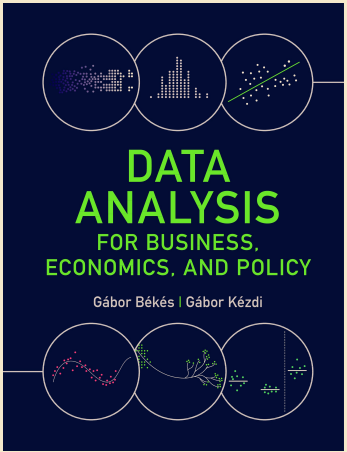
Békés-Kézdi: Data Analysis, Chapter 17: Probability Prediction and Classification



Data Analysis for Business, Economics, and Policy

Gábor Békés (Central European University)
Gábor Kézdi (University of Michigan)

Cambridge University Press, 2021

gabors-data-analysis.com

Central European University

Any comments or suggestions:
gabors.da.contact@gmail.com

Motivation

- Work for a consultancy helping a bank
- Many SME firms as clients, applying for loans
- Decide which firms should get a loan.

Plan

- Part I: Theory
  - Part II: Case study in detail

Prediction with qualitative target

- Y is qualitative
  - Whether a debtor defaults on their loan
  - Email is spam or not
  - Game result is win / lose / draw.
- We consider binary (two-class) Y only
  - $Y = 0$ or 1 (yes or no)
  - Class prevalence (p) - frequency of 1.

Prediction with qualitative target

Two different actions

- Predicting probability of $Y = 1$
  - The probability (chance) a debtor will default
- Assigning classes to $Y =$ **classification**
- Need to put target observation in a "class"
  - $\hat{Y}_i = 0$ or $\hat{Y}_i = 1$
- Could be multiple classes, like color
- Today: Y binary

The process

- Predict probability
  - As we have done in DA2/week 5
- Predicted probability between 0 and 1
  - Probability of an event happening
- For each observation we predicted a probability. Often that is it.
- Loss function is Brier score = RMSE
- Sometimes we will go further and classify observations into 0 and 1 = classification

Refresher: Probability Models

- LPM - not this time
- Logit
  - Nonlinear probability models
  - Logit $\Pr[y_i = 1|x_i] = \Lambda \times (\beta_0 + \beta_1 x_i) = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$
  - Predicted probability between 0 and 1
- Logit
  - Starts with a linear combination of the explanatory variables
    - Multiplies them with coefficients, just like linear regression
  - And then transforms that into something
    - That is always between 0 and 1
    - And that thing is the predicted probability.

What's 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

What's NOT new with Binary target?

- Need best fit
- With highest external validity
- Usual worries: overfit
  - Cross-validation helps avoid worst overfit
- Models similar to those used earlier
  - Regression-like models (probability models)
  - Tree-based models (CART, Random Forest)

Theory: Probability prediction

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

Probability prediction process

- Build models
  - several Logit models by domain knowledge
  - LASSO - Logit LASSO
  - CART/Random Forest (discuss later)
- Pick the best model via cross-validation
  - Loss function is Brier score = RMSE
  - Could be other, not today

Classification process

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

Classification process

- Predicted probabilities –> classifications by applying a classification threshold.
  - Below threshold obs –> 0
  - Above threshold obs –> 1
- Picking a threshold is not at all trivial
  - An intuitive threshold is 0.5 – assign class that is more likely.
  - But, maybe not. For rare events, that happen only 10% of the time, maybe 0.1 is better.
- More to come on this....

Classification Table

	$y_j = 0$ 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*)	TN + FN + FP + TP (*N, all observations*)

Classification Table: making errors

	$y_j = 0$ 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*)	TN + FN + FP + TP (*N, all observations*)

Classification Table: making errors

	$y_j = 0$ Actual negative	$y_j = 1$ Actual positive	Total
$\hat{y}_j = 0$ Predicted negative	Predict firm stay (*Firm did stay*)	Predict firm stay (*Firm exited*)	TN + FN (*all classified stay*)
$\hat{y}_j = 1$ Predicted positive	Predict firm exit (*Firm stayed*)	Predict firm exit (*Firm did exit*)	FP + TP (*all classified exit*)
Total	TN + FP (*all actual stay*)	FN + TP (*all actual exit*)	TN + FN + FP + TP (*N, 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 $y = 0$

Theory: The ROC

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

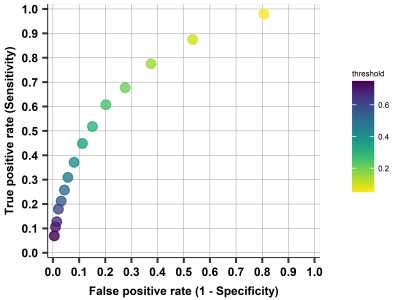
- For each threshold, we can compute confusion table –> calculate sensitivity and specificity
- Show in graph - illustrate (non-linear) trade-off

- ROC curve – choosing a threshold value creates a trade-off between how well a probability prediction leads to correct classification of y = 1 observations versus y = 0 observations.
  - The curve shows this across all possible threshold values.
  - The ROC curve does not show the threshold values themselves.

ROC Curve

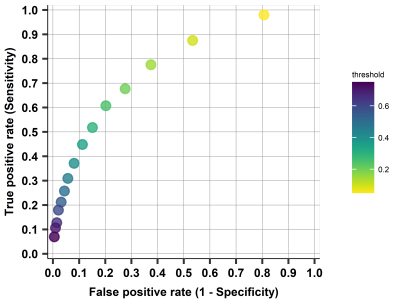
- The *ROC curve* is a popular graphic for simultaneously displaying specificity and sensitivity for all possible thresholds.
  - ROC: Receiver operating characteristic curve
  - Name from engineering
- For each threshold, we can compute confusion table –> calculate sensitivity and specificity
- Show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

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

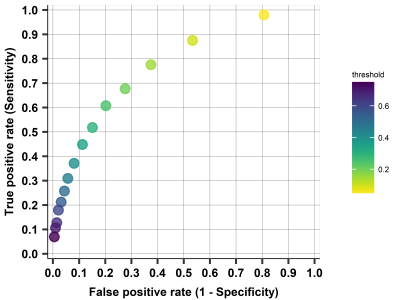
ROC Curve Intuition



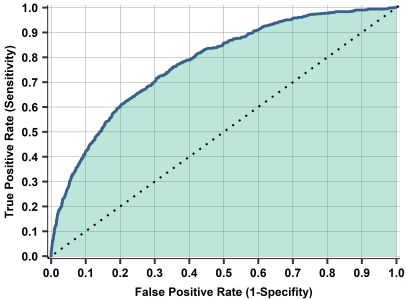
- Neither axis shows the value used for the threshold directly, but both decrease in threshold value.
- ROC curve is for all possible thresholds - many thresholds shown by dots
  - From 0 to 1
- Higher threshold means fewer positives and thus fewer false positives and/or fewer true positives.
- As we lower the threshold, we move to right and up.
- ROC curve – how true positives and false positives increases relative to

ROC Curve Intuition

(a) ROC curve points for various thresholds



(b) Continuous ROC curve



ROC Curve Intuition - the 45 degree line

- Vertical axis: $\Pr[(Correct1|y = 1)]$
- Horizontal axis: $\Pr[(False1|y = 0)]$
- 45 degree line = if classification totally random with true probability p
- Consider a case with $p = 40\%$ and $y = 1$
- all individuals are classified randomly to 1 or 0 with $p = 40\%$ chance
  - $\Pr[(Correct1|y = 1)] = \Pr[(False1|y = 0)] = p = 0.4$
  - Why? $\Pr[(Correct1)] = p$ whether observation has $y = 1$ or $y = 0$
- Any threshold may be applied here as classification is not based on any particular threshold
- That's the 45 degree line.

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 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 = Area Under the ROC Curve
- 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.

Model selection Nr.1: Probability models

- Model selection when we have no loss function, based on probability models
- Predict probabilities
  - No actual classification
- Use predicted probability to calculate RMSE
- Pick by smallest RMSE
  - When users rely on probabilities

- Draw up ROC curve and get AUC
- Pick the model with the largest AUC
  - More frequently used in practice
  - Has nice interpretation
  - Less sensitive to class imbalance
- In practice, AUC is more frequently used

Another argument for AUC

The AUC statistic has other, rather useful interpretation.

- Let us have two classes, 0 and 1, and let us randomly pick two observations belonging to different classes, $x_i = 1$ and $x_j = 0$.
- The AUC is the probability that our classification model will assign a higher probability of being in the class 1 to x_i than it will for x_j.
- For a case when AUC=0.8,
  - we randomly compare two observations (one in class 0, the other in class 1),
  - 4 times out 5, we will assign a higher probability of being in the class 1 to the observation actually in class 1.

Another argument for AUC

- The AUC does not depend on class prevalence (p) in the data.
- This is important - the biggest problem with accuracy as measure has been this: we could get high accuracy by betting on the class with higher frequency.

Theory: Classification, loss function

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?

Classification: how to select the threshold

- We see there is a trade-off
- How to select threshold?
- Majority voting? (50%)
- Match frequency in data (20%)

Classification: select the threshold with loss function

- Find optimal threshold with loss function.
- A loss function is a dollar (euro) value assigned to false positive and false negative.
  - It is actually the ratio of FN/FP that matters.
- Most often the costs of FP and FN are very different.

How to select the threshold

- Find optimal classification threshold with loss function
- Find threshold with lowest expected loss
- Two key inputs: relative prevalence of FP and loss due to errors

$$E[loss] = \Pr[FN] \times loss(FN) + \Pr[FP] \times loss(FP)$$

- How to find best threshold based on loss? Two options
- Formula
- Algorithm

How to select the threshold: Algorithm

- Algorithm looks over all possible thresholds and picks the best option
- Minimizing expected loss

- Technical note
- search for the optimal classification threshold does not look for the smallest expected loss.
- Instead, they search for the threshold that maximizes the probability cost function or the **cost-sensitive Youden index**
- Max J = Min expected loss (See Appendix 17.U2)

How to select the threshold: Formula

- Formula
  - When dataset is "large"
  - When our model has a "good" fit

$$Threshold_{minE(loss)} = \frac{loss(FP)}{loss(FN) + loss(FP)}$$

- In practice
  - Pro: easy to use, often close enough
  - Con: not the best cutoff, especially for smaller data, and poorer model

Model selection Nr.2: Loss function driven

- Model selection process when we have a loss function
- Directly based on classification
  1. Predict probabilities
  2. Use predicted probabilities and loss function to pick optimal threshold
     - Algo or formula
  3. Use that threshold to calculate expected loss
  4. Pick model with smallest expected loss (in 5-fold CV).

Theory: Now, with trees

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.

New loss function

- 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

- Technically two options:
- Probability forest + threshold search with algorithm
- Classification forest + threshold formula

Random forest: probability forest

Probability forest + threshold search / algo

- Predicted probabilities
- Use them to find threshold or use formula to classify
- Aggregates the probability predictions of each tree by averaging them across all trees.
- The model's predicted probabilities are simply these averages.
- For predicting probabilities – this is the version to use.
- For classification – can be used, too, by simply applying the optimal classification threshold to the predicted probabilities.

Random forest: classification forest

Classification forest + threshold formula

- Carries out the classification at the end of each individual tree + aggregates those classifications –> final classification
- Input formula based threshold as tuning parameter
- For predicting probabilities, this is not a good approach.
- For classification, this is the right model

- 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.

Random forest : key technical insight

- Two options yield results that are very close
  - Not the same
  - Both are okay to use

Random forest : key technical insight

- Two options yield results that are very close
  - Not the same
  - Both are okay to use

- Do not use "majority voting"!!!
- Default for R classification random forest is $t = 0.5$. Python is better.
- Loss(FN) = loss(FP) - Called "majority voting"
- Seems convincing. But it's misleading!
  - Loss function could be anything!!!

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

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

Case study

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.

Firm exit case study: business case

- Suppose we work for a consultancy, whose aim is to advise banks on client selection or purchasing managers on supplier selection.
- "We do business with a firm, is this firm going to be around in the near future?" - they may ask.

- Our aim is to predict corporate default - exit from the market.
- That is it.
  - Not more specific.
  - We have to figure out and decide on target, features, etc.

Firm exit case study: **bisnode-firms** dataset

- Firm data
- Many different type of variables
  - Financial
  - Management
  - Ownership
  - Status (HQ)
- 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.

Firm exit case study: Label (target) engineering

- Defining our target.
- In the data, there is no "exit" - we have to define it!
- A firm is operational in year t, but is not in business in $t + 2$.
- The target is hence a binary variable called exit,
  - 1 if the firm exited within 2 years
  - 0 otherwise.
- This definition is broad
  - Defaults / forced exit
  - Orderly closure
  - Acquisitions

Firm exit case study: Sample design

- Look at a cross section
  - Year=2012
  - status_alive=1
  - Keep if established in 2012
- We do not care about all firms. Not very small and very large
  - Below 10 million euros
  - Above 1000 euros
- Hardest call: keep when important variables are not missing
  - Balance sheet like liquid assets
  - Ownership like foreign
  - Industry classification
- End with 19K observation, 20% default rate

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 case study: Features - overview

- We consider key predictors and do feature engineering
- Feature engineering -What's the role, why matter?

Firm exit case study: Features - overview

- Great deal of unknown characteristics
- Dataset is small/medium sized for the task
  - 19K rows and dozens of columns
- Functional form could matter greatly for financial variables

- 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.
  - Plenty of analyst calls.

Firm exit case study: Feature engineering

- Growth rates
  - 1 year growth rate of sales. Log difference.
  - Could use longer time period. Lose observations
  - Should depend on client needs. Maybe: I am interested in 3y+ firms.
- Ownership, management info
  - Keep if well covered, impute some. Could drop additional firms if key vars missing
  - Again, depends on business
- Sometimes simplify (unless big data)
  - Binary: ceo_young = ceo_age_mod<40 & ceo_age_mod>15
  - Industry categories - too many, need merge
  - Foreign ownership - above a threshold
- Numerical variables from balance sheet
  - Check functional form - logs, polynomials

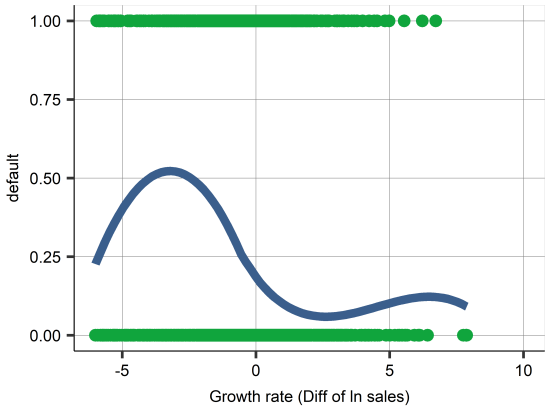
Firm exit case study: Feature engineering tools

- Check coverage
  - Decide on imputation vs drop
- Categorical (factor) variables
  - Tabulate
- Numerical variables
  - Check functional form - logs, polynomials
  - Look at relationships in scatterplot, loess and decide

Firm exit case study: Feature engineering

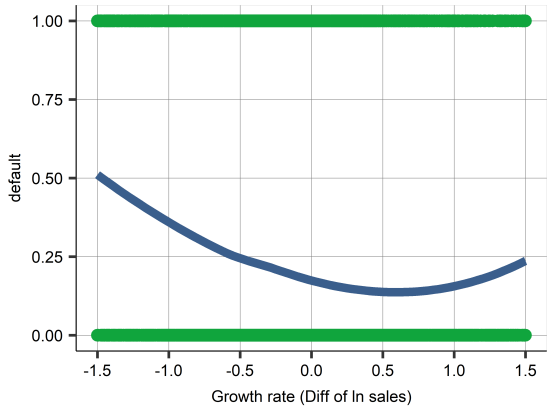
- May need to make cleaning steps.
- Create binary variables (flags) when implementing changes to values.
- When financial values are negative: replace with zero and add a flag to capture imputation.
- Make changes
  - for values that may have additional information in non-linear way
  - Value is exactly 0

Firm exit case study: Firm sales growth



- Annual growth in sales (difference in log sales) vs default
- Weird shape...

Firm exit case study: Firm sales growth

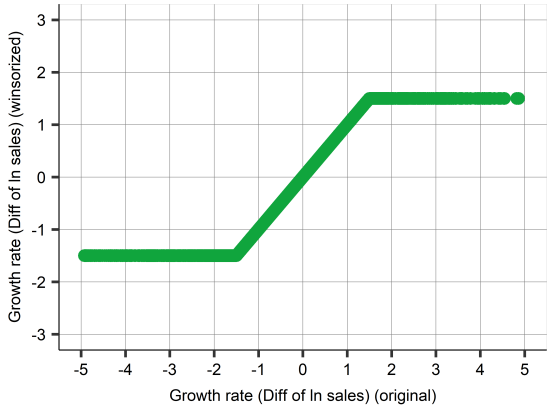


- Annual growth in sales vs default
- Weird shape...
- ... because of extremes, really
- few firms below, say -1.5 and above 1.5
- The rest looks ok

Firm exit case study: Winsorizing

- When edge of a distribution is weird
  - Not just a u-shaped polynomial
  - Domain knowledge helps!
- Winsorizing is a process to keep observations with extreme values in sample
- for each variable, we
  - identify a threshold value, and replace values outside that threshold with the threshold value itself
  - and add a flag variable.
- Two ways to do it:
  - an automatic approach, where the lowest and highest 1 percent or 5 percent is replaced and flagged.
  - Pick thresholds by domain knowledge as well as by looking at lowess. Preferred.

Firm exit case study: Firm sales growth



- The winsorized value simply equals original value in a range and flat below/after.

Case study: 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.

Firm exit case study: 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.

Firm exit case study: 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)

Firm exit case study: Data

- $N = 19,036$
- $N = 15,229$ 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.

Firm exit case study: 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*

Firm exit case study: 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

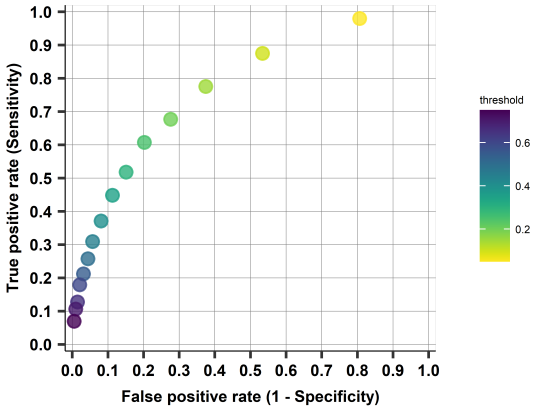
- *5-fold cross-validated on work set, average RMSE*

Will use Logit M4 model as benchmark

Classification

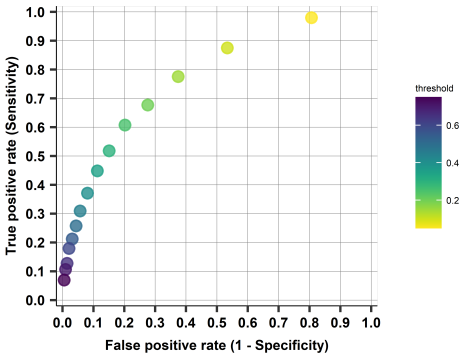
- Picked a model on RMSE/Brier score
- For classification, we will need a threshold

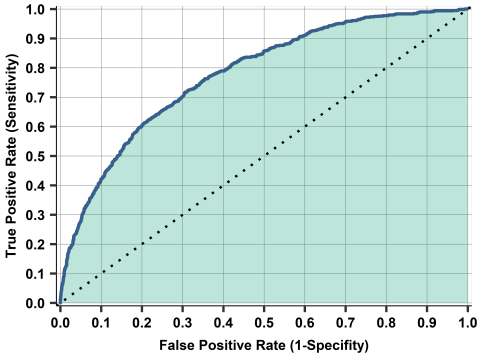
Firm exit case study: ROC curve



- ROC curve shows trade-off for various values of the threshold
  - Go through values of the ROC curve for selected threshold values,
  - between 0.05 and 0.75, by steps of 0.05

Firm exit case study: ROC curves





Firm exit case study: AUC

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

- Can calculate the AUC for all our models
- Model selection by RMSE or AUC
- Here: same (could be different if close)

Firm exit case study: Comparing two thresholds

- Take the Logit M4 model, predict probabilities and use that to classify on the holdout set
- Two thresholds: 50% and 20%
- Predict exit if probability > threshold

Firm exit case study: Comparing two thresholds

- Predict exit if probability > threshold

	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%

Firm exit case study: 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.

Summary

First option: no loss fn

- On the work set, do 5 fold CV and loop over models
  - Do Probability predictions
  - Calculate average RMSE on test for each fold
  - Draw ROC Curve and calculate AUC for each fold
- Pick best model based on avg RMSE
- Take best model and estimate RMSE on holdout–>best guess for live data performance
- Output: probability ranking - most likely to least likely.
- Show ROC curve and confusion table with logit on holdout 4 at $t = 0.5$ and $t = 0.2$ - to illustrate trade-off.

Firm exit case study: The loss function

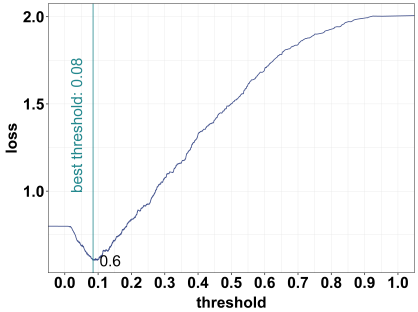
- Loss function = FN, FP
  - What matters is FN/FP
- FN=10
  - If the model predicts staying in business and the firm exits the market (a false negative), the bank loses all 10 thousand euros.
- FP=1
  - If predict exit and the bank denies the loan but the firm stays in business in fact (a false positive),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

Firm exit case study: Finding the threshold

- Find threshold by formula or algo

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

Firm exit case study: 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.*

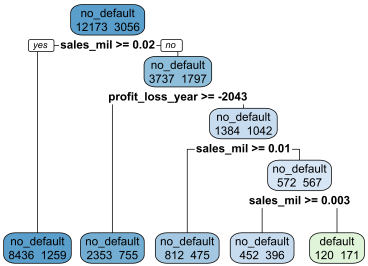
Firm exit case study

- 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.

Firm exit case study:Summary of process with loss function

- On the work set, do 5 fold CV and loop over models
  - Do Probability predictions
  - Calculate average RMSE on each test folds
  - Draw ROC Curve and find optimal threshold with loss function (1,10)
  - show: threshold search - loss plots and ROC curve for fold 5
- Summarize: for each model: average of optimal thresholds, threshold for fold 5, average expected loss, expected loss for fold Nr.5.
- Pick best model based on average expected loss
- Take best model, re-estimate it on work set + find optimal threshold and estimate expected loss on holdout set
- Confusion table on holdout with optimal threshold–>what to expect in live data.

Firm exit case study: CART



- 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

Firm exit case study: 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 (0.642)
- For the random forest we re-estimate the model on work set, and do prediction on holdout set.
  - Holdout RMSE RF is 0.358 (vs 0.366 best logit)
  - Holdout AUC is 0.808 vs 0.784 for best logit.

Firm exit case study: 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%

Firm exit case study: Random Forest

- Note that finding the optimal threshold is rather important.
- 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% - NO!!!!!
  - Don't use it!!!!!!
    - Unless loss function: FN=FP

Repetition for sake of argument

If you don't have a loss function, you can't classify.

Firm exit case study: Random Forest

- No loss function
  - Predict probabilities
- Loss function
  - Predict probabilities
  - Take these probabilities and classify by threshold selected
- Alternative: use threshold and change the classification rule
  - Can be done in caret/ranger

Firm exit case study: Comparing two thresholds

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

	actual stay	actual exit
predicted stay	33.6%	1.3%
predicted exit	45.4%	19.7%

Firm exit case study: 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*

Firm exit case study: 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).

Firm exit case study: Business application

- Consider this setup

- For each firm we review, we get 1000 euros in revenues,
- Loss function: loans to bad companies $= -10,000$ euros,
- missed loans to good ones $= -1,000$ euros.

Firm exit case study: Business application

- Simplest model 1 classifies with expected loss 0.722 euro per firm, the Random Forest model has 0.587 euro.
- Building a better model yields 135 euros higher profit per firm

$$(0.722 - 0.587) \times 1000 = 135$$

- If we do 1000 deals, it is 135,000 euros in profit.
- If a regulator asks for an interpretable model, we shall compare with the logit M4 model and have 103,000 euros in expected profit.
- Why does it matter?

Firm exit case study: Business application

- Random Forest gets us 135K profit, best logit is 103K compared to some simple model.
- We can take this and compare to development costs
- Profit for good analysis.

Summary

- Decide whether the goal is predicting probabilities or classification.
- The outcome of prediction with a binary target variable is always the predicted probabilities as a function of predictors.
- When our goal is probability prediction, we should find the best model that predicts probabilities by cross-validation + RMSE/AUC.
- When our goal is classification, we should find the best model that has the smallest expected loss.
  - With formula for threshold or search algorithm
- Finding the optimal classification threshold needs a loss function.

Summary

- Without a loss function, no classification.
  - If you don't have one, make it up.
  - Don't rely on default 0.5.