

Приложение 1

Подробный план работы:

1. Загружаем и обрабатываем входящие датасеты

1.1. Удаляем неинформативные столбцы 1.2. Объединяем датасеты по методу INNER

2. Проводим разведочный анализ данных:

2.1. Данные в столбце "Угол нашивки" приведём к 0 и 1

2.2. Изучим описательную статистику каждой переменной - среднее, медиана, стандартное отклонение, минимум, максимум, квантили

2.3. Проверим датасет на пропуски и дубликаты данных

2.4. Получим среднее, медианное значение для каждой колонки (по заданию необходимо получить их отдельно, поэтому продублируем их только отдельно)

2.5. Вычислим коэффициенты ранговой корреляции Кендалла

2.6. Вычислим коэффициенты корреляции Пирсона

3. Визуализируем наш разведочный анализ сырых данных (до выбросов и нормализации)

3.1. Построим несколько вариантов гистограмм распределения каждой переменной

3.2. Построим несколько вариантов диаграмм ящиков с усами каждой переменной

3.3. Построим гистограмму распределения и диаграмма "ящик с усами" одновременно вместе с данными по каждому столбцу

3.4. Построим несколько вариантов попарных графиков рассеяния точек (матрицы диаграмм рассеяния)

4. Проведём предобработку данных (в данном пункте только очистка датасета от выбросов)

4.1. Проверим выбросы по 2 методам: 3-х сигм или межквартильных расстояний

4.2. Посчитаем распределение выбросов по каждому столбцу (с целью предотвращения удаления особенностей признака или допущения ошибки)

4.3. Исключим выбросы методом межквартильного расстояния

4.4. Удалим строки с выбросами

4.5. Визуализируем датасет без выбросов, и убедимся, что выбросы еще есть.

4.6. Для полной очистки датасета от выбросов повторим пункты (4.3 – 4.5) ещё 3 раза.

4.7. Сохраняем идеальный, без выбросов датасет

4.8. Изучим чистые данные по всем параметрам

4.9. Визуализируем «чистый» датасет (без выбросов)

5. Проведём нормализацию и стандартизацию (продолжим предобработку данных)

5.1. Визуализируем плотность ядра

5.2. Нормализуем данные с помощью MinMaxScaler()

5.3. Нормализуем данные с помощью Normalizer()

5.4. Сравним с данными до нормализации

5.5. Проверим перевод данных из нормализованных в исходные

5.6. Рассмотрим несколько вариантов корреляции между параметрами после нормализации

5.7. Стандартизируем данные

5.8. Визуализируем данные корреляции

5.9. Посмотрим на описательную статистику после нормализации и после стандартизации

6. Разработаем и обучим нескольких моделей прогноза прочности при растяжении (с 30% тестовой выборки)

6.1. Определим входы и выходы для моделей

6.2. Разобьём данные на обучающую и тестовую выборки

6.3. Проверим правильность разбивки

6.4. Построим модели и найдём лучшие гиперпараметры (задача по заданию):

6.5. Построим и визуализируем результат работы метода опорных векторов

6.6. Построим и визуализируем результат работы метода случайного леса

6.7. Построим и визуализируем результат работы линейной регрессии

- 6.8. Построим и визуализируем результат работы метода градиентного бустинга
- 6.9. Построим и визуализируем результат работы метода К ближайших соседей
- 6.10. Построим и визуализируем результат работы метода дерева решений
- 6.11. Построим и визуализируем результат работы стохастического градиентного спуска
- 6.12. Построим и визуализируем результат работы многослойного перцептрона
- 6.13. Построим и визуализируем результат работы лассо регрессии
- 6.14. Сравним наши модели по метрике MAE
- 6.15. Найдём лучшие гиперпараметры для случайного леса
- 6.16. Подставим значения в нашу модель случайного леса
- 6.17. Найдём лучшие гиперпараметры для К ближайших соседей
- 6.18. Подставим значения в нашу модель К ближайших соседей
- 6.19. Найдём лучшие гиперпараметры метода дерева решений
- 6.20. Подставим значения в нашу модель метода дерева решений
- 6.21. Проверим все модели и процессинги и выведем лучшую модель и процессинг

7. Разработаем и обучим нескольких моделей прогноза модуля упругости при растяжении (с 30% тестовой выборки)

- 7.1. Определим входы и выходы для моделей
- 7.2. Разобьём данные на обучающую и тестовую выборки
- 7.3. Проверим правильность разбивки
- 7.4. Построим модели и найдём лучшие гиперпараметры (задача по заданию)
- 7.5. Построим и визуализируем результат работы метода опорных векторов
- 7.6. Построим и визуализируем результат работы метода случайного леса
- 7.7. Построим и визуализируем результат работы линейной регрессии
- 7.8. Построим и визуализируем результат работы метода градиентного бустинга
- 7.9. Построим и визуализируем результат работы метода К ближайших соседей
- 7.10. Построим и визуализируем результат работы метода дерева решений
- 7.11. Построим и визуализируем результат работы стохастического градиентного спуска
- 7.12. Построим и визуализируем результат работы многослойного перцептрона
- 7.13. Построим и визуализируем результат работы лассо регрессии
- 7.14. Сравним наши модели по метрике MAE
- 7.15. Найдём лучшие гиперпараметры для случайного леса
- 7.16. Подставим значения в нашу модель случайного леса
- 7.17. Найдём лучшие гиперпараметры для К ближайших соседей
- 7.18. Подставим значения в нашу модель К ближайших соседей
- 7.19. Найдём лучшие гиперпараметры метода дерева решений
- 7.20. Подставим значения в нашу модель метода дерева решений
- 7.21. Проверим все модели и процессинги и выведем лучшую модель и процессинг

8. Нейронная сеть для рекомендации соотношения матрица-наполнитель

- 8.1. Сформируем входы и выход для модели
- 8.2. Нормализуем данные
- 8.3. Построим модель, определим параметры
- 8.4. Найдём оптимальные параметры для модели
- 8.5. Посмотрим на результаты
- 8.6. Повторим шаги 8.4 – 8.5 до построения окончательной модели
- 8.7. Обучим нейросеть 80/20
- 8.8. Оценим модель
- 8.9. Посмотрим на потери модели
- 8.10. Посмотрим на график результата работы модели
- 8.11. Посмотрим на график потерь на тренировочной и тестовой выборках
- 8.12. Сконфигурируем другую модель, зададим слои
- 8.13. Посмотрим на архитектуру другой модели
- 8.14. Обучим другую модель

- 8.15. Посмотрим на потери другой модели
- 8.16. Посмотрим на график потерь на тренировочной и тестовой выборках
- 8.17. Зададим функцию для визуализации факт/прогноз для результатов моделей
- 8.18. Посмотрим на график результата работы модели
- 8.19. Оценим модель MSE
- 8.20. Сохраняем вторую модель для разработки веб-приложения для прогнозирования соотношения "матрица-наполнитель" в фреймворке Flask

9. Создаём приложение

- 9.1. Импортируем необходимые библиотеки
- 9.2. Загрузим модель и определим параметры функции
- 9.3. Получим данные из наших форм и положим их в список
- 9.4. Укажем шаблон и прототип сайта для вывода
- 9.5. Запустим приложение
- 9.6. Откроем <http://127.0.0.1:5000/>
- 10. Создание удалённого репозитория и загрузка результатов работы на него
- 10.1. <https://github.com/AlekseyGrudin/VKR>
- 10.2. Создадим README <https://github.com/AlekseyGrudin/VKR#readme>
- 10.3. Выгрузим все необходимые файлы в репозитор