POLYTECHNIC OF MEĐIMURJE IN ČAKOVEC

COURSE SYLLABUS

ACADEMIC YEAR: 2020/2021

1. GENERAL COURSE INFORMATION

1.1 Course name	**Pattern recognition**			
1.2 Study program/s	**Undergraduate professional study of Computer Science**			
1.3 Course status (O,E)	**Electoral**	**1.6 Mode of instruction (number of hours)**	**Lectures**	30
1.4 Course code			**Exercises**	30
1.5 Course abbreviation	RU		**Seminars**	
1.6 Semester	VI		**E-learning**	
1.7 ECTS	4	**1.7 Place and time of instruction**	Premises of the Polytechnic of Međimurje in Čakovec, according to the schedule published on the website	

2. TEACHING STAFF

2.1 Course leader/s-title	Željko Knok/ Master of Science	**contact**	zknok@mev.hr
		contact	
2.2 Assistant/s- title		**contact**	
		contact	
2.3 Instruction held by- title		**contact**	

3. COURSE DESCRIPTION

3.1 Course goals	After completing the course, the student will be able to use Python to recognize patterns. Knowledge in the field of artificial intelligence and machine learning is acquired
3.2 Prerequisites	Knowledge of the Python programming language is desirable for taking the course, but it is not necessary
3.3 Course outcomes	After successfully completing the course, students will be able to: O1 - Prepare tools for machine learning and pattern recognition O2 - Identify data sources required for machine learning O3 - Solve simple linear regression and classification tasks, clustering using Python programming language O4 - Solve simple tasks using a simple neural network and decision tree
3.4 Course content	The course presents contents related to the concept, possibilities and role of the database. Special attention is given to data search using SQL language, modeling and database maintenance. In the practical part, open source tools are used.

3.5 Types of coursework										
	x	Lectures	x	Exercises		Blended e-learning	x	Individual activities		Laboratory
		Seminars and workshops		Distant learning		Field classes	x	Multimedia and network		Mentorship
		Other								

3.6 Language of instruction	Croatian /English					
3.7 Monitoring students' work (enter the	1,00	Class attendance		Seminars		Essay
		Class activity		Project		Report/paper

number of ECTS credits for each activity so that the total number of ECTS credits is equal to the total ECTS value of the course, 1 ECTS = 30 hours)

2,00	Midterm exams	1,00	Practical task		Continuous knowledge check
	Written exam		Experimental work		
	Oral exam		Research		

3.8 Assessment and evaluation of students' work during classes and at the final exam

Activity specification	Percent %	Points
Assessment during instruction		
Attendance	5%	5
Class activity	5%	5
Seminar/ project/ essay	30%	30
Midterm exam 1	30%	30
Midterm exam 2	30%	30
Exam assessment for the students who failed to fullfil all the obligatory requirements during the semester		
Written exam	*60%*	*60*
Total:	**100%**	**100**

3.9 Assessment criteria – analysis per learning outcomes

Ways of evaluating learning outcomes						
	Attendance	**Activity**	**Mid-term exam 1**	**Mid-term exam 2**	**Practical work**	**Total**
Outcome 1			15		10	25
Outcome 2			15		5	20
Outcome 3				15	5	20
Outcome 4				15	10	25
Outcome not-related						10
Total	5	5	30	30	30	100

Grading of outcomes (in order to pass the mid-term exam/exam the student must achieve at least 50% points for each learning outcome)

Points	Grade
89 – 100	excellent (5)
76 – 88	very good (4)
63 – 75	good (3)
50 – 62	pass (2)
0 – 49	fail (1)

3.10 Specific features related with taking the course

If a student collects 50% of the points of each outcome, he / she directly takes the exam, provided that he / she has done practical work (exercises). A student cannot access the exam period if he / she has not achieved min. 60% correct answers. Practical work-exercises are made according to the instructions published on the Merlin system and are submitted by posting on the Merlin. Checking the completed exercises is done in the exercise classes after prior preparation with the teacher. During the semester, the student is required to perform five exercises independently. Practical work (completed exercises) is taught until the last week of lectures. During the exam, it is possible to orally check the knowledge from practical work (exercises).

If a student does not achieve a sufficient number of points on the midterm exam, he / she cannot take the next midterm exam.

Once achieved points in intermediate exams for each learning outcome are no longer deleted unless the student decides to correct the result for each learning outcome, whereby the points won until then are deleted and newly achieved points for that learning outcome are entered.

	The final grade is obtained on the exam period and is the sum of points earned during classes. Students who did not take the colloquium access the written part of the exam where all learning outcomes are checked, and are required to have completed exercises before taking the exam.	
3.11 Students obligations	Full-time students are required to attend at least 70% of the total number of hours of lectures and exercises in order to exercise the right to take the exam. Part-time students are required to attend at least 30% of the total number of hours of lectures and exercises in order to exercise the right to take the exam. If the student has not fulfilled all the obligations set by the course, he is obliged to attend the lectures again and meet the conditions for taking the exam. Attendance can be offset by online tuition, organised webinars and added assignments given by teachers. One lesson lasts 45 minutes, and several hours form a teaching unit. Absence from one teaching unit is counted as one absence. Delays and apologies are recorded separately. In that case, if the student missed more than 50% of classes, and has a justifiable reason/apology, the request should be submitted to the Department Council, which then decides on the justification of student absences with the obligatory opinion of the course leader.	
3.12 Written assignments		
3.13 Required reading	1.	Zsolt Nagy, Artificial Intelligence and Machine Learning Fundamentals 1st Edition, 2018.
	2.	
3.14 Additional reading	1.	https://scikit-learn.org/stable/auto_examples/index.html#examples
	2.	https://jakevdp.github.io/PythonDataScienceHandbook/index.html
4 ADDITIONAL COURSE INFORMATION		
4.1 Quality control	The quality of the program, teaching process, teaching skills and level of mastery of the material will be established by conducting a written evaluation based on questionnaires, and in other standardised ways and in accordance with the by-laws of the Polytechnic of Međimurje in Čakovec.	
4.2 Contact the teacher	Students can contact the teacher during the office hours and during classes, while for short questions and explanations they can contact him/her any day during working hours by coming in person or by landline. It is also possible to ask questions by e-mail, which will be answered in 48 hours at the latest. It is desirable for students to come as often as possible for any possible questions during the teacher's office hours.	
4.3 Information about the course	It is the obligation of each student to be regularly informed about the course. All notifications about the classes or possible postponement of classes will be posted on the bulletin board and on the website of the Polytechnic at least 24 hours in advance.	

4.4 Course contribution to the study program	Apply the acquired learning skills, basic knowledge of the profession and problem solving necessary for continuing studies at a higher level. Apply relevant mathematical and statistical methods in software engineering.

5. ANALYSIS OF COURSE TOPICS (the number of hours is equal to the number of lectures and exercises of the course)

LECTURES

Week	Topic and description	Method • Direct teaching (lecture, instruction, pp presentation) • Discovery learning (individual, lead, discussion) • Group learning • Case study • Field classes…	Learning outcomes	Course outcome
1.	Introduction to the course content, the concept of analysis and data processing (Data science)	Direct teaching (lecture, instruction, pp presentation)	Distinguish concepts in the field of Data science	O1
2.	Introduction to Python programming languages, Numpy data structure	Direct teaching (lecture, instruction, pp presentation)	Uses commands within the Numpy module	O2
3.	Advanced Numpy Methods. Advanced indexing. The notion of random numbers	Direct teaching (lecture, instruction, pp presentation)	Use advanced methods and commands	O2
4.	Introduction to data visualization modules	Direct teaching (lecture, instruction, pp presentation)	Use different modules to visualize the data	O2
5.	Basic algorithms for data processing. Linear classification and linear regression	Direct teaching (lecture, instruction, pp presentation)	Use LK and LR	O3
6.	Mid-term exam 1			O1-O2
7.	Data structures in the form of tables. Pandas Library	Direct teaching (lecture, instruction, pp presentation)	Use the Pandas library	O3
8.	Advanced data processing in the Pandas library. Filtration and aggregation	Direct teaching (lecture, instruction, pp presentation)	Use Pandas for filtering and aggregation	O3
9.	Advanced regression and classification methods	Direct teaching (lecture, instruction, pp presentation)	Use advanced regression and classification methods	O3
10.	Clustering process and linking to classification	Direct teaching (lecture, instruction, pp presentation)	Apply clustering with associated classification	O3
11.	Fundamentals and concept of neurocomputing	Direct teaching (lecture, instruction, pp presentation)	Explain the concept of neurocomputing	O4

12.	Neural network architecture and learning algorithms	Direct teaching (lecture, instruction, pp presentation)	Explain the types and architectures of neural networks	O4
13.	Decision trees (entropy. Gini Impurity)	Direct teaching (lecture, instruction, pp presentation)	Apply decision trees	O4
14.	Examples from practice	Direct teaching (lecture, instruction, pp presentation)	Recognize the application of machine learning in practice	O4
15.	Mid-term exam 2			O3-O4
EXERCISES/ SEMINARS				
Week	**Topic and description**	**Method** • Direct teaching (lecture, instruction, pp presentation) • Discovery learning (individual, lead, discussion) • Group learning • Case study • Field classes…	**Learning outcomes**	**Course outcome**
1.	Installation of the Anaconda programming environment and associated library modules	Direct teaching (lecture, instruction, pp presentation)	Install the necessary tools to work	O1
2.	Simple examples of using the Numpy module	Guided task, code examples	Apply commands within the Numpy module	O2
3.	Examples with matrices and random numbers	Guided task, code examples	Apply commands to work with matrices and random numbers	O2
4.	Exercise 1	Independent preparation of the exercise	Create a task using the Numpy module and matrices	O2
5.	Example of using matplotlib module for data visualization	Guided task, code examples	Apply matplotlib module in tasks	O2
6.	Example with LDA analysis	Guided task, code examples	Use LDA	O3
7.	Exercise 2	Independent preparation of the exercise	Create an example with the matplotlib module	O3
8.	Example of filtering data from a set of input data	Guided task, code examples	Apply data filtering functions	O3
9.	Exercise 3	Independent preparation of the exercise	Create an example with data filtering	O3
10.	Data example - Iris	Guided task, code examples	Uses data classification functions	O3
11.	Exercise 4	Independent preparation of the exercise	Apply the classification	O3

			function within Python	
12.	Examples for two types of clustering	Guided task, code examples	Apply the clustering procedure	O3
13.	A simple example of interpreting the operation of neural computer networks	Guided task, code examples	Explain the operation of neural networks	O4
14.	An example of classifying a car using a decision tree	Guided task, code examples	Create a classifier using the decision tree	O4
15.	Exercise 5	Independent preparation of the exercise	Create a classifier using the Random forest procedure	O4