

ANALIZA I PRZETWARZANIE DANYCH

Efekty uczenia się i treści programowe zajęć:

Nazwa zajęć: Analiza danych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi dokonać sortowania, filtrowania, podsumowywania oraz grupowania danych w postaci tabelarycznej w programie R.
- Umie wizualizować dane w programie R.
- Potrafi generować dane w programie R i wykorzystać je do przeprowadzenia symulacji stochastycznej.
- Potrafi dokonać analizy błędu pomiarowego. Rozumie ideę propagacji błędu oraz rozróżnia typy błędów pomiarowych.
- Potrafi konstruować liniowe oraz nieliniowe modele zależności zjawisk. Zna podstawowe metody wnioskowania dotyczące analizy regresji.
- Zna podstawowe metody analizy przeżycia.
- Potrafi dokonać analizy szeregów czasowych, ich modelowania oraz generowania. Zna podstawowe testy statystyczne dotyczące szeregów czasowych.
- Umie dokonać redukcji wymiaru danych. Potrafi ocenić jakość przeprowadzonej redukcji.
- Potrafi przeprowadzić klasyfikację korzystając z próby uczącej oraz podział danych na rozłączne skupienia. Potrafi ocenić jakość przeprowadzonych działań.

Treści programowe dla zajęć:

- Podstawy programu R oraz jego rozszerzenia w pakiecie dplyr
- Graficzne możliwości pakietu R ze szczególnym uwzględnieniem pakietu ggplot2
- Generatory liczb pseudolosowych w programie R, implementacja symulacji stochastycznej
- Podstawy rachunku błędów, obserwacje odstające, Kryterium Chauveneta, test Dixona, test Grubbsa
- Regresja liniowa i wielokrotna, regresja krokowa, regresja odporna, regresja składowych głównych, regresja częściowych najmniejszych kwadratów, regresja grzbietowa, regresja nieliniowa, regresja logistyczna
- Funkcja przeżycia, funkcja hazardu, estymator Kaplana-Meiera, model Coksa.
- Szeregi czasowe i ich analiza: modelowanie, wygładzanie, stacjonarność oraz generowanie szeregów czasowych, autokorelacja, autoregresja, rzędy procesu
- Techniki redukcji wymiarowości: analiza składowych głównych, analiza czynnikowa, skalowanie wielowymiarowe, analiza korespondencji.
- Podstawowe i zaawansowane metody klasyfikacji, np. LDA, QDA, kNN, NB. Hierarchiczna i niehierarchiczna analiza skupień.

Nazwa zajęć: Analiza danych przestrzennych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna podstawowe oraz zaawansowane struktury danych tabelarycznych oraz przestrzennych jak również podstawowy zakres wykonywanych na nich operacji.
- Zna podstawowe konstrukcje programistyczne oraz pojęcia składni i semantyki języków programowania wykorzystywane w procesie eksploracji danych przestrzennych w języku R lub Python. Zna zaawansowane metody informatyczne rozwiązywania problemów, po określeniu celu realizacji projektu zaliczeniowego potrafi dobrać adekwatne narzędzia do jego realizacji.
- Zna podstawowe źródła danych przestrzennych i umie je pozyskać.
- Umie napisać raport zawierający odpowiednie wizualizacje oraz interpretację uzyskanych wyników.

Treści programowe dla zajęć:

- Wprowadzenie do danych przestrzennych, wektorowych oraz rastrowych; eksploracja danych przestrzennych oraz tabelarycznych
- Złączenia przestrzenne, agregacje danych przestrzennych, selekcja przestrzenna, transformacje geometrii,
- Źródła danych przestrzennych administracyjnych, bazy danych obiektów topograficznych, pozyskiwanie danych satelitarnych, cyfrowych modeli terenu oraz otwartych danych z Open Street Map
- Konsultacje w ramach realizacji projektu zaliczeniowego

Nazwa zajęć: Algorytmy i struktury danych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna i stosuje podstawowe konstrukcje algorytmiczne, zapisuje je w pseudokodzie i wybranym języku programowania.
- Wykorzystuje procedury i funkcje do formułowania algorytmów. Ma wiedzę w zakresie znaczenia i wykorzystania pojęcia rekurencji.
- Zna i stosuje proste i złożone struktury danych, w tym struktury dynamiczne.
- Zna podstawowe techniki projektowania algorytmów i stosuje wiedzę matematyczną do formułowania i rozwiązywania prostych zadań algorytmicznych.
- Konstruuje i implementuje w wybranym języku programowania algorytmy dla średnio zaawansowanego problemu algorytmicznego.
- Ocenia poprawność i złożoność czasową i pamięciową algorytmów, potrafi krytycznie ocenić skonstruowany algorytm.
- Ma świadomość ważności algorytmiki w informatyce i rozumie potrzebę dalszego kształcenia algorytmicznego.

Treści programowe dla zajęć:

- Język algorytmiczny
- Pojęcie zmiennej, instrukcja przypisania, instrukcje warunkowe, iteracje, operatory specjalne
- Pojęcie struktury tablicowej
- Przykłady i implementacje prostych problemów algorytmicznych na tablicach 1 i 2-wymiarowych, wyszukiwanie liniowe i binarne
- Pojęcie procedury
- Deklaracja, parametry formalne, wywołanie, przykłady prostych procedur i funkcji
- Rekurencja
- Pojęcie rekurencji, przykłady procedur rekurencyjnych, programowanie dynamiczne
- Algorytmy sortowania
- Sortowanie przez wstawianie, bąbelkowe, przez scalanie, szybkie, przez zliczanie (elementy różne, przypadek ogólny)
- Analiza algorytmów
- Notacja asymptotyczna, złożoność pamięciowa i czasowa algorytmów, złożoność optymistyczna, pesymistyczna i średnia, twierdzenie o rekurencji uniwersalnej, klasy złożoności, złożoność poznanych algorytmów
- Stosy, kolejki, listy
- Pojęcie zbioru dynamicznego. Tablicowa implementacja stosu i operacje na stosie. Tablicowa implementacja kolejki i operacje kolejkowe. Lista dwukierunkowa z dowiązaniem. Operacje na listach. Listy z wartownikiem.
- Pojęcia teorii grafów
- Graf prosty, drzewa i ich podstawowe własności, drzewa ukorzenione
- Kopce
- Podstawowe operacje na kopcach binarnych, sortowanie przez kopcowanie
- Drzewa wyszukiwań binarnych
- Podstawowe własności, operacje słownikowe przechodzenie drzewa BST
- Metoda zachłanna
- Problem minimalnego drzewa rozpinającego, problem najkrótszych połączeń, kody Huffmana
- Metoda z nawrotami
- Problem n królowych, przykłady prostych problemów optymalizacyjnych
- Stabilność numeryczna algorytmów
- Arytmetyka stała- i zmiennopozycyjna, pojęcie błędu bezwzględnego i względnego, przykłady algorytmów numerycznie niestabilnych

Nazwa zajęć: Chmura obliczeniowa

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna podstawowe zagadnienia związane z przetwarzaniem w chmurze i technologiami chmurowymi
- Zna zasady działania oraz architekturę różnych usług chmurowych
- Zna teorię konstrukcji rozwiązań chmurowych i dopasowania ich do specyfiki działalności przedsiębiorstwa
- Potrafi podać, ocenić i przeanalizować przykłady istniejących rozwiązań chmurowych
- Umie wykorzystać metody poznane na zajęciach w celu pracy z podstawowymi usługami chmurowymi
- Potrafi zastosować poznane narzędzia do rozwiązywania rzeczywistych problemów wymagających technologii chmurowych

Treści programowe dla zajęć:

- Przetwarzanie w chmurze – podstawowe informacje, terminologia, modele chmurowe
- Rodzaje chmur, trendy w technologiach chmurowych, argumenty za i przeciw rozwiązaniom chmurowym
- Modele usług SaaS, PaaS, IaaS i inne
- Amazon Web Services – podstawowe informacje, idea, rozwiązania, cele, usługi
- Amazon Web Services – rozpoczęcie pracy, założenie konta, EC2 i S3 z wykorzystaniem strony internetowej AWS
- EC2 i S3 z wykorzystaniem klienta AWS
- EC2 i S3 z wykorzystaniem języka Python
- Praca w grupach: analiza Case Study i przedstawienie go w formie prezentacji

Nazwa zajęć: Statystyka nieparametryczna

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna i rozumie skale pomiarowe oraz elementy statystyki opisowej.
- Zna i umie zastosować estymację jądrową.
- Zna i rozumie pojęcie dystrybuanty empirycznej oraz umie ją zastosować do testowania wybranych hipotez statystycznych.
- Rozumie pojęcie zależności zmiennych losowych i umie ją badać za pomocą odpowiednich współczynników oraz sprawdzić ich statystyczną istotność za pomocą testów statystycznych.
- Zna i umie wykorzystać metody bazujące na rangach i znakach.
- Potrafi wykorzystać test serii w różnych problemach.
- Potrafi wykorzystać testy chi-kwadrat w różnych zagadnieniach.
- Zna, rozumie i umie wykorzystać testy statystyczne w przypadku cech o rozkładzie zero-jedynkowym.
- Zna i potrafi wykorzystać metody repróbkiwania.
- Rozumie i zna testy bazujące na randomizacji i permutacji danych.

Treści programowe dla zajęć:

- Skale pomiarowe, charakterystyki szeregu rozdzielczego, statystyka opisowa, wykresy statystyczne
- Estymacja jądrowa, różne estymatory jądrowe.
- Testy statystyczne oparte na dystrybuancie empirycznej (test Kołmogorowa i Kołmogorowa-Smirnowa), dystrybuanta empiryczna.
- Współczynniki mierzące zależność (Spearmana, Kendalla), testy istotności dla tych współczynników, współczynniki zgodności opinii i konsystencji
- Rangowanie. testy oparte na rangach i znakach (mediana, Fischera, Wilcoxon, U Manna-Whitneya, rangowanych znaków Wilcoxon), testy rangowe dla wielu grup (Kruskala-Wallis, Friedmana)
- Testy serii (losowości próby, zgodności rozkładów, Walda-Wolfowitza, losowości reszt w regresji)
- Testy chi-kwadrat (zgodności, niezależności, jednorodności, dla tablic kontyngencji)
- Testy dla proporcji (dla prób niezależnych, McNemara, chi-kwadrat)
- Metody: bootstrap, jackknife, Monte Carlo
- Testy randomizacyjne i permutacyjne

Nazwa zajęć: Etyka danych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna etyczne podstawy gromadzenia, przetwarzania oraz modelowania danych jak również ich wpływ na sferę społeczno-gospodarczą. Wie, czym są dane wrażliwe, jakie dane przestrzenne mogą zostać zaklasyfikowane jako dane wrażliwe oraz jak należy się z nimi obchodzić.
- Wie czym są błędy poznawcze i jaki mogą mieć wpływ na cały proces gromadzenia, przetwarzania i modelowania zjawisk.
- Wie w jaki sposób wizualizacja danych może szerzyć dezinformację oraz jakie metody wizualizacji są poprawne w odniesieniu do wybranych zjawisk.

Treści programowe dla zajęć:

- Wprowadzenie do danych etyki danych – problematyka pozyskiwania danych, RODO, prywatności danych oraz wpływu przetwarzania danych na środowisko
- Znaczenie błędów poznawczych w kontekście pozyskiwania, przetwarzania i interpretacji danych
- Etyczne formy prezentacji danych i wykorzystywania w procesie podejmowania decyzji

Nazwa zajęć: Gromadzenie i eksploracja danych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna wielowymiarowy model danych. Umie zaprojektować hurtownię wykorzystującą model OLAP. Potrafi formułować zapytania w językach SQL MDX oraz DAX.
- Potrafi integrować dane pochodzące z różnych źródeł, transformować je do wspólnej spójnej postaci oraz wypełniać nimi hurtownie danych.
- Zna wybrane modele i algorytmy zgłębiania danych oraz potrafi je wykorzystać do tworzenia systemów analitycznych. Potrafi wykorzystać język DMX do wypełniania, trenowania i predykcji w systemach analitycznych.
- Umie tworzyć raporty z wykorzystaniem standardowych narzędzi (Access, Excel, narzędzia tabel przestawnych) czerpiąc dane zarówno ze źródeł relacyjnych, jak i analitycznych.
- Zna wybrane metody reprezentacji danych w formacie XML, ich transformację z i do danych w modelu relacyjnym. Umie przetwarzać dokumenty XML.

Treści programowe dla zajęć:

- Systemy transakcyjne (OLTP) a analityczne (OLAP); porównanie; obszary zastosowań
- Hurtownie danych. Model danych wielowymiarowych – kostka, miary, wymiary, hierarchie, komórki; tabele faktów i tabele wymiarów; schemat gwiazdy i płatka śniegu; sposoby składowania: ROLAP, MOLAP, HOLA
- Tabele przestawne, PowerPivot, elementy języka DAX (Data Analysis eXpressions). Wykorzystanie Power BI
- Tworzenie hurtowni danych
- Język zapytań MDX (MultiDimensional eXpressions)
- Integracja i transformacja danych
- Eksploracja danych - pojęcie eksploracji danych; eksploracja danych w systemach odkrywania wiedzy; systemy uczące się
- Wybrane algorytmy eksploracji danych i ich implementacja w SSAS - drzewa decyzyjne; reguły asocjacyjne; wzorce sekwencji; analiza skupień
- Język DMX (Data Mining Extensions) w systemie zgłębiania danych.
- Tworzenie raportów z baz danych; raporty lokalne; wsparcie ze strony serwerów bazodanowych; serwery raportów
- Wykorzystanie serwerów OLE do tworzenia analiz i raportów.
- XML w bazach danych i systemach integracji danych. Reprezentacja danych w standardzie XML; modele DOM i SAX; opis schematu dokumentu XML –DTD i XML Schema.

Nazwa zajęć: Modelowanie informacji w ekonomicznych modelach decyzyjnych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna podstawowe zagadnienia związane z tworzeniem i zastosowaniem modeli decyzyjnych, zna podstawy teorii decyzji
- Zna budowę oraz zasadę działania systemów eksperckich
- Zna teorię konstrukcji sterownika rozmytego, bazy reguł, automatycznego dowodzenia twierdzeń
- Potrafi podać, ocenić i przeanalizować przykłady istniejących i wykorzystywanych modeli decyzyjnych wraz z zasadami ich działania
- Umie wykorzystać metody teorii decyzji, teorii gier i badan operacyjnych do rozwiązywania prostych problemów decyzyjnych.
- Potrafi zastosować narzędzia zbiorów rozmytych oraz istniejące modele decyzyjne do rozwiązywania prostych problemów decyzyjnych.

Treści programowe dla zajęć:

- Teoria decyzji – podstawowe pojęcia, podstawowe modele decyzyjne, rys historyczny modeli decyzyjnych
- Proste modele decyzyjne – metoda sympleks, zamknięte zadania transportowe, optimum Pareto
- Badanie operacyjne – podstawowe metody algorytmicznego rozwiązywania prostych problemów decyzyjnych w ekonomii
- Proste modele decyzyjne – otwarte zadania transportowe, zagadnienie najkrótszej drogi, zastosowanie dodatku Solver do rozwiązywania problemów metodą sympleks
- Teoria gier – podstawowe pojęcia, rodzaje gier, przykłady gier, równowaga Nasha
- Teoria gier – zapoznanie z multimedialnymi materiałami doszkalającymi (interaktywne gry typu dylemat więźnia, symulacja gry kooperacyjnej), postać strategiczna gry, poszukiwanie równowagi Nasha, poszukiwanie strategii dominujących i słabo dominujących
- Reguły decyzyjne – rodzaje reguł, tworzenie reguł, wnioskowanie automatyczne, automatyczne dowodzenie twierdzeń
- Reguły wnioskowania, wnioskowanie automatyczne – budowa prostej bazy reguł, automatyczne dowodzenie twierdzeń przy pomocy tablic Betha

- Systemy eksperckie – podstawowe pojęcia, budowa, podział, zastosowania, narzędzia tworzenia, zasady działania, przykłady systemów eksperckich
- Model Markowitza – podstawowe pojęcia, zasada działania, opis ryzyka, portfele efektywne
- Model Markowitza – podstawy zarządzania portfelem, szukanie portfeli efektywnych, ocena ryzyka
- Model rozmyty – podstawowe pojęcia związane ze zbiorami rozmytymi, rozmyta arytmetyka portfela, rozmytość a ryzyko, przykład portfela rozmytego
- Zbiory rozmyte i zmienna lingwistyczna – proste zadania związane z definiowaniem zmiennej lingwistycznej oraz tworzenia i operacji na zbiorach rozmytych
- Model stanowy – przykład działającego, innowacyjnego modelu decyzyjnego – opis działania, zastosowania
- Portfel rozmyty i sterownik rozmyty – przykład prostego sterownika rozmytego i próba zamodelowania szkieletu systemu eksperckiego dla rozmytego portfela

Nazwa zajęć: **Programowanie w języku Python**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna i rozumie podstawowe techniki i pojęcia programowania proceduralnego oraz obiektowego.
- Potrafi czytać i analizować kod napisany w języku Python.
- Potrafi samodzielnie rozwiązać proste problemy z wykorzystaniem języka Python.

Treści programowe dla zajęć:

- Stałe i zmienne. Typy danych prostych. Złożone typy danych: sekwencje, słowniki, operacje na sekwencjach. Instrukcja przypisania.
- Operatory. Wyrażenia.
- Operacje wejścia/wyjścia.
- Sterowanie przebiegiem programu. Instrukcje warunkowe, złożone wyrażenia logiczne. Pętle. Listy składane.
- Funkcje. Definicja funkcji. Argumenty. Zakresy. Funkcje rekurencyjne. Operator lambda.
- Pliki i sposoby dostępu do nich. Operacje na plikach. Instrukcja try. Obsługa błędów i wyjątków.
- Moduły w Pythonie. Korzystanie z wybranych modułów. Tworzenie własnych modułów.
- Iteratory.
- Paradygmaty programowania obiektowego. Dziedziczenie, hermetyzacja i polimorfizm. Różnice między programowaniem strukturalnym a obiektowym.
- Klasy, obiekty, pola i metody.
- Debugowanie.
- Dokumentacja.

Nazwa zajęć: **Projekt magisterski 1**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi przygotować środowisko pracy w projekcie magisterskim.
- Potrafi zaprojektować użyteczny system informatyczny na potrzeby przygotowywanej pracy magisterskiej.
- Potrafi dostarczać częściowe rezultaty prac wykonywanych w ramach projektu magisterskiego.
- Potrafi współpracować przy i kierować pracą nad projektem informatycznym.

Treści programowe dla zajęć:

- Treści kształcenia ustala prowadzący w zależności od tematyki projektu magisterskiego.

Nazwa zajęć: **Projekt magisterski 2**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi zaimplementować użyteczny system informatyczny na potrzeby przygotowywanej pracy magisterskiej.
- Potrafi zapewnić wysoką jakość systemu informatycznego działającego na potrzeby przygotowywanej pracy magisterskiej.
- Potrafi dokumentować architekturę i działanie systemu informatycznego działającego na potrzeby przygotowywanej pracy magisterskiej.
- Potrafi wizualizować w przygotowanym systemie informatycznym informacje odpowiadające na problemy i pytania badawcze poruszane w pracy magisterskiej.
- Potrafi współpracować przy i kierować pracą nad projektem informatycznym.

Treści programowe dla zajęć:

- Treści kształcenia ustala prowadzący w zależności od tematyki projektu magisterskiego.

Nazwa zajęć: **Seminarium magisterskie 1**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Umie sprecyzować swoje zainteresowania zagadnieniami matematycznymi i informatycznymi związanymi z analizą i przetwarzaniem danych.
- Poglębia wiedzę z zakresu analizy i przetwarzania danych i jej zastosowań związaną z problematyką pracy magisterskiej.
- Potrafi dokonać krytycznej oceny źródeł wykorzystywanych w samokształceniu i opracowaniach własnych.
- Potrafi przygotować i zaprezentować krótkie opracowanie konkretnego problemu dotyczącego analizy i przetwarzania danych.
- Docenia znaczenie uczciwości intelektualnej i zna zasady etyczno-prawne przestrzegania dobrych obyczajów w nauce i biznesie.

Treści programowe dla zajęć:

- Treści kształcenia ustala prowadzący seminarium w zależności od tematyki seminarium powiązanej z tematami przygotowywanych prac magisterskich

Nazwa zajęć: Seminarium magisterskie 2

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna szczegółowe wymogi merytoryczne i formalne stawiane pracom magisterskim.
- Potrafi opracować szczegółowy konspekt przygotowywanej pracy magisterskiej.
- Potrafi zrozumieć treść publikacji specjalistycznych dotyczących zagadnień podejmowanych w pracy magisterskiej.
- Posiada pogłębioną wiedzę z tych dziedzin analizy i przetwarzania danych, które istotnie wiążą się z problematyką pracy magisterskiej.
- Swobodnie i w pełni potrafi korzystać z baz danych zawierających informacje o publikacjach naukowych i ze źródeł literaturowych.
- Potrafi przedstawiać ustnie zagadnienia poruszane w przygotowywanej pracy magisterskiej, dyskutować o nich, formułować własne wnioski i komentarze.
- Docenia znaczenie uczciwości intelektualnej i zna zasady etyczno-prawne przestrzegania dobrych obyczajów w nauce i biznesie.

Treści programowe dla zajęć:

- Treści kształcenia ustala prowadzący seminarium w zależności od tematyki seminarium powiązanej z tematami przygotowywanych prac magisterskich

Nazwa zajęć: Seminarium magisterskie 3

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi zrozumieć treść publikacji specjalistycznych dotyczących zagadnień podejmowanych w pracy magisterskiej
- Posiada pogłębioną wiedzę z tych dziedzin analizy danych i jej zastosowań, które istotnie wiążą się z problematyką pracy magisterskiej
- Potrafi przedstawiać ustnie, w sposób kompleksowy, najważniejsze zagadnienia poruszane w przygotowanej pracy magisterskiej, dyskutować o nich, formułować własne wnioski i komentarze
- Potrafi uzasadnić przyjętą koncepcję realizacji tematu (w szczególności metodykę, dobór i zakres wykorzystania literatury, kolejność prezentacji zagadnień)
- Posiada ugruntowaną wiedzę z zakresu przedmiotów obowiązkowych dla studiów drugiego stopnia na poziomie wymaganym na egzaminie magisterskim

Treści programowe dla zajęć:

- Treści kształcenia ustala prowadzący seminarium w zależności od tematyki seminarium powiązanej z tematami przygotowywanych prac magisterskich

Nazwa zajęć: Statystyka z językiem R

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna podstawy programu R. Umie utworzyć struktury danych w tym programie oraz wykonywać odpowiednie operacje na nich.
- Potrafi programować w języku R, a w szczególności pisać własne funkcje w tym programie.
- Umie wykorzystać narzędzia programu R do analizy statystycznej.
- Umie opisać rozkład empiryczny badanej cechy za pomocą odpowiednich tabel, wykresów oraz statystyk opisowych.
- Potrafi dobrać odpowiedni do danego zagadnienia model statystyczny. Potrafi dokonać estymacji punktowej i przedziałowej parametrów przyjętego modelu. Umie zbadać własności estymatorów.
- Zna konstrukcje i własności testów statystycznych. Potrafi dobrać odpowiedni test do rozważanego zagadnienia. Umie zbadać własności testów statystycznych.

Treści programowe dla zajęć:

- Program R: RStudio, system pomocy, pakiety, wektory atomowe, operatory arytmetyczne, logiczne, relacyjne, indeksowanie wektorów, wybrane funkcje wbudowane, listy, macierze, czynniki, ramki danych, odczytywanie i zapisywanie danych ze źródeł zewnętrznych
- Programowanie w języku R: instrukcje warunkowe, pętle, funkcje
- Opis rozkładu empirycznego badanej cechy za pomocą odpowiednich tabel, wykresów oraz statystyk opisowych, np. szereg rozdzielczy, histogram, wykres słupkowy, wykres kołowy, średnia, mediana, wariancja, odchylenie standardowe
- Model statystyczny (model normalny, wykładniczy, dwumianowy, Poissona, jednostajny, Rayleigha). Estymacja punktowa i przedziałowa parametrów modelu
- Testy statystyczne: hipotezy statystyczne, obszar krytyczny, błędy pierwszego i drugiego rodzaju, poziom istotności testu, p-wartość, test ilorazu wiarygodności, wybrane testy statystyczne

Nazwa zajęć: Język SQL w analizie danych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna podstawowe cechy i zadania systemu zarządzania relacyjną bazą danych; rozumie istotę relacyjnych baz danych i ma świadomość istnienia innych, pozarelacyjnych, modeli danych; wie jakie są najnowsze trendy w systemach bazodanowych i w jaki sposób tematyka baz łączy się z innymi dziedzinami
- Zna składowe relacyjnego modelu danych, w szczególności ograniczenia integralnościowe oraz jego podstawę teoretyczną
- Wykonuje podstawowe i zaawansowane operacje na bazie danych z wykorzystaniem języka SQL; programuje serwer bazodanowy.
- Wykorzystuje funkcje analityczne (funkcje okna) w tworzeniu wydajnych zapytań SQL.

Treści programowe dla zajęć:

- Omówienie historii baz danych; Podstawowe pojęcia relacyjnego modelu danych: relacja, atrybut, krotka, klucz podstawowy, klucz obcy, inne ograniczenia integralnościowe; algebra relacji i rachunki relacji; standardy SQL; przegląd systemów zarządzania bazami danych i klientów bazodanowych
- Język SQL – polecenie SELECT - filtrowanie, projekcja, sortowanie; podzapytania; złączenia wewnętrzne i zewnętrzne, samo-złączenia oraz anty-złączenia; funkcje agregujące oraz grupowanie; operacje na zbiorach
- Język SQL – polecenia DDL oraz DML; tworzenie obiektów bazodanowych, odczytywanie metadanych
- Język SQL – widoki, wyrażenia tablicowe CTE, tabele i zmienne tablicowe
- Język SQL – elementy programowania bazy danych: funkcje i procedury użytkownika, skrypty, operator LATERAL; procedury wyzwalane.
- Język SQL – funkcje analityczne (funkcje okna).
- No-SQL, ostatnie trendy w systemach baz danych, dziedziny i dalsze zagadnienia związane z bazami danych.

Nazwa zajęć: Systemy informatyczne

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Rozumie społeczne aspekty pracy zespołowej w projekcie informatycznym.
- Potrafi określić cechy innowacyjnego projektu informatycznego.
- Potrafi wizualizować system informatyczny za pomocą makiety.
- Potrafi przygotować się do procesu pozyskiwania inwestorów dla systemu informatycznego.
- Umie przygotować się do publicznej prezentacji koncepcji systemu informatycznego.
- Potrafi uczestniczyć w projekcie prowadzonym metodami zwinnymi.
- Potrafi uczestniczyć w procesie ciągłej integracji.
- Potrafi zintegrować system ciągłej integracji z systemem kontroli wersji.
- Potrafi opracować specyfikację zakresu systemu informatycznego.
- Potrafi organizować pracę w trakcie rozwoju systemu informatycznego.
- Potrafi zaprojektować użyteczny system informatyczny.
- Umie zorganizować proces przeprowadzenia testów systemu informatycznego.
- Potrafi uruchomić procesy prowadzące do pozyskania systemu informatycznego o wysokiej jakości.
- Potrafi planować zadania w projekcie informatycznym.
- Zna specyfikę zarządzania projektem informatycznym.
- Potrafi przedstawić cele i działanie systemu informatycznego jego interesariuszom.
- Potrafi przygotować demonstrację systemu informatycznego.
- Potrafi zaprezentować publicznie cele i działanie systemu informatycznego.

Treści programowe dla zajęć:

- Społeczne aspekty pracy zespołowej w projekcie informatycznym: Charakterystyka pamięci ludzkiej. Motywacja do pracy. Aspekty pracy zespołowej. Cechy lidera.
- Społeczne aspekty pracy zespołowej w projekcie informatycznym: Symulacja pracy zespołowej w projekcie informatycznym pod stresem czasowym.
- Charakterystyka innowacyjnego systemu informatycznego: Cechy innowacyjnego systemu informatycznego. Zasady osiągnięcia sukcesu na rynku nowych technologii.
- Charakterystyka innowacyjnego systemu informatycznego: Opracowanie koncepcji projektu spełniającego cechy innowacyjności. Wizualizacja systemu za pomocą makiety dynamicznej.
- Pozyskiwanie inwestorów: Typy i przykłady inwestorów. Sposoby prezentacji projektu: „elevator pitch”, prezentacja biznesowa, business plan.
- Pozyskiwanie inwestorów: Przygotowanie wypowiedzi typu "elevator pitch". Opracowanie slajdów prezentacji biznesowych dla potencjalnego inwestora.
- Publiczna prezentacja koncepcji projektu: Publiczne prezentacje biznesowe w obecności potencjalnych inwestorów i użytkowników projektowanych systemów.
- Publiczna prezentacja koncepcji projektu: Opracowanie business planu dla inwestorów. Określenie inwestora lub wskazanie klienta tworzonego oprogramowania.
- Metodyki zwinne: Podstawowe koncepcje metodyk zwinnych. Metodyka SCRUM.
- Metodyki zwinne: Poznanie narzędzia Jira. Zakładanie projektu w systemie Jira. Stworzenie dziennika projektu.
- Ciągła integracja: Metody prototypowania. Pojęcie ciągłej integracji. Współczesne systemy kontroli wersji. Projektowanie ciągłej integracji.
- Ciągła integracja: Określenie metody prototypowania. Otwarcie projektu w systemie kontroli wersji. Założenie projektu w serwerze ciągłej integracji. Integracja systemu kontroli wersji z systemem ciągłej integracji. Zaplanowanie pierwszego sprintu.
- Specyfikacja zakresu systemu informatycznego: Charakterystyka użytkowników. Lista aktor – cel. Lista in – out. Wymagania funkcjonalne i niefunkcjonalne. Formalny opis przypadków użycia.
- Specyfikacja zakresu systemu informatycznego: Opracowanie charakterystyki użytkowników, listy aktor – cel, listy in – out, wymagań funkcjonalnych i niefunkcjonalnych oraz przypadków użycia dla realizowanych projektów. Zaplanowanie drugiego sprintu.
- Wybrane zagadnienia użyteczności systemu informatycznego: Wprowadzenia pojęcia użyteczności systemu informatycznego. Omówienie aspektów użyteczności: kontekst, wprowadzanie danych, wyprowadzanie danych, responsywność, łączność z siecią, zasoby.
- Wybrane zagadnienia użyteczności systemu informatycznego: Modyfikacja dziennika projektu pod kątem użyteczności. Opracowanie szablonu raportu użyteczności dla projektowanego systemu.
- Zaplanowanie trzeciego sprintu.
- Wybrane zagadnienia testowania systemu informatycznego: Określenie pojęcia testowania. Tworzenie przypadków testowych. Typy testowania. Planowanie testów. Testowanie automatyczne.
- Wybrane zagadnienia testowania systemu informatycznego: Opracowanie planów testowania dla realizowanego projektu. Opracowanie testu automatycznego dla wybranej funkcjonalności projektu. Zaplanowanie czwartego sprintu.
- Ocena jakości systemu informatycznego: Określenie pojęcia jakości systemu informatycznego. Jakość systemu a jakość kodu. Miary jakości. Składowe jakości oprogramowania. Schematy oceny jakości.
- Ocena jakości systemu informatycznego: Opracowanie metryki jakości dla systemu podobnego do projektowanego. Ewaluacja jakości systemu podobnego. Opracowanie schematu oceny projektowanego systemu. Zaplanowanie piątego sprintu.
- Planowanie projektu: Zasady tworzenia harmonogramu projektu. Siatka podziału zadań. Zależności między zadaniami. Alokacja zasobów. Wykres Gantta. Narzędzie MS-Project.
- Planowanie projektu: Opracowanie harmonogramu wstecznego dla projektowanego systemu za pomocą programu MS-Project. Obliczenie wartości finansowej projektu. Zaplanowanie szóstego sprintu.
- Zarządzanie projektem informatycznym: Specyfika zarządzania zespołem informatycznym. Sterowanie presją. Czynniki decydujące o wydajności pracy.
- Zaliczenie działającej wersji prototypu systemu informatycznego realizowanego w ramach projektu.
- Publiczna demonstracja systemów opracowanych podczas semestru dla inwestorów i użytkowników.
- Podsumowanie wyników pracy podczas kursu

Nazwa zajęć: **Uczenie maszynowe - zastosowania**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Rozumie rolę i znaczenie uczenia, potrafi wskazać przykłady zastosowań uczenia maszynowego.
- Potrafi wyróżnić podstawowe typy zadań uczenia maszynowego i wskazać ich przykłady.

- Umie korzystać z podstawowych narzędzi biblioteki NumPy oraz elementów języka Python przydatnych do implementowania rozwiązań z dziedziny uczenia maszynowego.
- Umie przetwarzać dane przechowywane w tekstowych formatach tabelarycznych (CSV/TSV).
- Umie wizualizować dane, korzystając z biblioteki Matplotlib.
- Rozumie zagadnienie regresji liniowej jednej i wielu zmiennych.
- Rozumie metodę gradientu prostego.
- Umie zaimplementować algorytm gradientu prostego do znalezienia rozwiązania problemu regresji liniowej.
- Rozumie zagadnienie regresji logistycznej.
- Umie zaimplementować algorytm gradientu prostego do znalezienia rozwiązania problemu regresji logistycznej.
- Rozumie znaczenie ewaluacji algorytmów uczenia maszynowego i zna jej podstawowe metody.
- Rozumie rolę zbiorów danych: uczącego, walidacyjnego i testowego, i potrafi z nich korzystać.
- Zna podstawowe miary jakości stosowane przy ewaluacji algorytmów uczenia maszynowego.
- Potrafi korzystać z modułów pakietu Scikit-Learn do implementacji rozwiązań uczenia maszynowego.
- Potrafi dokonać ewaluacji zaimplementowanego rozwiązania.
- Rozumie zjawiska nadmiernego i niedostatecznego dopasowania.
- Zna metody regularyzacji.
- Umie zapobiegać nadmiernemu i niedostatecznemu dopasowaniu w implementowanych przez siebie rozwiązaniach.
- Umie poprawnie reprezentować dane różnych typów i korzystać z nich do rozwiązywania problemów metodami uczenia maszynowego.
- Rozumie znaczenie optymalizacji i zna jej podstawowe metody.
- Umie stosować metody optymalizacji uczenia maszynowego.
- Rozumie ideę uczenia nienadzorowanego i zna najważniejsze algorytmy uczenia nienadzorowanego.
- Potrafi zaimplementować przykładowy algorytm uczenia nienadzorowanego.
- Rozumie zasadę działania naiwnego klasyfikatora bayesowskiego.
- Rozumie zasadę działania algorytmu k najbliższych sąsiadów.
- Rozumie zasadę działania sztucznych sieci neuronowych, w tym wielowarstwowych.
- Potrafi wykorzystywać metodę propagacji wstecznej do uczenia wielowarstwowych sieci neuronowych.
- Potrafi implementować sieci neuronowe z wykorzystaniem biblioteki Keras.
- Rozumie zasadę działania i potrafi wskazać zastosowania rekurencyjnych sieci neuronowych.
- Rozumie zasadę działania i potrafi wskazać zastosowania spłotowych sieci neuronowych.
- Rozumie zasadę działania i potrafi wskazać zastosowania modeli typu encoder-decoder.
- Rozumie ideę uczenia przez wzmacnianie i zna podstawowe paradygmaty uczenia przez wzmacnianie.

Treści programowe dla zajęć:

- Wprowadzenie do uczenia maszynowego. Czym jest uczenie maszynowe? Uczenie maszynowe a analiza danych. Przegląd zastosowań i metod uczenia maszynowego. Podstawowe pojęcia związane z uczeniem maszynowym.
- Elementy języka Python przydatne przy implementowaniu algorytmów uczenia maszynowego. Bibliotek NumPy.
- Wczytywanie i prezentowanie danych. Formaty CSV i TSV. Tworzenie wykresów przy pomocy biblioteki Matplotlib.
- Regresja liniowa jednej zmiennej. Funkcja kosztu. Metoda gradientu prostego. Regresja liniowa wielu zmiennych. Normalizacja danych.
- Implementacja regresji liniowej jednej zmiennej w języku Python przy pomocy metody gradientu prostego.
- Dwuklasowa i wieloklasowa regresja logistyczna. Metoda gradientu prostego dla regresji logistycznej.
- Ewaluacja algorytmów uczenia maszynowego. Metodologia testowania. Zbiory uczący, testowy i walidacyjny. Walidacja krzyżowa. Miary jakości. Obserwacje odstające.
- Implementacja regresji liniowej wielu zmiennych w języku Python wraz z ewaluacją otrzymanego modelu.
- Implementacja regresji logistycznej.
- Regresja wielomianowa. Problem nadmiernego i niedostatecznego dopasowania. Metody regularyzacji. Krzywa uczenia się.
- Implementacja regresji wielomianowej. Problem nadmiernego i niedostatecznego dopasowania w praktyce. Implementacja metod regularyzacji.
- Naiwny klasyfikator bayesowski.
- Sposoby reprezentacji danych.

- Algorytm k najbliższych sąsiadów.
- Korzystanie z gotowych implementacji algorytmów na przykładzie pakietu scikit-learn. Implementacja naiwnego klasyfikatora bayesowskiego. Implementacja algorytmu k najbliższych sąsiadów.
- Metody uczenia nienadzorowanego. Algorytm k średnich. Algorytm analizy głównych składowych.
- Implementacja metod uczenia nienadzorowanego na przykładzie algorytmu k średnich i algorytmu PCA.
- Wprowadzenie do sieci neuronowych. Perceptron. Funkcje aktywacji. Wielowarstwowe sieci neuronowe.
- Propagacja wsteczna. Uczenie wielowarstwowych sieci neuronowych.
- Przegląd funkcji aktywacji. Wielowarstwowe sieci neuronowe w praktyce. Odmiany metody gradientu prostego. Algorytmy optymalizacji.
- Implementacja wielowarstwowych sieci neuronowych. Korzystanie z pakietu Keras.
- Rekurencyjne sieci neuronowe. Sieci LSTM i GRU.
- Implementacja rekurencyjnych sieci neuronowych na przykładzie sieci LSTM.
- Splotowe sieci neuronowe.
- Implementacja splotowych sieci neuronowych.
- Modele encoder-decoder. Autoencoder. Word embeddings. Tłumaczenie neuronowe.
- Uczenie przez wzmacnianie. Podstawy systemów dialogowych.

Nazwa zajęć: **Warsztat programisty**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi korzystać z podstawowych i średniozaawansowanych poleceń powłoki systemowej Bash, w tym zarządzać systemem operacyjnym i pisać proste skrypty.
- Potrafi korzystać z podstawowych poleceń interpretera poleceń Microsoft PowerShell.
- Potrafi zainstalować, uruchomić i korzystać z maszyny wirtualnej oraz z oprogramowania do konteneryzacji.
- Potrafi korzystać z rozproszonych systemów kontroli wersji.

Treści programowe dla zajęć:

- Linux – podstawowe komendy powłoki
- Linux – edytory tekstu
- Linux – strumienie plików i potoki; zmienne środowiskowe; przetwarzanie tekstu
- Linux – wyrażenia regularne
- Linux – operacje na plikach
- Linux – informacje o systemie; dowiązania twarde i symboliczne; wyszukiwanie aplikacji i plików; środowisko użytkownika; uprawnienia plików
- Linux – procesy
- Linux – skrypty
- Linux – przeglądarki internetowe i programy do pobierania z sieci; transfer danych przez sieć
- Linux – konfiguracja sieci; rozwiązywanie problemów; narzędzia DNS
- Linux – kompilacja; biblioteki; linkowanie
- Linux – konto root, su i sudo; instalacja i aktualizacja oprogramowania; konta, użytkownicy i grupy; planowanie przyszłych procesów; systemy plików
- Microsoft PowerShell – podstawowe polecenia
- Maszyny wirtualne, Windows Subsystem for Linux oraz konteneryzacja
- Systemy rozproszonej kontroli wersji; Git

Nazwa zajęć: **Wstęp do rachunku prawdopodobieństwa**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna podstawowe pojęcia rachunku prawdopodobieństwa. Rozumie pojęcie przestrzeni probabilistycznej.
- Potrafi stosować różne definicje prawdopodobieństwa.
- Rozumie pojęcie niezależności zdarzeń i jej znaczenia. Zna podstawowe schematy rachunku prawdopodobieństwa.
- Zna pojęcie zmiennej losowej oraz jej rozkładu. Umie wyznaczyć rozkład zmiennej losowej. Zna podstawowe rozkłady ciągłe i dyskretne.
- Potrafi wyznaczać podstawowe momenty zmiennych losowych.
- Rozumie prawa wielkich liczb oraz centralne twierdzenie graniczne wraz z ich zastosowaniem.
- Zna pojęcie wektora losowego, jego rozkładu oraz rozkładów brzegowych i warunkowych. Umie wyznaczać te rozkłady. Zna pojęcie niezależności zmiennych losowych i potrafi zweryfikować niezależność. Potrafi wyznaczać podstawowe momenty wektorów losowych.

Treści programowe dla zajęć:

- Przestrzeń zdarzeń elementarnych, zdarzenie losowe, algebra zdarzeń, przestrzeń probabilistyczna, własności prawdopodobieństwa
- Klasyczna przestrzeń probabilistyczna, dyskretna przestrzeń probabilistyczna, prawdopodobieństwo geometryczne, prawdopodobieństwo warunkowe, wzór na prawdopodobieństwo całkowite, wzór Bayesa
- Niezależność zdarzeń, schemat Bernoulliego
- Zmienna losowa, rozkład zmiennej losowej, dystrybuanta, gęstość, funkcja prawdopodobieństwa, rozkład dwupunktowy, dwumianowy, Poissona, geometryczny, jednostajny, wykładniczy, normalny
- Momenty zmiennych losowych, wartość oczekiwana, mediana, dominanta, wariancja
- Prawo wielkich liczb, mocne prawo wielkich liczb Kołmogorowa, centralne twierdzenie graniczne Lindeberga-Levy'ego
- Wektor losowy, rozkład wektora losowego, rozkład brzegowy, rozkład warunkowy, niezależność zmiennych losowych, kowariancja, korelacja

Nazwa zajęć: Zaawansowana wizualizacja danych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna i rozumie zasady komunikacji wizualnej w analizie danych, rozumie podstawy gramatyki wizualizacji (ang. grammar of graphics).
- Zna podstawowe składowe wykresów, rozumie ich strukturę wewnętrzną oraz relacje pomiędzy składowymi.
- Potrafi modyfikować gotowe szablony prezentacji danych stosując narzędzia programistyczne.
- Potrafi projektować własne style wykresów i je udostępniać.
- Zna zasady projektowania układów prezentacji danych oraz automatycznego dodawania elementów uzupełniających.
- Potrafi programować wielowarstwowe złożone układy prezentacji danych oraz modyfikować wykresy dostarczane przez zewnętrzne biblioteki.
- Potrafi zaprogramować własną klasę/moduł prezentacji danych i udostępnić ją do wykorzystania z zewnętrznymi zbiorami danych.

Treści programowe dla zajęć:

- Metody wizualizacji danych, typy wykresów w bibliotekach graficznych, komunikacja wizualna
- Anatomia wykresu, osie, koordynaty, skale i projekcje danych
- Programowanie składowych wykresów, dostosowanie wykresów do potrzeb, domyślna konfiguracja bibliotek graficznych
- Zarządzanie kolorem i skalami barwnymi, własne, nieliniowe skale barwne, style
- Projektowanie wykresu, układ, ornamentyka, podwykresy i okna, legendy
- Modyfikacja wykresów, wykresy złożone, warstwy danych, wykresy 3D, automatyzacja procesu wizualizacji
- Tworzenie własnych klas wykresów i ich udostępnianie - projekt zaliczeniowy

Nazwa zajęć: Język angielski 1

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi w sposób przystępny przedstawić fakty z zakresu informatyki i ogólnej terminologii naukowej, porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach w języku angielskim.
- Potrafi przygotować dokumentację, opracowania i raporty w języku angielskim.
- Potrafi przygotować wystąpienia ustne w języku angielskim, dotyczące zagadnień teoretycznych i praktycznych informatyki.
- Potrafi posługiwać się językiem obcym na poziomie B2+ Europejskiego Systemu Opisu Kształcenia Językowego oraz zna język angielski w stopniu umożliwiającym czytanie ze zrozumieniem dokumentacji oprogramowania.
- Zna i potrafi stosować słownictwo dotyczące informatyki oraz określenia używane w tekstach o charakterze naukowym.

Treści programowe dla zajęć:

- Opisywanie swojego zakresu pracy; Opisywanie sposobu działania (np. aplikacji, systemu)
- Cechy stylu naukowego. Stosowanie strony biernej do opisu procesów oraz wyrażania relacji przyczynowo-skutkowych
- Praca nad przygotowaniem indywidualnej prezentacji na temat związany z analizą i przetwarzaniem danych

- Praca z artykułami dotyczącymi Internetu, mediów społecznościowych, bezpieczeństwa informacji, nauka słownictwa za pomocą aplikacji opartych na interwałowym system powtarzania (Spaced Repetition System), takich jak Memrise i Anki (<http://ggregi.com/apps.htm>); korzystanie ze słowników internetowych i słowników zwrotów (collocations) (<http://www.freecollocation.com/>)

Nazwa zajęć: **Język angielski 2**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Potrafi w sposób przystępny przedstawić fakty z zakresu matematyki i analizy danych, porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach w języku angielskim.
- Potrafi przygotować dokumentację, opracowania i raporty w języku angielskim.
- Potrafi przygotować wystąpienia ustne w języku angielskim, dotyczące zagadnień teoretycznych i praktycznych matematyki oraz analizy danych.
- Potrafi posługiwać się językiem obcym na poziomie B2+ Europejskiego Systemu Opisu Kształcenia Językowego oraz zna język angielski w stopniu umożliwiającym czytanie ze zrozumieniem dokumentacji oraz artykułów dotyczących analizy danych.
- Zna i potrafi stosować słownictwo dotyczące analizy danych i matematyki.

Treści programowe dla zajęć:

- Praca nad wspólnym projektem, wymiana zdobytych informacji, zadania typu information gap.
- Tłumaczenie na język angielski dokumentacji i opracowań z wykorzystaniem cech stylu naukowego.
- Prezentacja wyników pracy nad projektem.
- Praca z artykułami dotyczącymi big data, wizualizacji danych oraz prawdopodobieństwa i statystyki
- nauka słownictwa za pomocą aplikacji opartych na interwałowym system powtarzania (Spaced Repetition System), takich jak Memrise i Anki (<http://ggregi.com/apps.htm>);
- korzystanie ze słowników internetowych i słowników zwrotów (collocations) (<http://www.freecollocation.com/>)

Nazwa zajęć: **Podstawy matematyki**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Zna symbole logiczne języków rachunku zdań oraz języka rachunku predykatów. Umie formułować wypowiedzi w językach rachunku zdań i rachunku predykatów. Zna pojęcie tautologii. Umie wartościować formuły języka rachunku zdań oraz sprawdzać, czy dane formuły są tautologiami. Umie negować formuły języka rachunku zdań oraz języka rachunku predykatów. Zna pojęcie zasięgu kwantyfikatora oraz zmiennej wolnej oraz zmiennej związanej. Umie wyznaczać zasięg kwantyfikatorów oraz zmienne wolne i zmienne związane. Zna pojęcie kwantyfikatorów o ograniczonym i nieograniczonym zakresie.
- Zna budowę oraz rodzaje definicji. Zna podstawowe rodzaje definicji. Zna budowę twierdzenia i umie wskazać założenia oraz tezę. Zna i rozumie sens pojęć warunek konieczny oraz warunek dostateczny w twierdzeniach w postaci implikacji. Zna i rozumie znaczenie dowodów w matematyce i informatyce.
- Zna pojęcie zbioru, należenia do zbioru, zawierania zbiorów i równości zbiorów. Umie opisać elementy należące do zbioru i określić, czy dwa zbioru są równe. Umie wykonywać podstawowe działania (w tym działania uogólnione) na zbiorach (suma, przekrój, różnica). Umie rysować na płaszczyźnie podwykresy i nawdwykresy funkcji rzeczywistych.
- Zna pojęcie relacji. Zna podstawowe własności relacji binarnych (tj. zwrotność, symetryczność, przechodniość, antysymetryczność, itp.). Umie sprawdzić, czy dana relacja posiada dane własności.
- Zna symbol sumy Σ i notację „sigma”. Umie posługiwać się zapisem sumy. Zna definicję silni oraz symbolu Newtona. Umie obliczać symbole Newtona oraz umie posługiwać się wzorem dwumianowym Newtona, korzystając z trójkąta Pascala.
- Zna algorytm dzielenia z resztą w pierścieniu liczb całkowitych. Zna pojęcie dodawania i mnożenia „modulo”. Umie dodawać i mnożyć „modulo” w pierścieniu liczb całkowitych.
- Zna zasadę mnożenia i dodawania. Umie wykorzystać zasadę mnożenia i dodawania przy zliczaniu obiektów. Zna pojęcia permutacji, wariacji i kombinacji (z powtórzeniami i bez). Umie dobrać właściwy schemat kombinatoryczny w zależności od sytuacji (wariacja vs. kombinacja) i za jego pomocą umie zliczać obiekty.
- Zna (naiwne, tzn. nieoparte na pojęciu relacji) pojęcie funkcji. Umie wskazać dziedzinę oraz przeciwdziedzinę funkcji. Umie wyznaczyć dziedzinę naturalną funkcji. Zna pojęcie równości dwóch funkcji. Umie sprawdzić, czy dwie funkcje są równe. Zna pojęcie złożenia funkcji. Umie wyznaczyć złożenie funkcji. Zna pojęcie monotoniczności funkcji. Umie w oparciu o definicję monotoniczności zbadać monotoniczność danej funkcji. Zna pojęcie funkcji okresowej. Umie sprawdzić, czy dana funkcja jest okresowa. Zna pojęcie funkcji parzystej i nieparzystej. Umie sprawdzić, czy dana funkcja jest

- parzysta/nieparzysta. Zna pojęcie iniekcji, suriekcji oraz bijekcji, a także funkcji odwrotnej. Umie sprawdzić, czy dana funkcja jest iniekcją/suriekcją/bijekcją, a w przypadku bijekcji potrafi wyznaczyć wzór funkcji odwrotnej. Potrafi zastosować pojęcie iniekcji/suriekcji/bijekcji w problemach wymagających analizy, np. w równaniach funkcyjnych. Zna kształt wykresów podstawowych funkcji. Zna pojęcie obrazu i przeciwobrazu funkcji. Potrafi wyznaczać obraz i przeciwobraz funkcji.
- Zna pojęcie wielomianu (w tym wielomianu zerowego) oraz równości dwóch wielomianów. Umie stwierdzić, czy dwa wielomiany są równe. Umie wskazać stopień wielomianu. Umie dzielić pisemnie wielomiany. Zna algorytm Hornera i umie zastosować go do dzielenia wielomianu przez dwumian. Rozumie, że algorytm Hornera wykorzystuje się w algorytmice do wyznaczania wartości wielomianu w punkcie. Zna pojęcie pierwiastka wielomianu i jego krotności. Umie obliczać pochodną formalną wielomianu i za jej pomocą potrafi wyznaczyć krotność danego pierwiastka. Umie znajdować pierwiastki wymierne wielomianów. Umie rozwiązywać równania i nierówności wielomianowe. Umie zamieniać postać ogólną trójmianu kwadratowego na kanoniczną lub iloczynową i odwrotnie. Zna wzory Viete'a i umie się nimi posługiwać. Wie, że wykresem funkcji kwadratowej jest parabola. Umie rysować wykresy funkcji kwadratowych. Zna wzory na wierzchołek paraboli i umie się nimi posługiwać.
 - Zna pojęcie funkcji wymiernej (w tym funkcji homograficznej). Umie wyznaczyć dziedzinę naturalną takiej funkcji. Umie szkicować wykres funkcji wymiernej (w tym funkcji homograficznej). Umie rozwiązywać równania i nierówności wymienne. Zna pojęcie ułamka prostego i umie rozkładać funkcję wymierną na sumę wielomianu i ułamków prostych.
 - Zna pojęcie potęgi i logarytmu (w tym logarytmu naturalnego). Zna własności działań na potęgach i logarytmach i umie się nimi posługiwać. Zna pojęcie funkcji wykładniczej i logarytmicznej. Zna i potrafi szkicować wykresy funkcji wykładniczej i logarytmicznej i podstawowe własności takich funkcji (związane np. z monotonicznością, iniektywnością, suriektywnością, bijektywnością). Umie rozwiązywać równania i nierówności wykładnicze i logarytmiczne.
 - Zna definicję wartości bezwzględnej i jej interpretację geometryczną. Wie, że za pomocą wartości bezwzględnej można liczyć odległość na prostej rzeczywistej. Zna podstawowe własności wartości bezwzględnej związane z równościami i nierównościami (w tym nierówność trójkąta). Umie rozwiązywać równania i nierówności z wartością bezwzględną.
 - Zna pojęcie miary łukowej kąta. Umie zamieniać miarę łukową na stopniową i odwrotnie. Zna definicję funkcji sinus, kosinus, tangens i kotangens dla dowolnego kąta. Potrafi szkicować wykresy tych funkcji i zna ich podstawowe własności. Umie wyznaczyć obraz punktu w obrocie o dany kąt. Umie wyznaczyć kąt obrotu. Umie obliczać wartości pozostałych funkcji trygonometrycznych na podstawie jednej danej. Zna podstawowe tożsamości trygonometryczne i umie je stosować przy wykazywaniu bardziej skomplikowanych. Umie obliczać wartości wyrażeń zawierających funkcje trygonometryczne. Umie rozwiązywać równania i nierówności trygonometryczne.
 - Zna pojęcie równoliczności dwóch zbiorów oraz mocy zbioru. Umie stwierdzać, czy dane zbiory są równoliczne (np. za pomocą konstrukcji odpowiedniej bijekcji). Zna pojęcie zbiorów przeliczalnych i nieprzeliczalnych. Potrafi sprawdzić, że dany zbiór jest przeliczalny. Zna pojęcie zbioru o mocy continuum. Umie sprawdzić, że dany zbiór jest mocy continuum.
 - Zna pojęcie ciągu, granicy ciągu oraz ciągu zbieżnego i rozbieżnego. Umie wyznaczać granice ciągów (korzystając z definicji, działań arytmetycznych na granicach lub twierdzenia o trzech ciągach).
 - Zna notację „duże O” i „małe o”. Potrafi posługiwać się notacją asymptotyczną.
 - Wie, czy jest szereg liczbowy oraz czym są sumy częściowe takiego szeregu. Zna pojęcie szeregu zbieżnego i rozbieżnego. Zna warunek konieczny oraz podstawowe kryteria zbieżności szeregów (kryterium porównawcze, kryterium d'Alemberta, kryterium Cauchy'ego). Umie zbadać, czy dany szereg o wyrazach nieujemnych jest zbieżny. Korzystając z teorii szeregów, potrafi wykazać, że pewne ciągi są zbieżne do zera.
 - Zna pojęcie granicy funkcji w punkcie (jednostronnej i obustronnej) i w nieskończoności. Umie wyznaczyć granice funkcji w punkcie i w nieskończoności (korzystając z definicji oraz działań arytmetycznych na granicach). Zna pojęcie funkcji ciągłej. Zna podstawowe własności funkcji ciągłych (np. działania arytmetyczne na funkcjach ciągłych, ciągłość złożenia, ciągłość funkcji elementarnych, itp.). Umie podać przykłady funkcji ciągłych oraz nieciągłych. Potrafi sprawdzić, czy dana funkcja jest ciągła w punkcie. W przypadku funkcji z parametrem, potrafi dobrać parametr tak, by funkcja była ciągła.
 - Zna pojęcie pochodnej pierwszego oraz wyższych rzędów oraz zna interpretację geometryczną pochodnej. Umie obliczać pochodną korzystając z definicji oraz ze wzorów. Zna podstawowe wzory służące obliczaniu pochodnych. Zna związek znaku pochodnej oraz monotoniczności funkcji. Umie zbadać monotoniczność funkcji bazując na analizie znaku pierwszej pochodnej. Zna pojęcie ekstremum lokalnego funkcji (maksimum oraz minimum lokalne). Zna warunek konieczny oraz dostateczny istnienia ekstremum lokalnego funkcji. Posługując się warunkiem dostatecznym (lub analizą znaku pochodnej), umie wyznaczyć ekstrema lokalne funkcji jednej zmiennej. Zna i umie posługiwać się regułą de

l'Hospitala przy obliczaniu granic funkcji. Zna pojęcie szeregu potęgowego. Wie, czym jest promień i koło zbieżności. Umie wyznaczyć promień zbieżności (korzystając ze wzoru Cauchy'ego-Hadamarda). Wie, czym jest szereg Taylora i szereg Maclaurina funkcji. Umie rozwijać m.in. funkcje wykładnicze i funkcje trygonometryczne (sinus i kosinus) w szeregi Maclaurina.

- Zna pojęcie funkcji pierwotnej i całki nieoznaczonej. Umie obliczać całki nieoznaczone posługując się m.in. metodami całkowania przez podstawianie i całkowania przez części, wykorzystując wzory na całki elementarne. Zna pojęcie całki oznaczonej (Riemanna) i wie, jaka jest jej interpretacja geometryczna. Umie obliczać pola powierzchni za pomocą całki funkcji jednej zmiennej. Zna pojęcie całki niewłaściwej Riemanna i umie się nim posługiwać.
- Wie, czym jest macierz i zna podstawowe rodzaje macierzy (macierze kwadratowe, symetryczne, itp.). Umie wykonywać podstawowe operacje na macierzach (dodawanie, odejmowanie, mnożenie, transponowanie). Zna pojęcie wyznacznika macierzy i umie go obliczać, korzystając m.in. z metody Sarrusa czy rozwinięcia Laplace'a. Zna pojęcie macierzy odwrotnej.
- Zna pojęcie pochodnych cząstkowych pierwszego oraz wyższych rzędów. Zna podstawowe wzory służące obliczaniu pochodnych cząstkowych. Umie obliczać pochodne cząstkowe korzystając ze wzorów. Zna pojęcie ekstremum lokalnego funkcji wielu zmiennych (maksimum oraz minimum lokalne). Zna warunek konieczny oraz dostateczny istnienia ekstremum lokalnego funkcji dwu zmiennych. Posługując się warunkiem dostatecznym, umie wyznaczyć ekstrema lokalne funkcji wielu zmiennych. Zna pojęcie całki iterowanej oraz całki wielokrotnej. Umie obliczać całki iterowane oraz całki podwójne po prostokątach i obszarach normalnych względem jednej z osi układu.
- Zna przykłady zastosowań omawianych metod/pojęć w analizie i przetwarzaniu danych.
- Umie zautomatyzować obliczenia za pomocą dostępnych programów lub/i samodzielnie napisanego skryptu/algorytmu.

Treści programowe dla zajęć:

- Elementy logiki matematycznej: Rachunek zdań (język rachunku zdań, wartościowanie formuł, tautologie); Rachunek predykatów (język rachunku predykatów, kwantyfikatory o ograniczonym i nieograniczonym zakresie, formułowanie wypowiedzi w języku predykatów)
- Metody matematyki: Definicje (budowa definicji, rodzaje definicji, przykłady definicji, błędy w definiowaniu); Twierdzenia (budowa twierdzenia, założenie, teza, warunek konieczny i dostateczny, twierdzenie proste/odwrotne/przeciwne/przeciwstawne, kwadrat logiczny); Dowody (rodzaje dowodów: wprost, nie wprost, niekonstruktywny);
- Elementy teorii zbiorów: Pojęcie zbioru (zasada ekstensjonalności) i należenia do zbioru, pojęcie zawierania (inkluzji), równość zbiorów, zbiór pusty; Działania na zbiorach (suma, przekrój, różnica, działania uogólnione); Podzbiory płaszczyzny (iloczyn kartezjański, nadwykresy i podwykresy funkcji)
- Relacje: Pojęcie relacji; Relacje binarne i ich własności
- Symbol sumy, dwumian Newtona: Symbol sumy Σ ; Silnia oraz symbol Newtona; Wzór dwumianowy Newtona
- Pierścienie typu Z_p : Dzielenie z resztą i bez reszty liczb całkowitych; Działania dodawania i mnożenia „modulo”
- Elementy kombinatoryki: Zasada dodawania i mnożenia; Permutacje; Wariacje; Kombinacje
- Funkcje: Pojęcie funkcji, dziedziny funkcji, argumentu oraz wartości funkcji; równość funkcji; Obraz i przeciwobraz; Złożenie funkcji; Monotoniczność funkcji (funkcje rosnące, malejące, nierosnące, niemalejące); Funkcje okresowe; Parzystość/nieparzystość funkcji; Iniekcja, suriekcja, bijekcja, funkcja odwrotna; Przekształcanie wykresów funkcji rzeczywistej jednej zmiennej rzeczywistej
- Wielomiany: Wielomian, współczynnik wielomianu, wyraz wolny, stopień wielomianu, wielomian zerowy, równość wielomianów; Dzielenie pisemne wielomianów, schemat Hornera; Pierwiastki wielomianów (krotność, pierwiastki wymierne, równania i nierówności wielomianowe); Trójmian kwadratowy (postać ogólna, kanoniczna, iloczynowa trójmianu kwadratowego, wzory na pierwiastki trójmianu kwadratowego, wzory Viete'a, wzory na wierzchołek paraboli)
- Funkcje wymierne: Funkcja wymierna, funkcja homograficzna; Miejsca zerowe funkcji wymiernej (równania i nierówności wymierne); Rozkład funkcji wymiernej na ułamki proste
- Funkcje wykładnicze i logarytmiczne: Własności działań na potęgach; Funkcja wykładnicza (definicja, wykres, własności); Własności logarytmów; Funkcja logarytmiczna (definicja, wykres, własności); Równania i nierówności logarytmiczne i wykładnicze
- Wartość bezwzględna: Pojęcie wartości bezwzględnej i funkcji wartość bezwzględna; Interpretacja geometryczna wartości bezwzględnej; Równania i nierówności z wartością bezwzględna
- Funkcje trygonometryczne: Funkcje trygonometryczne dowolnego kąta; Wykresy i własności funkcji sinus, kosinus, tangens, kotangens; Tożsamości trygonometryczne; Równania i nierówności trygonometryczne
- Teoria mocy: Pojęcie równoliczności zbiorów; Moc zbioru; Zbiory przeliczalne; Zbiory nieprzeliczalne

- Granice ciągów: Ciąg, wyraz ciągu, wyraz ogólny ciągu; Granica ciągu, ciąg zbieżny i rozbieżny; Metody obliczania granic ciągów (działania arytmetyczne na granicach, twierdzenie o trzech ciągach)
- Asymptotyka: Notacja „duże O”; Notacja „małe o”
- Szeregi: Szereg, wyraz ogólny szeregu, sumy częściowe szeregu; Szereg zbieżny i rozbieżny; Warunek konieczny zbieżności szeregów; Kryteria zbieżności szeregów o wyrazach; nieujemnych (kryterium porównawcze, kryterium d'Alemberta, kryterium Cauchy'ego)
- Granica funkcji, ciągłość funkcji: Granica funkcji w punkcie (granica obustronna, granica jednostronna); Granica funkcji w nieskończoności; Metody obliczania granic (działania arytmetyczne na granicach funkcji); Funkcja ciągła w punkcie i na zbiorze; Własności funkcji ciągłych (działania arytmetyczne na funkcjach ciągłych, ciągłość złożenia, ciągłość funkcji elementarnych)
- Rachunek różniczkowy funkcji jednej zmiennej: Pochodna funkcji jednej zmiennej pierwszego i wyższych rzędów; Metody obliczania pochodnych; Reguła de l'Hospitala; Zastosowanie pochodnych do badania funkcji (monotoniczność funkcji, ekstrema lokalne); Szeregi potęgowe (szeregi Taylora, szeregi Maclaurina)
- Rachunek całkowy funkcji jednej zmiennej: Funkcja pierwotna i całka nieoznaczona; Metody całkowania (całkowanie przez podstawienie, całkowanie przez części); Pojęcie i geometryczna interpretacja całki oznaczonej; Całka niewłaściwa
- Elementy algebry liniowej: Pojęcie macierzy, rodzaje macierzy (macierz kwadratowa, macierz symetryczna); Podstawowe operacje na macierzach (dodawanie, odejmowanie, mnożenie, transponowanie); Wyznacznik macierzy; Macierz odwrotna
- Rachunek różniczkowy i całkowy funkcji wielu zmiennych: Pojęcie funkcji wielu zmiennych; Pochodne cząstkowe pierwszego i wyższych rzędów; Zastosowanie pochodnych cząstkowych do badania funkcji wielu zmiennych (ekstrema lokalne); Całka iterowana; Całka podwójna na prostokątach i na obszarach normalnych

Nazwa zajęć: **Wstęp do informatyki**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student:

- Wie czym zajmuje się i jaką odpowiedzialność ponosi informatyk. Zna zasady tworzenia i rozwoju usług informatycznych.
- Zna podstawowe pojęcia z zakresu informatyki jako nauki, potrafi zapisywać liczby w różnych formatach stałopozycyjnych.
- Zna metody reprezentacji informacji i danych za pomocą liczb binarnych. Potrafi zapisywać liczby zmiennoprzecinkowe i wykonywać na nich operacje arytmetyczne.
- Zna podstawowe pojęcia teorii informacji, potrafi obliczyć ilość informacji w komunikacie, potrafi wyznaczyć entropię źródła informacji.
- Zna pojęcia średniej długości słowa kodowego, redundancji, różnych rodzajów kodów. Potrafi wyznaczać kod zwarty metodą Huffmana.
- Zna techniki kompresji danych stratnej i bezstratnej, potrafi obliczyć stopień kompresji danych.
- Zna metody przechowywania danych na poziomie logicznym oraz sprzętowym, zna podstawowe elementy formatu XML, potrafi zaprojektować prosty schemat XML
- Zna metody walidacji plików XML, potrafi zapisać schemat XML przy użyciu języka DTD
- Zna podstawowe techniki przetwarzania niskopoziomowego, pojęcie bramki logicznej, sumatora. Potrafi zapisać schematy funkcji logicznych przy użyciu bramek logicznych.
- Zna budowę lokalnych, średnich oraz rozległych sieci komputerowych, zna podstawowe urządzenia sieciowe, rozumie ogólne zasady działania sieci Internet, potrafi diagnozować urządzenia sieciowe
- Zna podstawowe protokoły sieciowe, potrafi opisać algorytm komunikacji w protokołach sieciowych.
- Zna najbardziej przydatne oprogramowanie, a w szczególności Google Search, Wolfram Alpha, Google Drive, potrafi wykorzystywać podstawowe oraz zaawansowane funkcje tych pakietów.
- Zna zasady formatowania tekstu przy użyciu LaTeX, potrafi przygotować prosty dokument LaTeX
- Zna szablon Beamer dla LaTeX, potrafi korzystać ze środowiska do przygotowywania dokumentów LaTeX
- Zna ideę repozytorium danych, podstawowe komendy pakietu Git, potrafi stworzyć i zarządzać repozytorium danych Git
- Zna zagadnienia bezpieczeństwa i wrażliwości danych, podstawowe algorytmy szyfrujące, algorytm RSA. Potrafi szyfrować i deszyfrować wiadomości przy użyciu oprogramowania GPG.

Treści programowe dla zajęć:

- Przedmiot informatyki, historia informatyki, firma informatyczna, usługi informatyczne
- Systemy pozycyjne, przeliczanie liczby dziesiętnej na binarną (część całkowita i ułamkowa), system ósemkowy, szesnastkowy, przeliczanie liczb z systemu dwójkowego na szesnastkowy i odwrotnie.

- Reprezentacja informacji: liczby stałe i zmiennoprzecinkowe, kodowanie napisów przy użyciu kodu ASCII, zapisywanie kolorów w modelu RGB (zadania przy tablicy), dźwięki (tw. o próbkowaniu), filmy
- Zapis liczb zmiennoprzecinkowych (IEEE754), działania arytmetyczne na liczbach zmiennopozycyjnych
- Teoria informacji, entropia źródła informacji
- Obliczanie ilości informacji, entropii źródła danych
- Kod zwarty, algorytm Huffmana
- Wyznaczanie kodu zwartego dla komunikatów
- Przetwarzanie informacji: kompresja danych tekstowych, obrazów oraz filmów
- Obliczanie stopnia kompresji danych dla znanych algorytmów kompresji tekstu i obrazów, kopiowanie danych przy użyciu rsync
- Przechowywanie danych - sposoby przechowywania danych, poziomy RAID, format XML
- Przygotowanie schematu XML dla wybranego modelu danych, np. dla księgarni
- Pliki XML - sposoby walidacji, zastosowanie
- Rozwijanie schematu XML dla wybranego modelu danych (DTD)
- Architektura komputera, programowanie niskopoziomowe (bramki logiczne)
- Funkcje i układy logiczne, kod Grey'a, sumatory
- Podstawowe wiadomości o sieciach komputerowych, topologie sieci, routing i sieć Internet
- Sprawdzanie konfiguracji urządzeń sieciowych, monitoring ruchu sieciowego
- Protokoły sieciowe
- Zapisywanie kroków komunikacji w najpopularniejszych protokołach sieciowych.
- Najbardziej przydatne oprogramowanie: Google Search, Drive, Wolfram
- Zaawansowane kwerendy do wyszukiwarek internetowych. Korzystanie z narzędzi Google (Google Drive, dokumenty, arkusze)
- Formatowanie tekstu przy użyciu LaTeX
- Przygotowywanie dokumentów przy użyciu LaTeX
- Prezentacje LaTeX beamer, środowiska do pracy w LaTeX
- Przygotowanie prezentacji przy użyciu szablonu Beamer
- Repozytoria danych (ze szczególnym uwzględnieniem Gita)
- Praca z systemem Git (ćwiczenia w laboratoriach)
- Bezpieczeństwo danych i szyfrowanie.
- Ćwiczenia z wykorzystaniem oprogramowania PGP.