create the model and find the right model, it's have been decided to split the dataset between a training set (80% of the dataset) and a testing set (20% of the dataset), the dataset has been randomised before the split.

In the case of the daily consumption that represents:

- Training set: 2337 samples
- Testing set: 585 samples

For the half hourly consumption:

- Training set: 112176 samples
- Testing set: 28080 samples

During the model development, the K-fold segmentation[10] is going to be used to create the best model based on the training set, the number of fold is going to be 10. The evaluation of the best model during the K-fold training is based on the best r^2 score.

Benchmark

There will be two models to surpass, one for the daily forecast and one for the half hourly forecast. For the daily consumption a piecewise linear approach is going to be used like explained in this paper[11].In this approach just the daily electricity consumption and the outdoor temperature (the Clermont Ferrand data) are going to be used. In the following figure there is an illustration of this model.

[10] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html
[11] ttp://www.ibpsa.org/proceedings/BS2015/p2854.pdf

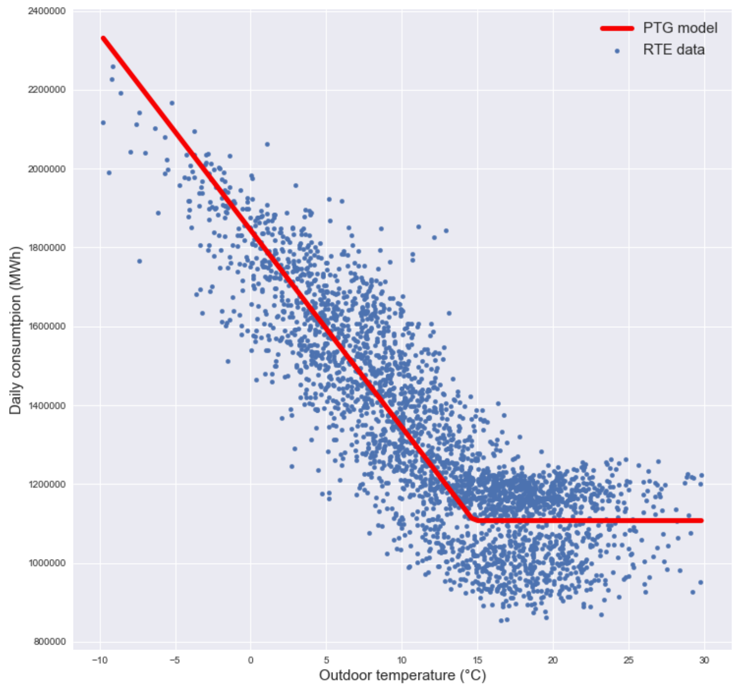

Figure 3: Illustration of the PTG model based on the RTE data and the outdoor temperature in Clermont-Ferrand

This approach illustrates the seasonal effect on the electricity consumption with the two parts on the model. In the following table there is an illustration of the efficiency of this model.

r^2 score	Time of execution (seconds)
0.757	0.021

Table 4:Metrics for the PTG model

The score for this model are not too bad with a r^2 score of 0.75 and the execution time is quite short.

For the half hourly forecast, this a time series problem so there is a model based on the moving average call ARIMA model[12] it's going to be our benchmark model of the half hourly consumption. It's only based on the electricity consumption. In the following table, there is a representation of his efficiency.

r^2 score	Time of execution (seconds)
-0.662	0.014

Table 5: Metrics for the ARIMA model

This model is not a good model, so it's going to be easy to create a better model and the works did on the daily consumption will give the keys to make this better model.

[12] https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

III. Methodology

Data Pre-processing

For this part, the pre-processing is quite straight forward. To create the model a join has to be done between the weather data and the RTE dataset for each approach the library pandas[13] is perfect to do this part of the project.

In the case of the daily consumption, a sum function has to be used on the electricity data and for the weather data an average function has to be used. It has been notice that there is an impact of the day of the week, the month of the year and the moment of the day so these features are going to be added in the dataset with the use of the built-in functions of pandas to convert a timestamp in another time index.

For the first part of the model only the outdoor temperature is going to be used but for the next steps when some extra features are going to be added like the weather parameters and time information the input date have to be normalised to limit the impact of some different order parameters.

Implementation and refinement

In each case for the two-forecast systems, the different are going to be tune and the best model are going to be selected based on the metrics chosen previously.

In the case of the polynomial regressor system, the degree used will be modified, the range of study will be between 0 and 15.

For the random forest and the decision tree regressors, the maximum depth of the tree will be modified between 2 and 15.

In the case of the K-nearest neighbours the number of neighbours will be modified between 1 and 30 and the weight functions will be the uniform functions or the distance functions.

For the neural network regressor, the number of hidden layers will be modified between 2 and 256, the activation functions could be the identity, the logistic, the tanh or the relu functions and for the solver is will lbfgs or adam.

There will be four steps on the analysis for each forecast system:

- Test with the outdoor temperature at Clermont-Ferrand
- Add of new features (find the right combinations of features)
- Limit the number of data for the training set
- Test a new weather dataset (the national one)

For each step, the process wants to have a grid search that can permit the evaluation of the accuracy of the model on the different configurations tested.

[13] http://pandas.pydata.org/

IV. Results

Model Evaluation and Validation (Daily forecast)

Model based on the outdoor temperature at Clermont-Ferrand

To present the result of the analysis, the representation will be the r^2score in function of the execution time but you can find in the appendix the table that contains all the result of the analysis (with the mean absolute error). In the following table there is the result for each main model used with their metrics associated and the parameter of the models.

Models	Parameters	r^2 score	Time of execution (s)
Polynomial regressor	Degree = 15	0.7504	0.005
Random forest regressor	Max depth = 4	0.7688	0.017
Decision tree regressor	Max depth = 4	0.7669	0.004
K-nearest neighbour	Number of neighbours =19 Weights = uniform	0.7652	0.005
Neural network MLP	Hidden layer size = 256 Activation = relu Solver = lbfgs	0.7713	0.651

Table 6: Best models for the daily forecast (based on outdoor temperature at Clermont-Ferrand)

If the comparison is based on the r^2 score, the polynomial regressor with a degree of 15 is the worst model but it's very close of the PTG model our benchmark model. In another metrics like the time of execution the neural network is very slow like 30 times the PTG model, the other models are quite fast. The more appropriate models seems to be:

- Random forest regressor
- Decision tree regressor
- K-nearest neighbour

Model based on some extra weather features

In this part, the outdoor humidity and the wind speed will be added in the input features. In the following table, there is the results for the models.

Models	Parameters	r^2 score	Time of execution (s)
Polynomial regressor	Degree = 15	0.7667	0.099
Random forest regressor	Max depth = 4	0.7788	0.026
Decision tree regressor	Max depth = 4	0.7737	0.003
K-nearest neighbour	Number of neighbours =27 Weights = uniform	0.7781	0.003
Neural network MLP	Hidden layer size = 256 Activation = relu Solver = lbfgs	0.7847	1.699

Table 7: Models for the works on the extra weather features

In terms of r^2 score, all the model seems better than the PTG but the gain of efficiency it's not very important for each model. The only model that has change his parameter is the k-nearest neighbour that has a higher number of neighbours. Except for the neural network, the time of execution is quite similar than for the PTG even with more features. For the next steps, it will be chosen to not add the two new features on the model.

Model based on extra time features

Like for the previous part, two new features will be added to the input features the type of day and the month of the year. In the following table, there is the results for the models.

Models	Parameters	r^2 score	Time of execution (s)
Polynomial regressor	Degree = 15	0.902	0.108
Random forest regressor	Max depth = 4	0.877	0.017
Decision tree regressor	Max depth = 4	0.856	0.004
K-nearest neighbour	Number of neighbours =19 Weights = uniform	0.917	0.002
Neural network MLP	Hidden layer size = 256 Activation = relu Solver = lbfgs	0.921	1.826

Table 8: Models for the works on the extra time features

The models are improving with these new features, it's kind of normal after the different observations that have been done on the seasonal and weekday effect. This point is very important for the next steps of the analysis.

Let's see the impact of the size of the training set on the different models

Impact of the training set size

In the following figure, there is a representation of the impact of the number of points in the r^2 score of the model on the testing set.

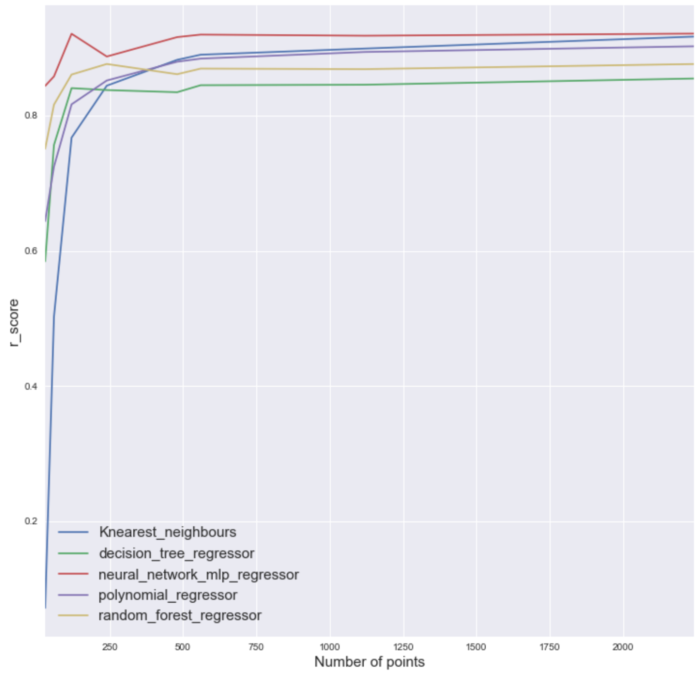

Figure 4: Evolution of the r^2 score in function of the number of points in the training set

This figure illustrates that the neural network is quite efficient quickly with a little amount of data, but basically after 2 years of data the model seems to reach his most efficiency parameter. The k-nearest neighbours is the model that seems to have big difficulty with a little amount of data.

Impact of the weather data (national dataset)

Models	Parameters	r^2 score (Clermont-Ferrand data)	r^2 score (National data)	r^2 score (National data + more weather data)	r^2 score (National data + more time data)
PTG		0.757	0.827		
Polynomial regressor	Degree = 15	0.75	0.811	0.829	0.92
Random forest regressor	Max depth = 4	0.768	0.831	0.836	0.91
Decision tree regressor	Max depth = 4	0.766	0.831	0.829	0.898
K-nearest neighbour	Number of neighbours =19 Weights = uniform	0.765	0.823	0.831	0.931
Neural network MLP	Hidden layer size = 256 Activation = relu Solver = lbfgs	0.771	0.829	0.84	0.904

Table 9: Impact of the weather data on the r^2 score

The models seem more efficient with this new input date, but when some new inputs weather features are added to the training set the gain are very limited. The addition of the time feature is very interesting. The conclusion on this part of the project is that the national weather data seems more efficient but the choice to use Clermont-Ferrand was not a bad first approximation.

The model that seems the most interesting in term of efficiency and execution is definitely the k nearest neighbours, this model doesn't to have a lot of parameters to be tuned to reach a very good score but it need a more important training set that the other.

Half hourly forecast

To make this forecast, and after the works on the daily forecast it has been decided to choose as inputs:

- The national weather data (only the outdoor temperature)
- The time feature (month, day if the week and half hour of the day)

To obtain a better model of the ARIMA, the focus will be done on the model that have been used for the daily forecast, the PTG is out of the race because to use the PTG it will be necessary to create a model for each half hourly slot in a day and it seems not very efficient to do that.

Models	Parameters	r^2 score	Time of execution (s)
Polynomial regressor	Degree = 8+	0.903	2.77
Random forest regressor	Max depth = 14	0.944	1.58
Decision tree regressor	Max depth = 13	0.936	0.21
K-nearest neighbour	Number of neighbours =28 Weights = distance	0.945	0.26
Neural network MLP	Hidden layer size = 256 Activation = logistic Solver = lbfgs	0.93	156.28

Table 10: Results for the best models of the half hourly forecast

This part illustrates that the models have a better r^2score than the ARIMA model, but their time to create is longer than for the ARIMA model.

Justification

In all the cases, the models tested are quite all more performant than the benchmark model proposed. The most efficient seems to be the K-nearest neighbours but this model need a proper training set to be efficient, the neural network is very performant but his execution time is quite long in comparison to the others.

V. Conclusion

Free-Form Visualization

In the following figure, there is representation of the efficiency of the K-nearest neighbours for the half hourly forecast system that have been develop.

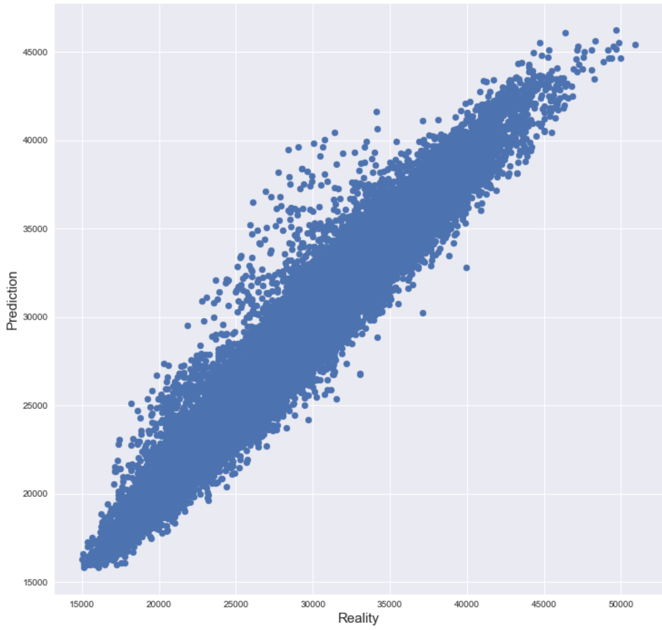

Figure 5: Representation of the best model for the half hourly forecast (K-nearest neighbour)

In both case, for the daily or the half hourly model, the models propose are very efficient like the one present in the figure above.

Reflection

For this project, the approach has been to cross the weather data and the electricity consumption to make a forecast engine that can be used to forecast the electricity consumption at a daily scale or 30 minutes scale. The approach for both of the forecast problem has been quite the same and the solution find are quite more performant that the most obvious technics used in this area.

The first part of the project has been to create the dataset, cross the different sources to create the dataset that has been used in the benchmarking of models to find the right model that fit our problem. The second part has been to find the models, test these models in different configurations to find the right one that is efficient and fast to build (or at least not too far from the know solution).

Improvement

It seems that the models build during this project are quite very efficient in general but there is a lot more technics to test like deep learning or reinforcement learning algorithm to train a better model.

Another part to improve is the input of the model, the weather data can be improved but maybe adding some extra information like demographic information on the department, information on the industry installed in the department can be a valuable asset to improve the model, and in term of time data some information on the holiday period could be great to add.

Appendix 1: Results

Model based on the outdoor temperature at Clermont-Ferrand

Polynomial regressor

models	r2_score	mean_abs_error	execution_time
PR: 0	-0,00028	223060	0,005019
PR: 1	0,696451	117764,5	0,004012
PR :2	0,746055	108075,4	0,005015
PR: 3	0,750357	107224,1	0,004015
PR: 4	0,750329	107243,9	0,004513
PR: 5	0,750445	107217,1	0,005514
PR: 6	0,750444	107217,5	0,004513
PR: 7	0,750448	107216,3	0,005014
PR: 8	0,750448	107216,3	0,004512
PR: 9	0,750448	107216,3	0,006016
PR: 10	0,750448	107216,3	0,004503
PR: 11	0,750448	107216,3	0,005515
PR: 12	0,750448	107216,3	0,005013
PR: 13	0,750448	107216,3	0,006517
PR:14	0,750448	107216,3	0,005013
PR:15	0,750448	107216,3	0,005014

Random forest regressor

models	r2_score	mean_abs_error	execution_time
RF:2	0,747262	108781,9	0,022092
RF:3	0,766777	105124,7	0,01504
RF:4	0,768864	104500,3	0,017734
RF:5	0,765735	105304,9	0,01905
RF:6	0,760363	106591,7	0,017576
RF:7	0,753032	108273	0,024596
RF:8	0,745756	109723	0,024065
RF:9	0,733702	112260	0,022057
RF:10	0,72283	113064,5	0,026069
RF:11	0,71027	115494	0,026604
RF:12	0,706594	116218,9	0,029079
RF:13	0,685869	120027,5	0,03363
RF:14	0,678777	120585,7	0,031552

Decision tree regressor

models	r2_score	mean_abs_error	execution_time
DT:2	0,721268	112554,6	0,005011
DT:3	0,759756	105876,3	0,004036
DT:4	0,766962	105140,4	0,00401
DT:5	0,765678	105271,1	0,004014
DT:6	0,752932	108286,1	0,004016
DT:7	0,740599	110027,8	0,00401
DT:8	0,72773	113297,2	0,005012
DT:9	0,711444	114928,9	0,005053
DT:10	0,700739	116494,2	0,00703
DT:11	0,679241	120256,2	0,006517
DT:12	0,672549	121764,6	0,004986

DT:13	0,65595	124924,9	0,00501
DT:14	0,646006	126137,8	0,005548
DT:15	0,642339	127282,5	0,00501

K-nearest neighbours

models	r2_score	mean_abs_error	execution_time
KNN:1/uniform	0,525106	145006,4	0,005039
KNN:1/distance	0,525106	145006,4	0,005989
KNN:2/uniform	0,654323	125133,6	0,004985
KNN:2/distance	0,619362	130605,1	0,004012
KNN:3/uniform	0,683304	120228,2	0,004011
KNN:3/distance	0,644205	127150,5	0,004514
KNN:4/uniform	0,701642	116914,2	0,004014
KNN:4/distance	0,65579	124999	0,005013
KNN:5/uniform	0,724702	112472,2	0,003982
KNN:5/distance	0,662985	123887,1	0,004978
KNN:6/uniform	0,732277	111670,4	0,004479
KNN:6/distance	0,668894	122674,1	0,005014
KNN:7/uniform	0,737447	109763,7	0,00448
KNN:7/distance	0,674098	121728,3	0,00498
KNN:8/uniform	0,744979	107895	0,004014
KNN:8/distance	0,677536	121012,6	0,00401
KNN:9/uniform	0,74743	107556,7	0,004011
KNN:9/distance	0,679697	120663,3	0,005014
KNN:10/uniform	0,752533	107003,6	0,004041
KNN:10/distance	0,681563	120300,3	0,004506
KNN:11/uniform	0,755094	106450	0,004012
KNN:11/distance	0,683273	119845	0,004011
KNN:12/uniform	0,755154	106526,5	0,004006
KNN:12/distance	0,684948	119543,5	0,005002
KNN:13/uniform	0,758673	105684,2	0,004548
KNN:13/distance	0,685669	119434,8	0,005013
KNN:14/uniform	0,758487	106052,7	0,003538
KNN:14/distance	0,686261	119341,7	0,004014
KNN:15/uniform	0,759612	105961,1	0,003989
KNN:15/distance	0,68732	119130,5	0,005013
KNN:16/uniform	0,758065	106135,6	0,004013
KNN:16/distance	0,687402	119148,3	0,00451
KNN:17/uniform	0,760401	105657,9	0,004013
KNN:17/distance	0,687969	119039,2	0,005014
KNN:18/uniform	0,759941	105814,9	0,005007
KNN:18/distance	0,688477	118950	0,005001
KNN:19/uniform	0,761403	105724,2	0,004013
KNN:19/distance	0,689419	118802	0,003983
KNN:20/uniform	0,760331	106201,1	0,004008
KNN:20/distance	0,690329	118668,5	0,004011
KNN:21/uniform	0,760956	106428,8	0,005013
KNN:21/distance	0,691069	118466,5	0,004514
KNN:22/uniform	0,761696	106184,4	0,004007
KNN:22/distance	0,691387	118411	0,005012
KNN:23/uniform	0,76276	105990	0,006037

KNN:23/distance	0,691842	118313	0,00401
KNN:24/uniform	0,76392	105589,8	0,004993
KNN:24/distance	0,69218	118278,4	0,004977
KNN:25/uniform	0,763955	105513,9	0,005007
KNN:25/distance	0,692377	118251,9	0,004977
KNN:26/uniform	0,76384	105782	0,005013
KNN:26/distance	0,692744	118176,5	0,004512
KNN:27/uniform	0,764424	105638,3	0,004014
KNN:27/distance	0,692997	118163,2	0,005014
KNN:28/uniform	0,765079	105427	0,005005
KNN:28/distance	0,693217	118116,9	0,004009
KNN:29/uniform	0,765267	105376,5	0,005008
KNN:29/distance	0,693451	118053,4	0,004977

Neural Network MLP

models	r2_score	mean_abs_error	execution_time
NN_MLP:2/identity/lbfgs	0,695703	117750,2	0,008021
NN_MLP:2/identity/adam	-24,9376	1323274	0,482284
NN_MLP:2/logistic/lbfgs	-0,00028	223060	0,01805
NN_MLP:2/logistic/adam	-24,9382	1323291	0,475263
NN_MLP:2/tanh/lbfgs	-0,00028	223060	0,014037
NN_MLP:2/tanh/adam	-24,9382	1323290	0,483285
NN_MLP:2/relu/lbfgs	0,695703	117750,2	0,014539
NN_MLP:2/relu/adam	-24,9376	1323274	0,481281
NN_MLP:4/identity/lbfgs	0,695703	117750,2	0,008021
NN_MLP:4/identity/adam	-24,9367	1323253	0,463232
NN_MLP:4/logistic/lbfgs	-0,00028	223060	0,02256
NN_MLP:4/logistic/adam	-24,938	1323285	0,489301
NN_MLP:4/tanh/lbfgs	-0,00028	223060	0,016043
NN_MLP:4/tanh/adam	-24,9379	1323283	0,517376
NN_MLP:4/relu/lbfgs	0,695703	117750,2	0,011531
NN_MLP:4/relu/adam	-24,9375	1323274	0,488802
NN_MLP:8/identity/lbfgs	0,695703	117750,2	0,008523
NN_MLP:8/identity/adam	-24,9354	1323218	0,459723
NN_MLP:8/logistic/lbfgs	-0,00028	223060	0,046122
NN_MLP:8/logistic/adam	-24,9376	1323276	0,542444
NN_MLP:8/tanh/lbfgs	0,7642	104252,2	0,062667
NN_MLP:8/tanh/adam	-24,9376	1323274	0,612128
NN_MLP:8/relu/lbfgs	0,695703	117750,2	0,013035
NN_MLP:8/relu/adam	-24,9366	1323249	0,513867
NN_MLP:16/identity/lbfgs	0,695703	117750,2	0,009526
NN_MLP:16/identity/adam	-24,932	1323132	0,491808
NN_MLP:16/logistic/lbfgs	-0,00028	223060	0,061149
NN_MLP:16/logistic/adam	-24,937	1323259	0,599094
NN_MLP:16/tanh/lbfgs	-0,00028	223060	0,033591
NN_MLP:16/tanh/adam	-24,9368	1323253	0,650244
NN_MLP:16/relu/lbfgs	0,695703	117750,2	0,015542
NN_MLP:16/relu/adam	-24,9348	1323204	0,571519
NN_MLP:32/identity/lbfgs	0,695703	117750,2	0,012533
NN_MLP:32/identity/adam	-24,925	1322950	0,507887
NN_MLP:32/logistic/lbfgs	-0,00028	223060	0,024564

NN_MLP:32/logistic/adam	-24,9356	1323222	0,775563
NN_MLP:32/tanh/lbfgs	-0,00028	223060	0,06768
NN_MLP:32/tanh/adam	-24,935	1323205	0,838732
NN_MLP:32/relu/lbfgs	0,770608	103793,1	0,044117
NN_MLP:32/relu/adam	-24,9349	1323206	0,622154
NN_MLP:64/identity/lbfgs	0,695703	117750,2	0,028075
NN_MLP:64/identity/adam	-24,9119	1322611	0,604608
NN_MLP:64/logistic/lbfgs	-0,00028	223060	0,100266
NN_MLP:64/logistic/adam	-24,9328	1323147	0,904405
NN_MLP:64/tanh/lbfgs	-0,00028	223059,9	0,109794
NN_MLP:64/tanh/adam	-24,9317	1323119	1,118977
NN_MLP:64/relu/lbfgs	0,770703	103913,6	0,20705
NN_MLP:64/relu/adam	-24,9258	1322969	0,77456
NN_MLP:128/identity/lbfgs	0,695703	117750,2	0,051637
NN_MLP:128/identity/adam	-24,8849	1321914	0,659755
NN_MLP:128/logistic/lbfgs	-0,00028	223060,2	0,333387
NN_MLP:128/logistic/adam	-24,9272	1322999	1,300959
NN_MLP:128/tanh/lbfgs	-0,00028	223060	0,246155
NN_MLP:128/tanh/adam	-24,925	1322940	1,735116
NN_MLP:128/relu/lbfgs	0,770745	103880,3	0,414603
NN_MLP:128/relu/adam	-24,9157	1322710	1,009684
NN_MLP:256/identity/lbfgs	0,695703	117750,2	0,088234
NN_MLP:256/identity/adam	-24,831	1320521	0,75952
NN_MLP:256/logistic/lbfgs	0,722619	107850	2,559307
NN_MLP:256/logistic/adam	-24,916	1322704	1,838891
NN_MLP:256/tanh/lbfgs	-0,00028	223060	0,503338
NN_MLP:256/tanh/adam	-24,9116	1322585	3,03106
NN_MLP:256/relu/lbfgs	0,771357	103648,8	0,654742
NN_MLP:256/relu/adam	-24,8896	1322035	1,453366