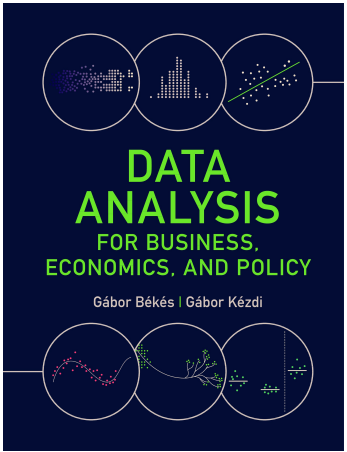
17. Probability, Prediction and Classification

Gábor Békés

Data Analysis 3: Prediction

2020

Slideshow for the Békés-Kézdi Data Analysis textbook



- Cambridge University Press, 2021
- **gabors-data-analysis.com**
  - Download all data and code: gabors-data-analysis.com/data-and-code/
- This slideshow is for **Chapter 17**

Predicting a binary y: probability prediction and classification

- With a binary y variable, we can make two kinds of predictions
- Probability prediction: predicting the probability of $y = 1$ for each observation, $\hat{y}^P$, which is the shorthand notation for $\hat{P}[y = 1|x]$
- Classification: predicting whether $\hat{y} = 0$ or $\hat{y} = 1$ for each observation

What's New and not new with binary target?

- Probability predicted not value
- Desire to classify
  - assign 0 or 1
  - based on a probability that comes from a model
  - But how?
- New measures of fit
  - Some based on probabilities
  - Others based on classification
- But: Still need best fit, with highest external validity
- Models similar to those used earlier
  - Regression-like models (probability models)
  - Tree-based models (CART, Random Forest)

Firm exit case study: Case study: background

- Banks and business partners are often interested in the stability of their customers.
- Predicting which firms will be around to do business with is an important part of many prediction projects.
- Working with financial and non-financial information, your task may be to predict which firms are more likely to default than others.

Data tables for firm-level data

Data table	Unit of observations	Important variables
Financial data	firm–year	revenue, material cost, EBITDA, assets,
Managers	person–time interval–firm	CEO age, gender
Employment	firm–year	average wages
Corporate registry	firm	city of headquarters, year of firm birth
Industry classification	firm	NACE industrial classification

Data work

- Firm data
- Dataset is a panel data
  - We created earlier
  - Rows are identified by company id (comp-id) and year.
- We'll focus on a cross-section of 2012.
- The target is hence a binary variable called exit,
  - 1 if the firm exited within 2 years
  - 0 otherwise.
- Plenty of other sample design, feature engineering steps - please read 18.A1 and look at the code.

Firm exit case study: Features - overview

- Key predictors
  - size: sales, sales growth
  - management: foreign, female, young, number of managers
  - region, industry, firm age
  - other financial variables from the balance sheet and P&L.
- For financial variables, we use ratios (to sales or size of balance sheet).
- Here it will turn out be important to look at functional form carefully, especially regarding financial variables.
- Mix domain knowledge and statistics.

firm exit: Model features 1

- **Firm**: Age of firm, squared age, a dummy if newly established, industry categories, location regions for its headquarters, and dummy if located in a big city.
- **Financial 1**: Winsorized financial variables: fixed, liquid (incl current), intangible assets, current liabilities, inventories, equity shares, subscribed capital, sales revenues, income before tax, extra income, material, personal and extra expenditure.
- **Financial 2**: Flags (extreme, low, high, zero - when applicable) and polynomials: Quadratic terms are created for profit and loss, extra profit and loss, income before tax, and share equity.
- **Growth**: Sales growth is captured by a winsorized growth variable, its quadratic term and flags for extreme low and high values.

Model features 2

- **HR**: For the CEO: female dummy, winsorized age and flags, flag for missing information, foreign management dummy; and labor cost, and flag for missing labor cost information.
- **Data Quality**: Variables related to the data quality of the financial information flag for a problem, and the length of the year that the balance sheet covers.
- **Interactions**: Interactions with sales growth, firm size, and industry.

Probability prediction

We build models to predict probability when:

- aim is to predict probabilities – this is what we do
- aim is to classify (predict 0 or 1) – this is the first step
  - build probability models, select the best one
  - use a loss function to classify

Predicting probabilities

- Mostly covered in Chapter 11
- Use the logit model
- Loss function is RMSE = Brier score
- Calibration curve will be useful

Models

Models (number of predictors)

- Logit M1: handpicked few variables ($p = 11$)
- Logit M2: handpicked few variables + Firm ($p = 18$)
- Logit M3: Firm, Financial 1, Growth ($p = 35$)
- Logit M4: M3 + Financial 2 + HR + Data Quality ($p = 79$)
- Logit M5: M4 + interactions ($p = 153$)
- Logit LASSO: M5 + LASSO ($p = 142$)
    - Number of coefficients = N of predictors +1 (constant)

Probability model specifications: number of variables in models

	Hand-picked	Firm	Financial 1	Financial 2	Growth	HR	Data quality	Interactions
Logit M1	x							
Logit M2	x	x						
Logit M3		x	x		x			
Logit M4		x	x	x	x	x	x	
Logit M5		x	x	x	x	x	x	x
Logit LASSO		x	x	x	x	x	x	x

Cross validation

- $N = 19,036$
- $N = 15,229$ in work set (80%)
  - Cross validation 5x training + test sets
  - Used for cross-validation
- $N = 3,807$ in holdout set (20%)
  - Used only for diagnostics of selected model.

Simple logit model coefficients

Variable	Coefficient	Marginal difference
Age	-0.035	-0.005
Current liabilities	0.171	0.023
Curr. liab. flag error	0.318	0.043
Curr. liab. flag high	0.135	0.018
Change in log sales	-0.482	-0.065
Fixed assets	-0.811	-0.109
Foreign management	0.216	0.029
Electrical equipment	0.145	0.018
Machinery and equipment	0.033	0.004
Motor vehicles	0.406	0.052
Other transport equipment	-0.033	-0.004
Repair and installation	-0.158	-0.018
Accommodation services	0.129	0.015
Food & beverage services	0.464	0.061
Profit loss last year	-0.450	-0.060
Log sales	-0.180	-0.024
Log sales sq	0.015	0.002
Share equity	-0.388	-0.052

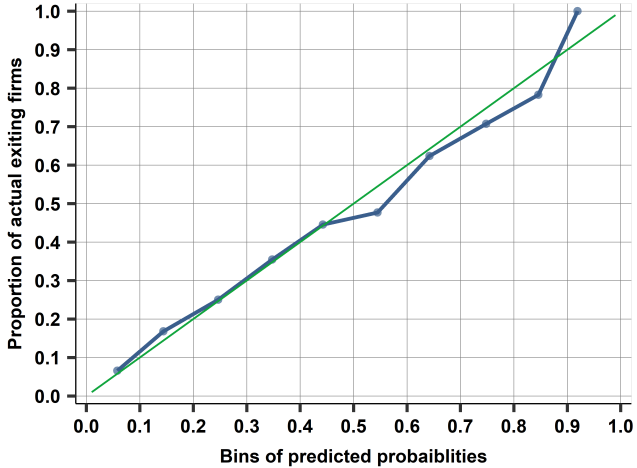
Comparing model fit

	Variables	Coefficients	CV RMSE
Logit M1	4	12	0.374
Logit M2	9	19	0.366
Logit M3	22	36	0.364
Logit M4	30	80	**0.362**
Logit M5	30	154	0.363
Logit LASSO	30	143	0.362

5-fold cross-validated on work set, average RMSE

Will use Logit M4 model as benchmark.

Calibration curve



Classification process

- Predict probability
- Make into 0/1 predictions - classifications
- We can make errors
  - False negative
  - False positive

Classification and the confusion table

- The confusion table shows the number of observations by their predicted class ($\hat{y} = 0$ or $\hat{y} = 1$) and actual class ($y = 0$ or $y = 1$)
- In the table, correctly classified observations are called true positives (TP; $\hat{y} = 1, y = 1$) and true negatives (TN; $\hat{y} = 0, y = 0$)
- Incorrectly classified observations are called false positives (FP; $\hat{y} = 1, y = 0$) and false negatives (FN; $\hat{y} = 0, y = 1$)

The confusion table for binary y

	$y_j = 0$ Actual negative	$y_j = 1$ Actual positive	Total
$\hat{y}_j = 0$ Predicted negative	TN (*true negative*)	FN (*false negative*)	TN + FN (*all classified negative*)
$\hat{y}_j = 1$ Predicted positive	FP (*false positive*)	TP (*true positive*)	FP + TP (*all classified positive*)
Total	TN + FP (*all actual negative*)	FN + TP (*all actual positive*)	N = TN + FN + FP + TP (*all observations*)

Measures of classification

- **Accuracy =(TP+TN)/N**
  - The proportion of rightly guessed observations
  - Hit rate
- **Sensitivity =TP / (TP+FN)**
  - The proportion of true positives among all actual positives
  - Probability of predicted y is 1 conditional on $y = 1$
- **Specificity = TN/(TN+FP)**
  - The proportion of true negatives among all actual negatives
  - Probability predicted y is 0 conditional on $y = 0$

Measures of classification

- The key point is that there is a trade-off between making false positive and false negative errors.
- This is the essential insight in classification
- This can be expressed with specificity and sensitivity.

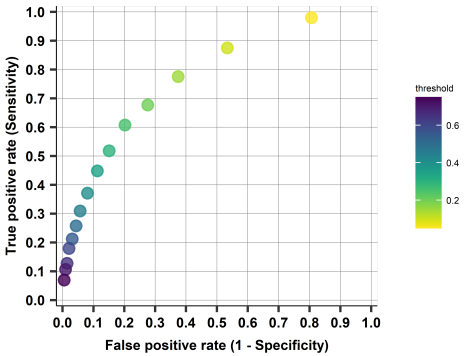
ROC Curve

- The *ROC curve* is a popular graphic for simultaneously displaying specificity and sensitivity for all possible thresholds.
  - ROC: Receiver operating characteristic curve
  - Name from engineering
- For each threshold, we can compute confusion table –> calculate sensitivity and specificity
- Show in graph - illustrate (non-linear) trade-off

Illustrating the trade-off between different classification thresholds: the ROC curve

- The ROC curve illustrates the trade-off between false positives and false negatives for when various thresholds are used to turn probability predictions from an estimated model into classification The ROC curve shows the proportion of false positives among all $y = 0$ observations (one minus specificity) and the proportion of true positives among all $y = 1$ observations (sensitivity); the threshold values are not shown on the graph
- The ROC curve of a completely random probability prediction is the 45 degree line
- The ROC curve of a perfect probability prediction would jump from zero to one and stay at one (running along the upper edge of the box)

ROC Curve: a two-dimensional plot



- Horizontal axis: False positive rate (one minus specificity) = the proportion of FP among actual negatives
- Vertical axis: is true positive rate (sensitivity) = proportion of TP among actual positives
- For classifications from a single probabilistic forecast as the threshold is moved from 0 to 1

Area Under ROC Curve

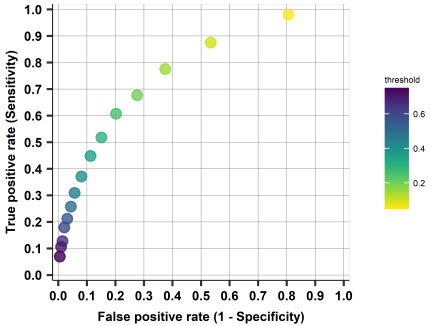
- ROC curve: the closer it is to the top left column, the better the prediction.
  - Perfect model: horizontal line at TPR=1
- Area under ROC curve (AUC) summarizes quality of probabilistic prediction
- For all possible threshold choices
- Area = 0.5 if random classification
- Area > 0.5 if curve mostly over 45 degree line
- AUC is a good statistic to compare models
- Defined from a non-threshold dependent model (ROC)
- The larger the better
  - Ranges between 0 and 1.

Predicting firm exit: probability and classification

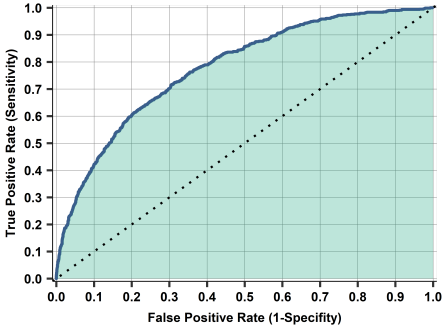
- Illustrate ROC curve based on the logit M4 model, estimated on one of the training sets; predictions on the corresponding test set.

ROC curve for predicting firm exit

(a) ROC curve points for various thresholds



(b) Continuous ROC curve



RMSE and AUC for various models of firm exit

Model	RMSE	AUC
Logit M1	0.374	0.738
Logit M2	0.366	0.771
Logit M3	0.364	0.777
Logit M4	0.362	0.782
Logit M5	0.363	0.777
Logit LASSO	0.362	0.768

Note: Models are described in the text. Five-fold cross-validated RMSE and AUC (area under the ROC curve) values.

Confusion tables with thresholds 0.5 and 0.2

	Threshold: 0.5			Threshold: 0.2		
	Actual stay	Actual exit	Total	Actual stay	Actual exit	Total
Predicted stay	75%	15%	90%	57%	7%	64%
Predicted exit	4%	6%	10%	22%	14%	36%
Total	79%	21%	100%	79%	21%	100%

Note: Percentage of observations in cells. Firm exit predictions based on the logit M4 model. Classification on the holdout sample; two confusion tables next to each other.

Accuracy, sensitivity, and specificity (in percent)

	Threshold: 0.5	Threshold: 0.2
Accuracy	81%	71%
Sensitivity	28%	68%
Specificity	95%	72%

Note: Firm exit predictions based on the Logit M4 model. Classification on the holdout sample based on two thresholds.

Threshold choice consequences

- Having a higher threshold leads to
  - fewer predicted exits:
    - 10% when the threshold is 50% (36% for threshold 20%).
  - fewer false positives (4% versus 22%)
  - more false negatives (15% versus 7%).
- The 50% threshold leads to a higher accuracy rate than the 20% threshold
  - 50% threshold: 75% + 6% = 81%
  - 20% threshold: 57% + 14% = 71%
  - even though the 20% threshold is very close to the actual proportion of exiting firms.

Classification

- How we make classification from predicted probability?
- Set a threshold!

- The process of classification
- If probability of event is higher than this threshold–> assign (predict) class 1; and 0 otherwise.

- Who sets the threshold?
- We need a loss function.
- A loss function is a dollar (euro) value assigned to false positive and false negative.

Loss function and finding the optimal classification threshold

- A loss function for classification attaches loss values for incorrect predictions (*FP* and *FN*)
- The best classification produces the lowest expected loss:
  $E[loss] = P[FN] \times loss(FN) + P[FP] \times loss(FP)$
- Based on predicted probabilities, the best classification is achieved with the optimal classification threshold, which minimizes expected loss

Loss function and finding the optimal classification threshold

- One way to find the optimal threshold is to use the formula $\frac{loss(FP)}{loss(FP) + loss(FN)}$
- Another method of finding the optimal threshold is a search algorithm, which selects the probability model and the optimal classification threshold together
  - cross-validated search over a variety of thresholds given the loss function
  - Cost-sensitive Youden index
  - More complicated, more robust way

The loss function

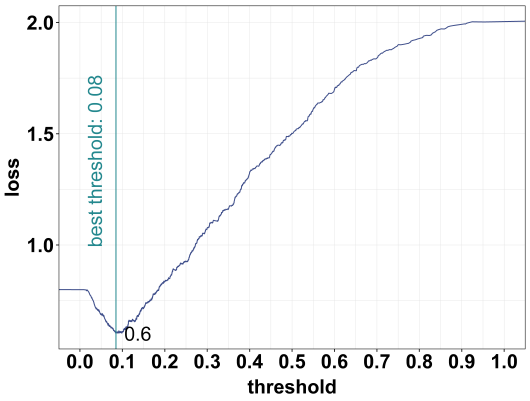
- Loss function = FN, FP
  - What matters is FN/FP
- FN=10
  - If the model predicts staying in business and the firm exits the market (a false negative), the bank loses all 10 thousand euros.
- FP=1
  - If predict exit and the bank denies the loan but the firm stays in business in fact (a false positive), the bank loses the profit opportunity of 1 thousand euros.
- With correct decisions, there is no loss.

Finding the threshold

- Find threshold by formula or algo

- Formula: the optimal classification threshold is $1/11 = 0.091$
- Algo: search thru possible cutoffs

Finding the threshold



- Consider all thresholds $T = 0.01, 0.02$—1
- Calculate the expected loss for all thresholds
- Pick when loss function has the minimum
- *Done in CV, this is fold Nr.5.*

Ranking of classifications (models and classification thresholds)

	RMSE	Optimal threshold	Expected loss
Logit M1	0.374	0.089	0.722
Logit M2	0.366	0.096	0.660
Logit M3	0.364	0.092	0.630
Logit M4	0.362	0.082	0.619
Logit M5	0.363	0.091	0.637
Logit LASSO	0.362	0.104	0.665

Note: Models are described in the text. RMSE, optimal threshold selection, and expected loss are results of 5-fold cross-validation (averages).

The process review

- Model selection process
  - Predict probabilities
  - Use predicted probabilities and loss function to pick optimal threshold
  - Use that threshold to calculate expected loss
  - Pick model with smallest expected loss (in 5-fold CV)
- We run the threshold selection algorithm on the work set, with 5-fold cross-validation.
  - Best is model Logit M4
  - the optimal classification threshold by algo is 0.082. Close to formula (0.091)
  - The average expected loss of 0.64.

Classification tree

- Classification tree, predict the class (0/1)
- Same: Building trees with recursive binary splitting
- Different: prediction is not the mean of values, but the share of $y = 1$
- Probability<–>Frequency
- Based on threshold
- Different: Loss function

New loss function

- In a classification tree, the measure of fit is **node impurity**.
- Extent to which nodes contain observations with both $y = 0$ and $y = 1$ or only $y = 0$ or $y = 1$.
- A widely used measure is the **Gini index of node impurity**.
- Let's consider a split, for node m, and let $\widehat{p_m}$ represent the share of observations with $y = 1$.

$$Gini = 2\widehat{p_m}(1 - \widehat{p_m})$$

- The index is very small if all observations have either $y = 0$ or all have $y = 1$.
- The closer $\widehat{p_m}$ to 0.5 the larger the value of the index.
- Thus, a small value implies that the node is made up entirely of a single class.

- It turns out so using the Gini index of node impurity or using MSE to find the best fit leads to the same result.
  - See Appendix Ch17.U2

Random forest

- Similar approach to regression trees
- Do classification trees, on bootstrapped datasets, and aggregate them.
- Often perform better than logit models.
  - Similarly to OLS vs Random Forest
- No need for model building
- Better probability prediction
- Slower

- Boosting can also be used for binary y.

Random forest: two options

- Similar approach to regression trees
- Do classification trees, on bootstrapped datasets, and aggregate them Two options

(a technical issue):

- Probability forest + threshold search with algorithm
- Classification forest + threshold formula
- For classification, we can use probability or classification forest.
  - Results tend to be very similar
  - We have to find the optimal classification threshold using a loss function.

- Do not use default setting of "majority voting"!
  - Default for classification random forest is $t = 0.5$

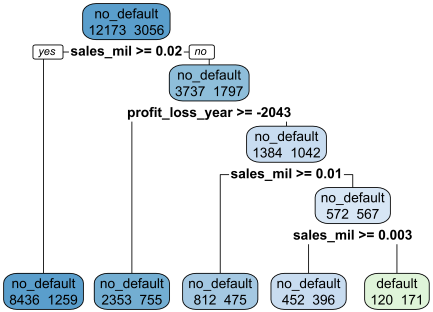
Random Forest summary

- Random Forest works well for prediction when target is binary
- May always use for probability prediction
- Use for classification only with an explicit loss function

Random Forest

- Building block is a CART - classification tree with exit probabilities
- Build a probability random forest
- Same variables as in M4
- Compare models and choose

CART Classification tree



- CART
- a small tree we built for illustration purposes
- with only three variables:
  - firm size (sales),
  - binary variable for having a foreign management
  - Binary if the firm is new.
- Terminal nodes with share of exit predictions

Random Forest

- The model outperforms the logit models, with a cross validated RMSE of 0.358 and AUC of 0.808.
- We used predicted probabilities to find the optimal thresholds, and used this to make the classification.
- The expected loss: 0.587
- smaller than for the best logit (0.619)

Random Forest

- Note that finding the optimal threshold is rather important.
- If used a 0.5 threshold, the expected loss jumped to -1.540 vs -0.587 for the best threshold model.
- This is 2.6 times the loss from the optimal threshold.
- The default option in random forest (and many ML models) for classification is majority voting
- Majority voting is threshold=50%

Comparing two thresholds

- Predict exit if probability>10.9%
- Expected loss: (1.33 x $\underline{10}$ + 45.4 x $\underline{1}$)/100 = 0.587

	Actual stay	Actual exit
Predicted stay	33.6%	1.3%
Predicted exit	45.4%	19.7%

Summary of process with RF

- Run probability forest on work set with 5-CV
- Get average (ie over the folds) RMSE and AUC
- Now use loss function (1,10) and search for best thresholds and expected loss over folds
- Show ROC, loss on fold 5
- Optimal Threshold, average expected loss is calculated
- Take model to holdout and estimate RMSE, AUC and expected loss–>what you expect in live data
- *+1 Show expected loss with classification RF and default majority voting to compare*

Summary of model for **model selection**

Model	Preds	Coeffs	RMSE	AUC	threshold	Exp. loss
Logit M1	11	12	0.374	0.736	0.089	0.722
Logit M4	36	79	0.362	0.784	0.082	0.619
Logit LASSO	36	143	0.362	0.768	0.106	0.642
RF probability	36	n.a.	0.354	0.808	0.098	0.587

RMSE, AZC, Threshold, Loss: all 5-fold CV results (averages).

Class imbalance

- A potential issue for some dataset - relative frequency of the classes.
- Class imbalance = the event we care about is very rare or very frequent ($\Pr(y = 1)$ or $\Pr(y = 0)$ is very small)
  - Fraud
  - Sport injury
- What is rare?
  - Something like 1%, 0.1%. (10% should be okay.)
  - Depends on size: in larger dataset we can identify rare patterns better.
- Consequence: Hard to find those rare events.

Class imbalance: the consequences

- Methods we use not good at handling it.
- Both for the models to predict probabilities, and for the measures of fit used for model selection.
  - The functional form assumptions behind the logit model tend to matter more, the closer the probabilities are to zero or one.
- Cross-validation can be less effective at avoiding overfitting with very rare or very frequent events if the dataset is not very big.
- Usual measures of fit can be less good at differentiating models.
- Consequence
  - Poor model performance
  - Model fitting and selection setup not ideal

Class imbalance: what to do

- What to do? Two key insights.
- 1: Know when it's happening. Ready for poor performance.
- 2: May need an action: **rebalance** sample to help build better models
- Downsampling – randomly drop observations from frequent class to balance out more
  - Before: $100,000$ observations 1% event rate ($99,000$ $y = 1$, $1,000$ $y = 0$)
  - After $10,000$ observations 10% event rate ($9,000$ $y = 1$, $1,000$ $y = 0$)
- Over-sampling of rare events
- Smart algorithms
  - Synthetic Minority Over-Sampling Technique (SMOTE)
  - Others

Main takeaways

- With a binary target variable, we can carry out two kinds of prediction: probability prediction (the probability that an observation has $y = 1$) and classification (whether an observation has y = 1 or $y = 0$)
  - Predicting the probability that $y = 1$ is very similar to predicting the expected value of y, but it may require a different model
  - We can classify by using a threshold value for predicted probabilities
  - We need a loss function for finding the optimal classification threshold value and classify accordingly