Polnish Client

Identification of insolvent clients

Agenda

- Situation
- Solution
- Results
- Next Steps

Situation

Small number of insolvencies have a potentially high impact on costs

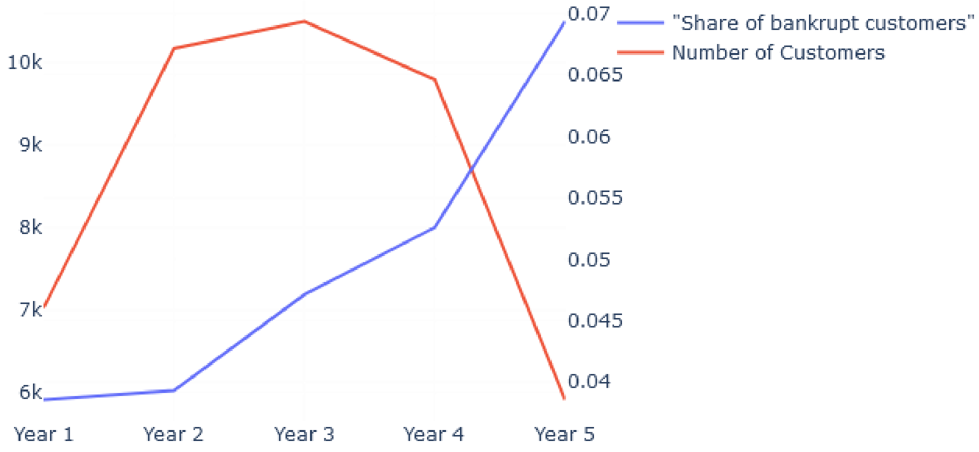
Problems

Unidentified insolvencies lead to a higher rate of compensations than planned

Wrongly identified bankruptcies cause higher insurance premiums which may lead to higher contract terminations

Un- or bad estimated insolvency rates require higher equity shares. Therefore minimize the ability to leverage the available capital.

Data – Starting point



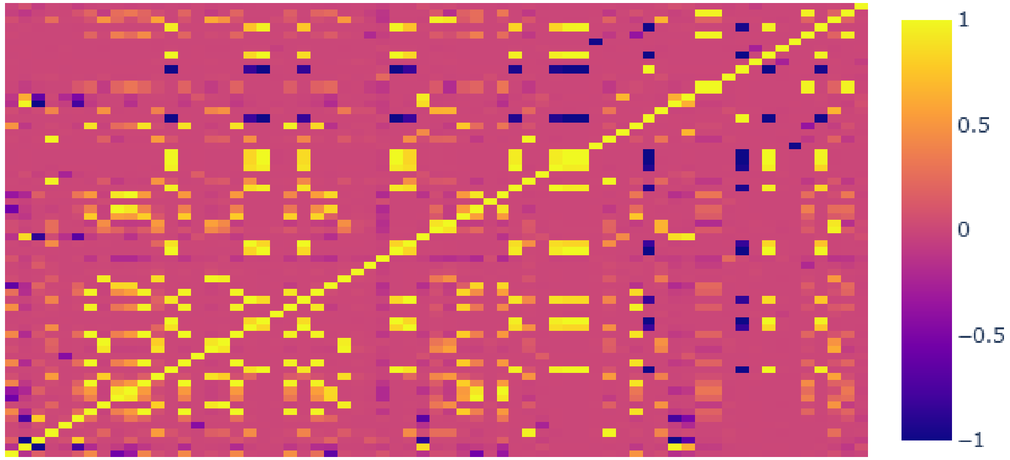
- Historical Data of survived and insolvent companies after one to five years.
- Data seems to be consistent since the share of bankrupt companies increases with time

Large deviations in the prediction of bankruptcy causes several negative cost and sales effects.

Situation

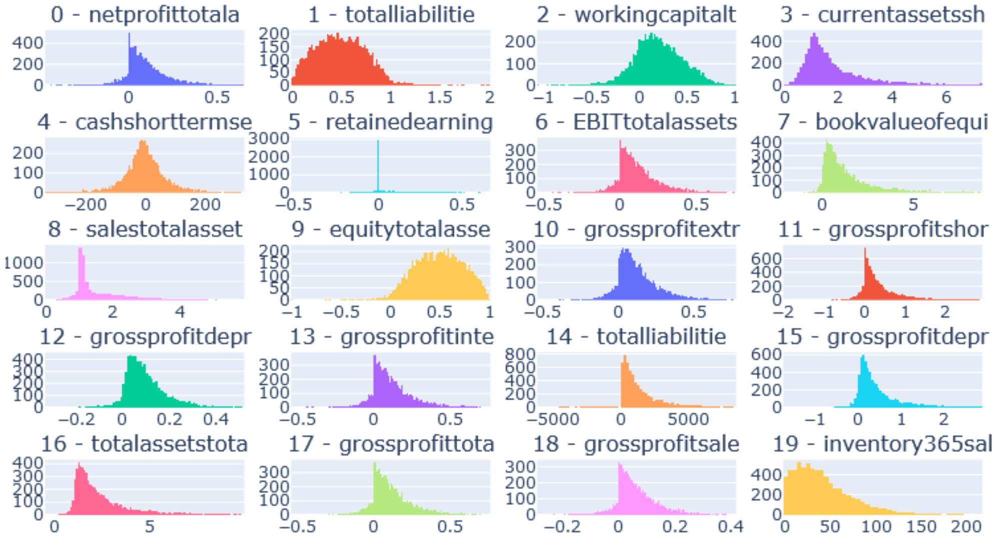
64 financial kpis are available and mostly complete (1.24% missing data in total)

Correlations



While strong corrlation between some financial kpis are usual they may decrease the performance of the Classification Model.

Distributions



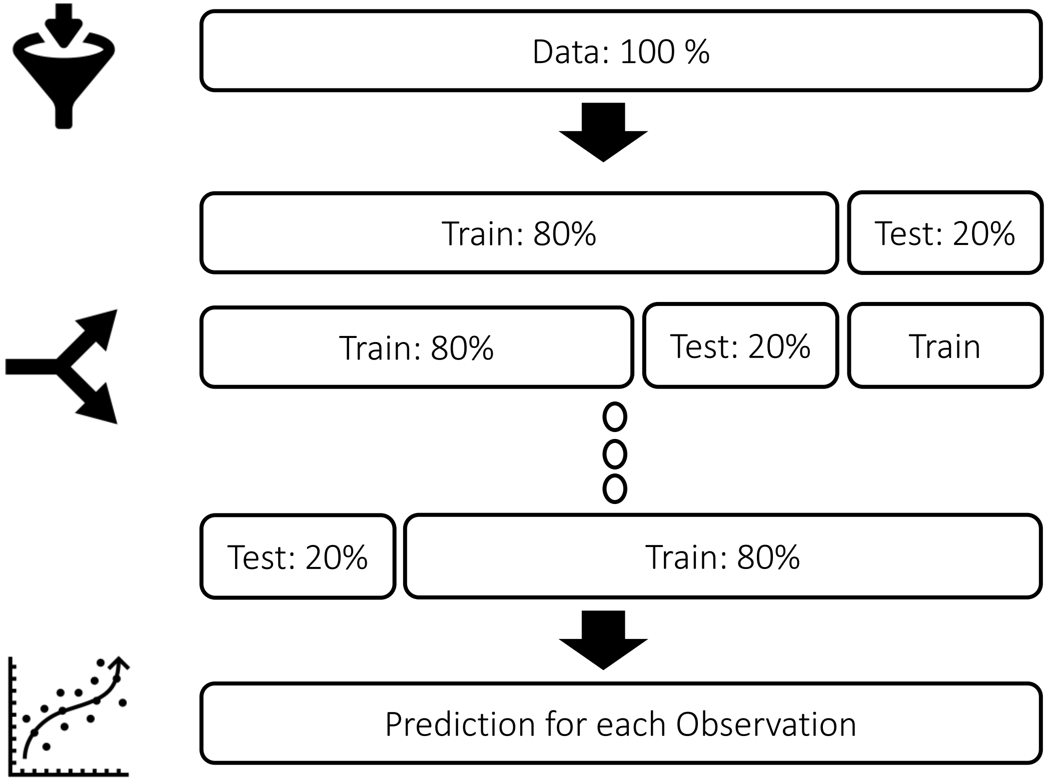
Some Features have a non normal distribution which may aggravate the training of the classification model. Since this is not necessary it is an optional preprocessing step.

Correlations should be eleminated and features if possible transformed to normal distribution

Solution

To maximize the generalization of our models we use cross validation

Validation framework



Data: 100 %

Train: 80% | Test: 20%

Train: 80% | Test: 20% | Train

Test: 20% | Train: 80%

Prediction for each Observation

Algorithms

Logistic Regression

- Used for baseline prediction
- Linear model that is easy to interpret since we have one coefficient per feature
- Not prone for overfitting cause of the linearity
- Needs several preprocessing steps

Extreme Gradient Boosting

- Captures non-linear relations
- Ensemble of Trees makes it harder to interpret
- Trees are able to fit to any function. Therfore, it´s prone to overfitting
- Does not need any preprocessing

Solution

The most important choice is the evaluation function. Especially, when handling imbalanced datasets.

Evaluation

- We deal with 5-7% of insolvencies for each year. Therefore, a model that predicts no insolvencies already achieves an accuracy of 93-95%.
- The choice of the evaluation measure dependes on the cost that is caused by false predictions
- Its far more expensive to pay compensation than to lose a customer due to high insurance premiums. Therefore **Recall** is a good base measurement for this case in combination with a high **Area-under-curve**
- Furthermore, we created the measure „Weigthed Accuracy“ which weights the positive cases by a numerical factor.

Results – Best Model

Logistic Regression

- max_iter: 1000
- Preprocessor:
  - PCA with 99,9% variance
  - Downsampling
  - MeanReplacement
  - Standardizing

- AUC: 0.765
- WA: 0.713

Extreme Gradient Boosting

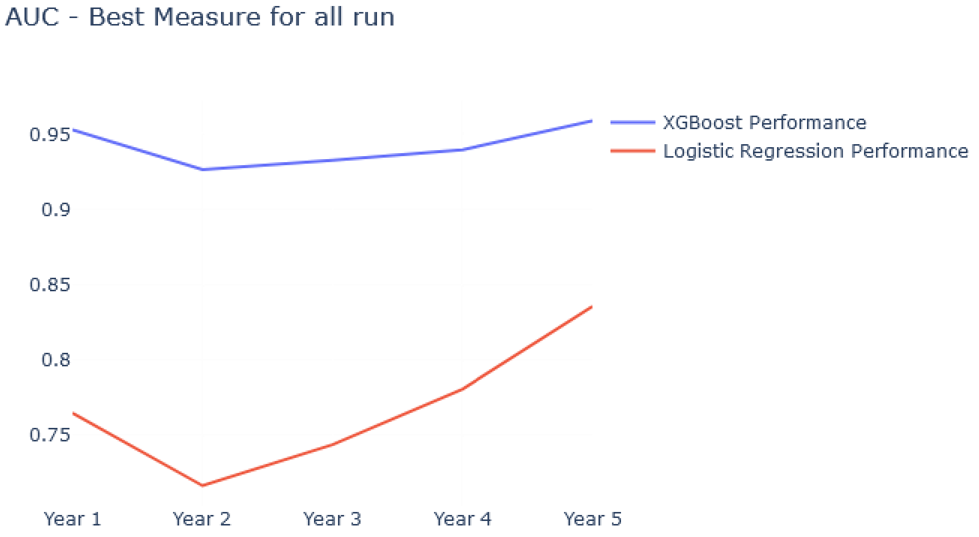
- n_rounds: 50
- lambda (L2): 5
- max_depth: 4
- Column- and rowsample: 0.7

- AUC: 0.952
- WA: 0.881

Solution

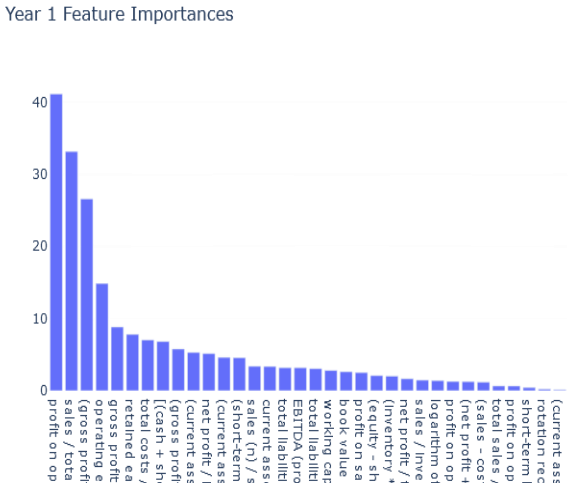
There are some single Features that have a strong impact on the model performance

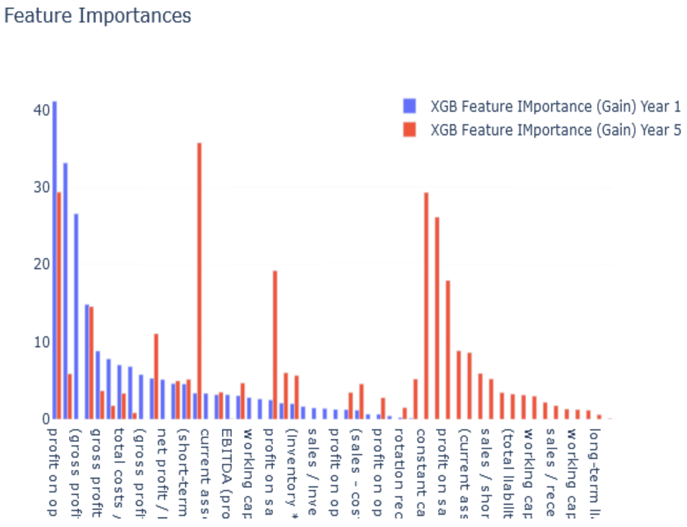
Evaluation



- XGBoost outperforms Logistic Regression by more than 20%
- Both have a similar pattern

Feature Importance





- Most of the information is covered by a small amount of features
- Important features for Year 1 differ from the features of Year 5 (right figure)

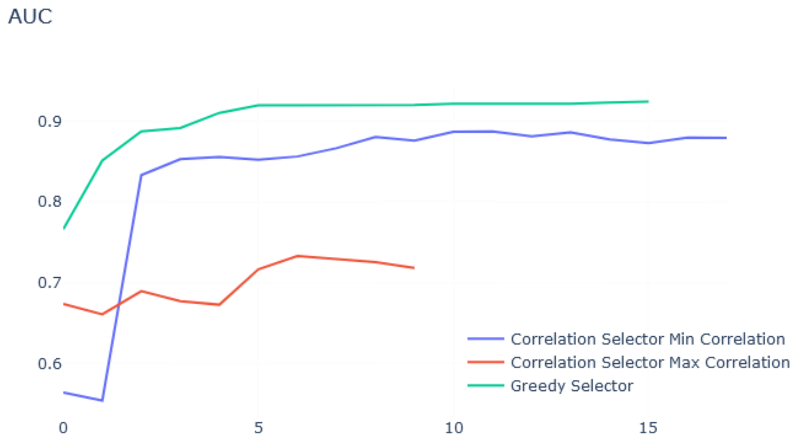
There is potential in Feature Selecting since we a small amount of variables with a high information value and a lot of values with high correlations to each other

Solution

The number of required features can be significantly reduced with only a small decrease in performance

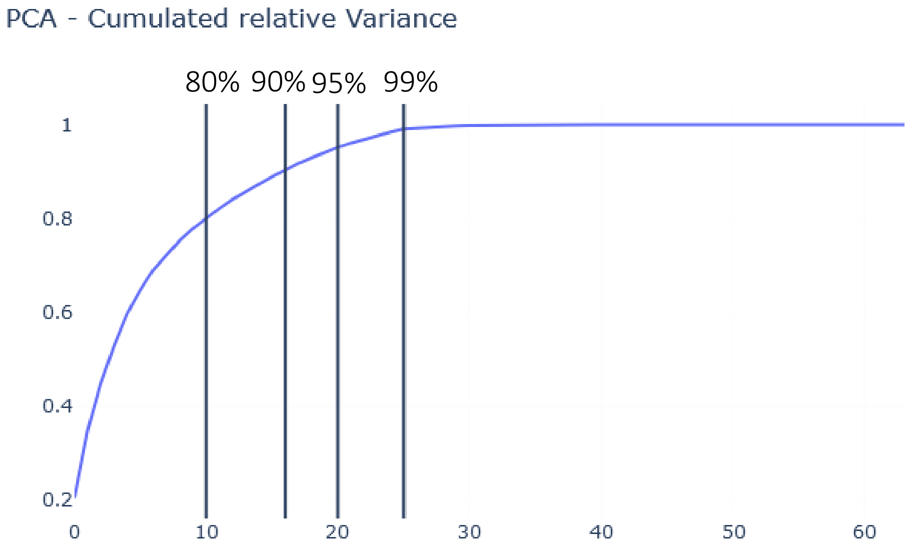
Feature Selector

- We implemented two methods which iteratively reduce the amount of features
- “Correlation Selection” selects features based on their sum of squared correlation to the remaining features.
- “Greedy Selection” chooses iteratively the feature that improves the measure the most



While both Selector choose less than 25% of all variables till they abort the search, the greedy selector achieves the best measure with an AUC of 0.92.

With a decrease from 0.95 to 0.92 the AUC barely changes while we are able to save costs for gathering all features.



Next Steps – Future Projects

Knowing which company is likely default is one thing, knowing how much we can recover is another

This Project:

- This project should be put into production to be able to raise the identified cost savings
- As companies are no homogenous mass it isn’t enough to just predict which company is likely to default. We can build on this project to predict the recovery rate of the potential bankrupt companies. Some companies may be able to cover 90% of their liabilities after insolvency while other companies have too much liabilities and are barely able to cover more than 10%

New Projects:

- Cross- und Upselling. Can we raise sales by identifying customers that probably need other insurances as well.
- Fraud detection: All transactions and compensation requests can be preclassified whether they are likely to be fraudulent
- Churn-Classification: Prevent the termination of contracts

Planned – but not implementd

Some planned features could have improved the measure or increase the generality of the results

- Normalizing of features
- Additional Test split that purpose is the last evaluation step to prove how well our model is able to generalize
- Feature Selection: Costs of feature gathering as an additional regularization term of the cost function.
- Run Greedy Selector for all Years
- DALEX Plots for Evaluation of Feature Contribution to a single prediction
- Watch out for potential causalities

Excourse: Weighted Accuracy

Weighted accuracy captures cost differences between the positive and negative class

$$Weighted\ Accuracy = \frac{\sum((y \wedge 1) \wedge (\hat{y} \wedge 1)) \ * \ \alpha + \sum((y \wedge 0) \wedge (\hat{y} \wedge 0))}{\sum(y \wedge 1) * \alpha + (y \wedge 0)}$$

$$with\ \alpha = \frac{cost\ not\ identified\ positive\ case}{cost\ not\ identified\ negative\ case}$$

- If alpha goes against infinity the weighted accuracy will only the accuracy of matching all positive cases
- If alpha goes against negative infinity the weighted accuracy will only be the accuracy of matching all negative cases