Artificial Intelligence in Finance

Lesson 3 – part A

Martina Halousková
508104@mail.muni.cz

Department of Finance, Faculty of Economics and Administration, MU

December 2, 2023

Plan for today

- More information about Midterm.

Part A:

- Classification, Binary logistic regression, Evaluation of binary outcomes (predictions, confusion matrix, ROC curve, loss functions), Data imbalance.
- script lesson3A.R

Part B:

- Regularized logistic regression, Bagging, Tree-based methods: classification trees, Bagging decision trees, Random decision forest.
  [a selection from a longer presentation]
- script lesson3B.R

Midterm

Assignment in this folder. There, you will find:

1. **R script file** with a detailed description of the tasks you should complete and the number of points awarded for each task. The main goal of the MidTerm will be to design a prediction model.
2. csv file with the **dataset**,
3. and an Excel file with a short **description of the data**.

Midterm

Instructions:

1. Upload your solution to this depository vault.
2. The deadline is Sunday, 10.12. 23:59 - you have one week to work on your solutions.
3. You can use all scripts/codes/functions from our lectures - you do not need to create the code entirely from scratch. You can also use Google to help you with coding issues or with any errors.
4. If you have any special requirements, let me know.

This MidTerm will form 30% of your grade.

Please submit your own original solutions and do not copy from other students.

Outline for Section 1

Introduction

Instead of predicting a specific value (on an interval) for a **continuous target** variable, we might want to predict a **qualitative variable** (e.g. color, social status, ...). More broadly, we are interested in **classification problems**:

- The patient is: i) healthy, ii) has a common cold, iii) flue, iv) COVID-19 or v) something else?
- The respondent is willing to vote for candidate: i) No. 1, ii) No. 2., ...
- Is it likely that the company will have financial distress (1 - yes, 0 - no)?
- Is the customer going to buy the product (1 - yes, 0 - no)?
- Is the borrower going to repay the loan (1 - yes, 0 - no)?

Introduction

A specific case of a classification problem is related to **binary decision** (1 - Yes, 0 - No).
Classification **is distinct** from continuous outcome prediction. We have **different models** and a different concepts of **what constitutes a good prediction**. Some methods:

- Logistic regression.
- Penalized logistic regressions:
  - LASSO.
  - RIDGE.
  - Elastic net.
- Tree based methods.
- Support Vector Machines.
- K-Means clustering (sort of).
- Neural networks and other methods...

Outline for Section 2

Probability linear model

Let $Y_i, i = 1, 2, ..., n$ denote a bi-variate outcome $1 - survived, 0 - notsurvived$ sinking of the titanic and X_i the age of the person. The following model is the **probability linear model** and is estimated via OLS:

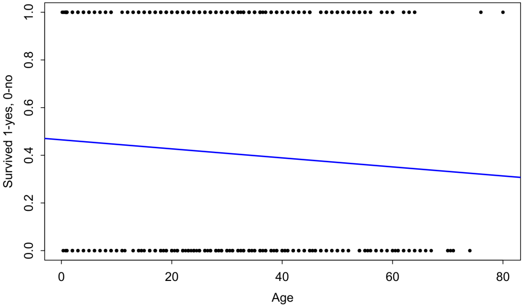
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{1}$$

with following estimates:

$$Y_i = 0.46 - 0.001894 + \hat{\epsilon}_i \tag{2}$$

Probability linear model

The estimated regression line shows you why such a model **might not be the best idea**:



Issues:

- The model is **heteroscedastic almost by design**.
- Predicted values might **fall below 0** and **exceed 1**.

Binary logistic regression

Ideally you might want to model the probability directly. Let $h(\boldsymbol{X}_i;\beta)$ be a **link-function** that includes k features in vector $\boldsymbol{X}$ and corresponding k parameters in vector β. The probability that event happens $Y_i = 1$ is:

$$P(Y_i = 1|\boldsymbol{X}_i;\beta) = h(\boldsymbol{X}_i;\beta) \tag{3}$$

For probability that event will not happen we have:

$$P(Y_i = 0|\boldsymbol{X}_i;\beta) = 1 - P(Y_i = 1|\boldsymbol{X}_i;\beta) = 1 - h(\boldsymbol{X}_i;\beta) \tag{4}$$

We can combine both equations into:

$$P(Y_i|\boldsymbol{X}_i;\beta) = h(\boldsymbol{X}_i;\beta)^{Y_i}(1 - h(\boldsymbol{X}_i;\beta))^{(1-Y_i)} \tag{5}$$

This is a Bernoulli trial.

Binary logistic regression

The Bernoulli trial:

$$P(Y_i|\boldsymbol{X}_i;\beta) = h(\boldsymbol{X}_i;\beta)^{Y_i}(1-h(\boldsymbol{X}_i;\beta))^{(1-Y_i)} \tag{6}$$

assuming **independence** between outcomes, leads to a **Binomial process** and we can combine multiple observations of the outcome into a **likelihood function**:

$$L(\beta) = P(Y|\boldsymbol{X};\beta) = \prod_{i=1}^{n} h(\boldsymbol{X}_i;\beta)^{Y_i}(1-h(\boldsymbol{X}_i;\beta))^{(1-Y_i)} \tag{7}$$

The goal is to find such parameters of β that lead to the highest possible value of the $L(\beta)$. Why?

Binary logistic regression

The maximization process is over parameters β:

$$\max_{\boldsymbol{\beta}} \rightarrow L(\beta) = \prod_{i=1}^{n} h(\boldsymbol{X}_i;\beta)^{Y_i}(1-h(\boldsymbol{X}_i;\beta))^{(1-Y_i)} \tag{8}$$

Instead of working with the product a more convenient method is to use **log-likelihood**:

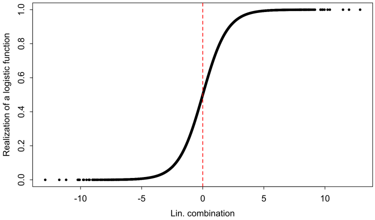
$$\max_{\boldsymbol{\beta}} \rightarrow LL(\beta) = \sum_{i=1}^{n} Y_i log[h(\boldsymbol{X}_i;\beta)] + (1-Y_i) log[(1-h(\boldsymbol{X}_i;\beta))] \tag{9}$$

We have to figure out, how should the $h(.)$ function look like. A popular option is a form of a **sigmoid function**.

Binary logistic regression

Specifically, a popular option is the **logistic function**; hence the **logistic regression**. Let denote $\sum_{j=1}^{k} \beta_j X_{i,j}$ simply as x. The logistic function has a form:

$$P_i = P(Y_i = 1|\boldsymbol{X}_i; \boldsymbol{\beta}) = h(.) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \tag{10}$$



$$\max_{\boldsymbol{\beta}} \rightarrow LL(\boldsymbol{\beta}) = \sum_{i=1}^{n} Y_i \left(\sum_{j=1}^{k} \beta_j X_{i,j} \right) - log\left(1 + e^{\sum_{j=1}^{k} \beta_j X_{i,j}}\right) \tag{11}$$

Binary logistic regression

Recall: In machine learning applications we do not care so much about parameter estimates. Still if you want to interpret coefficients, remember the from $h(.)$ you can get to P_i. A popular approach is to look at **odds**:

$$O_i = \frac{P_i}{1 - P_i} = e^x \tag{12}$$

This looks better, now taking the (natural) log leads to the **logit**:

$$log\left(\frac{P_i}{1 - P_i}\right) = e^x = x \tag{13}$$

and it looks similar to linear regression.

Outline for Section 3

Prediction model

Turning back to the survivors of the sinking of the titanic (yR0lWICH3rY). We use a training sample and consider the following specification (with estimates):

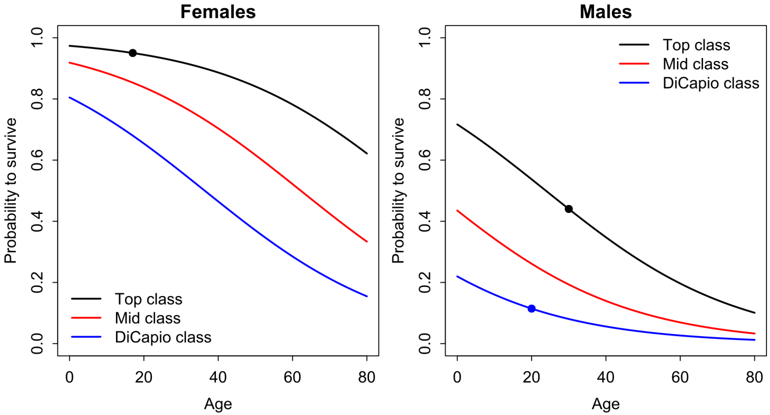
$$\sum_{j=1}^{k} \hat{\beta}_j X_j = -1.27 + 2.19 Top_i + 1.01 Mid_i + 2.68 Female_i - 0.04 Age_i + 0.94 Parent_i$$

- How would you estimate the effect of Age on the probability of surviving?
  - Use logistic regression.
  - The effect is **non-linear** and depends on other variables!

Prediction model

Assuming that the person has following characteristics, Top = 1, Mid = 0, Female = 1, Age = 30, Parent = 0, the probability tu survive is given by:

$$0.92 = \left(1 + e^{-1.27+2.19+2.68-0.04\times 30}\right)^{-1} \tag{14}$$



How accurate are forecasts?

Consider $Y_i = 1$ to be a positive and $Y_i = 0$ a negative outcome. Let's the probability prediction be $\hat{p}_i$ and assume to have a given threshold $p_T \in (0, 1)$ such, that if $\hat{p}_i > p_T \rightarrow \hat{Y}_i = 1$. Given a sample of observations in the testing sample indexed as $i = 1, 2, ..., n$ we can construct the following **confusion matrix**, predictions from *plm* and $p_T = 0.5$:

	Observed $Y_i = 0$	Observed $Y_i = 1$
Predicted $\hat{Y}_i = 0$	118	76
Predicted $\hat{Y}_i = 1$	4	12

- True positives? $TP = 12$.
- True negatives? $TN = 118$.
- False positives? $FP = 4$.
- False negatives? $FN = 76$.

How accurate are forecasts?

	Observed $Y_i = 0$	Observed $Y_i = 1$
Predicted $\hat{Y}_i = 0$	118	76
Predicted $\hat{Y}_i = 1$	4	12

- **Accuracy** $= \frac{TP+TN}{TP+TN+FP+FN} = \frac{118+12}{18+12+4+76} = 0.62$
- **Sensitivity** $=$ Recall $=$ TPR $= \frac{TP}{TP+FN} = \frac{12}{12+76} = 0.14$
- **Specificity** $=$ TNR $= \frac{TN}{TN+FP} = 0.97$
- **Precision** $= \frac{TP}{TP+FP} = \frac{12}{12+4} = 0.75$
- **Balanced accuracy** $= \frac{\text{Sensitivity}+\text{Specificity}}{2} = \frac{0.14+0.97}{2} = 0.55$
- **F1** $= 2 \times \frac{\text{Precision}\times\text{Sensitivity}}{\text{Precision}+\text{Sensitivity}} = 2 \times \frac{0.75\times0.14}{0.75+0.14} = 0.24$

How accurate are forecasts?

Compare the confusion matrix from the *plm* model:

	Observed $Y_i = 0$	Observed $Y_i = 1$
Predicted $\hat{Y}_i = 0$	118	76
Predicted $\hat{Y}_i = 1$	4	12

To the confusion matrix from the logistic regression: *plm* model:

	Observed $Y_i = 0$	Observed $Y_i = 1$
Predicted $\hat{Y}_i = 0$	102	29
Predicted $\hat{Y}_i = 1$	20	59

Which model leads to **better** predictions?

How accurate are forecasts?

Which model leads to **better** predictions? It depends right? Still the differences appear to be substantial:

Models	PLM	LR
Accuracy	0.62	**0.77**
Sensitivity	0.14	**0.67**
Specificity	0.97	0.83
Precision	0.75	**0.74**
Balanced accuracy	0.55	**0.75**
F1 score	0.24	**0.70**

Note, that the threshold $p_T = 0.5$ was set arbitrarily. In fact, it might be considered to be a **hyperparameter** that you need to tune using **cross-validation**.

Changing threshold

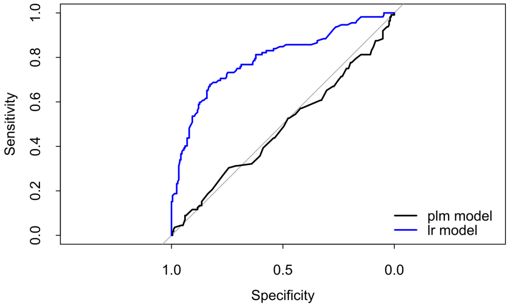
Let's change the threshold to $p_T = 0.45$.

Models	PLM	LR
Accuracy	054	**0.76**
Sensitivity	0.30	**0.69**
Specificity	0.72	0.80
Precision	0.43	**0.72**
Balanced accuracy	0.50	**0.75**
F1 score	0.35	**0.69**

Not an improvement for LR! The model is the same, only the threshold changed.

ROC

The Receiver Operating Characteristic curve displays two types of errors for all possible thresholds (James et al. 2018, [3]).



The overall performance of a classifier across all possible thresholds is the **area under the ROC**, denoted as **AUC**. In cases above $AUC_{plm} = 0.51$ and $AUC_{lr} = 0.84$.

Brier score

There are several popular alternatives to evaluate classification forecasts that can be used in the model confidence set framework as well. The **Brier** (1950, [1]) **score** for two class problems is given by:

$$S_B = n^{-1} \sum_{i=1}^{n} (\hat{p}_i - Y_i)^2 \tag{15}$$

, which is the mean squared error between the predicted probability ($\hat{p}_i$) and the observed outcome (Y_i).

In our examples above we have $S_{B,plm} = 0.24$ and $S_{B,lr} = 0.16$ and only the logistic regression model is in the set of superior models.

Cross entropy

The **Cross-entropy** is likely the most popular measure for classification purposes. For two class problems it is given by:

$$S_E = n^{-1} \sum_{i=1}^{n} - [log(\hat{p}_i)Y_i + log(1 - \hat{p}_i)(1 - Y_i)] \tag{16}$$

The two terms are switched on/off depending on whether the observed event happened or not. **You get penalized if you are confident and wrong**.

In our examples above we have $S_{E,plm} = 0.68$ and $E_{B,lr} = 0.48$ and only the logistic regression model is in the set of superior models.

Finance related cost functions

A threshold and loss functions should be driven by the **domain knowledge**. The mapping $D : \hat{p}_i \to \hat{Y}_i, \hat{Y}_i \in \{0, 1\}$ should not be driven by purely statistical measures.

Consider a loan market with three participants, lender, borrower and investor. Lender and investor are designing credit-scoring models.

- What should be the criterion for the lender?
- What should be the criterion for the investor?

Outline for Section 4

Intuition

In the titanic dataset, 40.82% survived. This is **not overly imbalanced**. However, using the Zopa dataset ('zsnew.csv'), we have only 8.155% of defaulted loans. This is a severaly imbalanced dataset, where the majority class (good loans) has significant representation in the data.

Imbalanced data might lead to **accuracy paradox**. Say you predict a stock to default in the next year. You have 99.5% of firms that have not defaulted (**majority** class) and only 0.05% that have (**minority** class):

- How accurate is a prediction that will unconditionally always predict a non-default (i.e. 0)?
- Your model will have a tendency to learn from mostly successful companies that are over-represented in the sample.
- The accuracy of the model is likely to reflect the underlying distribution imbalance.

Intuition

Possible solutions:

- **Under-sampling** the majority class.
- **Over-sampling** the minority class.
- **Under**-sampling the majority **and Over**-sampling the minority class.
- Use **cost weighted learning** - more weight given to the minority class.
- Use synthetic minority over-sampling technique (SMOTE) of Chawla et al., (2002, [2]).
- Adjustment of the decision threshold p_T.
- Instance hardness threshold of Smith et al., (2014, [5]).
- balance cascade of Liu et al., (2009, [4]).

[1] Glenn W Brier et al. "Verification of forecasts expressed in terms of probability". In: *Monthly weather review* 78.1 (1950), pp. 1–3.

[2] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[3] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[4] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. "Exploratory undersampling for class-imbalance learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2008), pp. 539–550.

[5] Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. "An instance level analysis of data complexity". In: *Machine learning* 95.2 (2014), pp. 225–256.

MUNI
ECON

Artificial Intelligence in Finance

Lesson 2 – part B

Martina Halousková
508104@mail.muni.cz

Department of Finance, Faculty of Economics and Administration, MU

December 2, 2023

Outline for Section 1

Introduction

The logistic regression is a **popular benchmark**:

- It is simple to interpret.
- It is fast to estimate - no tuning required.
- It leads to **nonlinearities** along and across features.
- You can still attempt to **enhance** the model via:
  - Feature transformations (although difficult to interpret see Mood (2010, [3]).
  - Feature interactions.
  - Bagging.

There are some **limitations** if many features are included:

- Estimation uncertainty - too many parameters.
- **Over-fitting** is likely.

Introduction

Following the regularization approach for regression one can **adjust the logistic regression** as well:

- LASSO logistic regression (LLR).
- RIDGE logistic regression (RLR).
- Elastic Net logistic regression (ENLR).
- Complete subset logistic regression.

Outline for Section 2

LASSO logistic regression

Recall that the **log-likelihood to maximize** for logistic regression is given by:

$$\max_{\boldsymbol{\beta}} \rightarrow LL(\boldsymbol{\beta}) = \sum_{i=1}^{n} Y_i \left(\sum_{j=1}^{k} \beta_j X_{i,j} \right) - log\left(1 + e^{\sum_{j=1}^{k} \beta_j X_{i,j}}\right) \tag{1}$$

After adding the **penalty term**, the expression becomes:

$$\begin{aligned} \max_{\boldsymbol{\beta}} \rightarrow LL(\boldsymbol{\beta}) &= \left[\sum_{i=1}^{n} Y_i \left(\sum_{j=1}^{k} \beta_j X_{i,j} \right) - log\left(1 + e^{\sum_{j=1}^{k} \beta_j X_{i,j}}\right) \right] \\ & -\lambda \sum_{j=1}^{k} |\beta_j| \end{aligned} \tag{2}$$

LASSO logistic regression

As before the λ parameter needs to be estimated $\rightarrow$ **cross-validation**:

$$\max_{\boldsymbol{\beta}} \rightarrow LL(\boldsymbol{\beta}) = \left[\sum_{i=1}^{n} Y_i \left(\sum_{j=1}^{k} \beta_j X_{i,j}\right) - log\left(1 + e^{\sum_{j=1}^{k} \beta_j X_{i,j}}\right)\right] - \lambda \sum_{j=1}^{k} |\beta_j| \tag{3}$$

What classification **accuracy** measure **to use**?

LASSO logistic regression

What classification **accuracy** measure **to use**?

- **Deviance** (cross-entropy):

$$D_i = -2\left[log(\hat{p}_i)Y_i + log(1-\hat{p}_i)(1-Y_i)\right] \tag{4}$$

- AUC - does not require explicit threshold.
- **Custom based**:
  - Profit, revenue.
  - Balanced accuracy.
  - Precision (depends)....

RIDGE logistic regression

Using the **penalty term** from RIDGE leads to:

$$\max_{\boldsymbol{\beta}} \rightarrow LL(\boldsymbol{\beta}) = \left[\sum_{i=1}^{n} Y_i \left(\sum_{j=1}^{k} \beta_j X_{i,j}\right) - log\left(1 + e^{\sum_{j=1}^{k} \beta_j X_{i,j}}\right)\right] - \lambda \sum_{j=1}^{k} \beta_j^2 \quad (5)$$

Elastic Net logistic regression

Combining the LASSO and RIDGE **penalty terms** leads to:

$$\max_{\boldsymbol{\beta}} \rightarrow LL(\boldsymbol{\beta}) = \left[\sum_{i=1}^{n} Y_i \left(\sum_{j=1}^{k} \beta_j X_{i,j}\right) - log\left(1 + e^{\sum_{j=1}^{k} \beta_j X_{i,j}}\right)\right] - \lambda \left[\frac{1-\alpha}{2} \sum_{j=1}^{k} \beta_j^2 + \alpha \sum_{j=1}^{k} |\beta_j|\right] \quad (6)$$

Apart from λ we need to estimate or assume $\alpha \in (0, 1)$ as well.

Titanic dataset

How are regularization methods doing in our datasets?
Titanic dataset:

- Predicting the survival of a passenger.
- No imbalance procedures.
- Threshold set to 0.427 the proportion of survived passengers.
- 1046 obs. originally, 836 in mildly unbalanced training and 210 in testing dataset with 5 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.828	0.743	0.786
LLR	0.829	0.743	0.786
RLR	0.814	0.750	0.782
ENLR	0.829	0.743	0.786

P2P Zopa dataset

How are regularization methods doing in our datasets?
Zopa dataset:

- Predicting the default of a loan.
- 8% bad loans $\rightarrow$ unbalanced data!
- We have enough 20000 observations $\rightarrow$ under-sample the majority.
- 2609 in training and 653 in testing data with 160 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.73	0.75	0.74
LLR	0.68	0.82	0.75
RLR	0.70	0.76	0.73
ENLR	0.68	0.82	0.75

Firm defaults

How are regularization methods doing in our datasets?
Firm defaults dataset:

- Predicting the default of a firm in next period.
- Random under-sampling of majority and random over-sampling of minority.
- 6819 obs. originally, 1400 in balanced training and 1364 in testing dataset and 91 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.73	0.87	0.80
LLR	0.82	0.86	0.84
RLR	0.82	0.85	0.84
ENLR	0.82	0.86	0.84

Outline for Section 3

Bagging revisited

Bagging is a based on estimating a model on a **bootstrapped** sample. That is, we create a *new* dataset by randomly selecting (with replacement) observations from the original dataset. Repeating the bootstrapping (sampling) $\rightarrow$ estimation $\rightarrow$ prediction sample B times, leads to a **distribution** of predictions for every testing observation i. Specifically, i.e. $\hat{p}_{i,b}, i = 1, 2, ...; b = 1, 2, ..., B$.

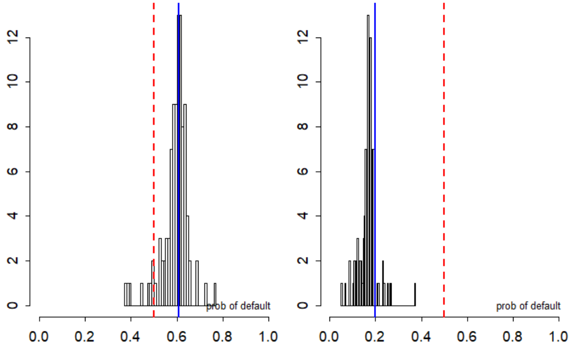
This is especially interesting for **classification** tasks. Why?

Bagging revisited

This is specifically interesting for **classification** tasks. Why?

- Bagging can improve prediction accuracy - **averaging many over-fitted models**.
- It gives us an estimate of **confidence in our predictions**.

The P2P loan default:



P2P loan

As an investor, you invest if $\hat{Y}_i = 0$: Using LASSO you face:

	Observed $Y_i = 0$	Observed $Y_i = 1$
Predicted $\hat{Y}_i = 0$	262	105
Predicted $\hat{Y}_i = 1$	59	227

This leads to a 71.39% success across 367 loans. Let's be more conservative and:

- Invest **only** into loans, where the 95^{th} quantile of the predicted probability is below the threshold, i.e. $\sum_{b=1}^{B} I(\hat{p}_{i,b} > 0.5) \leq 0.95$.

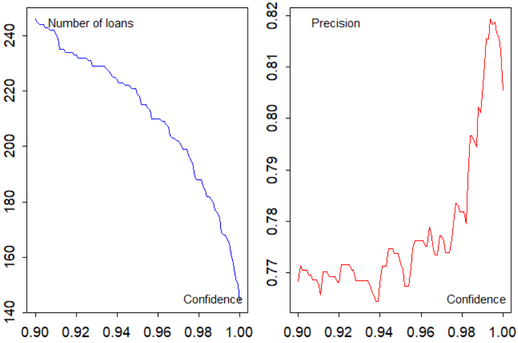
The confusion matrix investor faces is:

	Observed $Y_i = 0$	Observed $Y_i = 1$
Predicted $\hat{Y}_i = 0$	169	50
Predicted $\hat{Y}_i = 1$	0	0

The success goas to 77.16% but **only** across 219 loans.

P2P loan

The higher the confidence the higher should be negative predicted value - **there is a price we pay**.



Does that mean that loan market is **inefficient** see Lyócsa and Výrost (2018, [2])? Not necessarily.

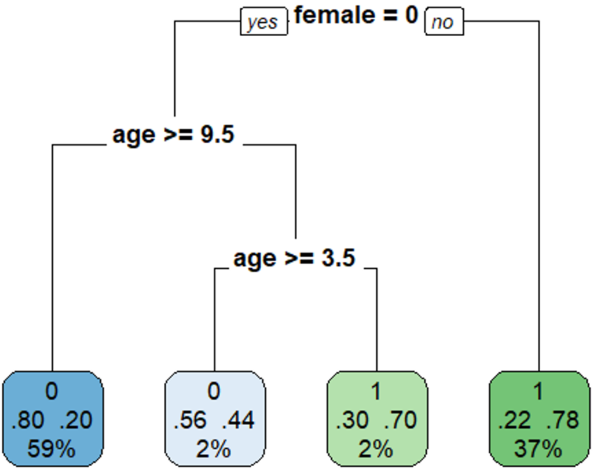
Outline for Section 4

Introduction

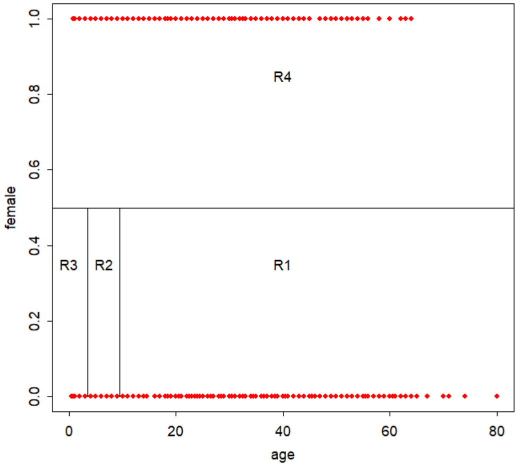
- Involve **stratifying a feature space** into simpler regions - subsets of data.
- Prediction for a specific observation from the testing sample is usually the **most occurring class** in the **terminal region**.
- Simple decision trees can be improved via:
  - pre-prunning.
  - post-prunning.
  - bagging.
  - bagging and randomization - random forest.
  - boosting.

Example

Shallow tree

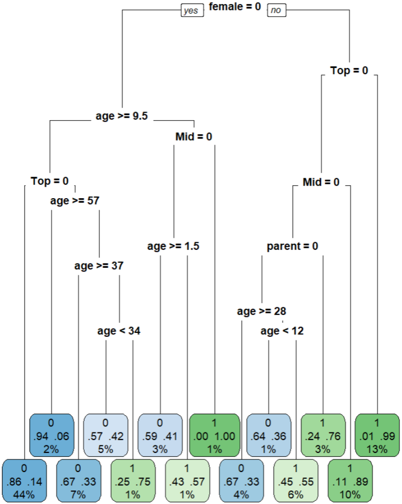


Example



Example

Deeper tree



Splitting the nodes

How do we **find splitting points**?

To split a node (t), measures employed are based on **degree of impurity** of a node(s). Highest impurity is (0.5, 0.5), lowest is (0.0, 1.0) or (1.0, 0.0), i.e. the smaller the degree of impurity, the more **skewed** the **class distribution** (Tan et al., 2016, [4]).

Let $p(c|t)$ denote the proportion of observations of class c in node t:

- **Classification error** for given node (t):

$$I_{CE} = 1 - \max_c p(c|t) \tag{7}$$

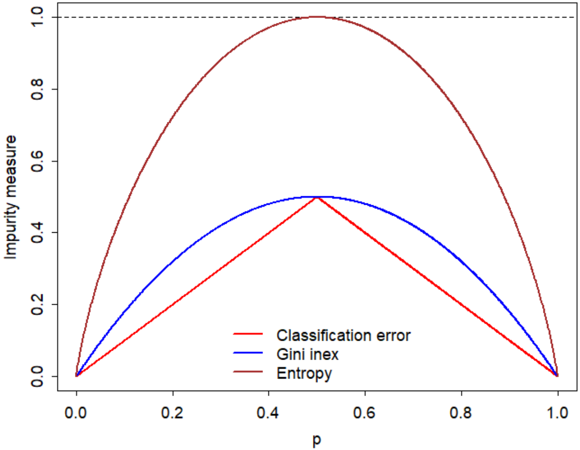
- **Gini index** (we will use) for node (t):

$$I_{CE} = 1 - \sum_c p(c|t)^2 \tag{8}$$

- **Entropy** for node (t):

$$I_{CE} = -\sum_c p(c|t) log_2 p(c|t) \tag{9}$$

Splitting the nodes



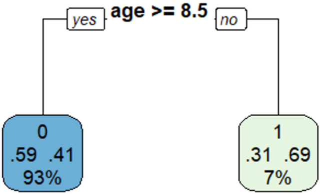
Splitting the nodes

Similarly as for regression trees, in **decission trees** the ultimate goal is to find terminal regions - $R_1, R_2, ..., R_J$ that minimize some loss functions (James et al., 2013, [1]). As before, a greedy approach is used, mostly the the **recursive binary splitting** algorithm.

A split is decided by **comparing the degree of impurity of the parent node with child nodes**. Let k be number of classes (2 for **binary splits**), N_j number of observations in child node j and $I(t_j)$ the impurity measure of child note j. The goal is to find a split that maximizes:

$$\Delta = I(t) - \sum_{j=1}^{k} \frac{I(t_j)N_j}{N} \tag{10}$$

Splitting the nodes - Example



$$\Delta = I(t) - \sum_{j=1}^{k} \frac{I(t_j)N_j}{N}$$

$$0.489 - \frac{0.429 \times 61}{836} - \frac{0.482 \times 775}{836} = 0.0107$$

Prunning

The **pre-pruning** approach (early stopping rules):

- Limit the **maximum depth** of the tree.
- Set a **minimum number** needed to consider a **split**.
- Set a **minimum number** of observations in a terminal **region** (bucket size).

The **post-pruning** approach (bottom-up from a deep tree):

- Introducing penalization for too complex trees.

Outline for Section 5

Bagging for decision trees

Recall that bagging is based on the idea that averaging unbiased but potentially over-fitted model's predictions will reduce the out-of-sample error.

Let's have data denoted as Z with $i = 1, 2, ..., N$ observations. In a non-parametric bootstrap:

1. Each observation in Z has the same probability of being selected.
2. **Randomly** select N observations from Z, **with replacement** and create a new dataset Z^*.
3. Estimate a given model/statistics using the dataset Z^*.
4. Repeat step 2 and 3 until we have B models/statistics.

Bagging: Introduction

Using data from the training sample, for each bootstrap sample you estimate a complete (deep) tree T^{*b} and generate a corresponding forecast for observation i that belongs to the testing dataset, $\hat{p}_i^{*,b} \in (0,1)$. The prediction using **bagging** is given by a simple average:

$$\hat{p}_i = B^{-1} \sum_{b=1}^{B} \hat{p}_i^{*,b} \tag{11}$$

This approach should work well for deep trees - why? Predictions from such trees have **low bias but high variance**.

Titanic dataset

How are regularization methods doing in our datasets?
Titanic dataset:

- Predicting the survival of a passenger.
- No imbalance procedures.
- Threshold set to 0.427 the proportion of survived passengers.
- 1046 obs. originally, 836 in mildly unbalanced training and 210 in testing dataset with 5 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.828	0.743	0.786
LLR	0.829	0.743	0.786
RLR	0.814	0.750	0.782
ENLR	0.829	0.743	0.786
DC-BAG	0.757	0.821	0.789

P2P Zopa dataset

How are regularization methods doing in our datasets?
Zopa dataset:

- Predicting the default of a loan.
- 8% bad loans → unbalanced data!
- We have enough 20000 observations → under-sample the majority.
- 2609 in training and 653 in testing data with 160 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.73	0.75	0.74
LLR	0.68	0.82	0.75
RLR	0.70	0.76	0.73
ENLR	0.68	0.82	0.75
DC-BAG	0.80	0.75	0.78

Firm defaults

How are regularization methods doing in our datasets?
Firm defaults dataset:

- Predicting the default of a firm in next period.
- Random under-sampling of majority and random over-sampling of minority.
- 6819 obs. originally, 1400 in balanced training and 1364 in testing dataset and 91 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.73	0.87	0.80
LLR	0.82	0.86	0.84
RLR	0.82	0.85	0.84
ENLR	0.82	0.86	0.84
DC-BAG	0.82	0.90	0.86

Outline for Section 6

Random decision forest

Similarly as with random forest for regressions, random forest combines bagging with a random selection of features to consider at each split $\rightarrow$ **decorrelated trees**. Key parameters to hyper-tune:

- Depth of the trees (should be deep).
- Number of random features selected at each split.
- Number of trees.

Other pre-prunning parameters can be hyper-tuned as well.

Titanic dataset

How are regularization methods doing in our datasets?

Titanic dataset:

- Predicting the survival of a passenger.
- No imbalance procedures.
- Threshold set to 0.427 the proportion of survived passengers.
- 1046 obs. originally, 836 in mildly unbalanced training and 210 in testing dataset with 5 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.828	0.743	0.786
LLR	0.829	0.743	0.786
RLR	0.814	0.750	0.782
ENLR	0.829	0.743	0.786
DC-BAG	0.757	0.821	0.789
RF	0.814	0.778	0.796

P2P Zopa dataset

How are regularization methods doing in our datasets?

Zopa dataset:

- Predicting the default of a loan.
- 8% bad loans $\rightarrow$ unbalanced data!
- We have enough 20000 observations $\rightarrow$ under-sample the majority.
- 2609 in training and 653 in testing data with 160 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.73	0.75	0.74
LLR	0.68	0.82	0.75
RLR	0.70	0.76	0.73
ENLR	0.68	0.82	0.75
DC-BAG	0.80	0.75	0.78
RF	0.80	0.72	0.76

Firm defaults

How are regularization methods doing in our datasets?
Firm defaults dataset:

- Predicting the default of a firm in next period.
- Random under-sampling of majority and random over-sampling of minority.
- 6819 obs. originally, 1400 in balanced training and 1364 in testing dataset and 91 **features**.

Models	Sensitivity	Specificity	Balanced Acc
LR	0.73	0.87	0.80
LLR	0.82	0.86	0.84
RLR	0.82	0.85	0.84
ENLR	0.82	0.86	0.84
RF	0.78	0.93	0.85

[1] Gareth James et al. *An introduction to statistical learning*. Springer, 2013.

[2] Štefan Lyócsa and Tomáš Vrost. "To bet or not to bet: a reality check for tennis betting market efficiency". In: *Applied Economics* 50.20 (2018), pp. 2251–2272.

[3] Carina Mood. "Logistic regression: Why we cannot do what we think we can do, and what we can do about it". In: *European sociological review* 26.1 (2010), pp. 67–82.

[4] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

Thank You for Your Attention!

MASARYK
UNIVERSITY