

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ
по дисциплине «Неконтролируемое машинное обучение»

для студентов направления подготовки
09.03.02 «Информационные системы и технологии»
Профиль «Прикладное программирование в интеллектуальных
информационных системах»

Ставрополь, 2026

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
ЛАБОРАТОРНАЯ РАБОТА 1. ИССЛЕДОВАТЕЛЬСКИЙ	
АНАЛИЗ ДАННЫХ	3
ЛАБОРАТОРНАЯ РАБОТА 2. ПОИСК АНОМАЛИЙ.....	12
ЛАБОРАТОРНАЯ РАБОТА 3. ИЗОЛИРУЮЩИЙ ЛЕС	18
ЛАБОРАТОРНАЯ РАБОТА 4. УМЕНЬШЕНИЕ	
РАЗМЕРНОСТИ.....	24
ЛАБОРАТОРНАЯ РАБОТА 5. КЛАСТЕРИЗАЦИЯ	34
ЗАКЛЮЧЕНИЕ	44
СПИСОК ЛИТЕРАТУРЫ	45

ЛАБОРАТОРНАЯ РАБОТА 1. ИССЛЕДОВАТЕЛЬСКИЙ АНАЛИЗ ДАННЫХ

Цели и задачи

Цель лабораторной работы: изучение программных средств для организации рабочего места специалиста по анализу данных и машинному обучению.

Основные задачи:

- установка и настройка среды разработки Python;
- изучение принципов загрузки и очистки данных;
- получение навыков по предварительной обработке данных на языке Python;
- изучение основных библиотек Python для работы с данными.

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с базовыми принципами языка Python, используя следующие источники: [1-5].

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): MS Visual Studio 2019 и выше; среда разработки Java, интерпретатор Python (Anaconda).

Методика и порядок выполнения работы

Перед выполнением индивидуального задания рекомендуется выполнить все пункты учебной задачи.

Учебная задача

Для начала работы с любой библиотекой необходимо импортировать ее:

```
import pandas as pd
```

Теперь нужно загрузить данные в структуру Pandas датафрейм (Dataframe). Она представляет собой двумерную структуру данных с именными столбцами потенциально разных типов – ее можно воспринимать как файл Excel, но на языке Python.

Для загрузки данных можно использовать функцию `pd.read_csv(path, delimiter)`. Для успешного использования необходимо указать путь (path) к файлу. В общем случае путь - это строка, содержащая полный путь к файлу и название файла.

Можно также указать разделитель (delimiter), в нашем случае это «;». Но для разных файлов могут использоваться и другие типы разделителей, такие как «;», « » (пробел), «\t» (табуляция). Вы можете проверить используемый в файле разделитель, предварительно открыв его в текстовом редакторе, например Notepad++.

Следующая строка загрузит файл с названием `cars.csv`, который лежит в папке `/content/`, разделитель – запятая:

```
df = pd.read_csv('/content/cars.csv', delimiter = ',')
```

Для того чтобы визуализировать содержание загруженного датафрейма в блокнотах Google Colab, достаточно просто запустить отдельную ячейку, в которой прописана переменная, содержащая датафрейм. В других случаях можно воспользоваться функцией `display`. На рисунке 1.1 представлена визуализация датафрейма набора Cars.

	Make	Model	Year	Style	Distance	Engine_capacity(cm3)	Fuel_type	Transmission	Price(euro)
0	Toyota	Prius	2011	Hatchback	195000.0	1800.0	Hybrid	Automatic	7750.0
1	Renault	Grand Scenic	2014	Universal	135000.0	1500.0	Diesel	Manual	8550.0
2	Volkswagen	Golf	1998	Hatchback	1.0	1400.0	Petrol	Manual	2200.0
3	Renault	Laguna	2012	Universal	110000.0	1500.0	Diesel	Manual	6550.0
4	Opel	Astra	2006	Universal	200000.0	1600.0	Metan/Propan	Manual	4100.0
...	...	...	...	...	...	...	...	...	...
41002	Dacia	Logan Mcv	2015	Universal	89000.0	1500.0	Diesel	Manual	7000.0
41003	Renault	Modus	2009	Hatchback	225.0	1500.0	Diesel	Manual	4500.0
41004	Mercedes	E Class	2016	Sedan	50000.0	1950.0	Diesel	Automatic	29500.0
41005	Mazda	6	2006	Combi	370000.0	2000.0	Diesel	Manual	4000.0
41006	Renault	Grand Scenic	2006	Minivan	300000.0	1500.0	Diesel	Manual	4000.0

41007 rows x 9 columns

Рисунок 1.1. Визуализация датафрейма набора Cars

Чтобы получить общую информацию о содержимом датафрейма, можно применить к нему метод `.info()`. Должна появиться следующая информация (рис. 1.2): количество столбцов, их имена, тип данных в каждом столбце, количество пропущенных значений для каждого столбца.

```

RangeIndex: 41007 entries, 0 to 41006
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Make                                  41007 non-null  object
1   Model                                41007 non-null  object
2   Year                                  41007 non-null  int64
3   Style                                41007 non-null  object
4   Distance                              41007 non-null  float64
5   Engine_capacity(cm3)                 41007 non-null  float64
6   Fuel_type                            41007 non-null  object
7   Transmission                         41007 non-null  object
8   Price(euro)                          41007 non-null  float64
dtypes: float64(3), int64(1), object(5)
memory usage: 2.8+ MB

```

Рисунок 1.2. Информация о датафрейме набора Cars

Повторяющиеся строки в наборе данных могут возникать по многим причинам: программная ошибка, человеческая ошибка и т. д. При этом никакой пользы для алгоритмов машинного обучения от включения повторных строк не будет.

Вы можете проверить количество повторяющихся строк, применив метод `.duplicated()` к датафрейму. В результате получится столбец, который содержит информацию о том, является ли данная строка повторной или нет. Можно применить метод `.sum()`, чтобы узнать количество повторных строк:

```
df.duplicated().sum()
```

Для датафрейма набора Cars имеется 3743 дубликата, почти 1/10 от всех данных. Чтобы удалить дубликаты, используем следующую строку:

```
DF = df.drop_duplicates().reset_index(drop=True)
```

Она означает, что для оригинального датафрейма `df` мы просим Python использовать метод удаления дубликатов `.drop_duplicates()` и сбросить индексы датафрейма `.reset_index(drop=True)` – в качестве альтернативы можно сохранить исходные индексы датафрейма. Результаты вышеупомянутого преобразования сохраняются в новый датафрейм `DF`.

Хорошей практикой является размещение результата значительного преобразования в отдельном датафрейме, чтобы всегда можно было вернуться к исходному. Вы можете проверить, что все сделано правильно, оценив количество строк дубликатов в новом датафрейме.

Для сохранения датафрейма в файл можно воспользоваться методом `.to_csv(path, index)`, которому необходимо указать полный путь и название файла, в который вы хотите сохранить данные (`path`). Кроме того, можно указать, необходимо ли сохранять индексы строк (булева переменная `index`). Так, следующая строка сохранит датафрейм без дубликатов в папку `/content/` с названием `'cars_no_dup.csv'`:

```
DF.to_csv('/content/cars_no_dup.csv', index=False)
```

Для визуализации данных можно использовать «стандартную» для Python библиотеку визуализации Matplotlib. Однако при использовании датафреймов Pandas целесообразно пользоваться библиотекой Seaborn:

```
import seaborn as sns
```

Seaborn – это библиотека для создания разнообразной графики на Python, которая тесно интегрирована со структурами данных Pandas.

Один из лучших графиков для ознакомления с данными – это парный график (pairplot). Он отображает все возможные попарные комбинации числовых признаков набора данных в виде скаттерограмм, а по диагонали строятся гистограммы распределений. Но если признаков много, то и времени на это уйдет немало - лучше разделить датафрейм на несколько частей. Категориальные признаки можно визуализировать, используя, например, цвет данных. При этом Seaborn позволяет реализовать визуализацию достаточно просто, буквально в одну строку:

```
sns.pairplot(data = DF, hue = 'Transmission')
```

Для того чтобы построить график, достаточно указать датафрейм, из которого будут взяты численные признаки в переменную data, дополнительно можно указать в качестве переменной hue (цвет) один из столбцов с категориальными данными. На рисунке 1.3 представлен pairplot для набора данных Cars.

После ознакомления с данными необходимо выполнить их предварительную обработку. Она зависит от типа данных. Для того чтобы определить типы данных, можно воспользоваться следующим кодом:

```
cat_columns = []
num_columns = []

for column_name in df.columns:
    if (df[column_name].dtypes == object):
        cat_columns += [column_name]
    else:
        num_columns += [column_name]
```

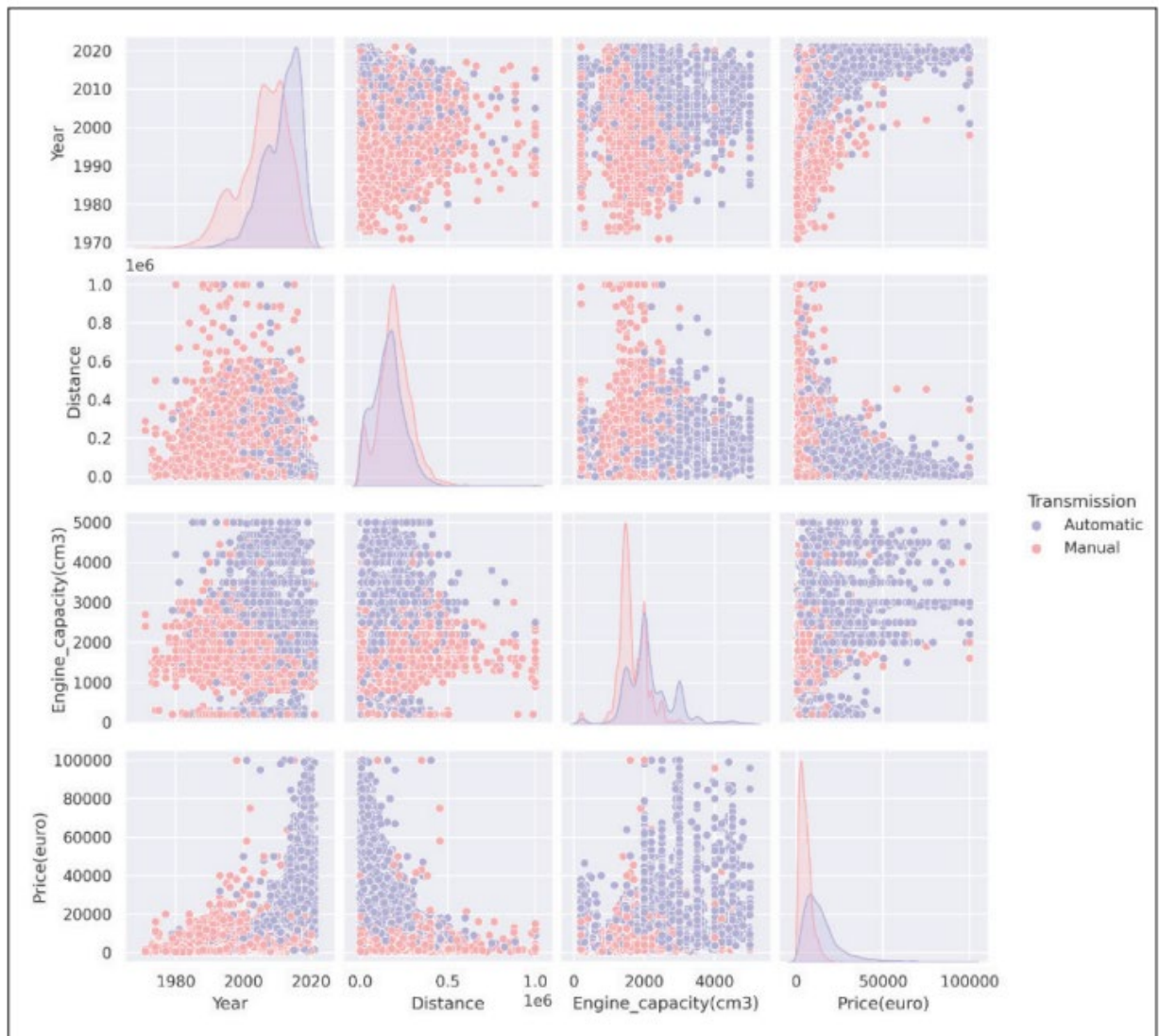


Рисунок 1.3. Визуализация pairplot для набора данных Cars

В переменную `cat_columns` попадут названия всех колонок с категориальными признаками, а в `num_columns` – с числовыми признаками. Обсудим методы предварительной обработки, характерные для разных типов данных.

Начать анализ числовых данных стоит с оценки статистических показателей. В библиотеке Pandas это можно сделать в одну строку с использованием метода `.describe()`:

```
DF[num_columns].describe()
```

На рисунке 1.4 представлен результат применения метода `.describe()` к набору данных Cars.

	Year	Distance	Engine_capacity(cm3)	Price(euro)
count	37264.000000	3.726400e+04	37264.000000	3.726400e+04
mean	2007.709264	4.758488e+05	1858.932535	9.569387e+03
std	8.295806	4.591520e+06	707.662731	5.283315e+04
min	1900.000000	0.000000e+00	0.000000	1.000000e+00
25%	2004.000000	9.000000e+04	1499.000000	3.300000e+03
50%	2009.000000	1.700000e+05	1800.000000	6.490000e+03
75%	2014.000000	2.300000e+05	2000.000000	1.179900e+04
max	2021.000000	1.000000e+08	9999.000000	1.000000e+07

Рисунок 1.4. Результат применения метода .describe()
к числовым признакам набора данных Cars

В результате демонстрируются следующие оценки: количество данных в столбце (count), среднее значение (mean), стандартное отклонение (std), минимальное значение (min), 25, 50 и 75 перцентили (25%, 50%, 75%), максимальное значение (max).

Индивидуальное задание

1. Скачайте набор данных по согласованию с преподавателем и используйте функции и методы библиотеки Pandas для загрузки и начальной работы с данными.
2. Выполните визуализацию данных с использованием библиотеки Pandas. Попробуйте построить разные виды графиков для числовых признаков – скаттерограммы, гистограммы и т. д. Для скаттерограмм попробуйте использовать категориальные данные для таких параметров графиков, как оттенок (hue), размер маркера (size), тип маркера (style). Таким образом, вы можете объединить информацию о нескольких признаках в один двумерный график.

3. Попробуйте добавить в модель дополнительные признаки на основе имеющихся. Проверьте корреляцию новых признаков с добавленными.
4. Выполните предварительную обработку данных. Сохраните результаты разных методов предварительной обработки в разные файлы, чтобы потом была возможность протестировать различные гипотезы.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Какие инструментальные средства используются для организации рабочего места специалиста Data Science?
2. Какие библиотеки Python используются для работы в области машинного обучения? Дайте краткую характеристику каждой библиотеке.
3. Почему при реализации систем машинного обучения широкое распространение получили библиотеки Python?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1-3, 9, 10].

ЛАБОРАТОРНАЯ РАБОТА 2. ПОИСК АНОМАЛИЙ

Цели и задачи

Цель лабораторной работы: изучение алгоритмов поиска аномалий.

Основные задачи:

- получение навыков создания одномерных данных;
- изучение основных типов регрессии;
- получение навыков реализации модели линейной регрессии.

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с базовыми принципами языка Python, используя следующие источники: [1-5]. Особое внимание необходимо уделить репозитарию [5] с исходными кодами.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): MS Visual Studio 2015 и выше; среда разработки Java, интерпретатор Python (Anaconda) с библиотеками matplotlib, seaborn, numpy.

Методика и порядок выполнения работы

Перед выполнением индивидуального задания рекомендуется выполнить все пункты учебной задачи.

Учебная задача

Перед тем как использовать алгоритм на реальных данных, в учебных целях бывает полезно тестировать его на данных, которые можно менять самостоятельно: чтобы рассмотреть различные ситуации и увидеть, как это влияет на результат.

Реализуем с помощью функции `true_fun` создание одномерных данных, которые могут быть некоторыми функциями от входных данных X :

```
def true_fun(x, a=np.pi, b = 0, f=np.sin):
    x = np.atleast_1d(x)[:]
    a = np.atleast_1d(a)
    if f is None: f = lambda x:x
    x = np.sum([ai*np.power(x, i+1) for i,ai in enumerate(a)],
axis=0)
    return f(x+ b)
```

В этой функции x – входные данные, a – коэффициент, на который данные умножаются, b – константная добавка, f – функция, которая применяется к входным данным. Стоит отметить, что для полиномиальных зависимостей достаточно использовать в качестве a список коэффициентов. Созданные данные могут быть как линейными, $f = \text{None}$, так и гармоническими (`np.sin`, `np.cos`) или экспоненциальными `np.exp`.

В учебных целях (допускающих не 100%-ю точность) мы также добавим шумы к данным с помощью функции `noises`. Аргументами этой функции являются `shape` (размер массива) и `noise_power` (величина шума):

```
def noises(shape , noise_power):
    return np.random.randn(*shape) *noise_power
```

Соберем это все в рамках единой функции `dataset`. Новыми аргументами в этой функции становятся N – количество точек, которые мы хотим сгенерировать, x_max – максимальное значение входных данных и булева переменная `random_x`. Если `random_x = True`, тогда будут случайным образом сгенерированы N точек, иначе данные будут распределены равномерно в диапазоне от 0 до x_max .

```
def dataset(a, b, f = None, N = 250, x_max = 1, noise_power
= 0, random_x = True, seed = 42):
    np.random.seed(seed)

    if random_x:
        x = np.sort(np.random.rand(N)) * x_max
    else:
        x = np.linspace(0, x_max, N)

    y_true = np.array([])

    for f_ in np.append([], f):
        y_true = np.append(y_true, true_fun(x, a, b, f_))

    y_true = y_true.reshape(-1, N).T
    y = y_true + noises(y_true.shape, noise_power)

    return y, y_true, np.atleast_2d(x).T
```

Выходными параметрами этой функции являются `y_true` (истинный вариант зависимости), `y` (зашумленный вариант), `x` (входные данные).

Приведем примеры команд для генераций данных и их визуализации:

```
y, y_true, x = dataset(a = 2, b = 2, f = None, N = 100, x_
max = 1, noise_power = 0.1, seed = 42)
```

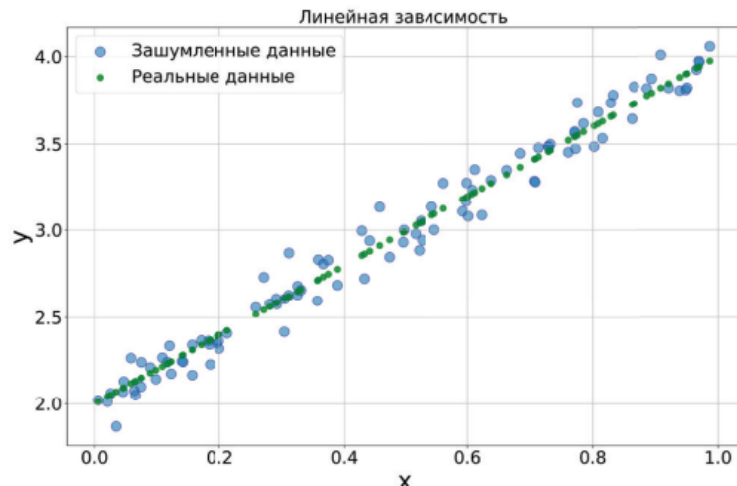


Рисунок 2.1. Сгенерированные данные с линейной зависимостью

```
y, y_true, x = dataset(a = [-1, 2, -2], b = -1, f = None, N = 250, x_max = 1.25, noise_power = 0.05, seed = 42)
```

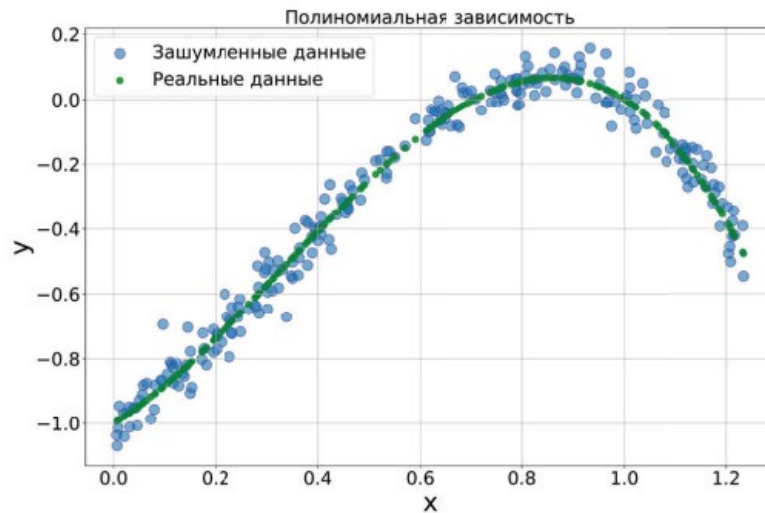


Рисунок 2.2. Сгенерированные данные с полиномиальной зависимостью

```
y, y_true, x = dataset(a = 3*np.pi, b = 0, f = np.cos, N = 25, x_max = np.pi/4, noise_power = 0.1, seed = 42)
```

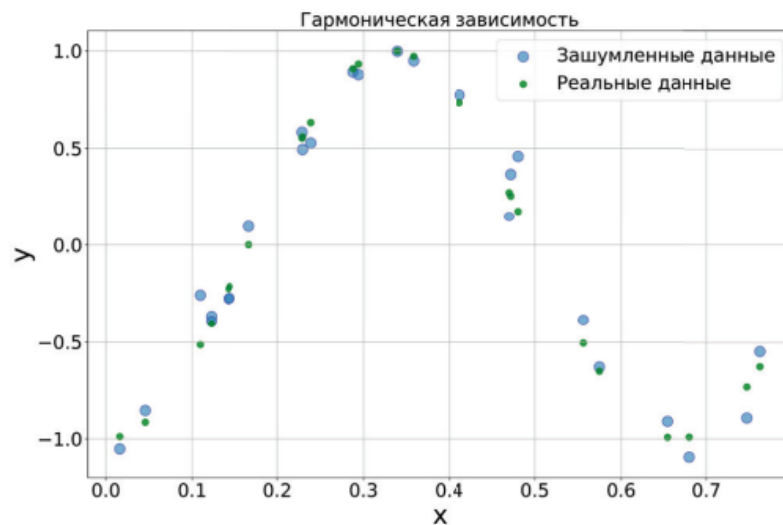


Рисунок 2.3. Сгенерированные данные с гармонической зависимостью

Индивидуальное задание

1. Подберите набор данных и согласуйте свой выбор с преподавателем.
2. Используя функцию `dataset`, сгенерируйте линейные зависимости с другими параметрами a , b , N , `noise_power` и проверьте модель линейной регрессии на этих данных.

3. Используя функцию `dataset`, сгенерируйте гармонические зависимости с другими параметрами и проверьте различные степени полиномов исходных данных и разные типы регуляризации. Посмотрите, что произойдет, если увеличить регуляризационный множитель.
4. Используйте любую из подготовленных вами моделей линейной регрессии для предсказания цены автомобилей в наборе данных `Cars`. Для оценки качества модели используйте отложенную выборку и несколько метрик регрессии. Сравните результаты модели при использовании только числовых признаков и при добавлении категориальных признаков с помощью One-Hot-кодирования.
5. Сравните работу реализованных алгоритмов с функциями библиотеки `scikit-learn`:
 - простая линейная регрессия через метод наименьших квадратов `sklearn.linear_model.LinearRegression`;
 - простая линейная регрессия через градиентный спуск `sklearn.linear_model.SGDRegressor`;
 - регрессия с регуляризацией Тихонова `sklearn.linear_model.Ridge`;
 - регрессия с L1-регуляризацией `sklearn.linear_model.Lasso`;
 - эластичная регуляризация `sklearn.linear_model.ElasticNet`.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Какие инструментальные средства используются для организации рабочего места специалиста Data Science?

2. Какие библиотеки Python используются для работы в области машинного обучения? Дайте краткую характеристику каждой библиотеке.

3. Почему при реализации систем машинного обучения широкое распространение получили библиотеки Python?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1-4].

ЛАБОРАТОРНАЯ РАБОТА 3. ИЗОЛИРУЮЩИЙ ЛЕС

Цели и задачи

Цель лабораторной работы: изучение принципов построения информационных систем с использованием метрических методов классификации.

Основные задачи:

- изучение инструментария Python для реализации алгоритмов метрической классификации;
- изучение методов оптимизации параметров метрической классификации;
- освоение модификаций kNN-метода.

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с теорией построения метрических классификаторов, используя следующие источники: [1-5]. Особое внимание необходимо уделить репозитарию [5] с исходными кодами.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): MS Visual Studio 2013 и выше; среда разработки Java, интерпретатор Python (Anaconda).

Методика и порядок выполнения работы

Перед выполнением индивидуального задания рекомендуется выполнить все пункты учебной задачи.

Учебная задача

Как и в случае с линейной регрессией, сначала сгенерируем данные. В этот раз мы воспользуемся готовыми функциями, которые реализованы в библиотеке `scikit-learn`, а именно `make_classification` для линейно разделимых данных, `make_moons` для данных, распределенных в виде знака инь-ян, и `make_circles` для данных, распределенных в виде концентрических кругов.

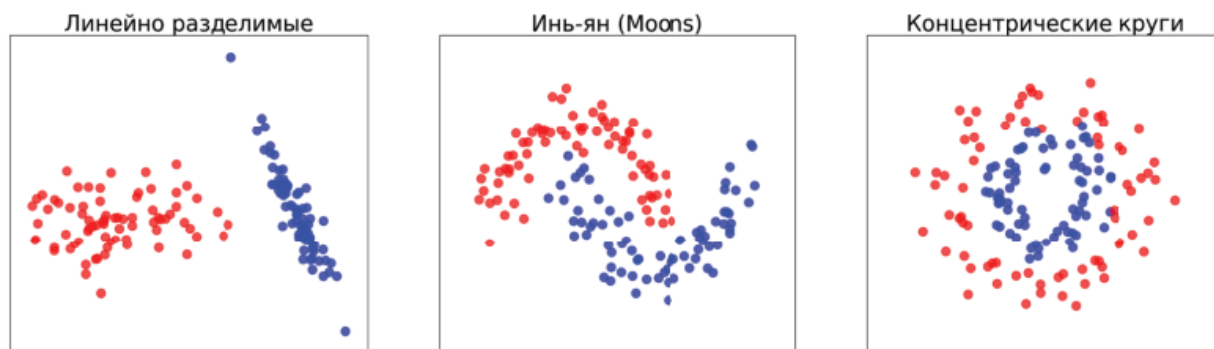


Рисунок 3.1. Типы генерируемых данных

Запишем отдельную единую функцию, которая будет генерировать данные. Входными параметрами этой функции будет N – количество точек, `method` – тип данных, который мы хотим сгенерировать, `noises` – уровень шума в данных. Выходными параметрами будут признаки – матрица X с размерами $N \times 2$ и вектор y меток классов для каждой точки.

```

from sklearn.datasets import make_moons, make_
circles, make_classification
def make_bin_clf(N, method = 'line', noises = 0.15, random_
state = 42):

    if random_state: rng = np.random.
RandomState(seed = random_state)

    if method == 'line' or method is None:
        X, y = make_classification(n_samples=N, n_
features=2, n_redundant=0, n_informative=2, n_clusters_per_
class=1, class_sep=2)
        X += np.random.randn(*X.shape) *noises

    elif method == 'moons':
        X, y = make_moons(n_samples=N, noise=noises)

    elif method == 'circles':
        X, y = make_circles(n_
samples=N, noise=noises, factor=0.5)

    return X,y

```

Запишем функцию определения класса:

```

def to_class(logit, threshold = 0.7):
    return (logit>=threshold)*1

```

В описанном смысле можно говорить о том, что результат применения функции сигмоиды – это вероятность того, что аргумент функции z - i принадлежит одному из классов. Такой аргумент принято называть логит. Отметим, что для расчета функции потерь не следует пользоваться округлением до классов. Теперь запишем функцию потерь. В значениях логарифма мы ввели небольшую константу ϵ с целью исключить ошибку вида «логарифм нуля»:

```

def bce_loss(yhat, y):
    eps = 1e-6
    return -(y*np.log(yhat + eps)+(1-y)*np.log(1-yhat+eps))
.mean()

```

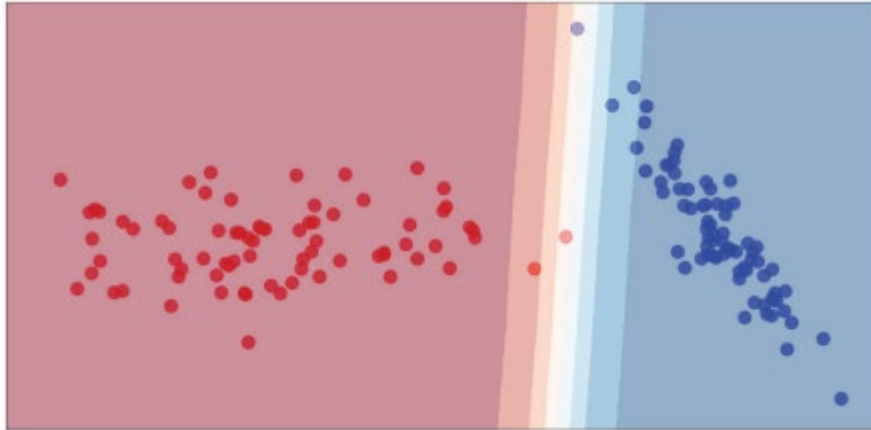


Рисунок 3.2. Визуализация функции принятия решений для линейно разделимых данных

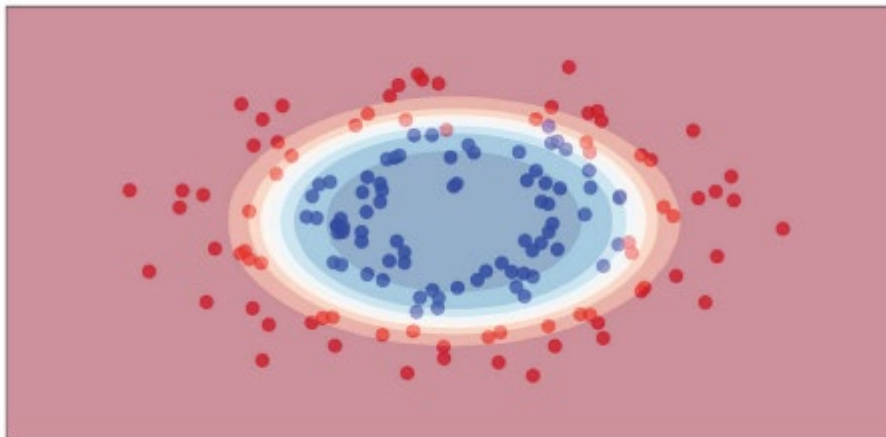


Рисунок 3.3. Визуализация функции принятия решений для концентрических кругов

Индивидуальное задание

1. Сгенерируйте линейно разделимые данные с другими параметрами и проверьте модель логистической регрессии на этих данных. Проанализируйте метрики классификации.
2. Сгенерируйте данные, распределенные как знак инь-ян или концентрические круги. Проверьте различные степени полиномов исходных данных и различные типы регуляризации для достижения наилучшего качества классификации. Проанализируйте метрики классификации.

3. Используйте модель логистической регрессии для предсказания. Для оценки качества модели используйте отложенную выборку и несколько метрик классификации. Сравните результаты модели при использовании только числовых признаков и при добавлении категориальных признаков с помощью One-Hot-кодирования.
4. Сравните работу реализованных алгоритмов с функцией библиотеки `scikit-learn` – логистической регрессией `sklearn.linear_model.LogisticRegression`.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Допустим, тест на некое заболевание R дал положительный ответ, хотя на самом деле у испытуемого нет этого заболевания. Какую ошибку допустил тест?
2. Пусть в матрице ошибок $TP = 5$, $TN = 90$, $FP = 10$, $FN = 5$. Оцените метрики классификации для такой матрицы ошибок.

3. Перечислите возможные гиперпараметры модели логистической регрессии.
4. В чем заключается метод парзеновского окна?
5. Поясните принцип метода потенциальных функций.
6. Назовите, какие параметры оптимизируют в методах kNN?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1-5].

ЛАБОРАТОРНАЯ РАБОТА 4. УМЕНЬШЕНИЕ РАЗМЕРНОСТИ

Цели и задачи

Цель лабораторной работы: изучение принципов построения информационных систем с использованием логических методов классификации.

Основные задачи:

- освоение метода разложения матриц (метод главных компонент);
- освоение технологии внедрения алгоритмов на основе решающих деревьев в приложения;
- изучение параметров логической классификации;
- освоение модификаций логических методов классификации.

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с теорией построения логических классификаторов, используя следующие источники: [1-5].

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): MS Visual Studio 2013 и выше; среда разработки Java, интерпретатор Python (Anaconda).

Методика и порядок выполнения работы

Учебная задача

Рассмотрим пример на данных, структуру которых мы можем контролировать. Это можно представить так: возьмем двумерную фигуру — эллипс с разными радиусами (рис. 4.1, а). При его рассмотрении может оказаться, что вторая ось системы координат не нужна. Таким образом, этот случай можно рассмотреть как одномерный (рис. 4.1, б).

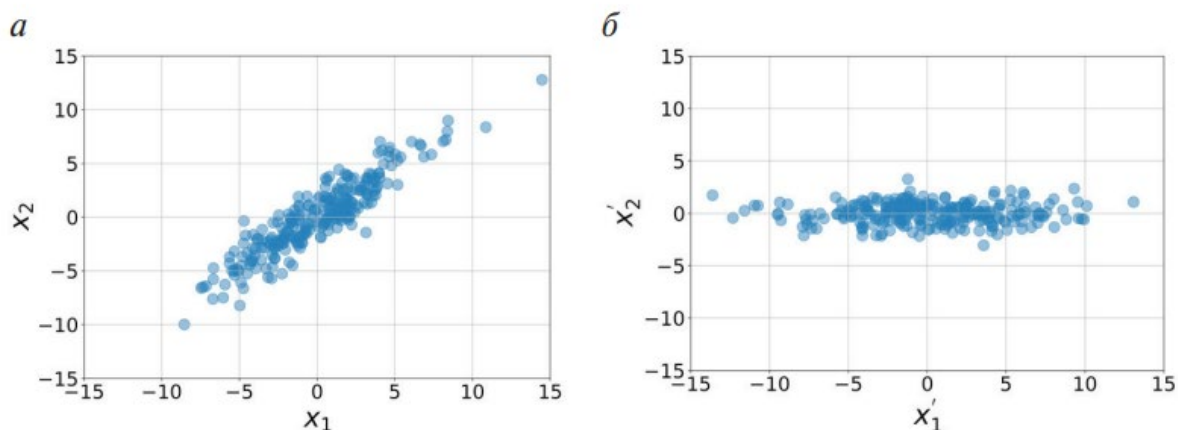


Рисунок 4.1. Уменьшение размерности двумерных данных

Сгенерировать данные, распределенные подобным образом, можно с использованием функции `create_ellipsoid_data`, где $C1$ и $C2$ — координаты центра, $S1$ и $S2$ — радиусы эллипса, θ — угол наклона, N — количество точек, `random_state` — фиксированный сид случайных чисел (для повторяемости генерируемых данных):

```
def create_ellipsoid_data(C1 = 0, C2 = 0, S1 = 5, S2 = 1, theta = 45, N = 250, random_state = 42):
    if random_state: np.random.seed(random_state)

    theta = np.pi * theta / 180
    Centers = np.array([C1, C2])
    Sigmas = np.array([S1, S2])

    R = np.array([[np.cos(theta), -np.sin(theta)],
                  [np.sin(theta), np.cos(theta)]])

    return (R @ np.diag(Sigmas) @ np.random.randn(2, N) + np.diag(Centers) @ np.ones((2, N))).T
```

В нашем примере одна ось, в которой расположена основная часть фигуры, будет главной компонентой. Совокупность главных компонент

образует так называемое собственное подпространство. Вторая ось останется шумовым подпространством. Как видно из примера на рис. 4.1, б, размерность фигуры в каждой из обозначенных осей будет соответствовать важности этой оси.

Другими словами, можно сказать, что разброс значений в каждой оси будет соответствовать ее важности. Такой разброс значений по каждой оси будет называться собственными значениями. Сортируя собственные значения по убыванию, мы можем определить те из них, которые следует оставить, и те, которые следует убрать.

В случае данных большей размерности происходят схожие преобразования: находятся оси многомерных эллипсов, в которых данные изменяются в большей степени. При этом уменьшение размерности можно реализовать за счет выбора не всех главных компонент, а только тех, которые имеют наибольшее собственное значение.

Возможно, поиск осей в двумерном эллипсе не выглядит как что-то впечатляющее. Посмотрим возможности метода главных компонент на более сложной задаче, наборе данных MNIST — это набор рукописных чисел от 0 до 9. Общее число изображений в наборе данных — 70000. Примеры изображений, которые содержатся в этом наборе данных, представлены на рис. 4.2.



Рисунок 4.2. Примеры изображений в наборе данных MNIST

Изображения в наборе MNIST – одноканальные, размером 28×28 пикселей. Воспользуемся простым подходом и будем рассматривать эти изображения как вектор признаков из 784 параметров. Используем функцию `fetch_openml` для загрузки данных с сайта OpenML, имя набора данных `'mnist_784'`. Помимо набора данных MNIST, сайт OpenML содержит большое число других наборов данных, которые можно использовать для практики алгоритмов машинного обучения.

В коде ниже укажем: необходимо, чтобы функция вернула только массив данных и вектор меток (`return_X_y=True`), а не полные данные в формате словаря. При этом добавим, что данные должны быть в формате массивов `numpy array`, а не датафреймов `Pandas` (`as_frame = False`).

```
from sklearn.datasets import fetch_openml
X, y = fetch_openml('mnist_784', version=1, return_X_y=True, as_frame = False)
```

Данные в наборе MNIST хранятся в формате 8 бит (т. е. интенсивность пикселя закодирована числом от 0 до 255). Нормализуем данные, для этого достаточно поделить все значения на 255. При этом данные будут изменяться в диапазоне от 0 до 1, где 0 – это черные пиксели, а 1 — белые. Промежуточные значения – 254 оттенка от черного к белому.

Воспользуемся подготовленным нами классом `PCA` и применим метод главных компонент к набору данных MNIST, указав `n_components` равным 100. На рисунке 4.3 представлена визуализация собственных значений для проведенного преобразования. Видно, что после 100-го собственного значения график асимптотически стремится к 0.

На рисунке 4.4 представлена визуализация скаттерграмм для первых четырех главных компонент. Каждая цифра представлена точкой соответствующего цвета. Несмотря на ограниченную размерность, ряд цифр сгруппированы «по группам». Так, цифры 0 и 1 стоят обособленно от остальных в координатах первой главной компоненты, вторая главная компонента позволяет отделить цифру 2 от остальных, а четвертая главная

компонента позволяет отсечь часть представлений цифры 6. При этом перемешанными оказались цифры 3 и 5, а также цифры 4, 7 и 9, что вполне объяснимо из-за схожести в их написании. Тем не менее это достаточно хороший результат для такой простой модели.

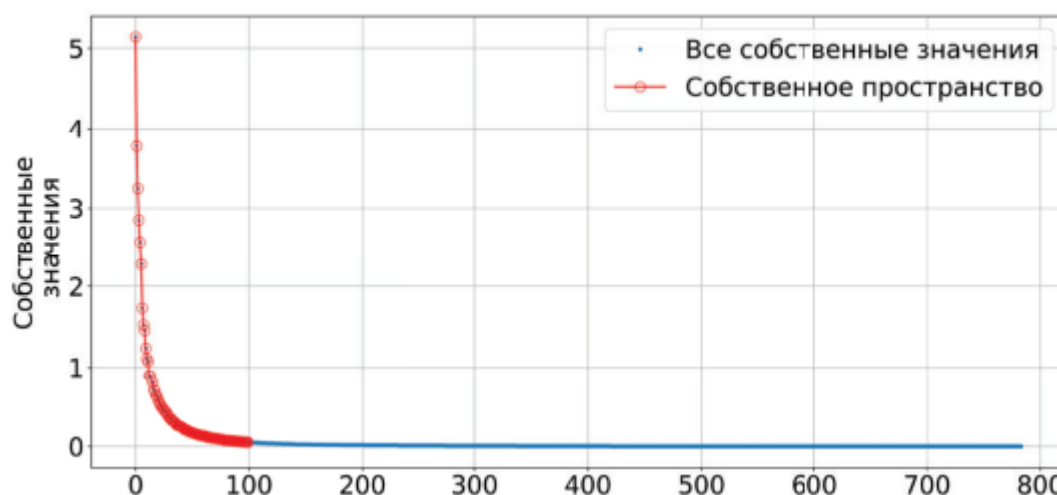


Рисунок 4.3. Визуализация собственных значений для набора данных MNIST

Попробуем интерпретировать полученные результаты. Для этого визуализируем собственные вектора. Сгруппируем их таким образом, чтобы можно было составить изображение, аналогичное исходным. На рисунке 4.5 красным цветом обозначены положительные значения, а синим — отрицательные. С учетом того, что главные компоненты — это линейная комбинация исходных признаков, «работу» первой главной компоненты можно интерпретировать следующим образом: берется центральная область изображения с положительными весами и окружающая ее область с отрицательными.

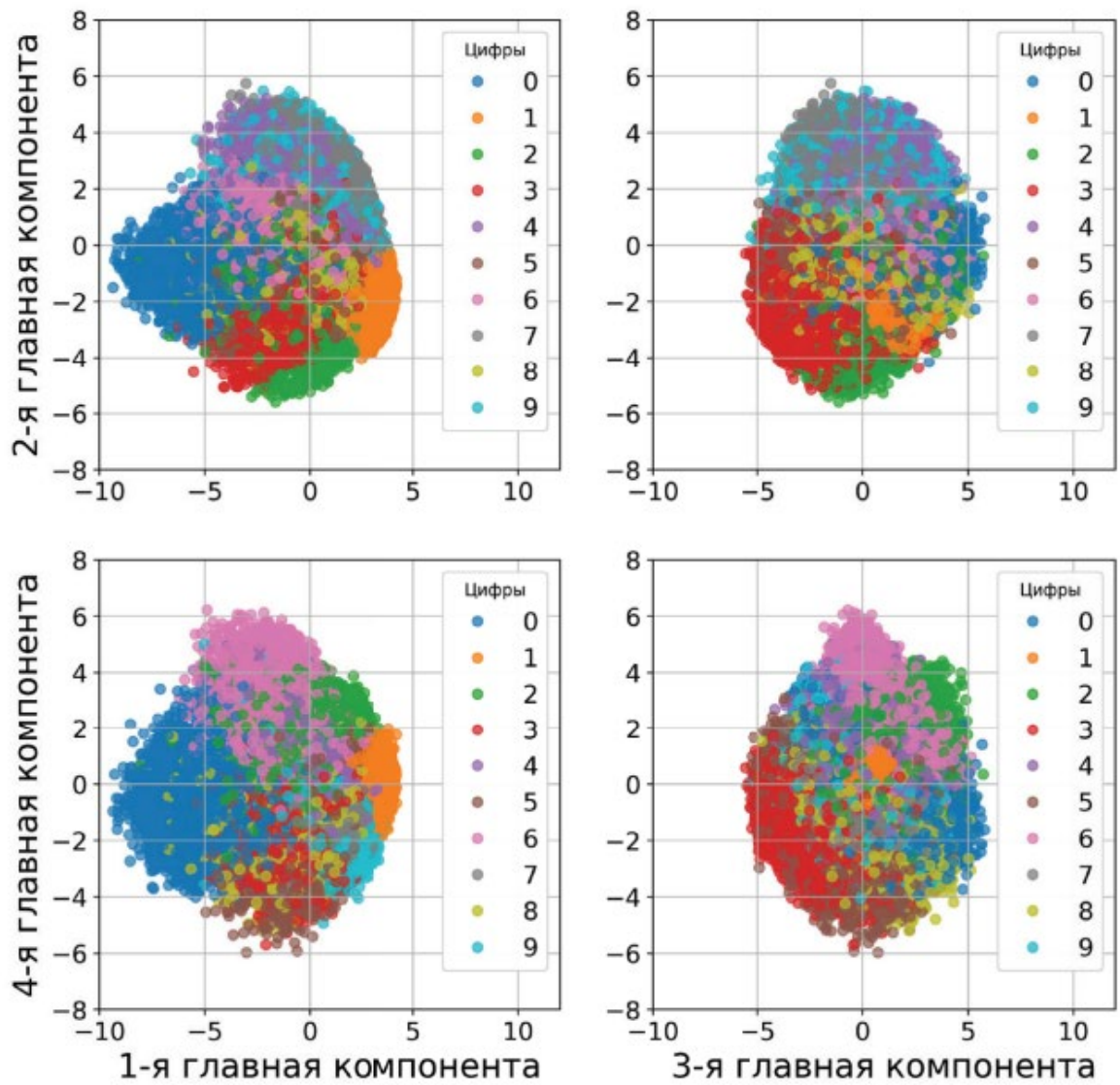


Рисунок 4.4. Визуализация пространства четырех главных компонент для набора данных MNIST

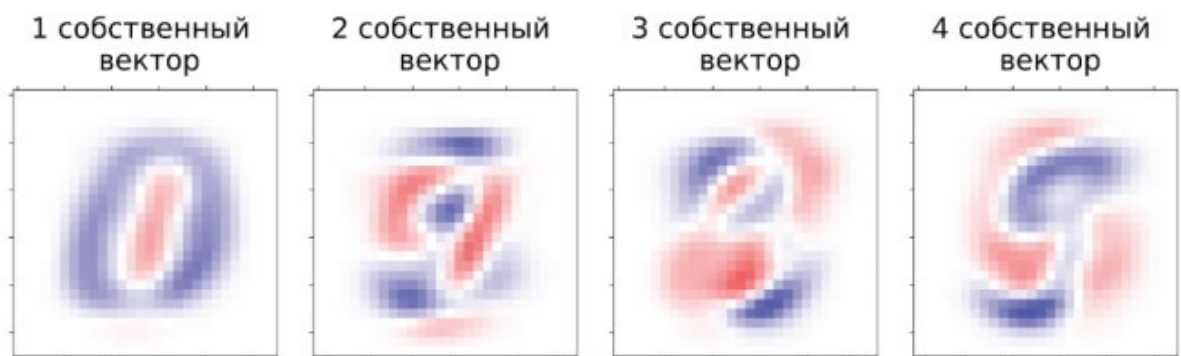


Рисунок 4.5. Визуализация четырех собственных векторов для набора данных MNIST

Таким образом, если изображение похоже на цифру 1, то для этой главной компоненты получится положительное число (светлые пиксели исходного изображения будут помножены на положительные веса). Если изображение похоже на цифру 0, то для этой главной компоненты получится отрицательное число (светлые пиксели исходного изображения будут помножены на отрицательные веса). Для других цифр результат зависит от того, сколько светлых пикселей попадут в зоны положительных и отрицательных весов.

Аналогичным образом можно попытаться интерпретировать другие главные компоненты. Однако, к сожалению, такой красивой и однозначной трактовки, как с первой главной компонентой и цифрами 0 и 1, не получится.

Наконец, посмотрим, как будут выглядеть изображения, если их реконструировать из 100 главных компонент. На рисунке 4.6 представлены реконструированные изображения и соответствующие им оригиналы. Видно, что в целом восстановленные изображения соответствуют оригиналам, хотя имеются шумы и артефакты.



Рисунок 4.6. Реконструкция изображения с использованием главных компонент

Однако стоит отметить ограниченность метода главных компонент в общем случае. Так, найденные собственные вектора будут относительно хорошо работать на новых данных, которые «похожи» на те, что были в

обучающем наборе. Но если цифры будут наклонены под другим углом, то результаты восстановления могут быть неожиданными. Кроме того, если изображения более сложные, например различные изображения котов, то собственные вектора могут носить случайных характер.

Индивидуальное задание

1. Сгенерируйте данные в виде эллипса с разными значениями радиусов и углов наклона. Примените метод главных компонент. Визуализируйте пространство главных компонент, оцените собственные значения и собственные вектора.
2. Загрузите данные MNIST. Поэкспериментируйте с количеством компонент при применении метода главных компонент. Оцените качество восстановления при разных значениях размерности собственного пространства. Визуализируйте разные пространства главных компонент.
3. Примените метод главных компонент для выбранного набора данных:
 - выполните визуализацию пространства главных компонент и оцените их связь с исходными признаками;
 - примените пространство главных компонент в качестве входных данных для алгоритмов;
 - сравните результаты модели при использовании только числовых признаков и при добавлении категориальных признаков с использованием One-Hot-кодирования.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Ответы на контрольные вопросы.

4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Как связаны главные компоненты с исходными данными?
2. Сделайте грубую оценку сжатия данных, если исходная матрица имела размерность $(4250, 7)$, а при восстановлении используются три главные компоненты.
3. Сгенерируйте данные в виде эллипса с центром в точке $(1.5, -2.5)$, радиусами $(3, 2.5)$, углом 65 и количеством точек 1100 . Оцените собственные вектора, собственные значения, максимальные и минимальные значения в пространстве главных компонент.
4. Что такое многоклассовая информативность? Для чего она применяется?
5. Поясните назначение и алгоритм бинаризации количественных признаков.
6. Поясните порядок поиска закономерностей в форме конъюнкций.

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1–5].

ЛАБОРАТОРНАЯ РАБОТА 5. КЛАСТЕРИЗАЦИЯ

Цели и задачи

Цель лабораторной работы: изучение принципов построения информационных систем с использованием линейных методов машинного обучения.

Основные задачи:

- освоение технологии внедрения алгоритмов линейной классификации в приложения;
- изучение основных приемов работы с разреженными матрицами в ходе машинного обучения;
- освоение техники построения, обучения и оценки модели логистической регрессии;
- освоение приемов работы с синтезированными признаками, масштабированием и настройкой гиперпараметров.

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с теорией построения регрессионных моделей машинного обучения, используя следующие источники: [1-5].

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): MS Visual Studio 2013 и выше; среда разработки Java, интерпретатор Python (Anaconda).

Методика и порядок выполнения работы

Учебная задача

Для простоты описания используем двумерный случай, однако все описанные метрики будут работать и в случаях пространств большей размерности.

На рисунке 5.1 представлены эквидистантные фигуры для различных метрик расстояний. Все точки, лежащие на прямых одного цвета, находятся на расстоянии 1 от центра координат с точки зрения соответствующей метрики расстояния. Запишем функцию для определения расстояния:

```
def distance(X1, X2, metric = 'euclidean', p = 2):
    if metric == 'euclidean':
        dist = np.sqrt(np.sum(np.square(X1 - X2).T,axis=0))
    if metric == 'cityblock':
        dist = np.sum(np.abs(X1 - X2).T,axis=0)
    if metric == 'Chebyshev':
        dist = np.max(np.abs(X1 - X2).T,axis=0)
    if metric == 'Minkowski':
        dist = np.power(np.sum(np.power(np.
abs(X1 - X2),p).T,axis=0),1/p)
    return dist
```

Функция работает как с векторами, так и с матрицами равной размерности.

Как правило, метрикой расстояния по умолчанию является евклидова метрика. Однако в разных задачах, особенно если анализируются категориальные переменные, могут хорошо проявлять себя и другие метрики. Выбор метрики расстояния тоже можно отнести к гиперпараметрам модели.

Кроме того, необходимо помнить о стандартизации или нормализации данных при использовании метрик расстояния. Иначе результаты вычисления

метрик могут быть некорректны, если используются данные, измеряющиеся в разных диапазонах.

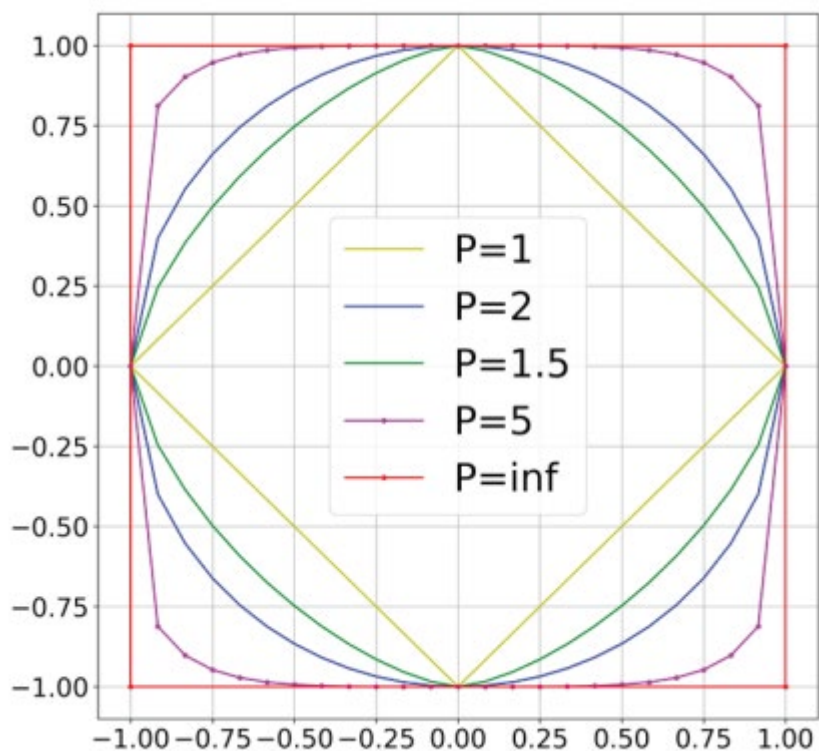


Рисунок 5.1. Эквидистантные фигуры для разных метрик расстояния

Одним из самых простых методов кластеризации является метод k -средних. Суть данного метода сводится к тому, чтобы найти заданное число кластеров (k) и их центры (так называемые центроиды) - такие, чтобы расстояние от центроидов до всех точек кластера было минимальны.

Прежде чем проводить кластеризацию, необходимо проинициализировать кластеры. Для этого выберем случайные индексы среди доступных в наборе данных:

```
def init_centroids(X, n_clusters):
    centroid_idx = np.random.randint(0, X.shape[0],
size = n_clusters)
    return X[centroid_idx,:]
```

Посмотрим, как это работает для двух кластеров. Проведем первую кластеризацию. Для этого возьмем каждый центроид и посчитаем расстояние

от него до всех записей набора данных. Индексы значений для каждого кластера выберем как индексы минимальных расстояний до соответствующего центроида. Таким образом, нулевой кластер будет включать те точки набора данных, в которых расстояние до нулевого центроида меньше, чем до первого центроида.

```
def predict(X, n_clusters, centroids, metric =
    'euclidean', p = 2):
    distances = np.zeros((X.shape[0], n_clusters))
    for i, centr in enumerate(centroids):
        distances[:, i] = distance(centr, X, metric, p)
    cluster_label = np.argmin(distances, axis = 1)
    return cluster_label
```

Посмотрим, как распределились результаты кластеризации после случайной инициализации центроидов (рис. 5.2). Получилось достаточно интересно: оба центроида оказались относительно близко друг к другу. Тем не менее это нулевая итерация алгоритма.

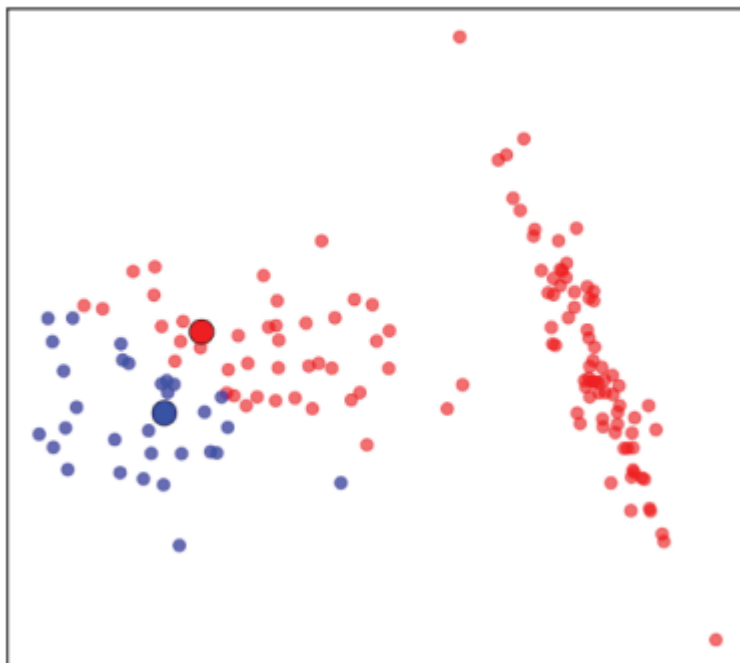


Рисунок 5.2. Результаты кластеризации после инициализации центроидов

Теперь выберем новые центроиды для каждого кластера как среднее значение по кластеру. В визуализации будут видны старые центроиды и новый

центроид с бóльшим радиусом (рис. 5.3). Видно, что центр для красного кластера значительно сместился.

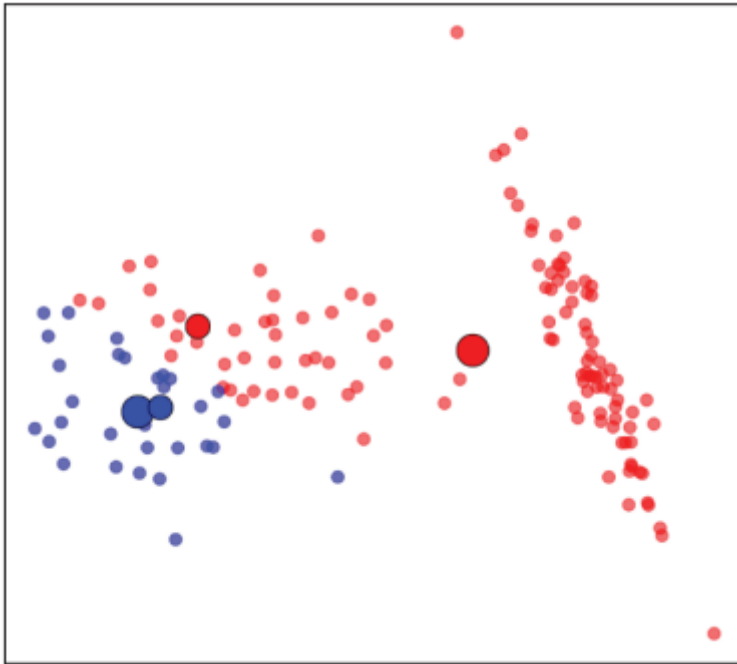


Рисунок 5.3. Результаты кластеризации после изменения центроидов

Рассчитаем относительное расстояние между старыми и новыми центроидами. Если расстояние между обновленными центроидами будет сравнительно небольшим, то есть центроиды перестанут менять позицию, то будем считать, что кластеризация закончена.

```
def delta_centroids(centroids,old_centroids,
metric = 'euclidean', p = 2):
    return (distance(centroids,old_centroids, metric, p)/
distance(old_centroids, np.mean(old_centroids), metric, p)).mean()
```

Для автоматизации процесса реализуем процедуру итерационной кластеризации. Укажем порог относительного изменения центроидов – tol. В конце процедуры выведем результирующий номер итерации и расстояние между последним изменением центроидов:

```

def fit(X, n_clusters, centroids, max_
iter=10, tol=0.01, metric = 'euclidean', p = 2):

    dcentr = np.inf

    for i in range(max_iter):

        old_centroids = np.copy(centroids)
        cluster_label=predict(X, n_clusters, centroids, metric, p)

        for k in range(n_clusters):
            c_idx = np.flatnonzero(cluster_label==k)
            centroids[k] = X[c_idx].mean(axis = 0)

        dcentr = delta_centroids(centroids,old_centroids, met-
ric, p)

        if dcentr<=tol:
            break

        print('Мы остановились на итерации:', i, ', относитель-
ное изменение центроидов: ',dcentr)

    return cluster_label

```

Проверим и визуализируем результаты. Буквально через пару итераций получим следующее распределение (рис. 5.4).

Теперь объединим все наши наработки в один класс KMeans. Центроиды на последней итерации хранятся в атрибуте `.centroids`. Для того чтобы получить предсказания номера кластера, необходимо воспользоваться методом `.fit_transform`. Кроме того, мы добавили в класс KMeans атрибут `.inertia` — сумму квадратов расстояния точек кластеров до соответствующих центроидов.

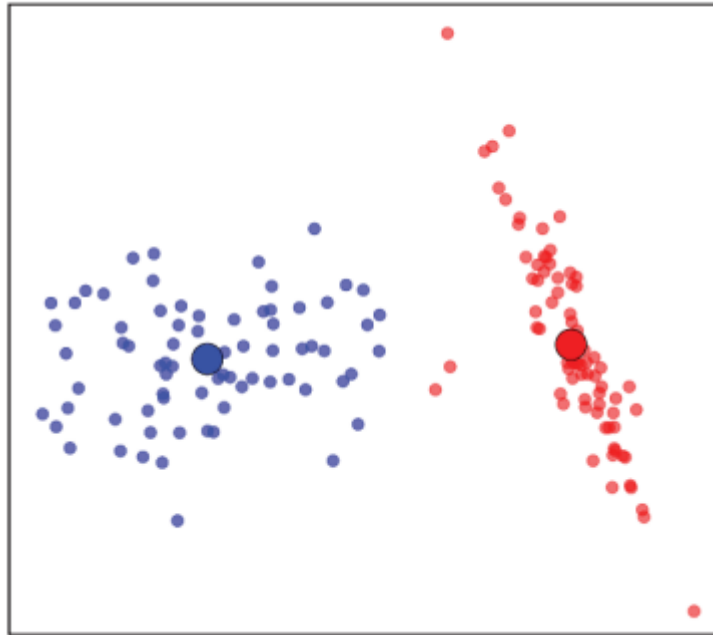


Рисунок 5.4 Результаты кластеризации после нескольких итераций

С помощью этого параметра можно попытаться ответить на вопрос, какое число кластеров оптимально для конкретных данных. Для этого можно использовать так называемый метод локтя (Elbow Method): проверяются различные количества кластеров и запоминаются конечные значения инерции для каждого числа кластеров, затем визуализируется зависимость (число кластеров, инерция). Как правило, получаются зависимости, подобные рис. 5.5.

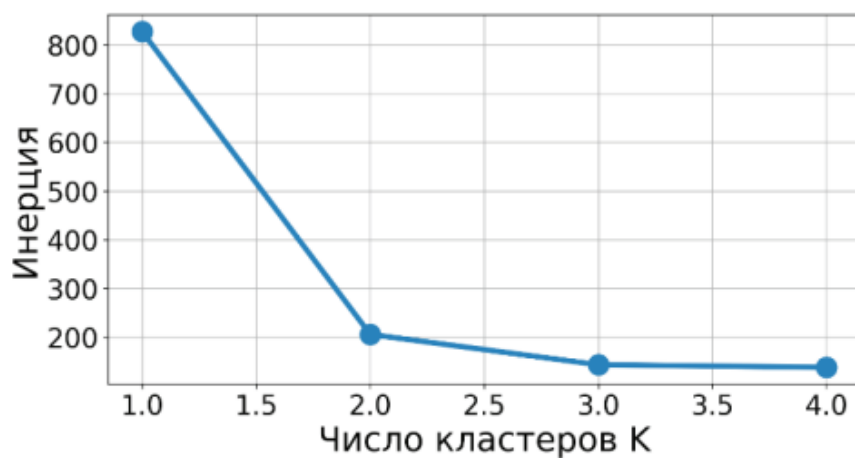


Рисунок 5.5 Визуализация метода локтя

Затем ищется такое число кластеров, которое напоминает перегиб согнутого локтя (в нашем случае это число $k = 2$). Большое количество кластеров неуместно: мы начинаем искусственно дробить большие кластеры на малые, нарушая целостную структуру данных (рис. 5.6). Метод k -средних можно применять не только к двумерным данным, но и к данным большей размерности. При этом рекомендуется пользоваться методами уменьшения размерности для возможности визуализации результата кластеризации.

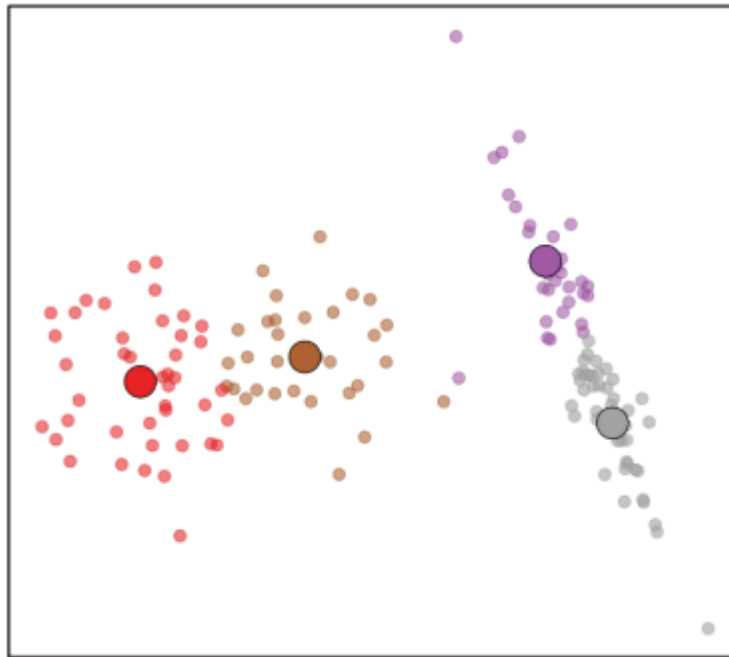


Рисунок 5.6. Кластеризация k -средних со слишком большим числом кластеров

В общем случае кластеризация применяется в контексте того, что истинные метки кластеров нам неизвестны. Именно поэтому эту задачу машинного обучения относят к задачам обучения без учителя. Однако иногда возникают тренировочные задачи, в которых метки известны или помимо числовых данных есть набор категориальных признаков, и требуется выяснить, связаны ли найденные кластеры с каким-то категориальным признаком

Индивидуальное задание

1. Студент самостоятельно выбирает набор данных и согласует свой выбор с преподавателем.
2. Сгенерируйте линейно разделимые данные с другими параметрами и проверьте, как работает алгоритм кластеризации k-средних на этих данных. Оцените оптимальное число кластеров по методу локтя.
3. Сгенерируйте данные, распределенные как знак инь-ян или концентрические круги, и проверьте, как работает алгоритм кластеризации k-средних на этих данных. Оцените оптимальное число кластеров по методу локтя.
4. Загрузите данные MNIST. Уменьшите размерность данных с использованием метода главных компонент. Примените кластеризацию k-средних. Оцените оптимальное число кластеров по методу локтя и связь кластеров с цифрами на изображениях.
5. Выполните кластеризацию для набора данных:
 - выполните кластеризацию для числовых признаков: используйте все числовые признаки, выполнив визуализацию в разных двумерных проекциях;
 - оцените оптимальное число кластеров по методу локтя;
 - оцените связь кластеров с категориальными признаками;
 - сравните результаты модели при использовании данных с применением метода главных компонент.
6. Сравните работу реализованных алгоритмов с функциями библиотеки `scikit-learn` – кластеризацией k-средних `sklearn.cluster.KMeans`.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.
4. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. . Оцените евклидово расстояние между векторами $x_1 \{2, 5, 3, 7\}$ и $x_2 \{2, 7, 1, 5\}$.
2. Оцените расстояние Чебышева между векторами $x_1 \{0, 10, 4, 9\}$ и $x_2 \{3, 7, 0, 2\}$.
3. Есть три центроида $c_1 \{1, 0, 0\}$, $c_2 \{0, 1, 1\}$, $c_3 \{1, 0, 1\}$ и точка x с координатами $\{2, 0, 2\}$. К какому кластеру следует отнести эту точку при использовании евклидовой метрики расстояния?
4. Как называется метод определения оптимального числа k (кластеров) с использованием анализа инерции?

Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [1–5].

ЗАКЛЮЧЕНИЕ

Учебное пособие (лабораторный практикум) по дисциплине «Неконтролируемое машинное обучение» для студентов направления 09.03.02 «Информационные системы и технологии». Пособие охватывает теоретические аспекты построения информационных систем на основе методов искусственного интеллекта, а также предлагает студентам практические рекомендации по разработке систем на основе методов искусственного интеллекта. Основное внимание уделяется теории обучения машин (машинное обучение, machine learning).

В пособии рассмотрены практические аспекты проектирования и разработки информационных систем для решения различных задач с использованием следующих подходов: линейных методов классификации, аппарата нейронных сетей, байесовских методов классификации, алгоритмов восстановления регрессии, алгоритмов логической классификации.

Многие задачи, возникающие в практических приложениях, не могут быть решены заранее известными методами или алгоритмами. Это происходит по той причине, что нам заранее не известны механизмы порождения исходных данных или же известная нам информация недостаточна для построения модели источника, генерирующего поступающие к нам данные. Как говорят, мы получаем данные из «черного ящика». В этих условиях ничего не остается, как только изучать доступную нам последовательность исходных данных и пытаться строить предсказания совершенствуя нашу схему в процессе предсказания. Подход, при котором прошлые данные или примеры используются для первоначального формирования и совершенствования схемы предсказания, называется методом машинного обучения (Machine Learning). Машинное обучение – чрезвычайно широкая и динамически развивающаяся область исследований, использующая огромное число теоретических и практических методов.

СПИСОК ЛИТЕРАТУРЫ

Список основной литературы

1. Уэс, Маккинли. Python и анализ данных Электронный ресурс / Маккинли Уэс ; пер. А. А. Слинкин. - Python и анализ данных, 2021-04-19. - Саратов : Профобразование, 2019. - 482 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0046-7, экземпляров неограниченно.

2. Сузи, Р.А. Язык программирования Python Электронный ресурс : учебное пособие / Р.А. Сузи. - Язык программирования Python, 2020-07-28. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2020. - 350 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-9556-0058-2, экземпляров неограниченно

Список дополнительной литературы

3. Стенли, Липпман. Язык программирования C++ Электронный ресурс : Полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. - Язык программирования C++, 2021-04-19. - Саратов : Профобразование, 2020. - 1104 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0136-5, экземпляров неограниченно

4. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2020. - 1773 с., экземпляров неограниченно

5. <https://archive.ics.uci.edu/ml/index.html> – Репозиторий наборов данных для машинного обучения (Центр машинного обучения и интеллектуальных систем).

6. <https://www.kaggle.com> – Портал и система проведения соревнований по проблемам анализа данных.

7. <https://www.mockaroo.com> – Сайт для генерации наборов данных.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

по организации и проведению самостоятельной работы
по дисциплине

«Неконтролируемое машинное обучение»

для студентов направления подготовки

09.03.02 «Информационные системы и технологии»

направленность (профиль)

**«Прикладное программирование в интеллектуальных информационных
системах»**

Ставрополь, 2026

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине «Неконтролируемое машинное обучение» направлена на формирование следующих **компетенций**:

Код, формулировка компетенции	Код, формулировка индикатора
ПК-3 Способен использовать интеллектуальные модели при решении задач профессиональной деятельности	ПК-3ид-3.1 – Демонстрирует знания теоретических основ функционирования интеллектуальных моделей и особенности их обучения и оптимизации

	ПК-3 _{ид-3.2} – Анализирует и применяет интеллектуальные методы согласно решаемой задаче
	ПК-3 _{ид-3.3} – Применяет инструментальные средства визуализации данных, в том числе больших данных
	ПК-3 _{ид-3.4} – Умеет настраивать параметры и гиперпараметры интеллектуальных моделей для достижения высокой эффективности обучения и функционирования
	ПК-3 _{ид-3.5} – Применяет инструментальные средства интеллектуальной обработки данных различной природы (текстовые, изображения, видео, аудио)
ПК-6 Способен проектировать информационные подсистемы обработки данных	ПК-6 _{ид-6.1} – Демонстрирует знания в области архитектуры систем управления базами данных, типологии баз данных, методов оптимизации баз данных
	ПК-6 _{ид-6.2} – Применяет инструментальные средства для проектирования баз данных различного типа, в том числе распределенных
	ПК-6 _{ид-6.3} – Владеет методикой оптимизации систем хранения данных на основе интерпретируемых критериев эффективности
	ПК-6 _{ид-6.4} – Анализирует предметную область в рамках решения профессиональных задач и проектирует архитектуру систем обработки данных на основании типов данных, конфигурации потоков данных

2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;

- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
6 семестр					
ПК-3, ПК-6	Самостоятельное изучение литературы и источников	Собеседование	12	4	16
ПК-3, ПК-6	Подготовка лабораторным занятиям	Защита ЛР	48	12	60
Итого за 6 семестр			60	16	76
Итого			60	16	76

4. Порядок выполнения самостоятельной работы студентом

4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и

уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста**:

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;

2. Выделите главное, составьте план;

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

4.4. Методические рекомендации по написанию научных текстов (докладов, докладов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т. п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

Требования к написанию доклада

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

Структура доклада:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.
- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.
- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса
- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

- Приложение (при необходимости).

Требования к оформлению:

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- доклад должен быть представлен в сброшюрованном виде.

Порядок защиты доклада:

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

Оценка доклада

Доклад оценивается по следующим критериям:

- соблюдение требований к его оформлению;
- необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;
- умение студента свободно излагать основные идеи, отраженные в докладе;

- способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

4.5. Методические рекомендации по выполнению исследовательских проектов

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Требования по выполнению и оформлению проекта

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

Критерии оценки работы участника проекта.

Для каждого из участников проекта оцениваются:

- профессиональные теоретические знания в соответствующей области;
- умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;
- умение работать с техническими средствами;
- умение пользоваться соответствующими выполняемому проекту информационными технологиями;

- умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;
- умение работать в команде;
- умение публично представлять результаты собственной деятельности;
- коммуникабельность, инициативность, творческие способности.

Критерии выставления оценки участникам проекта

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	Технические средства и информационные технологии освоены и использованы для реализации проекта полностью	Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности.	Проект представлен полностью и в срок.
«Хорошо»	Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий	Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	Проект представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетворительно»	Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующим и информационным и технологиями	Студент выполнил большую часть возложенной на него работы.	Проект сдан со значительным опозданием (более недели) и не полностью

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.	Навыков работы с техническими средствами нет, информационные технологии не освоены	Студент практически не работал, не выполнил свои задачи или выполнил лишь отдельные не существенные поручения в групповом проекте.	Проект не сдан.

Студенты должны: защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (если проект групповой).

4.6. Методические рекомендации по подготовке к экзаменам и зачетам

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неустойчивые занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала

было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

Список литературы для выполнения СРС

Перечень основной литературы:

1. Уэс, Маккинли. Python и анализ данных Электронный ресурс / Маккинли Уэс ; пер. А. А. Слинкин. - Python и анализ данных, 2021-04-19. - Саратов : Профобразование, 2019. - 482 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0046-7, экземпляров неограниченно.

2. Сузи, Р.А. Язык программирования Python Электронный ресурс : учебное пособие / Р.А. Сузи. - Язык программирования Python, 2020-07-28. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2020. - 350 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-9556-0058-2, экземпляров неограниченно

Перечень дополнительной литературы:

3. Стенли, Липпман. Язык программирования C++ Электронный ресурс : Полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. - Язык программирования C++, 2021-04-19. - Саратов : Профобразование, 2020. - 1104 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0136-5, экземпляров неограниченно

4. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2020. - 1773 с., экземпляров неограниченно

Методическая литература:

1. Методические рекомендации для самостоятельной работы студентов по дисциплине «Неконтролируемое машинное обучение»
2. Методические указания к лабораторным работам по дисциплине «Неконтролируемое машинное обучение»

Интернет-ресурсы:

1. <http://el.ncfu.ru/> – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Неконтролируемое машинное обучение».
2. <http://www.intuit.ru> – Интернет-Университет Компьютерных технологий.