Национальный исследовательский университет ИТМО

Факультет ПИиКТ

Отчет по дисциплине:   
«Системы искусственного интеллекта»  
Модуль 2

Работу выполнила:

Касьяненко В.М.

Группа:

P3320

Преподаватель:

Королёва Ю. А.

Санкт-Петербург,

2024

Лабораторная работа 1. Метод линейной регрессии

**Введение**

В этой лабораторной работе рассматривается применение метода линейной регрессии для анализа и прогнозирования данных. Целью работы является изучение основ метода линейной регрессии, понимание его работы и применение на практике для решения задачи прогнозирования на основе реальных данных.

**Описание метода**

Линейная регрессия — это статистический метод, используемый для моделирования взаимосвязи между зависимой переменной (целевой переменной) и одной или несколькими независимыми переменными (признаками). Основная цель линейной регрессии заключается в нахождении линейной зависимости, которая наиболее точно предсказывает значения целевой переменной на основе значений независимых переменных. Модель представлена в виде уравнения:

где у - зависимая переменная, х1, х2, … , - независимые переменные, b0 – смещение (константа), b1, b2, …, bn - коэффициенты, которые показывают, насколько сильно каждая из независимых переменных влияет на зависимую переменную, - случайная ошибка.

**Псевдокод метода**

1. Инициализация параметров (коэффициентов) модели.

2. Определение функции ошибки, например, средней квадратичной ошибки (MSE).

3. Выбор алгоритма оптимизации (например, метод градиентного спуска).

4. Для каждой итерации алгоритма:

4.1. Вычисление предсказанных значений на основе текущих параметров модели.

4.2. Подсчет ошибки на основе текущих предсказаний и истинных значений.

4.3. Обновление коэффициентов модели с использованием градиентного спуска для минимизации ошибки.

5. Проверка сходимости: остановка алгоритма при достижении заданной точности или по истечению максимального числа итераций.

6. Вывод итоговых коэффициентов модели.

**Результаты выполнения**

Проанализирован набор данных и вычислена статистика, включая количество, среднее значение, стандартное отклонение, минимум, максимум и различные квантили для каждого признака.  
Визуализированы данные, чтобы лучше понять их распределение и корреляцию между признаками. Примеры визуализации данных были представлены в отчете.

Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание  
Для каждой модели проведена оценка производительности, используя метрику коэффициента детерминации. Этот коэффициент измеряет, насколько хорошо модель соответствует данным, где 1 - идеальное соответствие, а 0 - модель не лучше, чем простое среднее значение.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

* **Модель №1 и Модель №3** показывают схожие результаты и имеют примерно одинаковое значение R^2 (66.36% и 66.85% соответственно), что говорит о хорошем уровне предсказания для обеих моделей.
* **Модель №2** значительно уступает остальным моделям с коэффициентом R^2 равным 29.1%, что говорит о недостаточной точности этой модели для рассматриваемых данных.

Таким образом, модели №1 и №3 показали наилучшие результаты предсказаний, и могут быть использованы для решения задачи линейной регрессии. Модель №2 не так эффективна и требует доработки.

Лабораторная работа 2. Метод k-ближайших соседей (k-NN)

**Введение**

Цель данной лабораторной работы — изучить и применить метод k-ближайших соседей (k-NN) для решения задачи классификации. Метод k-NN является одним из наиболее распространённых алгоритмов классификации, основанным на сходстве объектов в многомерном пространстве признаков. В ходе работы мы реализуем данный метод без использования сторонних библиотек, кроме NumPy и Pandas, протестируем его на двух моделях с разными наборами признаков и оценим производительность с помощью матриц ошибок.

**Описание метода**

Метод k-ближайших соседей (k-NN) — это алгоритм, который классифицирует объект, основываясь на классах его ближайших соседей в пространстве признаков. Основной принцип работы заключается в том, что объект получает класс, который чаще всего встречается среди его k-ближайших соседей. Основные этапы работы метода:

1. **Вычисление расстояний**: Для каждого нового объекта рассчитываются расстояния до всех точек в обучающем наборе данных. Наиболее часто используется евклидово расстояние
2. **Выбор k-ближайших соседей**: После расчёта расстояний выбираются k-ближайших точек в обучающем наборе.
3. **Голосование**: Класс нового объекта определяется на основе наиболее часто встречающегося класса среди k-ближайших соседей.

**Псевдокод метода**

1. Для каждой точки запроса q из тестового набора данных:

2. Рассчитать расстояние между q и каждой точкой в обучающем наборе данных.

3. Отсортировать все расстояния по возрастанию.

4. Выбрать первые k соседей с наименьшим расстоянием.

5. Определить класс q по наиболее часто встречающемуся классу среди выбранных соседей.

6. Присвоить точке q предсказанный класс.

7. Повторить для каждой точки тестового набора.

**Результаты выполнения**

Для тестирования метода были построены две модели:

* **Модель 1** с набором случайных признаков.
* **Модель 2** с фиксированным набором признаков.

Для каждой модели были рассчитаны матрицы ошибок при различных значениях k (3, 5, 10).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Класс 1 классифицируется безошибочно. Классы 2 и 3 также показывают чуть лучшие результаты, чем в Модели 1, с меньшим количеством ошибок (особенно для класса 3).

Модель 2 выглядит более стабильной по сравнению с Моделью 1. У неё меньше ошибок для класса 2 и лучше классифицируется класс 3.

**Примеры использования метода**

Метод k-ближайших соседей полезен в следующих ситуациях:

* **Классификация изображений**: k-NN часто применяется для классификации объектов на изображениях на основе их похожести с ранее известными образцами.
* **Медицинская диагностика**: в задачах, где необходимо предсказать диагноз на основе симптомов, k-NN может быть полезен для поиска пациентов с похожими симптомами.
* **Рекомендательные системы**: k-NN используется для поиска пользователей с похожими предпочтениями и рекомендаций на основе их выбора.

Лабораторная работа 3. Деревья решений

**Введение**

В данной лабораторной работе рассматривается метод построения деревьев решений — один из наиболее популярных и эффективных алгоритмов для решения задач классификации и регрессии. Основная цель работы заключается в изучении принципов работы алгоритма деревьев решений, его применения для классификации данных и оценки производительности модели с использованием метрик ROC и Precision-Recall.

**Описание метода**

Дерево решений — это структура данных, представляющая собой древовидную модель решений и их возможных последствий. Алгоритм разбивает данные на основе значений признаков, в результате чего создаются узлы дерева, представляющие принятие решения. Листья дерева содержат конечные предсказания классов. Алгоритм работает рекурсивно, выбирая на каждом уровне признак с наибольшим приростом информации (Information Gain) или наивысшим значением Gain Ratio.

**Псевдокод метода**

1. Начать с исходного набора данных.

2. Рассчитать энтропию (или другую метрику неопределенности) для всего набора данных.

3. Для каждого признака рассчитать прирост информации (Information Gain):

a. Для каждого возможного значения признака разбить данные на подмножества.

b. Для каждого подмножества рассчитать энтропию.

c. Рассчитать взвешенное среднее энтропий подмножеств — это значение энтропии после разбиения.

4. Выбрать признак с наибольшим приростом информации для текущего узла дерева.

5. Разделить данные на подмножества по выбранному признаку.

6. Повторить шаги 2–5 для каждого подмножества данных, пока:

a. Все примеры в подмножестве относятся к одному классу.

b. Нет оставшихся признаков для разделения.

c. Достигнута максимальная глубина дерева.

7. Присвоить листовым узлам классы на основе распределения классов в подмножестве.

**Результаты выполнения**

После применения метода деревьев решений на выбранном наборе данных (например, на данных о грибах), мы получили следующие результаты:

* **Матрица ошибок**:
  + True Positives (TP): 3663
  + False Positives (FP): 303
  + False Negatives (FN): 138
  + True Negatives (TN): 3208
* **Метрики классификации**:
  + **Точность (Precision)**: 92.36%
  + **Полнота (Recall)**: 96.37%
  + **Общая точность (Accuracy)**: 93.97%
* **ROC-кривая** и **AUC-ROC**:
  + Площадь под ROC-кривой (AUC): 0.994, что говорит о почти идеальной способности модели различать классы.
* **PR-кривая** и **AUC-PR**:
  + Площадь под PR-кривой (AUC): 0.9945, что свидетельствует о высоком качестве классификации даже при наличии несбалансированных данных.

TPR

FPR

Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Recall

Precision

Изображение выглядит как снимок экрана, линия, текст, График

Автоматически созданное описание

**Примеры использования метода**

Метод деревьев решений может быть полезен в различных ситуациях:

1. **Медицина**: Для классификации заболеваний на основе симптомов пациента. Дерево решений может строить последовательность вопросов, на основе которых система выдает диагноз, что позволяет принять решение об эффективном лечении.
2. **Маркетинг**: Деревья решений применяются для сегментации клиентов и предсказания вероятности покупки продукта. Это помогает бизнесам нацеливать рекламные кампании на потенциально заинтересованных клиентов.
3. **Финансы**: Для предсказания риска по кредитам. Банки могут использовать деревья решений для оценки кредитоспособности клиента, анализируя такие параметры, как доход, занятость и кредитную историю.

Лабораторная работа 4. Логистическая регрессия

**Введение**

Цель лабораторной работы — изучить метод логистической регрессии и применить его для решения задач классификации. В рамках работы изучаются два метода оптимизации: градиентный спуск и метод Ньютона, а также производится оценка их эффективности на наборе данных.

**Описание метода**

**Логистическая регрессия** — это алгоритм машинного обучения, который используется для бинарной классификации. Основная идея заключается в построении зависимости между входными признаками и вероятностью того, что объект принадлежит к одному из двух классов. Логистическая функция (сигмоида) преобразует линейную комбинацию признаков в значение вероятности.

**Псевдокод метода**

1. Инициализация весов w и смещения b.

2. Повторять для заданного числа итераций:

- Рассчитать линейную комбинацию признаков: t = X \* w + b

- Применить сигмоидную функцию для получения предсказанных вероятностей: z = σ(t)

- Вычислить градиенты по весам и смещению:

dw = (1 / N) \* X^T \* (z - Y)

db = (1 / N) \* Σ(z - Y)

- Обновить веса и смещение:

w = w - learning\_rate \* dw

b = b - learning\_rate \* db

3. После завершения итераций вернуть обученные веса и смещение.

**Результаты выполнения**

В ходе эксперимента применялись два метода оптимизации: **градиентный спуск** и **метод Ньютона**. Для градиентного спуска использовались различные значения скорости обучения и числа итераций. Оценивалась производительность модели с помощью метрик:

* **Точность (Accuracy)**
* **Точность (Precision)**
* **Полнота (Recall)**
* **F1-мера**

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описание

Результаты показали, что с увеличением скорости обучения и числа итераций F1-мера и другие метрики улучшались, однако метод Ньютона продемонстрировал более стабильные результаты за меньшее количество итераций.

**Примеры использования метода**

Метод логистической регрессии может применяться в следующих задачах:

1. **Медицинская диагностика** — предсказание наличия заболевания на основе симптомов (например, диабета на основе медицинских показателей).
2. **Классификация спама** — классификация писем на спам и не спам на основе содержимого письма.
3. **Маркетинговый анализ** — предсказание вероятности отклика клиента на маркетинговую кампанию.

Этот метод выбран благодаря своей интерпретируемости, простоте реализации и эффективности при работе с линейно разделимыми данными.

**Сравнение методов**

**Сравнительный анализ методов**

* **Градиентный спуск**: требует настройки скорости обучения и большого числа итераций для достижения сходимости. Чувствителен к выбору гиперпараметров, но проще в реализации и не требует больших вычислительных затрат.
* **Метод Ньютона**: сходится быстрее за счет использования второго порядка производных (Гессиана), однако вычисление обратной матрицы может быть дорогим по времени и памяти.

**Преимущества градиентного спуска**:

* Простота реализации.
* Хорошо работает с большими данными.

**Преимущества метода Ньютона**:

* Быстрая сходимость.
* Точность решения за меньшее количество итераций.

**Примеры лучшего использования каждого метода**

* **Градиентный спуск**: предпочтителен для задач с большими наборами данных, где вычисление обратной матрицы Гессиана становится слишком ресурсоемким. Особенно эффективен при работе с разреженными данными.
* **Метод Ньютона**: лучше всего использовать, когда размер данных относительно мал, а вычислительные мощности позволяют проводить сложные вычисления. Этот метод обеспечивает более быструю сходимость при наличии вычислительных ресурсов.

Заключение

Каждый метод имеет свои сильные и слабые стороны, и выбор метода зависит от характера данных и задачи классификации.

- Линейная регрессия и логистическая регрессия подходят для задач с линейной зависимостью между признаками и целевой переменной.

- Метод k-ближайших соседей может использоваться в случаях нелинейных зависимостей.

- Деревья решений могут быть полезны для задач, где интерпретируемость играет важную роль и необходимо учитывать разнообразные признаки.

Итак, выбор метода должен базироваться на конкретных особенностях задачи и данных, а также на требованиях к интерпретируемости и производительности модели.