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 Насыров Артур Ильдусович

Москва, 2023

# Содержание

Введение .......................................................................................................... 3

1. Аналитическая часть .................................................................................. 4

1.1. Описание задачи и исходных данных ...................................................4

1.2. Разведочный анализ и визуализация данных........................................4

2. Разработка моделей машинного обучения...........................................….12

2.1 Предобработка данных ..........................................................................12

2.2. Описание используемых методов........................................................ 16

2.2.1 Гребневая (Ridge) регрессия........................................................... 16

2.2.2 Метод опорных векторов для регрессии....................................... 16

2.2.3 Метод k-ближайших соседей.......................................................... 17

2.2.4 Случайный лес................................................................................. 17

2.2.5 Extremely randomized trees............................................................. 18

2.2.6 Градиентный бустинг...................................................................... 18

2.7 Voting Regressor................................................................................. 19

2.2.8 Нейронная сеть............................................................................... 19

2.3 Создание моделей .............................................................................. 20

2.3.1 Разбиение на тренировочную и тестовые выборки.................... 20

2.3.2. Поиск гиперпараметров c кросс-валидацией............................. 20

2.3.3 Метрики качества моделей..............................................................22

2.4 Тестирование моделей........................................................................... 23

3. Разработка приложения ............................................................................ 32

4. Создание удаленного репозитория............................................................. 33

Заключение ........................................................................................................ 34

Список используемой литературы и веб ресурсы…………………………... 35

# Введение

Тема данной работы - прогнозирование конечных свойств новых материалов (композиционных материалов).

Композиционный материал - это сочетание двух (или более) химически разнородных компонентов (фаз) с четкой границей раздела между ними. В большинстве композитов (за исключением слоистых) компоненты можно разделить на матрицу и включенные в нее элементы .

В изготовление современных композитов используются следующие материалы: полимеры, керамика, стеклянные и углеродные волокна.

Также возможно получить композиты с уникальными эксплуатационными свойствами. Но тут не все так просто: даже если мы и знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов и прогнозирование характеристик.

Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

Стоимость производства композитного материала высока. Зная характеристики компонентов, невозможно рассчитать свойства композита. Значит для получения заданных свойств требуется большое количество испытаний различных комбинаций. Сократить время и затраты на создание определенного материала могла бы помочь система поддержки производственных решений, построенная на принципах машинного обучения .

# 1. Аналитическая часть

## 1.1 Описание задачи и исходных данных

В данной работе исследуется композит с матрицей из базальтопластика и нашивками из углепластика.

Целью настоящей работы является прогнозирование характеристик «Модуля упругости при растяжении, ГПа» и «прочность при растяжении, МПа» и рекомендовать «соотношение матрица- наполнитель» материалов при помощи методов машинного обучения и построение моделей.

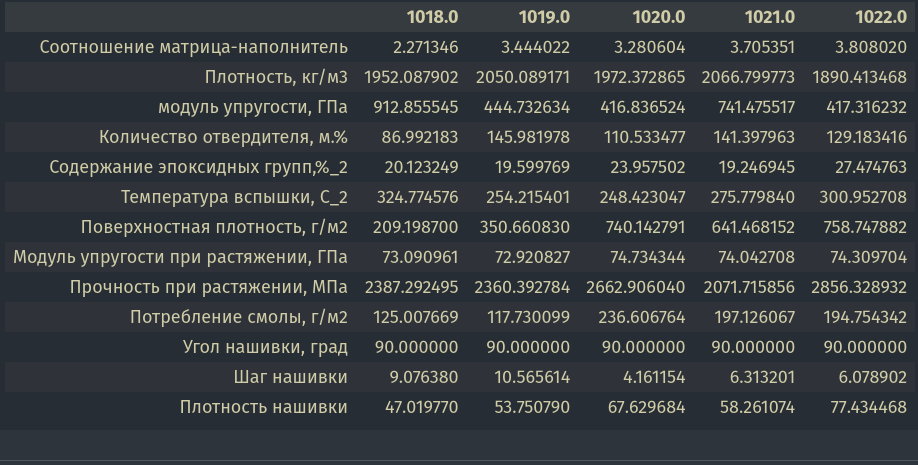
Данные предоставлены Центром НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

Имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим и т.д.). Один набор данных «X\_bp» содержит (10 переменных вещественного типа) содержащей 1023 строки данных. Второй набор данных «X\_nup» (3 переменные вещественного типа) содержащей 1040 строк данных. Данные представлены в Excel – файлах. Интересующие выходные характеристики (3 выходные переменные) содержатся в наборе «X\_bp». Для анализа данных будет использоваться язык программирования Python **.** Этому способствует простота языка, а также большое разнообразие открытых библиотек.

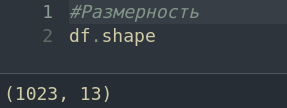
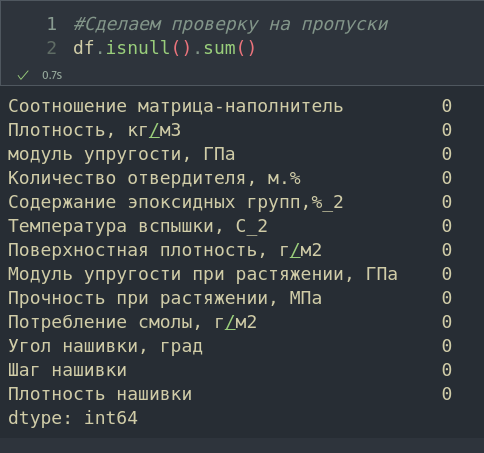
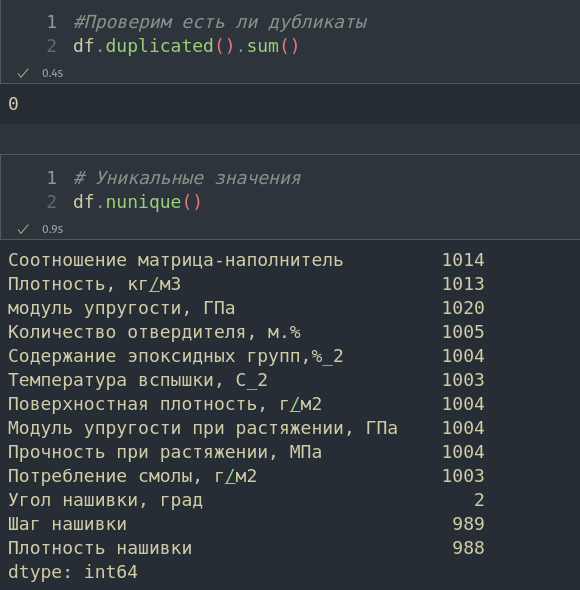
1.2 Разведочный анализ и визуализация данных

При разведочном анализе учитывается и сравнивается большое число признаков, а для поиска закономерностей используются самые разные методы. Результаты разведочного анализа помогут в разработке наилучшей стратегии углубленного анализа, выдвижение гипотез, уточнение особенностей применения тех или иных математических методов и моделей.

Известно, что файлы требуют объединения с типом INNER по индексу. После объединения часть строк из файла X\_nup была отброшена. И дальнейшие исследования проводим с объединенным датасетом.

Рисунок 1 - Наименование параметров

Описание признаков объединенного датасета приведено на рисунке 2. Все признаки имеют тип float64. Пропусков и дубликатов в данных нет. Объединенный датасет, содержит 13 столбцов и 1023 строк . Все признаки, кроме «Угол нашивки», являются непрерывными, количественными.



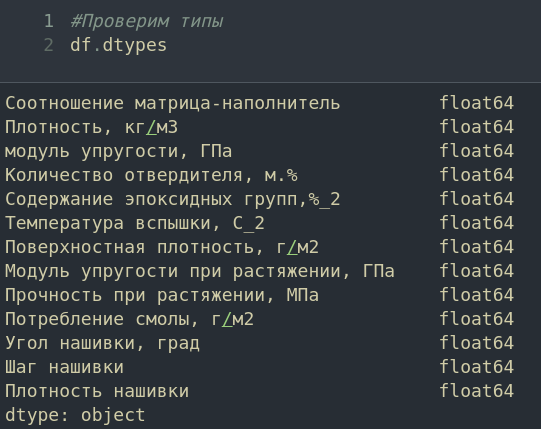
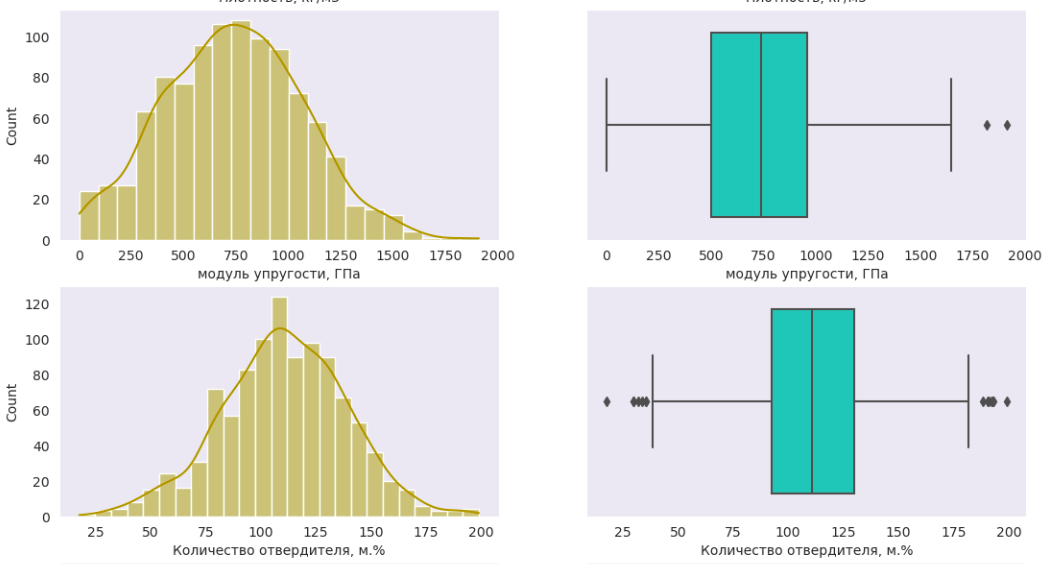
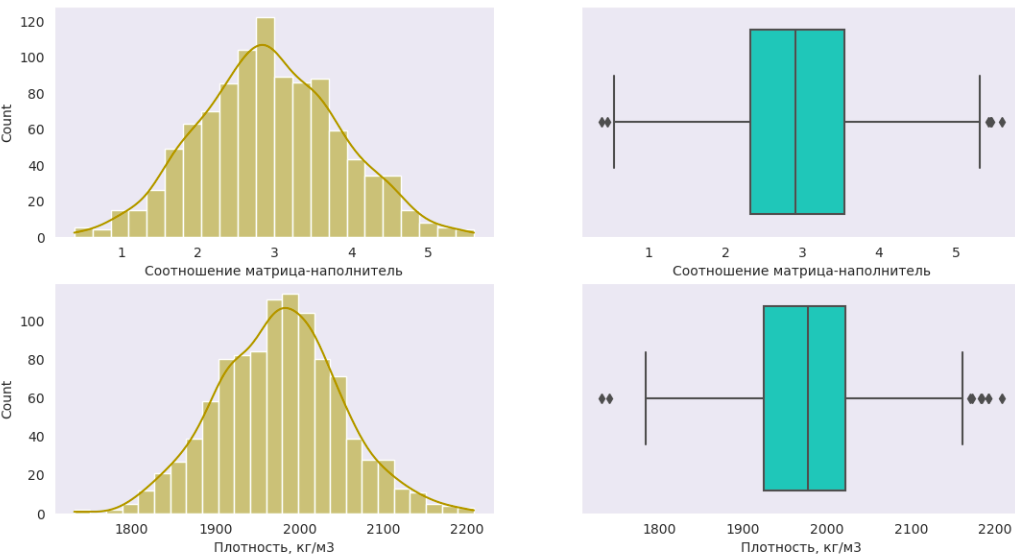
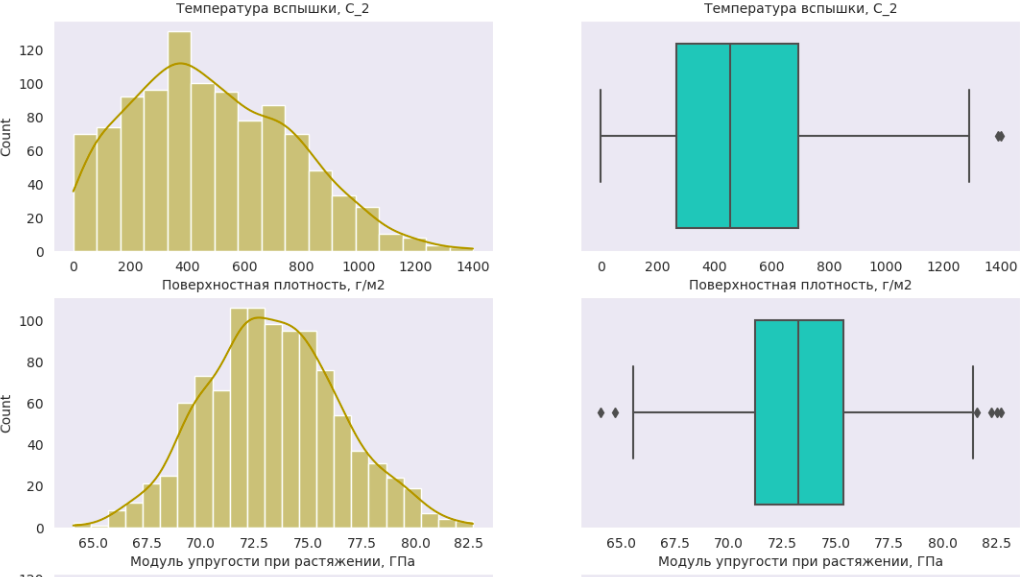
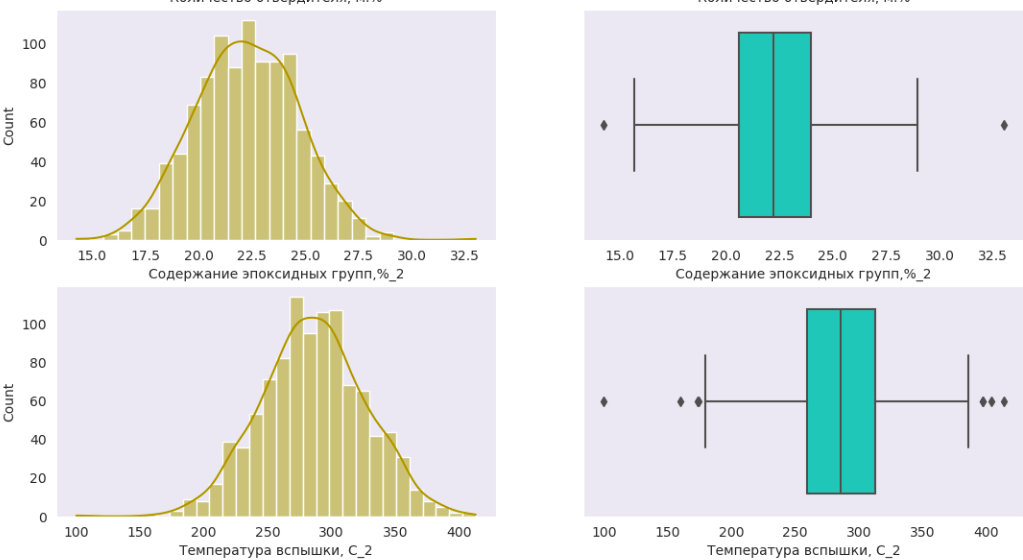
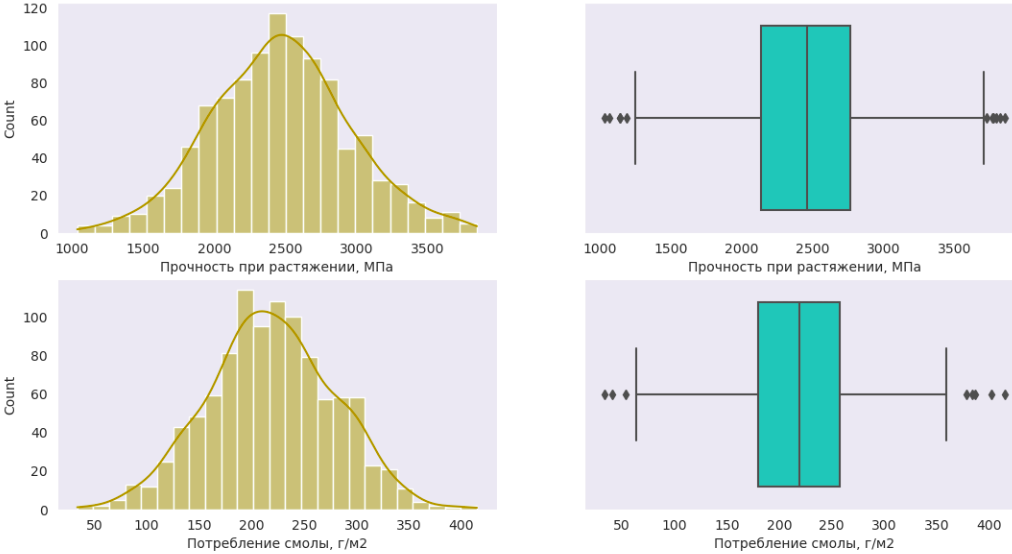
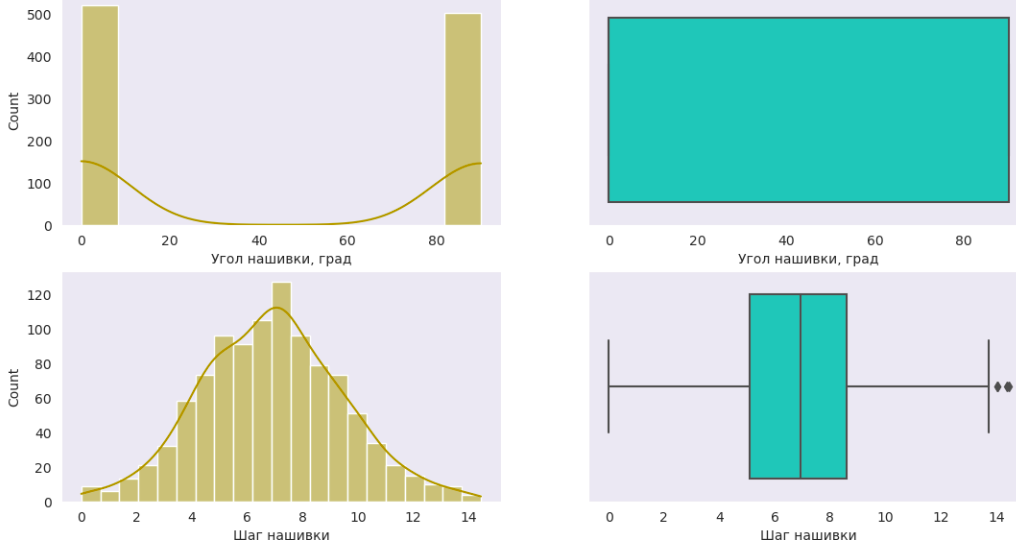


Рисунок 2 — Описание признаков датасета

Гистограммы распределения переменных и диаграммы «ящик с усами» приведены на рисунке 3. 







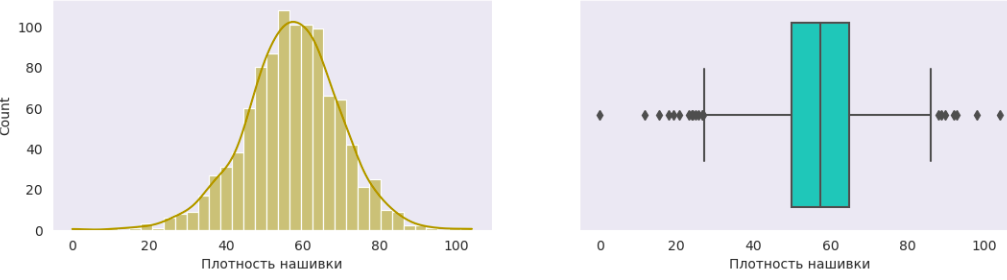


Рисунок 3 - Гистограммы распределения переменных

и диаграммы «ящик с усами»

Известно, что датасет был предварительно подготовлен, поэтому отсутствие пропусков не удивило.

Также нас интересует описательная статистика датасета,которая в численном виде отражает то, что мы видим на гистограммах. Она представлена на рисунке 4. 

Рисунок 4 - Описательная статистика признаков датасета

Также была проведена оценка признаков на нормальное распределение

результат показан на рисунке 5, где видно что все признаки кроме 'Поверхностная плотность, г/м2', 'Угол нашивки, град', 'Плотность нашивки'

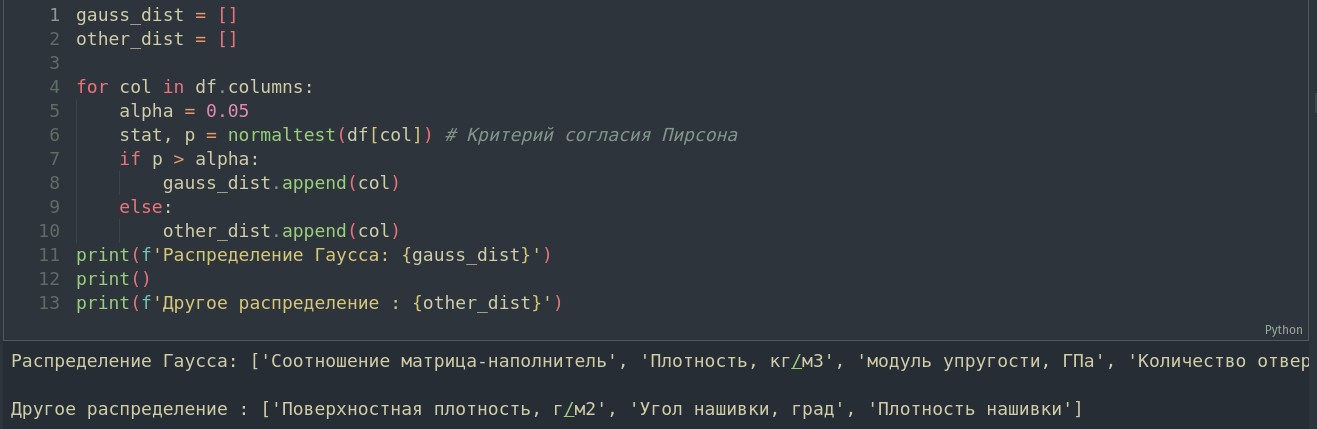
имеют нормальное распределение.

Рисунок 5 - Оценка признаков на нормальное распределение

Для анализа корреляционных признаков были построены корреляционная тепловая карта и диаграмма рассеивания. Разведочный анализ данных показал, что линейной связи между любыми переменными нет, корреляция равна 0 это наглядно видно на рисунках 6 и 7.

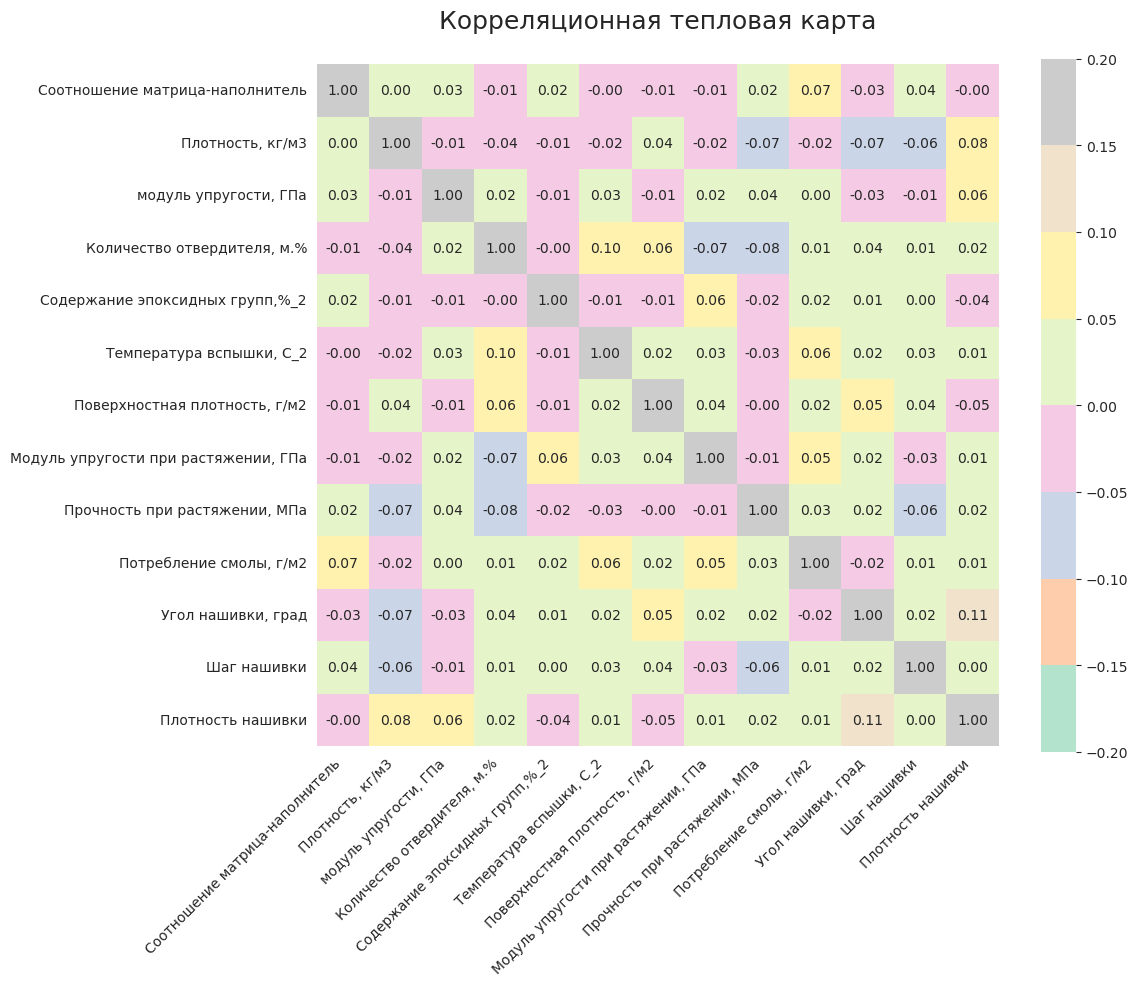


Рисунок 6 - Корреляционная тепловая карта

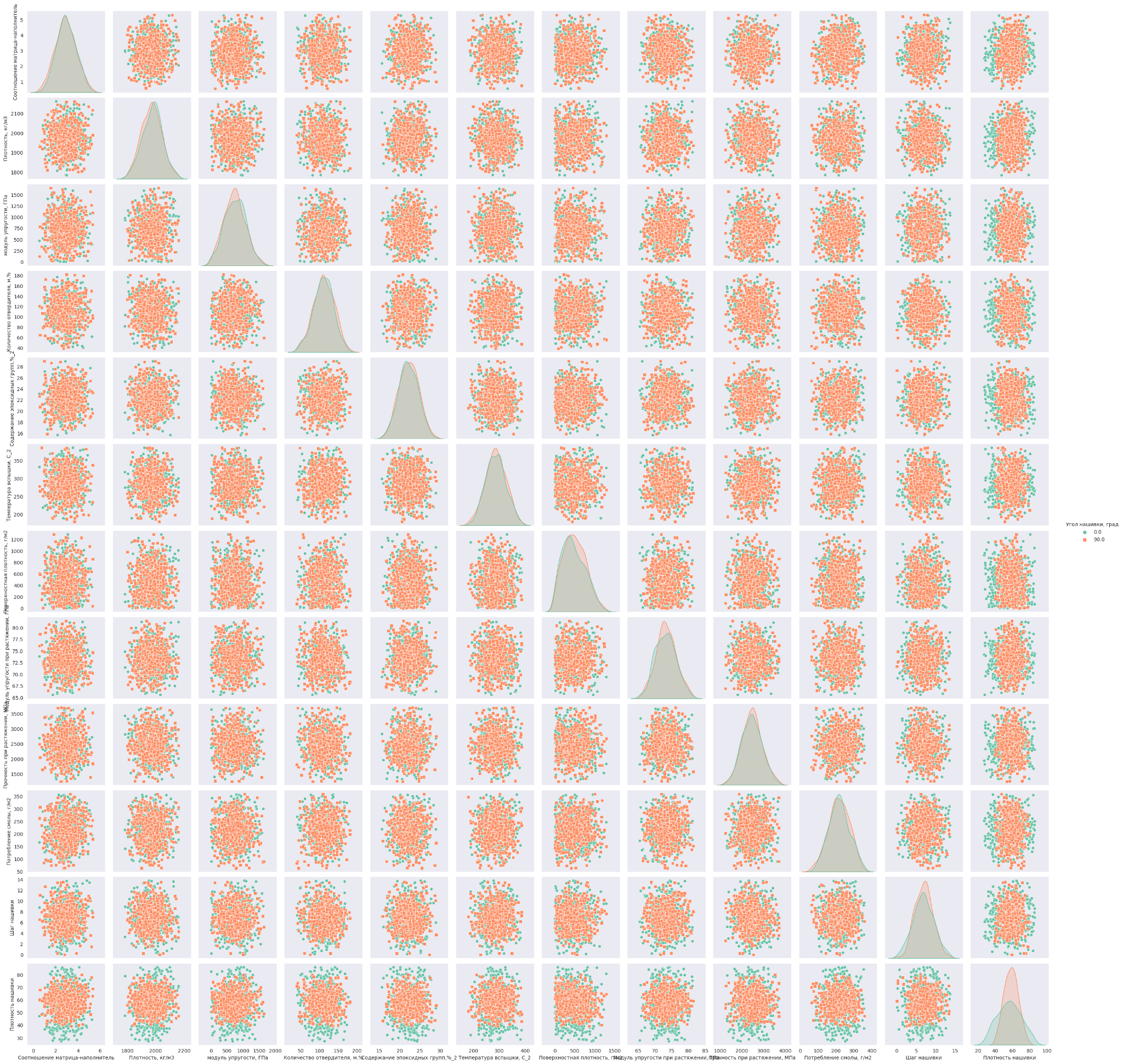


Рисунок 7 - Попарные графики рассеяния точек

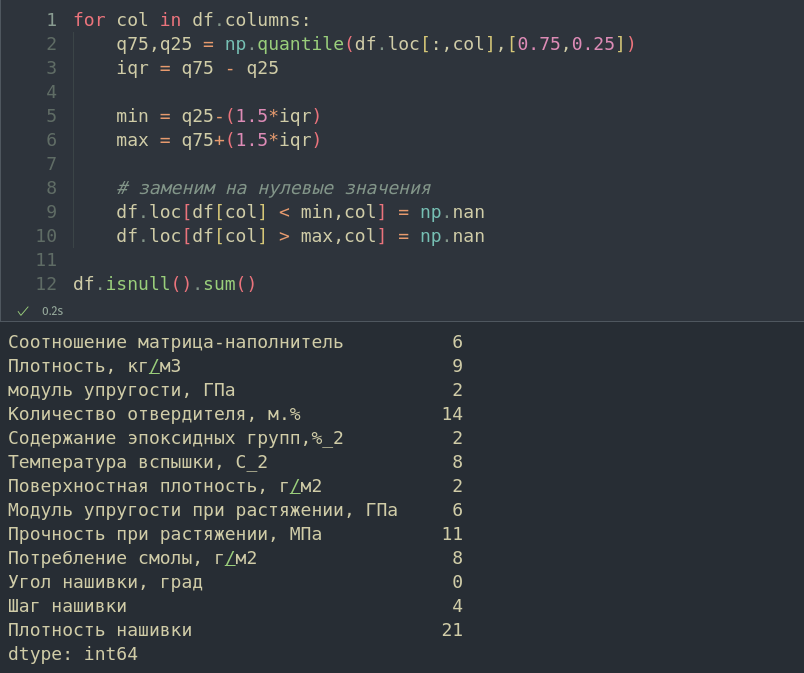
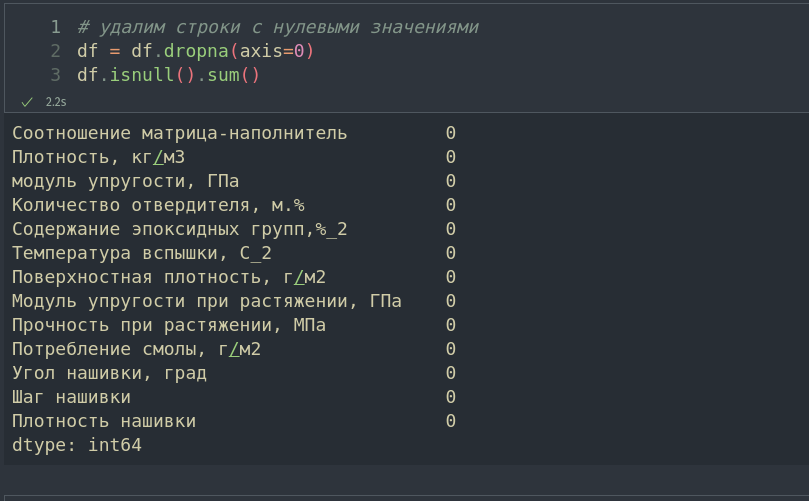
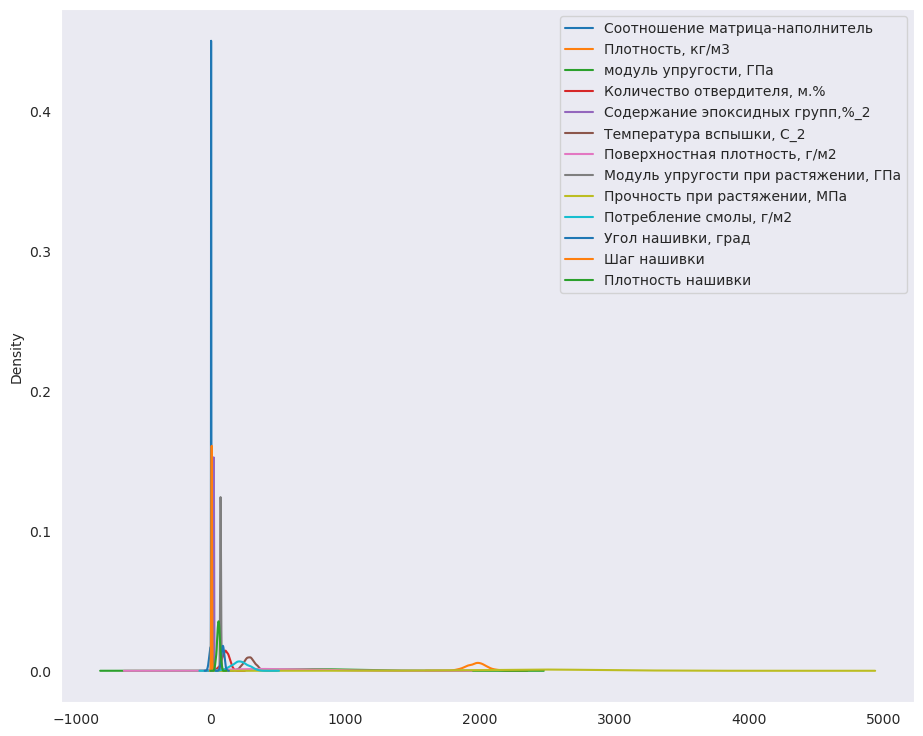
Удаление выбросов осуществлено методом межквартильного расстояния, что показано на рисунке 8.

Рисунок 8 - Удаление выбросов

Оценка плотности ядра на рисунке 9 показала, что данные требуют нормализации.

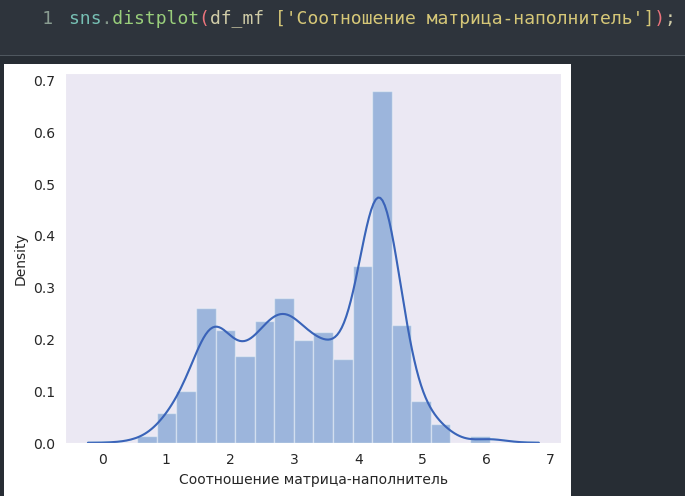
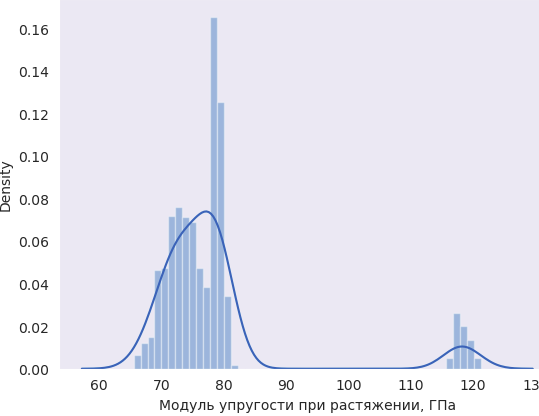
Рисунок 9 - Оценка плотности ядра

2. Разработка моделей машинного обучения

2.1. Предобработка данных

Разведочный анализ данных показал, что линейной связи между любыми переменными нет, корреляция равна 0, данных для выборки недостаточно, не все признаки имеют одинаковое распределение, значения необходимые для прогнозирования, редки и необычны т.е датасет не сбалансирован, то для дальнейшей задачи было решено создать 3 новые датасета для каждой модели и добавить синтетические данные, для этого была использована библиотека ImbalancedLearningRegression.

После предобработки целевые переменные стали иметь следующий вид показанного на рисунке 10



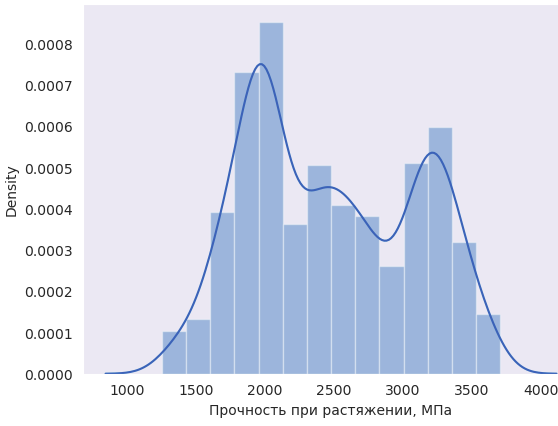
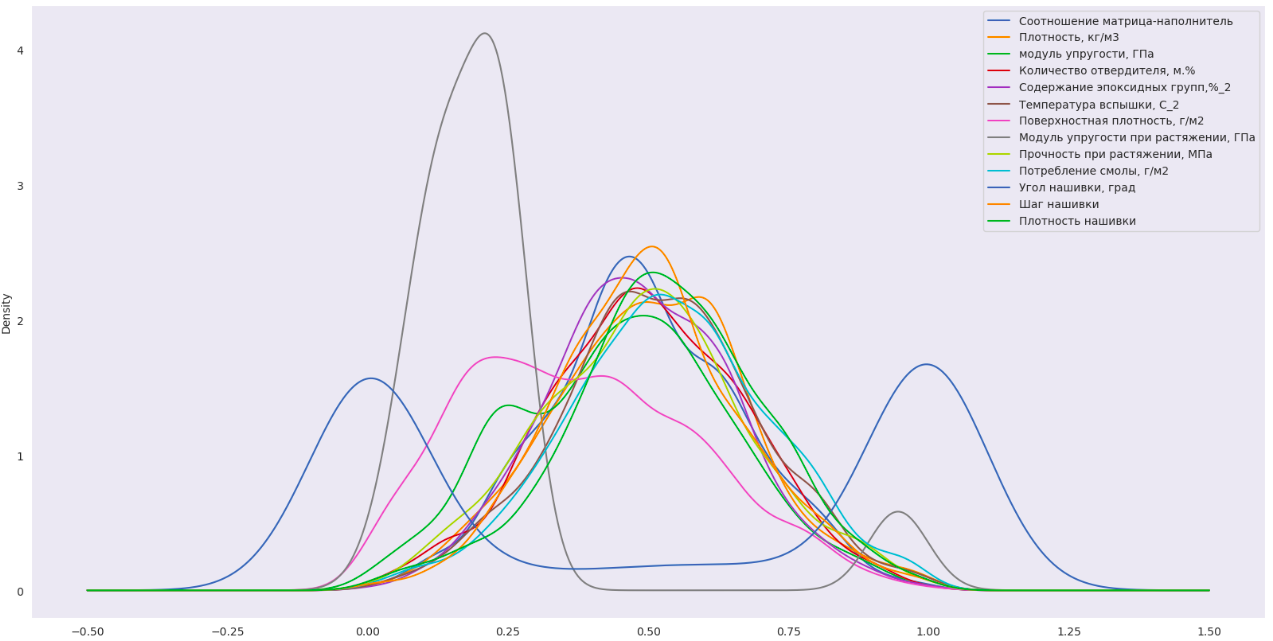
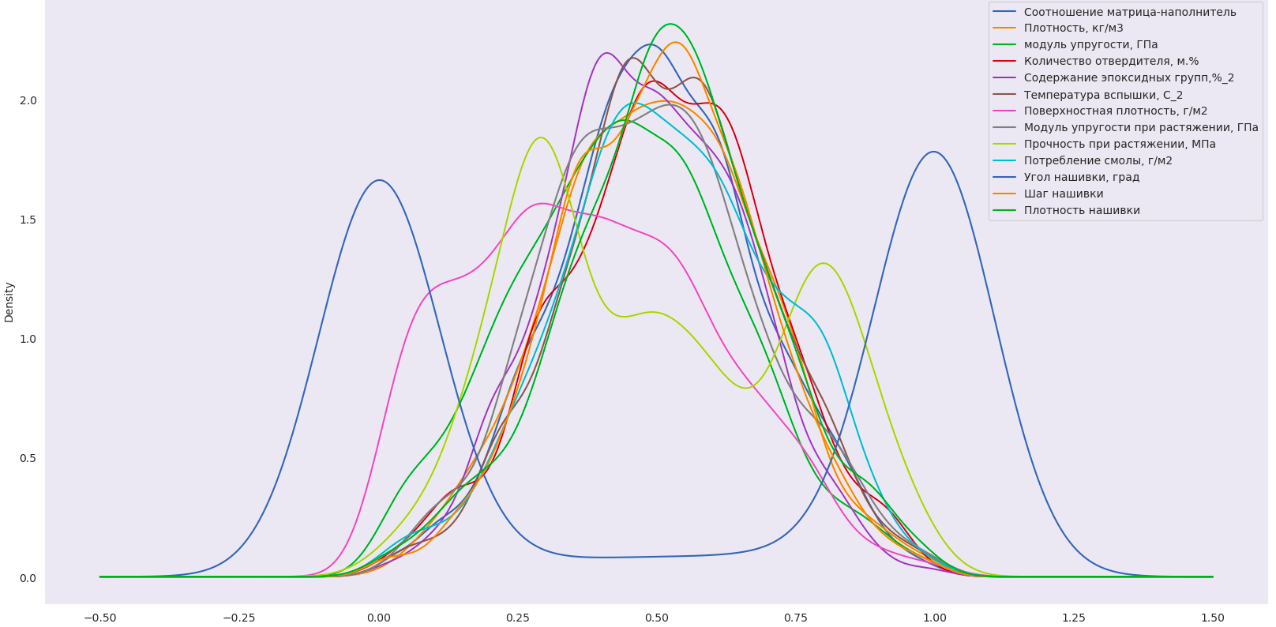


Рисунок 10 - Целевые переменные после добавления синтетических данных

Для нормализации использовался MinMaxScaler() в библиотеке Sklearn. Таким образом мы преобразовали значения в диапазон от 0 до 1.

Теперь плотность ядра имеет следующий вид, что на рисунке 11



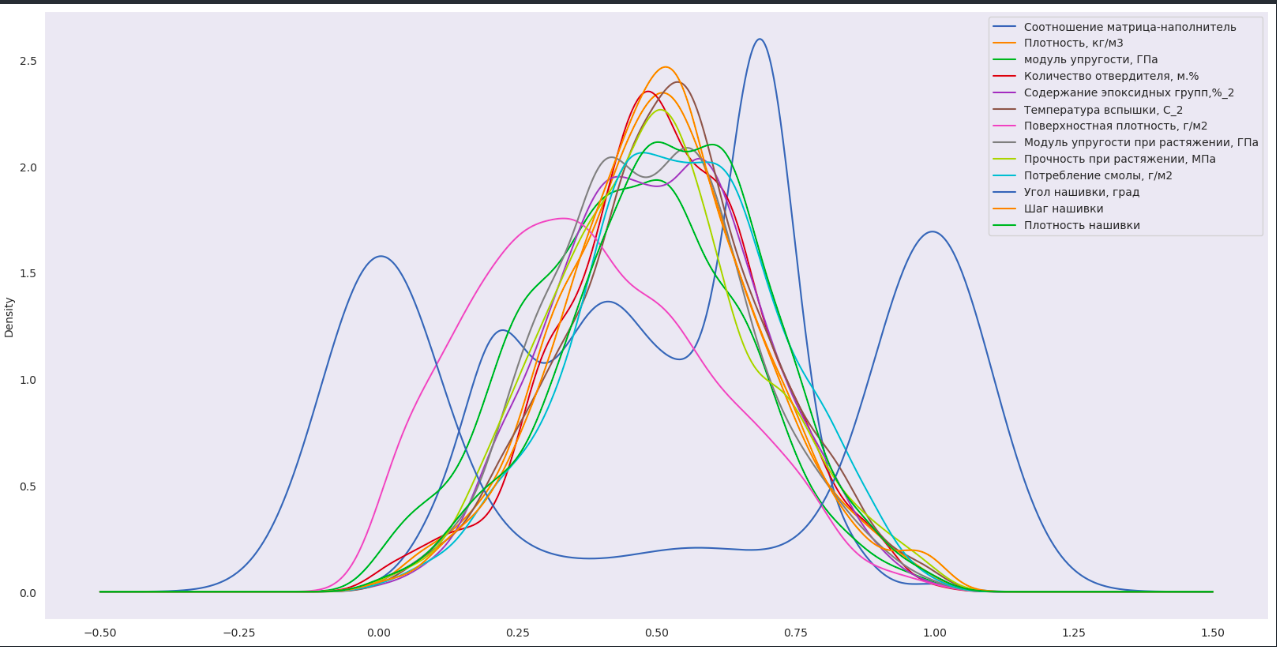


Рисунок 11 - Плотность ядра для 3-х датасетов после нормализации

Описательная статистика после предобработки данных показана на рисунках 11, 12, 13

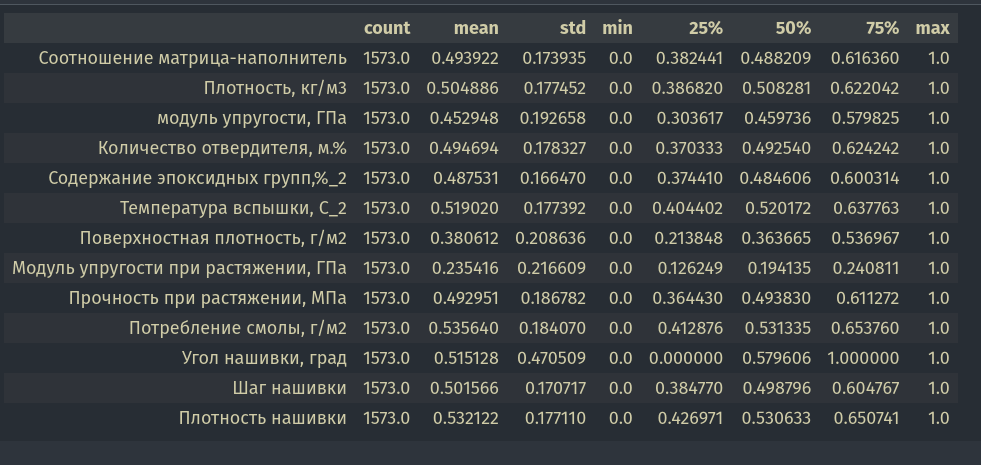


Рисунок 11 - Описательная статистика для датасета с целевой переменной “Модуль упругости при растяжении, ГПа”

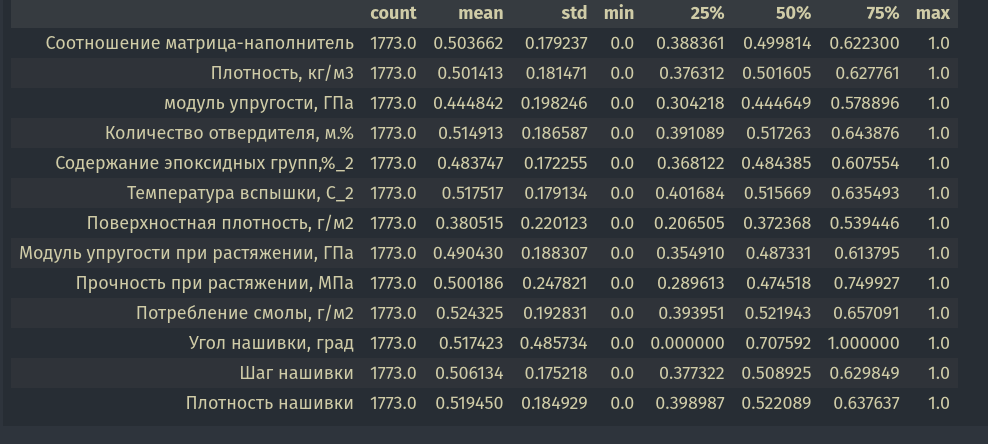


Рисунок 12 - Описательная статистика для датасета с целевой

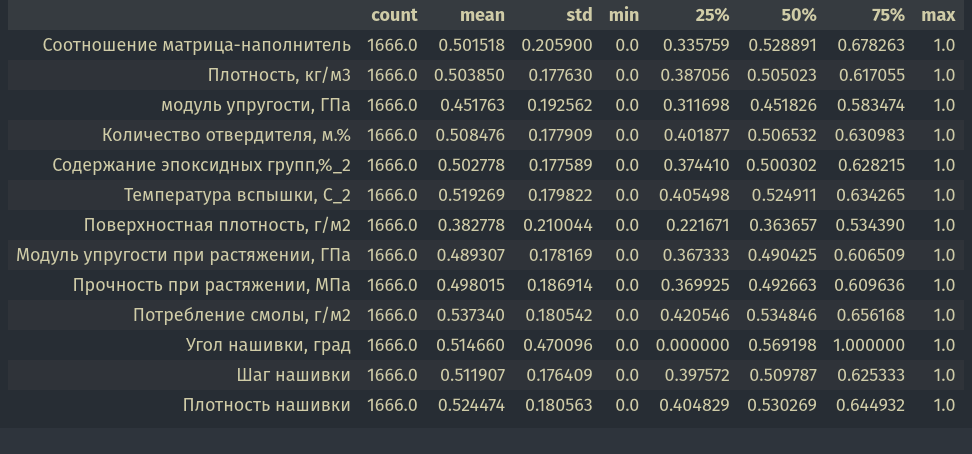
переменной “Прочность при растяжении, МПа”

Рисунок 13 - Описательная статистика для датасета с целевой

переменной “Соотношение матрица-наполнитель”

Для удобства работы, экономии времени и вычислительных ресурсов вычислительной машины, датасеты сохранены в отдельные файлы в формате csv. Дальнейшая работа по обучению моделей будет осуществляться в отдельных 3-х ноутбуках.

## 2.2. Описание используемых методов

Предсказание значений вещественной, непрерывной переменной — это задача регрессии. Эта зависимая переменная должна иметь связь с одной или несколькими независимыми переменными, называемых также предикторами или регрессорами. Регрессионный анализ помогает понять, значение зависимой переменной изменяется при изменении независимых переменных.

## 2.2.1 Гребневая (Ridge) регрессия

Гребневая регрессия — вариация линейной регрессии. Она применяет сжатие и хорошо работает для данных, которые демонстрируют сильную мультиколлинеарность. Гребневая регрессия использует регуляризацию L2, которая взвешивает ошибки по их квадрату, чтобы сильнее наказывать за более значительные ошибки. Регуляризация позволяет интерпретировать модели. Если коэффициент близким к 0 , значит данный входной признак не является значимым. Этот метод реализован в sklearn.linear\_model.Ridge.

## 2.2.2 Метод опорных векторов для регрессии

Метод опорных векторов (support vector machine, SVM) — один из наиболее популярных методов машинного обучения. Он создает гиперплоскость или набор гиперплоскостей в многомерном пространстве, которые могут быть использованы для решения задач классификации и регрессии. Основная идея заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом. Интуитивно, хорошее разделение достигается за счет гиперплоскости, которая имеет самое большое расстояние до ближайшей точки обучающей выборке любого класса. Максимально близкие объекты разных классов определяют опорные вектора.

Если в исходном пространстве объекты линейно неразделимы, то выполняется переход в пространство большей размерности.

Вариация метода для регрессии называется SVR (Support Vector Regression). Реализован в sklearn.svm.SVR.

## 2.2.3 Метод k-ближайших соседей

Метод k-ближайших соседей (k Nearest Neighbors). На интуитивном уровне суть метода проста: посмотри на соседей вокруг, какие из них преобладают, таковым ты и являешься. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Для реализации метода необходима метрика расстояния между объектами. Используется, например, евклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных. Реализован в sklearn.neighbors.KneighborsRegressor.

## 2.2.4 Случайный лес

Случайный лес (RandomForest) — представитель ансамблевых методов, разновидность бэггинга.

Каждое дерево в ансамбле строится из выборки, взятой с заменой (т. е. начальной выборки) из обучающей выборки. Кроме того, при разделении каждого узла во время построения дерева наилучшее разделение находится либо из всех входных признаков, либо из случайного подмножества размера max\_features.

Цель случайности — уменьшить дисперсию оценки леса. Действительно, отдельные деревья решений обычно демонстрируют высокую дисперсию и имеют тенденцию к переобучению. Внедренная случайность в лесах дает деревья решений с несколько несвязанными ошибками прогнозирования. Взяв среднее значение этих прогнозов, некоторые ошибки могут быть уравновешены. Случайные леса обеспечивают снижение дисперсии за счет объединения разных деревьев, иногда за счет небольшого увеличения смещения. Использую реализацию sklearn.ensemble.RandomForestRegressor и sklearn.ensemble.ExtraTreesRegressor

## 2.2.5 Extremely randomized trees

* 1. Как и в случайных лесах, используется случайное подмножество объектов-кандидатов, но вместо поиска наиболее отличительных пороговых значений пороговые значения выбираются случайным образом для каждого объекта-кандидата, и лучший из этих случайно сгенерированных пороговых значений выбирается в качестве правила разделения. Обычно это позволяет еще немного уменьшить дисперсию модели за счет чуть большего увеличения смещения. Реализация sklearn.ensemble.ExtraTreesRegressor

## 2.2.6 Градиентный бустинг

Градиентный бустинг (GradientBoosting) — еще один представитель ансамблевых методов.

В отличие от случайного леса, где каждый базовый алгоритм строится независимо от остальных, бустинг воплощает идею последовательного построения линейной комбинации алгоритмов. Каждый следующий алгоритм старается уменьшить ошибку предыдущего.

Чтобы построить алгоритм градиентного бустинга, нам необходимо выбрать базовый алгоритм и функцию потерь или ошибки (loss). Loss-функция – это мера, которая показывает насколько хорошо предсказание модели соответствуют данным. Используя градиентный спуск и обновляя предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых loss минимальна.

Бустинг, использующий деревья решений в качестве базовых алгоритмов, называется градиентным бустингом над решающими деревьями. Он отлично работает на выборках с неоднородными данными и способен эффективно находить нелинейные зависимости в данных различной природы. На настоящий момент это один из самых эффективных алгоритмов машинного обучения.

В этой работе я использую реализацию градиентного бустинга из библиотеку XGBoost и CatBoost от компании Яндекс.

2.7 Voting Regressor

Идея VotingRegressor состоит в том, чтобы объединить концептуально разные регрессоры машинного обучения и вернуть средние прогнозируемые значения. Такой регрессор может быть полезен для набора одинаково хорошо работающих моделей, чтобы сбалансировать их индивидуальные недостатки.

* 1. Реализация sklearn.ensemble.VotingRegressor

## 2.2.8 Нейронная сеть

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Вычислительная единица нейронной сети — нейрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учетом значимости (веса) каждого входа.

Смещение – это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения.

Также у нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть. Примеры активационных функций: relu, сигмоида.

У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

* входной слой — его размер соответствует входным параметрам;
* скрытые слои — их количество и размерность определяем специалист;
* выходной слой — его размер соответствует выходным параметрам.

Прямое распространение – это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением.

Прогнозируемое значение сравниваем с фактическим с помощью функции потери. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась.

Для обновления весов в модели используются различные оптимизаторы.

Количество эпох показывает, сколько раз выполнялся проход для всех примеров обучения.

В этой работе я использую библиотеку TensorFlow.

* 1. 2.3 Создание моделей

## 2.3.1 Разбиение на тренировочную и тестовые выборки

* 1. Для этого были созданы отдельные новые ноутбуки для каждой модели.Загруженные датасеты были преобразованы в массивы Numpy для лучшей оптимизации. Далее разбили массивы на тренировочные и тестовые в соотношении 0.7 на тренировочную и 0.3 на тестовую методом train\_test\_split.

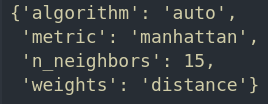
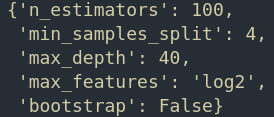
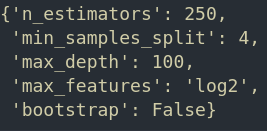
## 2.3.2. Поиск гиперпараметров c кросс-валидацией

## Поиск гиперпараметров был осуществлен c кросс-валидацией по сетке методом GridSearchCV из Sklearn для RidgeRegression и KneighborsRegressor. Для остальных методов использовалась специальная библиотека Optuna для более быстрого вычисления поскольку gridsearch слишком долго подбирает гиперпараметры и не подходит для ансамлевых методов в особенности бустига.

* 1. Для прогнозирования «Модуля упругости при растяжении, гпа » поиск по сетке с перекрестной проверкой показал:
  2. Ridge regression

Support Vector Regrresion



Kneighbors regression Random forest regression Extra tree regression

XGBoost Regressor CatBoost Regressor

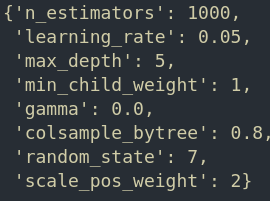
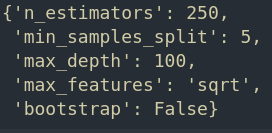
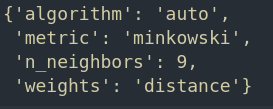
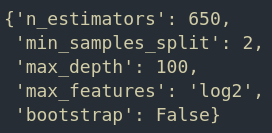


Рисунок 14 - Лучшие гиперпараметры для «Модуля упругости при растяжении, гпа »

* + 1. Для прогнозирования «Прочности при растяжении, мпа» поиск по сетке с перекрестной проверкой показал:
    2. Ridge regression

Support Vector Regrresor

Kneighbors regressor Random forest regressor Extra tree regressor



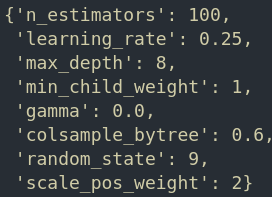
XGBoost Regressor CatBoost Regressor

Рисунок 15 - Лучшие гиперпараметры для «Прочности при растяжении, мпа»

2.3.3 Метрики качества моделей

Существует множество различных метрик качества, применимых для регрессии. В этой работе я использую:

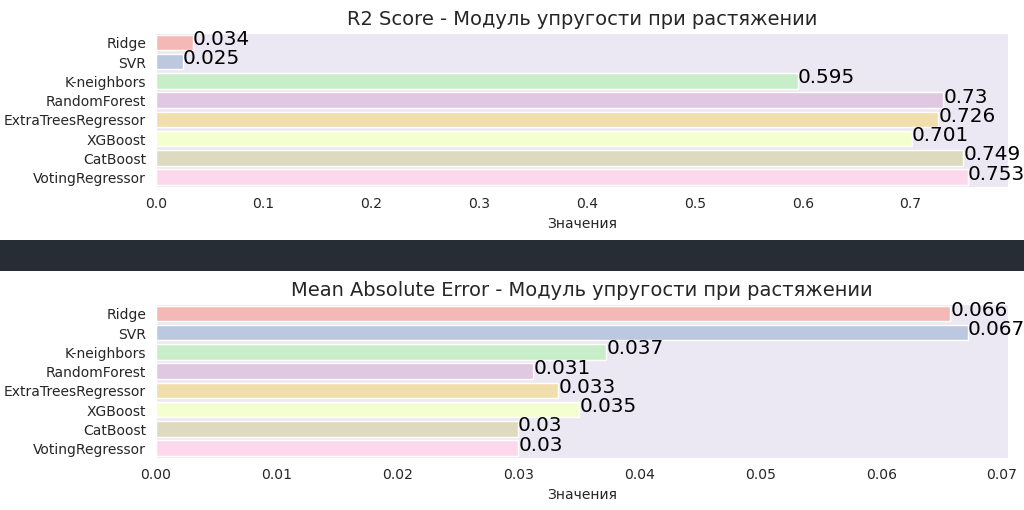
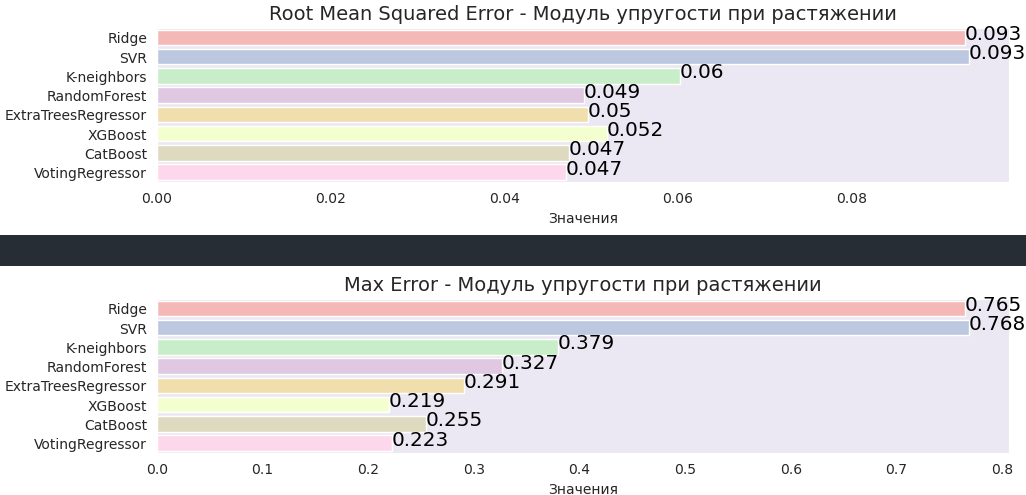
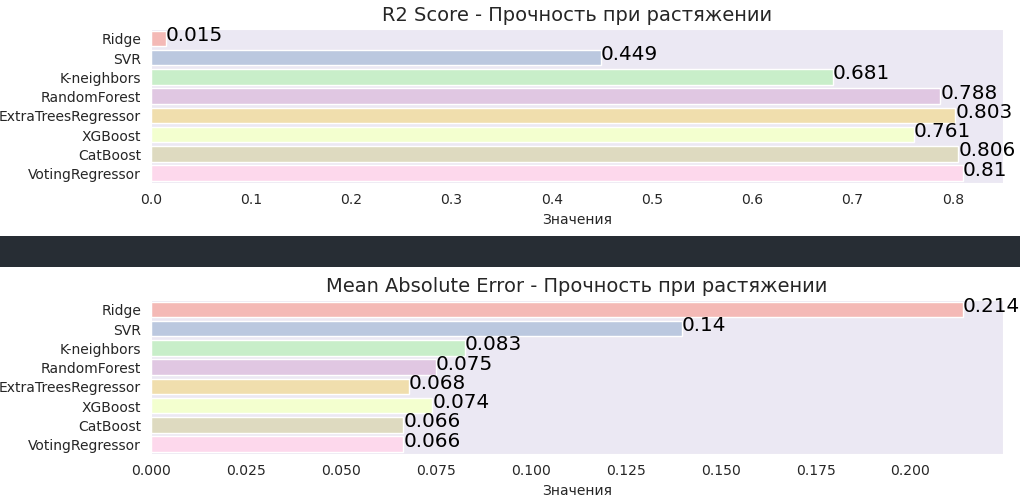
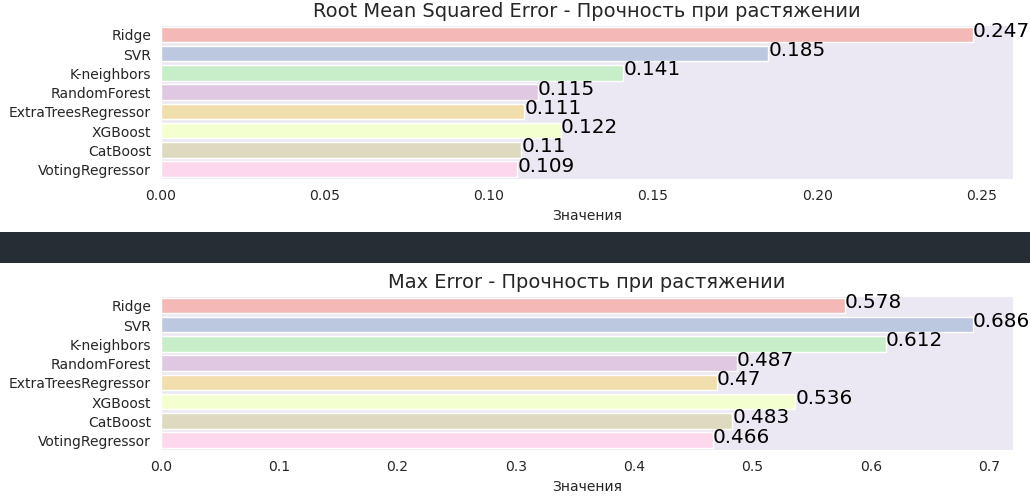
* R2 или коэффициент детерминации измеряет долю дисперсии, объясняемую моделью, в общей дисперсии целевой переменной. Если он близок к единице, то модель хорошо объясняет данные, если же он близок к нулю, то прогнозы сопоставимы по качеству с константным предсказанием;
* RMSE (Root Mean Squared Error) или корень из средней квадратичной ошибки принимает значениях в тех же единицах, что и целевая переменная. Метрика использует возведение в квадрат, поэтому хорошо обнаруживает грубые ошибки, но сильно чувствительна к выбросам;
* MAE (Mean Absolute Error) - средняя абсолютная ошибка так же принимает значениях в тех же единицах, что и целевая переменная;
* MAX(Max Error) - максимальная ошибка данной модели в единицах измерения целевой переменной.

RMSE, MAE и MAX принимают положительные значения. R2 в норме должная принимает положительные значения. Эту метрику надо максимизировать. Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели

Для рекомендации «Соотношение матрица-наполнитель ». был построен регрессор DNN. Архитектура нейронной сети: последовательная входной слой содержит 128 нейронов и активационную функцию «tanh», слой Dropout между входным и скрытым слоем с частотой 80 процентов, второй скрытый слой содержит 64 нейрона и активационную функцию «tahn» и один выходной слой. С помощью метода compile() настроен процесс компиляции это необходимо перед обучением. Количество эпох 1100, batch size 500. В качестве функции ошибки будет использоваться средняя абсолютная ошибка (MAE) и оптимизатор rmsprop.

2.4 Тестирование моделей

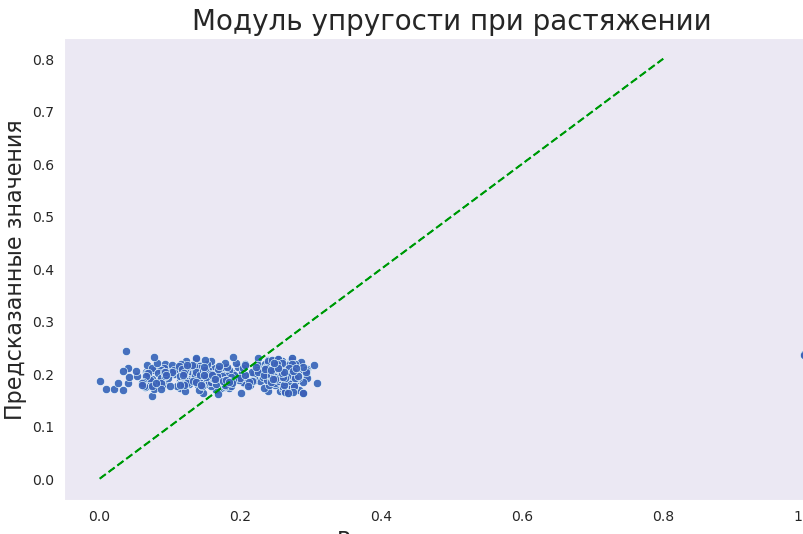
Обучение моделей регрессии подобранных с использованием поиска гиперпараметров , проводились в среде Jupyter Notebook. Оценка точности предсказаний проводилась с использованием оставшихся 30 % тестовых данных.

2. Визуализация соотношений, тестированных данных и предсказанных моделями для «Модуля упругости при растяжении, гпа » и «Прочности при растяжении , мпа» 
3. 
4. Рисунок 16 - Результаты метрик моделей для «Модуля упругости при растяжении, гпа »
5. 
6. 
7. Рисунок 17 - Результаты метрик моделей для «Прочности при растяжении, мпа»

Теперь про результаты, как видно из рисунка 16 и 17 модели Ridge регрессии и SVR плохо спрогнозировали это подтверждает и коэффициент детерминации R2 который составил менее 0.5 , и большой средней абсолютной и средней квадратичной ошибкой. Эти модели не справились со своей задачей .

1. Гораздо лучше результат показала модель KneighborsRegressor ее результат R2 показал более 0.5. Но лучше всех результат показали ансамблевые модели
2. Их R2 составил более 0.7 , а некоторых моделей и более 0.8. Лучше всего результат показал “суперансамбль” VotingRegressor из комбинации CatBoostRegressor и RandomForestRegressor и ExtraTreeRegressor.

Визуализация самой лучшей и худшей моделей для «Модуля упругости при растяжении, гпа » и «Прочности при растяжении , мпа» представлена на

рисунках 17 , 18, 19 и 20.

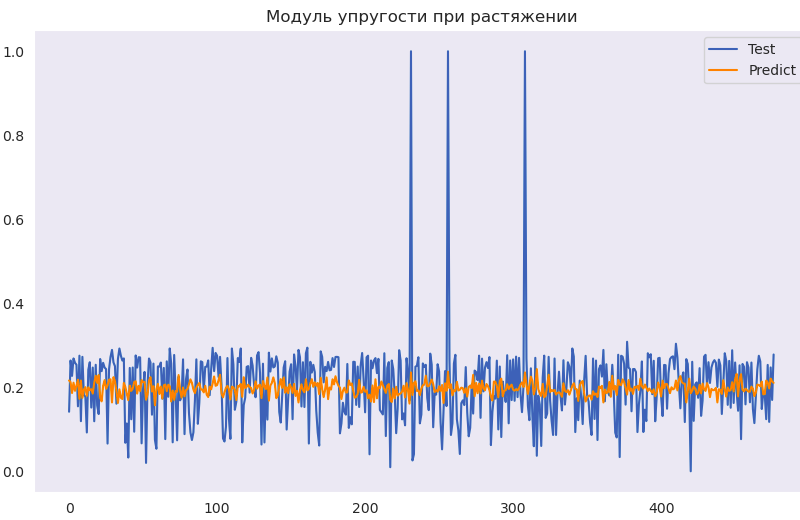
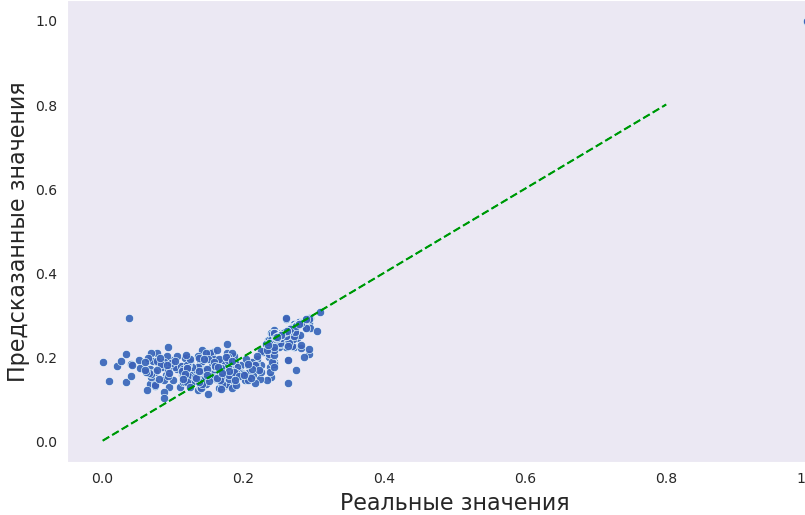
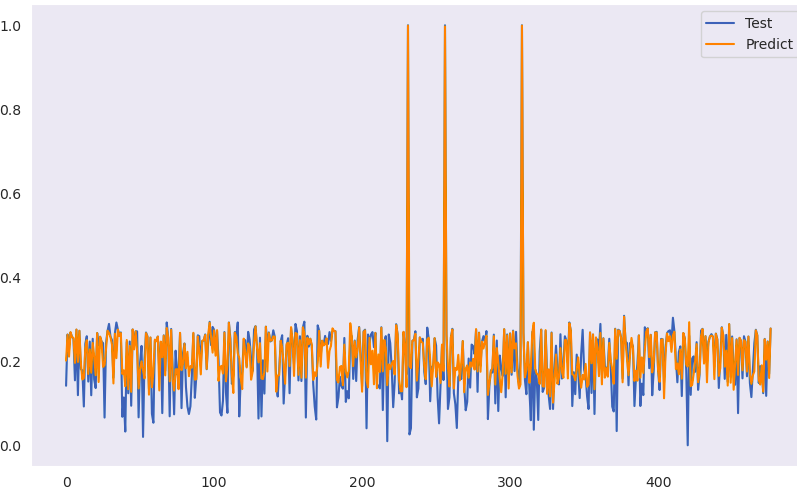
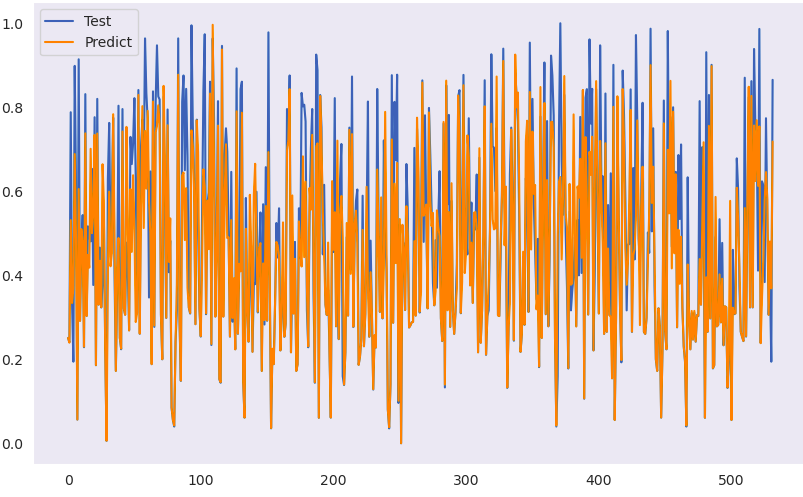
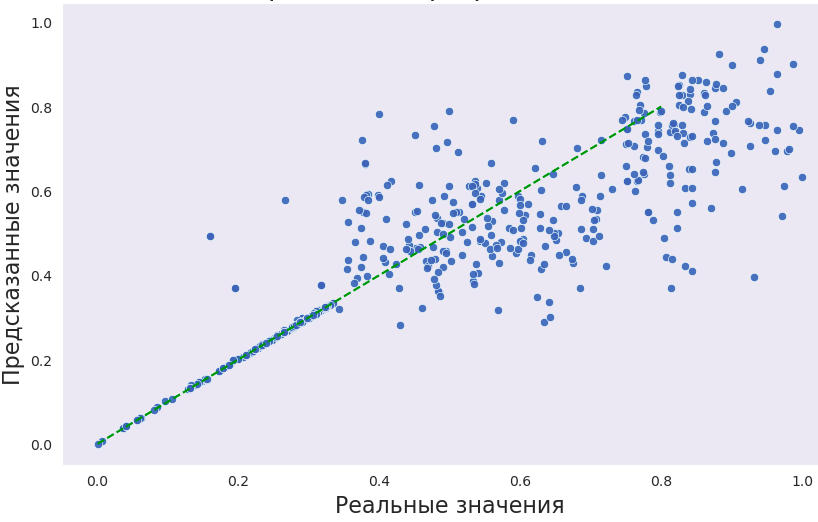
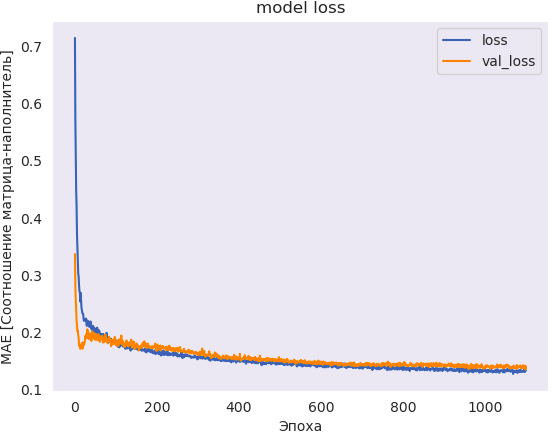
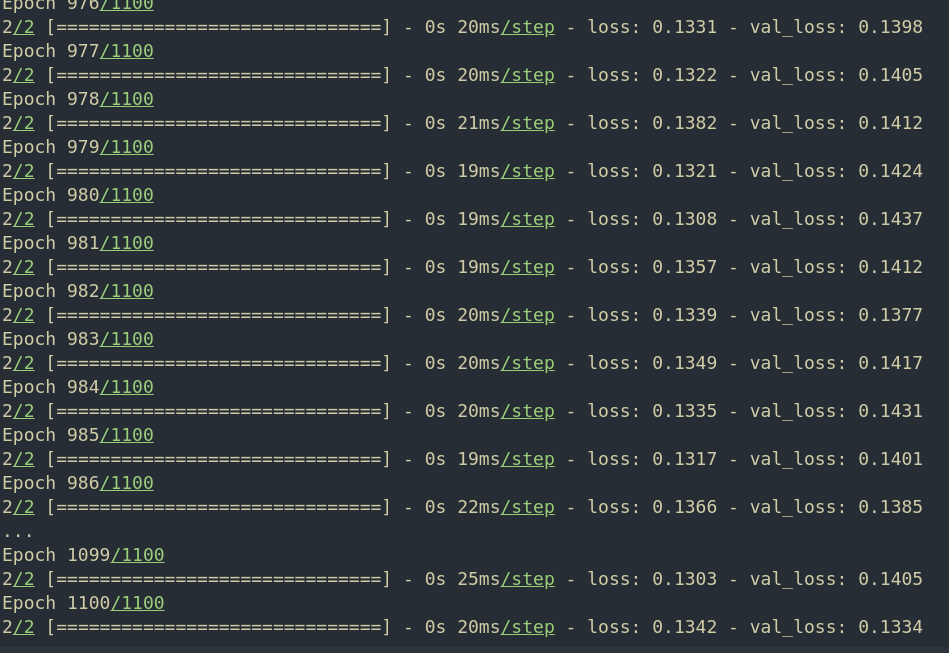


Рисунок 17 - Визуализация для Ridge Regressor «Модуля упругости при растяжении, гпа »

## 

## 

Рисунок 18 - Визуализация для Ridge Regressor «Прочности при растяжении, мпа»

* 1. Рисунок 19 - Визуализация для CatBoostRegressor «Модуля упругости при растяжении, гпа » 
  2. Рисунок 20 - Визуализация для CatBoostRegressor «Прочности при растяжении, мпа»
  3. Визуализация соотношений значений тестовых данных и значений, полученных обучением модели регрессии DNN для «Соотношение матрица наполнитель»

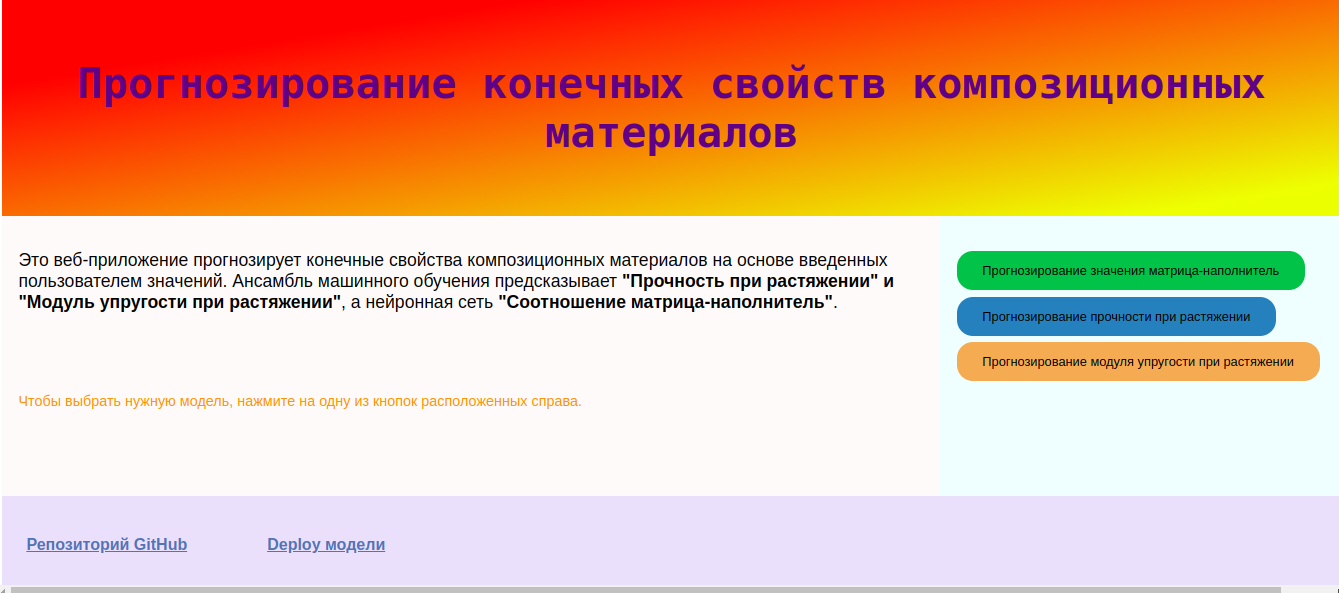
## Рисунок 21 - Визуализация для нейронной сети «Соотношение матрица-наполнитель»

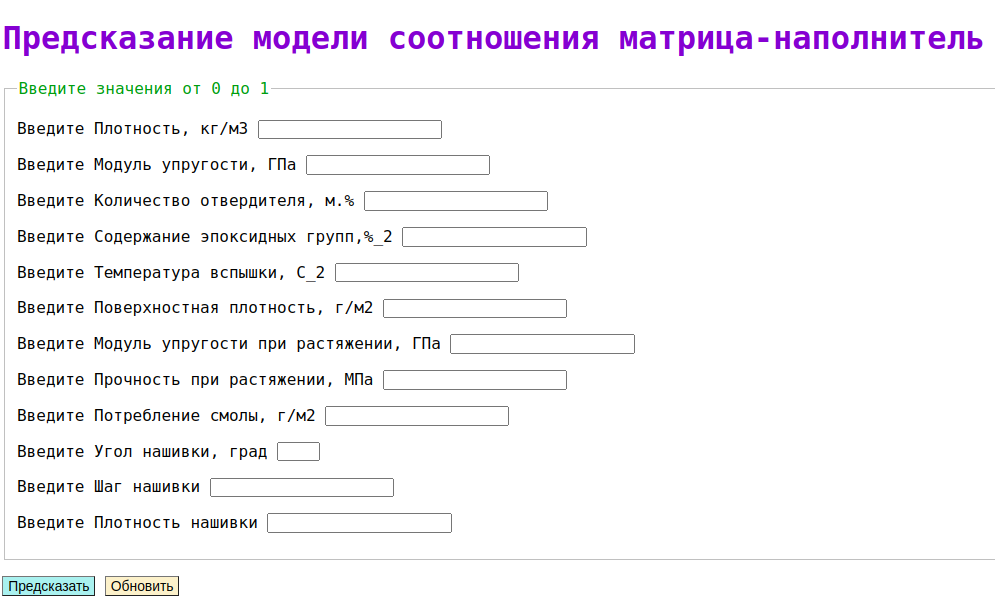
* 1. Нейронная сеть не совсем справилась с поставленной задачей рекомендация соотношение матрица-наполнитель это подтверждает рисунок 21.
  2. Как видно ошибка стремительно уменьшилась, а дальше если и уменьшалась, то совсем незначительно. Реальные значения и прогнозируемые значения сильно разбросаны вдоль линии.

## 3. Разработка приложения

Решено разработать веб-приложение с помощью языка Python, фреймворка Flask и шаблонизатора Jinja.

В приложении необходимо реализовать следующие функции:

* выбор целевой переменной для предсказания;
* ввод входных параметров;
* проверка введенных параметров;
* загрузка сохраненной модели, получение и отображение прогноза выходных параметров.



## Рисунок 22 - Страницы из веб-приложения

* 1. Готовая модель введена в продакшн и выгружена на хостинг: <https://vkr-deploy.onrender.com/>

## 4. Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу <https://github.com/qalansiyah/my_vkr>. На него были загружены результаты работы: исследовательский notebook, код приложения.

# Заключение

В ходе выполнения данной работы был выполнен пайплайн построения моделей машинного обучения.

Этот поток операций и задач включает:

* изучение теоретических методов анализа данных и машинного обучения;
* изучение основ предметной области, в которой решается задача;
* извлечение и транформацию данных. Здесь нам был предоставлен готовый набор данных, поэтому через трудности работы с разными источниками и парсингом данных мы еще не соприкоснулись;
* проведение разведочного анализа данных статистическими методами;
* DataMining — извлечение признаков из датасета и их анализ;
* разделение имеющихся, в нашем случае размеченных, данных на обучающую, валидационную, тестовую выборки;
* выполнение предобработки (препроцессинга) данных для обеспечения корректной работы моделей;
* построение аналитического решения. Это включает выбор алгоритма решения и модели, сравнение различных моделей, подбор гиперпараметров модели;
* визуализация модели и оценка качества аналитического решения;
* сохранение моделей;
* разработка и тестирование приложения для поддержки принятия решений специалистом предметной области, которое использовало бы найденную модель;
* внедрение решения и приложения в эксплуатацию.

Список используемой литературы и веб ресурсы

* 1. 1 Композиционные материалы : учебное пособие для вузов / Д. А. Иванов, А. И. Ситников, С. Д. Шляпин ; под редакцией А. А. Ильина. — Москва : Издательство Юрайт, 2019 — 253 с. — (Высшее образование). — Текст : непосредственный.
  2. 2.ГрасД. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербурr, 2021. - 416 с.: и
  3. 3. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.
  4. 4. Язык программирования Python- Режим доступа: https://www.python.org/.
  5. 5. Библиотека Pandas- Режим доступа: https://pandas.pydata.org/.
  6. 6. Библиотека Matplotlib- Режим доступа: https://matplotlib.org/.
  7. 7. Библиотека Seaborn- Режим доступа: https://seaborn.pydata.org/.
  8. 8. Библиотека Sklearn- Режим доступа: <https://scikit-learn.org/stable/>.
  9. 9. Библиотека Tensorflow - Режим доступа: <https://www.tensorflow.org/>
  10. 10. Документация по библиотеке numpy: Режим доступа:<https://numpy.org>
  11. 11.Библиотека CatBoost - Режим доступа:<https://catboost.ai/>
  12. 12.Библиотека XGBoost - Режим доступа: <https://xgboost.readthedocs.io/en/stable/>
  13. 13.Библиотека ImbalancedLearningRegression -Режим доступа:
  14. <https://imbalancedlearningregression.readthedocs.io/en/latest/>
  15. 14.Библиотека Optuna - Режим доступа: <https://optuna.org/>
  16. 15. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>

# 16.Создание веб-приложения с помощью Flask в Python 3 - Режим доступа:

* 1. <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-ru>
  2. 17.Python Data Science Handbook: - Режим доступа: [**https://jakevdp.github.io/PythonDataScienceHandbook/**](https://jakevdp.github.io/PythonDataScienceHandbook/)