Lab Class ML:VI

Until Wednesday, Feb. 4th, 2026, 11:59 pm CET, solutions to the following exercises must be submitted as one zip-file named `ML25-ex6-group<your-group-number>.zip` via Moodle:
1 2, 4, 5, 6, 7, 8.

Exercise 1 : Probability Basics (3 Points)

Explain the following concepts briefly:

(a) Ω

(b) Event A

(c) $\emptyset$

(d) $P(A \mid B)$

(e) Statistical independence of two events

(f) Mutually exclusive events

Exercise 2 : Probability Basics (1 Points)

Which of the following statements are true?

- ☐ According to the Kolmogorov axioms the statement $P(A) - P(\overline{A}) = 0$ holds.
- ☐ A function that fulfills the Kolmogorov axioms is a probability measure.
- ☐ Two events are statistically independent $\Leftrightarrow P(A \cap B) = P(A) + P(B)$.
- ☐ Each subset A of a sample space Ω is an event.

Exercise 3 : Probability Basics (Kolmogorov) (0 Points)

Prove the implications of the Kolmogorov axioms from the lecture (Theorem 7).

Exercise 4 : Bayes' Rule (2+3=5 Points)

A hospital database contains diagnoses ($C_1 \ldots C_5$) for 8 patients along with binary observations of symptoms $S_1 \ldots S_9$:

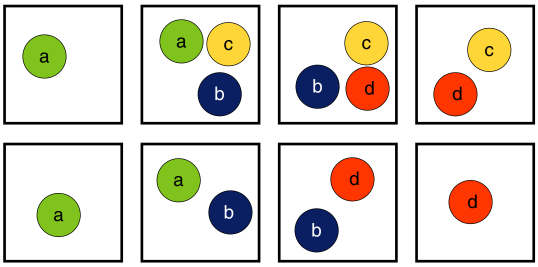
Patient	**Diagnosis**	**Symptoms**								
		S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
1	C_1	1	0	1	0	1	0	0	0	0
2	C_2	0	1	0	1	1	0	1	0	0
3	C_3	1	0	1	0	0	1	0	1	0
4	C_4	0	1	0	1	1	0	1	0	0
5	C_3	1	0	1	0	0	0	0	1	0
6	C_5	0	0	0	0	1	0	0	0	1
7	C_3	1	0	1	0	0	1	0	0	0
8	C_2	0	1	0	0	0	0	1	0	0

(a) Compute based on the database the prior probabilities $P(C_i)$ for each diagnosis.

(b) Compute based on the database the posterior probabilities $P(C_i \mid S_4)$ for each diagnosis.

Exercise 5 : Probability Basics (Conditional Independence) (1+1+1+1+1+1=6 Points)

There are eight boxes containing different colored balls as shown in the illustration below:



The balls can be green, blue, yellow, or red (also marked a, b, c, d in the figure). When picking one of the eight boxes at random, let A refer to the event "box contains a green ball," B to the event "box contains a blue ball," C to the event "box contains a yellow ball," and D to the event "box contains a red ball." Hence, $A \cap B$ is the event "box contains both a green and a blue ball," etc.

(a) Calculate $P(A)$, $P(B)$, $P(C)$, and $P(D)$.

(b) Calculate $P(A \cap B)$, $P(A \cap C)$, $P(B \cap C)$, and $P(B \cap D)$.

(c) Check all that apply:

- ☐ The events A and B are statistically independent.
- ☐ The events A and C are statistically independent.
- ☐ The events B and C are statistically independent.
- ☐ The events B and D are statistically independent.

(d) Calculate $P(A \mid C)$, $P(B \mid C)$, and $P(A \cap B \mid C)$.

(e) Calculate $P(B \mid D)$, $P(C \mid D)$, and $P(B \cap C \mid D)$

(f) Check all that apply:

- ☐ The events A and B are conditionally independent given C.
- ☐ The events B and C are conditionally independent given D.

Exercise 6 : Naïve Bayes (3+2=5 Points)

Given is the following dataset to classify whether a dog is dangerous or well-behaved in character:

Color	**Fur**	**Size**	**Character** (C)
brown	ragged	small	well-behaved
black	ragged	big	dangerous
black	smooth	big	dangerous
black	curly	small	well-behaved
white	curly	small	well-behaved
white	smooth	small	dangerous
red	ragged	big	well-behaved

(a) Determine the parameters $P(A_i)$ and $P(B_{j=x_j} \mid A_i)$ for a Naïve Bayes classifier on this dataset. Use a table like the following to report the $P(B_{j=x_j} \mid A_i)$:

Attribute	Value (x_j)	Class (c_i)	$P(B_{j=x_j} \mid A_i)$
Color	brown	well-behaved	...
Color	brown	dangerous	...
...	...	...	...

(b) Classify the new example $\mathbf{x} = (\mathsf{black}, \mathsf{ragged}, \mathsf{small})$ using a Naïve Bayes classifier with the parameters you calculated in (a).

Exercise 7 : Regularized Linear Regression (2+1+1=4 Points)

Suppose we are estimating the regression coefficients in a linear regression model by minimizing the regularized objective function

$$\mathcal{L}(\mathbf{w}) = \mathsf{RSS}_{\text{tr}}(\mathbf{w}) + \lambda \cdot R(\mathbf{w}),$$

(a) You are given a trained linear regression model with weight vector

$$\mathbf{w} = (w_0, w_1, w_2, w_3, w_4)^T = (10, 50, 4, -3, 5)^T$$

and the following statistics:

- Training RSS: $\mathsf{RSS}_{\text{tr}}(\mathbf{w}) = 120$
- Test RSS: $\mathsf{RSS}_{\text{test}}(\mathbf{w}) = 145$

Compute the regularized training objective $\mathcal{L}(\mathbf{w})$ for both ridge and lasso regression with $\lambda = 0.5$. Show all intermediate calculations.

(b) Without computing exact values, explain qualitatively how the following quantities change if λ is increased:

- the training RSS,
- the test RSS,

(c) You have a dataset with 100 features, but you suspect that only about 10 features are truly relevant.

Which regularization method would you choose, ridge or lasso, and why?

Exercise 8 : $\boxed{\text{P}}$ Classification with Naïve Bayes (1+1+1+1+1+1=6 Points)

In this exercise, you will implement the Naïve Bayes classifier for predicting whether a given text was written by a human or generated by a language model. To make this task a bit easier, you will use a modified version of the dataset where all texts have been converted to Bag-of-words representations. As usual, there are test cases provided to check your implementation steps. Submit the file with your predictions for the test set along with your other solutions.

Download and use these files from Moodle:

- *Text files for training, validation and test sets*:
  `texts-train.tsv`, `texts-val.tsv`, `texts-test.tsv` – These files contain the texts of the examples in the training, validation, and test sets, respectively. Use these files to extract features for the classification task.
- *Bag-of-words features for training, validation and test sets*:
  `bow-features-train.npy`, `bow-features-val.npy`, `bow-features-test.npy` – The function to load these features is already implemented in the template.
- *Labels for the training and validation sets*:
  `labels-train.tsv`, `labels-val.tsv` – The function to load the classes is already implemented in the template.
- *Template for the programming exercise:*
  `programming_exercise_statistical_learning.py`. It contains function stubs for each function mentioned below. To run the program, use the following command (make sure the above-mentioned files are located in the `data` folder):

  ```
  python3 programming_exercise_statistical_learning.py
  ```

(a) Implement the function `class_priors` to estimate the prior probabilities $P(A_i)$, where A_i is the event that an example has the class c, for all possible classes c occurring in a dataset D. The function receives an array of values $c(\mathbf{x})$ for the $\mathbf{x} \in D$ as input, and returns a Python dictionary mapping the distinct classes c to their prior probabilities.

(b) Implement the function `conditional_probabilities` to estimate the conditional probabilities $P(B_{j=x_j}|A_i)$, where $B_{j=x_j}$ is the event that the j'th feature has the value x_j in an example $\mathbf{x}$. The function receives two arrays as input, and returns a nested dictionary with the mapping class $c \rightarrow$ feature index $j \rightarrow$ feature value $x_j \rightarrow$ probability $P(B_{j=x_j}|A_i)$. Consider using the class `collections.defaultdict` from the Python standard library to make this easier.

(c) Implement the method `fit` of the class `NaiveBayesClassifier` using the functions implemented so far.

(d) Implement the method `predict` of the class `NaiveBayesClassifier`, which takes as input a single feature vector $\mathbf{x}$ and returns the most probable class according to the Naïve Bayes model.

(e) Implement the function `train_and_predict`, which shall fit a Naïve Bayes model on the training set, evaluate it on the validation set and return an array of predictions on test samples, similarly to previous exercises.

(f) Implement the function `extract_features` to extract useful features from the provided texts for the classification task. Re-run the training and prediction steps using these new features. Aim to achieve the highest possible performance on the validation set by carefully selecting and engineering features that enhance the classifier's performance.

Bonus: Challenge *(This exercise is optional)*
As in the previous exercise, optimize and tweak your implementation using everything you have learned. Once satisfied with the performance, classify the examples in the separate test dataset, and submit your classifications. *Hints:*

- Consider special handling for the case where a particular feature-value combination does not occur in the training data. Keyword: Laplace smoothing.