МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

УТВЕРЖДАЮ
Директор ИЭиАС
С.И. Лукьянов

26.02.2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ТЕХНОЛОГИИ DATA MINING И BIG DATA

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения
заочная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	5

Магнитогорск
2020 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования
19.02.2020 г. протокол № 5

Зав. кафедрой _____ О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭ и АС
26.02.2020 г. протокол № 5

Председатель _____ С.И. Лукьянов

Рабочая программа составлена:
ст. преподаватель кафедры ВТиП, _____ М.В. Зарецкий

Рецензент:
Начальник отдела технологических платформ
ООО «Компас Плюс» , канд. техн. наук _____ Д.С. Сафонов

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Цель освоения дисциплины "Технологии Data Mining и Big Data":

- формирование у студентов представления о типах задач, возникающих в области интеллектуального анализа данных (Технологии Data Mining и Big Data);

- освоение основных подходов, применяемых при решении задач Data Minig и Big Data;

- освоение современных программных средств, применяемых при решении задач Data Minig и Big Data;

- получение навыков применения парадигм Data Minig и Big Data при решении задач в различных предметных областях.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Технологии Data Mining и Big Data входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Методы управления знаниями

Обработка экспериментальных данных на ЭВМ

Базы данных OLTP-систем

Базы и хранилища данных

Программные решения для бизнеса

Моделирование

Функциональное программирование

Объектно-ориентированное программирование

Философия

Прикладная математика

Программирование

Численные методы

Элементы линейной алгебры

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Выполнение и защита выпускной квалификационной работы

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Технологии Data Mining и Big Data» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-6 Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями	
ПК-6.1	Оценивает качество математической модели при формализации задачи предметной области
ПК-6.2	Оценивает качество разработанных алгоритмов для последующего кодирования
ПК-6.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 акад. часов, в том числе:

– контактная работа – 10,7 акад. часов:

– аудиторная – 10 акад. часов;

– внеаудиторная – 0,7 акад. часов

– самостоятельная работа – 93,4 акад. часов;

Форма аттестации - зачет с оценкой

Раздел/ тема дисциплины	Курс	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Концептуальные основы. Программный								
1.1 Данные, информация, знания.	5	0,5	1		6	Самостоятельное изучение учебной и научной литературы.	Беседа – обсуждение. Устный опрос	ПК-6.1, ПК-6.2
1.2 Основы языка R. Среда RStudio (RStudio Cloud). Хранилище CRAN и работа с ним.		0,5	1		10	Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы.	Беседа – обсуждение. Анализ программного кода. Устный опрос.	ПК-6.1, ПК-6.2
Итого по разделу		1	2		16			
2. Предварительная обработка данных. Проверка гипотез. Кластеризация.								
2.1 Предварительная обработка данных. Преобразование Raw Data в Tidy Data. Анализ выбросов.	5	0,5	1		14	Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы.	Беседа – обсуждение. Анализ программного кода. Устный опрос.	ПК-6.1

2.2 Проверка статистической гипотезы о параметрах генеральной совокупности. Проверка статистической гипотезы о законе распределения. Кластеризация.		0,5	1		16	Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы.	Беседа – обсуждение. Анализ программного кода. Устный опрос.	ПК-6.1
Итого по разделу		1	2		30			
3. Построение статистических зависимостей. Анализ и прогнозирование временных рядов. Обработка текстовой информации.								
3.1 Построение статистических зависимостей. Анализ временных рядов. Нахождение тренда.	5	1	1/1И		23	Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы.	Беседа – обсуждение. Анализ программного кода. Устный опрос	ПК-6.3
3.2 Обработка "сырого" текста. Разметка по частям речи. Лемматизация и стеммирование. Построение корпусов текстов. Выявление именованных сущностей.		1	1/1И		24,4	Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы.	Беседа – обсуждение. Анализ программного кода. Устный опрос.	ПК-6.3
Итого по разделу		2	2/2И		47,4			
4. Закрепление изученного материала и контроль качества усвоения								
4.1 Закрепление изученного материала. Контроль качества усвоения.	5					Изучение современных программных реализаций.	Критическое рассмотрение применения методов Data Mining и Big Data в реальных задачах.	ПК-6.1, ПК-6.2, ПК-6.3
Итого по разделу								
Итого за семестр		4	6/2И		93,4		зао	
Итого по дисциплине		4	6/2И		93,4		зачет с оценкой	

5 Образовательные технологии

1. Традиционные образовательные технологии ориентируются на организацию образовательного процесса, предполагающую прямую трансляцию знаний от преподавателя к студенту (преимущественно на основе объяснительно-иллюстративных методов обучения). Учебная деятельность студента носит в таких условиях, как правило, репродуктивный характер.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Семинар – беседа преподавателя и студентов, обсуждение заранее подготовленных сообщений по каждому вопросу плана занятия с единым для всех перечнем рекомендуемой обязательной и дополнительной литературы.

Практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе личностно значимого для них образовательного результата. Наряду со специализированными технологиями такого рода принцип интерактивности прослеживается в большинстве современных образовательных технологий. Интерактивность подразумевает субъект - субъектные отношения в ходе образовательного процесса и, как следствие, формирование саморазвивающейся информационно-ресурсной среды.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция–пресс-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении специализированных программных сред и технических средств работы с информацией.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Радченко И.А. Технологии и инфраструктура Big Data: Учебное пособие. [Электронный ресурс]. / И.А. Радченко, И.Н. Николаев – СПб.: Университет ИТМО, 2018. – 55 с. Режим доступа http://books.ifmo.ru/book/2138/tehnologii_i_infrastructura_Big_Data:_uchebnoe_posobie.htm

б) Дополнительная литература:

1. Шитиков В.К. Классификация, регрессия и другие алгоритмы Data Mining с использованием R [Электронный ресурс]./ В.К. Шитиков, С.Э. Мастицкий - Тольятти, Лондон, -2017, 351 с. Режим доступа: https://ranalytics.github.io/data-mining

в) Методические указания:

Горюнов Ю.В. Практикум по машинному обучению [Электронный ресурс]. / Ю.В. Горюнов, А.Н. Половинкин, Н.Ю. Золотых. Режим доступа: http://www.uic.unn.ru/~zny/ml/

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7 Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
Deductor Studio Academic	Согашение о сотрудничестве №06-2901\08 от 29.01.2008	бессрочно
Anaconda Python	свободно распространяемое ПО	бессрочно
Scilab Computation Engine	свободно распространяемое ПО	бессрочно
MathWorks MathLab v.2014 Classroom License	К-89-14 от 08.12.2014	бессрочно
NotePad++	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Лекционная аудитория ауд. 282 – Мультимедийные средства хранения, передачи и представления информации;

Компьютерные классы Центра информационных технологий ФГБОУ ВПО «МГТУ им. Г.И. Носова» – Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники;

Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки – ауд. 282 и классы УИТ и АСУ;

Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации – классы УИТ и АСУ;

Помещения для хранения и профилактического обслуживания учебного оборудования – Центр информационных технологий – ауд. 372

Приложение 1. Учебно-методическое обеспечение самостоятельной работы обучающихся.

Задание к лабораторной работе по теме:

Данные информация, знания.

Проанализировать совокупность текстовых, графических и аудиовизуальных данных, размещенных на одной из страниц WEB-ресурсов:

1. www.yandex.ru;
2. www.mail.ru;
3. www.rambler.ru;
4. www.gazeta.ru;
5. www.ura.ru;
6. www.znak.com;
7. www.mk.ru;

Задание к лабораторной работе по теме:

Основы языка R. Среда RStudio (RStudio Cloud). Хранилище CRAN и работа с ним.

1. Установить один пакет из CRAN в RStudio.
2. Загрузить файл в RStudio с локального носителя.
3. Скачать файл из RStudio на локальный носитель.
4. Написать функцию для суммирования элементов числового массива в R.
5. Написать функцию для нахождения среднего арифметического числового массива.
6. Написать функцию для нахождения стандартно отклонения числового массива.
7. Написать для нахождения суммы числовых элементов двух массивов.

Задание к лабораторной работе по теме:

Предварительная обработка данных. Преобразование Raw Data в Tidy Data. Анализ выбросов.

Установить необходимые пакеты.

1. Выполнить в среде R:

```
library(tibble)
tibble(
  a = 1:3,
  b = 1,
  c = -1:1
)
```

Проанализировать результат.

2. Выполнить в среде R:

```
dfr = data.frame(a = 1:3, b = 1,  c = -1:1)
as_tibble(dfr)
```

Проанализировать результат.

3. Выполнить в среде R.

```
tribble(
  ~a, ~b, ~c,
   1,  1, -1,
   2,  1,  0,
   3,  1,  1
)
```

Проанализировать результат.

4. Выполнить в среде R.

```
library(readr)
(okruga = read_csv('data/okruga.csv'))
```

Проанализировать результат.

5. Выполнить в среде R.

```
library(ISLR)
str(Carseats)
```

Проанализировать результат.

6. Выполнить в среде R.

```
library(ISLR)
income <- imputate_na(carseats, Income, US, method = "rpart")
income
```

Проанализировать результат.

7. Выполнить в среде R.

```
library(mice)
urban <- imputate_na(carseats, Urban, US, method = "mice")
```

Проанализировать результат.

Задание к лабораторной работе по теме:

Проверка статистической гипотезы о параметрах генеральной совокупности. Проверка статистической гипотезы о законе распределения. Кластеризация.

Для заданной выборки проверить гипотезу (уровень значимости принять равным 0,95):

1. О равенстве математического ожидания данной величине.

2. О том, что генеральная совокупность подчинена нормальному закону распределения.
3. О том, что генеральная совокупность подчинена закону распределения Пуассона.
4. О том, что генеральная совокупность подчинена логнормальному закону распределения.
5. О том, что генеральная совокупность подчинена гамма закону распределения.
6. О том, что генеральная совокупность подчинена экспоненциальному закону распределения.
7. Для данной выборки построить кластеризацию по 2 критериям.

Задание к лабораторной работе по теме:

Построение статистических зависимостей. Анализ и прогнозирование временных рядов. Нахождение тренда.

1. Для двух заданных выборок построить выборочный коэффициент корреляции. Проверить его значимость.
2. Для двух заданных выборок построить ранговый выборочный коэффициент корреляции. Проверить его значимость.
3. Для двух заданных выборок построить линейную регрессионную зависимость. Проанализировать остатки, оценить значимость каждого из коэффициентов и всего уравнения (зависимой считаем первую выборку).
4. Для двух заданных выборок построить нелинейную регрессионную зависимость. Проанализировать остатки, оценить значимость каждого из коэффициентов и всего уравнения (зависимой считаем первую выборку).
5. Для нескольких заданных выборок построить линейную регрессионную зависимость. Проанализировать остатки, оценить значимость каждого из коэффициентов и всего уравнения (зависимой считаем первую выборку).
6. Для нескольких заданных выборок построить нелинейную регрессионную зависимость. Проанализировать остатки, оценить значимость каждого из коэффициентов и всего уравнения (зависимой считаем первую выборку).
7. Для заданного временного ряда определить коэффициент автоковариации, построить уравнение тренда .

Задание к лабораторной работе по теме:

Обработка «сырого» текста. Разметка по частям речи. Лемматизация и стемминирование. Построение корпусов текстов. Выявление именованных сущностей.

Рассмотреть предложенный пример на языке Python, предназначенный для работы с сырым текстом и текстовыми корпусами.

1.

```
from nltk.corpus import gutenberg

from nltk import FreqDist

def Ling_01():

    print(gutenberg.fileids())
```

```
def Ling_02():
    fd = FreqDist()
    for word in gutenberg.words('austen-persuasion.txt'):
        fd.inc(word)
    print(fd.N())
    print(fd.B())
```

2.

```
from nltk.corpus import gutenberg
from nltk import FreqDist
def Ling_01():
    print(gutenberg.fileids())

def Ling_02():
    fd = FreqDist()
    for word in gutenberg.words('austen-persuasion.txt'):
        fd[word]+=1
    print(fd.N())
    print(fd.B())
```

3.

```
from nltk.book import *
def Sample_01():
    print(text1)
    print(text2)
    print(text3)

def Sample_02():
    C1 = text1.concordance('monstrous')
    C2 = text3.concordance('God')
```

```
    print(C1)
    print(C2)
```

4.

```
from nltk.book import *
def Sample_02():
    print('For monstrous')
    text1.concordance('monstrous')
    print('For great')
    text2.concordance('great')
    print('For God')
    text3.concordance('God')
```

5.

```
from nltk.book import *
def Sample_03():
    text1.similar('monstrous')
    text2.similar('little')
    text3.similar('God')
```

6.

```
from nltk.book import *
def Sample_04():
    text1.common_contexts(['monstrous', 'very'])
    text2.common_contexts(['little', 'great'])
    text3.common_contexts(['God', 'devil'])
```

7.

```
import nltk
def corp_01():
```

```
    ff = nltk.corpus.gutenberg.fileids()
    print(ff)
def corp_02():
    emma = nltk.corpus.gutenberg.words('austen-emma.txt')
    ll = len(emma)
    print(qq)
```

8.

```
import nltk
def corp_01():
    ff = nltk.corpus.gutenberg.fileids()
    print(ff)
def corp_02():
    emma = nltk.corpus.gutenberg.words('austen-emma.txt')
    ll = len(emma)
    print(ll)
def corp_03():
    emma = nltk.Text(nltk.corpus.gutenberg.words('austen-emma.txt'))
    emma.concordance('surprise')
```

9.

```
def WEB_02():
    url = "http://gutenberg.spiegel.de/buch/belagerung-von-mainz-3641/1"
#    url = "http://gutenberg.spiegel.de/buch/achilleis-7287/1"
    html=request.urlopen(url).read().decode('utf8')
    print(html[:60])
    raw=BeautifulSoup(html, 'html.parser').get_text()
    print(raw)
    tokenizer = TreebankWordTokenizer()
    tokens = tokenizer.tokenize(raw)
```

```
    tokens = tokens[110:390]
    print(tokens)
    text = nltk.Text(tokens)
    print(text)
    conc = text.concordance('gene')
    print(conc)
```

10.

```
def Disk_01():
    f = open('C:/Python_Prog/Deutsch/Goethe_02.txt','r',encoding='utf-8')
    raw = f.read()
    print(raw)
    f.close()
    german_tokenizer = nltk.data.load(
    'tokenizers/punkt/german.pickle')
    tokens = german_tokenizer.tokenize(raw)
    tokens = tokens[:500]
    print(tokens)
    text = nltk.Text(tokens)
    print(text)
    conc = text.concordance('das',lines=100)
    print(conc)
```

Индивидуальные задания к разделу 1.

Самостоятельно подключить необходимые библиотеки

```
library(xts)
library(lubridate)
library(plm)
library(forecast)
library(corrplot)
flu <- read.csv('http://www.google.org/flutrends/about/data/flu/ru/data.txt', skip = 10)
```

1.

```
plot(flu$Russia, type='l')
```

Проанализировать результат

2.

```
plot(flu$Date, flu$Russia, type='l')
```

Проанализировать результат

3.

```
TS <- ts(flu$Russia, frequency = 52, start = c(2004,10,3))
str(TS)
```

Проанализировать результат

4.

```
w <- chickwts$weight # Сохраняем веста в переменную w'
hist(w, breaks = 20) # Строим гистограмму с 20 колонками
hist(w, breaks = 20, freq = FALSE)
points <- seq(min(w), max(w), length.out = 100)
lines(points, dnorm(points, mean = mean(w), sd = sd(w)), col=2)
```

Проанализировать результат

5.

```
w <- chickwts$weight
plot(ecdf(w), do.points=FALSE, verticals = TRUE)
mean(w)
## [1] 261.3099
median(w)
## [1] 258
var(w)
## [1] 6095.503
sd(w)
```

Проанализировать результат

6.

```
set.seed(my.seed)
train <- sample(c(T, F), nrow(Hitters), rep = T)
test <- !train
# обучаем модели
regfit.best <- regsubsets(Salary ~ ., data = Hitters[train, ],
                          nvmax = 19)
```

Проанализировать результат

7.

```
k <- 10
```

```
set.seed(my.seed)
folds <- sample(1:k, nrow(Hitters), replace = T)
cv.errors <- matrix(NA, k, 19, dimnames = list(NULL, paste(1:19)))
```

Проанализировать результат

Индивидуальные задания к разделу 2.

1.

```
x <- model.matrix(Salary ~ ., Hitters)[, -1]
y <- Hitters$Salary
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
```

Проанализировать результат

2.

```
set.seed(my.seed)
train <- sample(1:nrow(x), nrow(x)/2)
test <- -train
y.test <- y[test]
ridge.mod <- glmnet(x[train, ], y[train], alpha = 0, lambda = grid,
                    thresh = 1e-12)
plot(ridge.mod)
```

Проанализировать результат

3.

```
set.seed(my.seed)
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 0)
plot(cv.out)
```

Проанализировать результат

4.

```
set.seed(my.seed)
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
plot(cv.out)
```

Проанализировать результат

5.

```
set.seed(2)a
```

```
pcr.fit <- pcr(Salary ~ ., data = Hitters, scale = T, validation = 'CV')
summary(pcr.fit)
```

Проанализировать результат

6.

```
set.seed(my.seed)
pcr.fit <- pcr(Salary ~ ., data = Hitters, subset = train, scale = T,
               validation = 'CV')
validationplot(pcr.fit, val.type = 'MSEP')
```

Проанализировать результат

7.

```
set.seed(my.seed)
pls.fit <- plsr(Salary ~ ., data = Hitters, subset = train, scale = T,
                validation = 'CV')
summary(pls.fit)
```

Проанализировать результат

Индивидуальные задания к разделу 3.

1.

```
my.seed <- 12345
train.percent <- 0.85
fileURL <- 'https://sites.google.com/a/kiber-guu.ru/msep/mag-
econ/salary_data.csv?attredirects=0&d=1'
wages.ru <- read.csv(fileURL, row.names = 1, sep = ';', as.is = T)
wages.ru$male <- as.factor(wages.ru$male)
wages.ru$educ <- as.factor(wages.ru$educ)
wages.ru$forlang <- as.factor(wages.ru$forlang)
```

Проанализировать результат

2.

```
my.seed <- 12345
n <- 100
train.percent <- 0.85
set.seed(my.seed)
```

```
x1 <- rnorm(20, 3.7, n = n)
set.seed(my.seed + 1)
x2 <- rnorm(50, 3.3, n = n)
rules <- function(x1, x2){
   ifelse((x1 > 20 & x2 < 50) | (x1 < 18 & x2 > 52), 1, 0)
}
```

Проанализировать результат

3.

```
set.seed(my.seed)
inTrain <- sample(seq_along(x1), train.percent*n)
x1.train <- x1[inTrain]
x2.train <- x2[inTrain]
x1.test <- x1[-inTrain]
x2.test <- x2[-inTrain]
y.train <- rules(x1.train, x2.train)
y.test <- rules(x1.test, x2.test)
df.train.1 <- data.frame(x1 = x1.train, x2 = x2.train, y = y.train)
df.test.1 <- data.frame(x1 = x1.test, x2 = x2.test)
```

Проанализировать результат

4.

```
<- 100              # наблюдений всего
train.percent <- 0.85  # доля обучающей выборки
set.seed(my.seed)
class.0 <- mvrnorm(45, mu = c(23, 49),
                   Sigma = matrix(c(3.5^2, 0, 0, 3.4^2), 2, 2,
                                  byrow = T))
set.seed(my.seed + 1)
class.1 <- mvrnorm(55, mu = c(15, 51),
                   Sigma = matrix(c(2^2, 0, 0, 2.5^2), 2, 2,
                                  byrow = T))
x1 <- c(class.0[, 1], class.1[, 1])
x2 <- c(class.0[, 2], class.1[, 2])
y <- c(rep(0, nrow(class.0)), rep(1, nrow(class.1)))
```

Проанализировать результат

5.

```
set.seed(my.seed)
x <- matrix(rnorm(20*2), ncol = 2)
y <- c(rep(-1, 10), rep(1, 10))
x[y == 1, ] <- x[y == 1, ] + 1
plot(x, pch = 19, col = (3 - y))
```

Проанализировать результат

6.

```
xtest <- matrix(rnorm(20*2), ncol = 2)
ytest <- sample(c(-1,1), 20, rep = TRUE)
xtest[ytest == 1, ] <- xtest[ytest == 1, ] + 1
testdat <- data.frame(x = xtest, y = as.factor(ytest))
ypred <- predict(bestmod, testdat)
table(predict = ypred, truth = testdat$y)
```

Проанализировать результат

7.

```
set.seed(my.seed)
x <- matrix(rnorm(200*2), ncol = 2)
x[1:100, ] <- x[1:100, ] + 2
x[101:150, ] <- x[101:150, ] - 2
y <- c(rep(1, 150), rep(2, 50))
dat <- data.frame(x = x, y = as.factor(y))
plot(x, col = y, pch = 19)
```

Проанализировать результат

Индивидуальные задания к разделу 4.

1.

```
from nltk.corpus import gutenberg
from nltk import FreqDist
def Ling_01():
    print(gutenberg.fileids())
```

```
def Ling_02():
    fd = FreqDist()
    for word in gutenberg.words('austen-persuasion.txt'):
        fd.inc(word)
    print(fd.N())
    print(fd.B())
```

Проанализировать результат

2.

```
from nltk.corpus import gutenberg
from nltk import FreqDist
def Ling_01():
    print(gutenberg.fileids())


def Ling_02():
    fd = FreqDist()
    for word in gutenberg.words('austen-persuasion.txt'):
        fd[word]+=1
    print(fd.N())
    print(fd.B())
```

Проанализировать результат

3.

```
from nltk.book import *
def Sample_01():
    print(text1)
    print(text2)
    print(text3)


def Sample_02():
    C1 = text1.concordance('monstrous')
```

```
    C2 = text3.concordance('God')
    print(C1)
    print(C2)
```

Проанализировать результат

4.

```
from nltk.book import *
def Sample_02():
    print('For monstrous')
    text1.concordance('monstrous')
    print('For great')
    text2.concordance('great')
    print('For God')
    text3.concordance('God')
```

Проанализировать результат

5.

```
from nltk.book import *
def Sample_03():
    text1.similar('monstrous')
    text2.similar('little')
    text3.similar('God')
```

Проанализировать результат

6.

```
from nltk.book import *
def Sample_04():
    text1.common_contexts(['monstrous', 'very'])
    text2.common_contexts(['little', 'great'])
    text3.common_contexts(['God', 'devil'])
```

Проанализировать результат

7.

```
import nltk
def corp_01():
```

```
    ff = nltk.corpus.gutenberg.fileids()
    print(ff)
def corp_02():
    emma = nltk.corpus.gutenberg.words('austen-emma.txt')
    ll = len(emma)
    print(qq)
```

Проанализировать результат.

Приложение 1. Оценочные средства для проведения промежуточной аттестации.

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-6: Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями		
ПК-6.1	Оценивает качество математической модели при формализации задачи предметной области	*Дано задание на разработку системы анализа производственных данных.* *1. Провести первичный разведочный анализ.* *2. Представить данные графически.* *3. Проверить гипотезу о законе распределения.* *4. На основании проведенного анализа оценить качество рассматриваемой математической модели.*
ПК-6.2	Оценивает качество разработанных алгоритмов для последующего кодирования	*1. Подобрать для алгоритмов обработки данных реализации в программных пакетах (работаем с хранилищами CRAN для R или GitHub для Python)* *2.С помощью программных средств из выбранных пакетов реализовать алгоритмы выявления зависимостей, оценки их параметров.* *3.Определить целесообразность применения непараметрических методов.* *4. Проверить работоспособность алгоритмов с помощью прототипа программы.*
ПК-6.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями	*Создать прототип программного продукта с использованием средств языка R и средств языка Python.* *Оценить основные характеристики программных средств:* *организация ввода и хранения данных;* *выполнение упорядочения данных “From Raw Data to Tidy Data”;* *выполнение обработки данных;* *возможность гибкого выбора наиболее приемлемой процедуры обработки;* *вывод результатов в текстовом и графическом виде.*

.