Analysis of Building Data for Final Project(MAE600)

Matthew Conrad[1]
Constantine Hadjidimoulas[1]
Kamalendu Paul[1]
[1]Syracuse University, New York, USA

Abstract

The report highlights a problem of linear regression of building data. There are 8 features are the various weather conditions and the target is the aggregated meter reading of the building. The report focuses on the simple ideas of regression including data cleaning, feature selection and tuning hyper parameters. It also takes a look at some of the advanced neural network techniques and a simple vector machine regression. Finally, these results are compared and is followed by a discussion on the numbers in the models.

Introduction

This is a study of the meter readings and the weather condition variable for a large office building in San Antonio. The time-frame of the data collected is from January of 2019 to June of 2019 at 5 minute intervals.

The data set was provided to me as part of the data used for this midterm test. The original data set was a comma separated values(csv) file that is contained the following columns in order.

- *Date/Time*
- Atmospheric pressure
- Ambient temperature
- Relative humidity
- Dew point temperature
- Solar radiation
- Wind speed
- Gust speed
- Wind direction
- **Meter reading**

The problem that I am interested in working on is a regression problem. Here the features that are in the data set are represented in the normal font(in the list above), and the target is shown in bold(Meter reading). The date/time stamp is ultimately omitted and not considered a part of the data set for our problem at hand. They are, however, used to sort the data into weekdays and weekends.

The report is going to be a supplement to the python code that is being used to make the analysis. The report is structured into sections. The first section is dedicated to the process and logic behind data cleaning. This is followed by a discussion on the relationships between the features and the target and among the features themselves. The next section deals with the decision making on the type of model and the procedure of the methods used to tackle the regression problem. The final section of the paper deals with listing the results and a brief discussion on the same.

Data cleaning and relationship between datasets

The data cleaning process was a five step process which will be laid out in the following paragraphs.

The first step of the data cleaning process was removing all of the points that contained "NaN" or "Not a Number" points. The second step was splitting the data into weekdays and weekends. This was done so because it was theorized that an office building would have different energy levels during the weekdays, when it was occupied, as compared to the weekends when it was unoccupied. The corresponding correlation matrices including the original data frame and its corresponding correlations as compared to the dataframe after it was split into both a weekdays and weekends dataframe. This is shown in Figures 1, 2 and 3.

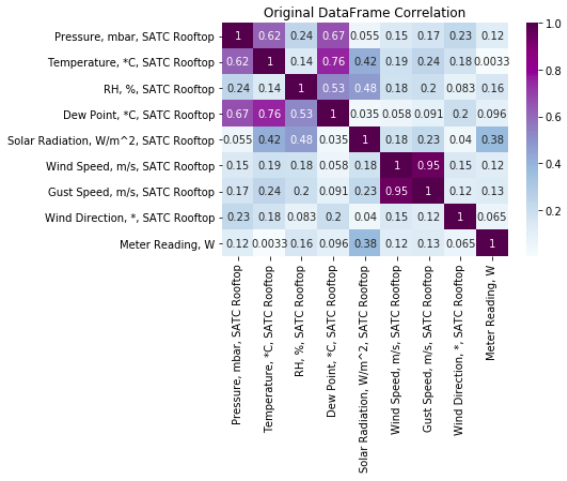

Figure 1: Full Data Correlation

For the purpose of the task we are trying to accomplish, predicting the meter reading (our target data) the main column we are concerned with in this correlation matrix is the last column. As can be seen from these correlation matrices the weekdays have a

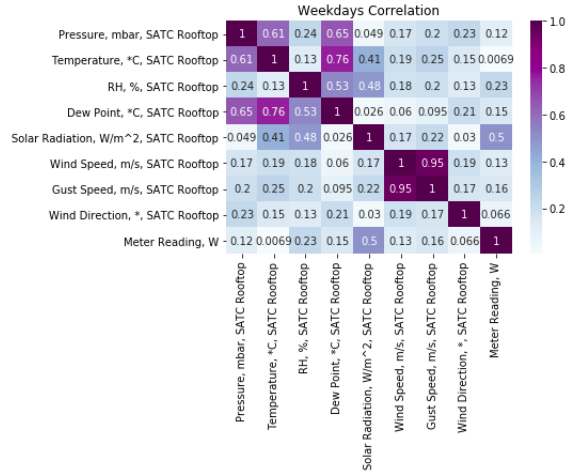

Figure 2: Weekday Correlation

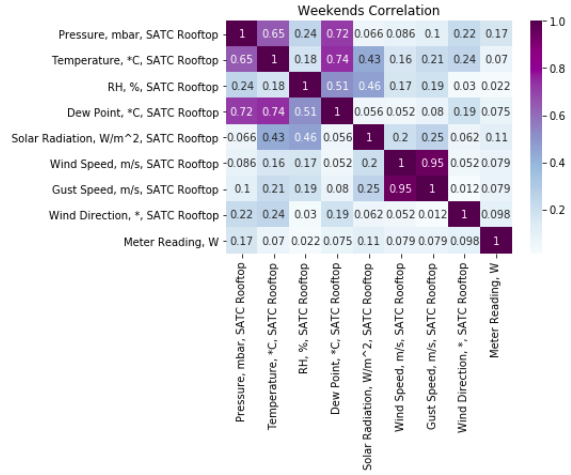

Figure 3: Weekend Correlation

stronger correlation to the target than the original data frame. Furthermore, it is seen that the weekends have a consistently weaker correlation to the target data than the weekends. This is not ideal but it was deemed acceptable because although the weekends have a weaker correlation they also had a lower variance. Therefore it was believed that the models that will be used will still be able to predict relatively accurately the points that are of interest (this will be discussed in greater depth in the linear regression models section).

The data was also split into working hours and non working hours. This seemed to be a little too fine tuned, however, because there appeared to be a "grace" period as to when workers entered and left the office. Therefore we opted to stay with the weekdays/weekends split instead of the work hours/off hours split

From here the data was further cleaned by removing features that had low correlation to the target data. The features that were removed were: pressure with a correlation of 0.12, temperature with a correlation of 0.0033, dew point with a correlation of 0.096, and wind direction with a correlation of 0.065. Next every four points were averaged together. This reduced the size of the data set by a factor of four. Instead of having a time series split up into 5 minute intervals we instead had a time series which had data points occurring every 20 minutes. This served to both normalize the variance in the data as well as to enable our machine learning algorithms to parse through the data more efficiently.

Finally outliers were removed from the dataset. This was done by removing any data point that had a z-score greater than 3. This served to clean the data of any points that could have been incorrectly recorded either through inaccuracies in the data capturing devices or through some error elsewhere. The final correlation matrices can be seen in Figures 4 and 5.

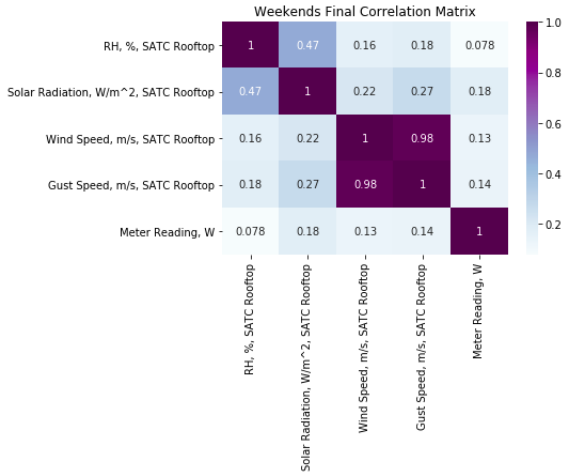

Figure 4: Weekend Correlation Final

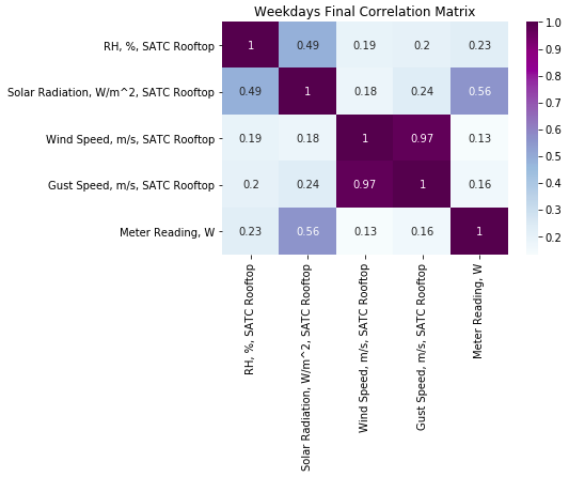

Figure 5: Weekday Correlation Final

As can be seen from these above matrices compared to the original matrices, the correlation between the features and the target is much better. This will be very important when moving forward and applying the machine learning algorithms to said data.

Problem statements

The problem is a very simple one. It is a regression problem. We try to use various data cleaning techniques to choose the best way to categorize the data and select the features.

The various ways we could clean our data for better results were based around the fact that this being an office building has weekdays and weekends. The results presented in the report is for the weekdays because of two reasons. Firstly, it was a much larger set of data and secondly, because the results obtained were much reliable for the weekdays.

The we use several models, in the category of regression, neural networks and support vector machines and try to make a comparisons by looking at the error metrics.

To be precise, the regression models used are linear, lasso, ridge and elastic net. The neural networks used are deep, recurrent and long short term memory. The SVM used is the support vector regression.

Finally at the end of the training and testing, the results are used to summarize and look for over-fitting and under-fitting.

Model architecture and hyper parameters

With the Training, Validation and Testing data sets established, the regression models need to be applied. The different five regression models take different approaches to the problem and help identify the best fit for the prediction data to the actual recorded data. The models will be evaluated based on their performance and their produced errors (error metrics: RMSE, MAPE, MAE, MBE, R2).

The four regression models are the following: **Linear Regression, Ridge Regression, Lasso Regression, and Elastic Net Regression**.

Linear Regression is used to show or predict the relationship between two variables or factors. The factor that is being predicted is called the dependent variable. It is a simple linear approach.

Ridge Regression is a variant of linear regression, a technique for the analysis of multiple regression data that experience multicollinearity.

Lasso Regression is a regression analysis method that performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the resulting model.

Elastic Net Regression is a commonly used model of regression which incorporates penalties from both lasso and ridge regularization. With the help of these models the best performance of the training and testing model will be identified.

These models are applied on the different sets of data the team created for weekdays and weekends.

The next model in line is the neural network models and the support vector machine regression. For the neural network we used deep neural network, recurrent neural network and long short term memory. In the following few paragraphs we will discuss the architecture and the hyper-parameters of the models.

For the **deep neural network** I made a 3 layer deep network. The activation function used is **RELU** activation. The optimizer used is **ADAM**. The network is a fully connected one. The hyper-parameters are the number of epochs, the number of nodes in the first two layers. The third layer is the one node layer because it is a regression problem with one output. The results are computed using a 5 fold cross validation.

The **recurrent neural network** is also implemented using the number of layers as the hyper-parameter. The number of hidden layers used for the same is 50 and the input nodes and output nodes are 5 for the features and target. The number of epochs is again 100 and the 400 data points are chosen for a week's worth of data and prediction with 25 percent for testing. The sequence length is 10 and the machine learns to predict the next data point

The **LSTM** has the same attributes as the **RNN**. The hyper parameters again is the number of layers.

The **SVM** model has a fairly straightforward and simple implementation. We use the support vector regression module from **sklearn**. The kernel used for the model is *rbf*. The hyper-parameters for the model are the **C** and the **gamma**. We use a cross validated hyper-parameter tuning for bettwe final results.

Model methodology and results

A correlation matrix is the first step needed in the classification of data before the training and testing starts. Now based on these data values the models will perform the regression with a **learning rate of 0.01** and a penalty $\alpha = 1$(for ridge, lasso and elastic net).

Running the **linear regression** model with 10,000 epochs, the model's performance is very accurate. As it can be seen from running the predictions against the actual data for weekdays in Figure 6. The testing and training RMSE, MAPE, MAE, and MBE are $(64843.337, 67430.765)$, $(28.073, 23.863)$, $(53369.472, 45392)$ and $-53224.819, -44466.082)$. The R2 testing is 0.676.

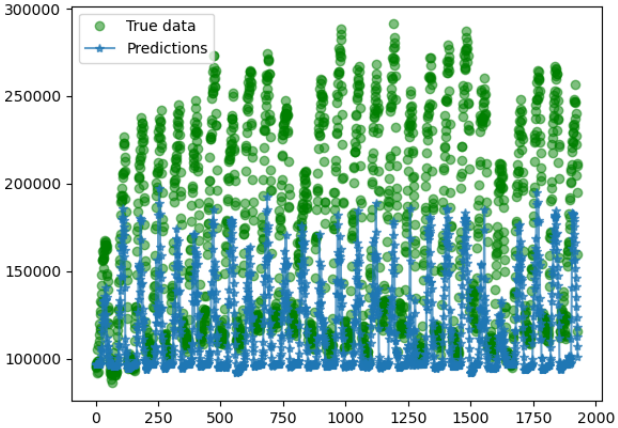

Figure 6: Linear regression for weekday data

The **ridge regression** for the weekdays is shown in Figure 7. The testing and training RMSE, MAPE, MAE, and MBE are $(65022.503, 67322.837)$, $(28.224, 23.785)$, $(53593.740, 45279.795)$ and $(-53462.227, -44312.504)$. The R2 testing is 0.676.

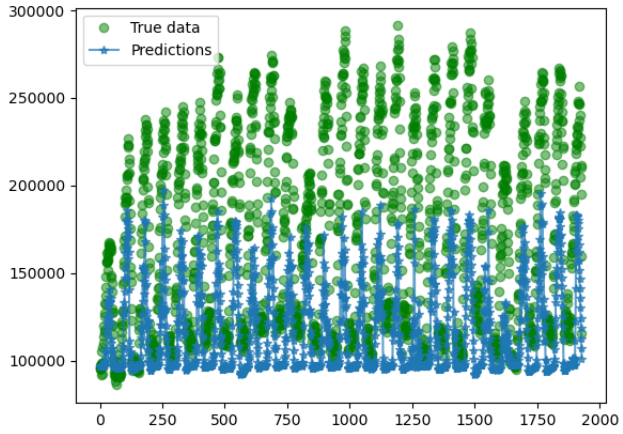

Figure 7: Ridge regression for weekday data

The **lasso regression** for the weekdays is shown in Figure 8. The testing and training RMSE, MAPE, MAE, and MBE are $(35892.584, 67677.031)$, $(19.162, 24.051)$, $(30845.287, 45688.977)$ and $(-7445.524, -44832.998)$. The R2 testing is 0.679.

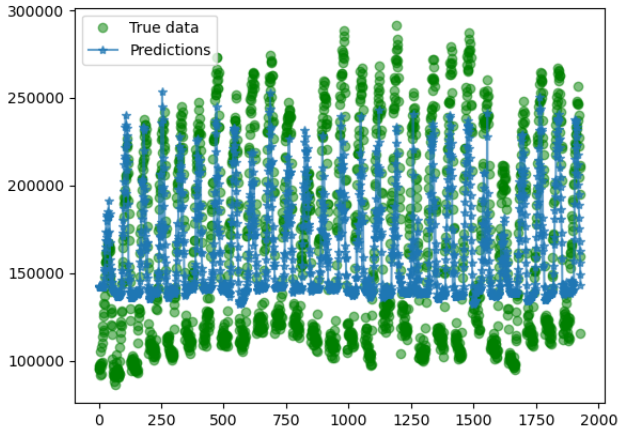

Figure 8: Lasso regression for weekday data

Elastic net regression model utilized 80% lasso and 20% ridge. After multiple tests this ratio was determined as the most accurate that can showcase the characteristics of the combining the two models. The results are shown in Figure 9 for the weekdays. The testing and training error are as follows. The training RMSE, MAPE, MAE, and MBE are 62051.026, 34.499, 49165.608, and −17471.179. The following testing errors are 71691.211, 37.156, 58090.644 and −14471.179.The R2 testing is 0.694.

Moving on to the *neural network* models. For the **deep neural network**, we started out by using the full dataset for the work hours as a training point. The results are shown in Figure 10.

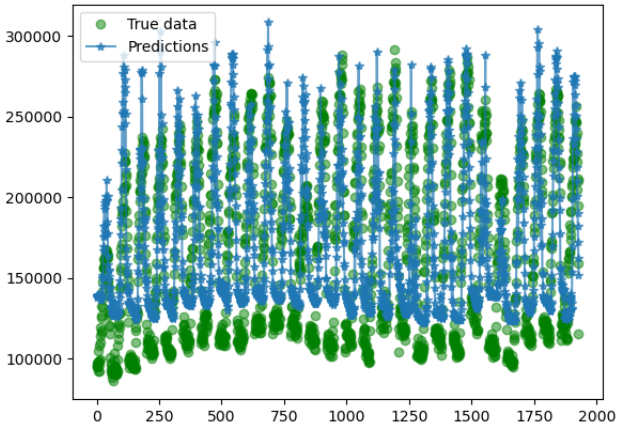

Figure 9: Elastic net regression weekdays

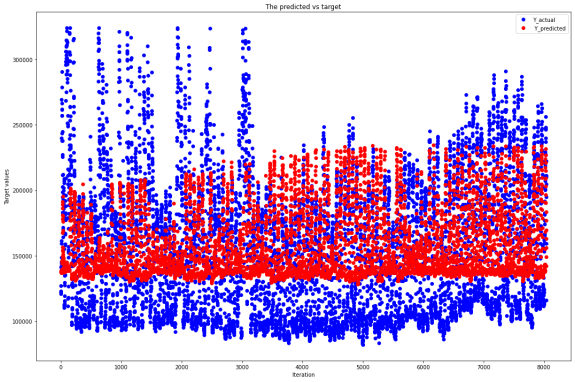

Figure 10: DNN full data set results

After the presentation it was pointed out that it was better to train the initial few and the later few data sets separately. This is reflected in Figure 11 and Figure 12.

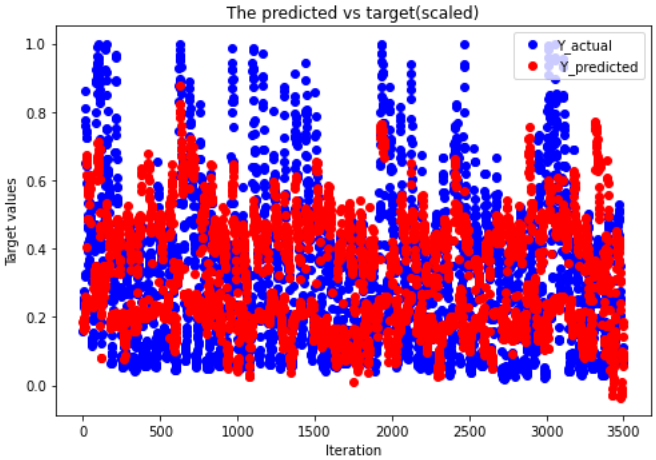

Figure 11: DNN first half data set results

Also, the hyper-parameters for the model are the nodes in the first two layers as discussed before-hand. These are denoted by (H1, H2). We made a grid search through 1 to 8. The learning rate is chosen to be 0.01 with 1000 epochs. The error for the RMSE, MAPE and R2 are shown for the testing ones in Figures 13, 14 and 15. The training and the second half figures are presented in the appendix so as to not clutter the report. The ideal hyper-parameters for

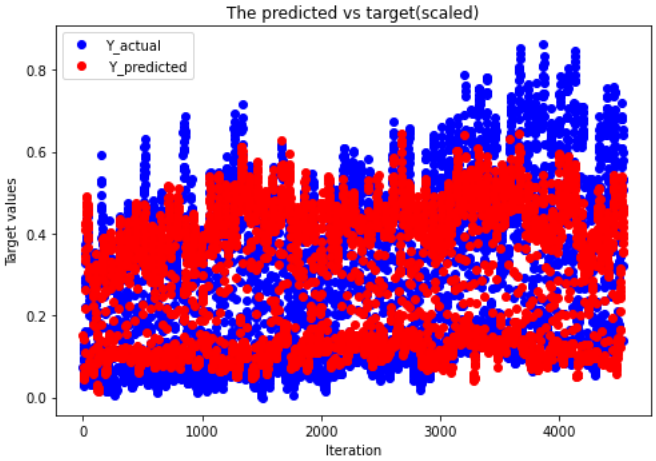

Figure 12: DNN second half data set results

the first half of the data-set has (9,6) and the second half has (7,6). These can be easily gathered from the error metric plots in Figures 13, 14 and 15.

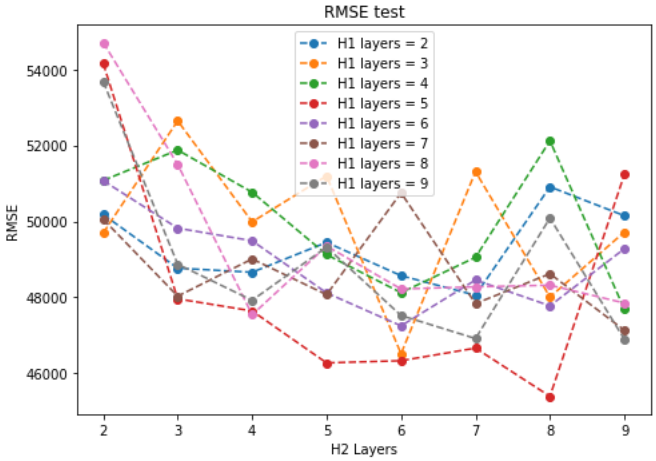

Figure 13: RMSE testing first half data set results

For the **RNN** the hyper-parameter, number of layers, was run through values of 1 to 4. It was found out that 1 layers gives the better result. The prediction is shown in Figure 16. The RMSE and MAPE for the same are 72987.931 and 41.165.

For the **LSTM** the hyper-parameter, number of layers, was run through values of 1 to 4. It was found out that 2 layers gives the better result. The prediction is shown in Figure 17. The RMSE and MAPE for the same are 38950.078 and 29.276.

For the **SVM** model we do a similar analysis. We start with the whole weekly dataset. Then we split it in search of better results. Then do a similar analysis. The hyper-parameters in this case is C and gamma with values running from C = $[0.1, 1, 10, 100]$ and gamma = $[1, 0.1, 0.01, 0.001, 0.00001]$. The best results are for C = 100 and gamma = 1.

The full data-set(Figure 18) had an testing RMSE of 36400.739 and testing MAPE of 16.215. The first half of the data-set(Figure 19) had an testing RMSE of 39996.123 and testing MAPE of 17.753. The second half of the data-set(Figure 20) had an testing RMSE

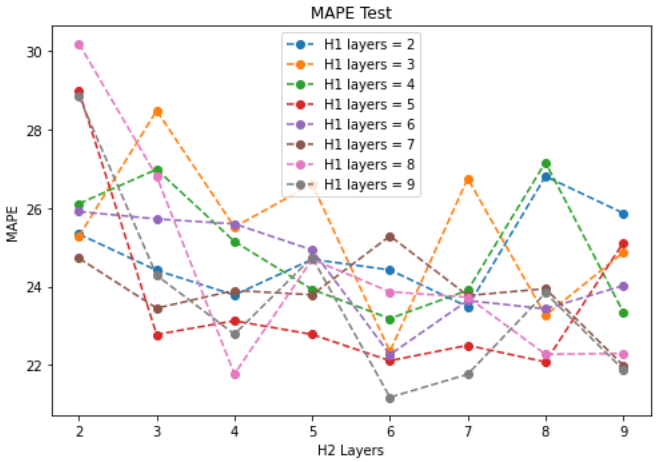

Figure 14: MAPE testing first half data set results

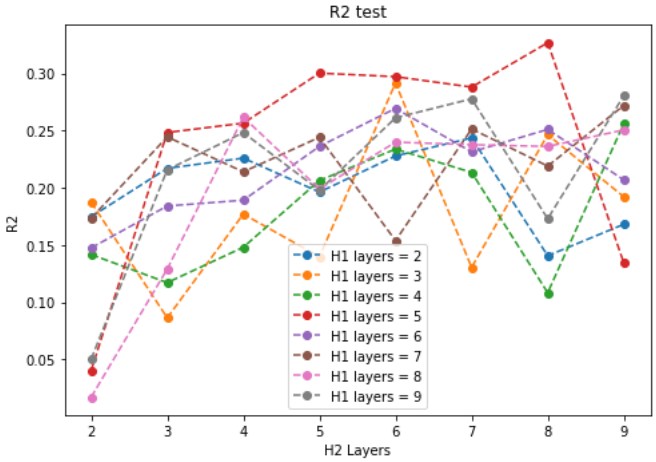

Figure 15: R2 testing first half data set results

of 24102.611 and testing MAPE of 13.064. So, as can be seen from this case. Splitting the dataset into too different portion having separate trends as seen previously can lead to better results.

Conclusion

When analysing different machine learning models there is a lot to keep in mind. By just looking at the graphs we cannot notice any obvious differences in our four models. When looking at the error metrics results someone needs to identify what error is of primary concern. Having split the data into weekdays and weekends we were able to achieve better results and better predictions than before with the one dataset. The team concentrated more on the Mean Absolute Percentage Error (MAPE) and R2 where a clear improvement was observed. The **cross validated error metrics** confirmed the model's performance improvement. The error metrics for the testing models slightly vary in a small range for MAPE and R2. For MAPE the testing error varies in each case, with lower values being considered ideal, different models affect differently this error metric as it can be observed in the graphs. The Elastic Net regression model resulted in MAPE values in the high 30% range, when compared the low 20% and high

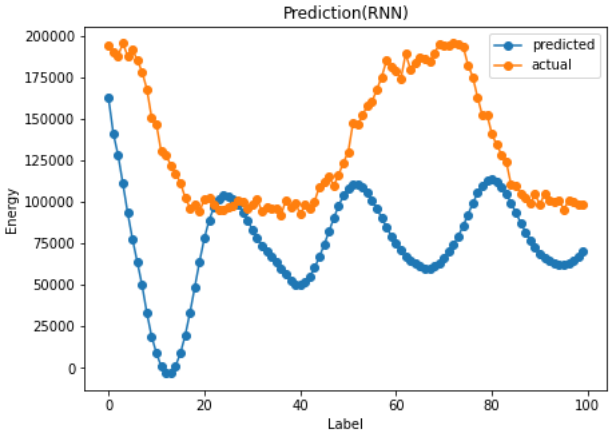

Figure 16: RNN results

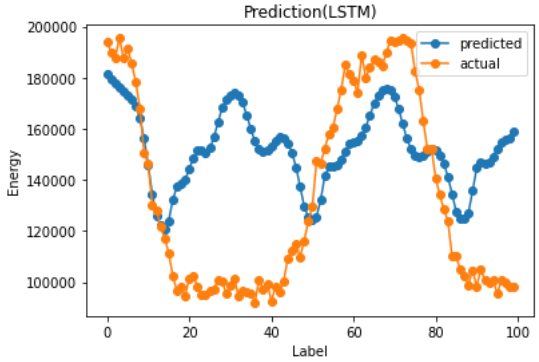

Figure 17: LSTM results

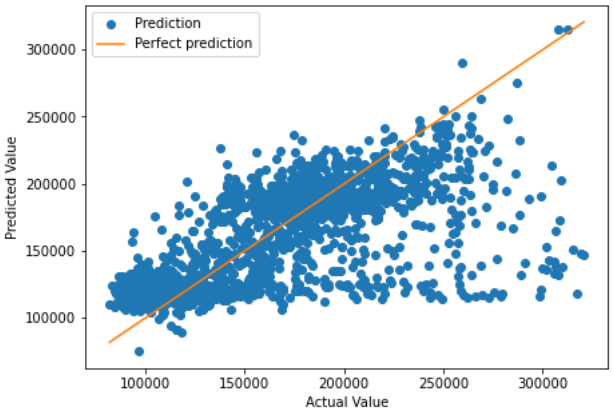

Figure 18: SVM on the full dataset

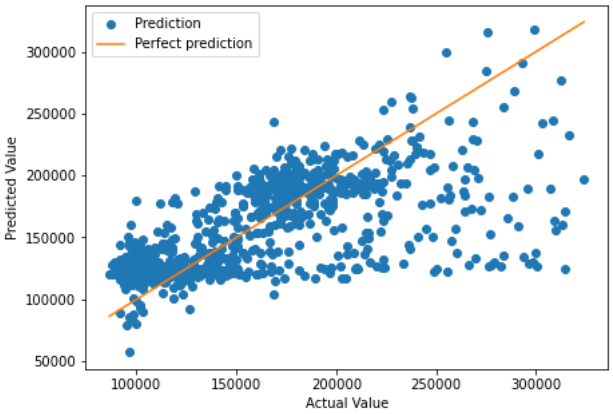

Figure 19: SVM on the first half dataset

10% ranges for the Linear, Lasso and Ridge regression models. These results for both testing and training for the weekdays show that our models are operating in good prediction accuracy. The Elastic Net model combines previous models that can be easily tuned and modified to satisfy any data variation. Looking at all the error metrics it outperforms all the first three models by utilizing the best of the Ridge and Lasso Regressions. Overall, there is no significant overfitting or underfitting anywhere in the data, so the statistical model did not begin to describe the random error in the data rather than the relationships between variables. This is something good and desired. Now considering the testing and training data. The goal of model testing is to make the testing as accurate as possible. Now for the weekend data, the results obtained were significantly worse to the weekday predictions. This was mainly since the correlation of some of the features to the target was different to those features obtained in the weekday data set. Here the trend is the same for the testing and training data, with an improvement of MAPE in the training data set. However, we do notice a better fitting for the Elastic Net model over the weekend data(see appendix), something that was not as significant in the weekdays data set. The results for testing and training were not a clear indicator of the models' training and testing performance. The slightly better results for testing indicate an underfitting character for the predictions. Trying to understand why the testing results are slightly better than training, the team suspects that it may have been caused by the nature of the data variations plus other factors like the change in seasons, that could affect the dry climate of San Antonio. Perhaps a separation of data based on the temperature variations could lead to more uniform data, that would give better results.

For the neural network models, a few of the discussion that we would like to base things around is how one chooses the data-set to train and test. As can be gather the splitting gives better results. This is attributed to the fact that there is a gradual shift of the power consumption due to seasonal changes.

The reason I decided to stick with the office hours, firstly because it had more data. Also, after feedback, I decided to split the data set into two parts for better training and results. This is apparent when we compare the full dataset being trained as opposed to parts being trained differently, especially in cases of deep neural networks and support vector machines.

In case of the RNN and the LSTM the choice of 400 datapoints revolved around the fact that 20 minute add up to a 5 day week. Since this is mainly good for time series data, this helps to predict the last 25 percent of the data much better. This in a way translates to predicting the results for a day, Friday, when you have knowledge of the past days in the week, Monday through Thursday.

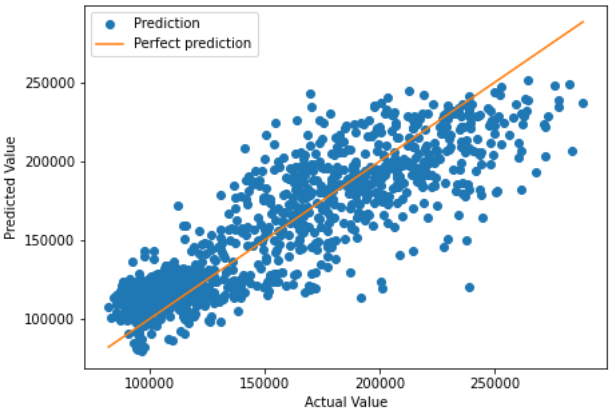

Figure 20: SVM on the second half dataset

Among all the methods in the neural network and the support vector machines, the DNN was much better than the SVM and, the LSTM was much better in terms of the time series forecasting. This is said by comparing the MAPE and RMSE errors for the models.

References

Most of the concepts have been adapted from the lecture notes of the class. I have also reused sections of the code from the laboratory class.

Appendix

This section is just used to put in results for the weekends for the regression methods. I also put in figures for the second half of the DNN and the epochs to loss function for the same.

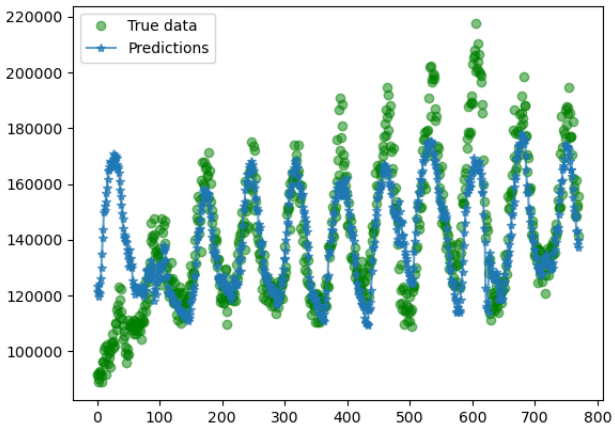

Figure 21: Elastic Net Weekends

The RMSE, MAPE, MAE, MBE and R2 for elastic net training and testing are $(33682.898, 31276.830)$, $(0.151, 0.181)$, $(19957.142, 25134.391)$, $(-13772.994, -17671.179)$ and $(,0.521)$.

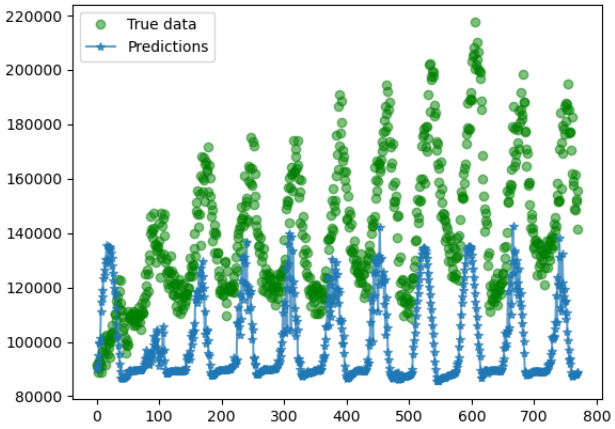

Figure 22: Lasso Weekends

The RMSE, MAPE, MAE, MBE and R2 for lasso training and testing are $(33682.898, 46492.496)$, $(0.151, 0.287)$, $(19957.142, 41859.566)$, $(-13772.994, -40123.582)$ and $(,0.174)$.

The RMSE, MAPE, MAE, MBE and R2 for ridge training and testing are $(33764.133, 46609.176)$, $(0.152, 0.288)$, $(20064.831, 41989.523)$, $(-13895.702, -40249.316)$ and $(,0.174)$.

The RMSE, MAPE, MAE, MBE and R2 for linear regression training and testing are $(33741.468, 46487.547)$, $(0.152, 0.287)$, $(20035.588, 41853.763)$, $(-13859.500, -40113.038)$ and $(,0.174)$.

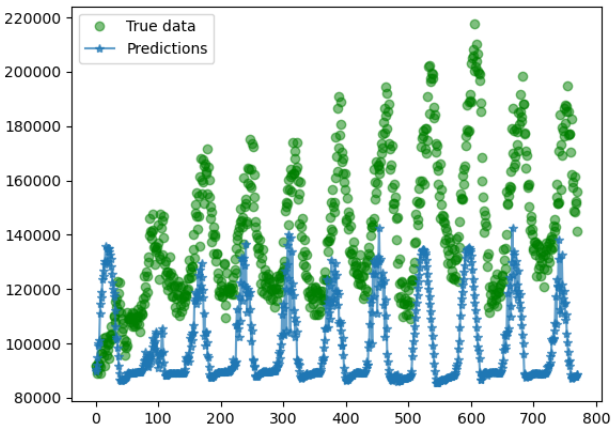

Figure 23: Ridge Weekends

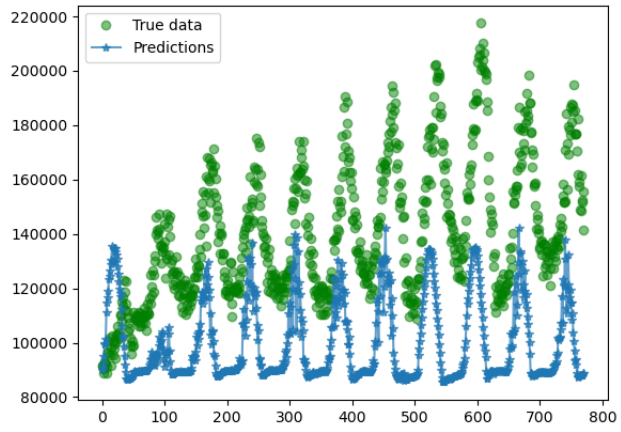

Figure 24: Linear Regression Weekends

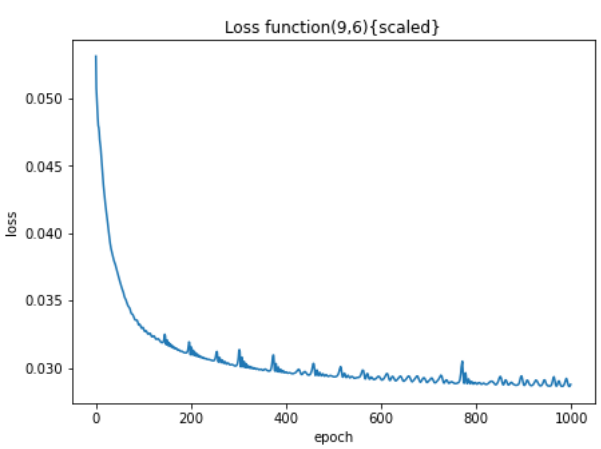

Figure 25: Loss epoch of first half DNN

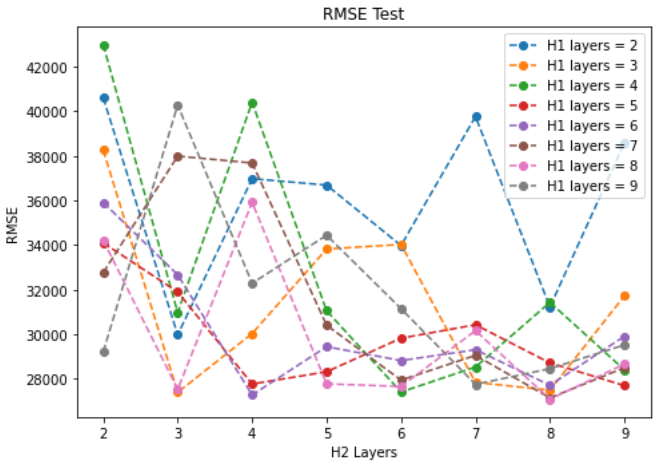

Figure 26: RMSE testing second half data set results

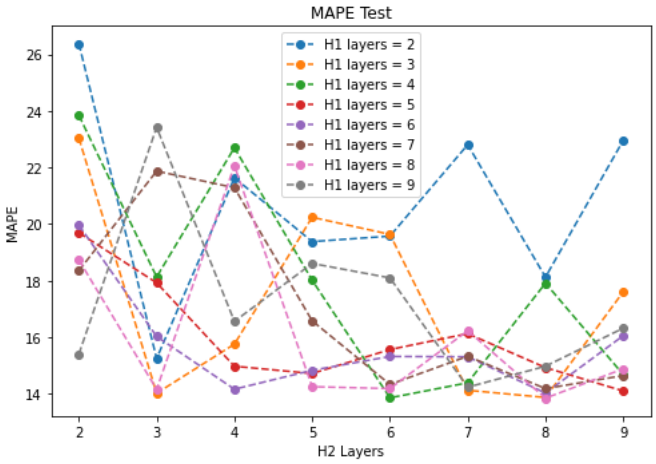

Figure 27: MAPE testing second half data set results

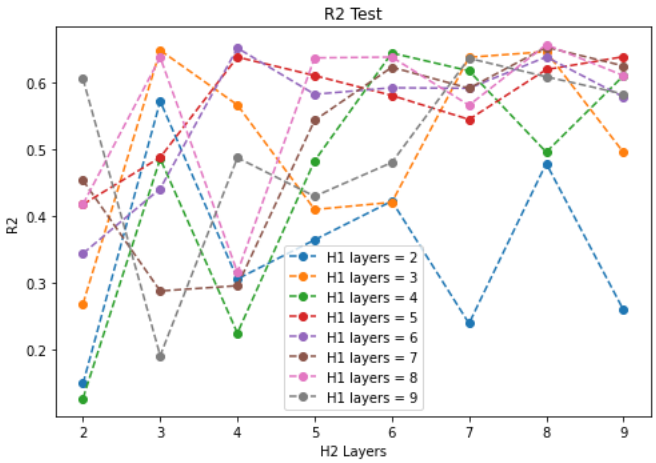

Figure 28: R2 testing second half data set results

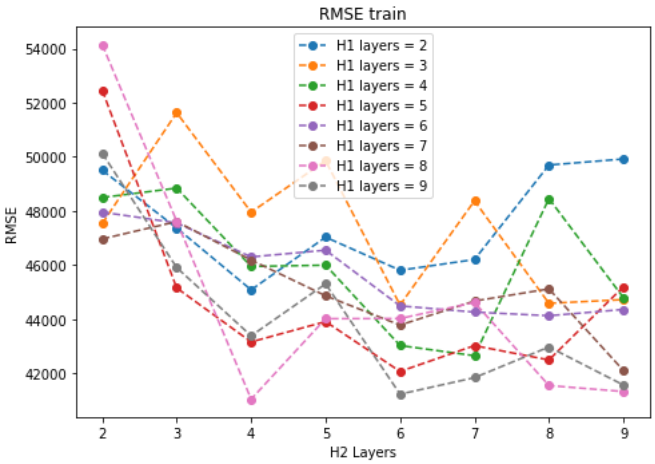

Figure 29: RMSE training first half data set results

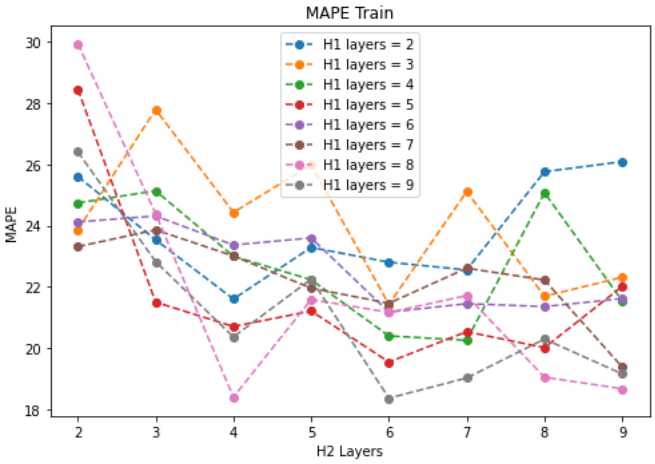

Figure 30: MAPE train first half data set results

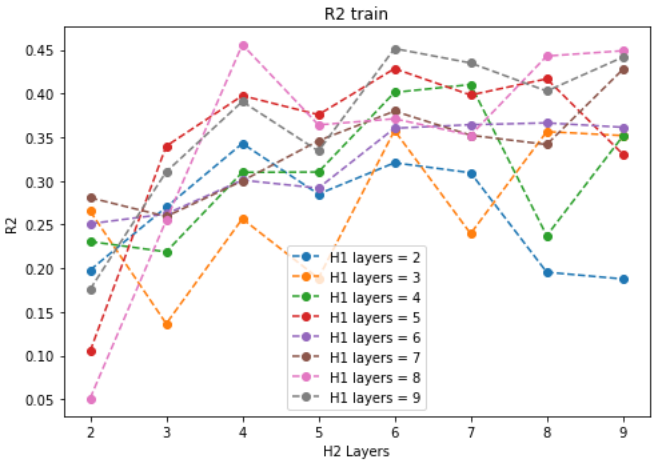

Figure 31: R2 training first half data set results

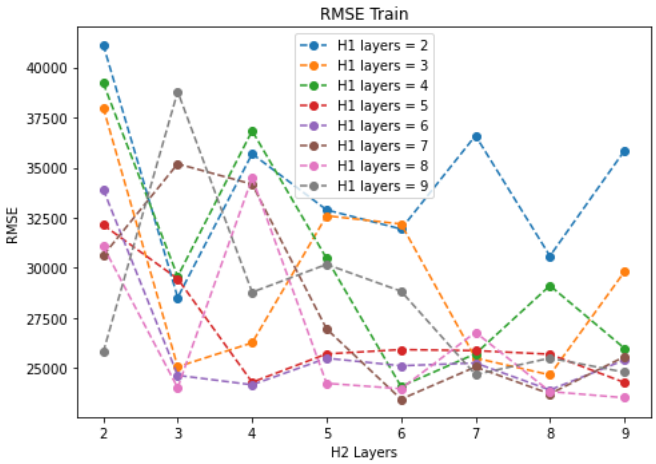

Figure 32: RMSE training second half data set results

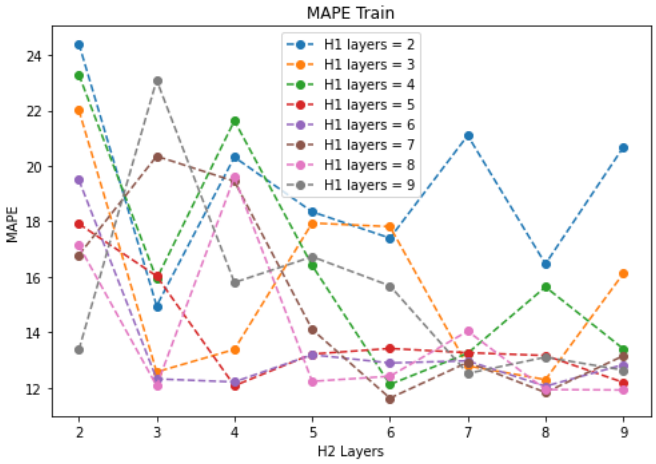

Figure 33: MAPE training second half data set results

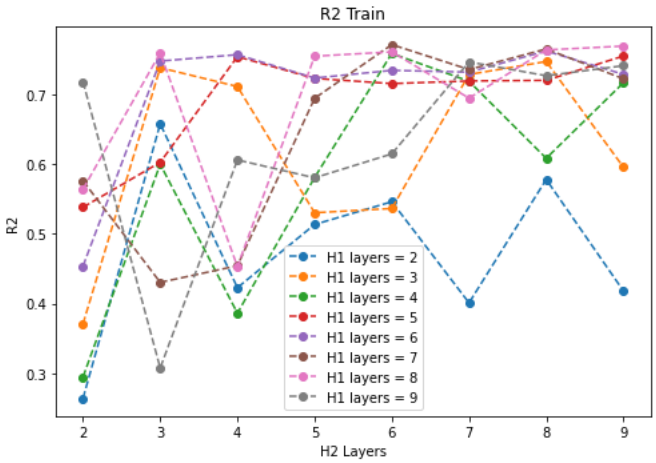

Figure 34: R2 train second half data set results