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»

**Тема:** «**Прогнозирование конечных свойств новых материалов**

**(композиционных материалов)»**

Слушатель Дементьева Ольга Андреевна

# 

Москва, 2023

# **Содержание**

[Содержание 2](#_Toc131283806)

[Введение 3](#_Toc131283807)

[1. Аналитическая часть 4](#_Toc131283808)

[1.1. Постановка задачи 4](#_Toc131283809)

[1.2. Описание используемых методов 5](#_Toc131283810)

[1.3. Разведочный анализ данных 11](#_Toc131283811)

[2. Практическая часть 15](#_Toc131283812)

[2.1. Предобработка данных 15](#_Toc131283813)

[2.2. Разработка и обучение модели 17](#_Toc131283814)

[2.3. Тестирование модели 19](#_Toc131283815)

[2.4. Написать нейронную сеть, которая будет рекомендовать 20](#_Toc131283816)

[2.5. Количество слоев и количество нейронов в слое 21](#_Toc131283817)

[2.6. Функции активации 22](#_Toc131283818)

[2.7. Batch Normalization 24](#_Toc131283819)

[2.8. Метод оптимизации 24](#_Toc131283820)

[2.9. Learning rate 26](#_Toc131283821)

[2.10. Dropout 27](#_Toc131283822)

[2.11. Создание удалённого репозитория и загрузка 29](#_Toc131283823)

[3. Заключение 29](#_Toc131283824)

[Список используемой литературы и веб ресурсы 30](#_Toc131283825)

# **Введение**

Композиционные материалы - это материалы, которые состоят из двух или более компонентов, соединенных друг с другом. Обычно одна из компонентов является матрицей, а другая - армирующим волокном. Матрица предназначена для переноса нагрузки и защиты армирующего материала от повреждений, а армирующий материал предназначен для принятия нагрузки.

Примерами композиционных материалов являются стеклопластик, углепластик, арамидный композит и многие другие. Они используются в авиационной, автомобильной, морской и других отраслях промышленности, где требуются легкие, прочные и износостойкие материалы.

Композиционные материалы имеют много преимуществ по сравнению с традиционными материалами, такими как металлы и керамика. Они имеют более высокую прочность и устойчивость к разрыву, легче и могут быть формованы в более сложные формы. Кроме того, они могут быть более устойчивы к воздействию различных химических веществ и коррозии, что делает их более долговечными.

Однако, производство композитных материалов обычно более затратно, чем производство традиционных материалов. Кроме того, из-за их особенностей, требуются специальные методы и оборудование для их обработки и сборки, что может усложнить их производство и ремонт. Композиционные материалы бывают различных типов и классифицируются по различным критериям. Например, по типу матрицы композиты могут быть полимерными (например, стеклопластик), металлическими (например, металлические матрицы с углеродными наполнителями) и керамическими (например, карбид кремния).

По направлению распределения наполнителя композиты могут быть однородными (волокна распределены равномерно), направленными (волокна направлены вдоль одного направления) или неориентированными.

Также композиционные материалы могут быть классифицированы по целевому применению, например, на материалы для авиационной и космической техники, материалы для строительства, материалы для медицинских применений и т.д.

В процессе исследовательской работы будут разработаны несколько моделей для прогнозирования композитных материалов их свойств. Эти модели могут использоваться для оптимизации состава и структуры композитных материалов, чтобы достичь наилучших результатов для конкретных приложений. Например, модели могут предсказывать модуль упругости при растяжении и прочности при растяжении. Это позволяет ускорить процесс разработки новых композитных материалов, снизить затраты и повысить их производительность и конкурентоспособность на рынке.

1. **Аналитическая часть**
   1. **Постановка задачи**

Обычно исследовательские работы в области композитных материалов имеют следующие цели:

1. Разработка новых композитных материалов с определенными свойствами, которые могут использоваться в конкретных приложениях, например, в авиации, автомобильной промышленности, медицине и т.д.;
2. Оптимизация состава и структуры композитных материалов для повышения их производительности и конкурентоспособности на рынке;
3. Исследование физико-химических и механических свойств композитных материалов, чтобы лучше понимать их структуру и свойства.

Для каждой конкретной исследовательской работы могут быть даны свои уникальные задачи и цели в зависимости от требований исследования.

Для исследовательской работы были даны 2 файла: X\_bp.xlsx и X\_nup.xlsx. В них хранилась информация о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). Соединив два файла, мы получили таблицу:



Рисунок 1 – общая информация о композитах

Цель работы: разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель».

Для начала необходимо провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Затем нужно найти выбросы и их обработать, после обучить несколько моделей на предложенных данных.

* 1. **Описание используемых методов**

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем и традиционно это задача регрессии. Цель любого алгоритма обучения с учителем — определить функцию потерь и минимизировать её, поэтому для наилучшего решения в процессе исследования были применены следующие методы:

* LightGBM;
* дерево решений;
* CatBoost

Градиентный бустинг (Gradient Boosting) — это метод машинного обучения, который используется для задач классификации и регрессии. Он является общей концепцией, объединяющей различные алгоритмы построения ансамблей моделей.

Идея градиентного бустинга заключается в последовательном добавлении слабых моделей (например, решающих деревьев) к ансамблю с целью улучшения качества прогнозирования. На каждом шаге алгоритм обучает новую модель, которая будет пытаться исправить ошибки предыдущих моделей. Таким образом, каждая новая модель добавляет свой вклад в общую прогностическую силу ансамбля.

Градиентный бустинг работает следующим образом:

1. Обучается базовая модель, например, решающее дерево, на всей обучающей выборке;
2. Вычисляется ошибка предсказания для каждого объекта в обучающей выборке;
3. Строится новая модель, которая будет пытаться исправить ошибки первой модели, на основе этих ошибок. Например, можно обучить новое дерево на остатках первой модели (т.е. разнице между истинными значениями и предсказаниями первой модели);
4. Новая модель добавляется к ансамблю, и процесс повторяется. Каждая следующая модель обучается на остатках предыдущих моделей, пока не будет достигнута заданная точность или не будет достигнуто максимальное число итераций.

Преимущества градиентного бустинга включают в себя:

1. Высокое качество прогнозирования: градиентный бустинг дает высокую точность прогнозирования в сравнении с другими методами машинного обучения;
2. Работа с разнородными данными: градиентный бустинг может обрабатывать как категориальные, так и числовые признаки;
3. Отсутствие предположений о распределении данных: градиентный бустинг не требует знания о распределении данных и может работать с неравномерными выборками.

Недостатки градиентного бустинга:

1. Градиентный бустинг может быть склонен к переобучению, особенно если используется сложная модель;
2. Время обучения может быть долгим.

Одним из используемых методов градиентного бустинга является LightGBM — это быстрый и эффективный алгоритм над решающими деревьями, который был разработан Microsoft и выпущен как open-source проект.

LightGBM отличается от других алгоритмов градиентного бустинга тем, что он использует уникальный подход к построению деревьев, называемый Gradient-based One-Side Sampling (GOSS) и Exclusive Feature Bundling (EFB). GOSS позволяет уменьшить количество выборок, используемых для обучения, что повышает скорость и эффективность алгоритма. EFB объединяет различные признаки для уменьшения размерности и улучшения точности модели.

LightGBM также обладает несколькими другими функциями, которые делают его популярным в машинном обучении:

1. Работа с категориальными признаками: LightGBM позволяет обрабатывать категориальные признаки напрямую, без необходимости их кодирования;
2. Поддержка распределенного обучения: LightGBM поддерживает распределенное обучение, что позволяет использовать многоядерные процессоры и кластеры для обучения более крупных и сложных моделей;
3. Высокая скорость работы: благодаря своей эффективной реализации LightGBM может быть значительно быстрее других алгоритмов градиентного бустинга.

LightGBM часто используется в задачах регрессии и классификации, но также может использоваться и в других задачах машинного обучения.

Дерево решений (Decision Tree) — это метод машинного обучения, который используется для решения задач классификации и регрессии. Этот метод основан на построении дерева, в котором каждый узел представляет собой некоторый признак, а каждое ребро соответствует значению этого признака.

При обучении дерево решений по данным, он разбивает выборку на все более мелкие подмножества, используя различные признаки, пока не достигнет условия остановки. Критерием разбиения выборки на каждом шаге может служить, например, максимизация информационного выигрыша, критерия Джини или энтропии.

После обучения дерева решений может быть использовано для классификации или регрессии новых объектов, используя последовательность принятия решений, заданную структурой дерева.

Преимущества деревьев решений включают в себя:

1. Легкость интерпретации: деревья решений могут быть легко визуализированы и интерпретированы, что позволяет исследователям и экспертам в данной области понимать, какие признаки и значения играют наиболее важную роль в принятии решений;
2. Эффективность: деревья решений могут быть обучены быстро и могут использоваться для прогнозирования новых объектов с высокой скоростью;
3. Работа с разнородными данными: деревья решений могут обрабатывать как категориальные, так и числовые признаки.

Однако, у деревьев решений есть и некоторые недостатки:

1. Они могут быть склонны к переобучению, особенно если их структура слишком сложна и настраивается под обучающую выборку;
2. Могут возникать проблемы с классификацией объектов, которые не встречались в обучающей выборке;
3. Они могут быть неустойчивыми к шуму и изменениям в данных.

Тем не менее, деревья решений остаются популярным методом машинного обучения и используются в различных областях, таких как финансы, медицина, биология, маркетинг, и т.д.

CatBoost - это градиентный бустинговый алгоритм машинного обучения, который специально разработан для работы с категориальными признаками. Он был разработан компанией Яндекс и представлен в 2017 году. CatBoost использует различные техники, такие как градиентный бустинг, регуляризацию, стохастический градиентный спуск и другие, чтобы улучшить качество модели.

CatBoost автоматически обрабатывает категориальные признаки, без необходимости предварительной обработки данных, такой как кодирование категориальных признаков или заполнение пропущенных значений. Он также имеет встроенные механизмы для обработки пропущенных значений и выбросов.

Одна из основных особенностей CatBoost - это его способность работать с большими наборами данных с высокой размерностью. Он имеет реализацию для параллельного обучения, что позволяет ускорить процесс обучения на больших наборах данных.

CatBoost также предоставляет различные методы для регуляризации модели, такие как L1, L2 и ElasticNet. Он также может использовать симметричную и асимметричную обработку градиента

CatBoost и LightGBM являются современными алгоритмами градиентного бустинга, используемыми в машинном обучении для решения задач классификации и регрессии. Однако у них есть некоторые отличия:

1. Обработка категориальных признаков: LightGBM и CatBoost поддерживают обработку категориальных признаков "из коробки". Однако в LightGBM для этого признаки должны быть заранее закодированы, а в CatBoost - достаточно просто указать их индексы.
2. Обработка пропущенных значений: LightGBM позволяет работать с пропущенными значениями без дополнительных манипуляций, в то время как в CatBoost нужно задать специальный код (например, -999).
3. Работа с большими наборами данных: LightGBM работает быстрее на больших наборах данных, в то время как CatBoost показывает более высокую точность.
4. Поддержка многоклассовой классификации: в отличие от LightGBM, CatBoost может использоваться для решения задач многоклассовой классификации без необходимости применения One-vs-Rest метода.

В качестве метрики выберем RMSE (Root Mean Squared Error, средняя квадратичная ошибка) — это метрика, которая используется для измерения качества модели регрессии. RMSE вычисляется как квадратный корень из среднеквадратической ошибки (MSE), которая представляет собой среднее значение квадратов разностей между прогнозируемыми значениями модели и фактическими значениями.

Формально, RMSE определяется как: RMSE = , где y - фактические значения целевой переменной, - предсказанные значения целевой переменной, n - количество примеров в выборке.

RMSE измеряется в тех же единицах, что и целевая переменная, и чем меньше значение RMSE, тем более точна модель. RMSE является чувствительной к выбросам, поэтому важно убедиться в отсутствии выбросов в данных или принять меры для их обработки.

RMSE имеет ряд свойств, которые делают его полезным для оценки качества модели регрессии:

1. RMSE всегда неотрицательно и может быть равным нулю только в случае, когда предсказанные значения совпадают с фактическими значениями;
2. RMSE является дифференцируемой функцией, что позволяет использовать его в градиентном спуске и других оптимизационных алгоритмах;
3. RMSE является одним из самых часто используемых критериев качества в задачах регрессии, поэтому его легко интерпретировать и сравнивать с другими моделями.

Недостатки MSE:

1. RMSE чувствителен к выбросам. Один или несколько значительных выбросов в выборке могут значительно увеличить значение RMSE и, следовательно, привести к искажению оценки качества модели;
   1. **Разведочный анализ данных**

Разведочный анализ данных (Exploratory Data Analysis, EDA) — это процесс исследования данных для получения понимания о том, как данные устроены и как связаны различные переменные в выборке. Цель EDA состоит в том, чтобы выявить закономерности, тренды, аномалии и паттерны в данных, которые могут помочь в дальнейшем анализе и принятии решений.

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их от различных аномалий, выбросов, пропущенных значений и других ошибок. Это позволяет получить более точные и надежные результаты при обучении моделей и повысить качество прогнозов.

Обработка и очистка данных включает в себя следующие этапы:

1. Удаление дубликатов: исключение повторяющихся записей из данных;
2. Обработка пропущенных значений: заполнение или удаление записей с пропущенными значениями;
3. Удаление выбросов: исключение экстремальных значений, которые могут искажать результаты модели;
4. Приведение данных к нужному формату: преобразование данных в формат, пригодный для использования моделями машинного обучения, например, преобразование категориальных переменных в числовые;
5. Масштабирование данных: приведение всех признаков к одному масштабу, чтобы избежать влияния больших числовых значений на результаты модели;
6. Проверка корректности данных: проверка на соответствие ожидаемым значениям и наличие ошибок в данных.

Обработка и очистка данных являются важным этапом в построении моделей машинного обучения, так как качество входных данных напрямую влияет на точность и надежность прогнозов, которые делает модель.

Дубликатов и пропусков в наших данных не обнаружено. Изучим наши данные, применив метод describe.

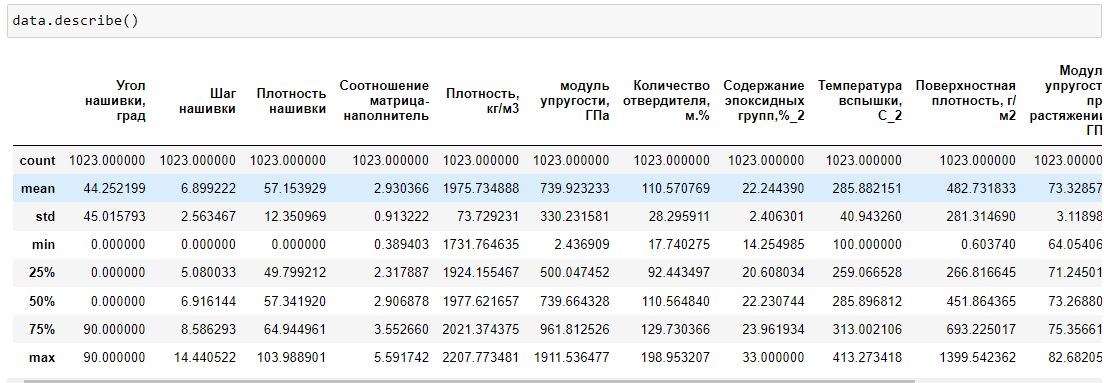


Рисунок 2 - Описательная статистика датасета

Теперь посмотрим более детально на каждый столбец предложенных данных, построив гистограмму и ящик с усами для каждого из них.

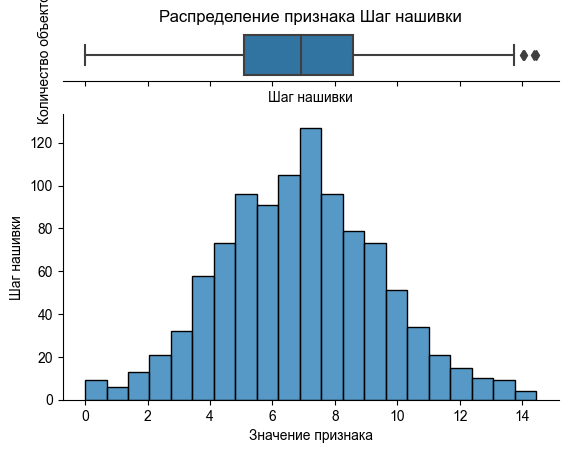


Рисунок 3 – Анализ столбца «Шаг нашивки»

Проанализировав данные, можем выделить, что столбец «Угол нашивки» можно отнести к категориальным переменным.

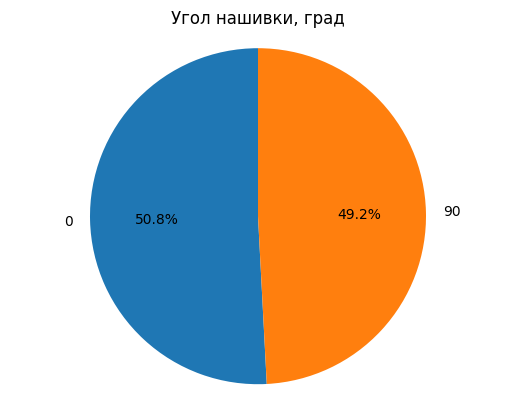


Рисунок 4 – Анализ столбца «Угол нашивки»

Остальные столбцы имеют вещественный тип и преимущественно нормальное распределение, что дает нам право обработать их с помощью метода «трех сигм». Правило трех сигм (или правило 3-х сигм) — это статистическое правило, которое гласит, что в любом нормальном распределении приблизительно 68% значений лежат в пределах одного стандартного отклонения от среднего, 95% значений лежат в пределах двух стандартных отклонений от среднего, а 99,7% значений лежат в пределах трех стандартных отклонений от среднего.

Данное правило широко используется в качестве инструмента для анализа данных и выявления выбросов. Если какое-то значение выходит за пределы трех сигм, то это может быть сигналом наличия аномалии в данных или ошибки в измерениях.

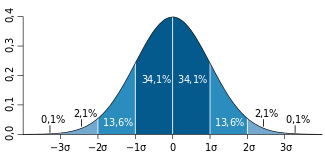


Рисунок 5 – График «трех сигм»

Следовательно, если у нас есть какая-либо точка данных, которая более чем в 3 раза превышает стандартное отклонение, то эти точки, скорее всего, будут аномальными или выбросами.

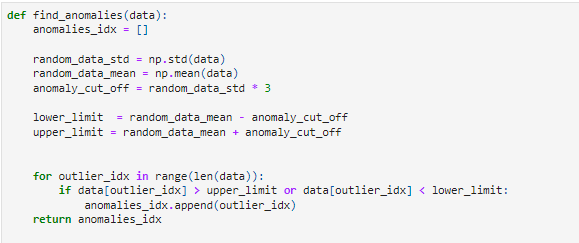


Рисунок 6 – Код для поиска аномальных данных

В результате удаления аномалий мы смогли избавиться от 24 записей.

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

В ходе проведённого анализа принимаем решение столбец "Угол нашивки" привести к категориальному типу и в будущем закодировать его, используя горячее кодирование. OneHotEncoder (горячее кодирование)— это метод кодирования категориальных признаков, который преобразует каждое уникальное значение категориального признака в бинарный вектор. Это делается для того, чтобы категориальный признак мог быть использован в алгоритмах машинного обучения, которые требуют числовых данных.

Концептуально, каждое уникальное значение категориального признака заменяется на вектор, состоящий из нулей и единиц. В этом векторе для каждого возможного значения категориального признака есть отдельный столбец, и в этом столбце на месте соответствующего значения категориального признака стоит 1, а в остальных столбцах - 0.

Например, если у нас есть категориальный признак "цвет", который может принимать значения "красный", "зеленый" и "синий", OneHotEncoder преобразует этот признак в три бинарных признака: "цвет\_красный", "цвет\_зеленый" и "цвет\_синий". Каждый из этих признаков будет иметь значение 1, если соответствующее значение категориального признака равно "красный", "зеленый" или "синий", и 0 в остальных случаях.

OneHotEncoder может быть применен в Pipeline, чтобы автоматически кодировать категориальные признаки при обучении модели. Он также может быть использован для преобразования данных перед обучением модели, чтобы модель могла использовать категориальные признаки в числовом виде.

Pipeline — это последовательность шагов обработки данных и моделирования, которые объединяются в один объект для упрощения работы с данными. Он представляет собой удобный способ комбинирования преобразований данных и моделирования, позволяя выполнять все необходимые операции в едином потоке, а также повторно использовать код для обработки данных и построения моделей.

В Pipeline обычно включают следующие шаги:

1. Предварительная обработка данных, включая масштабирование, преобразование и удаление ненужных признаков;
2. Построение моделей, включая выбор алгоритма и оптимизацию параметров модели;
3. Оценка качества модели с использованием кросс-валидации или отложенной выборки.

Pipeline позволяет объединить эти шаги в единый объект, который может быть использован для обучения и прогнозирования на новых данных. Кроме того, Pipeline облегчает процесс подбора гиперпараметров, так как позволяет определять множество возможных комбинаций преобразований данных и моделей.

Также для обработки данных будем использовать ColumnTransformer. Это классификатор в библиотеке scikit-learn, который позволяет применять различные преобразования к различным столбцам данных в зависимости от их типа. Он часто используется вместе с Pipeline, чтобы применять различные преобразования к разным наборам признаков, прежде чем данные будут переданы модели машинного обучения.

ColumnTransformer может быть использован для применения различных преобразований к разным типам признаков, например, категориальным, числовым или текстовым. Он также может быть использован для выполнения нескольких преобразований для одного и того же типа признаков, например, для нормализации и выброса выбросов.

Когда ColumnTransformer применяется вместе с Pipeline, он может быть использован для построения полной конвейерной системы, в которой данные преобразуются и передаются в модель машинного обучения для обучения и предсказания. Это позволяет автоматизировать процесс обработки данных и упрощает процесс создания и настройки моделей машинного обучения.

* 1. **Разработка и обучение модели**

Разработка и обучение моделей машинного обучения осуществлялась для двух столбцов: «Прочность при растяжении» и «Модуль упругости при растяжении» отдельно. Для того, чтобы предсказывать несколько переменных воспользуемся MultiOutputRegressor. Это классификатор в библиотеке scikit-learn, который позволяет обучать несколько регрессионных моделей одновременно на одном и том же наборе данных для предсказания нескольких выходных значений. Это означает, что он может быть использован для решения задачи многоклассовой регрессии.

MultiOutputRegressor работает путем обучения отдельных регрессионных моделей на каждый из выходных параметров. Каждая модель может быть обучена независимо друг от друга, что позволяет использовать различные модели для каждого выходного параметра, чтобы получить наилучшие результаты. После обучения всех моделей MultiOutputRegressor использует их для предсказания значений выходных параметров для новых наблюдений.

MultiOutputRegressor может быть использован с любым регрессионным алгоритмом в scikit-learn, например, с LinearRegression, RandomForestRegressor, GradientBoostingRegressor, и т.д. Он может быть особенно полезен, когда каждый из выходных параметров зависит от различных наборов признаков, и требуется модель, которая может учитывать эти зависимости и предсказывать значения выходных параметров для новых наблюдений. Для решения применим все методы, описанные выше.



Рисунок 7- Поиск гиперпараметров для LGBMRegressor

Порядок обработки данных таков: сначала мы делим данные на обучающую и тестовую выборки в соотношении 70/30. Далее преобразовываем данные используя ColumnTransformer. С помощью него мы масштабируем численные данные и кодируем категориальные. Далее данные поступают в модель, где и происходит обучение при использовании 10 фолдов. Подбор параметров осуществим с помощью GridSearchCV, который принимает на вход модель, набор гиперпараметров и метод оценки качества модели, а также данные для обучения и тестирования модели. Он перебирает все возможные комбинации гиперпараметров, обучает модель на каждой комбинации и оценивает ее производительность с помощью заданной метрики качества. Затем он выбирает наилучшие значения гиперпараметров на основе оценки производительности модели и возвращает модель с оптимальными значениями гиперпараметров.



Рисунок - наилучшие гиперпараметры LGBMRegressor

Поэкспериментировав с другими моделями, мы получили, что LGBMRegressor показал самый лучший результат на кросс валидации. Именно его мы и будем тестировать.



Рисунок 8 - Поиск гиперпараметров для DecesionTree и CatBoostRegressor

* 1. **Тестирование модели**

После обучения моделей была проведена оценка точности этих моделей на обучающей и тестовых выборках. В качестве параметра оценки модели использовалась средняя квадратическая ошибка (MSE).

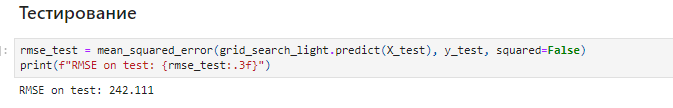


Рисунок 9 - результат оценки точности по RMSE

* 1. **Написать нейронную сеть, которая будет рекомендовать**

**соотношение «матрица – наполнитель».**

Для построения нейронной сети будем использовать библиотеку глубокого обучения Keras. Она предоставляет простой и понятный интерфейс для определения архитектуры нейронной сети, обучения модели и оценки ее производительности.

Данная нейронная сеть должна рекомендовать соотношение «матрица – наполнитель». Рассмотрим данный столбец более подробно:

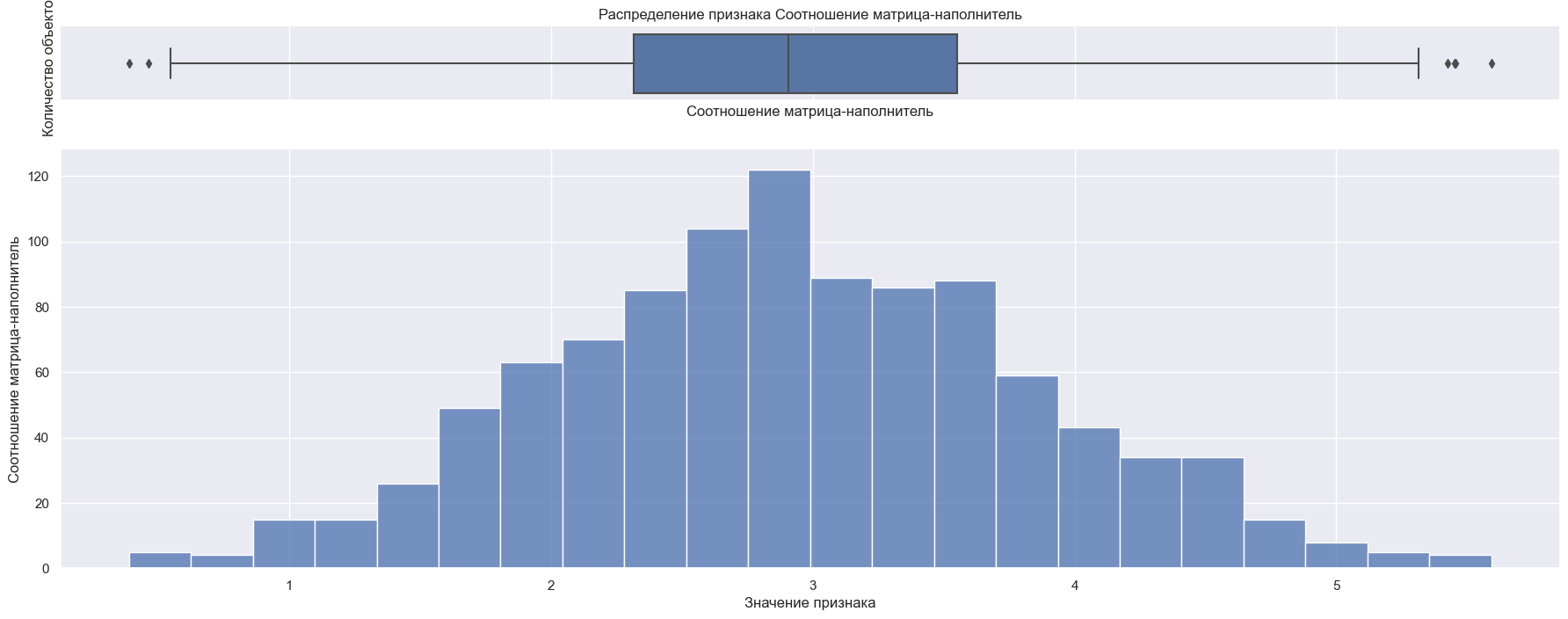


Рисунок 9 – Анализ столбца соотношение «матрица – наполнитель».

Для начала создадим Baseline, который постараемся улучшить, подбирая разные гиперпараметры.

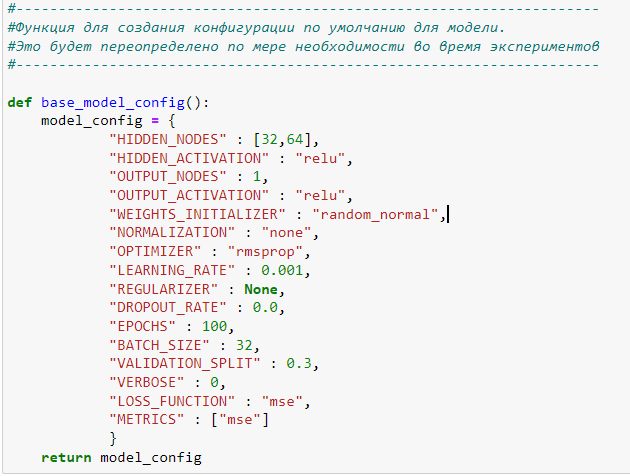


Рисунок 10 – параметры Baseline модели

В качестве удобства будем использовать EarlyStopping это техника регуляризации в машинном обучении, когда метрика перестает улучшаться в течение заданного количества эпох, EarlyStopping останавливает обучение модель, тем самым помогая сэкономить время на обучении и избежать переобучения модели.

Подбираемые параметры:

* Кол-во слоев и кол-во нейронов
* Функции активации
* Batch Normalization
* Метод оптимизации
* Темп обучения (learning rate)
* Dropout

# **Количество слоев и количество нейронов в слое**

Переберем от 3 до 4 слоев и от 5 до 30 нейронов с шагов в 5. По итогу получилось, что при большом количестве параметров модель переобучается очень быстро. По графику видно, что самым лучшим сочетанием оказалась комбинация 4 слоя и 5 нейрона, что дало 161 обучаемых параметров.

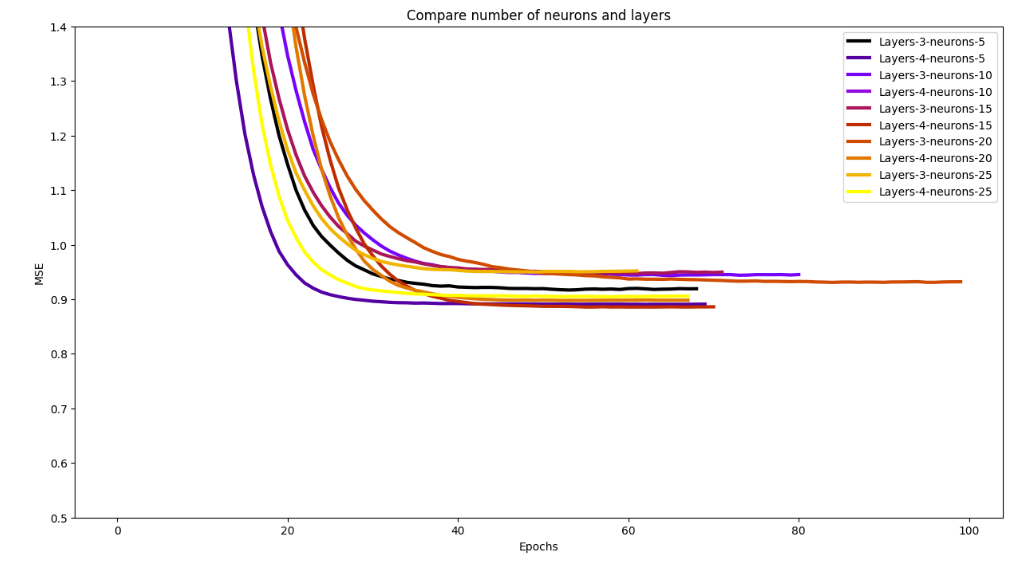


Рисунок 11 – Сравнение количества слоев и количества нейронов в слое

# **Функции активации**

Функция активации — это нелинейная функция, которая применяется к выходу нейрона в нейронной сети, чтобы добавить нелинейность и способствовать более сложному выражению зависимостей между входом и выходом.

В данном пункте рассмотрим, какая же функция активации покажет наилучший результат. Будем рассматривать ReLU, LeakyReLU, ELU:

1. ReLU (Rectified Linear Unit) — это простая функция, которая возвращает максимум из 0 и входного значения. Она позволяет ускорить обучение и справляется с проблемой исчезающего градиента;
2. LeakyReLU — это вариант функции ReLU, который позволяет сохранять градиенты для отрицательных значений входа, чтобы избежать проблемы «мертвых нейронов»;
3. ELU (Exponential Linear Unit) — это функция активации, которая была предложена в 2015 году и быстро стала популярной в области глубокого обучения. Она была разработана для решения некоторых проблем, связанных с другими функциями активации, такими как ReLU. ELU очень похож на ReLU для положительных значений входа, но для отрицательных значений ELU экспоненциально приближается к -1. Это позволяет избежать проблемы "мертвых нейронов", связанной с ReLU, когда нейрон перестает активироваться при отрицательных значениях входа. Кроме того, ELU имеет другие преимущества. Она имеет непрерывную производную, что делает ее более устойчивой к градиентному исчезновению. Она также имеет более гладкую кривую активации, что улучшает скорость сходимости и уменьшает вероятность переобучения.

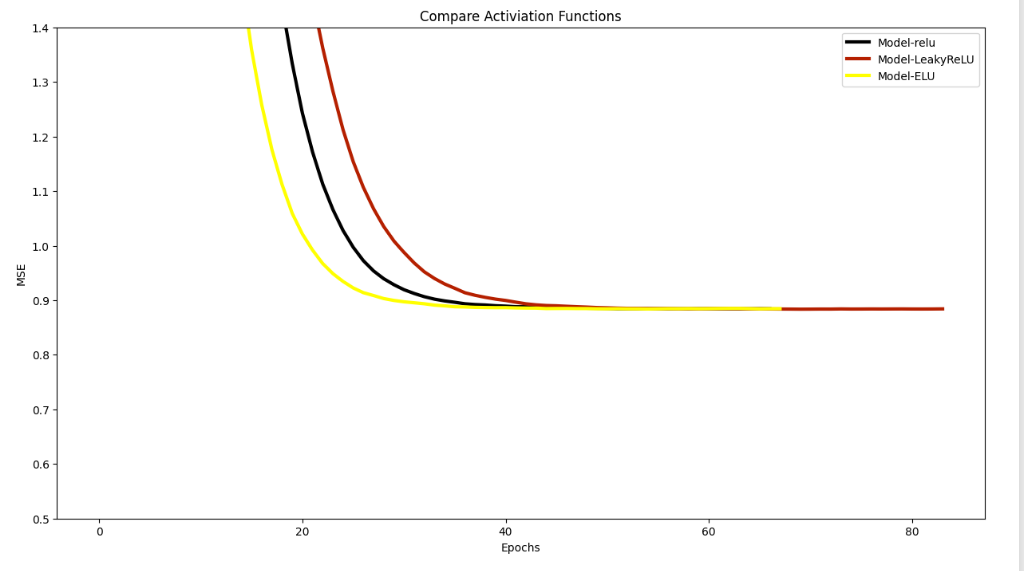


Рисунок 12 – Сравнение функций активации

Как мы можем увидеть на графике, самой лучшей функцией активацией стала ELU.

# **Batch Normalization**

Batch Normalization (BN) — это техника, которая позволяет ускорить и улучшить обучение глубоких нейронных сетей. Она заключается в нормализации входных данных каждого слоя в нейронной сети, чтобы среднее значение стало близким к 0, а стандартное отклонение близким к 1. Это делается для каждого батча данных, который подается на вход сети.

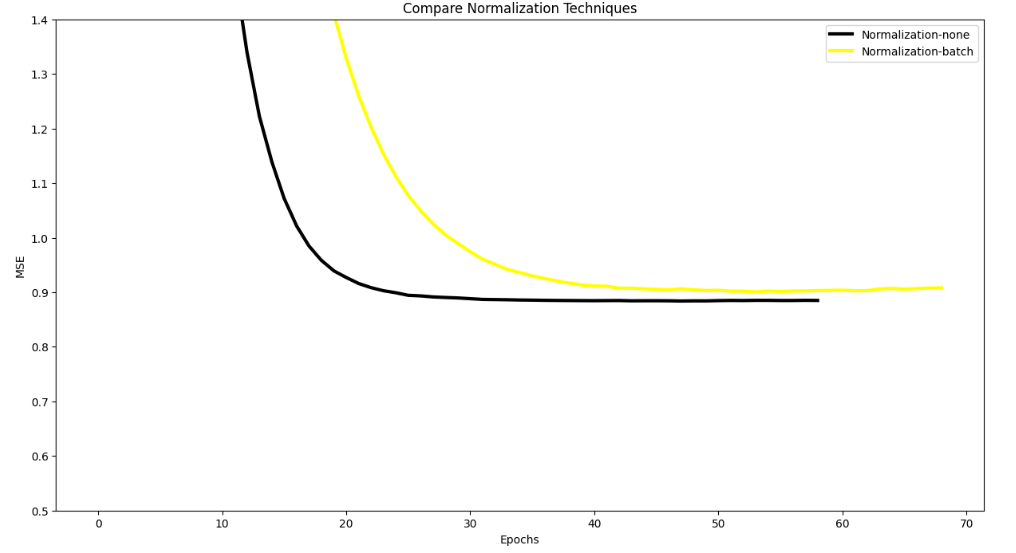


Рисунок 13 – Сравнение Batch Normalization

В ходе обучения можно заметить, что Batch Normalization вовсе не нужна, она только делает модель хуже.

# **Метод оптимизации**

Методы оптимизации нейронной сети нужны для настройки весов модели с целью минимизации функции потерь. Функция потерь измеряет, насколько хорошо модель предсказывает значения целевой переменной, и цель оптимизации заключается в том, чтобы найти такие значения весов, при которых функция потерь будет минимальна.

Мы будем рассматривать SGD и Adam.

SGD (Stochastic Gradient Descent) — это простой и популярный метод оптимизации для обучения нейронных сетей. Он использует градиент функции потерь для обновления весов на каждой итерации обучения. Однако в отличие от обычного градиентного спуска, который вычисляет градиент по всей обучающей выборке на каждой итерации, SGD использует только случайно выбранный мини-набор данных для вычисления градиента на каждой итерации.

Это позволяет ускорить обучение и избежать застревания в локальных минимумах функции потерь. Кроме того, SGD хорошо масштабируется для больших обучающих наборов данных, так как он использует только часть данных для каждого обновления весов.

Adam (Adaptive Moment Estimation) — это метод оптимизации, который был разработан для обучения нейронных сетей. Он является комбинацией методов градиентного спуска и адаптивных методов оптимизации, таких как RMSprop.

Adam использует градиент функции потерь, как и SGD, но в отличие от SGD, Adam поддерживает адаптивную скорость обучения и имеет меньшую склонность к застреванию в локальных минимумах. Он также автоматически вычисляет и подстраивает скорость обучения на каждой итерации, что делает его более эффективным для обучения нейронных сетей.

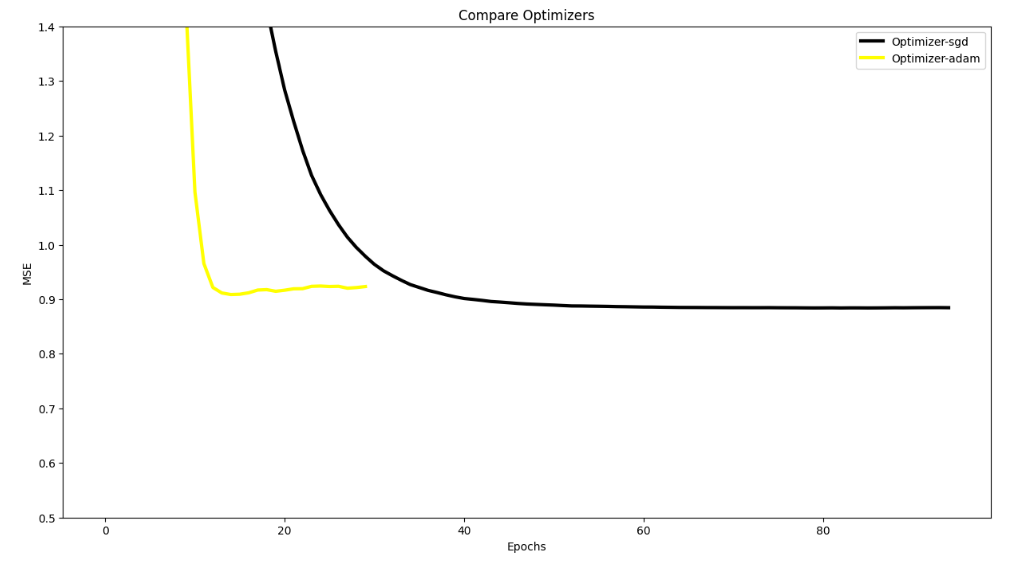


Рисунок 14 – Сравнение методов оптимизации

SGD оказался более стабильным, ошибка убывает плавно, без скачков

# **Learning rate**

Learning rate (скорость обучения) - это параметр, который определяет, насколько быстро нейронная сеть будет обучаться. Он определяет размер шага, который алгоритм оптимизации делает при обновлении весов нейронной сети на каждой итерации.

Слишком высокая скорость обучения может привести к тому, что алгоритм оптимизации будет "перепрыгивать" оптимальные значения весов, из-за чего обучение будет неустойчивым и неэффективным. С другой стороны, слишком низкая скорость обучения может привести к тому, что обучение будет очень медленным и может потребоваться много времени, чтобы достичь оптимальных значений весов.

Оптимальное значение скорости обучения зависит от конкретной задачи, архитектуры нейронной сети и размера набора данных. Обычно начальное значение скорости обучения выбирают эмпирически и затем настраивают его в процессе обучения с помощью методов, таких как ранняя остановка (Early Stopping) и контроль скорости обучения (Learning Rate Decay).

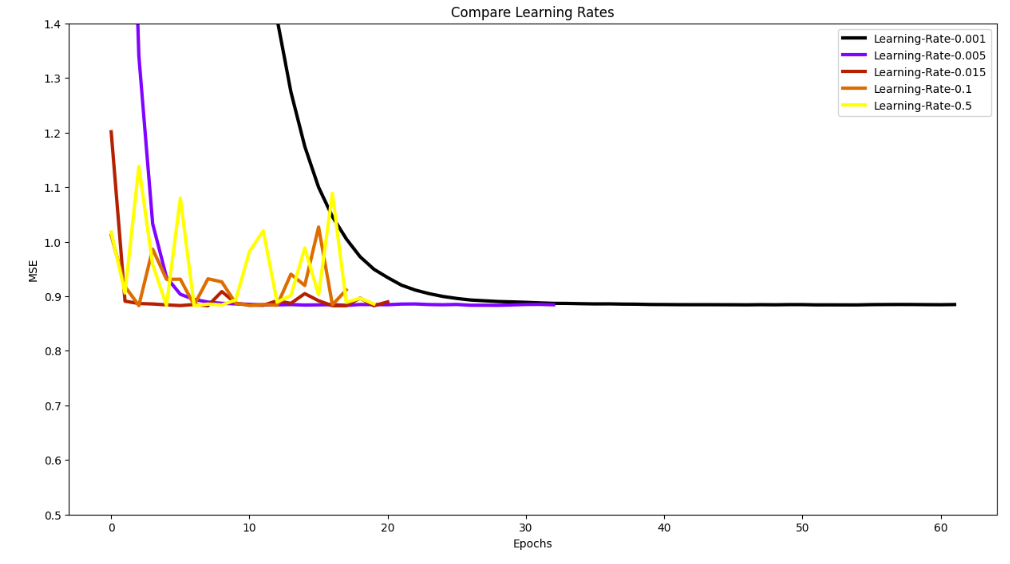


Рисунок 15 – Сравнение learning rate

Самый лучший learning rate оказался равным 0.015, именно при нем модель обучается лучше всего.

# **Dropout**

Dropout - это техника регуляризации, которая применяется в нейронных сетях, чтобы предотвратить переобучение. Она заключается в случайном "выключении" (отключении) некоторых нейронов во время обучения.

В процессе обучения каждый нейрон имеет определенный вес, который соответствует его важности в решении задачи. Когда мы случайно отключаем некоторые нейроны, мы уменьшаем их влияние на выход модели, что приводит к уменьшению переобучения и повышению обобщающей способности модели.

Применение Dropout позволяет уменьшить переобучение и повысить обобщающую способность модели. Однако, при этом может наблюдаться увеличение ошибки на обучающей выборке, что является нормальным эффектом, и не говорит о низком качестве модели.

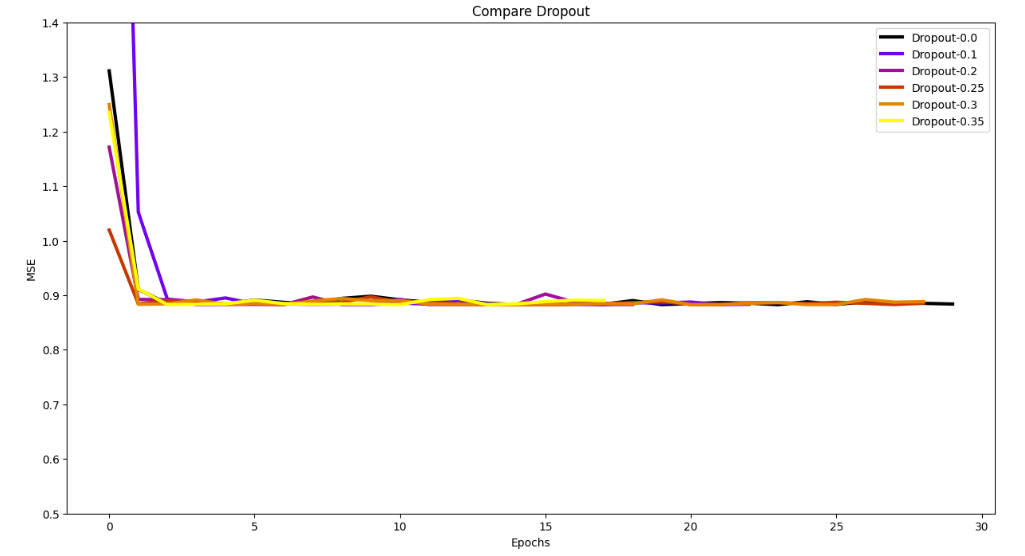


Рисунок 16 – Сравнение Dropout

При Dropout = 0.25 модель быстрее выучивает зависимости, поэтому будем использовать именно это значение.

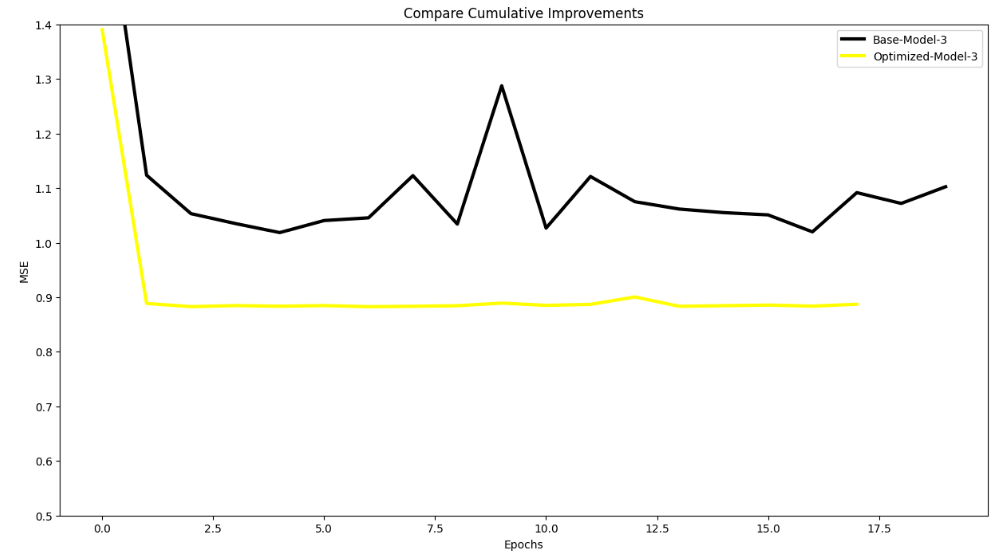


Рисунок 17 – Сравнение Baseline и оптимизированной модели

Как мы видим, оптимизированная модель показывает гораздо более лучший и стабильный результат, чем Baseline модель

**Разработка приложения**

Приложение работает из командной строки. Достаточно запустить его с помощью python и передать путь. Далее необходимо заполнить предлагаемые данные и на выходе получаем соотношение «матрица – наполнитель».

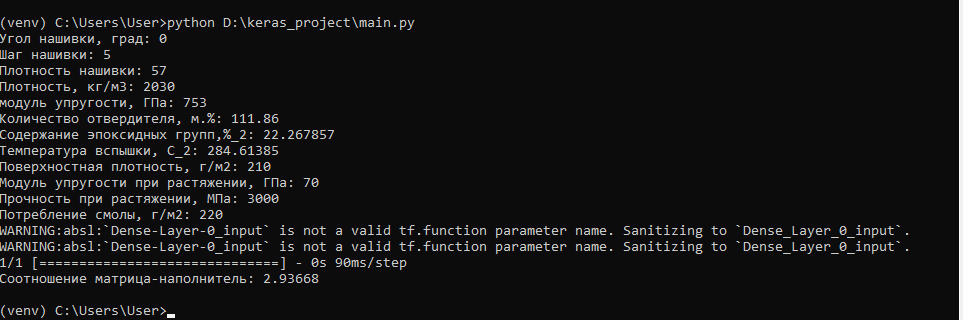


Рисунок 18 – Приложение с интерфейсом командной строки

* 1. **Создание удалённого репозитория и загрузка**

Репозиторий был создан на github.com по адресу: https://github.com/OlgaAndreevnaD/composite-materials

# **Заключение**

Прогнозирование конечных свойств новых материалов является важной задачей в материаловедении, которая может быть решена с помощью различных методов машинного обучения. Результаты исследования показали, что применение моделей регрессии для прогнозирования свойств композитов не всегда является эффективным подходом. Вместо этого, использование более сложных методов, таких как градиентный бустинг (LightGBM), дерево решений, стохастический градиентный спуск могут дать более точные прогнозы для определенных свойств материалов. Для получения более достоверных прогнозов, необходимо учитывать не только одиночные признаки, но и их комбинации, а также проводить анализ корреляций между ними. В целом, прогнозирование конечных свойств новых материалов является сложной задачей, требующей использования различных методов и тщательного анализа данных.

В ходе работы выяснилось, что LightGBM - лучшая модель для предсказания столбцов: «Модуль упругости при растяжении, Гпа», «Прочность при растяжении, Мпа».

Для прогнозирования столбца соотношение «матрица – наполнитель» были подобраны параметры, при которых модель показала наилучший результат. Данные параметры указаны в таблице ниже:

*Таблица 1*

|  |  |
| --- | --- |
| Количество слоев | 4 |
| Количество нейронов | 5 |
| Dropout | 0.25 |
| Batch Normalization | none |
| Learning rate | 0.015 |
| Функция активации | ELU |
| Метод оптимизации | Stochastic Gradient Descent |

**Список используемой литературы и веб ресурсы**

1. Alpaydin E. Introduction to machine learning. – MIT press, 2020.
2. Cherkassky V., Mulier F. M. Learning from data: concepts, theory, and methods. – John Wiley & Sons, 2007.
3. Quinlan J. R. Induction of decision trees //Machine learning. – 1986. – Т. 1. – С. 81-106.
4. Vapnik V. The nature of statistical learning theory. – Springer science & business media, 1999.