Capstone Project Summary – UK Triad Classification

Aim and Overview of the Project

The aim of this project is to build a predictive model that helps in correctly identifying which periods are going to be classified as Triads by the National Grid in the United Kingdom. There is a need for prediction because National Grid does not forecast the Triads in advance - they are only known post-February after the winter period has concluded. This is designed to encourage demand customers to avoid taking energy off the system during peak times, if possible, thereby reducing the need to build expensive infrastructure that all customers would need to pay for. This is where Machine Learning Algorithms come into play as it adds an analytical layer in decision making and accurately predicting which periods are going to be classified as **Triads**. The Triads are the three half-hour settlement periods of highest demand on the GB electricity transmission system between November and February (inclusive) each year, separated by at least ten clear days. National Grid uses the Triads to determine **TNUoS demand charges** for customers with **half hourly meters**. The scope of the project to begin was to predict the exact half-hour when the Triad was going to occur, however it evolved to predicting, if it was going to occur in the next 6 hours, mainly due to the extreme imbalanced nature of the data set and the fact that the class of Triads was extremely rare, if we use half-hourly data, there will only be 0.05% of the observations belonging to the triad class. Schneider Electric wants to predict these Triads, to provide service to their clients, so that the clients can avoid these extra charges.

Why am I trying to solve this problem?

I am trying to solve this problem, as it is an issue for our clients, and they can save the costs if we are able to predict when exactly those triads are going to occur. Our clients seek to receive an early alert for such period so they can reduce consumption during that time. Previously, alerts regarding potential peaks are being sent based on general trends, however there is no model created to tackle this problem. Creating an algorithm based on various important factors such as historic triads, weather, and historical demand, would allow us to be more accurate in terms of identifying these peaks, and improve the client satisfaction. Leading towards a potentially higher client retention and acquisition rate. As this would be an additional service that we will be providing to our clients. Some consultancies also provide a Triad forecasting service to notify their

customers when they believe a Triad is likely to occur. So rather than relying on those consultancies, we can directly provide our client accurate predictions, allowing them to save even more expenses.

Approach to tackle this problem and Result

The data that I used to create the model consisted of **Historical Demand**, **Triad History** and **Weather data (Temperatures, Dew Point)**. While the forecast will be done by using **Forecasted Demand** and **Weather Forecast Links,** which will contain both the temperature and dew points. First and foremost, I made sure that the quality of the data that I was using to solve this classification problem was up to the mark. I ensured that there were no missing or extreme values in the data set. The next step was to deep dive into the data and get to know it better and try to find out what the difference was between instances that were classified as triads and those which were not in the past. This was done by comparing means of both the classes and exploring their distributions in detail. Extensive feature engineering had to be performed as well, which included imputing dummy variables for blocks, after the data was aggregated to 6 hours. Including dummy variables for weekend and weekdays. Scaling the 3 main variables (Demand, Temperature, Dew Point) using the min-max normalization and then introducing 12 lags for the three key variables.

After all the data cleaning, feature engineering and exploratory data analysis, I finally moved on to the stage of running the models. The approach that I adopted was to run each model apart from linear, logistic, LASSO and Ridge, three times, each time adding additional features and variables to the models. This was to see how the model's performance changed after every iteration. I started off by running the most basic model, Linear Regression, as most of the times it alone is sufficient to provide us with appropriate results. After exploring Linear Regression, I ran some bit more complex models such as Logistic Regression (LOGIT), LASSO and RIDGE Score. Before finally moving to the more complex Machine Learning (ML) algorithms to see how well they perform for this classification problem that we have. All the models that I ran namely were, Linear Regression, Logistic Regression, LASSO, RIDGE, Random Forest Classifier, AdaBoost Classifier, CatBoost Classifier, Light GBM, XG Boost and Balanced Random Forest Classifier.

The model that performed the best was Balanced Random Forest Classifier. The metrics that were used to evaluate the performance of the models were AUC-ROC score, variable importance (F score) and number of False Negatives visualized through confusion matrix. This model gave a score of 0.95 in the final iteration and did not five any false negatives. This came as no surprise as Balanced Random Forest Classifier deals with the imbalance pretty well.

Limitations

The biggest Limitation that we faced in this whole project was the imbalanced nature of the data set. Due to this, the performance of several models was affected. Due to this limitation we also had to modify the scope of the project to predicting if the triad was going to occur in the next 6 hours rather than predicting the exact half hour triad period. Another limitation is that the predictions will be made on the predicted demands and predicted weather data, which even though is not that big of an issue due to the minimal deviation from the true values, however this could still be a factor towards misclassifying the Triads. Moreover, the ROC-AUC metric that we used to shortlist the models, does have some limitations as well. For instance, **Scale invariance is not always desirable.** For example, sometimes we really do need well calibrated probability outputs, and AUC won't tell us about that. Finally, there could have been more explanatory variables, such as the wind data, that could have helped in explaining the classification of triads even more. As when there is more wind, less electricity is demanded from the national grid.

Way Forward

Based on our series of classification models above we were able to determine that with our available imbalance data format we need to run a model which caters for this disproportionate ratio of observations. Our objective throughout this workflow is to cater for the minority class which is the occurrence of the triad. When dealing with imbalanced data we should not focus on the accuracy since machine learning algorithms are prone to optimize the overall accuracy. Therefore, use of confusion matrix, precision & recall score are considered to be a better indicator to evaluate our model performance. Currently our Balanced Random Forest model is preferred model based on the AUC and the confusion matrix results. Since the cost associated with false negative is higher than the false positive our aim was to minimize the occurrence of such misclassification. I also

will explore custom threshold with a self-supported loss function, which takes into account missing a triad is even more costly. However, we should not limit our analysis here but dig deeper into complex models like neural network. I was able to run a basic fully connected neural network as well, however not getting the desired results as it takes a lot of time and resources to fine tune the parameters and try different types of loaders to get the desired results. Stratified sampling or unsampling could have been two techniques to train the neural network with the data being imbalanced.

Neural network models are mostly trained with help of backpropagation of error algorithm. With the help of this approach, we are focusing on minimizing the misclassification error rather than squared error. Moreover, it allocates large error weightage to our minority class and smaller to our minority class making a balanced data distribution. This approach is also known as Cost sensitive Neural network. Furthermore, it adopts a regularization technique known as dropout layer which removes fixed number of entities in a network layer for each gradient step making it possible for the model to run on larger ensemble for smaller network. The process of tuning the parameters is tedious and requires time consuming trial and error procedures to reach to the optimize parameters. However, the performance of such neural networks can be used to further build more complex form of networks like fuzzy support vector machine which previously have been proven to work in the field of medical diagnostic where the data is heavily imbalanced, and misclassification are associated with higher cost.

Lessons Learned

- The quality of the data and feature engineering are extremely important while building any model (garbage in, garbage out)
- The importance of exploring and getting to know your data better, so that every decision taken is a conscious one
- The most complex models are not always the best ones
- How to deal with the imbalanced data set while creating classification models
- Learned in detail about the complex Machine learning algorithms and their evaluation metrics, especially in the case of classification problems
- The power of neural networks and their potential
- Reproducibility of your work should always be taken in account