Exercise Set 2

- Submit the answer via Moodle at the latest on Wednesday, 29 November 2023, at 23:59.
- You can answer anonymously or write your name on the answer sheet; it is your choice.
- One person should complete the assignment, but discussions and joint-solving sessions with others are encouraged. Your final solution must, however, be your own. You are not allowed to copy ready-made solutions or solutions made by other students. You are permitted to use external sources, web searches included.
- You can discuss the problems in the exercise workshop.
- Your answer will be peer-reviewed by you and randomly selected other students.
- The language of the assignments is English.
- The submitted report should be in a single Portable Document Format (pdf) file.
- Answer the problems in the correct order.
- Read Moodle's general instructions and grading criteria before starting the problems.
- The primary source material is ISLR_v2, Sections 4-5 and 8-9. However, please feel free to use other resources as well. "ISLR_v2" refers to James, Witten, Hastie, and Tibshirani, 2021. An Introduction to Statistical Learning with Applications in R, 2nd edition. Springer.
- Notice that you can submit your answer to Moodle well before the deadline and revise it until the deadline. Therefore, please submit your solution in advance after you have solved some problems! Due to the peer review process, we cannot grant extensions to the deadline. Even though the Moodle submission may occasionally remain open after 23:59, the submission system will eventually close. If you try to submit your answers late, you will **not** get any points (including peer-review points) from this Exercise Set. You have been warned.
- Please double-check that the submitted PDF is appropriately formatted and, e.g., contains all figures. Producing correctly formatted PDF files with Jupyter Notebooks can be challenging: remember to check the created PDF. We recommend using R Markdown instead of Jupyter notebooks.
- You can solve the problems at your own pace. However, we have provided a suggested schedule below. If you follow the plan, you can do the problems after we cover the topics in class.

Updates

- Updated information about `penalty=None` option in `sklearn.linear_model.LogisticRegression` for P8a.

Suggested schedule

Please see the Moodle for the reading list and topics of the classes. Before the classes, read the respective textbook sections and only attempt to do the problems afterwards! Also, notice that the lab sections at the end of the textbook chapters contain helpful hints for many problems requiring programming.

As a general guideline, if you get stuck on a problem or a task, please do the other problems or tasks first and then return to the complex problem. You can ask for help in Slack (channel #e2) or at the exercise workshops.

We recommend that you do the problems as follows:

After L5 subset selection and shrinkage methods, linear discriminative classifiers (15 Nov) and before the following lecture

Do problem 8.

After L6 generative classification methods (17 Nov)and before the following lecture

Do problems 9-12.

After L7 kNN and decision/regression trees (22 Nov) and before the following lecture

Do problems 13 and 14. You should finish problems 9-12 if you needed more time to complete them before L7.

After L8 advanced topics in machine learning - SVM and ensemble methods (24 Nov) and by 27 November

Do problem 15, write the learning diary (Problem 16) on 27 November, and submit your final solution.

You can still fine-tune your solution until the deadline.

Default policy in the case of force majeure reasons

If you need a special arrangement due to a sudden illness or other force majeure reason, contact us as soon as possible. As a default policy, we assume that in such cases, you have followed the schedule recommended above up until you inform us of the force majeure reason. For example, if you contact us on the last day before the deadline, we assume you have had time to solve the problems. Therefore, please solve the problems in time and submit your answer early (you can update the answer until the deadline)!

Problem 8

[5 points]

Topic: logistic regression, discriminative vs generative classifiers

In this problem, you will apply logistic regression (with an intercept term) to the *spam dataset*.

Section 4.7 (lab section) of ISLR_v2 (or ISLP) contains helpful information for solving this problem.

SPAM dataset

- We have constructed the spam dataset by applying the SpamAssassin spam filter on a subset of email messages from the Enron-Spam dataset.
- SpamAssassin analyzes the data using binary tests (for example, "Does the email have unusually many whitespace characters?") and then classifies the email as spam or ham (=not spam) based on the test outcomes.
- We will use five of these binary tests as covariates: `MISSING_FROM`, `FROM_ADDR_WS`, `TVD_SPACE_RATIO`, `LOTS_OF_MONEY`, and `T_FILL_THIS_FORM_SHORT` (for explanations of the tests, see the SpamAssassin documentation).
- The covariate vector $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) \in \{0, 1\}^5$ is therefore a 5-dimensional binary vector.
- The class variable $y \in \{0, 1\}$ (column SPAM in the CSV file) equals $y = 1$ if the message is spam and $y = 0$ if it is ham.
- You can find a training dataset of 100 emails from `spam_train.csv` and test data of 1000 emails from `spam_test.csv`.

Performance measures

In this exercise set, you will train probabilistic classifiers which estimate $\hat{p}(y \mid \mathbf{x})$: the probability of class y given the covariate vector $\mathbf{x}$. Two commonly used performance measures for probabilistic classifiers are *accuracy* and *perplexity*. We use the following notation:

- $y_i \in \{0, 1\}$: the true class of point i.
- $\hat{p}_i = \hat{p}(y = 1 \mid \mathbf{x}_i)$: the estimated probability for the ith point in a dataset of size n being spam.
- $\hat{y}_i$: the predicted class for point i which is $\hat{y}_i = 1$ if $\hat{p}_i \geq 0.5$ and $\hat{y}_i = 0$ otherwise.

We define the *accuracy* on a dataset of n items as follows:

$$\text{accuracy} = \sum\nolimits_{i=1}^{n} \mathrm{I}(y_i = \hat{y}_i)/n,$$

where $\mathrm{I}(z)$ is the indicator function which equals one if z is true and zero otherwise and the *perplexity* as:

$$\text{perplexity} = \exp\left(-\sum\nolimits_{i=1}^{n} \log \hat{p}(y = y_i \mid \mathbf{x}_i)/n\right).$$

Perplexity is a transformation of the likelihood (perplexity $= \exp(-\text{loglikelihood}/n)$), which may be the most commonly used performance measure on probabilistic classifiers. Example values are perplexity $= 1$ for a perfect classifier, which always predicts the probability of one to an actual class, and perplexity $= 2$ for coin flipping, which has a predicted class probability $\hat{p} = 1/2$.

Task a

Using one-hot encoding for y_i, train a logistic regression model *without* Lasso or Ridge regularisation on the training data. Then:

(i) Report the model coefficients.
(ii) Compute and report the accuracy and perplexity on the training and testing data. Make sure that you use no regularisation. Notice that you may get warnings about convergence; why?
(iii) Write down the equation for the predicted class probability, given the model coefficients β and the covariate vector $\mathbf{x}$.

Hint: In this task, with `sklearn.linear_model.LogisticRegression` you should turn of regularisation (on by default) with `penalty=None` option. You can also use R or `statsmodels`; see ISLR_v2 or ISLP, Section 4.7.2, for guidance.

Task b

Train a logistic regression model with Lasso regularisation. Find a regularisation coefficient value that performs better than the unregularised version on the test data and has some regression coefficients equal to zero. You can do this by trying various values; there is no need to be more sophisticated here.

Report your parameters, the regression coefficients, and the accuracies and perplexities on the testing data.

Look at the predicted class probabilities for the unregularized regressor in Task a, and your regularized regressors in Task b. How do the distributions of the probabilities differ?

Hint: For this task, you can use `statsmodels GLM.fit_regularised` or the R `glmnet` and `glmnetUtils` libraries (the latter of which provides a formula interface).

Problem 9

[6 points]

Objective: generative Bayes classifier

In this problem, you will study the quadratic discriminant analysis (QDA) classifier. Consider a simple case with two classes and only one feature ($K = 2$ and $p = 1$).

Task a

Prove that the QDA classifier is *not* linear if the class-specific variances differ ($\sigma_1^2 \neq \sigma_2^2$).

Hint: This problem is from the textbook (Problem 3, page 189). Please see the discussion in the textbook for hints and guidance. For this problem, you should follow the arguments laid out in Sect. 4.4.1 of the textbook, but without assuming that $\sigma_1^2 = \sigma_2^2$.

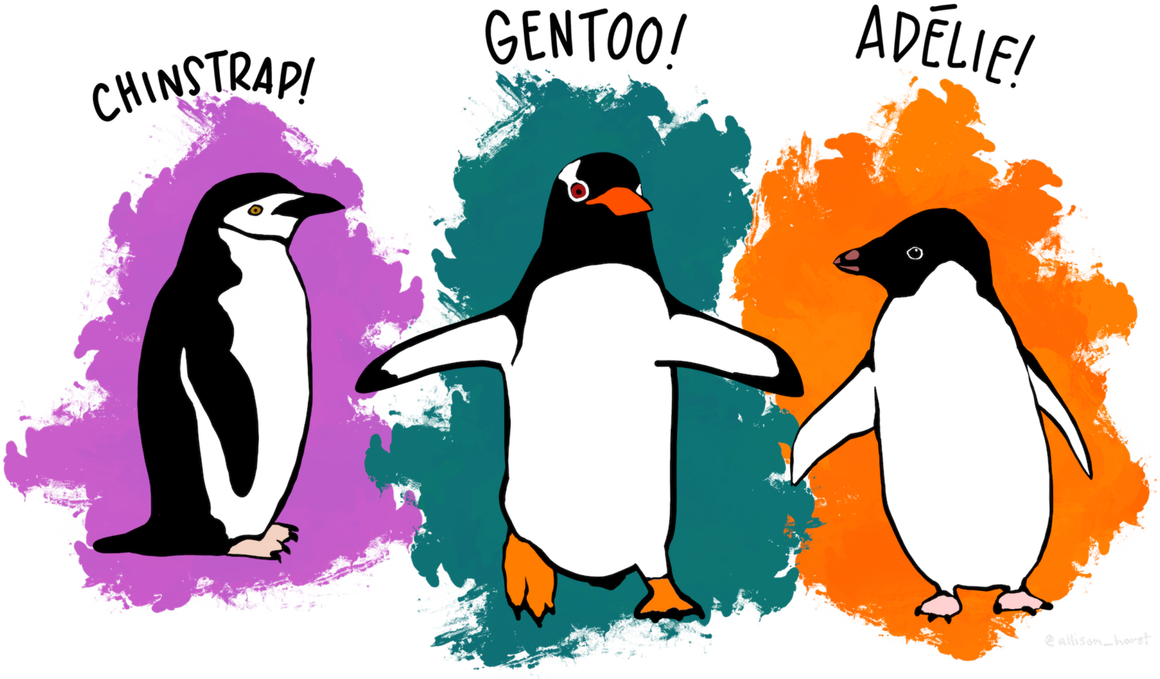
Problem 10

[6 points]

Objective: naive Bayes classifier

In this problem, you will study the Palmer penguins by building your own Naive Bayes classifier.

Palmer penguins dataset



Artwork by @allison_horst

- Dataset description.
- Use `penguins_train.csv` as your training data and `penguins_test.csv` as your testing data.
- Binary classification task: classify the penguin species as $y \in \{\text{Adelie}, \text{notAdelie}\}$ (`notAdelie` combining Gentoo and Chinstrap species) based on four morphological and weight measurements of the individual penguins, denoted by $\mathbf{x} = (x_1, x_2, x_3, x_4)^T \in \mathbb{R}^4$

Naive Bayes (NB) classifier

Your task is to build your own NB classifier; you should not use a ready-made classifier from a library. However, you do not need to create a generic classifier (such as `naiveBayes` in the R `e1071` library); it is enough that your classifier works for this particular task.

The idea of NB is that the dimensions are conditionally independent, given the class. Each class conditional feature distribution $p(x_i \mid y)$ is assumed to originate from an independent Gaussian distribution with its mean μ_{iy} and variance σ^2_{iy} for $i = 1, 2, 3, 4$.

Task a

Compute and report each attribute's means and standard deviations separately in the training set for both classes.

Estimate and report the class probabilities using Laplace smoothing with a *pseudocount* of 1 on the training set.

(You should produce a total of 18 numbers from this task.)

Task b

Now, you can find the class-specific expressions for $p(\mathbf{x} \mid y)$ needed by the NB classifier. Remember that according to NB assumption, the dimensions are independent, and hence, you can represent the class-specific $p(\mathbf{x} \mid y)$ likelihoods as products of 4 1-dimensional normal distributions.

Write down the formula needed to compute the posterior probability of the class being `Adelie` $\widehat{p}(y = \text{Adelie} \mid \mathbf{x})$ as a function of the four measurements in $\mathbf{x}$ and the statistics (means, standard deviations, class probabilities) you computed in the task a above.

Task c

Using the formula you derived in Task b, compute and report your classifier's classification accuracy on the test set. Additionally, calculate and report the probabilities $\widehat{p}(y = \text{Adelie} \mid \mathbf{x})$ for the three first penguins in the test set.

Hint: When computing classification accuracy, you can use the following rule to obtain "hard" classes:

$$\hat{y} = \begin{cases} \text{Adelie} & , \quad \widehat{p}(y = \text{Adelie} \mid \mathbf{x}) \geq 0.5 \\ \text{notAdelie} & , \quad \widehat{p}(y = \text{Adelie} \mid \mathbf{x}) < 0.5 \end{cases}$$

Problem 11

[5 points]

Objective: Understanding discriminative vs generative learning.

Download the reference below. You **do not need to read the full paper** or understand all the details! Instead, try to find the answers to the following questions.

Reference: Ng, Jordan (2001) On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. NIPS. http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf

Task a

Read the *Abstract* and *Introduction* (Sect. 1). According to the authors, is discriminative learning better than generative learning? Justify your answer.

Task b

By a "parametric family of probabilistic models", the authors mean a set of distributions where a group of parameters defines each distribution. An example of such a family is the family of normal distributions where the parameters are μ and Σ.

Ng and Jordan denote by h_{Gen} and h_{Dis} two models chosen by optimizing different objectives. Which two families do the authors discuss, and what are the (h_{Gen}, h_{Dis}) pairs for those models? What objectives are being optimised?

Task c

Study Figure 1 in the paper. Explain what it suggests (see the last paragraph of the Introduction). Reflect on what this means for the families in Task b.

Problem 12

[5 points]

Objective: comparing classifiers on synthetic data, application of different classifiers

In this problem, you will compare different classifiers using synthetic toy data sets.

Section 4.7.2 (Logistic regression) and 4.7.5 (NB) of ISLR_v2 (or ISLP) contain helpful information for solving this problem.

Toy data sets

We have generated ten training data sets of different sizes `toy_train_<n>.csv` for $n \in \{2^3, 2^5, \dots, 2^{12}\}$, and one test data set `toy_test.csv` with 10000 points.

Each toy data set has a binary class variable $y \in \{0, 1\}$ and two real-valued features $x_1, x_2 \in \mathbb{R}$.

The data are generated from the "true" model as follows:

- x_1 and x_2 are sampled from a normal distribution with zero mean and unit variance.
- The probability of y is given by:

$$p(y = 1 \mid x_1, x_2) = \sigma(-1/2 - x_1 + 3x_2/2 + x_1x_2/3),$$

  where $\sigma(t) = 1/(1 + e^{-t})$ is the standard logistic function.

Task a

Is the Naive Bayes (NB) assumption valid for the toy data set? Explain why or why not.

Task b

For each training set, train several classifiers that output probabilities (described below), and then report their accuracy and perplexity on the test set.

Produce the following table (or make a plot) for accuracy and for perplexity on the test set:

n	NB	LR	LRi	OptimalBayes	Dummy
8	?	?	?	?	?
16	?	?	?	?	?
32	?	?	?	?	?
64	?	?	?	?	?
128	?	?	?	?	?
256	?	?	?	?	?
512	?	?	?	?	?
1024	?	?	?	?	?
2048	?	?	?	?	?
4096	?	?	?	?	?

where the columns correspond to:

- Naive Bayes (NB) (e.g., `naiveBayes` from the library `e1071`)
- Logistic regression without an interaction term (e.g., `glm`)
- Logistic regression with an interaction term (e.g., `glm`)
- Optimal Bayes classifier that uses the actual class conditional probabilities (that you know in this case!) to compute $p(y \mid x_1, x_2)$ for a given (x_1, x_2) - no probabilistic classifier can do better than this

- "Dummy classifier" that does not depend on $\mathbf{x}$. It always outputs the probability $\widehat{p}(y = 1 \mid x_1, x_2)$ as the fraction of $y = 1$ in the training data. "Dummy" means that the classifier output does not depend on the covariates. Including a dummy classifier in your comparison is always a good idea! One way to get a dummy classifier here is to train a logistic regression with only the intercept term.

Task c

Report the logistic regression coefficients with interaction terms for the largest training data set. How do they compare with the coefficients of the actual model that generated the data?

Discuss your observations and what you can conclude.

- Which of the models above are probabilistic, discriminative, and generative?
- How do accuracy and perplexity (log-likelihood) compare?
- Is there a relation to the insights from the previous problem?
- Why does logistic regression with the interaction term perform so well for larger datasets?
- Does your dummy classifier ever outperform other classifiers, or do different classifiers outperform the optimal Bayes classifier?

Instructions

It is helpful to make a function that takes the training and testing data as input and outputs the probabilities $p(y = 1 \mid x_1, x_2)$. By using these probabilities you can quickly compute accuracy and perplexity.

In R, you can include the interaction term in logistic regression by writing ("*" implies that interaction should be included in the model instead of "+", which assumes only additive effects):

```r
# R
model <- glm(y ~ x1 * x2, data[[4]], family = "binomial")
```

In Python, you can create interaction terms with, e.g., sklearn.preprocessing.PolynomialFeatures (set `interaction_only=True`).

```python
# Python
from sklearn.preprocessing import PolynomialFeatures

create_inter = PolynomialFeatures(degree = 2, interaction_only = True)
data3_inter = create_inter.fit_transform(data[3].iloc[:, :2])
GaussianNB().fit(data3_inter, data[3]["y"])
```

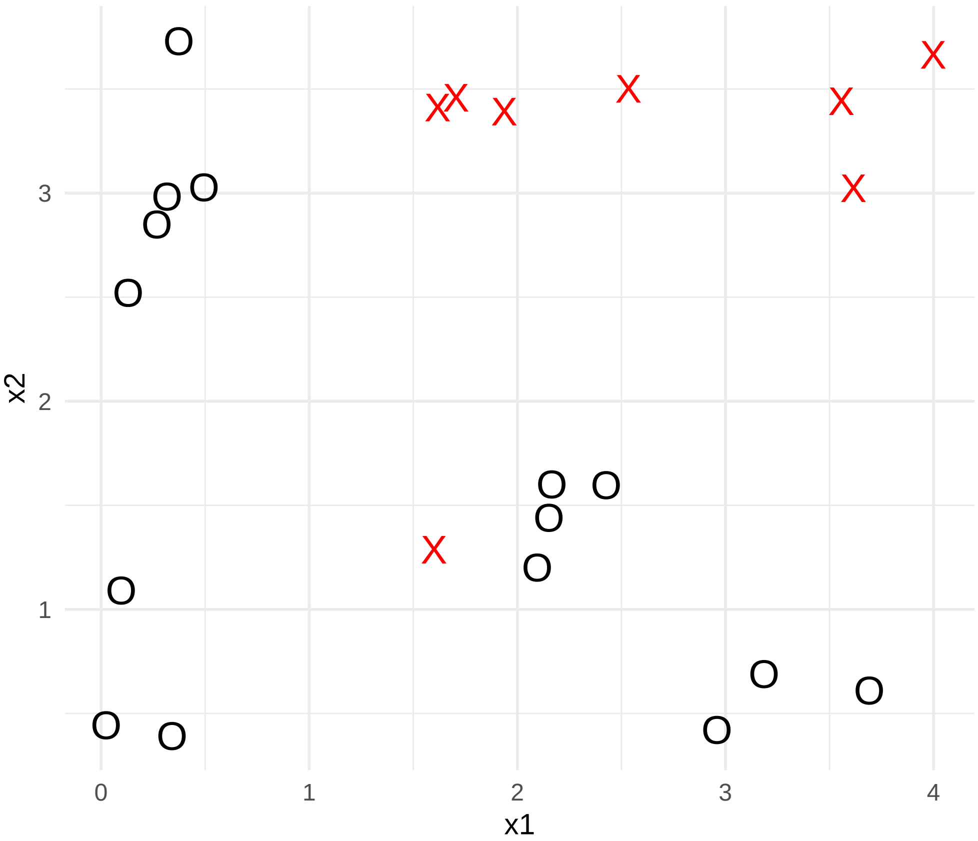
Problem 13

[6 points]

Objectives: basic principles of decision trees

In this task, you will simulate a decision tree algorithm by hand using the toy data shown in the figure.

Read Section 8.1 of ISLR_v2. Use the Gini index of Equation (8.6) as an impurity measure.



Task a

Sketch a run of the classification tree algorithm on the toy data and draw the resulting classification tree. For each split, report the Gini index value. Try to select the splits that obtain the best impurity measure.

(You do not need to worry about overfitting here: the resulting classification tree should have enough splits to fit the training data without error. Don't worry if your results are not optimal or super-accurate, as long as they are "in the ballpark".)

Problem 14

[5 points]

Learning objectives: basics of the k-NN method.

In this task, you will apply the *k-nearest neighbour* (k-NN) classifier by hand on a toy data set. You should be able to do this with pen and paper.

We will use the training dataset $D = \{(x_i, c_i)\}_{i=1}^{14}$, shown below, where $x_i \in \mathbb{R}$ are the covariates and $c_i \in \{-1, +1\}$ are the classes.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x_i	0.0	2.0	3.0	5.0	6.0	8.0	9.0	12.0	13.0	15.0	16.0	18.0	19.0	21.0
c_i	+1	+1	+1	-1	+1	+1	+1	-1	-1	-1	+1	-1	-1	-1

Task a

Where are the classification boundaries for the 1-NN and 3-NN classifiers? What are the respective classification errors on the training dataset?

Task b

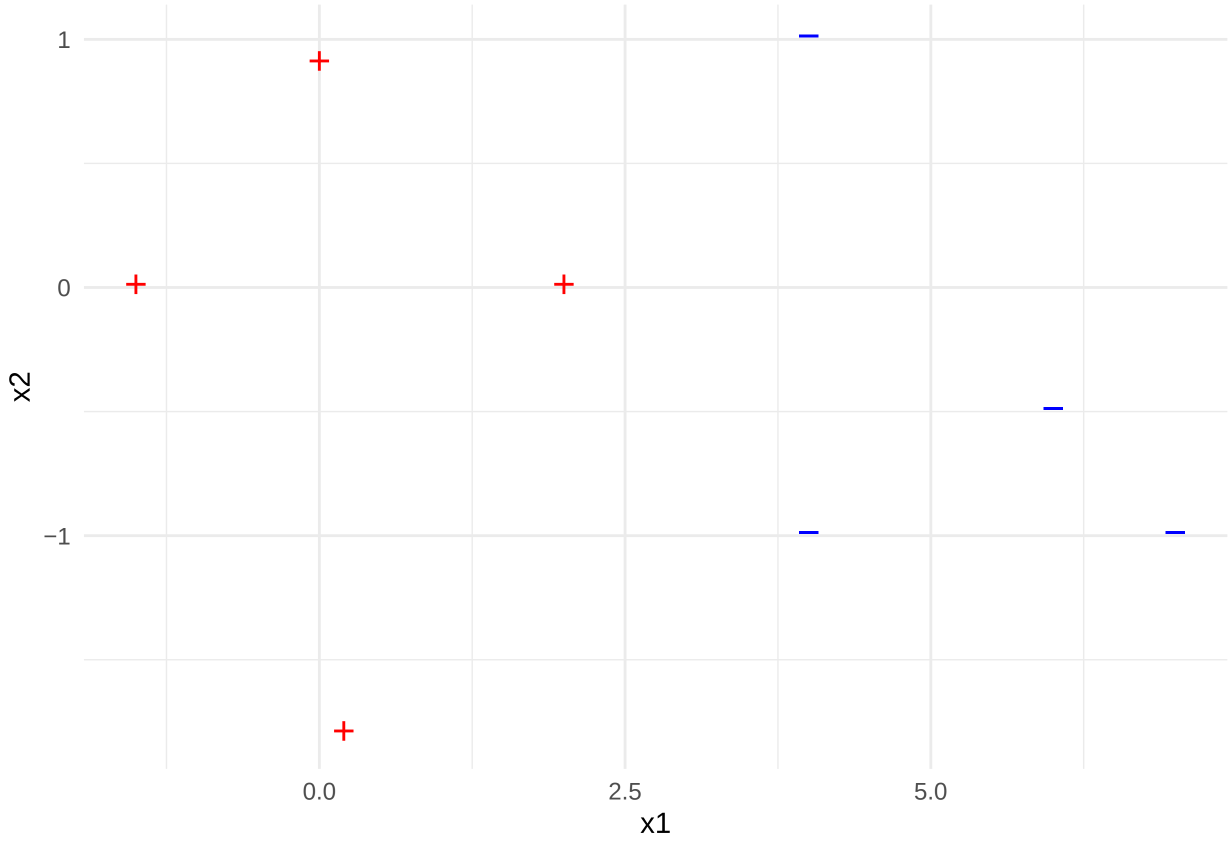
How does the choice of k in k-NN affect the classification boundary (not in the above example but in general)? Give examples of the behaviour for extreme choices (very small or large k).

Problem 15

[6 points]

Topic: SVM

In this problem, you will study the support vector machine (SVM) classifier on the toy data set shown below.



	x1	x2	class
A	-1.5	0.0	1
B	0.0	0.9	1
C	0.2	-1.8	1
D	2.0	0.0	1
E	4.0	1.0	-1
F	4.0	-1.0	-1
G	6.0	-0.5	-1
H	7.0	-1.0	-1

Task a

Find a separating hyperplane with the largest margin on the data set above. Write down the equation for this hyperplane and report the margin size.

Which of the points (A–H) are support vectors?

Hint: You can answer without mathematical proofs. You can do it simply by geometric intuition.

Problem 16

[2 points]

Objectives: self-reflection, giving feedback on the course

Task a

- Write a learning diary of the topics of lectures 5-8 and this exercise set.

Instructions

Guiding questions: What did I learn? What did I not understand? Was there something relevant to other studies or (future) work? Your reply should be 1-3 paragraphs of text. You can also give feedback on the course.

Task b

Give an estimate of the hours used in solving the problems in this exercise set.