**Deadline:** Thursday, December 14, 2023, 15:00

Before solving the exercises, read the instructions on the course website.

- For each theoretical problem, submit a single `pdf` file that contains your answer to the respective problem. This file may be a scan of your (legible) handwriting.
- For each practical problem, submit a single `zip` file that contains
  - the completed jupyter notebook (`.ipynb`) file,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows all your results.
- For the bonus question, submit a single `zip file` that contains
  - a `pdf` file that includes your answers to the theoretical part,
  - the completed jupyter notebook (`.ipynb`) file for the practical component,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows your results.
- Every team member has to submit a signed Code of Conduct.
- **IMPORTANT** You must make the team on CMS *before* you upload the solutions. If you upload the solutions first and create the team after it, the solution will not show for the new team member!

**Problem 1** (T, 4 Points). **Cross-Validation.**

1. [$2pts$] Explain the impact of the value for $k$ in $k$-fold cross validation. Where does $k$-fold CV fit in between the validation set approach and LOOCV and what is the advantage of using it?
2. [$2pts$] Explain how an outlier in a dataset can affect scores of LOOCV. In this setting, can $k$-fold cross-validation address the drawbacks of LOOCV?

**Problem 2** (T, 10 Points). **Model selection in Linear Regression**

Given is a training set consisting of samples $\mathbf{X} = (x_1, x_2, ..., x_N)^T$ with respective regression targets $\mathbf{y} = (y_1, y_2, ..., y_N)^T$ where $x_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. Alice fits a linear regression model $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$ to the dataset using the closed-form solution for linear regression (normal equations).

Bob has heard that by transforming the inputs $\mathbf{x}_i$ with a vector-valued function $\boldsymbol{\phi}$, he can fit an alternative function $g(\mathbf{x}_i) = \mathbf{v}^T \boldsymbol{\phi}(\mathbf{x}_i)$, using the same procedure (solving the normal equations). He decides to use a linear transformation $\boldsymbol{\phi}(\mathbf{x}_i) = \mathbf{A}^T \mathbf{x}_i$, where $\mathbf{A} \in \mathbb{R}^{D \times D}$ has full rank.

1. [$4pts$] Show that Bob's procedure will fit the same function as Alice's original procedure, that is $f(x) = g(x)$ for all $x \in \mathbb{R}^D$ (given that $w$ and $v$ minimize the training set error).
2. [$3pts$] Can Bob's procedure lead to a lower training set error than Alice's if the matrix $A$ is not of full rank i.e. non-invertible? Explain your answer.
3. [$3pts$] Explain if Alice and Bob will fit the same function or not in the case they perform ridge regression with the same regularization strength $\lambda$.

**Problem 3** (T, 6 Points). **The Bootstrap.**

We will now derive the probability that a given observation is part of a bootstrap sample of size $n$. Suppose that we obtain a bootstrap sample from a set of $n$ observations.

1. [$2pts$] What is the probability that the first bootstrap observation is the $j$th observation from the original sample? Justify your answer.
2. [$1pts$] Argue that the probability that an observation is not in the bootstrap sample is $(1 - 1/n)^n$.
3. [$1pts$] Derive the probability that an observation is there once in the bootstrap sample of size $n$.
4. [$2pts$] With the increase in sample size $n$, what is the behavior of the probabilities that an observation is not in the bootstrap sample, and an observation is in the bootstrap once? Comment.

**Problem 4** (P, 15 Points). **Programming Exercise.**
Download *exercisse.ipynb* from the CMS

1. [$2pts$] Implement the least squares regression using numpy functions with vectorization.
2. [$3pts$] Implement the fit ridge regression function using numpy functions with vectorization.
3. [$1pts$] Implement the generated predictions function using numpy functions with vectorization.
4. [$2pts$] Implement the mean square error function using numpy functions with vectorization.
5. [$3pts$] Implement the code to compute subset indices *train indices, val indices* for custom k-fold cross validation implementation.
6. [$1pts$] Plot the custom fold cross-validation implementation for $k = \{2, 3, \ldots, 10\}$
7. [$3pts$] Implement the code to compute the subsets $(X_{train}, y_{train})$ and $(X_{val}, y_{val})$ for custom LOOCV implementation

**Problem 5** (Bonus). **Comparing Linear Regression models**

1. **Theoretical 1**. In this problem, we compare linear regression models after feature transformation. We want to perform regression on a dataset consisting of $N$ samples $\boldsymbol{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} \in \mathbb{R}^N$). Assume that we have fitted an L2-regularized linear regression model and obtained the optimal weight vector $\boldsymbol{w}^* \in \mathbb{R}^D$ as

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{argmin} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{x}_i - y_i)^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w}$$

   Note that there is no bias term. Now, assume that we obtained a new data matrix $\boldsymbol{X}_{\text{new}}$ by scaling all samples by the same positive factor $a \in (0, \infty)$. That is, $\boldsymbol{X}_{\text{new}} = a\boldsymbol{X}$ (and respectively $\boldsymbol{x}_i^{\text{new}} = a\boldsymbol{x}_i$).

   (a) Find the weight vector $\boldsymbol{w}_{\text{new}}$ that will produce the same predictions on $\boldsymbol{X}_{\text{new}}$ as $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$.

   (b) Find the regularization factor $\lambda_{\text{new}} \in \mathbb{R}$, such that the solution $\boldsymbol{w}^*_{\text{new}}$ of the new $\ell_2$-regularized linear regression problem

$$\boldsymbol{w}^*_{\text{new}} = \underset{\boldsymbol{w}}{argmin} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{X}_i^{\text{new}} - y_i)^2 + \frac{\lambda_{\text{new}}}{2} \boldsymbol{w}^T \boldsymbol{w}$$

   will produce the same predictions on $\boldsymbol{X}_{\text{new}}$ as $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$. Provide a mathematical justification for your answer.

2. **Theoretical 2**. In this problem, we compare linear regression models after re-sampling our dataset. Let's assume we have a dataset where each data point $(x_i, y_i)$ is weighted by a scalar factor, which we will call $a_i \in \mathbb{R}_+$, i.e. we will assume that $a_i > 0$ for all $i$. This makes the sum of squares error function look like the following:

$$E_{\text{weighted}}(\boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{N} a_i (\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) - y_i)^2$$

   (a) Find the equation for the value of $\boldsymbol{w}$ that minimizes this error function.

   (b) Explain how this weighting or sampling factor, $a_i$, can be interpreted in terms of the variance of the noise on the data, and data points for which there are exact copies in the dataset.
   *Hint: The ordinary least squares can be modeled in a probabilistic context as i.i.d random variables* $y_i \sim N(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i), \beta^{-1})$ *with a common noise precision of* $\beta$.