МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science»

Слушатель Макаренко Геннадийй Анатольевич

Москва, 2022

СОДЕРЖАНИЕ

*Введение*

1. *Аналитическая часть*
   1. *Постановка задачи.*
   2. *Описание используемых методов*
   3. *Разведочный анализ данных*
2. *Практическая часть*
   1. *Предобработка данных*
   2. *Разработка и обучение модели*
   3. *Тестирование модели*
   4. *Написать нейронную сеть, которая будет рекомендовать соотношение матрица.*
   5. *Разработка приложения*
   6. *Создание удаленного репозитория и загрузка результатов работы на него.*

**Введение**

Данная выпускная квалификационная работа направлена на закрепление полученных знаний по курсу «Data Sсience». В работе показан поэтапный процесс загрузки исходных данных, их разведочного анализа и предобработки с последующим обучением нескольких моделей регрессии методом машинного обучения с учителем. Также показан пример сборки и обучения полносвязанной нейронной сети. В завершении квалификационной работы выполнена разработка конечного приложения и загрузка результатов работы на удаленный репозиторий.

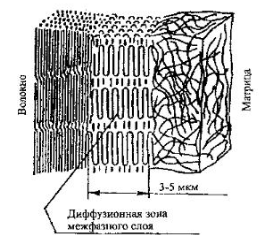
Термин «Data Sсience» или «Наука о данных» получил широкое распространение с начала 2000-х гг., когда привычные методы обработки и анализа данных, основанные на методах математической статистики, стали активно дополняться методами и технологиями искусственного интеллекта совместно с большими данными. Статистические методы и сегодня часто рассматриваются как важная компонента науки о данных, но неотъемлемой ее частью стали методы обучения машин, искусственные нейросети и другие технологии искусственного интеллекта. Основная цель универсального специалиста по анализу данных – выявить закономерности в имеющихся данных и извлечь скрытую в них информацию на базе широкого набора аналитических инструментов – от методов математической статистики до машинного обучения, искусственных нейросетей и технологий больших данных. В отличие от традиционного статистического анализа в науке о данных больше компетенций в информационных технологиях, программировании, методах визуализации.

**Аналитическая часть**

* 1. **Постановка задачи**

 Композиционные материалы и конструкции находят широкое применение в различных областях техники уже достаточно давно. Связано это в первую очередь с их более высокими удельными характеристиками, а также с возможностью изменить свойства материала в тех направлениях и местах конструкции, где это наиболее необходимо. Основными областями применения нанокомпозитных материалов в настоящее время являются автомобилестроение, авиастроение, космическая промышленность, производство упаковочных материалов, спортинвентаря.

Целью работы является: построение модели для прогнозирования конечных свойств новых композиционных материалов. Свойства композитов зависят в первую очередь от свойств исходных компонентов: армирующих элементов и матрицы. Кроме того, их соединение дает эффект синергизма, связанный с появлением у композита свойств не характерных для исходных компонентов. Пример строения композита представлен на рисунке 1.

Рисунок 1 - Структура композита

В качестве исходных данных для построения модели имеются наборы числовых значений с физическими характеристиками армирующих элементов (предположительно ткани) и матрицы, а также показателями прочности получаемого композита. Данные состоят из 2-х отдельных наборов размерностью 1023 х 10 и 1040 х 3 соответственно (рисунок 2).

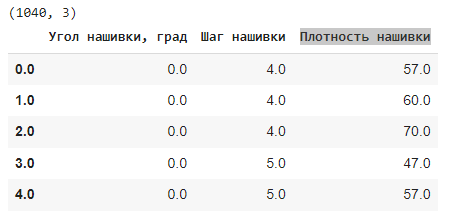


Рисунок 2. Исходные наборы данных.

Данные наборы данных не имеют общих колонок кроме индекса, их соединение в одну таблицу проведено методом INNER :

*rawdata = rawdata\_2.join(rawdata\_1, how='inner')*

Полученный набор данных содержит 11 входных и 2 целевых переменных. Входные переменные:

1. Угол нашивки (Stripe\_angle) - ориентация армирующих волокон слоя, в градусах, относительно базового направления.
2. Шаг нашивки (Stripe\_Step) – шаг расположения армирующих волокон в слое. В предоставленной документации не указаны единицы измерения данной переменной. Изучение тематики подсказывает, что шаг обычно измеряется в мм.
3. Плотность нашивки (Stripe\_density) – мера линейной плотности характерная для тканных материалов. Единицы измерения не указаны, может иметь различные единицы измерения в зависимости от технологии. Наиболее вероятная количество стежков на см2.
4. Поверхностная плотность (Surface\_density), г/м2 – типовая характеристика тканных материалов, характеризующая вес 1 квадратного метра полотна.
5. Соотношение матрица-наполнитель (Matrix\_ratio)– очевидно имеется в виду объемное соотношение матрицы к ее наполнителю.
6. Плотность, кг/м3 (Density) – нет конкретных пояснений. Предположительно подразумевается плотность углеволокна, что является входящим признаком модели.
7. Модуль упругости, Гпа (M\_Elasticity ) – порядок цифр указывает на то, что этот признак относится к наполнителю наподобие высокопрочного углеволокна. Принимаем как входящий.
8. Количество отвердителя, м.% (Hardener\_ratio) - типовая характеристика эпоксидных и им подобных матриц.
9. Содержание эпоксидных групп, %\_2 (Epoxy\_ratio) – то же.
10. Температура вспышки, С\_2 (Flash\_point) – то же.
11. Потребление смолы, г/м2 (Epoxy\_Consumption) – то же.

Целевые переменные:

1. Модуль упругости при растяжении, Гпа (Tension\_Elasticity) – способность композита упруго деформироваться.
2. Прочность при растяжении, Мпа (Tension\_Rigidity) – способность композита сопротивляться разрушению.

Перечисленные параметры не описывают всех свойств компонентов композитов, влияющих на их конечные прочностные характеристики. Разрабатываемая модель на основе этих данных будет нести учебную пользу, для построения модели промышленного применения необходимо более глубокое изучение основ композитных материалов с подбором дополнительных влияющих признаков.

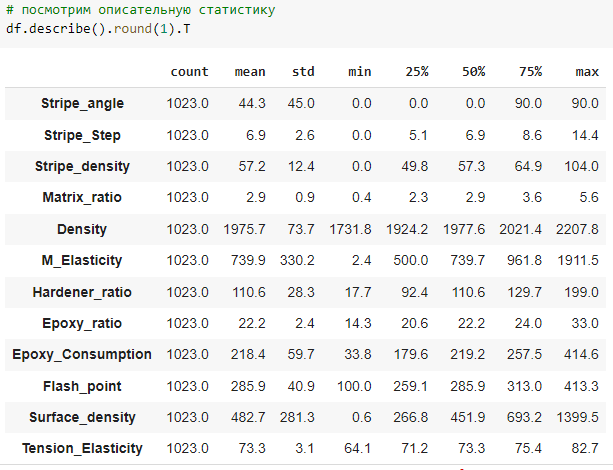
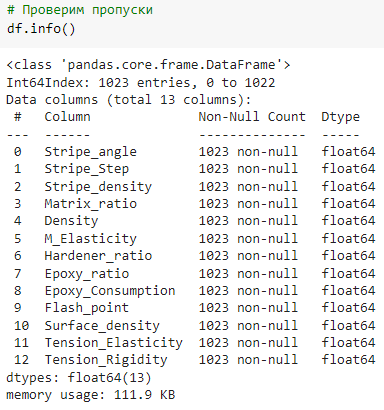
Выполним проверку данных. В наборе пропусков нет, но есть аномальные нулевые значения для Большинство регрессионных моделей чувствительны к наличию выбросов. В используемом наборе имеется значительное их число, рисунок 3.

Рисунок 3 – Количество и тип данных Рисунок 4 – Статистика

Проверка распределения данных с помощью диаграмм рассеяния на рисунке 5 не выявляет каких-то значимых закономерностей, в том числе и после отсеивания выбросов методом DBSCAN. Большинство пар признаков имеют равномерное рассеивание ввиду облака, при этом гистограмма распределения признака близка к нормальной. Данный факт ставит под сомнение возможность выявления и построения регрессионной модели, которая априори предполагает наличие зависимостей между признаками и целевыми переменными. Те же выводы подтверждает и корреляционная диаграмма на рисунке 6. Поэтому прежде, чем приступать к построению регрессионных моделей потребуется более глубокая очистка и подготовка данных. Основное внимание на этом этапе необходимо уделить первым 22 строкам исходного набора данных.

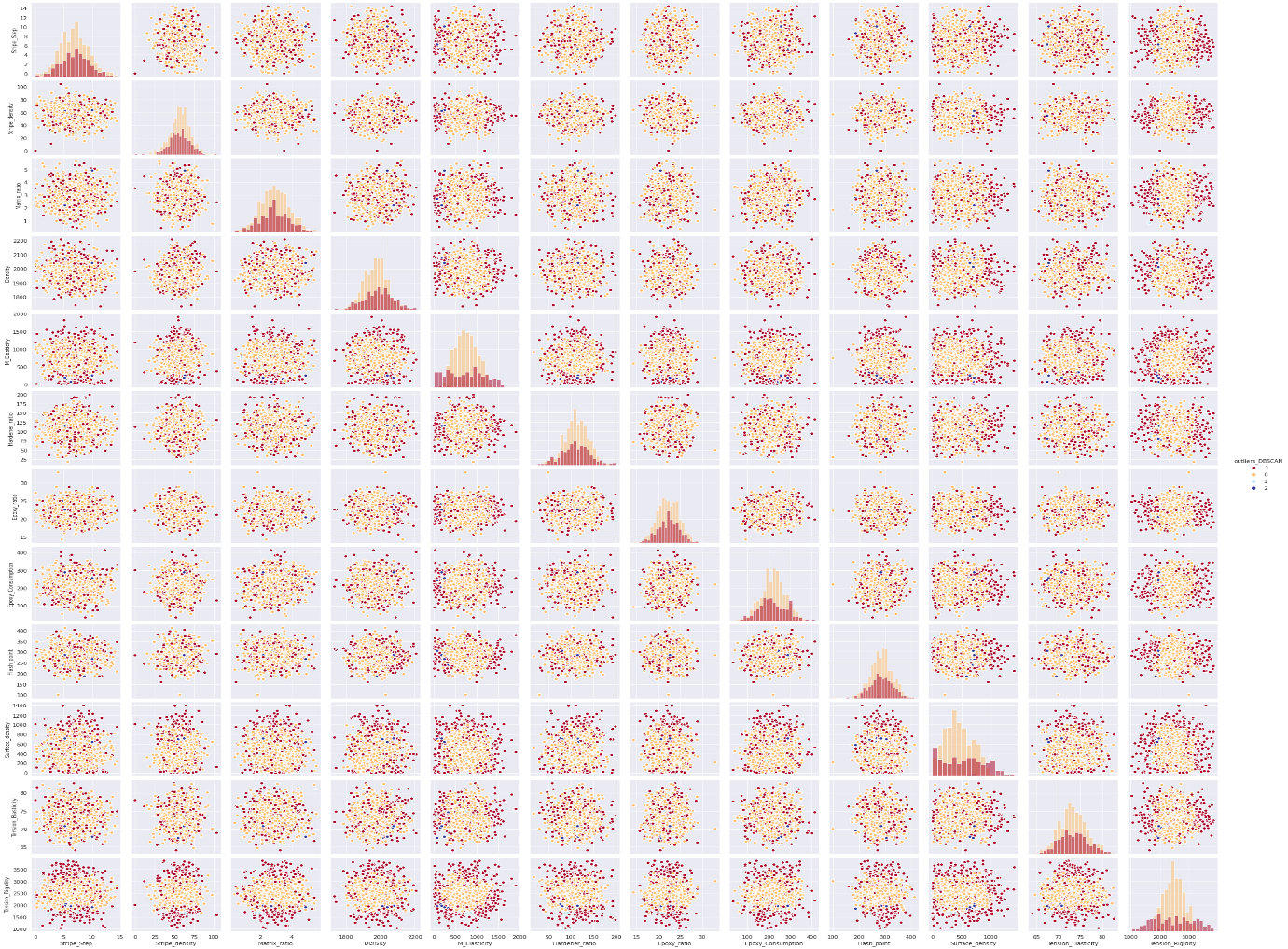
Рисунок 5 – Диаграмма рассеяния исходного набора данных



Рисунок 6 – Корреляционная матрица исходного набора данных

* 1. *Описание используемых методов*

Решение данной задачи может быть достигнуто методом регрессии.

Регрессия – это модель из категории «обучение с учителем», устанавливающая связь между одним или несколькими независимыми признаками (предикторами) и зависимым континуальным признаком. В линейной регрессии моделью зависимости является формула прямой линии. Ввиду того что явных зависимостей переменных не выявлено, то применение линейной модели регрессии нецелесообразно. Поэтому в дальнейшем исследовании большее внимание будет уделено нелинейным регрессионным моделям. Большинство методов регрессии позволяют строить модель только для одной целевой переменной, для регрессии по двум целевым переменным будут разработаны 2 параллельные модели по числу целевых переменных.

Опишем известные методы регрессий и их ключевые моменты.

1. [Метод наименьших квадратов](http://ru.wikipedia.org/wiki/%CC%E5%F2%EE%E4_%ED%E0%E8%EC%E5%ED%FC%F8%E8%F5_%EA%E2%E0%E4%F0%E0%F2%EE%E2)
   1. Простая (одномерная) линейная регрессия (модель с одним предиктором) аппроксимируется всем известной со школы функцией прямой линии с той лишь разницей, что теперь нужно добавить случайную ошибку epsilon «ϵ»:

*yi*​=*β*0​+*β*1​*xi*​+*ϵi*​, ,

где *yi*​ — зависимая переменная,

*xi*​ – объясняющая переменная,

*β*0​, *β*1​ — параметры модели, *ϵi*​ — случайная ошибка

* 1. Множественная линейная регрессия

*h*∈H, *h*(*x*)=*w*0​*x*0​+*w*1​*x*1​+*w*2​*x*2​+⋯+*wm*​*xm*​=∑*i*=0*m*​*wi*​*xi* =*xTw*​

* 1. Полиноминальная регрессия

Представляет самый общий случай регрессии, может быть нелинейной. Уравнение полинома имеет вид



где bj — параметры данного полинома. Среди них b0 — свободный член.

1. Логистическая регрессия

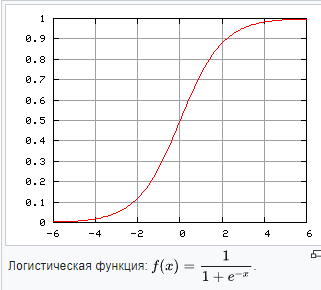
Применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков. Зависимая переменная y, принимающая лишь одно из двух значений —это числа 0 и 1, и множество независимых переменных, на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной. Основное назначение логистической регрессии – бинарная классификация, но она может применяться и для регрессионной модели, если закодировать целевую переменную методом LabelEncoder. Основное уравнение логистической функции, так называемая сигмоида, представлена на рисунке 7.

Рисунок 7

1. Регрессия Лассо.

Эта техника является одним из видов линейной регрессии и помогает уменьшить переобучение модели. В регрессии лассо, как и в гребневой, мы добавляем условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, мы используем смещение абсолютного значения:

min || Xw — y ||² + z|| w ||

Где X — это матрица переменных, w — веса, y — достоверные данные, z - смещение.

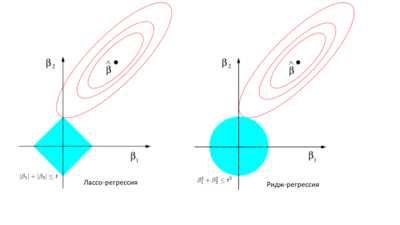
Значения данных сжимаются к центру, чтобы избежать переобучения. Лассо регрессия очень похожа на концепцию хребтовой регрессии (Ridge). Но регрессия Лассо может исключить из уравнения бесполезные переменные. Этот тип регрессии лучше, чем регрессия Ridge, и помогает уменьшить количество признаков в модели машинного обучения, которая содержит много признаков. Иллюстрация различия к регрессиям Лассо и Хребтовой представлена на рисунке 8.

Рисунок 8 – Регрессии с регуляризацией

1. Дерево принятия решений и Случайный лес.

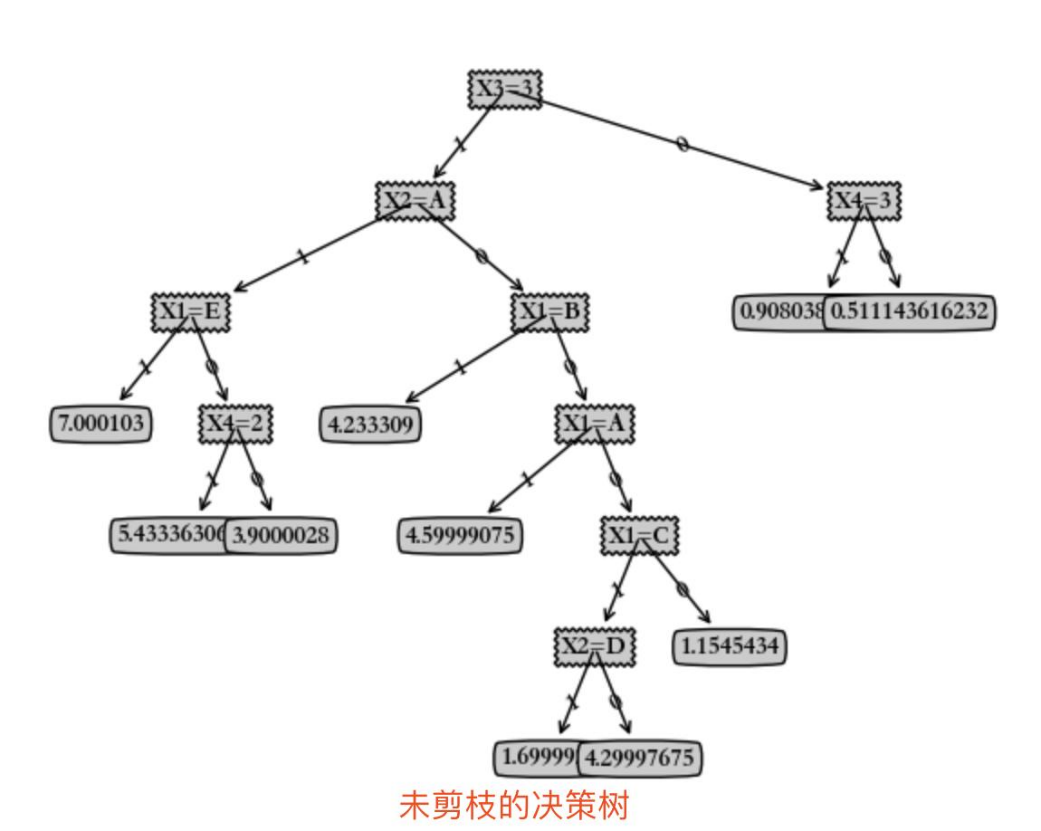
Дерево принятия решений – это представления правил, находящихся в последовательной, иерархической структуре, где каждому объекту соответствует узел, дающий решение. При построении дерева важно классифицировать атрибуты так, чтобы создать “чистые” узлы. То есть выбранный атрибут должен разбить множество так, чтобы получаемые в итоге подмножества состояли из объектов, принадлежащих к одному классу, или были максимально приближены к этому, то есть количество объектов из других классов в каждом из этих множеств было как можно меньше. “Случайные лес” – совокупность деревьев принятия решений. Входной вектор проходит через несколько деревьев решений. Для регрессии выходное значение всех деревьев усредняется.

Рисунок 8 – Пример регрессионного дерева решений.

1. Метод опорных векторов (SVM).

Это особый класс алгоритмов, который характеризуется использованием ядер, отсутствием локальных минимумов, и используется для решения задач классификации и регрессии. Графическое представление метода опорных векторов представлено на рисунке 9.



Рисунок 9 – Иллюстрация метода опорных векторов (SVM).

1. Квантильная регрессия

во многих обстоятельствах нас больше интересует медиана или определенный квантиль масштабированного результата. Квантильная регрессия моделирует взаимосвязь между набором переменных предикторов (независимых) и определенными процентилями (или "квантилями") целевой переменной (зависимой), чаще всего медианой.

1. Регрессия методом главных компонент

Эта регрессия начинается с использования главных компонент эндогенных переменных вместо самих эндогенных переменных. Так как главные компоненты некоррелированны, расчеты коэффициентов регрессии могут быть сильно упрощены. Если все главные компоненты включены в регрессию, то в результате полученная регрессионная модель эквивалентна модели, полученной непосредственно по системе эндогенных переменных методом наименьших квадратов, так что погрешности, вызванные мультиколлинеарностью, при этом не исчезают. Однако, вычисления оценок методом наименьших квадратов через регрессию главных компонент могут быть численно более стабильным, чем прямые расчеты.

1. Регрессия наименьших частичных квадратов

Регрессия частично наименьших квадратов оценивает регрессионные модели частично наименьших квадратов (PLS), также известные как модели "проекции на скрытую структуру". PLS представляет собой метод для предсказания, который является альтернативой обычной регрессии наименьших квадратов (OLS), каноническим корреляциям или построению моделей с помощью структурных уравнений. Он особенно полезен, когда предикторные переменные сильно коррелированы или когда число предикторов превышает число наблюдений. PLS соединяет свойства метода главных компонент и множественной регрессии. Сначала он выделяет набор скрытых факторов, которые объясняют как можно больше ковариации между независимыми и зависимыми переменными. Затем на шаге регрессии предсказываются значения зависимых переменных с использованием декомпозиции независимых переменных.

1. Порядковая регрессия.

Является расширением обобщенной линейной модели регрессии, в которой зависимая переменная измеряется в порядковой шкале. Независимые переменные в модели порядковой регрессии могут быть категориальными или количественными. Категориальные независимые переменные называют факторами. А количественные независимые переменные – ковариатами. Модель порядковой регрессии напоминает модель мультиномиальной логистической регрессии. В модели порядковой регрессии для каждой категории зависимой порядковой переменной (за исключением последней) строится уравнение регрессии, прогнозирующее накопленную вероятность принадлежности объекта наблюдения к данной категории. Отличие порядковой регрессии от мультиномиальной логистической регрессии в том, что в качестве связывающей зависимую переменную и независимые переменные функции может использоваться не только логистическая функция.

1. Регрессия Пуассона.

Это особый тип регрессионного анализа, который обычно используется для счетчиков моделей. Например, регрессия Пуассона может быть полезна в следующих случаях: Моделирование числа заболеваний, связанных с перелетами в самолетах.

1. Регрессия Кокса

Иначе модель пропорциональных рисков, — прогнозирование риска наступления события для рассматриваемого объекта и оценка влияния заранее определенных независимых переменных (предикторов) на этот риск. Риск рассматривается как функция, зависящая от времени. Ключевые особенности различных методов регрессии сведены в таблице ниже после изучения темы.

Таблица 1 – Ключевые особенности методов регрессии.

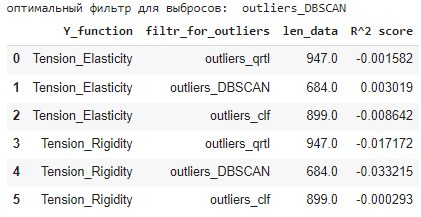
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование метода | Нелинейность выборки | Устойчивость к многомерным признакам | Устойчивость к коллинеарности | Устойчивость к выбросам | Без нормализации | Прочие отметки |
| Линейная регрессия | нет | нет | нет | нет | нет | простота |
| Множественная линейная регрессия | нет | нет | нет | нет | нет | простота |
| Полиноминальная регрессия | да | нет | нет | нет | нет | больше охват |
| Логистическая регрессия | да | нет | нет | нет | нет | нелинейные |
| Регрессия Лассо | да | да | да | нет | нет | улучшение регрессии |
| Случайный лес | да | да | нет | да | да | универсальность |
| Метод опорных векторов (SVM) | да | да | нет | нет | нет | для малых выборок |
| Квантильная регрессия | да | да | нет | да | нет | предсказать средние значения |
| Регрессия методом главных компонент | да | да | да | да | нет | не для объяснения фактических связей |
| Регрессия наименьших частичных квадратов | нет | да | да | да | нет | для малых выборок |
| Порядковая регрессия | да | нет | нет | нет | нет | для привязки к значениям шкал |
| Регрессия Пуассона | да |  | нет | нет |  | гетерогенных данных |
| Регрессия Кокса |  | нет | нет |  |  | для оценки времени до события |

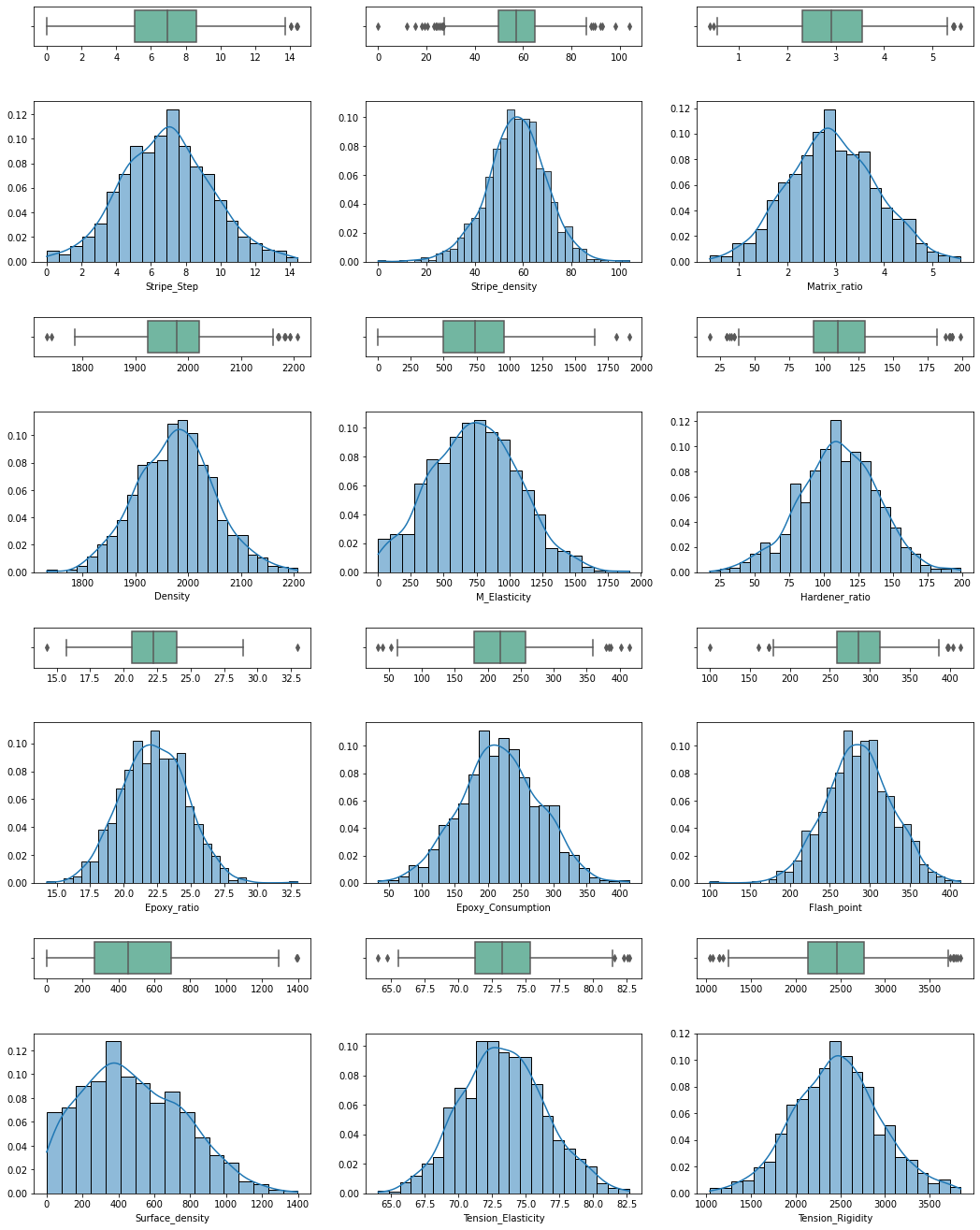
* 2. *Разведочный анализ данных*

Для разведочного анализа данных были широко использованы такие библиотеки Python как:

* Pandas:
  + метод info() для проверки пропусков и типов данных,
  + метод duplicated() для проверки повторяющихся строк или значений,
  + метод describe() для вывода описательной статистики,
  + corr() для создания матрицы корреляции,
  + shape для вывода размерности данных,
* Seaborn:
  + heatmap() для визуализации матриц через тепловую карту,
  + boxplot() для визуализации выбросов методом квартилей,
  + pairplot() для парной визуализации всех признаков,
* Matplotlib
  + plot(), scatter() визуализации распределений
* Sklearn:
  + DBSCAN, IsolationForest для многомерного анализа выбросов.

Разведочный анализ данных подтвердил высокую зашумленность данных, распределение их величин близкое к нормальному закону (рисунок 10), отсутствие какой-либо корреляции признаков и целевых переменных (рисунок 6), равномерное рассеяние данных в виде облака (рисунок 5). Гипотеза отсутствия значимых связей признаков с целевой переменной была подтверждена на тестировании обученной модели полиноминальной регрессии с оптимизацией Лассо. Было протестировано отсечение выбросов и аномалий тремя различными методами для которое не улучшило выявление зависимостей в данных. Результат сравнительного анализа этих методов представлен в таблице 2. Во время разведочного анализа данных была замечена аномальная разность в наборе данных до и после 22 строки. Разделение набора на 2 подвыборки по этому признаку и их последующий разведочный анализ вышеперечисленными методами показал выявление корреляции и взаимосвязей в первом наборе (рисунок 11, 12), но отсутствие взаимосвязей и равномерный шум во втором (рисунок 13). Опираясь на этот факт в дальнейшем обучении моделей, была использована первая подвыборка из набора данных.

Таблица 2 – сравнение разных фильтров

Рисунок 10 – распределение значений признаков

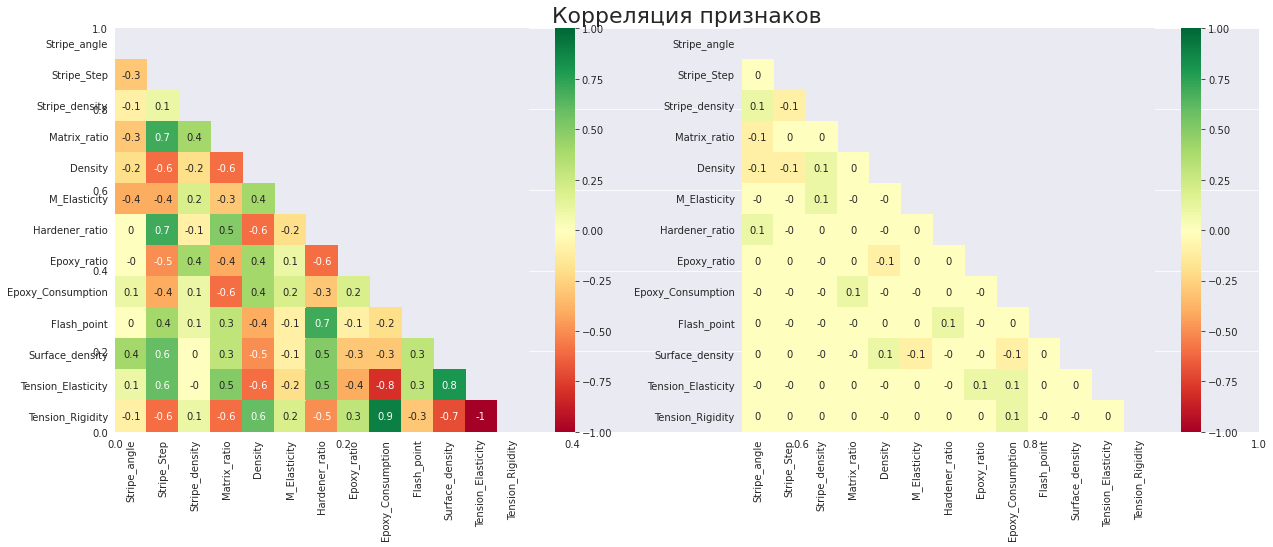


Рисунок 11 – матрица корреляции первой (слева) и второй (справа) подвыборок

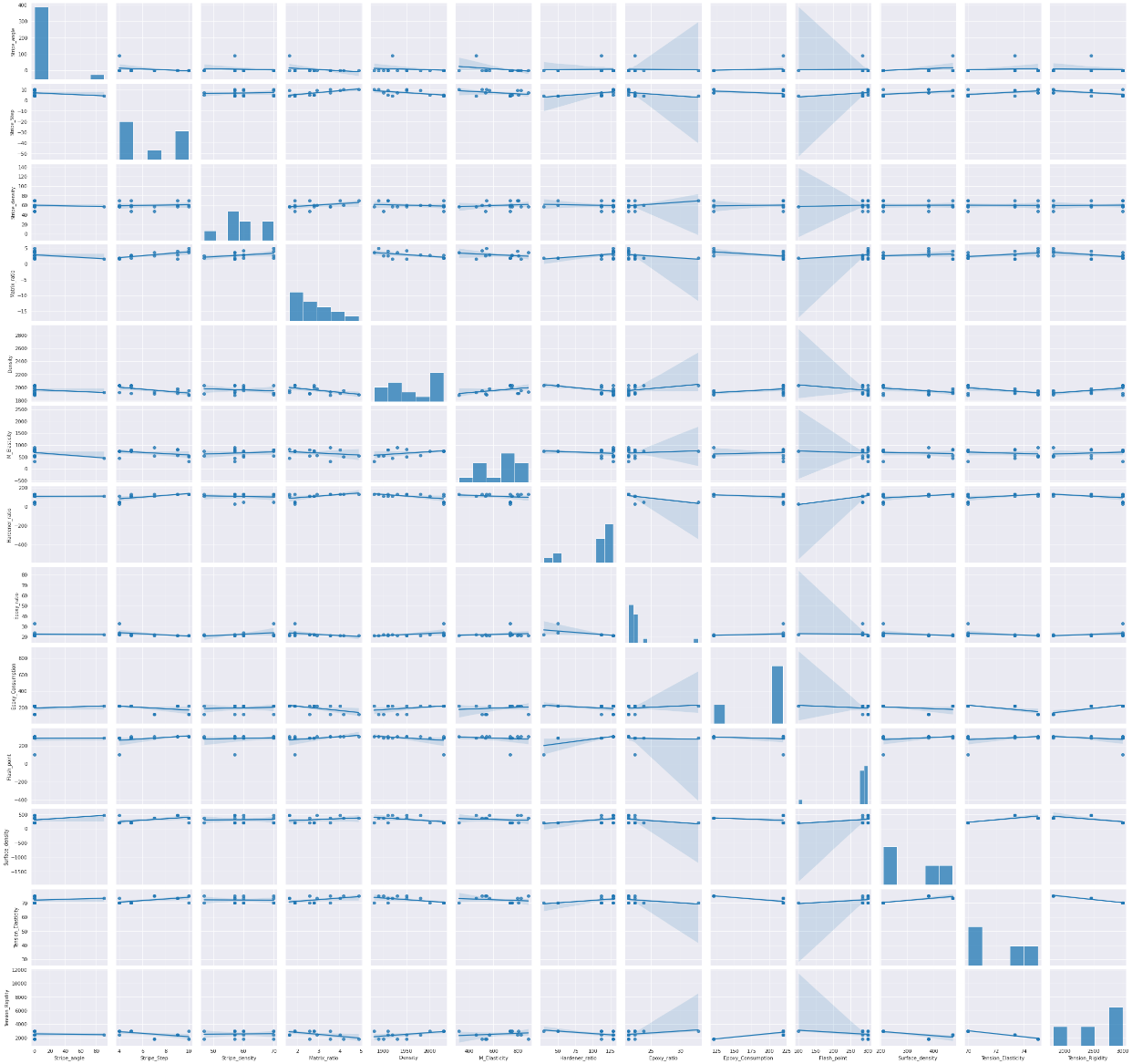


Рисунок 12 – диаграмма рассеяния на первой подвыборке

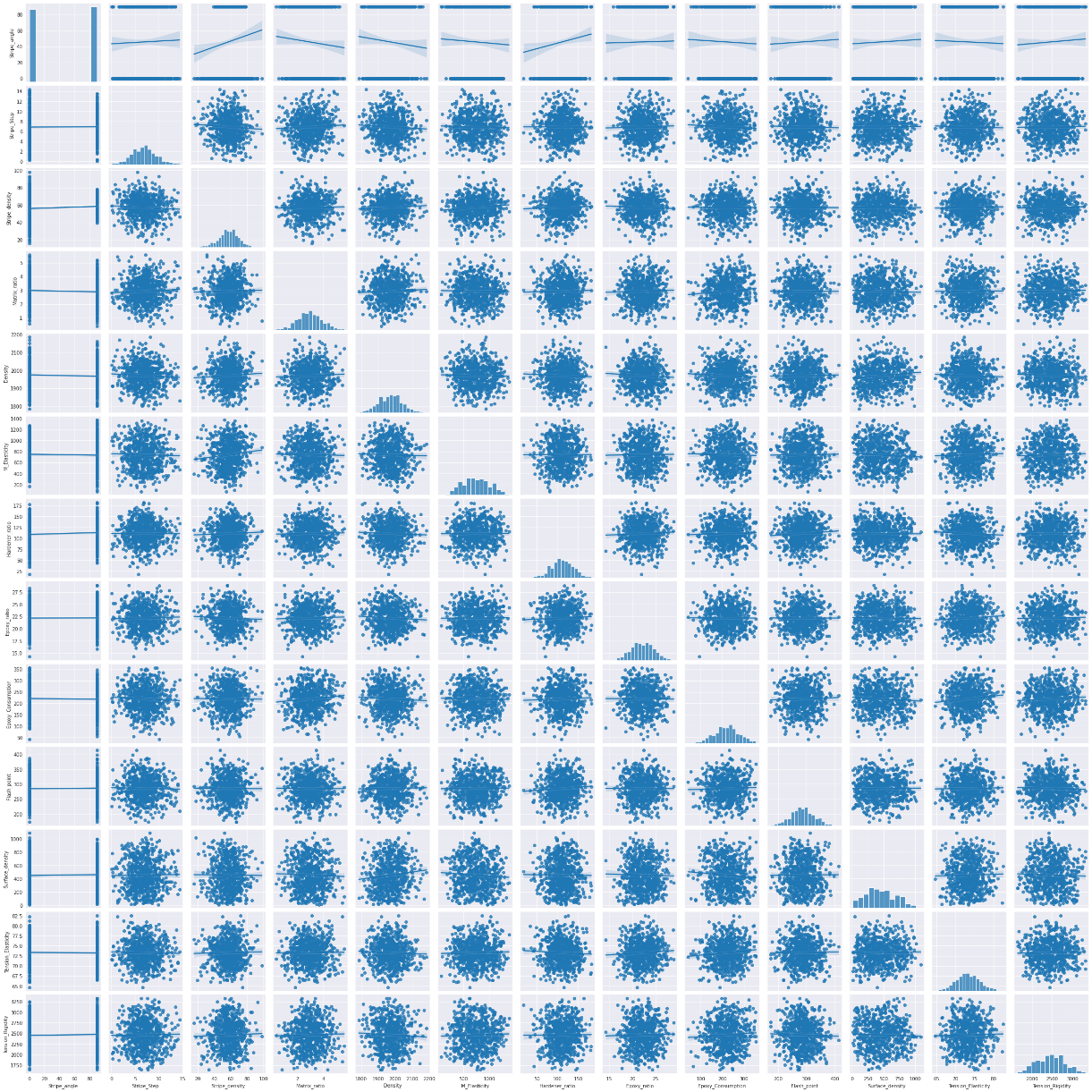


Рисунок 13 – Диаграмма рассеяния на второй подвыборке

Стоит отметить, что данные в первой подвыборке сильно разрежены и неравномерны, что следует учитывать при последующем подборе модели регрессии. Дополнительно был выполнен разведочный анализ методом Главных компонент (PCA) для проверки признаков на информативность или избыточность. Результат представлен на рисунках 14 и 15. На первой и второй подвыборках выявлено 6 и 11 (из 11) главных компонент соответственно. Ввиду малого размера первой подвыборки сокращение размерности нецелесообразно, так как может привести к потере значимых связей и ухудшению результата.

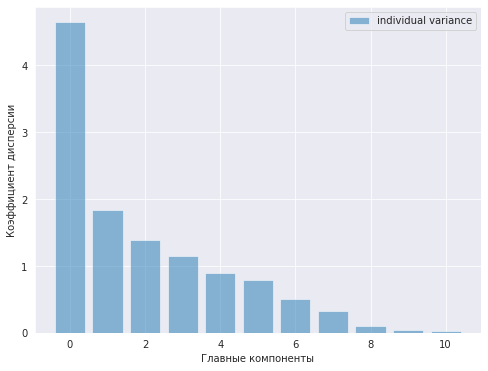
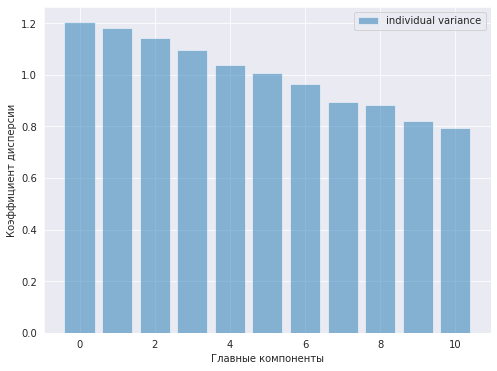


Рисунок 15 – Главные компоненты второй подвыборки

Рисунок 14 – Главные компоненты первой подвыборки

1. *Практическая часть*
   1. *Предобработка данных*

Разные модели требуют разной предобработки данных с учетом их свойств. В общем можно выделить несколько основных правил:

* разбалансированные наборы данных перед решением задач классификации потребуется выровнять,
* метод Главных компонент для больших мультиклассовые наборов данных улучшает работу многих алгоритмов, также сглаживает негативный эффект от выбросов,
* Категориальные переменные должны быть закодированы LabelEncoder (машинного обучения модели)
* Метки классов для нейросетей должны быть закодированы через OneHotEncoder,
* мультиклассовые метки высокой размерности необходимо векторизовать,
* в случае применения моделей классификации для целей регрессии необходимо закодировать непрерывную целевую переменную методом LabelEncoder,
* нормализация данных рекомендуется в случае различного масштаба значений мультипризнаков. Для нейросетей она обязательна для большинства функций активации,
* стандартизация данных при наличии аномальных выбросов предпочтительна,
* стандартизация предпочтительна при использовании алгоритмов, которые основываются на измерении расстояний, например, k ближайших соседей или метод опорных векторов (SVM).
* стандартизация неприемлема для моделей, у которых функции не принимают отрицательные числа,
* масштабирование может искажать некоторые свойства и связи переменных, особенно когда их распределение далеко от нормального. В этом случае некоторые алгоритмы, например деревья решений, не требуют масштабирования.

По результатам разведочного анализа необходимо выполнить в исходном наборе данных следующие преобразования:

1. удалить обнаруженные выбросы,
2. отсечь шумы (строки ниже 22),
3. нормализовать данные,
4. искусственно увеличить размер подвыборки методом бутстрэпа и аугментации (для нейросети).

Сравнительное распределение признаков до и после нормализации представлено на рисунках 16 и 17 соответственно. Описательная статистика до и после преобразования представлена в таблицах 3 и 4. Как видно масштаб значений признаков выровнялся.

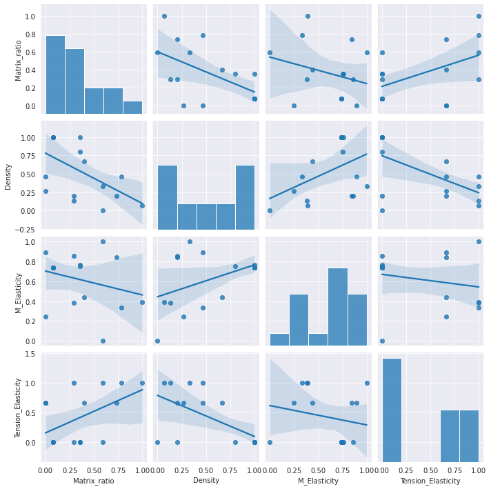
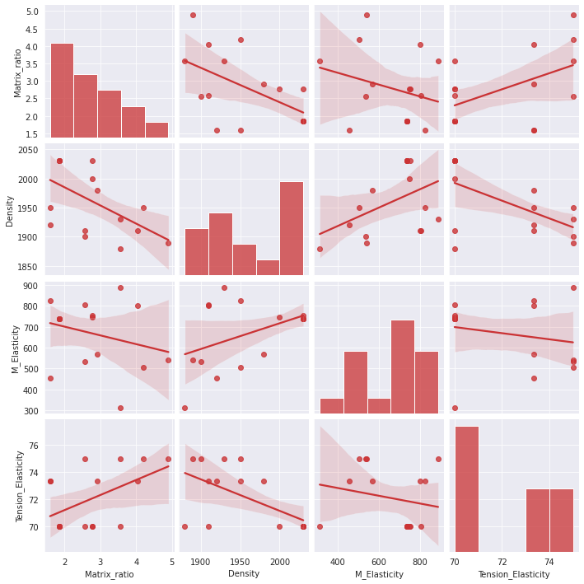
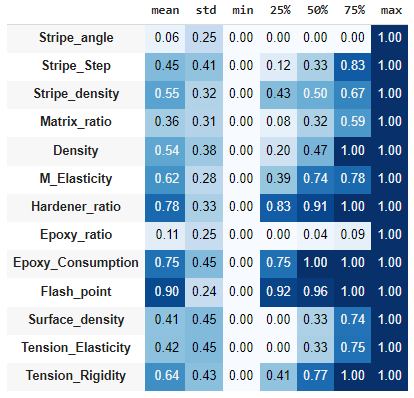
 

Рисунок 16 – Распределение до нормализации

Рисунок 17 – распределение после нормализации



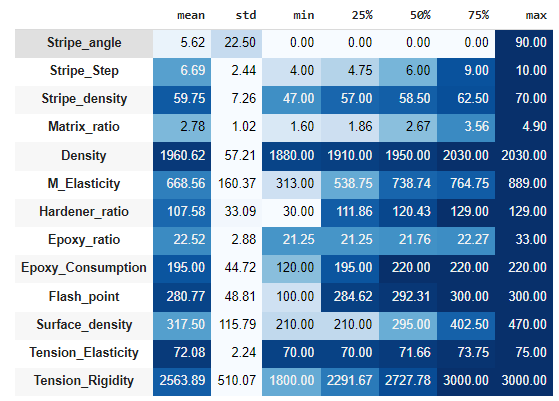


Таблица 3 – статистика до нормализации

Таблица 4 – статистика после нормализации

Для обучения нейросети потребуется искусственно увеличить размер выборки используя метод Pandas.sample с заменой и перемешиванием выбираемых данных. В результате получена новая выборка размерностью 200 на 13.

* 1. *Разработка и обучение модели*

Опираясь на разведочный анализ данных, для сравнения выбраны следующие модели машинного обучения:

* Регрессия Лассо с Полиноминальной (метод наименьших квадратов),
* Случайный лес,
* Метод Ближайших соседей,
* Метод опорных векторов,
* Логистическая регрессия

По результату их сравнения будет выбрана для последующего использования модель с наилучшими результатами. В качестве метрики качества модели будет использоваться коэффициент детерминации R^2.

Регрессия чувствительна к качеству выборки, особенно на маленькой выборке, поэтому используем регрессию с регуляризацией Лассо. Подготовим параметры для регрессии Лассо на выбранных данных: настроим коэффициент Альфа и определим степень полинома. Выбор коэффициента Альфа для Регрессии Лассо будет произведен с помощью перекрестной проверки методом координатного спуска, для сокращения кода использован метод создания конвейера: model = make\_pipeline(MinMaxScaler(), LassoCV(cv=5, max\_iter=5000)).fit(data, target\_func).

Результат подбора Альфа представлен на графике рисунка 18.

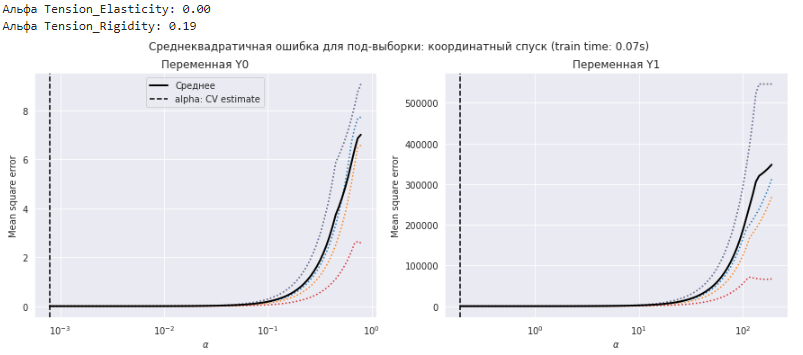


Рисунок 18 – Зависимость роста среднеквадратической ошибки от Альфа

Настройка параметров полиноминальной регрессии заключается в подборе степени полинома. Результат подбора представлен в таблице 5. Оптимальная степень полинома 1.

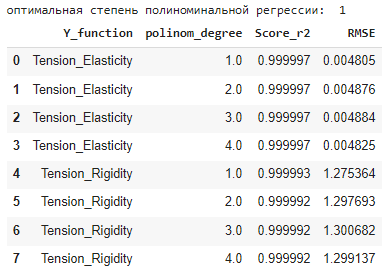


Таблица 5 – Подбор степени полинома

Деревья решений в классификации и регрессии очень похожи, поскольку работают путем построения деревьев с узлами «да/нет». Однако в то время, как конечные узлы классификации приводят к одному значению класса (например, 1 или 0 для задачи бинарной классификации), деревья регрессии заканчиваются значением в непрерывном режиме (например, 4593,49 или 10,98). Регрессия дерева решений не требует шкалирования данных на входе, так как при этом могут теряться некоторые свойства признаков. Подбор оптимальных параметров производится последовательным использованием алгоритмов RandomizedSearchCV и GridSearchCV.

RandomizedSearchCV - это случайный выбор значений параметров в пределах заданного диапазона параметров, а затем поиск наилучшего. Получен результат:

{'bootstrap': True,

'max\_depth': 30,

'max\_features': 'auto',

'min\_samples\_leaf': 1,

'min\_samples\_split': 2,

'n\_estimators': 1400}

GridSearchCV - объединить все заданные значения параметров, а затем найти лучшее. Подобранные параметры будут использованы при последующем построении модели:

{'bootstrap': True,

'max\_depth': 20,

'max\_features': 'auto',

'min\_samples\_leaf': 1,

'min\_samples\_split': 2,

'n\_estimators': 1200}

Принцип, лежащий в основе методов ближайшего соседа, состоит в том, чтобы найти предопределенное количество обучающих выборок, ближайших по расстоянию к новой точке, и предсказать метку по ним. Количество выборок может быть заданной пользователем константой (обучение k-ближайшего соседа) или изменяться в зависимости от локальной плотности точек (обучение соседей на основе радиуса). Расстояние, как правило, может быть любой метрической мерой: стандартное евклидово расстояние является наиболее распространенным выбором. Так как в основе алгоритма расчет расстояний, то понадобится стандартизация данных на входе. Для гиперпараметров применен тот же метод GridSearchCV.

{'algorithm': 'auto',

'leaf\_size': 1,

'n\_neighbors': 5,

'p': 1,

'weights': 'distance'}

Метод опорных векторов основан на построении разделяющей гиперплоскости, эффективно работает как с большими, так и малыми выборками когда количество признаков превышает количество их значений. Подобраны следующие гиперпараметры.

{'degree': 1, 'gamma': 'scale', 'kernel': 'linear'}.

Логистическая регрессия в общем случае является методом классификации. Но ее можно применить в качестве регрессора, закодировав целевую переменную через LabelEncoder().

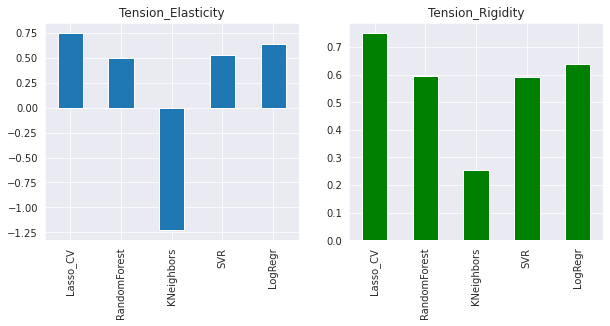
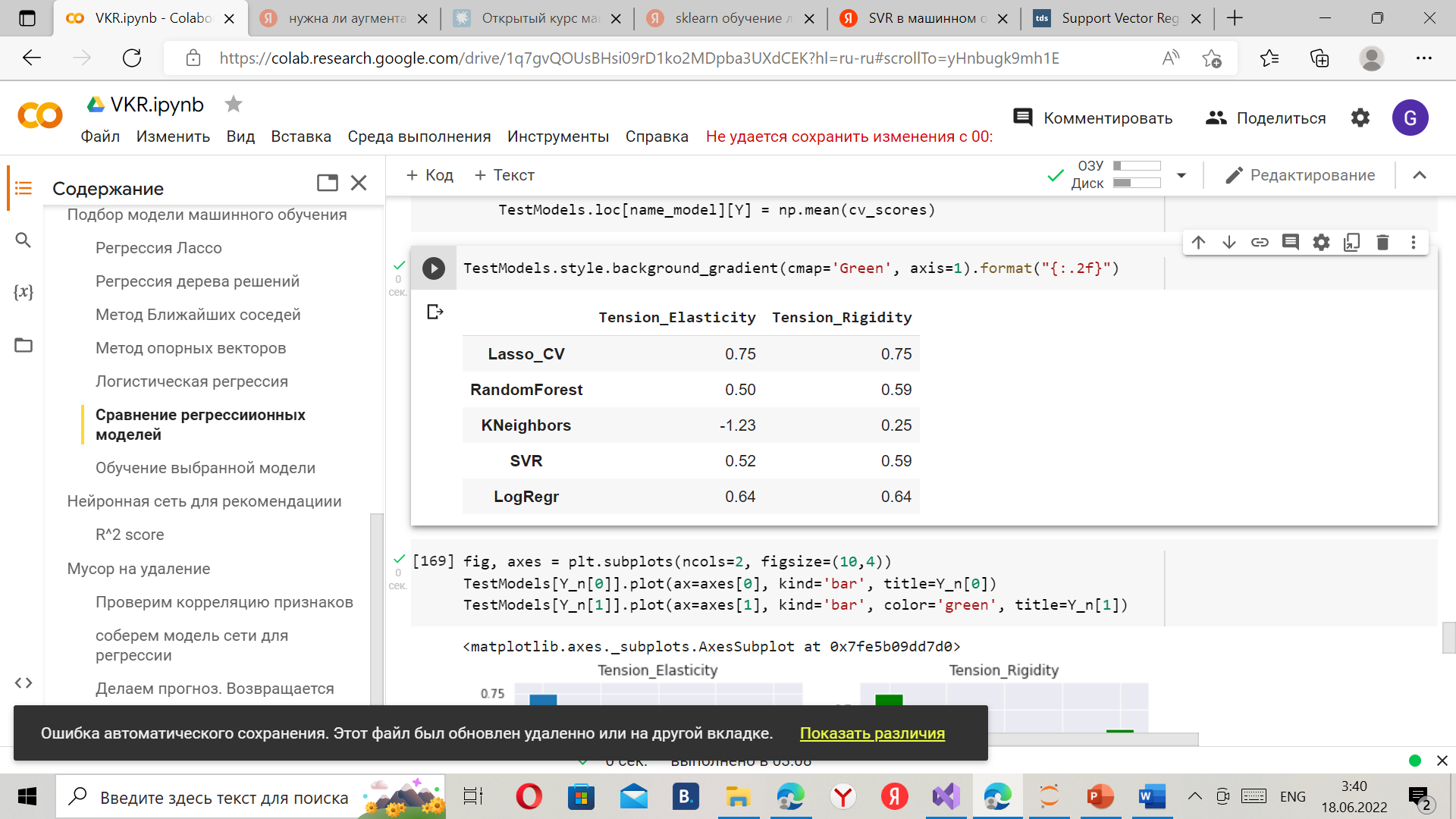
* 1. *Тестирование модели*

Для выборок малого размера повышение точности оценки модели может быть достигнуто методом кросс-валидации. Применен метод cv\_scores = cross\_val\_score(model, data, tgt\_func, cv=4, scoring='r2')

Результат сравнения коэффициентов детерминации по обучению разных моделей представлен в таблице 6 и на рисунке 19. Наилучший результат показывает модель Лассо с полиноминальной регрессией.

Таблица 6 – коэффициенты детерминации моделей

Рисунок 19 – коэффициенты детерминации моделей



* 1. *Нейросеть для рекомендации соотношения матрица и наполнителя*

В данной задаче требуется написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Для ее выполнения разделим данные на признаки и целевые переменные. В качестве целевой переменной берем 'Matrix\_ratio'. Применим нормализацию данных на входе. Особенностью нейросетей регрессии является то, что на выходе должен быть один нейрон с линейной функцией. На скрытых слоях рекомендуется использовать функцию ReLu, для обработки нелинейности включена функция Softmax. В качестве оптимизатора использована функция среднеквадратичное распространение корня Rmsprop, для вычисления ошибки среднеквадратическая функция ошибки MSE, в качестве метрики оценки качества функция средней абсолютной ошибки MAE. Структура слоев представлена в таблице 7.

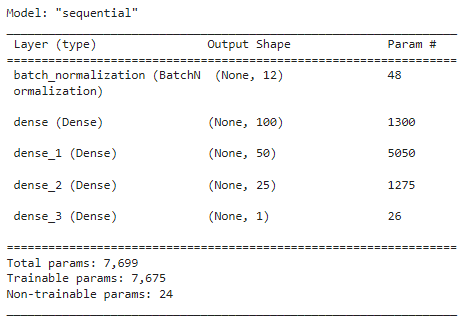


Таблица 7 – Структура слоев нейросети

При искусственном увеличении выборки, после ее разбиения на тестовую и тренировочную, некоторое количество строк продублируется между ними, что может привести к переобучению. Для компенсации этого применим метод K-Folds кросс-валидация с разбивкой на 5 групп. Он создает процесс, в котором каждый образец данных будет включен в набор тестов на некоторых этапах. Ход обучения контролируем с помощью анализа сходимости ошибки на тренировочной и тестовой выборках. Таким способом подбирается оптимальное количество эпох обучения чтобы избежать как недоученности, так и переобученной модели. Результат сходимости результатов обучений представлен на рисунке 20. Оптимальным является 200 эпох обучений.

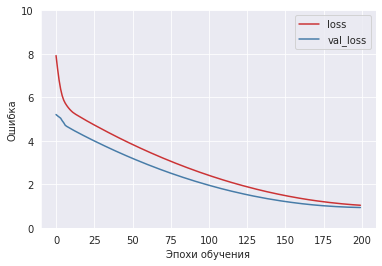


Рисунок 20 – эволюция ошибки модели с увеличением эпох обучения

* 1. *Разработка приложения*

Разработка приложения выполнена с помощью фрэйм-ворка Flask в среде Visual Studio. Flask это небольшой и легкий веб-фреймворк, написанный на языке Python, предлагающий полезные инструменты и функции для облегчения процесса создания веб-приложений с использованием Python. Он обеспечивает гибкость и является более доступным фреймворком для новых разработчиков, так как позволяет создать веб-приложение быстро, используя только один файл Python.

* 1. *Создание удаленного репозитория и загрузка результатов работы*

Чтобы иметь возможность совместной работы над каким-либо Git-проектом, необходимо управлять удалёнными репозиториями. Удалённые репозитории — это модификации проекта, которые хранятся в интернете или ещё где-то в сети. Их может быть несколько, каждый из которых, как правило, доступен для вас либо только на чтение, либо на чтение и запись. Совместная работа включает в себя управление удалёнными репозиториями и помещение (push) и получение (pull) данных в и из них тогда, когда нужно обменяться результатами работы.

git remote add [сокращение] [url]