Exercise collection – Supervised Classification

Contents

Lecture exercises **1**

Questions from past exams **9**

Ideas & exercises from other sources **17**

Lecture exercises

Exercise 1: logistic regression – thresholding

In logistic regression, we estimate the probability $\pi(\mathbf{x}) = \mathbb{P}(y = 1 \mid \mathbf{x})$. To decide if $\hat{y}$ is 0 or 1, we follow:

$$\hat{y} = 1 \quad \Leftrightarrow \quad \hat{\pi}(\mathbf{x}) \geq a$$

a) What happens if you are choosing $a = 0.5$? More precisely, from which value of $\boldsymbol{\theta}^T\mathbf{x}$ do you predict $\hat{y} = 1$ rather than $\hat{y} = 0$?

b) Explain (using words) why $a = 0.5$ is a sensible threshold.

Solution 1:

(a) For a binary classification problem the model can be written as:

$$\hat{y} = 1 \quad \Leftrightarrow \quad \pi(\mathbf{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T\mathbf{x})} \geq a$$

This can be reformulated, s.t. for $a \in (0,1)$

$$\begin{aligned}
&\frac{1}{1+\exp(-\boldsymbol{\theta}^T\mathbf{x})} \geq a \\
\Leftrightarrow\ & 1+\exp(-\boldsymbol{\theta}^T\mathbf{x}) \leq a^{-1} \\
\Leftrightarrow\ & \exp(-\boldsymbol{\theta}^T\mathbf{x}) \leq a^{-1} - 1 \\
\Leftrightarrow\ & -\boldsymbol{\theta}^T\mathbf{x} \leq \log(a^{-1} - 1) \\
\Leftrightarrow\ & \boldsymbol{\theta}^T\mathbf{x} \geq -\log(a^{-1} - 1)
\end{aligned}$$

For $a = 0.5$ we get:

$$\hat{y} = 1 \quad \Leftrightarrow \quad \boldsymbol{\theta}^T\mathbf{x} \geq -\log(0.5^{-1} - 1) = -\log(2-1) = -\log(1) = 0$$

This means the linear decision boundary is defined by a hyperplane equation, i.e., $\boldsymbol{\theta}^T\mathbf{x} = 0$ and it divides the input space in the "1"-space ($\boldsymbol{\theta}^T\mathbf{x} \geq 0$) and in the "0"-space ($\boldsymbol{\theta}^T\mathbf{x} < 0$).

(b) When the threshold $a = 0.5$ is chosen, the losses of missclassified observations, i.e., $L(\hat{y} = 0|y = 1)$ and $L(\hat{y} = 1|y = 0)$, are weighted equally. This means $a = 0.5$ is a sensible threshold if one does not need to avoid one type of misclassification more than the other.
Intuitively it makes sense to cut off at 0.5 because, if the probability for 1 is closer to 1 than to 0, we would intuitively choose 1 rather than 0.

Exercise 2: logistic regression – link function and likelihood

a) What is the relationship between softmax $\pi_k(x) = \frac{\exp(\theta_k^T x)}{\sum_{j=1}^{g}\exp(\theta_j^T x)}$ and the logistic function $\pi(\mathbf{x}) = \frac{1}{1+\exp(-\theta^T x)}$ for $g = 2$ (binary classification)?

b) The likelihood function of a multinomially distributed target variable with g target classes is given by

$$\mathcal{L}_i = \mathbb{P}(Y^{(i)} = y^{(i)}|x^{(i)}, \theta_1, \ldots, \theta_g) = \prod_{j=1}^{g} \pi_j(x^{(i)})^{\mathbb{1}_{\{y^{(i)}=j\}}}$$

where the posterior class probablities $\pi_1(x), \ldots, \pi_g(x)$ are modeled with softmax regression. Derive the likelihood function of n such independent target variables. How can you transform this likelihood function into an empirical risk function?
Hints:

- By following the maximum likelihood principle, we should look for parameters $\theta_1, \ldots, \theta_g$, which maximize the likelihood function.
- The expressions $\prod \mathcal{L}_i$ and $\log \prod \mathcal{L}_i$ (if this expression is defined) are maximized by the same parameters.
- The empirical risk is a *sum* of loss function values, not a *product*.
- Minimizing a scalar function multiplied with -1 is equivalent to maximizing the original function.

State the associated loss function.

c) Explain how the predictions of softmax regression (multiclass classification) look like (probabilities and classes) and define the parameter space.

Solution 2:

a) $\pi_1(x) = \frac{\exp(\theta_1^T x)}{\exp(\theta_1^T x)+\exp(\theta_2^T x)}$

$\pi_2(x) = \frac{\exp(\theta_2^T x)}{\exp(\theta_1^T x)+\exp(\theta_2^T x)}$

$\pi_1(x) = \frac{1}{(\exp(\theta_1^T x)+\exp(\theta_2^T x))/\exp(\theta_1^T x)} = \frac{1}{1+\exp(-\theta^T x)}$ where $\theta = \theta_1 - \theta_2$ and $\pi_2(x) = 1 - \pi_1(x)$

b) When using softmax regression the posterior class probability for the class k is modeled, s.t.

$$\pi_k(x) = \frac{\exp(\theta_k^T x)}{\sum_{j=1}^g \exp(\theta_j^T x)}.$$

A single observation is multinomially distributed, i.e.,

$$\mathcal{L}_i = \mathbb{P}(Y^{(i)} = y^{(i)} | x^{(i)}, \theta_1, \ldots, \theta_g) = \prod_{j=1}^g \pi_j(x^{(i)})^{\mathbb{1}_{\{y^{(i)}=j\}}}.$$

Under the assumption that the observations are conditionally independent the likelihood of the data can be expressed, s.t.

$$\mathcal{L} = \mathbb{P}(Y^{(1)} = y^{(1)}, \ldots, Y^{(n)} = y^{(n)} | x^{(1)}, \ldots, x^{(n)}, \theta_1, \ldots, \theta_g) = \prod_{i=1}^n \prod_{j=1}^g \pi_j(x^{(i)})^{\mathbb{1}_{\{y^{(i)}=j\}}}.$$

(By following the maximum likelihood principle, we should look for parameters $\theta_1, \ldots, \theta_g$, which maximize the expression above.)
Now we want the empirical risk to be a *sum* of loss function values, not a *product* recall:

$$\mathcal{R}_{\text{emp}} = \sum_{i=1}^n L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right).$$

We can turn the product into a sum by taking its log since the same parameters maximize the expressions. Finally, we convert the maximization into minimization by multiplying with -1. So we end up with the so-called cross-entropy loss function:

$$L(y, f(\mathbf{x})) = -\sum_{j=1}^g \mathbb{1}_{\{y=j\}} \log[\pi_j(x)].$$

We see that for the softmax regression the loss function is equal to the negative log-likelihood of one observation. Thus the associated empirical risk is the negative log-likelihood of the complete data set.

c) Since the subtraction of any fixed vector from all θ_k does not change the prediction, one set of parameters is "redundant". Thus we set $\theta_g = (0, \ldots, 0)$. Hence for g classes we get $g-1$ discriminant functions from the softmax $\hat{\pi}_1(x), \ldots, \hat{\pi}_{g-1}(x)$ which can be interpreted as probability. The probability for class g can be calculated by using $\hat{\pi}_g = 1 - \sum_{k=1}^{g-1} \hat{\pi}_k(x)$. To estimate the class we are using majority vote:

$$\hat{y} = \underset{k}{\arg\max}\, \hat{\pi}_k(x)$$

The parameter of the softmax regression is defined as parameter matrix where each class has its own parameter vector θ_k, $k \in \{1, \ldots, g-1\}$:

$$\theta = [\theta_1, \ldots, \theta_{g-1}]$$

Exercise 3: decision boundaries with `mlr3`

Choose some of the classifiers already introduced in the lecture and visualize their decision boundaries for relevant hyperparameters. Use `mlbench::mlbench.spirals` to generate data and use `plot_learner_prediction` for visualization. To refresh your knowledge about `mlr3` you can take a look at `https://mlr3book.mlr-org.com/basics.html`.

Solution 3:

See R code

Exercise 4: k-NN classification

Let the 2D feature vectors in the following figure be with two different class labels (triangles and circles). Classify the point (7,6) - represented by a square in the picture - with a k-nearest neighbor classifier. Distance function should be the L_1 norm (Manhattan distance):

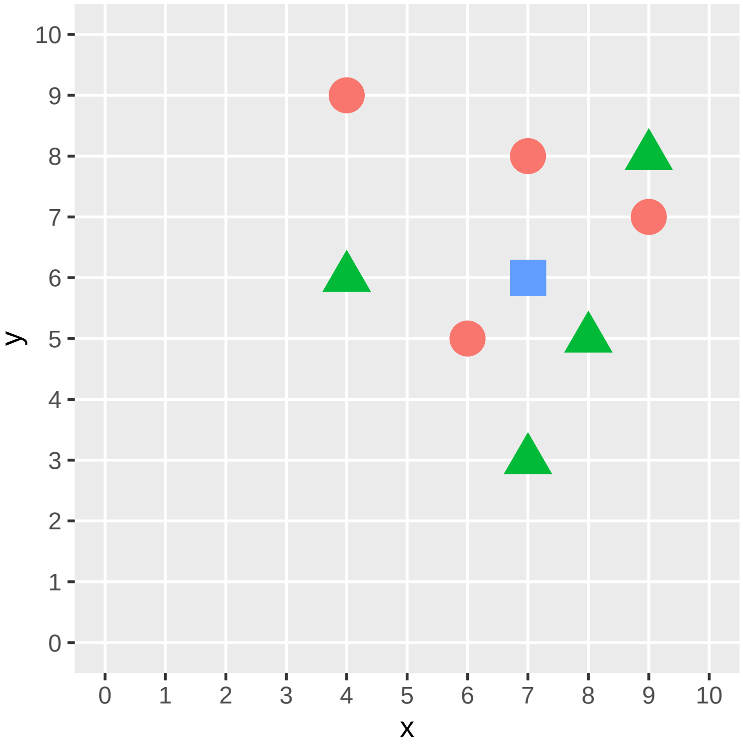
$$d_{\text{manhattan}}(x, \tilde{x}) = \sum_{j=1}^{p} |x_j - \tilde{x}_j|$$

As a decision rule, use the unweighted number of the individual classes in the k-next Neighbor Quantity, i. e. the point is assigned to the class that represents most k-nearest neighbors.

a) $k = 3$

b) $k = 5$

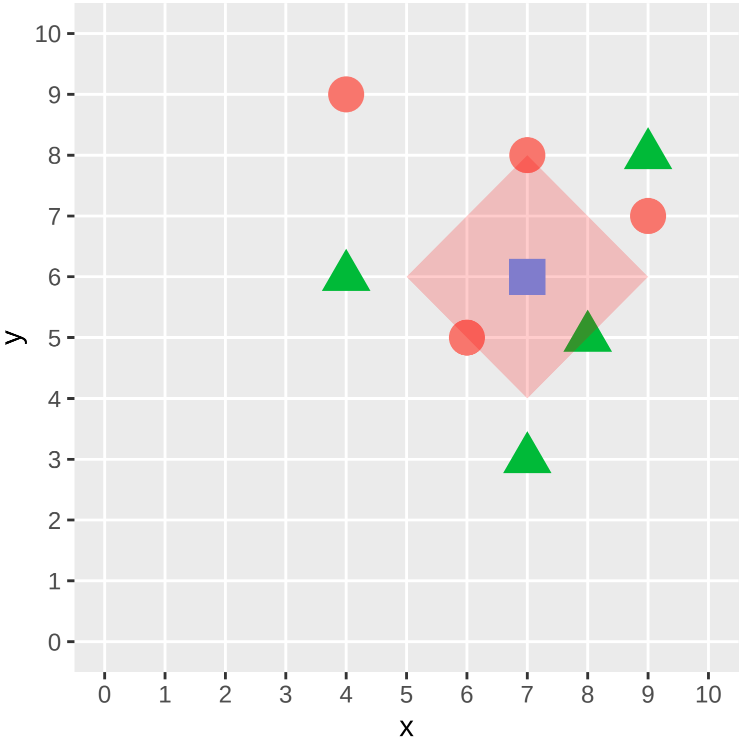
c) $k = 7$



Solution 4:

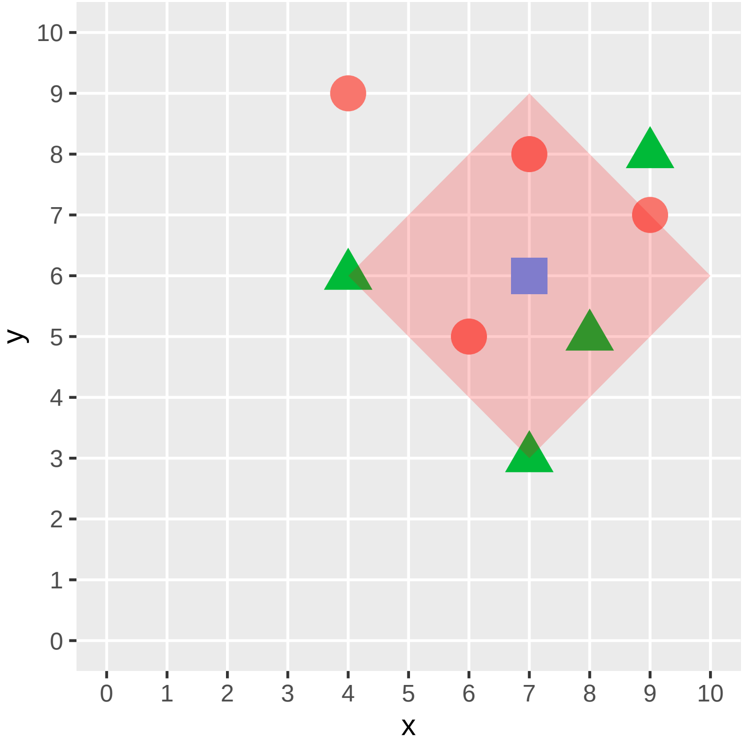
a) $k = 3$

2 circles and 1 triangle, so our point is also a circle



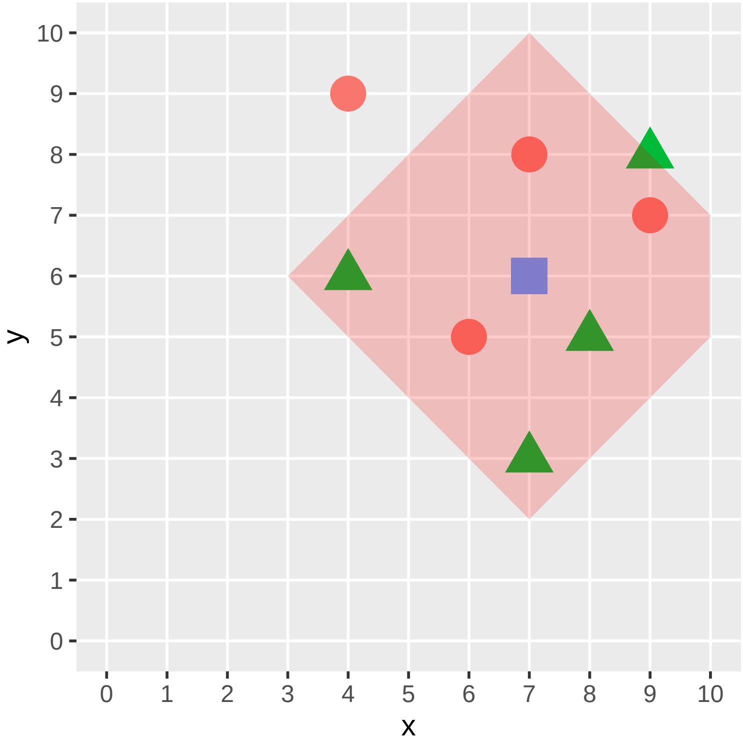
b) $k = 5$

3 circles and 3 triangles, we have to specify beforehand what to do in case of a tie



c) $k = 7$

3 circles and 4 triangles, so our point is also a triangle



Exercise 5: naive Bayes

You are given the following table with the target variable `Banana`:

ID	Color	Form	Origin	Banana ?
1	yellow	oblong	imported	yes
2	yellow	round	domestic	no
3	yellow	oblong	imported	no
4	brown	oblong	imported	yes
5	brown	round	domestic	no
6	green	round	imported	yes
7	green	oblong	domestic	no
8	red	round	imported	no

a) We want to use a naive Bayes classifier to predict whether a new fruit is a Banana or not. Calculate the posterior probability $\pi(x)$ for a new observation $(\text{yellow}, \text{round}, \text{imported})$. How would you classify the object?

b) Assume you have an additional feature "Length", which measures the length in cm. Describe in 1-2 sentences how you would handle this numeric feature with Naive Bayes.

Solution 5:

a) When using the naive Bayes classifier, the features $x := (x_{\text{Color}}, x_{\text{Form}}, x_{\text{Origin}})$ given the category $y \in \{\text{yes}, \text{no}\}$ are assumed to be conditionally independent of each other, s.t.

$$p((x_{\text{Color}}, x_{\text{Form}}, x_{\text{Origin}})|y = k) = p(x_{\text{Color}}|y = k) \cdot p(x_{\text{Form}}|y = k) \cdot p(x_{\text{Origin}}|y = k).$$

For the posterior probabilities $\pi_k(x)$ it holds that

$$\begin{aligned} \pi_k(x) &\propto \underbrace{\pi_k \cdot p(x_{\text{Color}}|y = k) \cdot p(x_{\text{Form}}|y = k) \cdot p(x_{\text{Origin}}|y = k)}_{=:\alpha_k(x)} \\ &\iff \exists c \in \mathbb{R} : \pi_k(x) = c \cdot \alpha_k(x), \end{aligned}$$

where π_k is the prior probability of class k. From this and since the posterior probabilities need to sum up to 1, it holds that

$$\begin{aligned} 1 &= c \cdot \alpha_{\text{yes}}(x) + c \cdot \alpha_{\text{no}}(x) \\ \iff c &= \frac{1}{\alpha_{\text{yes}}(x) + \alpha_{\text{no}}(x)}. \end{aligned}$$

This means in order to compute $\pi_{\text{yes}}(x)$ the scores $\alpha_{\text{yes}}(x)$ and $\alpha_{\text{no}}(x)$ are needed.
Now we want to compute for a new fruit the posterior probability $\hat{\pi}_{yes}((\text{yellow}, \text{round}, \text{imported}))$.
Note that we do not know the *true* prior probability and the *true* conditional densities. Here -since the target and the features are categorical- we can estimate them with the relative frequencies encountered in the data, s.t.

$$\begin{aligned} \hat{\alpha}_{\text{yes}}(x) &= \hat{\pi}_{yes} \cdot \hat{p}(\text{yellow}|y = \text{yes}) \cdot \hat{p}(\text{round}|y = \text{yes}) \cdot \hat{p}(\text{imported}|y = \text{yes}) \\ &= \hat{\mathbb{P}}(y = \text{yes}) \cdot \hat{\mathbb{P}}(x_{\text{Color}} = \text{yellow}|y = \text{yes}) \cdot \hat{\mathbb{P}}(x_{\text{Form}} = \text{round}|y = \text{yes}) \cdot \hat{\mathbb{P}}(x_{\text{Origin}} = \text{imported}|y = \text{yes}) \\ &= \frac{3}{8} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot 1 = \frac{1}{24} \approx 0.042, \\ \hat{\alpha}_{\text{no}}(x) &= \hat{\pi}_{no} \cdot \hat{p}(\text{yellow}|y = \text{no}) \cdot \hat{p}(\text{round}|y = \text{no}) \cdot \hat{p}(\text{imported}|y = \text{no}) \\ &= \hat{\mathbb{P}}(y = \text{no}) \cdot \hat{\mathbb{P}}(x_{\text{Color}} = \text{yellow}|y = \text{no}) \cdot \hat{\mathbb{P}}(x_{\text{Form}} = \text{round}|y = \text{no}) \cdot \hat{\mathbb{P}}(x_{\text{Origin}} = \text{imported}|y = \text{no}) \\ &= \frac{5}{8} \cdot \frac{2}{5} \cdot \frac{3}{5} \cdot \frac{2}{5} = \frac{3}{50} = 0.06. \end{aligned}$$

At this stage we can already see that the predicted label is ”no”, since $\hat{\alpha}_{\text{no}}(x) = 0.06 > \frac{1}{24} = \hat{\alpha}_{\text{yes}}(x)$. With this we can calculate the posterior probability

$$\hat{\pi}_{\text{yes}}(x) = \frac{\hat{\alpha}_{\text{yes}}(x)}{\hat{\alpha}_{\text{yes}}(x) + \hat{\alpha}_{\text{no}}(x)} \approx 0.41.$$

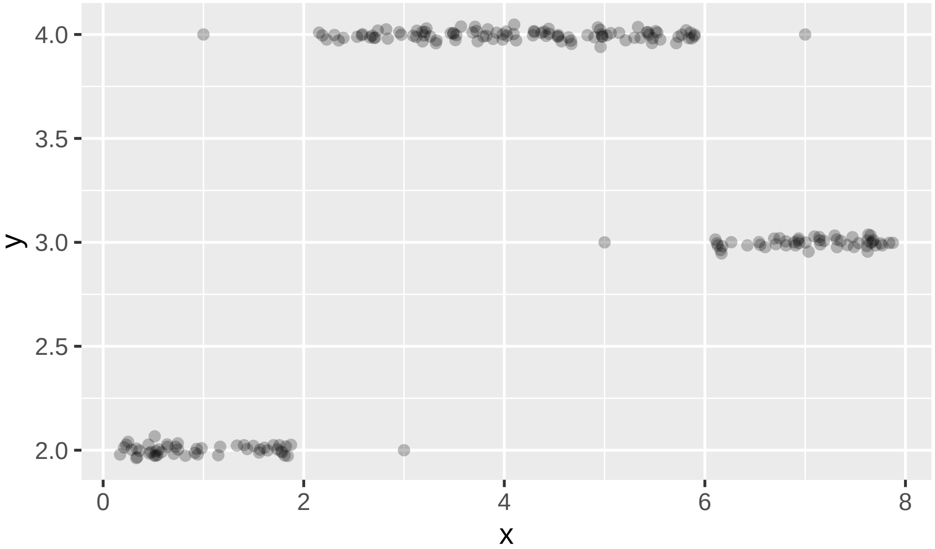
Corresponding R-Code:

```r
df_banana <- data.frame(
  Color = as.factor(
    c("yellow", "yellow", "yellow", "brown", "brown", "green", "green", "red")),
  Form = as.factor(
    c("oblong", "round", "oblong", "oblong", "round", "round", "oblong", "round")),
  Origin = as.factor(
    c("imported", "domestic", "imported", "imported", "domestic", "imported",
    "domestic", "imported")),
  Banana = as.factor(c("yes", "no", "no", "yes", "no", "yes", "no", "no"))
)

new_fruit <- data.frame(Color = "yellow", Form = "round", Origin = "imported", Banana = NA)
df_banana <- rbind(df_banana, new_fruit)

library(mlr3)
library(mlr3learners)

nb_learner <- lrn("classif.naive_bayes",
                  predict_type = "prob")

banana_task <- TaskClassif$new(
  id = "banana",
  backend = df_banana,
  target = "Banana"
)

nb_learner$train(banana_task, row_ids=1:8)

nb_learner$predict(banana_task, row_ids = 9)

## <PredictionClassif> for 1 observations:
##  row_ids truth response   prob.no  prob.yes
##        9  <NA>       no 0.5901639 0.4098361
```

b) For the distribution of a numerical feature given the the category we need to specify a probability distribution with continuous support. For example, for the information x_{Length} we could assume that $p(x_{\text{Length}}|y = \text{yes}) \sim \mathcal{N}(\mu_{\text{yes}}, \sigma^2_{\text{yes}})$ and $p(x_{\text{Length}}|y = \text{no}) \sim \mathcal{N}(\mu_{\text{no}}, \sigma^2_{\text{no}})$. (To estimate these normal distributions one would need to estimate their parameters $\mu_{\text{yes}}, \mu_{\text{no}}, \sigma^2_{\text{yes}}, \sigma^2_{\text{no}}$ on the data respectively)

Questions from past exams

Exercise 6: WS2020/21, main exam, question 1



The above plot shows $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$, a data set with $n = 200$ observations of a continuous target variable y and a continuous, 1-dimensional feature variable $\mathbf{x}$. In the following, we aim at predicting y with a machine learning model that takes $\mathbf{x}$ as input.

(a) Since the data seem to fall in 3 quite well-separable classes, we now want to apply a classification model instead of the regression model in a). To prepare the data for classification, we categorize the target variable y in 3 classes and call the transformed target variable z, as follows:

$$z^{(i)} = \left\{ \begin{array}{ll} 1, \text{ if} & -\infty < y^{(i)} \leq 2.5 \\ 2, \text{ if} & 2.5 \ < y^{(i)} \leq 3.5 \\ 3, \text{ if} & 3.5 < y^{(i)} < \infty \end{array} \right\}$$

Now we can apply quadratic discriminant analysis (QDA):

(i) Estimate the class means $\mu_k = \mathbb{E}(\mathbf{x}|z = k)$ for each of the three classes $k \in \{1, 2, 3\}$ visually from the plot. Do not overcomplicate this, a rough estimate is sufficient here.

(ii) Make a hand-drawn plot that visualizes the different estimated densities per class.

(iii) How would your drawing from (ii) change if we used linear discriminant analysis (LDA) instead of QDA? Explain your answer.

(iv) Why is QDA for this data preferable over LDA?

(b) Given are two new observations $\mathbf{x}_{*1} = -10$ and $\mathbf{x}_{*2} = 7$. State the prediction for each of the two models

(i) regression tree (from a))

(ii) QDA (from b))

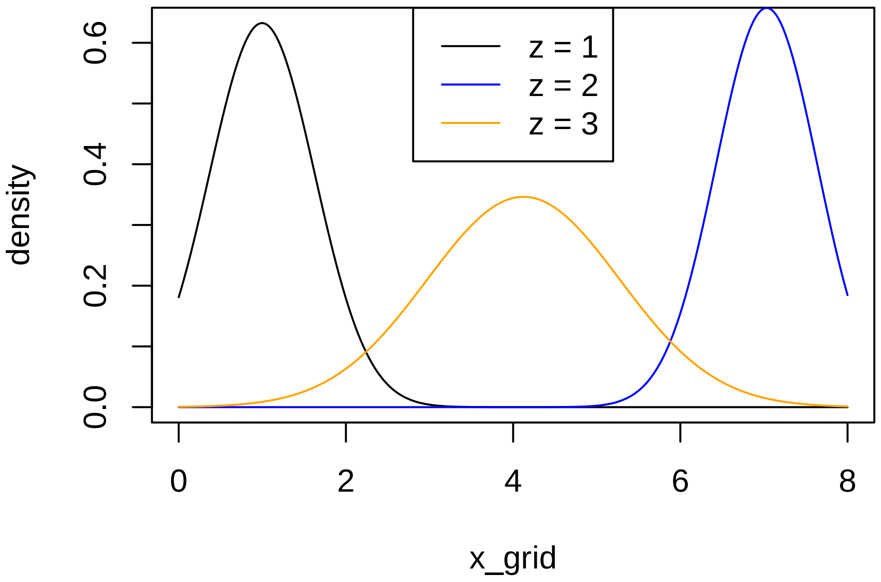
and explain how you derived the predictions.

(c) Discuss in 1-2 sentences which of the 2 models (regression tree, QDA) you would prefer for modeling the data and explain your decision.

Solution 6:

(a)

(i) $\mu_1 = 1$, $\mu_2 = 7$, $\mu_3 = 4$



(ii)

(iii) Variances would be all equal. Assumption of LDA is equal variances, i.e., estimatated models will always have equal variances, no matter if this fits the data or not

(iv) Variances seem not to be equal, this is only captured in QDA

(b) Given are two new observations $\mathbf{x}_{*1} = -10$ and $\mathbf{x}_{*2} = 7$. State the prediction for each of the two models

(i) $\hat{y}_{*1} = 2$, $\hat{y}_{*2} = 3$,

(ii) $\hat{z}_{*1} = 3$, since the variance of class 3 is higher, the density will overshoot the density of class 1. $\hat{z}_{*2} = 2$, obviously highest posterior here.

and explain how you derived the predictions.

(c) E.g.,

- CART better than LM because I do not have to specify those indicator functions manually and estimate the split points manually, CART does this data driven
- For QDA we have to throw away information of y, this favors CART
- QDA predicts the middle class (3) for very extreme observations, this does not seem right. However, we do not know how data behave outside the bounds of x.
- QDA assumes gaussian distributions which is clearly not the case.

Exercise 7: WS2020/21, main exam, question 3

Consider a binary classification algorithm that yielded the following results on 8 observations. The table shows true classes and predicted probabilities for class 1:

ID	True Class	Prediction
1	1	0.50
2	0	0.01
3	0	0.90
4	0	0.55
5	1	0.10
6	1	0.72
7	1	0.70
8	1	0.99

(a) Draw the ROC curve of the classifier manually. Explain every step thoroughly, and make sure to annotate the axes of your plot.

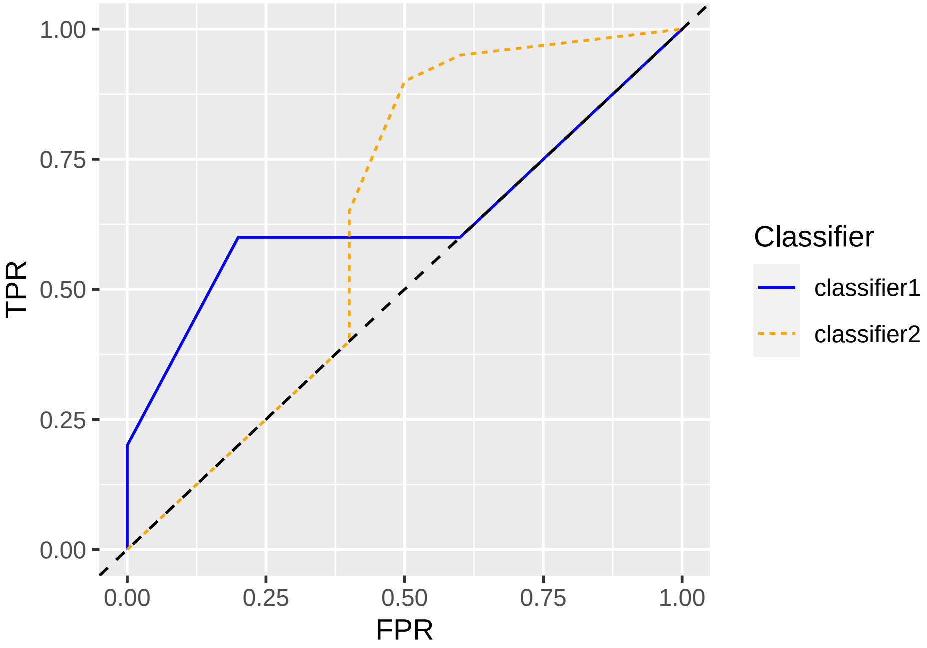
(b) Calculate the AUC of the classifier. Explain every step of your computation thoroughly.

(c) Calculate the partial AUC for FPR $\leq 1/3$. Explain every step of your computation thoroughly.

(d) Create a confusion matrix assuming a threshold of 0.75. Point out which values correspond to true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

(e) Compute sensitivity, specificity, negative predictive value, positive predictive value, accuracy and F1-score. State the respective formulas first.

(f) In the following plot, you see the ROC curves of two different classifiers with similar AUC's. Describe a practical situation where you would prefer classifier 1 over classifier 2 and explain why. (Note: The data in this question is not related to the data of the above questions.)



Solution 7:

(a) Scores:

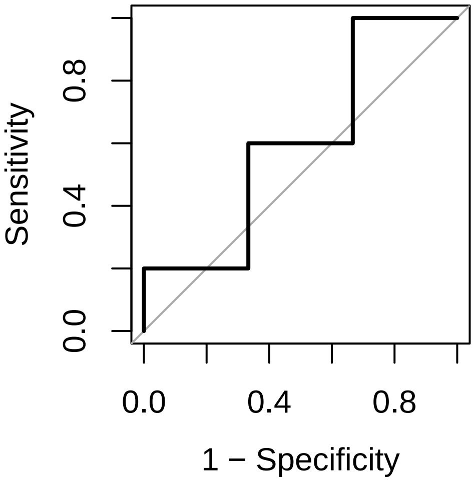
	true_labels	scores
8	1	0.99
3	0	0.90
6	1	0.72
7	1	0.70
4	0	0.55
1	1	0.50
5	1	0.10
2	0	0.01

Here we see that $\frac{1}{n_+} = \frac{1}{5} = 0.2$ and $\frac{1}{n_-} = \frac{1}{3}$. Now we follow the algorithm as described in the lecture slides:

- Set $\tau = 1$, so we start in $(0, 0)$; we predict everything as 1.
- Set $\tau = 0.95$ yields TPR $0 + \frac{1}{n_+} = 0.2$ and FPR 0. (Obs. 8 is '1')
- Set $\tau = 0.8$ yields TPR 0.2 and FPR $0 + \frac{1}{n_-} = 1/3$. (Obs. 3 is '0')
- Set $\tau = 0.71$ yields TPR $0.2 + \frac{1}{n_+} = 0.4$ and FPR $1/3$. (Obs. 6 is '1')
- Set $\tau = 0.6$ yields TPR $0.4 + \frac{1}{n_+} = 0.6$ and FPR $1/3$. (Obs. 7 is '1')
- Set $\tau = 0.52$ yields TPR 0.6 and FPR $1/3 + \frac{1}{n_-} = 2/3$. (Obs. 4 is '0')
- Set $\tau = 0.3$ yields TPR $0.6 + \frac{1}{n_+} = 0.8$ and FPR $2/3$. (Obs. 1 is '1')
- Set $\tau = 0.05$ yields TPR $0.8 + \frac{1}{n_+} = 1$ and FPR $2/3$. (Obs. 5 is '1')
- Set $\tau = 0$ yields TPR 1 and FPR $2/3 + \frac{1}{n_-} = 1$. (Obs. 2 is '0')

Therefore we get the polygonal path consisting of the ordered list of vertices

$$(0, 0), (0, 0.2), (1/3, 0.2), (1/3, 0.4), (1/3, 0.6), (2/3, 0.6), (2/3, 0.8), (2/3, 1), (1, 1).$$



(b) The AUC is the sum of three rectangles: $0.2 \cdot 1/3 + 0.6 \cdot 1/3 + 1 \cdot 1/3 = 0.6$

(c) The partial AUC is the area under the curve that is left from FPR $= 1/3$, so it is just the first rectangle: $0.2 \cdot 1/3 = 1/15$

(d)

	Actual Class - 0	Actual Class - 1
Prediction - 0	2	4
Prediction - 1	1	1

so we get

FN	FP	TN	TP
4	1	2	1

(e)

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{1}{5}$$
$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{2}{3}$$
$$\text{Negative Predictive Value} = \frac{\text{TN}}{\text{TN} + \text{FN}} = \frac{1}{3}$$
$$\text{Positive Predictive Value} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{1}{2}$$
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{3}{8}$$
$$\text{F1-score} = \frac{2 \cdot \text{PPV} \cdot \text{Sensitivity}}{\text{PPV} + \text{Sensitivity}} = (0.2)/(0.7) = 2/7$$

(f) Important point is to understand that classifier 1 does better in the 'high scores'. Or: For some threshold below 0.5 the precision is far better for classifier 1 than for classifier 2. For example if you want to output a certain number of 'most promising customers' this could be a good idea.

Exercise 8: WS2020/21, main exam, question 4

The table below shows $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$, a data set with $n = 8$ observations of a binary target variable y containing the information if the object is a `Banana` or not and a 4-dimensional feature vector $\mathbf{x}$, containing `Color`, `Form`, `Origin` (categorical features) and `Length` (continuous feature) of the object. In the following, we aim at predicting `Banana` with a machine learning model that takes $\mathbf{x}$ as input.

ID	Color	Form	Origin	Length [cm]	Banana
1	yellow	oblong	imported	15	yes
2	yellow	round	domestic	5	no
3	yellow	oblong	imported	10	no
4	brown	oblong	imported	17	yes
5	brown	round	domestic	16	no
6	green	round	imported	13	yes
7	green	oblong	domestic	25	no
8	red	round	imported	7	no

We want to use a naive Bayes classifier to predict the label of a new observation.

(a) Calculate the posterior probability $\pi(\mathbf{x}_*) = \mathbb{P}(y = \text{yes}|\mathbf{x}_*)$ for a new observation $\mathbf{x}_* = (\text{green}, \text{oblong}, \text{imported}, 14)^\top$. Explain every step thoroughly. (Hint: At some point you will have to compute values of Gaussian densities - use R for this step.)

(b) How would you classify the new observation? Explain your answer.

Solution 8:

(a) The features $\mathbf{x} := (x_{\text{Color}}, x_{\text{Form}}, x_{\text{Origin}}, x_{\text{Length}})$ given the category $y \in \{\text{yes}, \text{no}\}$ are assumed to be conditionally independent of each other (since we are using Naive Bayes), s.t.

$$p(\mathbf{x}|y = k) = p(x_{\text{Color}}|y = k) \cdot p(x_{\text{Form}}|y = k) \cdot p(x_{\text{Origin}}|y = k) \cdot p(x_{\text{Length}}|y = k).$$

For the posterior probabilities $\pi_k(\mathbf{x}) = \mathbb{P}(y = k|\mathbf{x})$ it holds with Bayes' Theorem:

$$\begin{aligned}\pi_k(\mathbf{x}) &\propto \underbrace{\pi_k \cdot p(\mathbf{x}|y = k)}_{=:\alpha_k(\mathbf{x})}\\ &\iff \exists c \in \mathbb{R} : \pi_k(\mathbf{x}) = c \cdot \alpha_k(\mathbf{x}),\end{aligned}$$

where $\pi_k = \mathbb{P}(y = k)$ is the prior probability of class k.
From this and since the posterior probabilities need to sum up to 1, it holds that

$$\begin{aligned}1 &= c \cdot \alpha_{\text{yes}}(\mathbf{x}) + c \cdot \alpha_{\text{no}}(\mathbf{x})\\ \iff c &= \frac{1}{\alpha_{\text{yes}}(\mathbf{x}) + \alpha_{\text{no}}(\mathbf{x})}.\end{aligned}$$

This means, the scores $\alpha_{\text{yes}}(x)$ and $\alpha_{\text{no}}(x)$ are needed to compute

$$\pi_{\text{yes}}(\mathbf{x}) = \frac{\alpha_{\text{yes}}(\mathbf{x})}{\alpha_{\text{yes}}(\mathbf{x}) + \alpha_{\text{no}}(\mathbf{x})}.$$

For the new observation $\mathbf{x}_*$ we have to estimate the respective probabilities and densities. For the categorical features, we can simply compute relative frequencies. For the continuous features `Length` we have to estimate the densities per class and evaluate those at the value 14. Doing this with R we end up with:

```
## [1] 0.1760327
## [1] 0.04863352
```

k	$\hat{\pi}_k$	$\hat{p}(\text{green}\|y = k)$	$\hat{p}(\text{oblong}\|y = k)$	$\hat{p}(\text{imported}\|y = k)$	$\hat{p}(14\|y = k)$
yes	3/8	1/3	2/3	3/3	0.176
no	5/8	1/5	2/5	2/5	0.049

$$\begin{aligned}\hat{\alpha}_{\text{yes}}(x) &= \hat{\pi}_{yes} \cdot \hat{p}(\text{green}|y = \text{yes}) \cdot \hat{p}(\text{oblong}|y = \text{yes}) \cdot \hat{p}(\text{imported}|y = \text{yes}) \cdot \hat{p}(14|y = \text{yes})\\ &= \frac{3}{8} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot 1 \cdot 0.176 \approx 0.0147,\\ \hat{\alpha}_{\text{no}}(x) &= \hat{\pi}_{no} \cdot \hat{p}(\text{green}|y = \text{no}) \cdot \hat{p}(\text{oblong}|y = \text{no}) \cdot \hat{p}(\text{imported}|y = \text{no}) \cdot \hat{p}(14|y = \text{no})\\ &= \frac{5}{8} \cdot \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{2}{5} \cdot 0.049 \approx 0.00098.\end{aligned}$$

With this we can calculate the posterior probability

$$\hat{\pi}_{\text{yes}}(x) = \frac{\hat{\alpha}_{\text{yes}}(x)}{\hat{\alpha}_{\text{yes}}(x) + \hat{\alpha}_{\text{no}}(x)} \approx 0.937.$$

(b) Classification as yes = `banana`. We have to define a threshold, observations with a posterior probability equal or above the threshold are hard labeled as yes, others as no. The optimal threshold has to be chosen, e.g., inspecting ROC measures. With default 0.5 we end up with the above classification.

Exercise 9: WS2020/21, retry exam, question 3

Consider a binary classification algorithm that yielded the following results on 8 observations. The table shows true classes and predicted probabilities for class 1:

ID	True Class	Prediction
1	1	0.30
2	0	0.91
3	0	0.03
4	0	0.55
5	0	0.45
6	0	0.65
7	1	0.71
8	1	0.98

(a) Draw the ROC curve of the classifier manually. Compute all relevant numbers explicitly, state the respective formulas first, and make sure to annotate the axes of your plot.

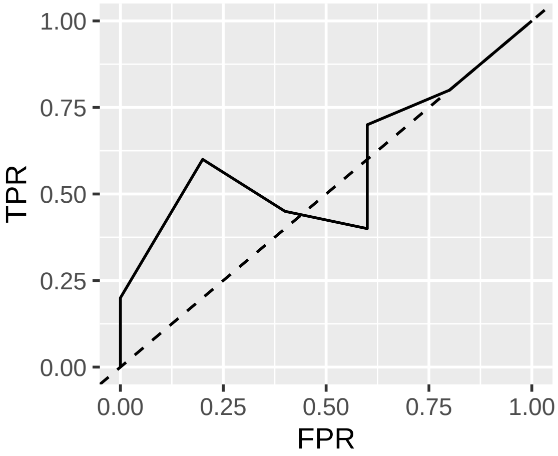
(b) Calculate the AUC of the classifier. Explain every step of your computation thoroughly.

(c) Calculate the partial AUC for TPR $\geq 2/3$. Explain every step of your computation thoroughly.

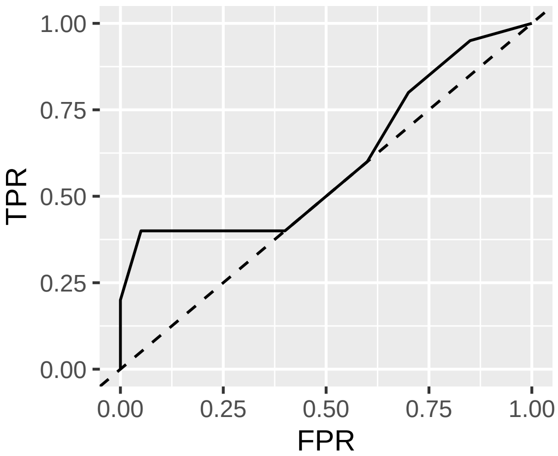
(d) Create a confusion matrix assuming a threshold of 0.7. Point out which values correspond to true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

(e) Compute sensitivity, specificity, negative predictive value, positive predictive value, accuracy and F1-score. State the respective formulas first.

(f) Now look at the following plot. Explain thoroughly why the solid line cannot be the ROC curve of a classifier.



(g) Imagine you are working in the marketing department of a large company. Your colleagues are developing a marketing campaign where they will call selected customers by phone in order to advertise a new product. Your company has 1,000,000 customers in the database and the budget of the marketing campaign allows to call 1,000 of these customers. Now it is your job to select the most promising 1,000 customers, i.e., those who will most likely buy the new product after having received the advertising phone call. Luckily, you have a large amount of similar data from older marketing campaigns that are representative for this campaign and can be used to train a supervised classification model with the target variable indicating if the customer did buy the product (1) or not (0). You train a random forest on the 1,000,000 observations and get the following cross-validated ROC curve. Assuming a balanced target variable: Do you think this model is fairly good for your purpose? Explain why and describe, how you would proceed from here to provide your colleagues with the 1,000 most promising customers.



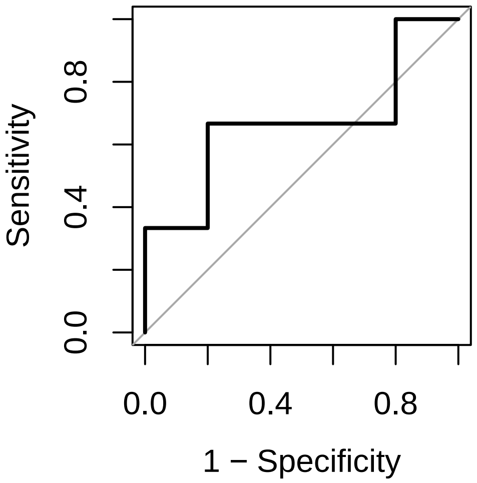
Solution 9:

(a) First we sort the results by score:

	true_labels	scores
8	1	0.98
2	0	0.91
7	1	0.71
6	0	0.65
4	0	0.55
5	0	0.45
1	1	0.30
3	0	0.03

Here we see that $\frac{1}{n_-} = \frac{1}{5} = 0.2$ and $\frac{1}{n_+} = \frac{1}{3}$. Now we follow the algorithm as described in the lecture slides:

- Set $\tau = 1$, so we start in $(0, 0)$; we predict everything as 1.
- Set $\tau = 0.95$ yields TPR $0 + \frac{1}{n_+} = 1/3$ and FPR 0. (Obs. 8 is '1')
- Set $\tau = 0.9$ yields TPR 1/3 and FPR $0 + \frac{1}{n_-} = 0.2$. (Obs. 2 is '0')
- Set $\tau = 0.70$ yields TPR $1/3 + \frac{1}{n_+} = 2/3$ and FPR 0.2. (Obs. 7 is '1')
- Set $\tau = 0.6$ yields TPR 2/3 and FPR $0.2 + \frac{1}{n_-} = 0.4$. (Obs. 6 is '0')
- Set $\tau = 0.5$ yi´elds TPR 2/3 and FPR $0.4 + \frac{1}{n_-} = 0.6$. (Obs. 4 is '0')
- Set $\tau = 0.4$ yields TPR 2/3 and FPR $0.6 + \frac{1}{n_-} = 0.8$. (Obs. 5 is '0')
- Set $\tau = 0.1$ yields TPR $2/3 + \frac{1}{n_+} = 1$ and FPR 0.8. (Obs. 1 is '1')
- Set $\tau = 0$ yields TPR 1 and FPR $0.8 + \frac{1}{n_-} = 1$. (Obs. 3 is '0')



(b) The AUC is the sum of three rectangles: $0.2 \cdot 1/3 + 0.6 \cdot 2/3 + 1 \cdot 0.2 = 2/3$

(c) The partial AUC is the area under the curve that is above TPR $= 2/3$, so it is just the small rectangle in the upper right corner: $0.2 \cdot 1/3 = 1/15$

(d)

	Actual Class - 0	Actual Class - 1
Prediction - 0	4	1
Prediction - 1	1	2

so we get

FN	FP	TN	TP
1	1	4	2

(e)

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{2}{3}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{4}{5}$$

$$\text{Negative Predictive Value} = \frac{\text{TN}}{\text{TN} + \text{FN}} = \frac{4}{5}$$

$$\text{Positive Predictive Value} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{2}{3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{6}{8}$$

$$\text{F1-score} = \frac{2 \cdot \text{PPV} \cdot \text{Sensitivity}}{\text{PPV} + \text{Sensitivity}} = (8/9)/(4/3) = 2/3$$

(f) TPR and FPR are both metrics that increase monotonically as the threshold c traverses from 1 to 0. The plot shows two instances of diminishing TPR, which would mean that, at the corresponding threshold, an observation that had previously been correctly classified as positive was not detected anymore. This is not possible with a binarization threshold.

(g) Yes. I would order the customers wrt the scores, then roughly the first 200,000 customers would be true positives (since the ROC is based on 1,000,000 customers and true class is balanced) and I would just select the top 1,000 customers. It doesn't matter that the classifiers gets bad later.

Ideas & exercises from other sources