<table>
<tr><th colspan="3">TOPIC PLAN</th></tr>
<tr><td>Partner organization</td><td colspan="2">Belgrade Metropolitan University</td></tr>
<tr><td>Topic</td><td colspan="2">Artificial intelligence</td></tr>
<tr><td>Lesson title</td><td colspan="2">Classification in machine learning</td></tr>
<tr><td>Learning objectives</td><td>Students can interpret basic concepts of classification.
Students can project Bayesian classifier.
Students are able to solve practical classification problems on artificial data and real-world datasets.</td><td rowspan="3">Methodology
☐Modeling
☐Collaborative learning
☐Project based learning
☐Problem based learning

Strategies/Activities
☐Graphic Organizer
☐Think/Pair/Share
☐Discussion questions

Assessment for learning
☐Observations
☐Conversations
☐Work sample
☐Conference
☐Check list
☐Diagnostics</td></tr>
<tr><td>Aim of the lecture / Description of the practical problem</td><td>The aim of this lecture is to learn basic concepts of classification, to gain knowledge for projecting Bayesian classifier and for solving practical classification problems on artificial data as well as on real-world datasets. In this lecture classification with Bayesian classifier is explained on two problems. In first problem there are artificial data where covariance matrices of classes are same and mathematical model for solving of this type of classification problem is presented. In second problem there are real data (famous Iris dataset) where covariance matrices of classes are different and mathematical model for solving of this type of classification problem is presented. Problems are practically solved in Python.</td></tr>
<tr><td>Previous knowledge assumed:</td><td>Basics of calculus.
Basics of linear algebra.
Basics of statistics.
Basic of Python programming.</td></tr>
</table>

<table>
<tr>
<td>Introduction / Theoretical basics</td>
<td>

RANDOM VECTORS AND THEIR PROPERTIES

In theory of statistical pattern recognition, measurements from one pattern or object are treated as random vectors. We will use term random variable in context of pattern recognition because it is clear that every time when we repeat the measurement, we will get some other value, which leads us to the idea of characterizing the measurement of physical quantity as some random variable that gives a new, different value in each realization. However, these values are not completely random, but subject to some laws. In order to formalize these laws, we will introduce two functions that will be joined to this random variable. First function is called distribution function, noted as $F_X(x)$, and second is called probability density function, noted as $f_X(x)$.

Distribution function is defined as:

$$F_X(x) = P_r\{X \leq x\}$$

and it represents probability that random variable X will take value that is less or equal to argument x. It has three main characteristics:

$$F_X(\infty) = 1 \Leftrightarrow P_r\{X \leq \infty\} = 1$$
$$F_X(-\infty) = 0 \Leftrightarrow P_r\{X \leq -\infty\} = 0$$
$$x_1 \leq x_2 \Rightarrow F_X(x_1) \leq F_X(x_2)$$

Probability density function is defined as:

$$f_X(x) = \frac{\partial F(x)}{\partial x}$$

Distribution function can be written as:

$$F_X(x) = \int_{-\infty}^{x} f_X(\tau) d\tau$$

Probability density function has two constraints:
</td>
<td>

Assessment as learning
☐Self-assessment
☐Peer-assessment
☐Presentation
☐Graphic Organizer
☐Homework

Assessment of learning
☐Test
☐Quiz
☐Presentation
☐Project
☐Published work
</td>
</tr>
</table>

$$F_X(\infty) = 1 \Rightarrow \int_{-\infty}^{\infty} f_X(x) = 1$$

$$(x_1 \leq x_2 \Rightarrow F_X(x_1) \leq F_X(x_2)) \Rightarrow (\forall x) f_X(x) \geq 0$$

Random vector *X* is completely described with distribution function or with probability density function. However, in many practical situations, these functions can't be determined or they are too complex. In those cases, we must choose some other parameters that are less informative, but much more convenient in numerical sense. First and most important parameter is mathematical expectation or mean value of random vector *X*:

$$M_X = E\{X\} = \int_{\sigma_X} x f_X(x) dx$$

where integration goes through whole space of random vector *X*. Conditional mathematical expectation of random vector *X* for class w_i is integral:

$$M_i = E\{X/w_i\} = \int x f(x/w_i) dx$$

Second very important parameter that characterize random vector *X* is covariance matrix:

$$\Sigma = E\left\{(X - M_X)(X - M_X)^T\right\}$$

$$\Sigma = E\left\{\begin{bmatrix} X_1 - m_1 \\ . \\ . \\ . \\ X_n - m_n \end{bmatrix} \begin{bmatrix} X_1 - m_1 & . & . & . & X_n - m_n \end{bmatrix}\right\} = \begin{bmatrix} c_{11} & . & . & . & c_{1n} \\ . & & & & \\ . & & & & \\ . & & & & \\ c_{n1} & . & . & . & c_{nn} \end{bmatrix}$$

Component c_{ij} of this matrix is:

$$c_{ij} = E\{(X_i - m_i)(X_j - m_j)\}\,;(i, j = 1, \ldots, n)$$

Thus, the diagonal elements of the covariance matrix form the variances of individual random variables in random vector, while non-diagonal elements represent covariances between random variables X_i and X_j. Although mathematical expectation and the covariance matrix are very important parameters which describe the distribution of a random vector, they are mostly unknown in practice and need to be estimated based on measured samples. This procedure is called sample estimation technique. This technique is most commonly used to estimate the mean and variance of a random variable. Namely, if it is necessary to estimate the mean value of the random variable Y whose realizations Y_i; *i*=1…, N are known, the simplest way is to determine arithmetic mean:

$$\hat{m_Y} = \frac{1}{N}\sum_{1}^{N} Y_i$$

Similarly, estimation of variance can be written as:

$$\hat{\sigma_Y^2} = \frac{1}{N}\sum_{1}^{N}(Y_i - \hat{m_Y})^2$$

HYPOTHESIS TESTING

The main goal of pattern recognition is to decide to which category observed sample belongs. Based on observation or measurement a measurement vector is formed. This vector serves as an input to the decision rule through which this vector joins one of the analyzed classes. ***Hypothesis testing*** is a whole family of methods solving of this type of a problem. These methods are very powerful, but they assume knowing of joined probability density functions from all classes and this information is often unknown in practice.

In pattern recognition theory, we deal with random vectors gained from different classes and each of them is characterized with its distribution function and probability density function. These functions are called conditional functions, and according to that, conditional probability density function for *i*-th class is noted as:

$$f(x/w_i) or f_i(x); i = 1, 2, \ldots, L$$

where w_i denotes class *i*, and *L* is overall number of classes. Unconditional probability density function of random vector X, which is often called mixed density function is given as:

$$f(x) = \sum_{i=1}^{L} P_i f_i(x)$$

where Pi denotes a priori probability of appearance of *i*-th class. A posteriori probability of class when random vector *X* is given is noted as $P(w_i/X)$ or $q_i(x)$ and can be calculated based on Bayesian theorem:

$$q_i(x) = \frac{P_i f_i(x)}{f(x)}$$

For our problem we will assume that measurement vector is random vector whose conditional probability density function depends from class that sample comes from. So far as these conditional probability density functions are known, then pattern recognition problem becomes statistical hypothesis testing problem. We will observe case where we have two classes w_1 and w_2 whose a priori probabilities P_1 and P_2 are known, as well as corresponding a posteriori probability density function $f_1(X) = f(X/w_1)$ and $f_2(X) = f(X/w_2)$.

Let assume that we have measurement vector X and our task is to determine to which of two classes this vector belongs. Simple decision rule can be based on conditional probabilities $q_1(X) = Pr(w_1/X)$ and $q_2(X) = Pr(w_2/X)$ as follows:

$$q_1(X) > q_2(X) \Rightarrow X \in w_1$$

$$q_1(X) < q_2(X) \Rightarrow X \in w_2$$

A posteriori probability $q_i(X)$ represents conditional probability that sample *X* comes from class w_i if its numerical value is known, i.e., realization. These probabilities can be calculated based on a priori probabilities of classes appearance Pi and a posteriori probability density function of measurement vectors $f_X = f_X(X/w_i)$, using Bayesian theorem:

$$q_i(X) = \frac{f_i(X)P_i}{f(X)} = \frac{f_i(X)P_i}{f_1(X)P_1 + f_2(X)P_2}$$

Because mixed (a priori) probability density function is positive and joint for both a posteriori probability, decision rule can be written as follows:

$$P_1 f_1(X) > P_2 f_2(X) \Rightarrow X \in w_1$$

$$P_1 f_1(X) < P_2 f_2(X) \Rightarrow X \in w_2$$

or we can write above equations as follows:

$$l(X) = \frac{f_1(X)}{f_2(X)} > \frac{P_2}{P_1} \Rightarrow X \in w_1$$

$$l(X) = \frac{f_1(X)}{f_2(X)} < \frac{P_2}{P_1} \Rightarrow X \in w_2$$

Expression *l*(*X*) is called likelihood ratio, and that is very important quantity in pattern recognition theory. Ratio P_2/P_1 is called ***threshold value*** in decision making. It is common practice to apply negative logarithm on likelihood ratio, and then decision rule has form:

$$h(X) = -ln(l(X)) = -ln(f_1(X)) + ln(f_2(X)) < ln(\frac{P_1}{P_2}) \Rightarrow X \in w_1$$

$$h(X) = -ln(l(X)) = -ln(f_1(X)) + ln(f_2(X)) > ln(\frac{P_1}{P_2}) \Rightarrow X \in w_2$$

A sign of inequality changed direction because of negative logarithm use. Expression ***h***(***X***) is called ***discrimination function***. Further on we will consider that $P_1 = P_2 = 0.5 \Rightarrow \ln\left(\frac{P_1}{P_2}\right) = 0$. Stated rules above are called ***Bayesian rule*** or ***minimal error decision test***. For analysis of stated rule, it is very important to determine probability of decision error. This classification rule can't ensure perfect classification (as well as other rules). When we say probability error, we consider probability of event that rule will bring wrong decision about measurement vector belonging to a class. Conditional probability of error for measurement vector, noted as *r*(*X*), is equal to smaller of probabilities $q_1(X)$ and $q_2(X)$, i.e.

$$r(X) = min[q_1(X), q_2(X)]$$

	Total error which is called Bayesian error, noted as ϵ can be calculated as follows: $$\epsilon = E\{r(X)\} = \int r(X)f(X)dX = \int min[q_1(X), q_2(X)]f(X)dX$$ $$= \int min[P_1 f_1(X), P_2 f_2(X)]dX = P_1 \int_{L_2} f_1(X)dX + P_2 \int_{L_1} f_2(X)dX$$ $$= P_1\epsilon_1 + P_2\epsilon_2$$ where $$\epsilon_1 = \int_{L_2} f_1(X)dX; \epsilon_2 = \int_{L_1} f_2(X)dX$$ Probability ε_1 is called ***Type I probability error*** and it represents probability that sample which comes from first class is wrongly classified. Similarly, probability ε_2 is called ***Type II probability error*** and it represents probability that sample which comes from second class is wrongly classified. In a total error relation first equality represents definition of this error, while other equality is applied Bayesian theorem. Integration area L_1 is the area from which the decision rule joins the measurement vector X to the class w_1 and analogously, integration area L_2 corresponds to those vectors X which the decision rule classifies into a class w_2. Consequently, these areas are often called w_1-area and w_2-area, respectively. For measurement vectors from L_1 area holds the relation $P_1 f_1(X) > P_2 f_2(X)$ and according to that conditional probability error is $r(X) = P_2 f_2(X)/f(X)$. Analogously, for vectors from L_2 area holds the relation $r(X) = P_1 f_1(X)/f(X)$. Based on that, we can say that Bayesian probability error consists of two terms. One of them refers to wrongly classified vectors from class w_1, while other refers to wrongly classified vectors from the class w_2.	

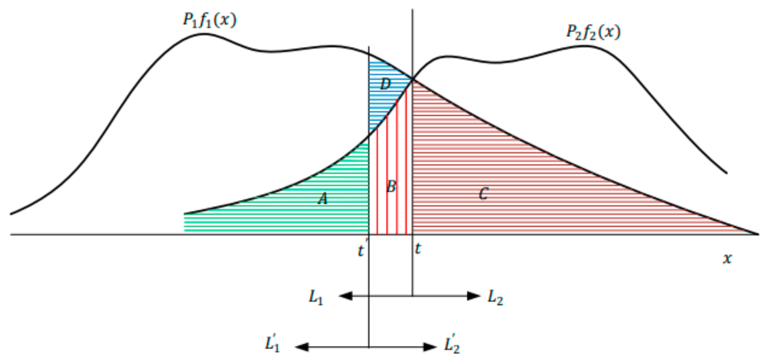

Figure 1. Illustration of one-dimensional random variables classification

On the Figure 2 Bayesian decision rule for one dimensional measurement vectors is illustrated: Decision boundary is set to *x*=*t*, and that is the point where $P_1f_1(X) = P_2f_2(X)$, and areas *x*<*t* and *x*>*t* are marked as L_1 and L_2 respectively. In this way, probability errors become $P_1\varepsilon_1 = B + C$ and $P_2\varepsilon_2 = A$. Total Bayesian error becomes *ϵ*=*A*+*B*+*C*, where A, B and C are marked areas, for example:

$$B = \int_t^{t'} P_1 f_1(X) dX$$

This decision rule generates minimal possible probability decision error. Calculating of Bayesian probability error is very complex problem, because this probability is calculated with probability density function integration (which is function of more variables) trough very complex areas. Because of that it is much easier to consider the problem in discrimination function domain, and integration can be performed through this function. Random variable is scalar, so the integration problem can be performed in one dimensional space. We can write this as:

$$\epsilon_1 = \int_{-\infty}^{ln(P_1/P_2)} f_h(h/w_1) dh$$

$$\epsilon_2 = \int_{ln(P_1/P_2)}^{\infty} f_h(h/w_2) dh$$

	Where $f_h(h/w_i)$ is a posteriori probability density function of discrimination function h for samples that come from class w_i. For our problem for artificially generated data random vector *X* is normally distributed. If random vector *X* is normally distributed, its probability density function can be written as: $N_X(M,\Sigma) = \frac{1}{(2\pi)^{n/2}\|\Sigma\|^{1/2}} e^{-\frac{1}{2}d^2(X)}$ $= \frac{1}{(2pi)^{n/2}\|\Sigma\|^{1/2}} e^{-\frac{1}{2}(X-M)^T\Sigma^{-1}(X-M)}$ Where $N_X(M,\Sigma)$ is shorted notation for normal distribution with mathematical expectation *M* and covariance matrix Σ. Function $d^2(X)$ is called statistical distance (or d^2 curve) of vector *X* from mathematical expectation vector M.	
Action	After introducing theoretical concepts, discussion with students will be performed for solving two posted problems. First problem is binary classification of artificially generated data. Data are normally distributed and data from first and second class have same covariance matrix. Second problem is classification of Iris dataset where data from classes have different covariance matrix. **SOLVING OF FIRST CLASSIFICATION PROBLEM** For our problem of artificially generated data, we have a posteriori probability density functions $f_i(X)$, where *i*=1,2. These functions are normal, with mathematical expectation vector M_i and covariance matrix Σ_i. Bayesian minimal error decision rule can be written in form: $h(x) = -ln(l(X)) = -ln(f_1(X)) + ln(f_2(X)) < ln(\frac{P_1}{P_2}) \Rightarrow X \in w_1$ Now we can replace $f_1(X)$ and $f_2(X)$ in upper equation and rewrite expression for *h*(*x*):	

$$h(x) = -ln(\frac{1}{(2pi)^{n/2}|\Sigma_1|^{1/2}}e^{-\frac{1}{2}(X-M_1)^T\Sigma_1^{-1}(X-M_1)}) +$$

$$+ ln(\frac{e^{-\frac{1}{2}(X-M_2)^T\Sigma_2^{-1}(X-M_2)}}{(2\pi)^{n/2}|\Sigma_2|^{1/2}})$$

We can write expression for discrimination function in final form:

$$h(x) = \frac{1}{2}(X - M_1)^T\Sigma_1^{-1}(X - M_1) -$$

$$-\frac{1}{2}(X - M_2)^T\Sigma_2^{-1}(X - M_2) + \frac{1}{2}ln(\frac{|\Sigma_1|}{|\Sigma_2|})$$

Last equation shows that decision boundary is quadratic function of X. If two classes have same covariance matrix, i.e. if $\Sigma_1 = \Sigma_2 = \Sigma$, decision boundary becomes linear function of *X* and can be written in next form:

$$h(x) = (M_2 - M_1)^T\Sigma^{-1}X + \frac{1}{2}(M_1^T\Sigma^{-1}M_1 - M_2^T\Sigma^{-1}M_2)$$

In our classification problem of artificially generated data classes have same covariance matrix and a priori probabilities of classes appearance are same. So, we can write expression for classification in the first class:

$$h(x) = (M_2 - M_1)^T\Sigma^{-1}X + \frac{1}{2}(M_1^T\Sigma^{-1}M_1 - M_2^T\Sigma^{-1}M_2) < 0 \Rightarrow X \in w_1$$

Similarly, we can write expression for classification in second class:

$$h(x) = (M_2 - M_1)^T\Sigma^{-1}X + \frac{1}{2}(M_1^T\Sigma^{-1}M_1 - M_2^T\Sigma^{-1}M_2) > 0 \Rightarrow X \in w_2$$

As we can see our classifier is line (we have linear function of *X*). I we want to draw our line, we can do that as follows. Decision of belonging to one of two classes is based on value of decision boundary (is it bigger or smaller than zero), so if we want to draw boundary itself, we must equal expression for *h*(*x*) with zero:

$$h(x) = (M_2 - M_1)^T\Sigma^{-1}X + \frac{1}{2}(M_1^T\Sigma^{-1}M_1 - M_2^T\Sigma^{-1}M_2) = 0$$

In order to draw our line, we must write expression for X_2 of X_1, because we are in X_1X_2 space. In other words, we have linear relationship and we want to express *y* of *x*, and in our case that is X_2 of X_1.

After explanation for discrimination function drawing, we implemented the solution in Python programming language and solved first classification problem. Gained results are on the figure below.

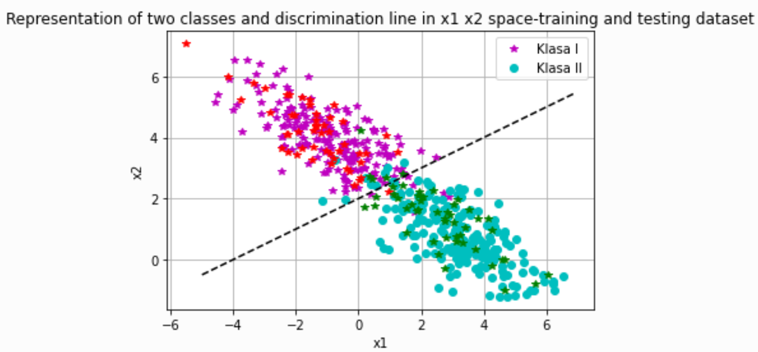

Figure 2. Bayesian classifier for binary classification of artificially generated data

After calculating of Type I error and Type II error on testing dataset we get 1 sample that is wrongly classified from class I, and 5 samples that are wrongly classified from class II. Classifier accuracy is 94%.

SOLVING OF SECOND CLASSIFICATION PROBLEM

In our second classification problem we have classes with different mean vectors and different covariance matrix. Here, we will have also have binary classification problem, but now we must fine two discrimination functions- first one between classes Iris Setosa and Iris Versicolor and second between classes Iris versicolor and Iris Virginica. When covariance matrix are not same, then discrimination function is not line, it is parable. Our discrimination function can be written as follows:

$$h(x) = \frac{1}{2}(X-M_1)^T\Sigma_1^{-1}(X-M_1) - \frac{1}{2}(X-M_2)^T\Sigma_2^{-1}(X-M_2) + \frac{1}{2}ln(\frac{|\Sigma_1|}{|\Sigma_2|})$$

Again, we implemented the solution in Python programming language and solved second classification problem. Gained results are on the figure below.

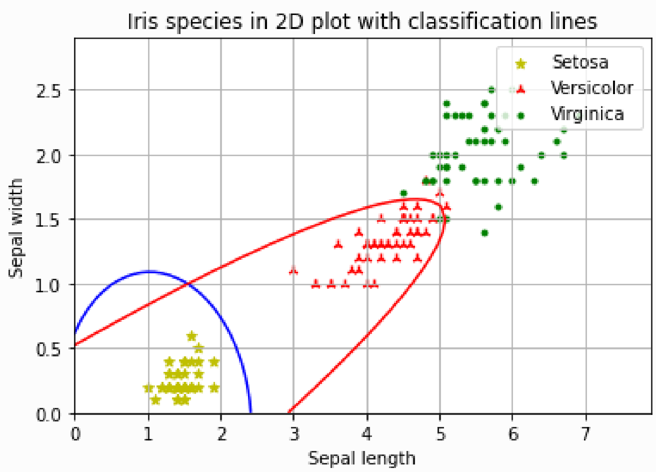

Figure 3: Classification of Iris dataset with Bayesian classifier

```
Classification of Setosa and Versicolor - Classification accuracy: 100.0%
Number of bad classified samples from class Setosa: 0
Number of bad classified samples from class Versicolor: 0
Classification of Versicolor and Virginica - Classification accuracy: 96.0%
Number of bad classified samples from class Versicolor: 3
Number of bad classified samples from class Virginica: 1
```

Figure 4: Classifier accuracy

At the end of lesson, we will discuss with students gain results and talk about conclusions after solving the classification problems.

Materials / equipment / digital tools / software

The materials for learning can be found as a part of references at the end of this topic plan;
Equipment: classroom, board, chalk;
Digital tools: laptop, projector;
Software: Python programming language.

Consolidation	• The teacher's discussion with the students through different questions; • Individually solving of simple tasks by the students under the supervision of the teacher; • Given of examples by the teacher for introducing a new concepts and creative discussion with the students; • Given homework by the teacher with a time limit until the next class.	
Reflections and next steps		
Activities that worked		**Parts to be revisited**
Introducing to classificiation basics. Solving problems in Python programming language.		Classification theoretical background.
References		

[1] Emilija Kisić, CS374 – Artificial Intelligence, Authorized Lectures on Metropolitan University Belgrade eLearning platform – LAMS, 2022.
[2] Artificial Intelligence: A Modern Approach. 4th Edition, S. Russell and P. Norvig. Prentice Hall, 2021.
[3] Artificial Intelligence: A New Synthesis, Nils Nilsson, Morgan Kaufmann, 1998