Exercise Set 1

- Submit the answer via Moodle at the latest on Wednesday, 15 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 workshops.
- Your answer will be peer-reviewed by you and randomly selected by 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.
- Main source material: ISLR_v2, Chapters 1-3 and 5. Please feel free to use other resources as well. ISLR_v2 refers to the course textbook: James et al., 2021. An Introduction to Statistical Learning with Applications in R, 2nd edition. Springer. ISLP refers to the Python variant ("In Introduction to Statistical Learning with Applications in Python"). The contents of the two books are almost identical, but there are some minor differences, e.g., in section numbering, i.e., you can mostly use the books interchangeably.
- 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. It is challenging to produce correctly formatted PDF files with Jupyter Notebooks: remember to check the created PDF. I recommend using R Markdown instead of Jupyter notebooks.
- You can solve the problems at your own pace. However, we have given a suggested study schedule below. If you follow the plan, you can do the problems after we have covered the topics in class.

How to prepare your solution

We recommend using RStudio Desktop and R Markdown to make PDF documents for your solutions. R Markdown supports LaTeX math notation and several output formats, including PDF, and languages including R and Python/SciPy. This file has been made using R Markdown. You can familiarise yourself with R Markdown by going through a brief tutorial course at https://rmarkdown.rstudio.com/lesson-1.html, after which you can use the book by Xie et al. for reference. An alternative to R Markdown with no dependency on R is Quarto.

Suggested study schedule for Block 1

Please find below a suggested study schedule for the first four lectures (L1-L4) and Exercise Set 1 (E1), with *planned* topics for each class (=there may be some minor changes).

1 November: L1 Practicalities and introduction

- Read ISLR_v2, Section 1, before the lecture
- Topics in the class:
  - Practical matters and conduct in the course
  - Introduction to other students
  - What is machine learning, and why is it important
  - Applications of machine learning
- Exercises from E1 to do after the lecture and before the following lecture:
  - P1

3 November: L2 data wrangling, statistical learning, and linear models

- Read ISLR_v2, Sections 2-2.2 and 3-3.1, before the lecture and do the quiz.
- Topics in the class:
  - Data wrangling
  - Tools of machine learning
  - Ingredients of machine learning: task, computational problem, model, data, algorithm, features
  - Introduction to supervised learning
  - linear models (will continue in L3)
- Exercises from E1 to do after the lecture and before the next lecture:
  - You start solving P2, but cross-validation will probably be covered on L3. If the concepts seem confusing, you may want to wait until after L3.

8 November: L3 evaluation

- Read ISLR_v2, Sections 3.2-3.3 and 5-5.1, before the lecture and do the quiz.
- Topics in the class:
  - linear regression (continuing from L2)
  - bias-variance tradeoff
  - validation-set approach and cross-validation
  - estimating model parameters (will continue in L4)
- Exercises from E1 to do after the lecture and before the following lecture:
  - P2 and P3
  - Start doing P4. Notice that P4 is a complex problem. If you feel stuck, do what you can and do the other problems first.
  - Attend the extra exercise workshop on 9 November!

10 November: L4 estimating model parameters and controlling flexibility

- Read ISLR_v2, Sections 5.2 and 6-6.2, before the lecture and do the quiz.
- Topics in the class:
  - estimating model parameters (continuing from L3)
  - controlling the model complexity
  - subset selection
  - regularisation
- Exercises from E1 to do after the lecture and before 15 November:
  - P5 and P6
  - Finish P4 after doing other problems first.
  - Finally, write the learning diary (P7)
  - Attend the exercise workshop on 13 November!

15 November: E1 Done

- Submit your solutions into Moodle.
- Review your answer and the solutions of others in Moodle by the given deadline.
- You can do the reviews in the exercise workshops.

Notice that the bulk of the workload is after L3 and L4. Please remember to book enough time in your schedule to solve the problems. It helps if you have read the respective book chapters recommended in the lecture schedule above and attended the classes and the exercise workshops. Avoid leaving everything to the last day!

Problem 1

[6 points]

Objective: familiarity with tools, basic description of the data set, familiarisation with the term project data

In this problem, you will preprocess and explore a data set about atomic structures of molecules. The data set `p1.csv` is a subset of the **GeckoQ** data set from the term project.

Lab section 2.3 of ISLR_v2 (or ISLP) contains useful information for solving this problem.

The instructions below are in R and Python. Before reading the data into R or Python, you can view it in Excel or a text editor. For a good book about the topic, we recommend R for Data Science.

To use Python in R Markdown, you *might* need to specify the path to your Python executable using `reticulate::use_python(path_to_python)`, unless your default Python environment has everything installed.

Task a

Read `p1.csv` into a data frame and familiarize yourself with the data. You may want to read the data description from the term project.

Drop the columns `["id", "SMILES", "InChlKey"]`.

Task b

Select the columns `["pSat_Pa","NumOfConf","ChemPot_kJmol"]` from the data frame and print their summary statistics.

Tip: In Python you can use `pd.describe()`. In R you can use `summary()`.

Task c

Extract the data in the column `ChemPot_kJmol` of the data frame to an array. Calculate the mean and standard deviation of this array.

Tip: In Python you can use `.values`. In R, you can extract a column from a data frame `df` into an array using `df$ChemPot_kJmol` or `df[,"ChemPot_kJmol"]`.

Note: In Python, did you notice a difference between the mean and standard deviation calculated using pandas in Task b and numpy? The latter uses `float.64` as the default `dtype` whereas pandas uses `float.32` for computation. As such, numpy often returns more precise results for computations.

Task d

Produce side-by-side plots of:

- a histogram of `pSat_Sa` in base 10 logarithmic units.
- a boxplot of `NumOfConf`.

Tip: In Python, you can use `hist()` and `boxplot()` in the `matplotlib` package. In R, you can use `hist` and `boxplot`. The command `par(mfrow=c(1,2))` divides the plot window into two regions so that you can visualize the 2 plots simultaneously.

Task e

Produce a scatterplot matrix of the variables `["MW", "HeatOfVap_kJmol", "FreeEnergy_kJmol"]`.

Tip: In Python you can use `seaborn.pairplot()`. In R, you can use `pairs()`.

Problem 2

[8 points]

Learning objective: learning linear regression, concrete use of validation set, k-fold cross-validation, using regression models from ML libraries, and generalisation

In this problem, you will fit regression models and study their losses. One of the purposes of this problem - in addition to theory - is to make you more comfortable with various machine learning workflows.

Sections 5.1 and 5.3.2 (lab section) of ISLR_v2 contain helpful information for solving this problem.

Tasks a-b use a synthetic data set, and Tasks c-d use a real data set:

- The synthetic data are given in the CSV files `train_syn`, `valid_syn`, and `test_syn` (the *training set*, *validation set*, and *test set* respectively).
- The real data are meteorological forecasts and geographic data from Cho et al. (2020)[1]. They are given in the CSV files `train_real` and `test_real` (the *training set* and *test set* respectively).

Task a

In this task, you will fit polynomials $\hat{y} = \sum_{k=0}^{p} w_k x^k$ to the synthetic data for several polynomial degrees p by using ordinary least squares (OLS) regression.

Produce the following table:

Degree	Train	Validation	Test	TestTRVA	CV
0	?	?	?	?	?
1	?	?	?	?	?
2	?	?	?	?	?
3	?	?	?	?	?
4	?	?	?	?	?
5	?	?	?	?	?
6	?	?	?	?	?
7	?	?	?	?	?
8	?	?	?	?	?

where:

- `Train` is the training loss (train model on training set, report error on *training* set)
- `Validation` is the validation loss (train model on training set, report error on *validation* set)
- `Test` is the testing loss (train model on training set, report error on *test* set)
- `TestTRVA` is another testing loss (train model on the *combined training and validation data*, report error on test set)
- `CV` is the MSE from 5-fold cross-validation on the combined training and validation data

Explain how you would choose the polynomial order if given a combined training and validation set when the losses on the test set would be unknown.

Task b

For each value of $p \in \{0, 1, 2, 3, 4, 8\}$, produce a plot showing the points (x_i, y_i) in the training set and the fitted polynomial in the interval $[-3, 3]$.

[1] Cho, D., Yoo, C., Im, J., Cha, D., 2020. Comparative Assessment of Various Machine Learning-Based Bias Correction Methods for Numerical Weather Prediction Model Forecasts of Extreme Air Temperatures in Urban Areas. Earth and Space Science 7. https://doi.org/10.1029/2019EA000740

Make sure to plot continuous curves for the polynomials (using, e.g., `x <- seq(from=-3, to=3, length.out=256)` in R or `x = np.linspace(start=-3, stop=3, num=256)` in Python) and not only lines connecting the values of x appearing in your data!

Task c

In this task, you will fit the following regressors to the real data to predict the next day's maximum temperature (variable `Next_Tmax`):

- dummy model (see the discussion below)
- OLS linear regression (simple baseline)
- random forest (RF)
- support vector regression (SVR)
- one more regression model implemented in your machine learning library not mentioned above.

Produce the following table:

Regressor	Train	Test	CV
Dummy	?	?	?
OLS	?	?	?
RF	?	?	?
SVR	?	?	?
?	?	?	?

where `Train` is the training loss, `Test` is the testing loss, and `CV` is the loss for 10-fold cross-validation.

Using the table, answer the following:

1. Which regressor is the best? Why?
2. How does `Train` compare to `Test`? How does `CV` compare to `Test`?
3. How can you improve the performance of these regressors (on this training set)?

About the dummy model

A *dummy model* is a supervised learning model that gives the same constant output regardless of the values of the covariates. The outcome might be such that it minimises the loss of the training data. For example, if we have OLS regression, this would be the mean of the dependent variable. Including the dummy model in your list is often helpful because it costs you almost nothing and gives you a good baseline for the performance of your "real" supervised learning models.

In addition to the dummy model, it is always helpful to include a simple *baseline*, such as OLS linear regression for regression problems and logistic regression for classification problems. It happens surprisingly often that your complex machine-learning model does not substantially outperform the simple baseline model.

Problem 3

[6 points]

Learning objectives: bias and variance and model flexibility

In this problem, you will study the bias-variance decomposition in the context of model selection.

Section 2.2 of ISLR_v2 will be helpful in solving this problem.

Task a

Describe the typical behaviour of the following terms, as we go from less flexible to more flexible statistical learning methods:

- training error and testing error
- (squared) bias
- variance
- irreducible (or Bayes) error.

Explain why each term has the described behaviour.

You can describe the behaviours in words or you can sketch them as curves. In the sketches the x-axis should represent the flexibility of the method, and the y-axis should represent the values for each term. There should be five curves in total so make sure to label each one.

Task b

In this task, you will test the bias-variance trade-off in practice using polynomial functions.

Assume a data point (x, y) is generated as $y = f(x) + \epsilon$ where $f(x) = 2 - x + x^2$, $\epsilon \sim \text{Normal}(0, 0.4^2)$, and $x \sim \text{Uniform}(-3, 3)$. Assume a polynomial regression function $\hat{f}$ of degree p is trained using a data set D of n data points (x, y).

According to Eq. (2.7) of ISLR_v2, you can decompose the expected squared loss at $x = 0$ as a sum of irreducible error, the bias term, and the variance term as

$$\text{squared loss} = \text{irreducible} + \text{bias}^2 + \text{var},$$

or:

$$E_D\left[(y_0 - \hat{f}_0)^2\right] = E_D\left[(y_0 - f_0)^2\right] + \left(E_D[\hat{f}_0] - f_0\right)^2 + E_D\left[(\hat{f}_0 - E_D[\hat{f}_0])^2\right],$$

where $f_0 = f(0)$ is the true function value at $x = 0$ and $\hat{f}_0 = \hat{f}(0)$ is the regression model prediction at $x = 0$. The expectation E_D is over the training data set D.

(i) Produce the following table:

Degree	Irreducible	BiasSq	Variance	Total	MSE
0	?	?	?	?	?
1	?	?	?	?	?
2	?	?	?	?	?
3	?	?	?	?	?
4	?	?	?	?	?
5	?	?	?	?	?
6	?	?	?	?	?

where

- `Degree` is the polynomial degree of the regression function.

- `Irreducible`, `BiasSq`, `Variance` are as described above. `MSE` is the mean squared error.
- `Total` is the sum of `Irreducible`, `BiasSq`, `Variance`.

(ii) Plot these four terms (squared loss, irreducible error, bias term, variance term) as a function of polynomial degree (i.e. make four curves).

(iii) Do the terms behave as you would expect from the discussion in Task a? Does `Total` approximately equal `SqLoss`?

How to compute the terms

To compute the expectations for degree p (one row in the table):

- Generate 1000 *training* sets each of which contains $n = 10$ data items, using the data generating process described above.
- For each training set:
  - train a polynomial regression function $\hat{f}(x)$ with degree p.
  - generate one *test* data point at $x = 0$, i.e., $(0, y_0)$, where $y_0 = f(0) + \epsilon$ where f and ϵ are as described above.
  - save the following numbers: $f(0)$, y_0, and $\hat{f}(0)$.

If you construct a table with 1000 rows and three columns $f(0)$, y_0 and $\hat{f}(0)$, then you can easily estimate the required expectations.

Problem 4 (Hard)

[6 points]

Topic: theoretical properties of generalisation loss and OLS linear regression [Ch. 2-3]

Consider a linear regression model $\hat{f}(\mathbf{x}) = \hat{\beta}^T\mathbf{x}$, where $\hat{\beta} \in \mathbb{R}^p$ is fit by ordinary least squares (OLS) to a set of training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where the pairs $(\mathbf{x}_i, y_i)$ have been drawn at random with replacement from a finite population, where $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, and $i \in \{1, \ldots, n\}$. Suppose we have a testing data $(\overline{\mathbf{x}}_1, \overline{y}_1), \ldots, (\overline{\mathbf{x}}_m, \overline{y}_m)$ drawn in the same way from the same population. Denote

$$L_{train} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\beta}^T\mathbf{x}_i\right)^2$$

and

$$L_{test} = \frac{1}{m}\sum_{i=1}^{m}\left(\overline{y}_i - \hat{\beta}^T\overline{\mathbf{x}}_i\right)^2.$$

The expectations below are defined over the resampling of both the training and testing data.

Task a

Prove that

$$E\left[L_{test}\right] = E\left[\left(\overline{y}_1 - \hat{\beta}^T\overline{\mathbf{x}}_1\right)^2\right].$$

Task b

Prove that L_{test} is an unbiased estimate of the generalisation error for the OLS regression.

Task c

Prove that $E\left[L_{train}\right] \leq E\left[L_{test}\right]$.

Task d

Explain how the task result above is related to the generalisation problem in machine learning.

About expectations

There are (at least) two ways to define the expectations $E\left[L_{train}\right]$ and $E\left[L_{test}\right]$:

1. sampling the testing data while keeping the training data fixed, in which case $\hat{\beta}$ is constant, resulting in $E\left[L_{test}\right]$ being the generalisation error for this particular training data and regression solution.
2. sampling both the training and testing data, in which $\hat{\beta}$ is a random variable and a function of the training data, resulting in $E\left[L_{test}\right]$ being the generalisation error averaged over all possible training data sets.

It is sometimes tricky to keep track of which is a random variable and over what to take expectations, and often, it is not explicitly mentioned. Both of the definitions above make sense but have slightly different interpretations. Please use the second definition in this task. You can read, e.g., Bengio et al. (2004)[2] for a more in-depth discussion of the expectations if interested, where "PE" of Eq. (1) corresponds to the first and "EPE" of Eq. (2) corresponds to the second definition.

[2]Bengio, Y., Grandvalet, Y., 2004. No unbiased estimator of the variance of K-fold cross-validation. J. Mach. Learn. Res. 5, 1089-1105. https://www.jmlr.org/papers/v5/grandvalet04a.html

Problem 5

[6 points]

Objective: properties of estimators [Ch 3]

So far, we have tried only to estimate the loss (to choose the best model). It is also essential to understand the model and diagnose potential problems.

In this problem, we study 1-variable linear regression $\hat{y} = w_0 + w_1 x$ using the four data sets named `d1.csv`, `d2.csv`, `d3.csv`, and `d4.csv`.

Task a

For each data set, fit an OLS linear regression and report:

- the intercept term estimate, standard error, and p-value,
- the slope term estimate, standard error, and p-value,
- the R-squared value of the model.

The slope term may be positive (or negative) with high confidence. Can you safely conclude that when `x` increases (or decreases), `y` tends to increase (and vice versa)?

Tip: See Section 3.6.2 of ISLR_v2.

Task b

Make a plot of each data set showing a scatterplot of `x` vs `y` along with the fitted regression line. What commonalities do you notice between the data sets and their fitted models?

Task c

Sect. 3.3.3 of ISLR_v2 lists six potential problems with linear regression models. Which of the six problems would (potentially) apply to each dataset? What tricks and plots did you use to detect and diagnose the problems? Produce at least one diagnostic plot that shows these problems (other than just plotting `x` vs `y`, which you can do here because the data is 1-dimensional and which would not work for higher-dimensional data sets).

Problem 6

[6 points]

Objective: properties of estimators and Bootstrap [Ch 3 & 5.2]

Task a

Compute the standard errors for the regression coefficient estimates for the data set `d2.csv` of the previous problem using bootstrap. Compare the bootstrap standard errors to the ones you got in Task A of the previous problem; which of the estimates is more trustworthy and why?

Task b

Describe briefly in your own words how the bootstrap algorithm computes the standard errors for the intercept and slope parameters in the task above.

Task c

In bootstrap, you sample n data points from a population of n points with replacement. Argue that the probability that the jth observation is *not* in the bootstrap sample is about 0.368 when n is very large.

Hints

Please see lab Section 5.3.4 of ISLR_v2 or 5.3.3 of ISLP, subsection "Estimating the Accuracy of a Linear Regression Model", for guidance for Task a. For Task c, you can find lots of hints in Problem 2 in Section 5 of ISLR_v2, and by observing that $\lim_{n\to\infty}(1-1/n)^n = 1/e \approx 0.368$.

Problem 7

[2 points]

Objectives: self-reflection, giving feedback on the course

Task a

Write a learning diary of the topics of lectures 1-4 and this exercise set.

Guiding questions: What did I learn? What did I not understand? Was there something relevant for other studies or (future) work? The length of 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.