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 Долматова Анастасия Александровна

Москва, 2022

# **Содержание**

[Содержание 2](#_Toc106106360)

[Введение 3](#_Toc106106361)

[1. Аналитическая часть 4](#_Toc106106362)

[1.1. Постановка задачи 4](#_Toc106106363)

[1.2. Описание используемых методов 6](#_Toc106106364)

[1.3. Разведочный анализ данных 11](#_Toc106106365)

[2. Практическая часть 16](#_Toc106106366)

[2.1. Предобработка данных 16](#_Toc106106367)

[2.2. Разработка и обучение модели 16](#_Toc106106368)

[2.3. Тестирование модели 17](#_Toc106106369)

[2.4. Нейронная сеть 18](#_Toc106106370)

[2.5. Разработка приложения 20](#_Toc106106371)

[2.6. Репозиторий и результаты 21](#_Toc106106372)

[Заключение 22](#_Toc106106373)

[Библиографический список 23](#_Toc106106374)

# **Введение**

Пояснительная записка подготовлена в рамках выпускной квалификационной работы по курсу «Data Science» на тему «Прогнозирование конечных свойств новых материалов (композиционных материалов)».

Для решения поставленной задачи изучены теоретические основы предметной области, изучены материалы, представленные заказчиком.

Проведены исследования и анализ предоставленных заказчиком данных с использованием методов, изученных на курсе «Data Science».

## **1. Аналитическая часть**

## **1.1. Постановка задачи**

Композиционные материалы — это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т. е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита - железобетон. Бетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри бетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства. Современные композиты изготавливаются из других материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов. Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

Актуальность: Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Датасет со свойствами композитов.

Число входных и выходных переменных:

Входные переменные:

1. Угол нашивки, град
2. Шаг нашивки
3. Плотность
4. Соотношение матрица-наполнитель
5. Плотность, кг/м3
6. Модуль упругости, Гпа
7. Количество отвердителя, м
8. Содержание эпоксидных групп,%\_2
9. Температура вспышки, С\_2
10. Поверхностная плотность, г/м2
11. Модуль упругости при растяжении, Гпа
12. Прочность при растяжении, Мпа
13. Потребление смолы, г/м2

Выходные переменные:

1. Соотношение матрица-наполнитель
2. Модуль упругости при растяжении, ГПа
3. Прочность при растяжении, МПа

Данные для решения поставленных задач представлены в виде двух датасетов разной длины. Для решения задач датасеты были предварительно объединены и приведены к одной длине в 1023 строки. По условию задачи объединение проводилось по индексу, тип объединения INNER.

При проведении анализа данных было выявлено отсутствие пропусков в датасете и незначительное количество выбросов, процент которых был достаточно мал и удален из выборки методом 3-х сигм.

## **1.2. Описание используемых методов**

Для решения поставленной задачи были использованы методы машинного обучения.

Задача регрессии в машинном обучении — это предсказание одного параметра (Y) по известному параметру X, где X — набор параметров, характеризующий наблюдение.

Для решения задач регрессии были использованы четыре метода машинного обучения:

1. Случайный лес.

Это алгоритм обучения с учителем. Его можно применять как для классификации, так и для регрессии. Также это наиболее гибкий и простой в использовании алгоритм. Лес состоит из деревьев. Говорят, что чем больше деревьев в лесу, тем он крепче. RF создает деревья решений для случайно выбранных семплов данных, получает прогноз от каждого дерева и выбирает наилучшее решение посредством голосования. Он также предоставляет довольно эффективный критерий важности показателей (признаков).

Алгоритм состоит из четырех этапов:

- Создайте случайные выборки из заданного набора данных.

- Для каждой выборки постройте дерево решений и получите результат предсказания, используя данное дерево.

- Проведите голосование за каждый полученный прогноз.

- Выберите предсказание с наибольшим количеством голосов в качестве окончательного результата.

1. К-ближайших соседей.

Метод k-ближайших соседей используется для решения задачи классификации. Он относит объекты к классу, которому принадлежит большинство из k его ближайших соседей в многомерном пространстве признаков. Это один из простейших алгоритмов обучения классификационных моделей.

Число k — это количество соседних объектов в пространстве признаков, которые сравниваются с классифицируемым объектом. Иными словами, если k=10, то каждый объект сравнивается с 10-ю соседями. Метод широко применяется в технологиях Data Mining.

В процессе обучения алгоритм просто запоминает все векторы признаков и соответствующие им метки классов. При работе с реальными данными, то есть наблюдениями, метки класса которых неизвестны, вычисляется расстояние между вектором нового наблюдения и ранее запомненными. Затем выбирается k ближайших к нему векторов, и новый объект относится к классу, которому принадлежит большинство из них.

Выбор параметра k противоречив. С одной стороны, увеличение его значения повышает достоверность классификации, но при этом границы между классами становятся менее четкими. На практике хорошие результаты дают эвристические методы выбора параметра k, например, перекрестная проверка.

Несмотря на свою относительную алгоритмическую простоту, метод показывает хорошие результаты. Главным его недостатком является высокая вычислительная трудоемкость, которая увеличивается квадратично с ростом числа обучающих примеров.

1. Градиентный бустинг.

Gradient Boosting Machine — это мощный алгоритм машинного обучения ансамбля, который использует деревья решений.

Boosting — это общий метод ансамбля, который включает в себя последовательное добавление моделей в ансамбль, где последующие модели корректируют производительность предыдущих моделей.

Алгоритм Gradient Boosting Machines.

Градиентное повышение относится к классу алгоритмов машинного обучения ансамбля, которые могут использоваться для задач классификации или регрессионного прогностического моделирования.

Ансамбли строятся из моделей дерева решений. Деревья добавляются по одному в ансамбль и подходят для исправления ошибок прогнозирования, сделанных предыдущими моделями. Это тип модели машинного обучения ансамбля, называемой бустингом.

Модели подходят с использованием любой произвольной дифференцируемой функции потерь и алгоритма оптимизации градиентного спуска. Это дает методу его название “gradient boosting”, поскольку градиент потерь минимизируется по мере подгонки модели, подобно нейронной сети.

1. Линейная регрессия.

За базовую модель для прогнозирования всех искомых параметров принята sklearn.linear\_model.LinearRegression. Обычная линейная регрессия методом наименьших квадратов. LinearRegression соответствует линейной модели с коэффициентами w = (w1, …, wp), чтобы минимизировать остаточную сумму квадратов между наблюдаемыми целями в наборе данных и целями, предсказанными линейным приближением.

Простая линейная регрессия имеет место, если рассматривается зависимость между одной входной и одной выходной переменными. Для этого определяется уравнение регрессии и строится соответствующая прямая, известная как линия регрессии .

Коэффициенты a и b, называемые также параметрами модели, определяются таким образом, чтобы сумма квадратов отклонений точек, соответствующих реальным наблюдениям данных, от линии регрессии была бы минимальной. Коэффициенты обычно оцениваются методом наименьших квадратов.

Если ищется зависимость между несколькими входными и одной выходной переменными, то имеет место множественная линейная регрессия. Соответствующее уравнение имеет вид:

где n - число входных переменных.

Очевидно, что в данном случае модель будет описываться не прямой, а гиперплоскостью. Коэффициенты уравнения множественной линейной регрессии подбираются так, чтобы минимизировать сумму квадратов отклонения реальных точек данных от этой гиперплоскости.

1. Нейронная сеть архитектуры многослойный персептрон.

Многослойный персептрон (MLP) — это алгоритм обучения с учителем, который изучает функцию  обучением на наборе данных, где m — количество измерений для ввода и o- количество размеров для вывода. Учитывая набор функций X=x1,x2,…xm и цель y, он может изучить аппроксиматор нелинейной функции для классификации или регрессии. Он отличается от логистической регрессии тем, что между входным и выходным слоями может быть один или несколько нелинейных слоев, называемых скрытыми слоями. На рисунке показан MLP с одним скрытым слоем со скалярным выходом.

Задачи решались с использованием следующих методов и библиотек Python:

* 1. NumPy — это общедоступный модуль для python, который предоставляет общие математические и числовые операции в виде пре-скомпилированных, быстрых функций, обеспечивает функционал, который можно сравнить с функционалом MatLab. NumPy (Numeric Python) предоставляет базовые методы для манипуляции с большими массивами и матрицами.
  2. Matplotlib — библиотека на языке программирования [Python](https://ru.wikipedia.org/wiki/Python) для визуализации данных [двумерной (2D) графикой](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0#%D0%94%D0%B2%D1%83%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0_(2D)) ([3D графика](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0#%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) также поддерживается). Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.
  3. [Pandas](http://pandas.pydata.org/) - данный пакет делает Python мощным инструментом для анализа данных. Пакет дает возможность строить сводные таблицы, выполнять группировки, предоставляет удобный доступ к табличным данным.
  4. Scikit-learn (sklearn) — это один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он содержит функции и алгоритмы для машинного обучения: классификации, прогнозирования или разбивки данных на группы.
  5. [Seaborn](https://seaborn.pydata.org/) — это библиотека для создания статистических графиков на Python. Она основывается на [matplotlib](https://matplotlib.org/) и тесно взаимодействует со [структурами данных pandas](https://livecodestream.dev/post/how-to-work-with-pandas-in-python/).
  6. TensorFlow — открытая [программная библиотека](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) для [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), разработанная компанией [Google](https://ru.wikipedia.org/wiki/Google_(%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F)) для решения задач построения и тренировки [нейронной сети](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия.
  7. Math - является самым базовым математическим модулем Python. Охватывает основные математические операции, такие как сумма, экспонента, модуль и так далее. Эта библиотека не используется при работе со сложными математическими операциями, такими как умножение матриц.

## **1.3. Разведочный анализ данных**

1. Описательная статистика.

Описательная статистика – это краткие описательные коэффициенты, которые суммируют данный набор данных, который может быть либо представлением всей совокупности, либо выборкой совокупности. Описательная статистика подразделяется на меры центральной тенденции и меры изменчивости (распространения). Меры центральной тенденции включают среднее, медиану и моду, в то время как меры изменчивости включают стандартное отклонение, дисперсию, минимальные и максимальные переменные, эксцесс и асимметрию.

Для разведочного анализа данных были использованы методы Python и описательной статистики.

- df.info() – выведена общая информация о датасете: тип данных, количество строк и столбцов, наименование переменных.

- sorted(df[column].unique() - Определена уникальность значений в датасете.

- df.isna().sum() – данные проверены на наличие пропусков. Пропусков не обнаружено.

- df.describe() – определены основные статистические характеристики, в том числе средние и медианные значениия характеристик. На рисунке 1 представлены некоторые из них.

Рисунок 1. Описательная статистика

1. Гистограмма плотности распределения.

Гистограмма плотности распределения — это столбиковая диаграмма, которая показывает, как данные распределяются по группам значений. Собранные данные представляют в виде ряда прямоугольников, одинаковых по ширине и различающихся по высоте. Анализ характера изменения высот позволяет оценить динамику процесса.

Гистограмму плотности распределения используют, чтобы наглядно показать, в каком интервале располагаются наиболее часто встречающиеся значения и как вообще распределяются данные. Кроме того, эта гистограмма позволяет определить наилучшие результаты процесса. Такое графическое изображение динамики процесса дает возможность наметить приоритетные задачи по его улучшению.

Построены гистограммы плотности распределения значений каждой переменной. Для большинства характеристик распределение близко к нормальному.

Угол нашивки, град - дискретное распределение, так как колонка содержит всего два уникальных значения. Выборочные результаты представлены на рисунке 2.

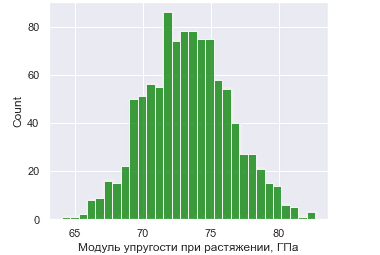


Рисунок 2. Нормальное распределение

1. Ящик с усами.

Ящик с усами, диаграмма размаха — график, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей.

Такой вид диаграммы в удобной форме показывает медиану (или, если нужно, среднее), нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы. Несколько таких ящиков можно нарисовать бок о бок, чтобы визуально сравнивать одно распределение с другим; их можно располагать как горизонтально, так и вертикально. Расстояния между различными частями ящика позволяют определить степень разброса (дисперсии) и асимметрии данных и выявить выбросы.

Для поиска выбросов построены диаграммы Ящика с усами, которые позволяют разбить данные на квантили 25, 50,75, 100. А также позволяют воспользоваться статистическим методом трех сигм.

Отклонение значения нормально распределённой случайной величины X от её математического ожидания М(х) не превосходит утроенного среднеквадратического отклонения σ с вероятностью около 0,9973. Иначе говоря, с вероятностью 0,9973 значение нормально распределённой случайной величины Х находится в интервале

*[М(х) - 3σ ... М(х) + 3σ]*,

Формула 1. Правило 3-сигм.

где *σ* - среднеквадратическое отклонение случайной величины.

На рисунках 3 и 4 представлены выборочные результаты исследования.

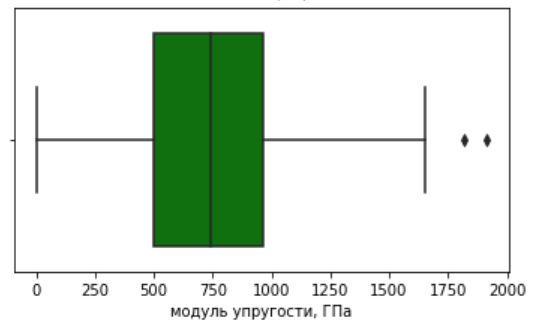


Рисунок 3. Диаграмма Ящик с усами с наличием выбросов в стороне больших значений.

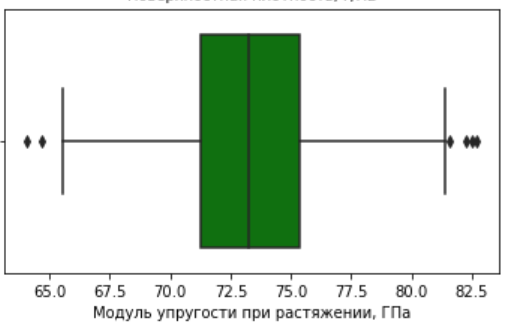


Рисунок 4. Диаграмма Ящик с усами с наличием выбросов с двух сторон.

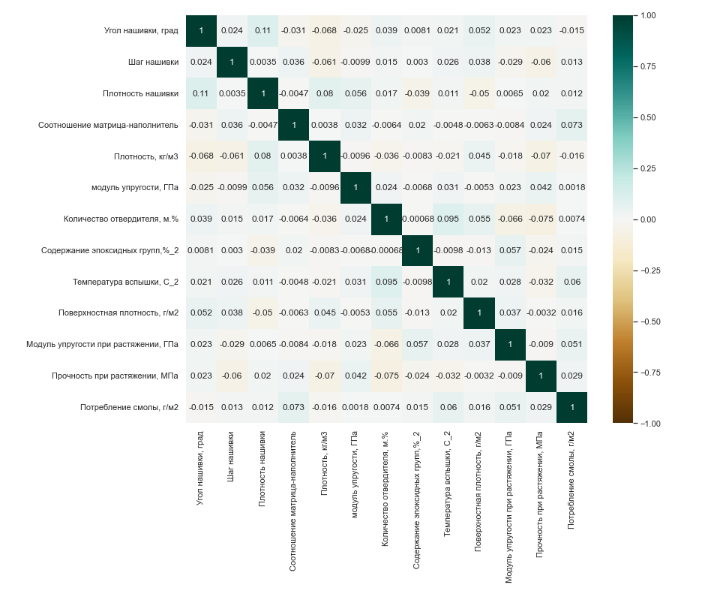
4. Тепловая карта корреляций.

Корреляция - важнейший фактор, лежащий в основе анализа данных. Он сообщает, как переменные в наборе данных связаны друг с другом и как они перемещаются относительно друг друга. Значение корреляции колеблется от -1 до +1. 0 Корреляция означает, что две переменные не зависят друг от друга. Положительная корреляция указывает на то, что переменные движутся в одном направлении, а отрицательная корреляция указывает на противоположное.

Тепловая карта используется для графического представления значений матрицы с разными цветовыми оттенками для разных значений. Он очень четко визуализирует общую матрицу.

Далее для исследования данных был рассчитан коэффициент корреляции для каждой пары входных параметров. Для этого были построены попарные графики точек рассеяния, а также тепловая карта коэффициентов корреляции.

На рисунке 5 представлена тепловая карта коэффициентов корреляции.

Рисунок 5. Тепловая карта коэффициентов корреляции.

Очень слабые прямые и обратные связи переменных в датасете. Это предобработанный датасэт с синтетическими данными. Применять линейные методы регрессии нет смысла.

# **2. Практическая часть**

# **2.1. Предобработка данных**

Предоставленные заказчиком данные для выпускной квалификационной работы были предварительно предобработаны.

На этом этапе было произведена очистка данных от выбросов методом 3-х сигм.

Общее количество выбросов составило 24.

Значения лежат в разных диапазонах. Для сопоставимости данных выполним нормализацию – приведение в диапазон от 0 до 1 с помощью MinMaxScaler.

Результат проведенной предобработки данных был проверен при помощи показателей описательной статистики и метода describe().Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6. Нормализированный датасет.

# **2.2. Разработка и обучение модели**

Для прогноза модуля упругости при растяжении и прочности при растяжении были использованы следующие модели:

- случайный лес

- к – ближайших соседей

- градиентный бустинг

- линейная регрессия.

Для обучения моделей были настроены гиперпараметры модели:

Для обучения моделей были настроены гиперпараметры модели:

Количество деревьев:

'n\_estimators': [60, 80, 100],

Поиск функций для лучшего разделения из набора следующих функций:

'max\_features': ['auto', 'sqrt', 'log2']

Глубина дерева (количество ветвлений или уровней деревьев:

'max\_depth' : [3,4,5,6]

Количество соседей:

'n\_neighbors': np.arange(1, 25)

Для определения лучших параметров выбран метод поиска лучших параметров по сетке из 10 блоков:

Функция GridSearchCV с параметром cv = 10 задано количество блоков.

В качестве метрик оценки качества были выбраны MAE и R2 .

## **2.3. Тестирование модели**

Для каждой модели были определены ошибки:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 7. Тестирование модели.

Модели показывают очень плохие результаты обучения.

## **2.4. Нейронная сеть**

По заданию для соотношения матрица-наполнитель необходимо построить нейросеть. Строю нейронную сеть с помощью класса keras.Sequential со следующими параметрами:

* входной слой нормализации 12 признаков;
* выходной слой для 1 признака;
* скрытых слоев: 2;
* нейронов в первом скрытом слое : 20;
* нейронов во втором скрытом слое : 10;
* добавляем слой Dropout между входным и первым скрытым слоем. Частота выпадения установлена ​​на 20%, то есть каждый пятый вход будет случайным образом исключен из каждого цикла обновления.
* активационная функция скрытых слоев: sigmoid;
* оптимизатор: RMSprop;
* loss-функция: MeanAbsoluteError.

Архитектура нейросети приведена на рисунке 8.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8. Архитектура нейросети

Для оценки качества модели был применена метрика MSE – среднеквадратическая ошибка.

Вышеперечисленные параметры являются гиперпараметрами настраиваемой модели.

Модель обучалась на 20 эпохах так как датасет не большой.

Результаты обучения модели характеризуются изменением среднеквадратической ошибки, представленном на рисунке 9:

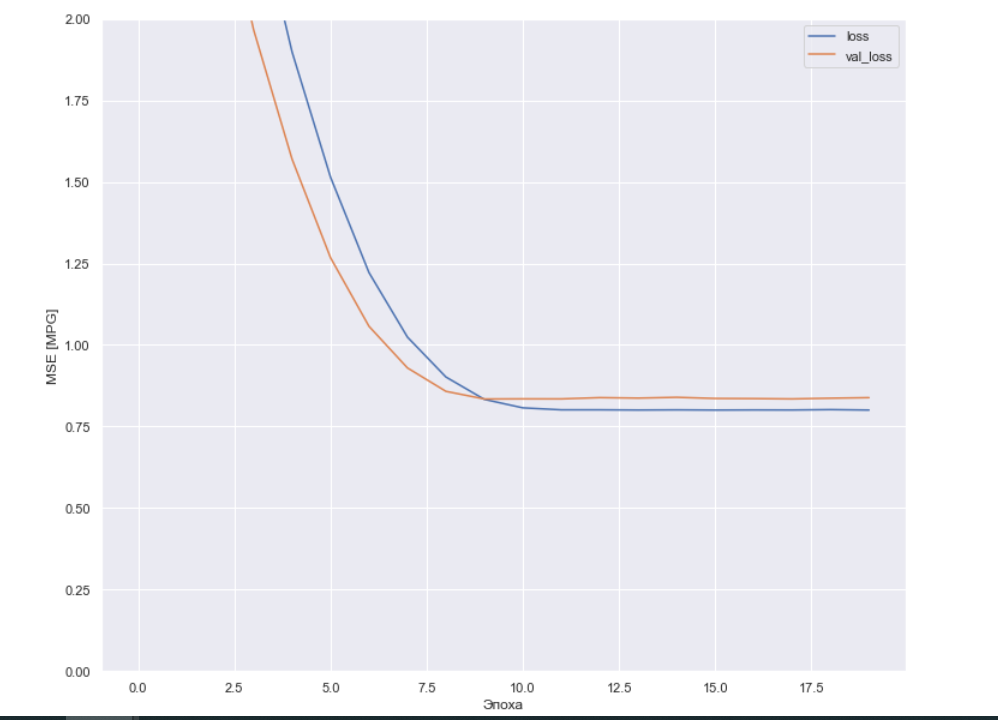


Рисунок 9. Характеристика изменения среднеквадратической ошибки при обучении модели.

## **2.5. Разработка приложения**

Разработано вэб-приложение для модели «Соотношение матрица-наполнитель». Приложение разработано в среде разработки PyCharm. Для разработки приложения был использован интерпретатор Python для запуска веб-приложения Flask. На рисунке 11. Отображено как выглядит приложение.

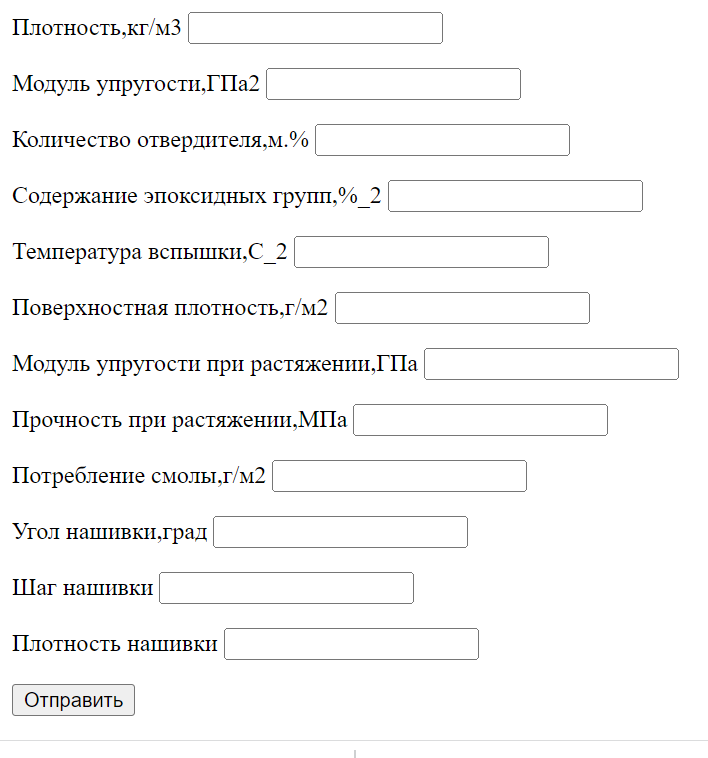


Рисунок 11. Внешний вид приложения

## **2.6. Репозиторий и результаты**

Создан репозиторий на GitHab, где размещен код исследования, оформлен файл README.

Созданный репозиторий:

https://github.com/AnastasiyaDolmatova/VKR\_Dolmatova

## **Заключение**

По результатам исследования были изучены теоретические основы методов машинного обучения, изучены основные библиотеки Python, как одного из основных инструментов для работы аналитика данных.

На основании практической задачи были апробированы методы машинного обучения и построения моделей для решения задач аналитики.

Результаты построения и обучения моделей не дали положительного результата, но позволили приобрести опыт по выбору модели для решения

задач регрессии, опыт по настройке таких моделей, опыт по оценке качества моделей и расчета различных метрик, характеризующих качество построенной модели, позволили оценить временной фактор на проведение подобного рода исследования.

## **Библиографический список**

* 1. Андерсон К. Аналитическая культура. От сбора данных до бизнес-результатов. Пер. с англ. – М: Манн, Иванов и Фербер, 2017 – 336с.
  2. Брюс П., Брюс Э. Практическая статистика для специалистов Data Science. Пер. с англ. – Спб.: БХВ-Петербург, 2018. – 304 с.
  3. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>.
  4. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>.
  5. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>.
  6. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>.
  7. Документация по библиотеке sklearn: – Режим доступа: <https://scikit-learn.org/stable/user_guide.html>.
  8. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>.
  9. Миронов А.А. Машинное обучение часть I ст.9 – Режим доступа: http://is.ifmo.ru/verification/machine-learning-mironov.pdf. (дата обращения 15.04.2022)
  10. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.htm>
  11. Шитиков В.К., Мастицкий С.Е. Классификация, регрессия и другие алгоритмы Data Mining с использованием R: - Режим доступа: <https://ranalytics.github.io/data-mining/index.html>
  12. Шоу Зед. Легкий способ выучить Python. Пер. с англ. –Москва: Эксмо, 2019 – 352 с.