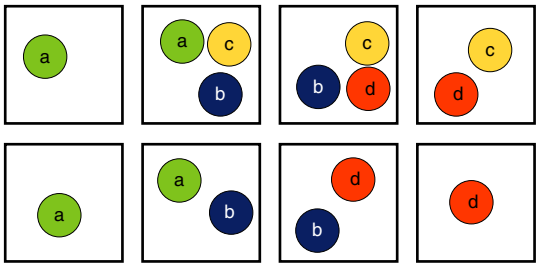
Lab Class ML:V

Exercise 1 : Probability Basics (Conditional Independence)

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 in case the colors are hard to distinguish). 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," and so on.

(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 gived D.

Exercise 2 : Bayes

A hospital database contains diagnoses (diseases) along with observed symptoms, collected during the past years. Let following representative dump be given, where the diseases are sorted temporally according to their appearances. Note, that the symptoms for a disease can change over different time periods.

Year	Diagnosis	Symptom	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
2001	D_1		X		X		X				
2002	D_2			X		X	X		X		
2003	D_3		X		X			X		X	
2004	D_4			X		X	X		X		
2005	D_3		X		X					X	
2006	D_5						X				X
2007	D_3		X		X			X			
2008	D_2			X					X		

(a) Compute the prior probabilities $P(D_i)$.

(b) Compute the posterior probabilities $P(D_i \mid S_4)$ of the diagnoses D_i given symptom S_4.

Exercise 3 : Probability Basics (Kolmogorov)

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

Exercise 4 : P Website Function Classifier

Your task is to implement a website classifier for the website main function dataset from the previous exercises, but this time using the Naive Bayes approach. The implementation of such a classifier requires (1) a loader for the dataset, (2) a vectorization procedure for turning the JSON into matrix form, (3) a feature scaling and discretization scheme, and (4) the actual Naive Bayes estimator.

Feel free to reuse code from previous exercises where applicable.

(a) Load the provided dataset and convert the samples and their features into an $n \times m$ matrix (use all features, so $m = 16$). Map each unique text label to an integer value and covert the ground-truth labels into an n-dimensional numerical vector. Using Pandas can be useful, but plain NumPy is totally sufficient.

(b) Split off 200 samples as a hold-out test set and use the rest as training data.

(c) In order to estimate the individual probabilities, the feature values need to be rescaled to a common scale and discretized. Develop a suitable scaling and discretization procedure with an adjustable number of buckets (resolution). A resolution of 20 means that after discretization, a real-valued feature $x_i \in [0, k]$ maps to one of 20 possible integer values $x_{d_i} \in [0, 1, \ldots, 19]$.

(d) Develop the class `NaiveBayesClassifier` with the following interface:

```
class NaiveBayesClassifier:
    def __init__(self):
        self.priors = None
        self.p_cond = None
        self.classes = []

    def fit(self, X, y):
        pass

    def predict(self, X):
        pass
```

`fit(X, y)` takes an $n \times m$ feature matrix and a numerical label vector and estimates the class priors $\hat{P}(A_m)$ as well the the conditional probabilities $\hat{P}(B_{i=x_i}|A_m)$. `predict(X)` takes another $n \times m$ feature matrix and returns the numerical representations of the most likely classes as an n-dimensional vector.

(e) Train your classifier with a discretization resolution of 20 buckets and evaluate it on the test split by calculating its prediction accuracy. Is the classification quality satisfying? Put the results into context and discuss.

(f) Give at least two reasons why the results may or may not be very good and which steps could be taken to influence them.

(g) Retrain the classifier with discretization resolutions of 50, 100, and 200. Evaluate the three new classifiers against the test set and against the training set itself. Explain the results.

Bonus: Challenge

(This exercise is not mandatory)

As in the previous exercise, optimize and tweak your implementation using everything you have learned. Areas of particular interest for optimization are the discretization scheme and resolution. Once satisfied with the performance, classify the examples in the test set from the previous bonus challenge and submit your classifications as a JSON file.

Hints:

- Do not over-optimize to a single test split. Consider the evaluation tips from last time, such as the use of cross-validation.
- If you are feeling lucky, you may try other probability estimation methods for different distributions from the Naive Bayes classifier family.
- Results should be submitted as a `pandas.DataFrame` written to a JSON file. It should contain the columns “page-id” and the corresponding “prediction.” To write the dataframe to file, use: `your_data_frame.to_json(filename, orient="records")`. Don’t forget to convert the numeric labels back to textual labels! The filename should be: “test_set_prediction_[group].json”, where [group] is **your** group number, for example “L01” for Leipzig or “W6-13” for Weimar.