Автономная некоммерческая образовательная организация
высшего образования
«Воронежский экономико-правовой институт»
(АНОО ВО «ВЭПИ»)

АНОО ВО "ВОРОНЕЖСКИЙ ЭКОНОМИКО-ПРАВОВОЙ ИНСТИТУТ", АНОО ВО "ВЭПИ", ВЭПИ

Подписано цифровой подписью: АНОО ВО "ВОРОНЕЖСКИЙ ЭКОНОМИКО-ПРАВОВОЙ ИНСТИТУТ", АНОО ВО "ВЭПИ", ВЭПИ
Дата: 26.09.2023 10:16:20 +03'00'

УТВЕРЖДАЮ
Проректор
по учебно-методической работе
_____ А.Ю. Жильников
«__» _____ 2023 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Б1.О.27 Системы искусственного интеллекта в профессиональной деятельности
(наименование дисциплины (модуля)

09.03.03 Прикладная информатика
(код и наименование направления подготовки)

Направленность (профиль) Прикладная информатика в экономике
(наименование направленности (профиля))

Квалификация выпускника Бакалавр
(наименование квалификации)

Форма обучения Очная, заочная
(очная, заочная)

Рекомендован к использованию Филиалами АНОО ВО «ВЭПИ»

Воронеж 2023

Методические рекомендации по выполнению лабораторных работ по дисциплине (модулю) рассмотрены и одобрены на заседании кафедры прикладной информатики.

Протокол от «01» сентября 2023г. № 1

Заведующий кафедрой М.С. Агафонова

Разработчики:

Профессор М.С. Агафонова

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1: Методы работы с таблицами в Python. Агрегация и визуализация данных. Проведение первичного анализа данных.

В данной лабораторной работе мы рассмотрим методы работы с таблицами в Python, включая агрегацию и визуализацию данных. Мы проведем первичный анализ данных, используя популярные библиотеки, такие как Pandas и Matplotlib. Работа включает в себя практические задания, которые помогут закрепить полученные знания и навыки.

Цели работы

1. Ознакомиться с библиотекой Pandas для работы с таблицами.
2. Научиться выполнять агрегацию данных.
3. Изучить методы визуализации данных с помощью Matplotlib.
4. Провести первичный анализ данных на примере реального набора данных.

Оборудование и программное обеспечение

- Компьютер с установленной операционной системой Windows, macOS или Linux.
- Установленный Python версии 3.6 и выше.
- Установленные библиотеки: Pandas, Matplotlib, NumPy.

Задания

Задание 1: Установка необходимых библиотек

Убедитесь, что у вас установлены необходимые библиотеки. Если нет, установите их с помощью pip:

```
pip install pandas matplotlib numpy
```

Задание 2: Загрузка данных

Скачайте набор данных (например, CSV-файл) и загрузите его в Pandas DataFrame. Пример кода:

```
import pandas as pd
data = pd.read_csv('path_to_your_file.csv')
print(data.head())
```

Задание 3: Первичный анализ данных

Проведите первичный анализ данных, используя следующие методы:

- data.info() — получение информации о DataFrame.
- data.describe() — получение статистических характеристик.
- data.isnull().sum() — проверка на наличие пропущенных значений.

Задание 4: Агрегация данных

Используйте методы агрегации для анализа данных. Например, сгруппируйте данные по определенному столбцу и вычислите среднее значение:

```
aggregated_data = data.groupby('column_name').mean()
```

```
print(aggregated_data)
```

Задание 5: Визуализация данных

Создайте графики для визуализации данных с помощью Matplotlib. Например, построим гистограмму:

```
import matplotlib.pyplot as plt
plt.hist(data['column_name'], bins=10)
plt.title('Гистограмма')
plt.xlabel('Значение')
plt.ylabel('Частота')
plt.show()
```

Задание 6: Интерпретация результатов

На основе полученных графиков и агрегированных данных сделайте выводы о ваших данных. Обсудите, какие интересные закономерности или аномалии вы заметили.

Заключение

В ходе лабораторной работы вы ознакомились с основными методами работы с таблицами в Python, научились выполнять агрегацию и визуализацию данных, а также провели первичный анализ данных. Эти навыки являются основополагающими для дальнейшего изучения анализа данных и машинного обучения.

Вопросы:

1. Методы машинного обучения.
2. Роль предобработка данных.

Содержание отчета:

1) цель работы;
2) задание на лабораторную работу для своего варианта;
3) алгоритм решаемого задания с необходимыми пояснениями;
4) выводы по работе.

Лабораторная работа № 2: Использование и сравнение алгоритмов классификации: kNN, решающие деревья и их ансамбли, логистическая регрессия.

В данной лабораторной работе мы рассмотрим три популярных алгоритма классификации: k ближайших соседей (kNN), решающие деревья и их ансамбли, а также логистическую регрессию. Мы проведем практические эксперименты, чтобы сравнить их эффективность на одном и том же наборе данных. В результате работы студенты смогут лучше понять принципы работы этих алгоритмов, их преимущества и недостатки, а также научатся применять их для решения задач классификации.

Цели работы

1. Изучить алгоритмы kNN, решающие деревья и логистическую регрессию.
2. Научиться применять данные алгоритмы на практике.
3. Сравнить эффективность алгоритмов на одном наборе данных.
4. Проанализировать результаты и сделать выводы.

Задания

Задание 1: Подготовка данных

1. Выберите набор данных для классификации. Рекомендуется использовать открытые наборы данных, такие как Iris, Wine или Titanic.
2. Загрузите данные и проведите предварительную обработку:
    - Заполните пропуски (если есть).
    - Преобразуйте категориальные переменные в числовые (если необходимо).
    - Разделите данные на обучающую и тестовую выборки (например, 80% на обучение и 20% на тестирование).

Задание 2: Реализация алгоритма kNN

1. Импортируйте необходимые библиотеки (например, sklearn).
2. Реализуйте алгоритм kNN с использованием библиотеки sklearn.
3. Обучите модель на обучающей выборке и сделайте предсказания на тестовой выборке.
4. Оцените качество модели с помощью метрик, таких как точность (accuracy), полнота (recall) и F1-мера.

Задание 3: Реализация решающих деревьев

1. Реализуйте алгоритм решающих деревьев с использованием библиотеки sklearn.
2. Обучите модель на обучающей выборке и сделайте предсказания на тестовой выборке.
3. Оцените качество модели с помощью тех же метрик, что и в предыдущем задании.

Задание 4: Реализация ансамблей решающих деревьев

1. Реализуйте алгоритм случайного леса (Random Forest) и/или градиентного бустинга (Gradient Boosting) с использованием библиотеки sklearn.
2. Обучите модель на обучающей выборке и сделайте предсказания на тестовой выборке.
3. Оцените качество модели с помощью тех же метрик.

Задание 5: Реализация логистической регрессии

1. Реализуйте алгоритм логистической регрессии с использованием библиотеки sklearn.
2. Обучите модель на обучающей выборке и сделайте предсказания на тестовой выборке.
3. Оцените качество модели с помощью тех же метрик.

Задание 6: Сравнение результатов

1. Сравните результаты всех реализованных моделей по метрикам точности, полноты и F1-меры.
2. Постройте графики для наглядного сравнения (например, столбчатые диаграммы).
3. Проанализируйте, какой алгоритм показал наилучшие результаты и почему.

Задание 7: Выводы

1. Напишите краткий отчет о проделанной работе.
2. Укажите, какие алгоритмы показали лучшие результаты и в каких случаях каждый из них может быть предпочтительнее.
3. Обсудите возможные улучшения и дальнейшие шаги для исследования.

Заключение

В ходе выполнения лабораторной работы студенты получат практический опыт работы с различными алгоритмами классификации, научатся их применять и сравнивать. Это поможет лучше понять, как выбирать подходящий алгоритм для решения конкретных задач в области машинного обучения.

Вопросы:

1. Влияние выбора параметра k на качество классификации при использовании алгоритма k-ближайших соседей.
2. Методы улучшения производительности алгоритма k-ближайших соседей для работы с большими объемами данных.

Содержание отчета:

1) цель работы;
2) задание на лабораторную работу для своего варианта;
3) алгоритм решаемого задания с необходимыми пояснениями;
4) выводы по работе.

Лабораторная работа № 3: Использование и оценка алгоритмов регрессии. Подбор оптимальных параметров регрессии.

В данной лабораторной работе рассматриваются основные алгоритмы регрессии, их применение и оценка качества. Учащиеся познакомятся с методами подбора оптимальных параметров регрессии, а также научатся использовать различные метрики для оценки производительности моделей. Работа включает в себя практические задания, которые помогут закрепить полученные знания.

Регрессия — это один из основных методов анализа данных, который используется для предсказания значений зависимой переменной на основе значений одной или нескольких независимых переменных. В этой лабораторной работе мы будем использовать Python и библиотеку scikit-learn для реализации алгоритмов регрессии.

Задание 1: Подготовка данных

1. Загрузите набор данных (например, boston housing dataset или diabetes dataset) из библиотеки sklearn.datasets.
2. Разделите данные на обучающую и тестовую выборки (например, 80% на обучение и 20% на тестирование).
3. Проведите предварительный анализ данных: проверьте наличие пропусков, визуализируйте распределение целевой переменной.

Задание 2: Реализация алгоритмов регрессии

1. Реализуйте следующие алгоритмы регрессии:
    - Линейная регрессия
    - Регрессия с использованием дерева решений
    - Регрессия с использованием случайного леса
    - Регрессия с использованием градиентного бустинга
2. Для каждой модели:
    - Обучите модель на обучающей выборке.
    - Сделайте предсказания на тестовой выборке.

Задание 3: Оценка качества моделей

1. Используйте следующие метрики для оценки качества моделей:
    - Средняя абсолютная ошибка (MAE)
    - Средняя квадратичная ошибка (MSE)
    - Коэффициент детерминации (R^2)
2. Сравните результаты для всех реализованных моделей и выберите наилучшую.

Задание 4: Подбор оптимальных параметров

1. Используйте метод кросс-валидации для подбора оптимальных гиперпараметров для модели, которая показала наилучшие результаты на предыдущем этапе.

2. Примените GridSearchCV или RandomizedSearchCV для поиска оптимальных параметров.

3. Обучите модель с оптимальными параметрами и оцените её качество на тестовой выборке.

Задание 5: Визуализация результатов

1. Постройте графики, показывающие:
   - Сравнение предсказанных и фактических значений для наилучшей модели.
   - Важность признаков для модели, использующей случайный лес или градиентный бустинг.
2. Напишите краткий вывод о том, какие факторы оказали наибольшее влияние на предсказания.

Заключение

В данной лабораторной работе были изучены основные алгоритмы регрессии, методы их оценки и подбора параметров. Учащиеся получили практический опыт работы с реальными данными и научились применять различные метрики для оценки качества моделей. Результаты работы помогут лучше понять, как выбирать и настраивать модели регрессии для решения практических задач.

Вопросы:

1. Упрощение выбора наиболее подходящей модели для конкретного набора данных при использовании различных метрик оценки регрессии, таких как MSE, MAE и R^2-коэффициент детерминации.

2. Особенности применения каждой из метрик MSE, MAE и R^2.

Содержание отчета:

1) цель работы;
2) задание на лабораторную работу для своего варианта;
3) алгоритм решаемого задания с необходимыми пояснениями;
4) выводы по работе.

Лабораторная работа № 4: Оптимизационные задачи и их решения. Подбор гиперпараметров алгоритма с помощью методов оптимизации

Методы безградиентной оптимизации

Данная лабораторная работа посвящена изучению методов безградиентной оптимизации, включая случайный поиск, метод "hill climb", отжиг и генетические алгоритмы. В ходе выполнения заданий студенты познакомятся с основами каждого метода, реализуют их на практике и сравнят эффективность различных подходов к оптимизации. Работа направлена на развитие навыков программирования и анализа алгоритмов.

Задание 1: Случайный поиск

Цель

Изучить метод случайного поиска и его применение для нахождения экстремума функции.

Описание

1. Выберите функцию для оптимизации (например, (f(x) = x^2 - 4x + 4)).
2. Реализуйте алгоритм случайного поиска:
    - Генерируйте случайные значения (x) в заданном диапазоне.
    - Вычисляйте значение функции для каждого (x).
    - Сохраняйте лучшее найденное значение и соответствующее (x).
3. Проведите 1000 итераций поиска.
4. Постройте график зависимости значений функции от итераций.

Ожидаемый результат

Получить график и значение, при котором функция достигает минимума.

Задание 2: Метод "hill climb"

Цель

Изучить метод "hill climb" и его применение для нахождения локального экстремума функции.

Описание

1. Используйте ту же функцию, что и в задании 1.
2. Реализуйте алгоритм "hill climb":
    - Начните с произвольного значения (x).
    - На каждой итерации изменяйте (x) на небольшое значение (например, (\Delta x = 0.1)).
    - Если новое значение функции меньше текущего, обновите (x).
    - Повторяйте процесс, пока не достигнете локального минимума.
3. Постройте график изменения (x) и значений функции.

Ожидаемый результат

Получить значение локального минимума и график изменения функции.

Задание 3: Метод отжига

Цель

Изучить метод отжига и его применение для нахождения глобального экстремума функции.

Описание

1. Используйте ту же функцию, что и в предыдущих заданиях.
2. Реализуйте алгоритм отжига:
    - Начните с произвольного значения (x) и начальной температуры.
    - На каждой итерации генерируйте новое значение (x') в окрестности текущего (x).
    - Вычисляйте разницу значений функции (\Delta f = f(x') - f(x)).
    - Если (\Delta f < 0), переходите к (x'); если (\Delta f \geq 0), переходите с вероятностью, зависящей от температуры.
    - Постепенно снижайте температуру.
3. Постройте график изменения температуры и значений функции.

Ожидаемый результат

Получить значение глобального минимума и график изменения функции.

Задание 4: Генетические алгоритмы

Цель

Изучить генетические алгоритмы и их применение для нахождения оптимальных решений.

Описание

1. Используйте ту же функцию, что и в предыдущих заданиях.
2. Реализуйте генетический алгоритм:
    - Создайте начальную популяцию случайных значений (x).
    - Определите функцию приспособленности (например, значение функции).
    - Реализуйте операции селекции, кроссовера и мутации.
    - Повторяйте процесс на протяжении нескольких поколений.
3. Постройте график изменения значений функции по поколениям.

Ожидаемый результат

Получить значение оптимума и график изменения функции.

Заключение

В результате выполнения лабораторной работы студенты должны освоить основные методы безградиентной оптимизации, реализовать их на практике и проанализировать полученные результаты. Это позволит лучше понять принципы работы алгоритмов и их применение в задачах оптимизации.

Вопросы:

1. Опишите основные принципы безградиентной оптимизации. В чем ее отличие от градиентного спуска?

2. Какие задачи можно решать с помощью безградиентной оптимизации? Приведите примеры.

3. Какие преимущества и недостатки у методов безградиентной оптимизации? Сравните их между собой.

4. Как выбрать оптимальный метод безградиентной оптимизации для конкретной задачи? Какие факторы нужно учитывать?

5. Что такое пространство поиска и целевая функция? Как они связаны с безградиентной оптимизацией?

Содержание отчета:

1) цель работы;

2) задание на лабораторную работу для своего варианта;

3) алгоритм решаемого задания с необходимыми пояснениями;

4) выводы по работе.

Лабораторная работа № 5: Классификация изображений и трансферное обучение

Данная лабораторная работа посвящена изучению методов классификации изображений с использованием трансферного обучения. Студенты познакомятся с основами работы с нейронными сетями, научатся применять предобученные модели для решения задач классификации, а также освоят практические навыки работы с библиотеками машинного обучения, такими как TensorFlow и Keras. В ходе выполнения заданий студенты будут работать с реальными наборами данных и оценивать качество своих моделей.

Задание 1: Введение в классификацию изображений

1. Изучите основные понятия, связанные с классификацией изображений. Определите, что такое классификация, какие существуют типы задач (бинарная и многоклассовая классификация).
2. Ознакомьтесь с архитектурами нейронных сетей, используемыми для классификации изображений (например, CNN - сверточные нейронные сети).
3. Прочитайте о трансферном обучении и его преимуществах. Объясните, как оно помогает в задачах классификации изображений.

Задание 2: Подготовка окружения

1. Установите необходимые библиотеки: TensorFlow, Keras, NumPy, Matplotlib.
2. pip install tensorflow keras numpy matplotlib
3. Создайте виртуальное окружение для работы над проектом.

Задание 3: Загрузка и предобработка данных

1. Выберите набор данных для классификации изображений (например, CIFAR-10, MNIST или собственный набор данных).
2. Загрузите данные и выполните их предобработку:
    - Измените размер изображений.
    - Нормализуйте пиксели изображений.
    - Разделите данные на обучающую и тестовую выборки.

Задание 4: Использование предобученной модели

1. Выберите предобученную модель (например, VGG16, ResNet50, InceptionV3) из библиотеки Keras.
2. Загрузите модель без верхнего слоя (классификатора) и добавьте свой собственный классификатор.
3. Обучите модель на своем наборе данных, используя трансферное обучение. Настройте параметры обучения (количество эпох, размер батча и т.д.).

Задание 5: Оценка модели

1. После завершения обучения оцените качество модели на тестовой выборке.

2. Используйте метрики, такие как точность (accuracy), полнота (recall) и F1-мера для оценки производительности модели.
3. Постройте графики потерь и точности для обучающей и тестовой выборок.

Задание 6: Визуализация результатов

1. Выберите несколько изображений из тестовой выборки и визуализируйте их с предсказанными метками.
2. Постройте матрицу путаницы для анализа ошибок классификации.

Заключение

1. Напишите краткий отчет о проделанной работе, в котором отразите:
   - Цели и задачи лабораторной работы.
   - Использованные методы и подходы.
   - Результаты и выводы по выполненным заданиям.
   - Возможные улучшения и дальнейшие направления работы.

Вопросы:

1. Применение архитектуры нейронных сетей для работы с изображениями.
2. Технологии передачи знаний между нейронными сетями.

Содержание отчета:

1) цель работы;
2) задание на лабораторную работу для своего варианта;
3) алгоритм решаемого задания с необходимыми пояснениями;
4) выводы по работе.

Лабораторная работа № 6: Работа с текстами и их векторными представлениями

Данная лабораторная работа посвящена изучению методов работы с текстами и их векторными представлениями. В ходе выполнения заданий студенты познакомятся с основными концепциями обработки естественного языка (NLP), а также научатся использовать различные алгоритмы для преобразования текстов в векторные формы. Работа включает в себя практические задания, которые помогут закрепить теоретические знания и развить навыки работы с текстовыми данными.

Цели и задачи лабораторной работы

1. Изучить основные методы обработки текстов.
2. Ознакомиться с векторными представлениями текстов.
3. Научиться применять алгоритмы для преобразования текстов в векторы.
4. Провести анализ текстовых данных с использованием векторных представлений.

Задание 1: Предварительная обработка текстов

Цель

Научиться очищать и подготавливать текстовые данные для дальнейшей обработки.

Описание

1. Выберите набор текстовых данных (например, статьи, отзывы, блоги).
2. Выполните следующие шаги:
    - Приведение текста к нижнему регистру.
    - Удаление пунктуации и специальных символов.
    - Токенизация (разделение текста на отдельные слова).
    - Удаление стоп-слов (например, "и", "в", "на").
    - Лемматизация или стемминг (приведение слов к их базовой форме).

Ожидаемый результат

Подготовленный набор текстов, готовый к дальнейшему анализу.

Задание 2: Векторизация текстов

Цель

Научиться преобразовывать тексты в векторные представления.

Описание

1. Используйте метод "мешка слов" (Bag of Words) для векторизации текстов:
    - Создайте словарь всех уникальных слов в вашем наборе текстов.
    - Для каждого текста создайте вектор, где каждое значение соответствует количеству вхождений слова из словаря.

2. Примените метод TF-IDF (Term Frequency-Inverse Document Frequency):
    - Рассчитайте TF-IDF для каждого слова в каждом тексте.
    - Создайте векторное представление на основе значений TF-IDF.

Ожидаемый результат

Векторные представления текстов в формате матрицы, где строки соответствуют текстам, а столбцы - словам из словаря.

Задание 3: Кластеризация текстов

Цель

Научиться группировать тексты на основе их векторных представлений.

Описание

1. Используйте алгоритм K-средних для кластеризации векторных представлений текстов:
    - Определите количество кластеров (k).
    - Примените алгоритм K-средних к вашей матрице векторных представлений.
    - Проанализируйте полученные кластеры и их содержание.
2. Визуализируйте результаты кластеризации с помощью метода t-SNE или PCA.

Ожидаемый результат

Группы текстов, сгруппированные по схожести, и визуализация кластеров.

Задание 4: Анализ результатов

Цель

Научиться анализировать и интерпретировать результаты обработки текстов.

Описание

1. Проанализируйте полученные векторные представления и кластеры:
    - Определите, какие слова или темы наиболее характерны для каждого кластера.
    - Обсудите, насколько хорошо тексты были сгруппированы.
2. Напишите отчет о проделанной работе, включая:
    - Описание использованных методов.
    - Результаты векторизации и кластеризации.
    - Выводы о качестве обработки текстов.

Ожидаемый результат

Отчет, содержащий анализ и интерпретацию результатов лабораторной работы.

Заключение

В ходе выполнения данной лабораторной работы студенты получат практические навыки работы с текстами и их векторными представлениями. Освоенные методы обработки текстов и анализа данных помогут в дальнейшем изучении и применении технологий обработки естественного языка.

Вопросы:

Общие вопросы:

1. Что такое обучение с подкреплением (Reinforcement Learning)? В чем его основная идея?
2. Какие элементы составляют систему обучения с подкреплением? Опишите роли агента, среды, состояния, действия и награды.
3. Что такое функция ценности состояния (Value function)? Как она связана с ожидаемой наградой агента?
4. Что такое функция качества действия (Q-function)? Чем она отличается от функции ценности состояния?
5. Как оптимизировать стратегию агента с помощью функций ценности и качества? Какие критерии используются для оценки качества стратегии?
6. В чем заключается задача обучения с подкреплением? Какую цель агент пытается достичь в процессе обучения?
7. Какие типы задач можно решать с помощью обучения с подкреплением? Приведите примеры.

Q-обучение:

1. Как работает алгоритм Q-обучения? Опишите его шаги и формулу обновления Q-функции.
2. Что такое скорость обучения (α) и коэффициент скидки (γ)? Как они влияют на процесс обучения?
3. Каковы преимущества Q-обучения? Какие проблемы решает Q-обучение?
4. Какие недостатки у Q-обучения? Когда Q-обучение может быть неэффективным?
5. Что такое SARSA? Чем SARSA отличается от Q-обучения?
6. Что такое Deep Q-learning? Как используются нейронные сети в Deep Q-learning?

Дополнительные вопросы:

1. Что такое марковское свойство? Как оно связано с обучением с подкреплением?
2. Как можно решить проблему "проклятия размерности" в обучении с подкреплением?
3. Какие методы используются для изучения модели среды в обучении с подкреплением?
4. Какие еще алгоритмы обучения с подкреплением существуют кроме Q-

обучения? (например, актор-критик, обучение с моделью)

5. Каковы принципы использования обучения с подкреплением в контексте искусственного интеллекта?

6. Как обучение с подкреплением может применяться для решения проблем в разных областях (игровые приложения, робототехника, финансы)?

7. Каковы перспективы развития обучения с подкреплением? Какие проблемы и направления исследований актуальны в этой области?

Содержание отчета:

1) цель работы;

2) задание на лабораторную работу для своего варианта;

3) алгоритм решаемого задания с необходимыми пояснениями;

4) выводы по работе.