Assignment 3 Technical Report[1]

by Evgeny Gushchin (December 14, 2023)

Task 1

Our main task is to build a model to predict fast growth of firms. In order to train and test the model we are using bisnode-firms data[2]. The original database contains 287,829 observations and 48 variables. We start with dropping the variables that have missing values for at least 90% of observations (these are: *COGS*, *finished_prod*, *net_dom_sales*, *net_exp_sales*, *wages*, *D*). Next, we filter our dataset to the period between 2010 and 2015[3]. We keep only the companies that are still operating:

data <- data %>% filter(is.na(exit_year))

Creating the target variable

The task of finding a good indicator for fast growth of firms is nontrivial. We are interested not just in the volume of growth but also in its quality. For this purpose we decided to construct and use Fixed Asset Turnover (FAT) Ratio[4]. FAT is useful in determining whether a company is efficiently using its fixed assets to drive net sales. FAT is calculated by dividing net sales by the average balance of fixed assets of a period:

$$FAT = \frac{Net\_Sales}{Average\_Fixed\_Assets}$$

Thus, using *sales* and *fixed_assets* variables from our dataset we can construct FAT ratio and use it for our analysis (see Figure 1). We should keep in mind that "though the ratio is helpful as a comparative tool over time or against other companies, it fails to identify unprofitable companies" (because it doesn't take debt into account).

data <- data %>% mutate(fat = ifelse(fixed_assets != 0, sales/((fixed_assets +lag(fixed_assets))/2), 0))

We take the logarithm of the new variable that we have created and then after grouping by company ID we take the first difference of the log(FAT ratio) in order get the percentage change in this indicator:

[1]When writing the code for this assignment I was using the help of ChatGPT 3.5. See link:

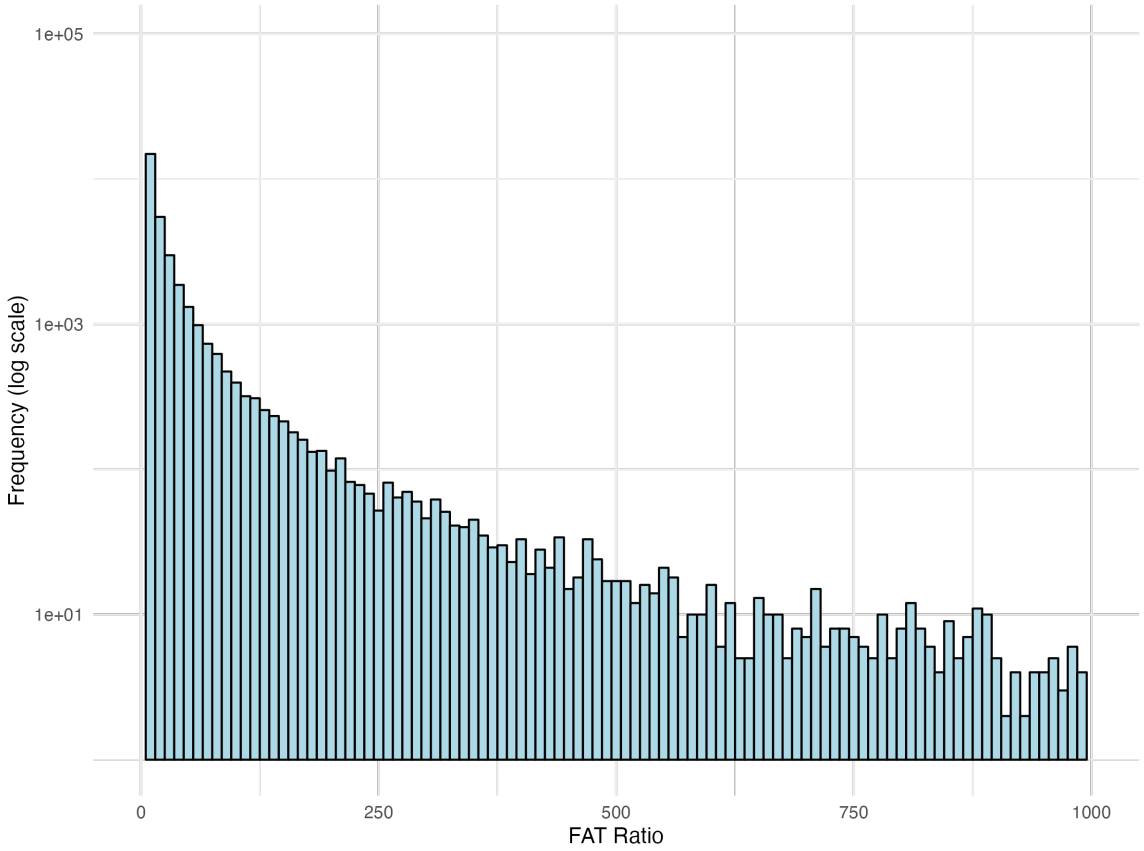

Figure 1: Constructed FAT Ratios

data <- data %>% group_by(comp_id) %>% mutate(d1_ln_fat = ln_fat - lag(ln_fat, 1)) %>% ungroup()

Afterwards we create *fast_growth* variable that takes the value of 1 when *d1_ln_fat* > 5 and 0 otherwise. To build our model we use the variable *future_fast_growth* that is a 1 period lead of *fast_growth*.

We can suggest two alternative ways of defining fast growth: 1. by using *sales* variable, yet this variable alone would not take into account the size of a company (for smaller companies it would be easier to show high growth rates of total sales) 2. by using

https://chat.openai.com/share/ed3e891f-76a6-43a0-979b-fbff3ff11448

[2]The dataset was downloaded on December 11, 2023. Link: https://osf.io/b2ft9/

[3]Link to my code on Github: https://github.com/evgeny-gushchin/DA3-phdma/blob/main/A3/Gushchin_A3_code.R

[4]More information about FAT ratio can be found here: https://www.investopedia.com/terms/f/fixed-asset-turnover.asp

labor_avg variable, that captures the changes in employment, yet this variable is noisy and has more than 50% of missing values.

In the end we restrict our sample only to 2012 data.

Dealing with unbalancedness

There is one problem in the target variable *future_fast_growth* that we have created. It is too restrictive, less that 1% of the firms in our sample become fast growing according to this measure. That is why we perform downsampling in order to improve the predictive power of our models:

majority_indices <- data %>% filter(future_fast_growth == 0) %>% sample_frac(0.05) rownames_to_column(var = "row_index") we keep only 5% of the companies that do not show future fast growth

data <- data %>% filter(future_fast_growth == 1 | row_number() %in% majority_indices$row_index)

Thus we increase the share of fast growing firms from 1% to 13%. But the total number of observations goes down to to <1,000.

Feature engineering

- we unite some industry category codes in *ind2*
- in *d1_ln_fat* we replace with 0 for new firms + add dummy to capture it
- we change negative values in assets to 0 and add a flag
- we generate total assets
- we divide all pl_names elements by sales and create new column for it
- divide all bs_names elements by total_assets_bs and create new column for it
- winsorizing tails and creating flags

Models

We use build and compare three prediction models:

- Logit (variables: *sales_mil_log*, *sales_mil_log_sq*, *d1_ln_fat_mod*, *profit_loss_year_pl*, *fixed_assets_bs*,*share_eq_bs*, *curr_liab_bs*, *curr_liab_bs_flag_high*, *curr_liab_bs_flag_error*, *age*, *foreign_management*, *ind2_cat*)
- Lasso logit
- Random Forest (we don't use interactions or modified features)

For each model we do 5-fold cross validation to estimate the best performance model. Also we take out 20% of observations as our holdout set.

PART I: Probability prediction

From Table 1 below we can see that Random Forest is the best model in terms of both RMSE (the lowest) and AUC (the highest). Thus this is the model we will use in Task 2.

Table 1: Model performance

	# of predictors	CV RMSE	CV AUC	CV threshold	CV expected Loss
Logit X2	18	0.325	0.738	0.167	407
Logit LASSO	1	0.337	0.699	0.149	490
RF prob.	36	0.294	0.862	0.164	264

PART II: Classification

For classification task we need the loss function. Lets assume that if we predict fast growth but in fact the company doesn't grow fast next period on average we lose 1,000 dollars, while in case we fail to identify the company that will be growing fast our expected loss is 5,000 dollars.

In Table 1 we saw that Random Forest has the lowest estimated loss among the 5 folds (264 dollars compared to 407 dollars in case of Logit and 490 dollars in case of Lasso).

PART III: Discussion of results

From Table 2 below we see the results of classification using our holdout sample (20% of our observations). Using the threshold that depends on our loss function. We can see that fast growing firms are still rare. So getting just one out of 4 fast growing firms right is already a good result. This result is proven when we check the expected loss in

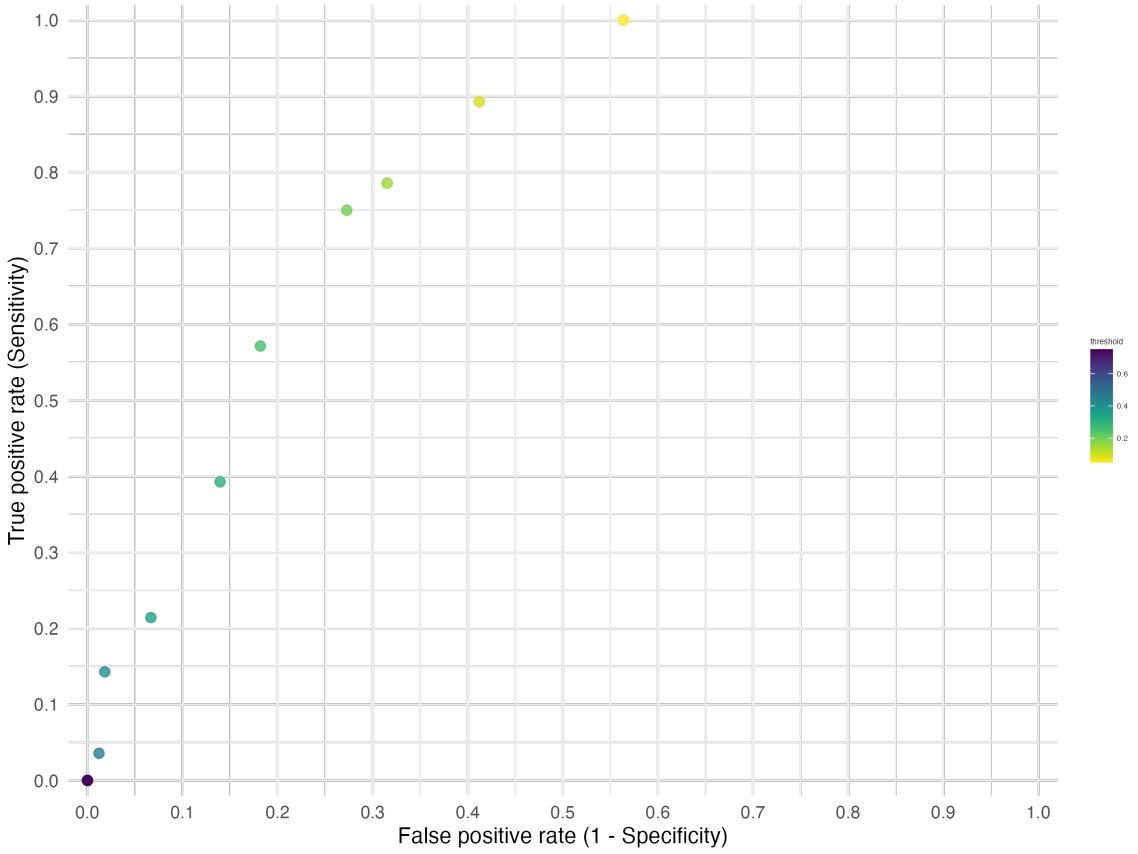

Figure 2: Discrete ROC for Logit

the holdout sets: Random Forest gives 363 dollars of loss and Logit gives 430 dollars of loss.

Task 2

Using *ind2_cat* variable we can devide our dataset into two: Manufacturing companies (NACE codes 26-33) and Accommodation and food service companies (NACE codes 55-56).

On both samples we run our best model (Random Forest) separately and compare the results.

In manufacturing sector our prediction model's performance is worse: expected loss in the holdout set is much greater (941 dollars) than in the case of services sector (526 dollars). One reason for this is that the share of fast-growing companies in the

Table 2: Confusion table for Random Forest

	Reference	
	no_fast_growth	fast_growth
Prediction		
no_fast_growth	162	27
fast_growth	3	1

manufacturing sample is much lower than in the services sample. Another possible explanation – there are predictors that are missing in our dataset that are particularly important for predicting fast growth in manufacturing sector, than in services.

Conclusions

To define fast growth we have constructed a variable for the annual growth rate of Fixed Asset Turnover ratio. We indicate that the firm is growing fast if its FAT ration increases by more than 5%. Due to unbalancedness we had to do downsampling. Among the three models that we used (Logit, Lasso and Random Forest) Random Forest proved to be the best one (in terms of RMSE, AUC and expected loss).

Yet, the performance of the model is not very stable if we divide our sample into two: manufacturing and service companies. The expected loss in manufacturing sectors is higher than in services.