IronHack 2024

Bankruptcy Prediction: Supervised Machine Learning Approach

Beata Taudul, Karina Zalavska

Agenda

01 Project Objectives

02 Data Overview and EDA

03 Models Investigation

04 Data Preprocessing

05 Models Implementation

06 Models Comparison Analysis

07 Final Conclusions

Project Objectives

Perform analysis of data collected from the Taiwan Economic Journal for the years 1999 to 2009. The data consists of information about the performance and stability of companies.

Companies bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange.

The goal is to evaluate a predictive models for bankruptcy classification.

Data Overview

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	0.796887	0.808809
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	0.797380	0.809301
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	0.796403	0.808388
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	0.796967	0.808966
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	0.797366	0.809304

6819 rows * 96 columns

Datatypes: float64 (int64)

Exploratory Data Analysis

DATA CLEANING

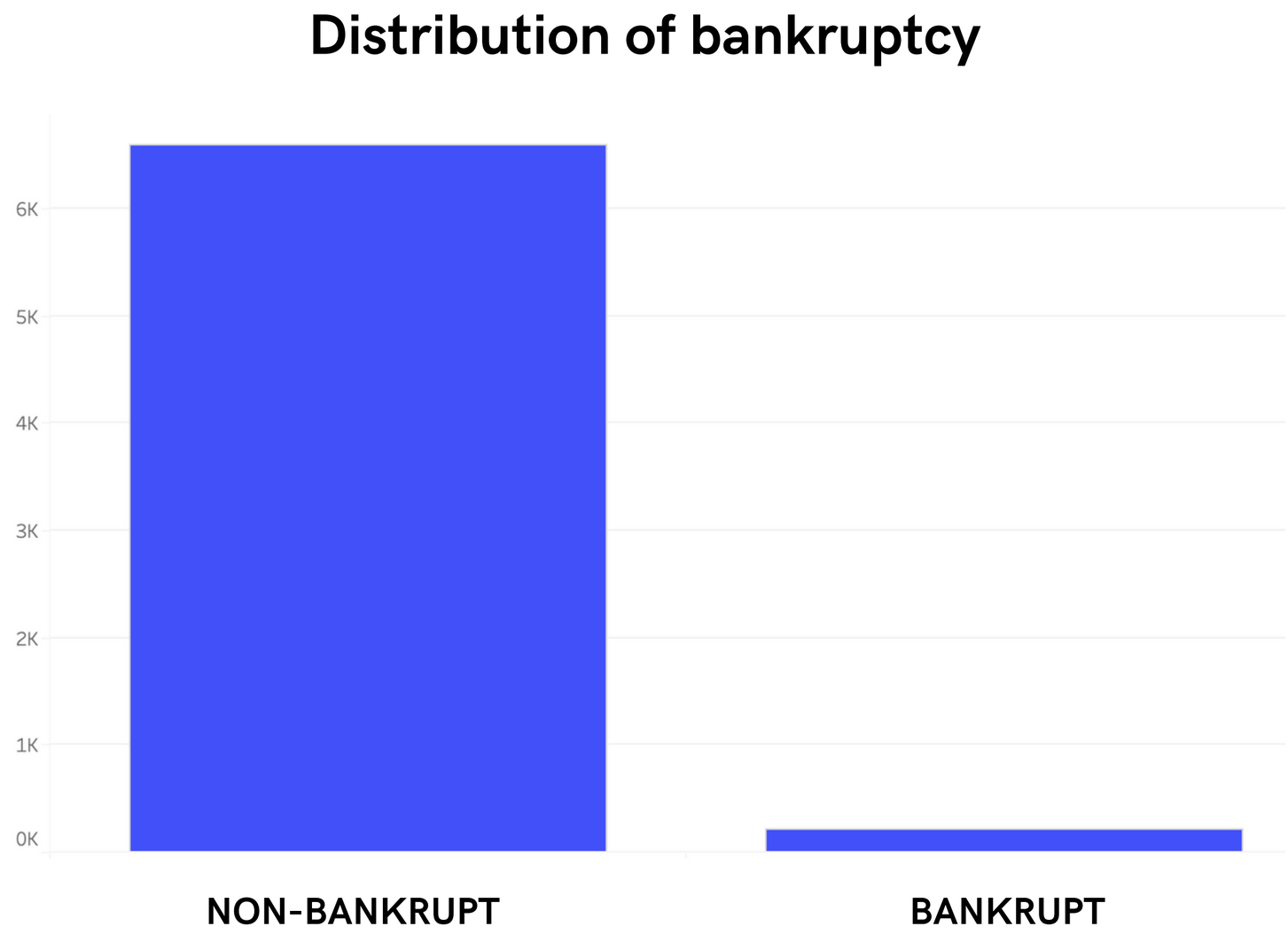
- Checked for missing values and duplicates
- Cleaned column names

```
'Bankrupt?'

' ROA(A) before interest and % after tax',
' ROA(B) before interest and depreciation after tax',
' Operating Gross Margin', ' Realized Sales Gross Margin',
' Operating Profit Rate', ' Pre-tax net Interest Rate',
```

- Dropped the columns: 'Net Income Flag', 'Liability-Assets Flag’
- Checked for outliers

Exploratory Data Analysis

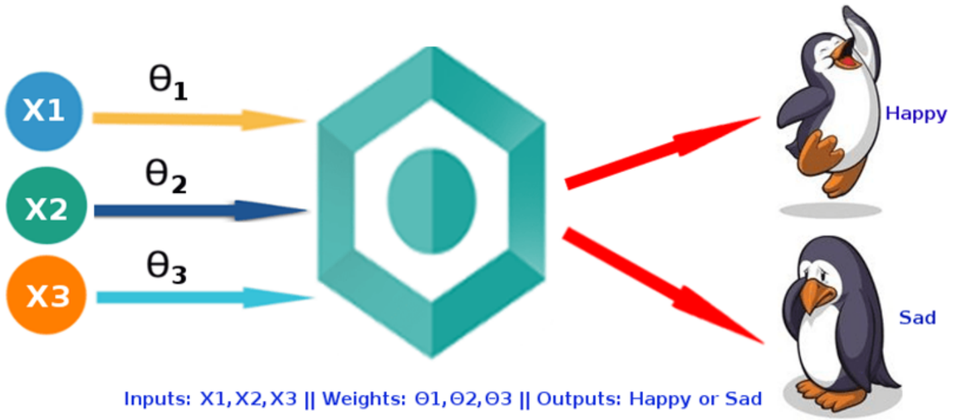


MODELS

- Logistic Regression
- XGBoost
- HistGradientBoostingClassifier
- Calibration

Logistic Regression

Addresses binary classification problems.
It takes input variables and transforms them into probability between 0 and 1

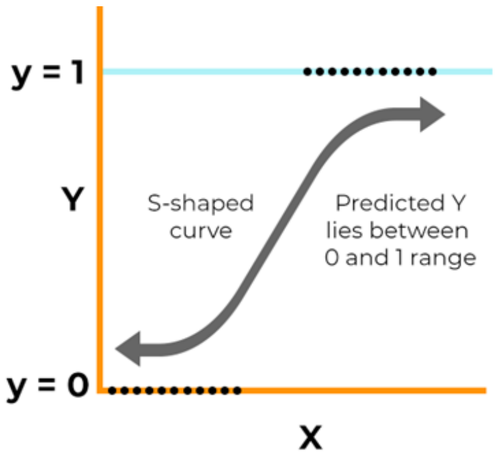


Assumptions

- No outliers
- No multicollinearity
- Independence of observations in dataset
- Larger samples sizes ensure stability
- Linearity (linear relathionship between the log-odds of target variable and the features)

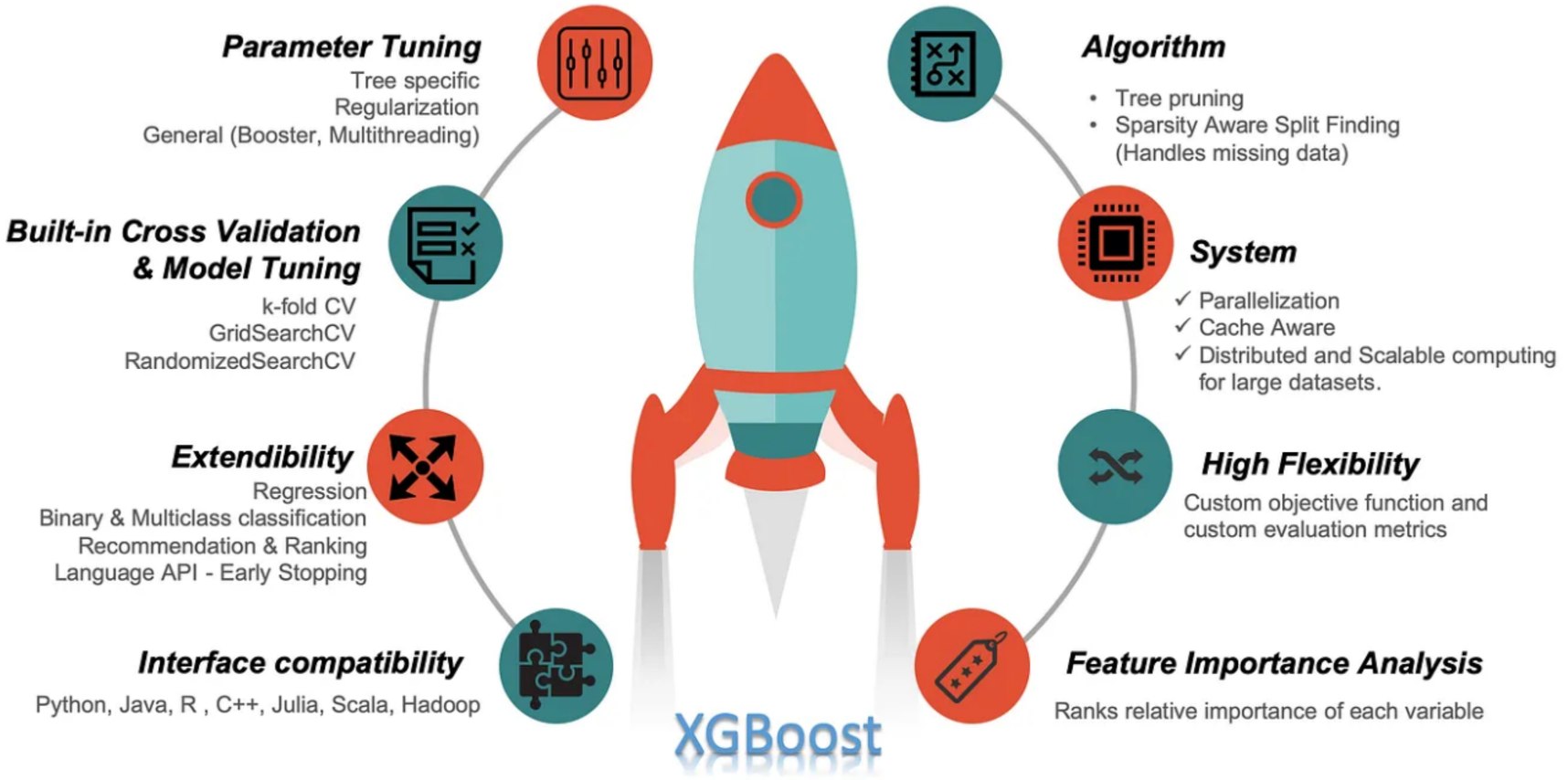
Logistic function to model the probability

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$



XGBoost Model

Combines multiple weak predictive models, typically decision trees, to create a stronger final model. Each new tree corrects the errors of the previous ones.



XGBoost Model

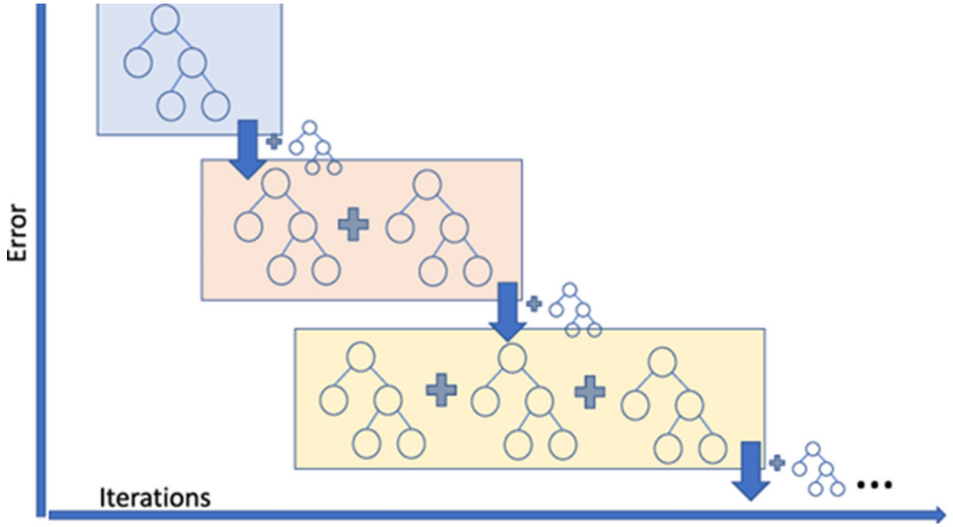
- An extension of the **gradient boosting algorithm**, combines the predictions of multiple weak learners (typically **decision trees**) to create a strong predictive model
- Use L1 (Lasso) and L2 (Ridge) regularization techniques to **prevent overfitting**
- Handle **missing data**
- Supports **parallel and distributed computing**
- Provides **feature importance scores** (identify the most influential variables in the model)

HistGradientBoostingClassifier

Histogram-based Gradient Boosting Classification Tree

Main elements:

- A loss function to be optimized.
- A weak learner to make predictions.
- An additive model to add weak learners to minimize the loss function.
- Employs histograms instead of using individual data points to construct decision trees



Advantages:

- It is a generalised algorithm which works for any differentiable loss function
- It often provides predictive scores that are far better than other algorithms
- It can handle missing data

Disadvantages:

- Sensitive to outliers.
- It is prone to overfit if number of trees is too large.

Calibration

CalibratedClassifierCV

- **adjusting the output** of a model to better align with the true probabilities of the events it predicts

- applicable to any algorithm that outputs probability estimates, such as logistic regression, support vector machines, or gradient boosting classifiers

Two main techniques

- **Platt Scaling (Logistic Regression Calibration)**
- **Isotonic Regression**

Other possibilities:

- **Beta Calibration**
- **Histogram Binning**
- **Platt Scaling with Regularization**
- **Temperature Scaling**
- **Dirichlet Calibration**

Data Preprocessing

How model performers initially? How balancing data changes output?

Start with full data set : 93 columns and imbalance between target values

Next: balance target values (**upscaling**)

Bankrupt	Train	Test
0	5285	1314
1	170	50

Can we get the same output but with less features?

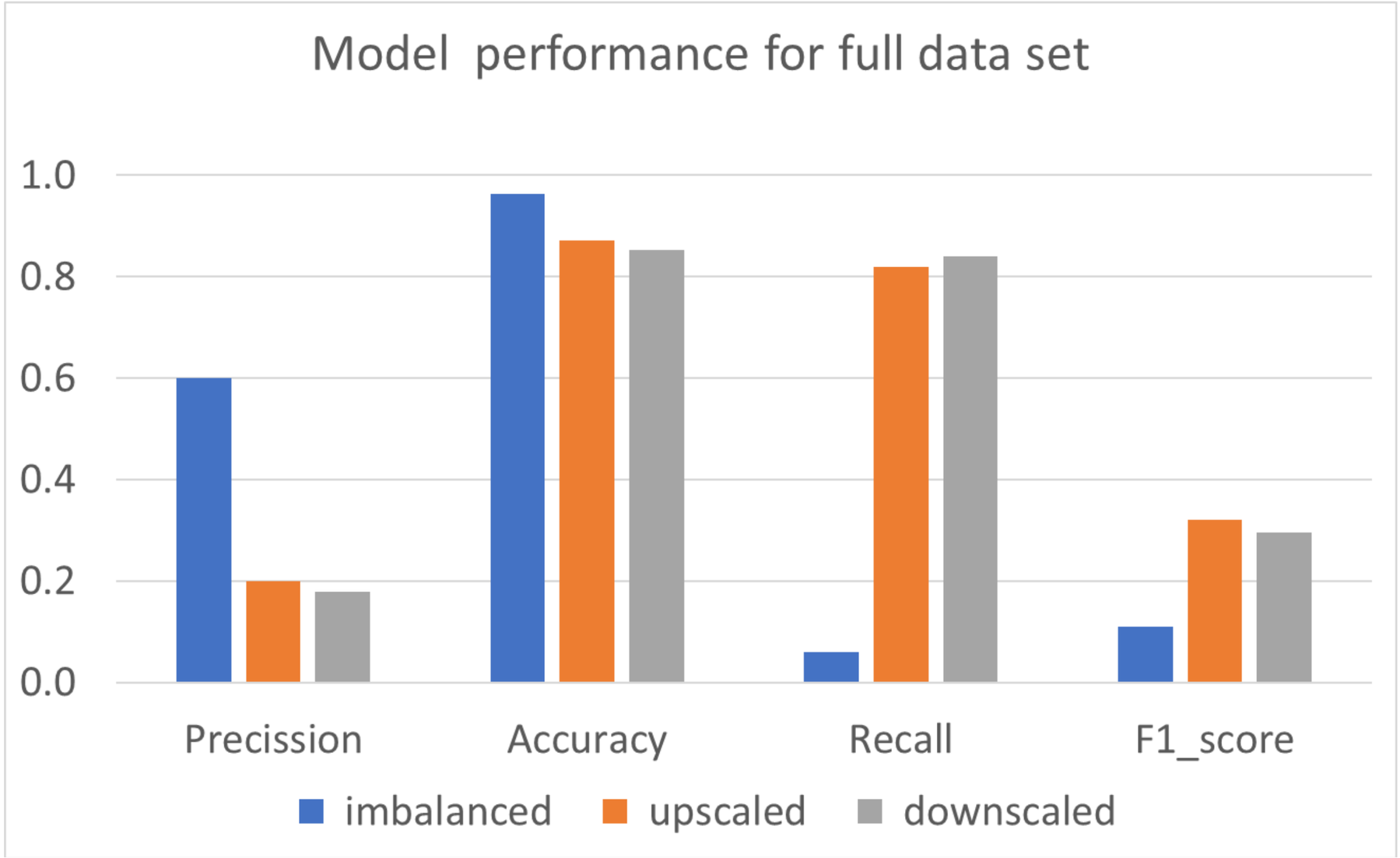
Remove features based on low variance, colinearity, PCA, RFE, KBest method….

KBest method (score_func=f_classif, k=30)

Remove collinearity - **18 columns left**

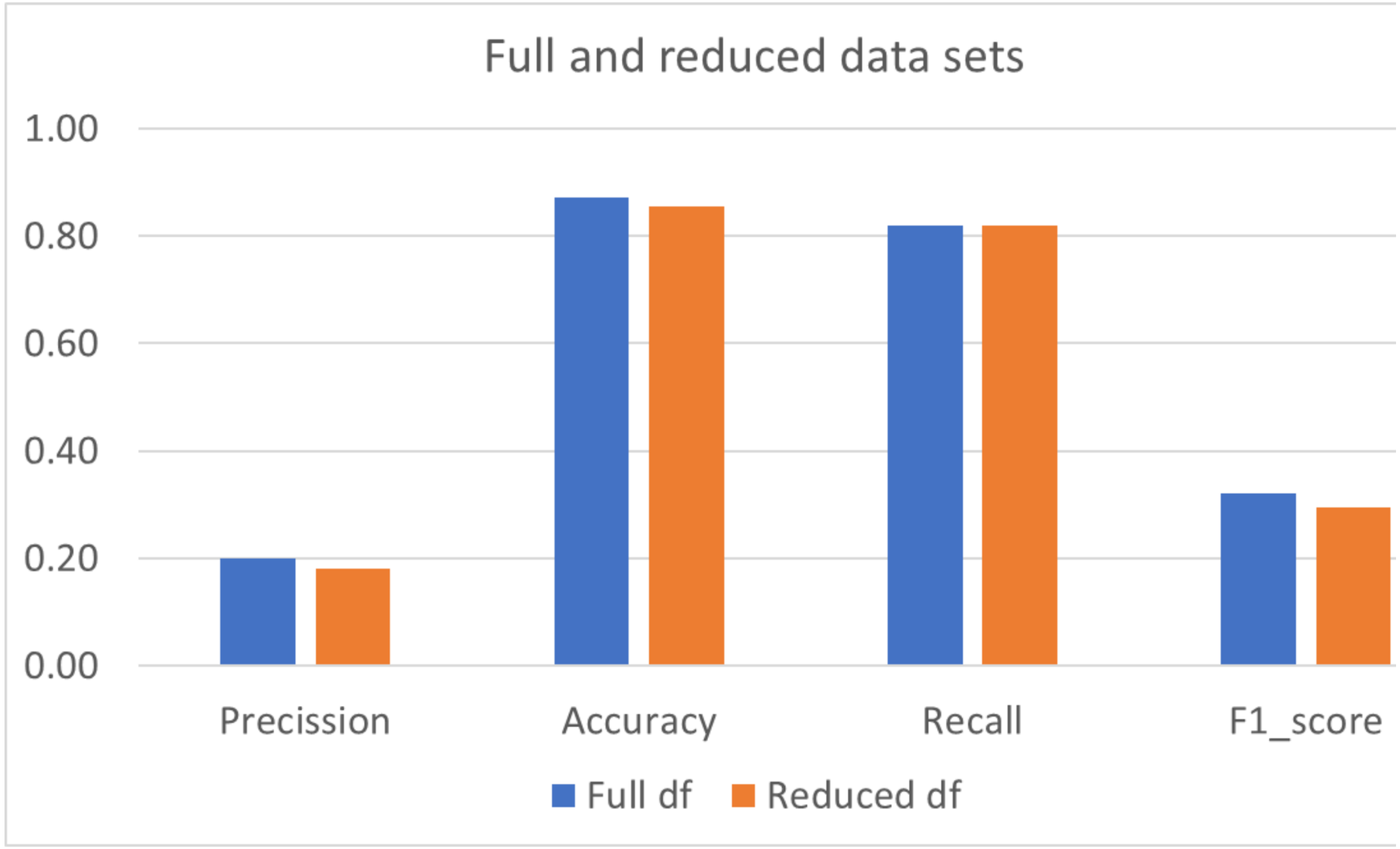
Finally: upscaled data and balance target values

LogisticRegression for full data set



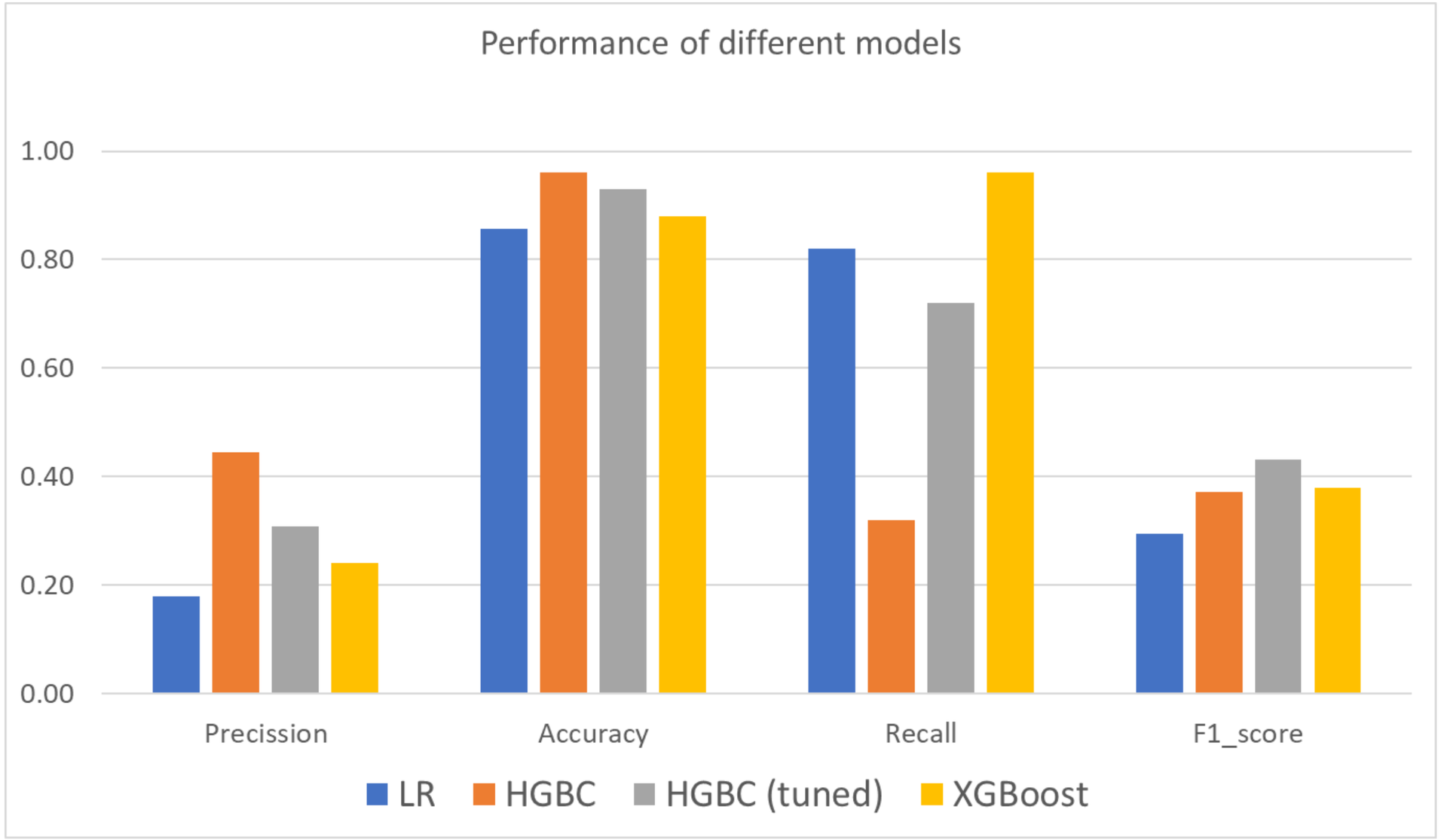
LogisticRegression for full and reduced data set

with balanced target values



Comparision of models

Reduced data set with balanced target values



Confusion matrix for models

	Imbalanced		Upscaled		Upscaled+calibration	
LR - reduced	**TN**	**FP**	**TN**	**FP**	**TN**	**FP**
	1313	1	1127	187	1128	186
	47	3	9	41	9	41
	FN	**TP**	**FN**	**TP**	FN	TP
HistGradient	**TN**	**FP**	**TN**	**FP**	**TN**	**FP**
	1310	4	1294	20	1311	3
	43	7	34	16	41	9
	FN	**TP**	**FN**	**TP**	**FN**	**TP**
XGBoost	**TN**	**FP**	**TN**	**FP**		
	1313	1	1162	152		
	41	9	3	48		
	FN	**TP**	**FN**	**TP**		

Confusion matrix for models

	Imbalanced		Upscaled		Tuned	
LR - reduced	TN	FP	TN	FP		
	1313	1	1127	187		
	47	3	9	41		
	FN	TP	FN	TP		
HistGradient	TN	FP	TN	FP	TN	FP
	1310	4	1294	20	1233	81
	43	7	34	16	14	36
	FN	TP	FN	TP	FN	TP
XGBoost	TN	FP	TN	FP		
	1313	1	1162	152		
	41	9	3	48		
	FN	TP	FN	TP		

Thank you!