Практическое задание № 5

ЛИНЕЙНАЯ РЕГРЕССИЯ

Цель работы: получить практику анализа статистических данных с использованием линейной регрессии с одной переменной и с множеством переменных.

Содержание задания

1. Общие сведения

1. Ознакомиться с презентацией о линейной регрессии.

2. Установить необходимое программное обеспечение. При выполнении задания наверняка понадобятся Python 3, NumPy и Matplotlib.

3. Ознакомиться с содержимым папки с заданием, которая включает в себя файлы, представленные ниже.

main_one.py – «основной» модуль, необходимый для выполнения первой части задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

main_multi.py – «основной» модуль, необходимый для выполнения второй части задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

data1.txt – база данных для выполнения первой части задания.

data2.txt – база данных для выполнения второй части задания.

plotData.py – модуль, содержащий функцию plotData, которая необходима для визуализации данных.

computeCost.py – модуль, содержащий функцию computeCost, которая необходима для вычисления значения стоимостной функции линейной регрессии.

gradientDescent.py – модуль, содержащий функцию gradientDescent, которая необходима для выполнения градиентного спуска с целью поиска параметров модели линейной регрессии.

featureNormalize.py – модуль, содержащий функцию featureNormalize, которая необходима для нормализации признаков.

normalEqn.py – модуль, содержащий функцию normalEqn, которая необходима для поиска параметров модели линейной регрессии с использованием нормальных уравнений.

4. Выполнить первую часть задания, связанную с реализацией и исследованием линейной регрессии с одной переменной.

5. Выполнить вторую часть задания, связанную с реализацией и исследованием линейной регрессии со множеством переменных.

2. Линейная регрессия с одной переменной

При выполнении данного задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: plotData, computeCost, gradientDescent.

1. При решении любой задачи с использованием инструментов машинного обучения важным является понимание структуры анализируемых данных и их визуализация в случае возможности. В первой части задания предлагается использовать базу данных из файла data1.txt. Данные представляют собой множество объектов, описываемых одним признаком (численность населения в некотором городе) и меткой (прибыль, которую можно получить при продаже определенного товара в городе с соответствующей численностью населения). Завершите программный код в модуле plotData.py, который позволит выполнять визуализацию данных. Завершение модуля подразумевает под собой написание строчек программного кода, которые позволят вызвать функцию из соответствующего модуля в файле main_one.py, позволяя решить определенный кусок настоящего задания. Например, в данном случае завершенный программный код будет выглядеть так, как представлено на рис. 1.

После завершения каждого блока кода интерпретируйте файл main_one.py с целью проверки правильности работы соответствующей части задания. Результат визуализации данных с использованием функции plotData представлен на рис. 2. В случае успешной интерпретации программного кода разрешается перейти к следующему пункту задания.

```python
import matplotlib.pyplot as plt
import numpy as np

def plotData(data):
    """
        Функция позволяет выполнить визуализацию данных в декартовой
        системе координат с подписанным осями (численность населения
        и прибыль)
    """

    # ==================== Ваш код здесь ====================
    # Инструкция: визуализируйте данные с использованием функций
    # figure и plot. Подпишите оси с использованием функций xlabel
    # и ylabel, предполагая, что аргументами этих функций являются
    # численность населения по x и прибыль по y

    plt.figure()                                                          # создание нового окна
    plt.plot(data[:, 0], data[:, 1], 'rx', markersize = 5, label = 'Тренировочные данные') # построение точек данных
    plt.legend(loc = 'upper right', shadow = True, fontsize = 12, numpoints = 1)  # построение легенды
    plt.xlabel('Численность населения в 10,000')                          # обозначение оси x
    plt.ylabel('Прибыль в $10,000')                                       # обозначение оси y
    plt.grid()                                                            # создание координатной сетки

    # =========================================================
```

Рис. 1. Завершенный программный код для функции plotData

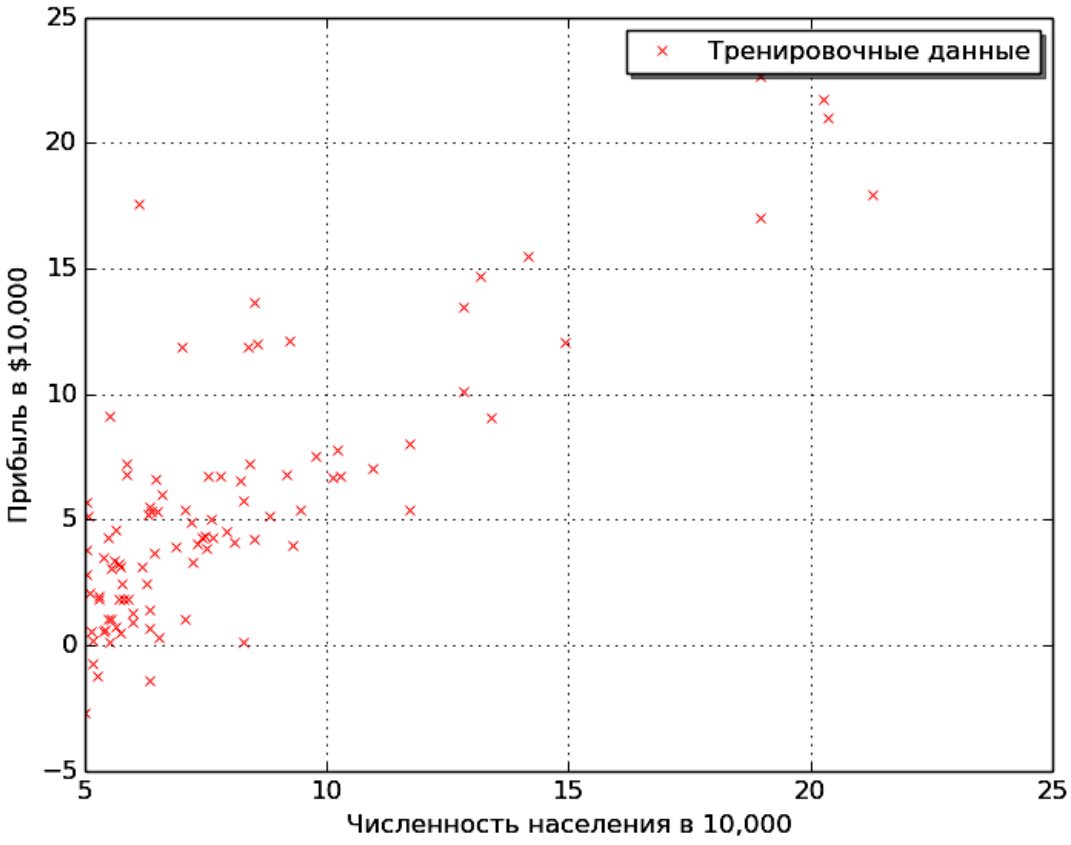

Рис. 2. Результат визуализации тренировочных данных

2. Завершите программный код в модуле computeCost.py, который позволит вычислить значение стоимостной функции для линейной регрессии. Формулы, описывающие ее вычисление, представлены в презентации. При

выполнении данной части задания могут понадобиться функции из библиотеки NumPy, представленные ниже.

dot – позволяет вычислить матричное произведение для двумерных массивов и скалярное произведение для одномерных массивов (без комплексного сопряжения).

sum – позволяет вычислить сумму элементов вдоль определенной размерности двумерного массива и сумму всех элементов для одномерного массива. Также полезным может оказаться оператор поэлементное возведение компонентов двумерного и одномерного массивов в квадрат: ** 2.

3. Завершите программный код в модуле gradientDescent.py, который позволит выполнить алгоритм градиентного спуска с целью обучения параметров модели линейной регрессии. Формулы, описывающие реализацию градиентного спуска, представлены в презентации. При выполнении данной части задания могут понадобиться следующие функции из библиотеки NumPy: dot и transpose.

transpose – позволяет выполнить транспонирование массива. Для одномерного массива данная функция не оказывает никакого действия, а для двумерного массива использование функции соответствует обычному матричному транспонированию.

4. После завершения предыдущих пунктов выполните предсказание прибыли от продажи товара в городах с численностью населения 35,000 и 70,000. При выполнении задания обратите внимание на то, что в матрице объекты-признаки, сформированной в файле main_one.py после загрузки базы данных из data1.txt, единственный признак объекта, описывающий численность населения в городе, является нормированным на значение 10,000.

3. Линейная регрессия со множеством переменных

При выполнении данного задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: featureNormalize, normalEqn.

1. Во второй части задания предлагается использовать базу данных из файла data2.txt. В этом случае данные представляют собой множество объектов, описываемых двумя признаками (площадь помещения в квадратных футах и число комнат в нем) и меткой (стоимость жилья для заданной площади и

числа комнат). Завершите программный код в модуле featureNormalize.py, который позволит выполнить нормализацию признаков на их математическое ожидание и среднеквадратическое отклонение. Для понимания выполнения данной процедуры и причин, по которой она используется, обратитесь к материалам в презентации. При выполнении данной части задания могут понадобиться функции из библиотеки NumPy, представленные ниже.

mean – позволяет вычислить арифметическое среднее вдоль определенной размерности.

std – позволяет вычислить среднеквадратическое отклонение вдоль определенной размерности. При вызове функции в настоящем задании формальному параметру ddof следует присвоить значение 1. Последнее требуется для получения несмещенной оценки среднеквадратического отклонения.

divide – позволяет выполнить поэлементное деление одного массива на другой.

repmat – выполняет повторение массивов размерности 0, 1 и 2 вдоль определенной размерности.

2. С использование ранее завершенных функций computeCost, gradientDescent выполните обучение параметров модели линейной регрессии со множеством переменных. Проведите небольшое исследование влияния параметра сходимости и числа итераций на качество сходимости градиентного спуска. Исследование можно выполнить, используя визуализацию изменения значения стоимостной функции в зависимости от числа итераций при фиксированном параметре сходимости.

3. После завершения предыдущих пунктов выполните предсказание стоимости жилья для площади 1650 квадратных футов и числа комнат 3. Обратите внимание на то, что перед выполнением процедуры предсказания требуется провести нормализацию признаков на соответствующие им математическое ожидание и среднеквадратическое отклонение.

4. Завершите программный код в модуле normalEqn.py, который позволит выполнить поиск параметров модели линейной регрессии с использованием нормальных уравнений. Для понимания выполнения данной процедуры обратитесь к материалам в презентации. При выполнении данной части

задания могут понадобиться следующие функции из библиотеки NumPy: dot, transpose и inv.

inv – позволяет выполнить вычисление обратной матрицы.

5. После завершения пункта 4 выполните предсказание стоимости жилья для площади 1650 квадратных футов и числа комнат 3. Обратите внимание на то, что при поиске параметров модели линейной регрессии с использованием нормальных уравнений, нормализация признаков не требуется. Сравните получившийся результат предсказания с результатом из пункта 3.