Prediction with Machine Learning for Economists 2021/22 Fall
Assignment 3 Summary Report

Target: Design the target (fast growth), then build models to predict **fast growth** of firms.
Data: bisnode-firms data. It is the detailed company data from a middle-sized country in the European Union. All registered companies in 2005-2016 in three selected industries (auto manufacturing, equipment manufacturing, hotels and restaurants).
Target variable: **fast growth (design by ourselves),** whether the company survives currently and current sales are 1.2 times the company's sales two years ago.
Predictors: sales, total_assets_bs, inc_bef_tax, curr_assets and others
Model selection: Logit, Random Forest, GBM
Data clean: Since the data is missing many values in some attributes, I need to delete the columns; for the more important attributes that cannot be deleted, delete the rows with missing values for that attribute.

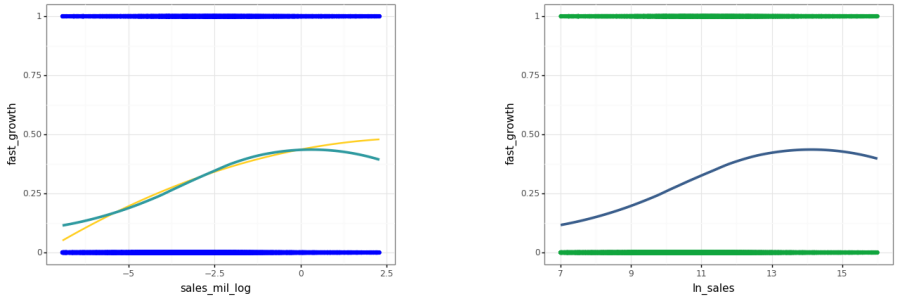

Figure1 Relation between sales_mil_log/ln_sales and fast_growth (target)

sales_mil_log is log(sales / 1000000), and ln_sales is log(sales). From the above two pictures, we can see that as the independent variable increases (sales will also increase), the target value fast_growth also slowly increases from 0 to around 0.5. It can be concluded that when sales increase, the possibility that the company does not grow rapidly decreases.

	Number of predictors	CV RMSE	CV AUC
X1	9.0	0.454952	0.617490
X2	16.0	0.454116	0.625344
X3	31.0	0.428237	0.737948
X4	74.0	0.428712	0.736242
X5	141.0	0.426765	0.741170

Table1 The results of the Logit model on the training set under different number of features

In the absence of our self-designed loss function, according to the table, it can be seen that when more features are used to predict the target value, both RMSE and AUC perform better. This is also in line with common sense. To a certain extent, it is often more accurate to use more angles (features) to analyze the results.

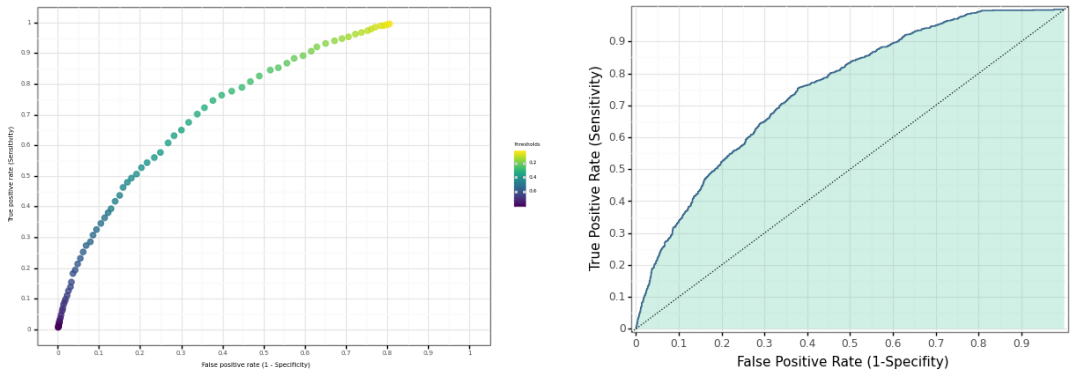

Figure2 ROC curve under different decision threshold

Under normal circumstances, when making probabilistic decisions, we will regard 0.5 as a watershed. When there is a certainty of 0.8 (probability), we will predict it as 1 (positive class), otherwise it will be 0 (negative class). In the above ROC curve, we need to obtain a threshold (not always 0.5) to make the False positive rate as small as possible and the True positive rate as large as possible.

	Predicted no fast_growth	Predicted fast_growth
Actul no fast_growth	2327	299
Actual fast_growth	733	418

	Predicted no fast_growth	Predicted fast_growth
Actul no fast_growth	2327	299
Actual fast_growth	733	418

	Predicted no fast_growth	Predicted fast_growth
Actul no fast_growth	1637	989
Actual fast_growth	293	858

Table2 From top to bottom, from left to right, the confusion matrix on the test set, the confusion matrix when the decision threshold is 0.5, and the confusion matrix when the decision threshold is the average value (0.31) of all the predicted probabilities of the test set.

It can be seen from the three tables that the results of the model classification are consistent with the prediction results when the threshold is 0.5. Due to too many negative classes, the average prediction probability of the test set is 0.31, and the correct rate obtained by the confusion matrix has dropped compared to before.

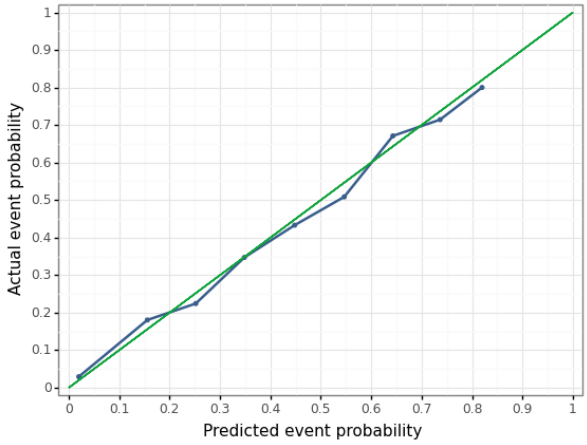

Figure3 Calibration curve between actual event probability and predicted event probability

Loss Function: In this experiment, the number of negative cases is more than the number of positive cases, so the penalty factor for judging a positive class as a negative class is greater than the cost of misjudging a negative class as a positive class. So the cost = FN/FP = 5/1, prevelance = y_train.sum()/len(y_train).

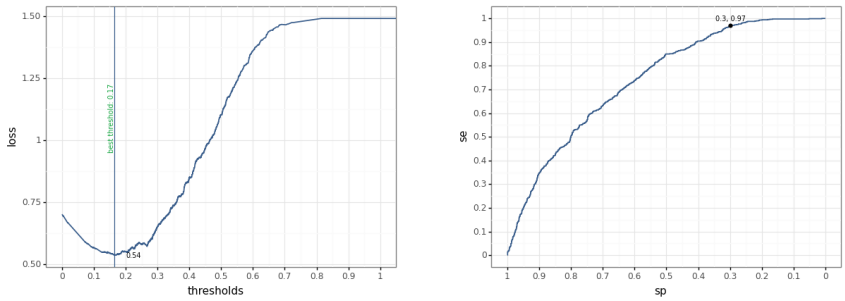

Figure4 Threshold selection and ROC results of cross-validation of the logit model on the training set

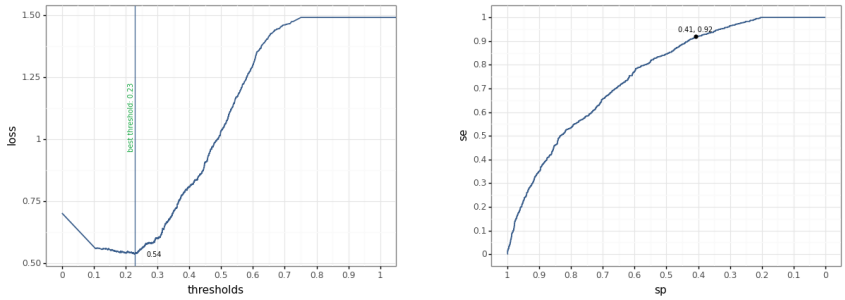

Figure5 Threshold selection and ROC results of cross-validation of Random Forest on the training set

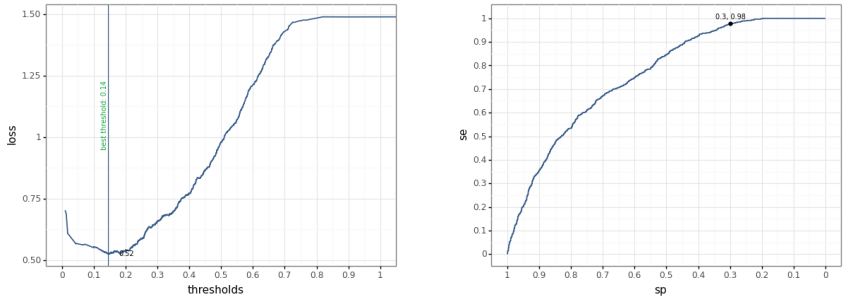

Figure6 Threshold selection and ROC results of cross-validation of GBM on the training set

Models		Predicted no fast_growth	Predicted fast_growth
Logit	Actual no fast_growth	812	1814
	Actual fast_growth	60	1091
Random Forest	Actual no fast_growth	975	1651
	Actual fast_growth	94	1057
GBM	Actual no fast_growth	781	1845
	Actual fast_growth	44	1107

Table3 Confusion matrix on the test set based on the above optimal threshold conditions

	Model	Number of predictors	CV RMSE	CV AUC	CV threshold	CV expected Loss
0	X1	9	0.454952	0.617490	0.177295	0.665829
1	X2	16	0.454116	0.625344	0.151930	0.673971
2	X3	31	0.428237	0.737948	0.143063	0.544619
3	X4	74	0.428712	0.736242	0.148817	0.547863
4	X5	141	0.426765	0.741170	0.159429	0.541970
5	rf_p	141	0.421505	0.757669	0.216233	0.536542
6	gbm_p	141	0.421926	0.755794	0.145089	0.534027

Table4 Summary CV result of Logit, Random Forest and GBM

It can be seen from Table 3 that under the best threshold on training set, Random Forest has the highest accuracy on the test set, reaching (975 + 1057) / (975 + 1651 + 94 + 1057) = 0.538. Logit comes next, and GBM is the worst.

From the final CV results, random forest performed best in RMSE and AUC, followed by GBM.