Prediction with Machine Learning for Economists assignment 3: Predicting fast growth of firms (2010-15)

Mihaly Hanics, MS in Social Data Science

This report is modelling and predicting whether firms are growing fast (as defined later), modelling and testing out different machine learning models.

Data: The data is gathered from OSF. I gathered inspiration from the notebooks from Chapter 17 of the book: Gábor Békés, Gábor Kézdi – Data Analysis for Business, Economics and Policy. All code used can be found at me9hanics/DA3-phdma.

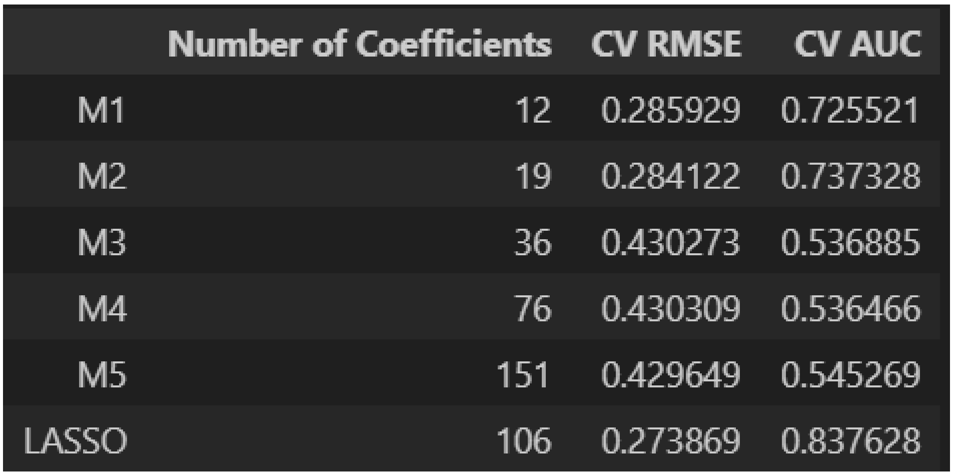
Data preprocessing: In the task, we filtered only for business data from 2010 to 2015, and for firms with sales higher than 1000 euros, but smaller than 10 million euros. The processing of data is analogous to the example shown in the book of predicting the exit of firms, it includes size, financial, and “historical”/cultural variables. However, after seeing the 0 correlation of my target variable and the CEO age statistics, I decided to drop attributes related to the CEO (except gender). Also, there were only 7 cases when there was a problem with an asset being negative, I dropped these cases and the “flag_asset_problem” variable. Modified data was saved to “data_2010_2015.csv”. Data was later split into training and holdout datasets.

The target variable for each model was the “fast_growth” variable, which was not originally in the dataset. I defined it for each company-year combination, that if the sales was at least 20% higher last year than 2 years ago, and at least 20% higher than last year, then the company in that year is defined as a fast grower. In other cases, it is defined as not a fast grower, so it is a binary variable. This led to about 9-10% (12k instances of 112k) of the data have a positive fast growth value. In the dataframe, the value of 1 accounts for a fast growth, and 0 means not fast growth. I defined the fast growth in this way because I wanted it to be a function of continuous growing for 2+ years, and 20% seemed like a good pick as there are not many but enough positive instances. Furthermore, we will compare two industries (manufacture vs services) and for modelling, turn the industry category into “dummy variables” (one-hot encoding each class, so e.g. turning NACE code 30 industry into a column, where the value is 1 if the instance’s industry code is 30. There were other variables that seemed mostly irrelevant for our analysis, however I kept everything else to make sure nothing is left out. LASSO will select the major components.

The models:

1) Logit models (1-5, from least complex to most complex): from 12 coefficients to 151, increasing in complexity. The most basic model only accounts for financial variables and industry. The models are cross-validated (with a 5-fold CV) for optimal performance.
2) LASSO’d logit model: This is a the most complex logit model (M5) with LASSO shrinkage, to reduce it to less attributes, only to the ones that matter above some percent.
3) Random forest: I ran a cross validation (for 90 minutes!) on maximum features: {5,6,7}, and minimum samples split: {11,16}. The best parameter grid turned out to be the maximal ones: 7 features, 11 min samples split.

The cross-validation results for logit models and LASSO logit:



	Number of Coefficients	CV RMSE	CV AUC
M1	12	0.285929	0.725521
M2	19	0.284122	0.737328
M3	36	0.430273	0.536885
M4	76	0.430309	0.536466
M5	151	0.429649	0.545269
LASSO	106	0.273869	0.837628

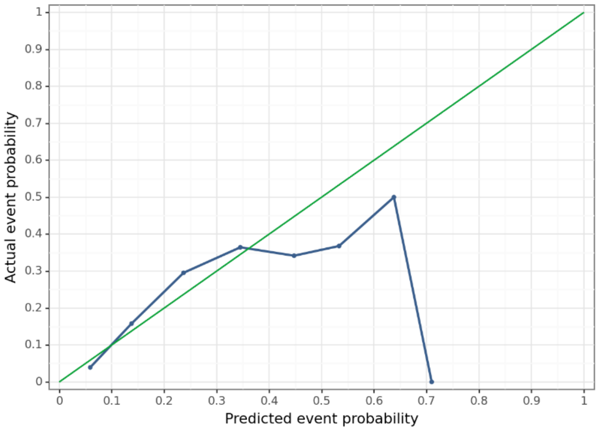
We see that models M3-M5 do very similarly, just as M1 and M2. Typically, for the "best model", the simplest model of the best performers is chosen, and I see that M1 is by far the most sensible choice. For this task, I choose one of the more complex models for "presentation", as smaller models have too low amount of attributes. Later, for the other task, I choose M1 to showcase the actual best model. Among M3-M5, because M3 had less than half parameters as M4, I choose M4 as a "golden middle" for the selected model: many parameters, but not too many. The LASSO model predicts a little bit better than the two simple (best) logit models, and has a very good AUC score.

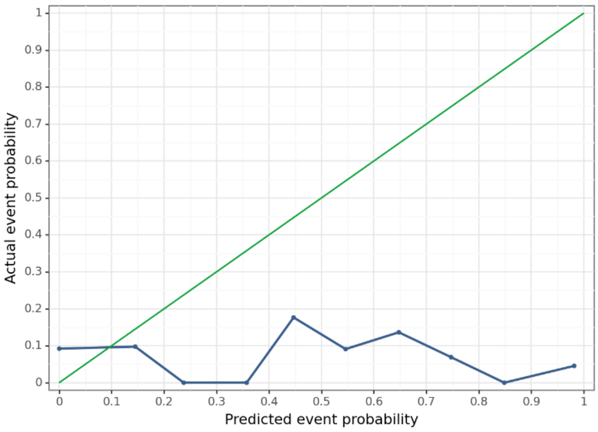
Probability predictions:
Mean-squared error and AUROC on the holdout set

The chosen logit model (M4) used for prediction on the holdout dataset led to around **0.28** RMSE, which is fairly good, better than expected, as for the cross validation, the RMSE was 0.43. But since our data is very unbalanced (90% vs 10% class distribution), we have to check if it doesn't just always predict "not fast growing". Since the ROC curve does not depend on class size distribution, we can measure the area under the ROC curve to see how well our model does realistically. The result came out to be 77%, which is good, better than I expected.

Now for the LASSO model: For some reason, the prediction's RMSE came out to be 0.944, which is almost 1 (the maximum value), screaming a terrible prediction. This must be an error. The ROC curve came out to be 49.5%, which is about as good as 50% that we get from tossing a coin. So there is a problem here. I checked, for some reason the model very often predicts "is growing fast", about as many times as it should actually predict that tbe firm is not growing fast. I swapped around the predictions: if LASSO prediction was that with p probability, the business is growing fast, now it is assumed that with 1-p probability is the business a fast grower. This led to a RMSE of 0.323, which is fine, but the area under ROC is about 50%. I sense a problem with the data provided for the LASSO testing, but I couldn't find an error in the code.

Callibration curves:





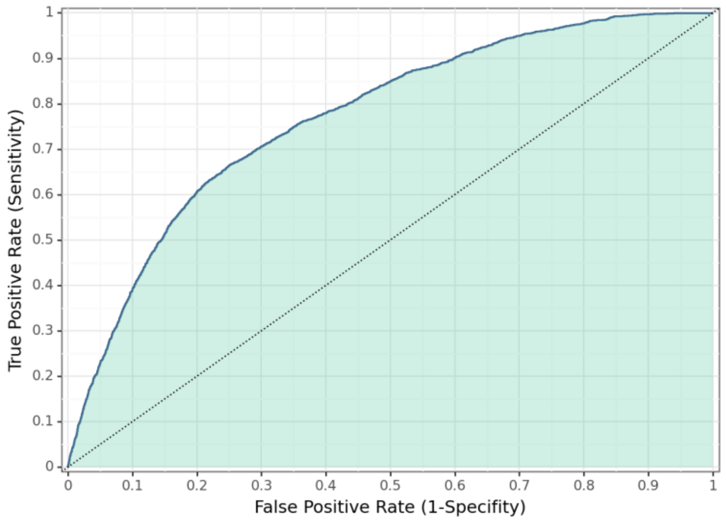
Again, a very weird observation on the LASSO model.

Confusion tables:

The confusion table values are as expected for the model, but for the lasso model, again roles had to be reversed. We got around 20000 true positive/negative predictions for both models at a threshold of 0.5, and around 2000 wrong predictions, thus our model does around 90% accurately – this of course can be "biased" because 90% of the elements are the same. This is somewhat the case – compared to the couple hundred "fast growth" predictions, there were 22k no-fast-growth predictions, so this distribution is even less than the original 9-10%. I played around with threshold values to separate on what is a no_fast_growth cs fast_growth prediction and found that the mean prediction value is not a good threshold, and the "optimal" threshold basically almost never guessed fast growth, so this is also bad. I choose the threshold to be 0.25 as a somewhat good value. The two models predicted around the same accuracy, but the (corrected) LASSO model predicted more fast growth firms.

ROCs:

The ROC for the best logit model:



The ROC for the LASSO logit model is just a 45° line, so again somehow equivalent to a biased coin… Maybe the evaluation of the model is actually wrong.

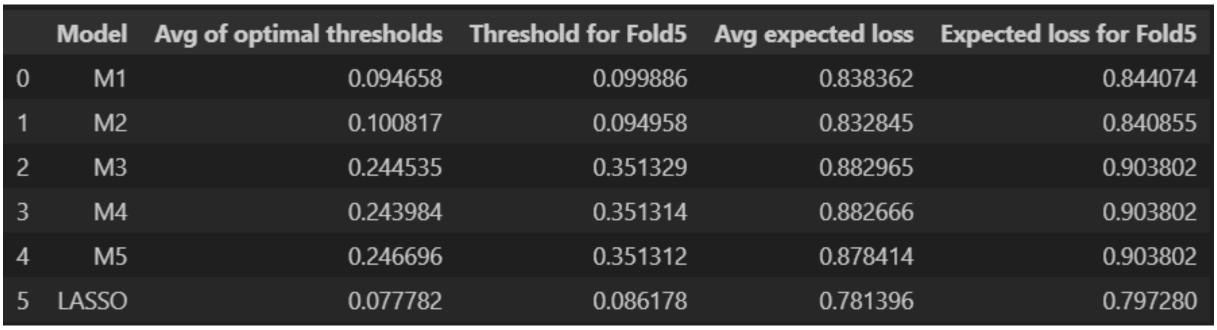
Loss function based classification:

We define a loss function to minimize for fine-tuning models. For that, we need to define a cost, which is not trivial.

Imagine we're an investing firm and we want to find promising firms to invest in. We want to find firms that will grow fast indeed, but we don't want to miss out on any firms that will grow fast. There is a reason to be careful with failing investments, so to avoid false positives, but there is also a reason to be careful with missing out on good investments, so to avoid false negatives. Naturally, because among the best investments a man can make, investing in startups is among the best returners as they can grow magnitudes in a span of a few years. So we'd rather have an increasing number of false positives to have less false negatives.

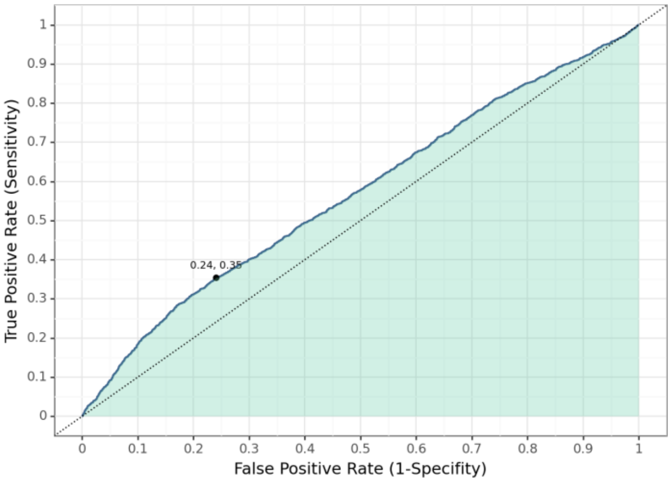
If we are talking big enough firms (which we do), then an investment must be big to have an impact, so in this case we can't really expect that high returns with all fairness. But we'd expect smaller losses aswell, which balances out thing. So I'd just set it at the reciprocate of the ratio of the occurrence of the two errors based on previous results, times 10, to have a some inclination towards false negatives.

However, this lead to many models having infinite optimal thresholds (bad computation), so I instead set the cost function to be 10 (FN/FP).



	Model	Avg of optimal thresholds	Threshold for Fold5	Avg expected loss	Expected loss for Fold5
0	M1	0.094658	0.099886	0.838362	0.844074
1	M2	0.100817	0.094958	0.832845	0.840855
2	M3	0.244535	0.351329	0.882965	0.903802
3	M4	0.243984	0.351314	0.882666	0.903802
4	M5	0.246696	0.351312	0.878414	0.903802
5	LASSO	0.077782	0.086178	0.781396	0.797280

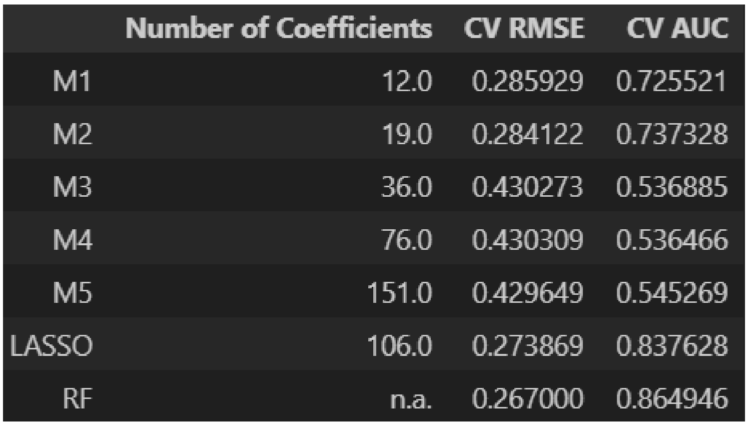
If we now examine M1 instead (as the “best model”), we see that the optimal threshold (based on the loss function) is ~0.095, with that threshold, we get this ROC:



Other models did worse on their optimum, but this is not super impressive ROC either. But it is optimalized for minimal loss, so in other terms it is as good as it can get.

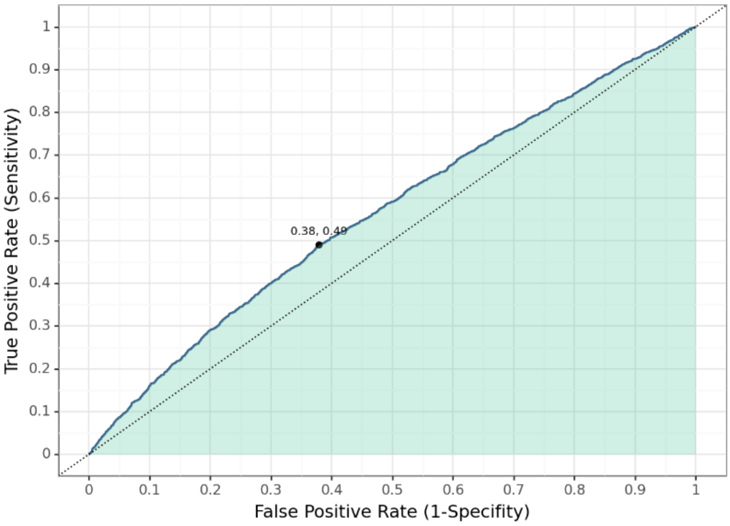
Prediction results for Random Forest:

Cross-validation results (including other models):



	Number of Coefficients	CV RMSE	CV AUC
M1	12.0	0.285929	0.725521
M2	19.0	0.284122	0.737328
M3	36.0	0.430273	0.536885
M4	76.0	0.430309	0.536466
M5	151.0	0.429649	0.545269
LASSO	106.0	0.273869	0.837628
RF	n.a.	0.267000	0.864946

On optimum, the RF model's ROC:



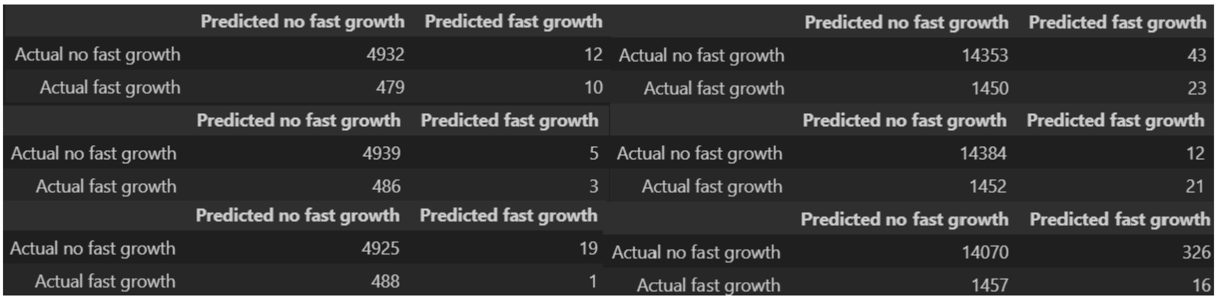
The RMSE on the holdout set is 0.263, whereas AUROC is 86.7%, the best in both categories.

Test (best) models on two different industries:

One industry is manufacturing, the other is services (food, accommodation).

Model	Manuf. RMSE	Manuf. AUROC	Services RMSE	Serv. AUROC
Best logit (M1)	0.276	76.1%	0.278	77.3%
LASSO	0.305	51% (?)	0.331	50% (?)
Random Forest	0.263	86.1%	0.264	86.9%

Confusion matrices for the three models and two different industries:



	Predicted no fast growth	Predicted fast growth
Actual no fast growth	4932	12
Actual fast growth	479	10

	Predicted no fast growth	Predicted fast growth
Actual no fast growth	14353	43
Actual fast growth	1450	23

	Predicted no fast growth	Predicted fast growth
Actual no fast growth	4939	5
Actual fast growth	486	3

	Predicted no fast growth	Predicted fast growth
Actual no fast growth	14384	12
Actual fast growth	1452	21

	Predicted no fast growth	Predicted fast growth
Actual no fast growth	4925	19
Actual fast growth	488	1

	Predicted no fast growth	Predicted fast growth
Actual no fast growth	14070	326
Actual fast growth	1457	16

Interpretation:

Whilst the LASSO model can be a strong tool, when we have way too many variables (see how model 1 with only 12 attributes did best, whilst LASSO still resulted in a 70-attribute model) and only a few matter, it is a not strong enough tool. Further analysis of the LASSO method is hard as we had problems with evaluating it, getting mixed results.

The best logit model did solidly, surprisingly it was the model with the least variables, and it did comparably well in RMSE on both the holdout set and CV results to the random forest (and even fairly well in AUC). But the random forest model is clearly the “winner” here, as it topped all categories, and it was very easy to build it and fine tune it.

Overall, a problem is that all models overpredict the most common case: no fast growth, even with a defined loss function. To address this, we may use over/undersampling, two common methods.