Paper pulp brightness prediction

DV2627 Advanced Machine Learning: Project

In collabortation with Södra

1st Oscar Cederlund
DVAMI19
Blekinge Institute of Technology
Karlskrona, Sweden
osce19@student.bth.se

I. Introduction

This project is done in collaboration with Södra (Södra Skogsägarna), a forestry cooperative that deals with three main business areas, timber goods, paper pulp and biofuel [1]. The assignment provided by Södra is to optimize the chemical consumption during the bleaching process of the paper pulp at Södra Cells pulp mill in Värö. These chemicals are essential to the bleaching process in order to break down and bleach the cellulose in the pulp

The issue at hand is that these chemicals are very expensive and there's a large potential to optimize the consumption during this process. This by minimizing the chemical consumption while maintaining the measured pulp brightness above the desired threshold. Machine learning is a great tool to investigate and potentially solve these problems as they are able to predict and forecast based on the parameters given from the data.

Today this process has been optimized with predictions by regression and tree models. While these do work for the task, these kinds of models are not able to take the time series aspect of the data into account. This could be improved on by using models that take the time series aspect into account.

In this project we have investigated the prediction ability of a regression model [2] compared to two other models, two neural networks and one VARMAX model.

II. Methodology

A. CRISP-DM

In order to conform to the CRISP-DM method and its iterative nature the project was treated much like one might treat the process of an agile software development project. This was done by setting up task and estimating their time of completion along with making sure that each step of the CRSIP-DM method was visited during the steps of each iteration. The iteration was not time limited but was treated as completed when the tasks set up were each completed and a meeting with project provider Södra had been conducted. This approach to CRISP-DM method provided the project with a great management tool which helped a lot with time, task, and iteration management. In total there were two iterations of the CRISP-DM cycle
that were completed. CRISP-DM iteration 1 features the following steps:

- Introductory business meeting
- Introductory data understanding
- Fundamental data preparation
- Fundamental model implementation
- Result evaluation

This iteration consisted primarily of getting the project fundamentals in place. The introductory business meeting served as a basis for the data understanding step of this iteration. Here Södra presented how the process works at the pulp mill, why each data feature in the data set is being collected and what purpose that feature fills in the bleaching process. During the introductory data understanding and fundamental data preparation steps the basics were set up to extract the wanted data from data set and was conducted in a quite exploratory fashion. For the fundamental model implementation the bare minimum models were implemented.

CRISP-DM iteration 2 features the following steps:

- Feedback business meeting
- Reworked input features
- Data cleaning
- Data prepossessing
- Model advancement

This iteration consisted of refining the work from the previous iteration. A feedback meeting with Södra supervisors was conducted and quite fruitful, with them sharing their domain experiences of data science, the dataset, pulp bleaching and the python environment that was provided by them. With their feedback the path forward was quite clear, even though some extra work had already been conducted because there was a lack of understanding on the python environment. Data cleaning became the next step as faults in the data set was revealed in the previous iteration and resulted in the inclusion of an extra data feature as much of the data quality was directly tied to that feature. After that the modeling step for advancing the models required some more data prepossessing to both

take in to head the data changes from the data cleaning and model updates.

B. Data handling

The data was provided in a relatively good quality, even though no prior work had been conducted it from our knowledge. The poor data quality that was found was often attributed when the pulp production had not been working correctly because of abnormal events, one instance of this was a maintenance stop of the pulp mill. During that period the data collected from various sensors was not representative of the data at working conditions of the mill, thus needed to be handled. The faults that the data would often contain were extreme values that would not be possible in practice, such as negative chemical feeds or values that are simply to large to be feasible. Multiple approaches to solves this problem were drafted among them were three major contenders:

1) Remove the none-representative data rows completely
2) Weighing the different data rows during model training.
3) Replacing the values with simulated ones.

The first and third options were not chosen as the thought was that they may hurt the data quality in some aspect of their own. Removing the faulty data rows would create a choppy data set and as the project was focused on time series analysis it was not clear how this would affect the data quality and model performance with incoherent data continuity. Replacing the values with simulated ones was not deemed feasible as they not be really representative of a functional production environment either. Thus the second option on weighing the faulty data rows using sample weights was chosen, this would allow data continuity while not allowing the faulty data to affect the model training to the same extent.

The data set contains data collected on a per minute frequency on a period that spans more then a year which, that combined with changes in the data not being very drastic made it suitable to aggregate the data rows from a minute basis to a 15 minute basis. This allows changes to be more easily detected and reduces the size of the data set, thus improving the ease of use on the data. This also became necessary for many of the models that were tested as it significantly reduced the training speed.

For the models there was a set of ten input features that had been deemed suitable for the task, these were selected early on in during the project and rarely changed afterwards. Five them were deemed essential and were included from the very start. The following ones very join in during development in CRSIP-DM cycle, often after meetings with Södra were they explained the bleaching process and what part each feature played in the process. In total there were 165 different data features within the data set, many of them were from different parts of the bleaching process.

C. Regression baseline

The regression model was supposed to serve as a baseline for the model evaluation and for that the a gradient boosting regressor model [2] from the sklearn package was chosen. The thought behind this decision was that this model with its ensemble nature and of this kind of model corresponds well to the current models in production at Södra

For the first iteration of the CRISP-DM cycle the the model was implemented in it's most basic state with rather arbitrary parameters. The performance results of this can be seen figure VIII. As expected the were results were quite poor, the prediction is quite stationary and not very responsive at all.

To solve the poor results in the second iteration CRSIP-DM additional data prepossessing was conducted as described in subsectionII-B. To complement this a grid search was conducted in order to tune hyper parameters of the model. The search space consisted of the following parameters:

- N_estimators: 100, 200, 300, 400, 500'
- Max_depth: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- Learning_rate: 0.001, 0.01, 0.1, 0.2, 0.3

The grid search also made use of a 5-fold cross validation test in order to provide an average result of each model. A higher fold had been more representative of each models performance but it was limited to five as the search space was already quite large compared to to computational resources that were available.

The resulting model had a significantly better performance compared to the model from the previous iteration as seen in figure VIII

D. Neural Networks

The neural network development was conducted in quite exploratory manner as we had limited domain knowledge, it was nonetheless a very rewarding experience. In the project proposal the first iteration of neural network was supposed a convolutions neural network. However when the development began on the neural network model started and the limited domain knowledge became apparent the scope was shrunk. Developed with the Tensorflow [3] package Keras [4] the model structure form the first CRISP-DM iteration was the following:

- Loss function: MeanSquaredError
- Optimizer: RMSprop(0.001)
- Model: Sequential
  - Dense(256, activation='relu', input_shape = (num_features))
  - Dense(256, activation='relu')
  - Dense(1)

The result from this model were found to greatly surpass the performance of the baseline model VIII of this iteration, as it was much more responsive towards the target in the test

set was able to more accurately predict the resulting changes based upon the test features.

During the second iteration it was decided to go back to the plan from the project proposal advance the model into a LSTM [5] model. This in order to hopefully incorporate the time series aspect and long term dependencies within the features in a greater extent. This type of model required a restructuring of the input data as to allow it to make use of data windows. A look back window of 72 was selected based upon the aggregation span, this makes the window span an entire day. For the parameters an exploratory search for parameters was made first and later transformed into another grid search.
The grid search space grew in to quite a large one as the there was a limited domain experience. The selected features for the search space were chosen based upon the exploratory testing that had been done beforehand. During the testing is was found that the activation functions and optimizers tended to make the largest difference upon the resulting model, thus they received more features. The batch sizes primarily kept large in order to reduce the training time for the models, one lower value as chosen with goal to investigate if the model would suffer because of the larger values:

- Learning rates: 0.001, 0.01
- Activation functions: softplus, elu, gelu
- Optimizers: Adam, RMSprop, Nadam
- Loss functions: MeanSquaredError, Huber
- LSTM units: 4, 64
- Batch size: 512, 256, 32

During development there were also two different models that were developed but it was found that the less complex the model were built, the more sensitive to changes in the data it became. Most likely because amount of units within the LSTM layers, it would later be found that we got better results by reducing them. It's prediction graph can be found in figure VIII. The first model that was built more complex had was structured according to the following:

- Loss function: MeanSquaredError
- Optimizer: RMSprop(0.001)
- Model: Sequential
  - LSTM(units=256, activation='gelu', return_sequences=True, input_shape=(look_back, num_classes))
  - Dropout(0.2)
  - LSTM(units=256, activation='gelu', return_sequences=True)
  - Dropout(0.2)
  - LSTM(units=256, activation='gelu', return_sequences=True)
  - Dropout(0.2)
  - Dense(1)

After noticing the issues with the previous model the development went back to the basics by reducing the model complexity. This would allow for a better understanding of how the model works and how the layers interact with each other. The Goal was to develop this further but as the time constraints of the project started to be noticeable it was settled for final model structure and the one that the grid search would be conducted upon(Model described contains the tuned parameters), the resulting model can be seen in figure VIII:

- Loss function: Huber
- Optimizer: Nadam(0.001)
- Model: Sequential
  - LSTM(units=64, activation='gelu', return_sequences=True,
  - Dense(1)

E. VARMAX

Two models were developed based upon the VARMAX package [6], one var and one varma model. Since there was some previous domain knowledge on univariate versions of these models it was believed that they would not be too difficult to build. After a large amount of time spent trying to get any useful predictions from them they were abandoned in favor of the neural network models. This is futher elaborated on in section VI.

For the models ACF and PACF were conducted upon the select features from the data set. Every feature had a very high had very high correlation values from the ACF test, meanwhile all of the features but two PACF plots showed only a single spike at their first lag. This shows that there might not be a strong trend or seasonality component within the data as might be expected. The two features that showed the opposite where the chemical feeds which makes sense. These graphs have not been included in the submission

With this knowledge the models were further adjusted with lags set accordingly in the second iteration but the results were not very different from the previous iteration. An example of this can be seen in figure VIII.

III. Results

The results showed that a robust regression is hard to beat, figure VIII. While the last iteration of the LSTM model was able to beat the gradient boosting model quite significantly error-wise, the predictions it made not quite representative of the data it's trying to predict as seen in figure VIII. Instead being more stationary and averaging out with a smaller value variance. This might be sign that the LSTM model might be over-fitted and thus might not be able to do these larger swings that the test data is doing. This may also be the result of the gird search that values low errors, thus promoting those models who are able to follow an average, What's even more interesting is the prediction capabilities of first iteration of a neural network model as seen in figure VIII, as this one is able follow the large value swings from the test data, thus showing that there's still potential in using models that utilizes a time series aspect.

IV. Analysis and further developments

To develop this project further there's definitely potential as the VARMAX models where never fully utilized and it would be very interesting to compare their results towards the ones from the neural networks as they might be able to keep the to avoid the averaging issue that the neural networks suffers, much like the gradient boosting ones. There's also potential in different model structures, the model from figure VIII is a very simple one and could be expanded on with convolutions components, merged with the LSTM or by simply tuning it's parameters.

V. Discussion

Despite the constant race against the clock the project manged to touch upon all segments of the CRISP-DM cycle but the deployment stage at least twice which became essential for understanding the process. The project had however major scope issues.

There were also problems with giving each model an equal amount of attention with the neural networks receiving more working hours compared to the baseline and VARMAX models. In total the following amount of time was spent on each model type across the two iterations:

- Gradient boosting: 11 hours
- Neural networks: 34 hours
- VARMAX: 23 hours
- Total project time: 155 hours

As seen above the time could have been distributed more evenly to give a better understanding of the effort to performance ratio on the models, as the Neural networks in total got more than three times as much work hours.

VI. Limitations

A. VARMAX and scope issues

As mentioned the methodology II-E the implementations of both the VAR and VARMAX model were unable provide any real predictions and along with having very long training times became a real hassle to work with. After a significant time was spent on these models without getting any proper results they were abandoned, which allowed for the scope to be reduced and more time could be spent on parts of the project that gave actual results, such as the neural network models and data processing. With the experience from adjusting the data set for the LSTM models it was considered to give the VARMAX models one last try. However as the time spent on the project started closing in the expected 146 hours it was realized that there would be any time left to spend upon it as there was still much left to evaluate on the other two models.

In the end the scope was most likely set too large during the project proposal. There are many probable reasons to why, two being a lack of hands on experience as there was only been one similar project conducted during previous courses at Blekinge technical institute with many other prior experiences being very contained in smaller assignments. A second reason was the varying group sizes in the course and the project proposal was not as tailored for a single student as it could have been.

B. SHAP

In the project proposal using SHAP as an evaluation tools for the models was set as a goal. Implementation of this was started as Södra had packaged some SHAP functions for evalutaion of regression models in their environment. These would how ever need to be adjusted in order to better incorporate the evaluation and support of the neural network models in SHAP, with project nearing it's end and little time left to be spent on the project this was cut from the final project.

C. LSTM grid search

The grid search for the LSTM model was never completed as it met an error when nearing the halfway mark. With about half of the models tested it was deemed enough as the project deadline and time limitations rapidly approached.

VII. Conclusion

To conclude the project we've just started to scratched the surface on topic and there's a lot more that could be done with a larger time budget. Despite this we were able to get feasible results which shows that there's potential in involving the time series aspect in pulp brightness prediction but in comparison the current baseline is able to perform well despite having less effort put into it that other models that perform seemingly worse.

References

[1] Södra, "About södra," https://www.sodra.com/sv/se/om-sodra/vara-verksamheter/, 2023, [Online; accessed 2023-02-08].

[2] Scikit-learn, "Scikit-learn - gradient boosting regressor," https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html, [Online; accessed 2023-03-24].

[3] Tensorflow, "Tensorflow - webpage," https://www.tensorflow.org/, [Online; accessed 2023-03-24].

[4] Keras, "Keras - about," https://keras.io/about/, [Online; accessed 2023-03-24].

[5] tutorialspoint, "Time series - lstm model," https://www.tutorialspoint.com/time_series/time_series_lstm_model.htm#, [Online; accessed 2023-02-10].

[6] Statsmodels, "Statsmodels - varmax," https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_varmax.html, [Online; accessed 2023-03-24].

VIII. Appendix

Fig. 1. Gradient boosting performance during iteration 1

Fig. 2. Gradient boosting performance during iteration 2
MAE: ≈ 1.633, MSE: ≈ 4.223

Fig. 3. Neural network performance during iteration 1

Fig. 4. LSTM model performance during iteration 2
MAE: ≈ 1.223, MSE: ≈ 2.501

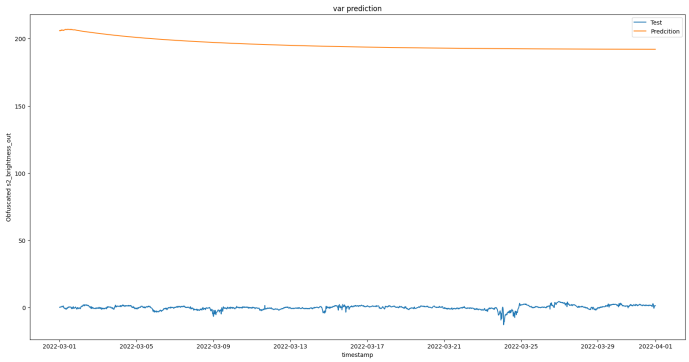

Fig. 5. VARMA model performance example