Aleph Omega Analytics - DB Bus Regio Challenge

Time Series Classification with XGB

Max Weber
Department of INF
TH Rosenheim
Rosenheim
Max.Weber.97@gmx.de

Florian Weiss
Department of INF
TH Rosenheim
Rosenheim
Florian-Weiss-web@web.de

John Lyons
Department of INF
TH Rosenheim
Rosenheim
John.Lyons.93@hotmail.com

ABSTRACT

This work deals with the prediction of passenger volumes in the city of Passau in the context of the AI Cup Passau 2022. The aim of the work is to classify the 50 busiest stops into different passenger densities depending on the date and stop. For this purpose, a model based on the extreme gradient boosting algorithm (XGB) was developed. This model considers temporal characteristics and the significant dysbalance of each class. The data set was provided by the Deutsche Bahn and supplemented by external weather data. To evaluate the performance of the model, a F1 score was used and averaged over all class forecasts. This resulted in a F1 score of 0.4227.

1 Introduction

Forecasting passenger densities on bus routes enables more precise demand calculation of necessary sizes of transport means. Incorrect forecasting is associated with economic and environmental costs and thus waste of resources. A correct forecast allows to detect this waste of resources and can offer potential savings.

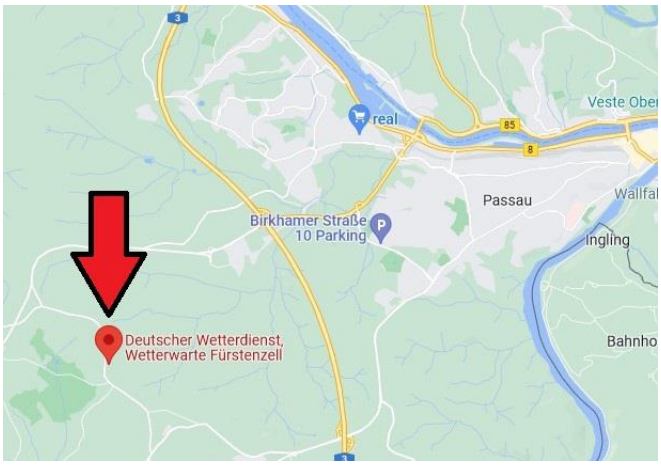
Deutsche Bahn provides data from the city of Passau over the period of one year for the "DB Regio Ticket Sales Prediction Challenge" competition. Two data sets were provided for this purpose. The first data set describes the passenger density of a regular bus line with regular stops. The second data set, on the other hand, describes the passenger density of an on-demand based bus service. In contrast to the regular service, the on-demand bus service only makes desired stops on a predefined route and omits stops that does not have online reservations.

These data sets contain for a given bus stop and date (day and hour) the volume of passengers divided into four classes. Classes 0, 1, 2 represents the number of passengers at a given time and end station. Class 3 represents three or more passengers. The target variable thus represents a multi-class problem.

2 Data Preprocessing

The data sets are divided into 12 folds. Each fold consists of three weeks of training and one week of prediction. These folds are continuous in time.

It can be assumed that the weather at a certain time of day has an influence on the behavior of passenger densities. For example, a rainy day may lead to a higher use of public transport, as an individual may prefer to use buses rather than a bicycle in this case. For this purpose, we extended both datasets with weather data. We use a weather station in Fürstenzell which is located close to Passau. A weather station in Passau is not active and therefore could not be used [1].



The data about Fürstenzell was provided publicly and free of charge by the German Weather Service in hourly basis on their website [2]. The weather data set includes various features like ambient temperature or air pressure.

We cleaned the weather data e.g., regarding duplicates and incorrect values. Furthermore, the time change in the weather station had to be considered.

Rebalancing of the data

The distributions of the classes were not identical. Class 0 has a relative frequency on the training data of about 80%, Class 1 about 9%, Class 2 about 4%, and Class 3 about 6%. Since the evaluation here is determined by the Mean F1 - Score for each class, it is critical that each class is given approximately equal prediction accuracy. The F1 - Score puts the Precision - Score and Recall in relation.

3 Model Development

For the model development we used the programming language R and the IDE RStudio. We used the packages "Dplyr" and "XGBoost".

It must be considered that for each forecast only the previous data before the forecast slot can be used. For example, the first fold has has three Weeks in hourly resolution which is equivalent to 504 data points per station for the training. The objective was to predict the following week (168 data points). From this and from the above-mentioned fact of unbalancing, a model with only a few trainable parameters should be chosen.

Therefore, we chose XGB because it performs well even if there a restricted amount of training data. Beside of that XGB is the state of in many classification tasks.

In the first approach, the model was to be made operational for the multiclass problem using a softmax function to output a probability for the forecast of each class. Then, the maximum over the probability of each class was selected as the forecast class. To get rid of the unbalancing problem, the training data for class 1, 2 and 3 were duplicated and added to the training data.

However, a much more successful approach, is to split the training data set per class and reformulate the problem as a one-vs-rest classification. This is done, for example, by transforming the problem into a One-hot encoding (e.g., class 3 or non-class 3) and then using a logit function to determine the probability. This is done four times for each class. The probabilities are cached. The advantage in this approach is that individual class probabilities can be edited to achieve higher accuracies. For example, probabilities can be subtracted from class 0 to penalize it against other class and focus on the other classes. Unbalancing was solved differently this time by giving weights to the gradient of the loss function, which is manipulate in the direction of the class which must be predicted in the one-vs-rest environment. During runtime, care was taken to output the error measure as AUC-ROC and AUC-precision-recall to be sure of the accuracy of the hyperparameters and adjust them accordingly.
The flow of the model was implemented to use all training data up to the fold to be predicted. Then, a one-vs-rest probability is determined for each class. After that, the probabilities are truncated by numerical values to penalize the classes according to their accuracy. Subsequently, the maximum of each class probability was selected, and this was used as the prediction class. This result was cached and later merged to the training dataset.

4 Results

It was shown during training that a small learning rate and high nround (trees) led to useful results. Gradient fitting with weights significantly improved the prediction accuracies with respect to the rarer classes in the training. High tree depth also helped. The test-train split was done by stratified sampling and showed how well each one-vs-rest probability predicted the class.
To assess the importance of each parameter in the posterior, we use Leo Breimann's importance metric for trees with fractional contribution of each feature to the model based on the total gain of this feature's splits [3]. Note that the XGB does not use entropy by default, but rather the so-called similarity score for logit-functions. The importance of each variable for the fully trained tree on the entire test dataset is given here below:

Shown for Class 0 (other classes give similar Outputs)

	Feature	Gain	Cover	Frequency
1	Zenit	1.631765e-01	1.661890e-01	9.917441e-02
2	hour_sin	8.125917e-02	1.541327e-01	5.295524e-02
3	Lufttemperatur	7.920637e-02	2.582335e-02	8.999877e-02
4	Windgeschwindigkeit	7.029640e-02	2.099259e-02	8.441233e-02
5	Globalstrahlung	6.733539e-02	1.864548e-02	7.361552e-02
6	relative_Feuchte	5.840605e-02	1.161639e-02	6.884746e-02
7	Windrichtung	5.530490e-02	1.146228e-02	6.641959e-02
8	week_sin	4.339258e-02	9.589281e-03	4.611200e-02
9	days_sin	3.363271e-02	3.468339e-02	2.410333e-02
10	Nummer	2.931796e-02	6.081706e-02	3.459228e-02
11	Sonnenscheindauer	2.615641e-02	3.984923e-03	2.912275e-02
12	days_cos	2.421429e-02	4.561642e-02	1.316038e-02
13	lat	2.015556e-02	1.704699e-02	2.944426e-02
14	long	1.905880e-02	1.215732e-02	2.721124e-02

5 Future Work

In our experiments we found that the prediction accuracy for the full training data for the on-demand dataset can be improved if each station is predicted individually compared to the simultaneous prediction of all stations. This may be because the XGB finds a significantly simpler split for the individual station than when all station data are used. This observation cannot be found with the Regular dataset.
It is possible to further improve the accuracies by using an additional model (stacking) to use the four forecast probabilities at the end and to train a model which considers the errors for a certain error decision. If the uncertainty is high (low probability for the four classes), a rejection can then be made to incorporate expert knowledge for individual decisions or to access another model for these exceptional cases (e.g., Seasonpredictor).
It was also shown that larger amounts of data led to better results. It can be assumed here that the months have a significant influence on the model, here especially the zenit indicates a seasonal influence on the passenger flow. Thus, it can be assumed that the

winter months have a different influence on passenger densities than the summer months. It is in this context that relevant features can be added to the XGB to better capture temporal dependencies. In forecasting bus density, it is conceivable that there exists a temporal dependency between the different passenger flows of the bus system. Weber showed in his work examining time series that tree-based methods produced similar or better results compared to classical time series methods. The modification of the XGB into a time series model could be implemented by introducing lag operators.

It is possible to use unsupervised learning algorithms (e.g., k-means clusters) to generate nontrivial divisions of the dataset into specific subgroups and pass these to the XGB. It can be argued that a division into end stations, e.g., in end stations with a higher population density and thus higher use of public transport compared to a lower population density, leads to improvements.

REFERENCES

[1] Bild Wetterstation. (o. D.). Google Maps. Abgerufen am 10. Juli 2022, von https://www.google.de/maps/place/Deutscher+Wetterdienst,+Wetterwarte+F%C3%BCrstenzell/@48.5538694,13.3466002,13z/data=!4m5!3m4!1s0x477459d36266b8bf:0xcb423f78a1103146!8m2!3d48.5452967!4d13.3535278

[2] N/A, N. A. (o. D.). Wetterdaten ID Fürstenzell: 05856. Deutscher Wetterdienst. Abgerufen am 10. Juli 2022, von https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/hourly/

[3] Tianqi Chen Et. Al. (2022, 16. April). XGBoost-Documentation. XGB-Boost-xgb.importance. Abgerufen am 10. Juli 2022, von https://cran.r-project.org/web/packages/xgboost/

[4] Weber, M. (2022). Anwendung unterschiedlicher Modellierungstechnik des maschinellen/statistischen Lernen zur Bereinigung der energetischen in-situ Messdaten bewohnter Gebäude unter Berücksichtigung der individuellen Nutzung und des Klimas. TH Rosenheim.