**INTRODUCTION MACHINE LEARNING**

**EXERCISE 6**

A red sign with white text

Description automatically generated

|  |
| --- |
| **TEACHER:** |
| Johannes Kiesel |

|  |
| --- |
| **GROUP:** |
| Group 16 |
|  |
| **SUBMITTED BY:** |
| Aaron Perez Herrera |
| Cesar Fernando Gamba Tiusaba |
| Chun Ting Lin |
| Olubunmi Emmanuel Ogunleye |

Exercise 1 : Probability Basics (1 Points)

Which of the following statements are true?

* According to the Kolmogorov axioms the statement P(A) - P(A) = 0 holds.
* A function that fulfills the Kolmogorov axioms is a probability measure.
* Two events are statistically independent P(An B) = P(A) + P(B).
* Each subset A of a sample space 12 is an event.

The true statements are.

* A function that fulfills the Kolmogorov axioms is a probability measure. (true)
* Each subset A of a sample space 1 is an event. (true)

Exercise 2 : Probability Basics (Kolmogorov) (0 Points)

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

The conditions to prove are the following ones:



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

A hospital database contains diagnoses (C1 ... C5) for 8 patients along with binary observations of symptoms S1 ... S9:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient** | **Diagnosis** | **Symptoms** | | | | | | | | |
| **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** |
| **1** | **C1** | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| **2** | **C2** | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| **3** | **C3** | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **4** | **C4** | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| **5** | **C3** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| **6** | **C5** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| **7** | **C3** | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| **8** | **C2** | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

1. Compute based on the database the prior probabilities P(Ci) for each diagnosis.

The total number of patients is 8.

* .The number of patients diagnosed with C1 is 1, so P(C1) = 1/8 = 0.125.
* The number of patients diagnosed with C2 is 2, so P(C2) = 2/8 = 0.25.
* The number of patients diagnosed with C3 is 3, so P(C3) = 3/8 = 0.375.
* The number of patients diagnosed with C4 is 1, so P(C4) = 1/8 =0.125.
* The number of patients diagnosed with C5 is 1, so P(C5) = 1/8 = 0.125.

1. Compute based on the database the posterior probabilities P(Ci | S4) for each diagnosis.

Posterior probabilities P(Ci|S4) of the diagnoses Ci given symptom S4 are calculated as follows:

The posterior probability of Ai given B is given by,

Hence, first P(B|Ai) is to be calculated.

Therefore, the posterior probability of S4 given C is calculated as:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

The next probabilities

Exercise 4

Part (a): Calculate P(A), P(B), P(C), and P(D):

Each event corresponds to the probability of a specific color of balls (green, blue, yellow, or red) being

in one of the eight boxes. Using the image provided:

Green balls (a) appear in 3 boxes.

Blue balls (b) appear in 4 boxes.

Yellow balls (c) appear in 2 boxes.

Red balls (d) appear in 4 boxes.

The probabilities are calculated as:

A math equations with numbers

AI-generated content may be incorrect.

Part (b): Calculate P(A and B), P(A and C), P(B and C), and P(B and D):

The joint probabilities are calculated by counting the boxes containing both colors:

A math equations with numbers

AI-generated content may be incorrect.

Part (c): Check for statistical independence:

Pair (A, B):

P(A) = 0.375, P(B) = 0.5, P(A and B) = 0.125 P(A) × P(B) = 0.375 × 0.5 = 0.1875

Since P(A and B) = 0.125 != 0.1875, A and B are not independent.

Pair (A, C):

P(A) = 0.375, P(C) = 0.25, P(A and C) = 0.125 P(A) × P(C) = 0.375 × 0.25 = 0.09375

Since P(A and C) = 0.125 != 0.09375, A and C are not independent.

Pair (B, C):

P(B) = 0.5, P(C) = 0.25, P(B and C) = 0.125 P(B) × P(C) = 0.5 × 0.25 = 0.125

Since P(B and C) = 0.125 = 0.125, B and C are independent.

Pair (B, D):

P(B) = 0.5, P(D) = 0.5, P(B and D) = 0.25 P(B) × P(D) = 0.5 × 0.5 = 0.25

Since P(B and D) = 0.25 = 0.25, B and D are independent.

Part (d): Calculate P(A|C), P(B|C), and P(A and B|C):

Conditional probabilities are calculated using the formula:

A black text on a white background

AI-generated content may be incorrect.

From the data:

A math equations with numbers

AI-generated content may be incorrect.

Part (e): Calculate P(B|D), P(C|D), and P(B and C|D):

Similarly, conditional probabilities given D are:

A black text on a white background

AI-generated content may be incorrect.

From the data:

A math equations with numbers

AI-generated content may be incorrect.

Part (f): Check for conditional independence:

Two events A and B are conditionally independent given C if:

A black text with a white background

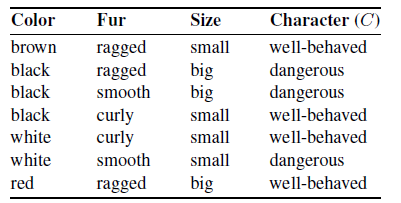
AI-generated content may be incorrect.

From (d), A number and exclamation mark

AI-generated content may be incorrect.Thus, A and B are not conditionally independent given C

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

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



1. Determine the parameters P(Ai) and P(Bj=xj | Ai) for a Naïve Bayes classifier on this dataset. Use a table like the following to report the P(Bj=xj | Ai):

Prior Probability each Class

|  |  |
| --- | --- |
| C1 (Well – Behaved) | = 4/7 = 0.57 |
| C2 (Dangerous) | = 3/7 = 0.43 |

Conditional Probability

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Value (xj)** | **Class (ci)** | **P (xj | ci)** |
| Color | Brown | Well-Behaved | 1/1 = 1.00 |
| Color | Brown | Dangerous | 0/1 = 0.00 |
| Color | Black | Well-Behaved | 1/3 = 0.33 |
| Color | Black | Dangerous | 2/3 = 0.67 |
| Color | White | Well-Behaved | 1/2 = 0.50 |
| Color | White | Dangerous | 1/2 = 0.50 |
| Color | Red | Well-Behaved | 1/1 = 0.00 |
| Color | Red | Dangerous | 0/1 = 0.00 |
| Fur | Ragged | Well-Behaved | 2/3 = 0.67 |
| Fur | Ragged | Dangerous | 1/3 = 0.33 |
| Fur | Smooth | Well-Behaved | 0/2 = 0.00 |
| Fur | Smooth | Dangerous | 2/2 = 1.00 |
| Fur | Curly | Well-Behaved | 2/2 = 1.00 |
| Fur | Curly | Dangerous | 0/2 = 0.00 |
| Size | Small | Well-Behaved | 3/4 = 0.75 |
| Size | Small | Dangerous | 1/4 = 0.25 |
| Size | Big | Well-Behaved | 1/3 = 0.33 |
| Size | Big | Dangerous | 2/3 = 0.67 |

1. Classify the new example x = (black, ragged, small) using a Naïve Bayes classifier with the parameters you calculated in (a).

X = (Color = Black, Fur = Ragged, Size = Small) 🡪 Y(x) predicted?

Exercise 6 : 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 in the data folder):

python3 programming\_exercise\_statistical\_learning.py

1. Implement the function class\_priors to estimate the prior probabilities P(Ai), where Ai 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(x) for the x ∈ D as input, and returns a Python dictionary mapping the distinct classes c to their prior probabilities.
2. Implement the function conditional\_probabilities to estimate the conditional probabilities P(Bj=xj |Ai), where Bj=xj is the event that the j’th feature has the value xj in an example x. The function receives two arrays as input, and returns a nested dictionary with the mapping class c→feature index j → feature value xj →probability P(Bj=xj |Ai). Consider using the class collections.defaultdict from the Python standard library to make this easier.
3. Implement the method fit of the class NaiveBayesClassifier using the functions implemented so far.
4. Implement the method predict of the class NaiveBayesClassifier, which takes as input a single feature vector x and returns the most probable class according to the Naïve Bayes model.
5. 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.
6. 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.

The solution to this part is attached with the Zip file.