Лабораторная работа 2. Регрессионный анализ

Задание для самостоятельного выполнения

Цель задания

Изучить основы регрессионного анализа в машинном обучении, освоить практические навыки построения и оценки регрессионных моделей для решения задач прогнозирования непрерывных величин.

Этапы выполнения:

1. Выбор датасета

Выберите датасет на сайте kaggle.com, подходящий для решения задач регрессии (предсказания непрерывных значений), если вариант с 1 ЛР работы подходит для данной задачи, можно оставить его.

2. Предобработка данных

Выполните необходимые шаги для подготовки данных к обучению моделей:

Обязательные шаги предобработки:

- Анализ пропущенных значений и их обработка (удаление или заполнение)
- Обнаружение и обработка выбросов (визуализация, анализ влияния)
- Кодирование категориальных переменных (если есть)
- Нормализация/стандартизация признаков (при необходимости)
- Разделение на обучающую и тестовую выборки

3. Построение и обучение моделей

Обучите и сравните следующие модели регрессии:

Обязательные модели для реализации:

1. Линейная регрессия (`LinearRegression`)

2. Полиномиальная регрессия степени (`PolynomialFeatures + LinearRegression`)
3. Регрессия с регуляризацией Ridge (`Ridge`)
4. Регрессия с регуляризацией Lasso (`Lasso`)

Дополнительные модели:

1. Случайный лес (`RandomForestRegressor`)
2. Градиентный бустинг (`GradientBoostingRegressor`)
3. Другие модели, которые по вашему мнению подходят для решения вашей задачи, но нужно суметь объяснить принцип их работы.

4. Оценка качества моделей

Для каждой модели вычислите следующие метрики:

Метрики для оценки:

- Средняя абсолютная ошибка (MAE)
- Среднеквадратичная ошибка (MSE)
- Среднеквадратическое отклонение (RMSE)
- Коэффициент детерминации R^2

Создайте сравнительную таблицу со всеми результатами.

5. Выбор лучшей модели и анализ результатов

Выберите лучшую модель на основе метрик качества и обоснуйте свой выбор.

Для лучшей модели выполните:

- Анализ важности признаков (для моделей, поддерживающих feature importance)
- Сравнение предсказаний с эталонными значениями на графиках

Дополнительные задания (по желанию):

- Тюнинг гиперпараметров лучшей модели
- Создание новых признаков
- Ансамблирование моделей

4. Контрольные вопросы

1. Что такое пропущенные значения? Какие основные способы их обработки вы знаете?
2. Что такое выбросы в данных?
3. Что такое категориальные переменные?
4. Что такое One-Hot Encoding?
5. Для каких моделей обязательна стандартизация признаков и почему?
6. Что такое регрессия в машинном обучении? Приведите пример задач регрессии.
7. В чём отличие регрессии от классификации?
8. Что означают коэффициенты в уравнении простой линейной регрессии?
9. Что такое множественная линейная регрессия? Как интерпретировать её коэффициенты?
10. Можно ли использовать линейную регрессию для нелинейных зависимостей? Если да, то как?
11. Что такое полиномиальная регрессия? Когда она применяется?
12. Как интерпретировать коэффициент детерминации R^2? Какие значения он может принимать?
13. В каком случае R^2 может быть отрицательным?
14. Что показывает метрика MAE? В каких единицах она измеряется?
15. В чём разница между MSE и RMSE? Когда лучше использовать каждую из них?
16. Что такое переобучение в контексте регрессии? Как его обнаружить?
17. Зачем нужно разделять данные на обучающую и тестовую выборки?
18. Объясните разницу между train_score и test_score. Что означает большая разница между ними?
19. Что такое кросс-валидация?
20. Чем Lasso отличается от Ridge?
21. Что такое гиперпараметры?

22. Что такое гиперпараметр alpha? Как он влияет на силу регуляризации?
23. Что означает Pipeline в scikit-learn? Зачем он используется?
24. Что такое мультиколлинеарность признаков? Как она влияет на линейную регрессию?
25. Как влияет масштабирование признаков на коэффициенты линейной регрессии?
26. Как можно визуально проверить, подходит ли линейная модель данным?
27. Чем отличаются параметры модели от гиперпараметров?
28. Что такое bias и variance? Как они связаны с качеством модели?
29. Какие способы борьбы с переобучением кроме регуляризации вы знаете?
30. Почему важно фиксировать random_state при обучении моделей?