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 Моисеев Герман Александрович

Москва 2022

Содержание

[Введение 3](#_Toc106265990)

[1. Аналитическая часть. 7](#_Toc106265991)

[1.1. Постановка задачи. 7](#_Toc106265992)

[1.2. Описание используемых методов. 8](#_Toc106265993)

[1.3. Разведочный анализ данных. 17](#_Toc106265994)

[2. Практическая часть 27](#_Toc106265995)

[2.1. Предобработка данных. 27](#_Toc106265996)

[2.2. Разработка и обучение модели. 30](#_Toc106265997)

[2.3. Тестирование модели. 32](#_Toc106265998)

[2.4. Построение нейронной сети. 32](#_Toc106265999)

[2.5. Разработка приложения 34](#_Toc106266000)

[2.6. Репозиторий 35](#_Toc106266001)

[Заключение 36](#_Toc106266002)

[Список литературы 37](#_Toc106266003)

# **Введение**

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

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

Композиционные материалы или композиты – это материалы, состоящие из двух и более компонентов (отдельных волокон или других армирующих составляющих и связующей их матрицы) и обладающие специфическими свойствами, отличными от суммарных свойств составляющих их компонентов.

Компоненты композитов не должны растворяться или иным способом поглощать друг друга. Они должны быть хорошо совместимы. Свойства композиционных материалов нельзя определить только по свойствам компонентов, без учета их взаимодействия.

Примеры композиционных материалов: пластик, армированный борными, углеродными, стеклянными волокнами, жгутами или тканями на их основе; алюминий, армированный нитями, стали, бериллия.

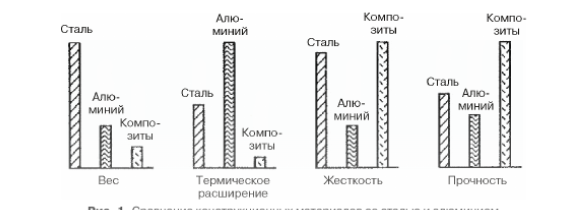


Рисунок 1- Сравнение конструкционных материалов   
со сталью и алюминием[[1]](#footnote-1)

Комбинируя объемное содержание компонентов, можно получать композиционные материалы с требуемыми значениями прочности, жаропрочности, модуля упругости, абразивной стойкости, а также создавать композиции с необходимыми магнитными, диэлектрическими, радиопоглощающими и другими специальными свойствами.

Традиционно композиты имеют такие структурные элементы, как матрица и набивка. В разных источниках их называют по-разному, но смысл от этого   
не меняется.[[2]](#footnote-2)

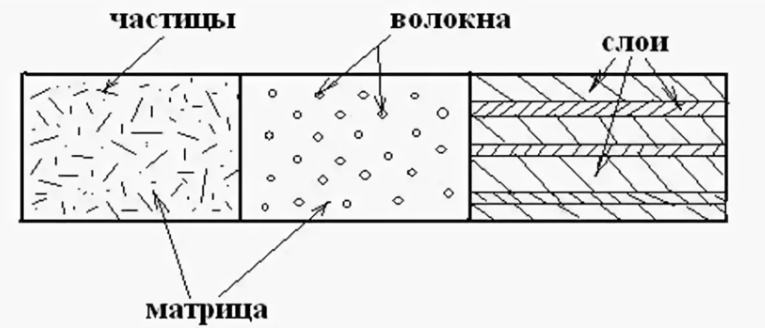


Рисунок 2 – Структурные элементы композитов

Матрицей принято называть материал, внутри которого располагается набивка. В случае железобетона матрицей является бетон. Характерной особенностью является то, что дополняющие материал или материалы как будто плавают в материале матрицы. Набивка или наполнитель — это компоненты, которые призваны усовершенствовать свойства. В случае железобетона —   
это арматура из металла, пронизывающая весь материал насквозь и придающая большую пластичность. Не обязательно, что это один вид материала   
и не обязательно, что используется он в единственном числе. Наряду   
с волокнами, в материал могут быть добавлены некоторые дисперсные частицы.

Вторая мировая война позволила армированным полимерам перекочевать из лабораторий в реальный мир.

Альтернативные материалы, позволяющие снизить вес конечного изделия, были необходимы в военном авиастроении. Очень быстро инженеры поняли преимущества композитов в плане их веса и прочности. [[3]](#footnote-3)

В свое время прогресс в космической сфере стал стимулом для развития композитной отрасли. В нашей стране лидером данного направления было и есть Обнинское научно-производственное предприятие «Технология». Первые обнинские композитные материалы еще в 1980-х годах были испытаны   
на стойкость на планете Венера – вошли в состав одноименных советских космических аппаратов. Чуть позже «Технология» присоединилась к самому масштабному космическому проекту страны – разработке корабля многоразового использования «Буран». Космоплан должен был выдержать температуру 1500 °С во время входа в атмосферу. Специалистами «Технологии» была создана специальная керамическая плитка для обшивки. Для «Бурана», который совершил полет в 1988 году, было выпущено почти 40 тыс. таких плиток.

В Обнинске были разработаны и крупногабаритные композиционные конструкции, например, головной обтекатель ракеты-носителя «Протон». Благодаря использованию композитов «Протон» «скинул» почти полторы тонны, что имело огромный экономический эффект. Ведь вывести на орбиту тонну груза стоит почти столько же сколько тонна золота.

Сегодня производятся множество композиционных конструкций для космоса и авиации. Это обтекатели для ракет-носителей, панели для самых современных спутников, детали для военных и гражданских самолетов. К примеру, на предприятии создаются композитные конструкции для новейшего авиалайнера МС-21 и двигателей самолета Sukhoi Superjet 100.

Композиты сейчас применяются не только в космической индустрии   
и авиастроении, их можно встретить везде. Например, еще в 1980-х годах   
они «перекочевали» в автопром. Наглядной рекламой композитов   
для автомобилестроения послужила тогда победа на международных автогонках первого углепластикового болида. С тех пор с каждым годом объем композитов в современных машинах растет. Их использование позволяет снизить массу автомобиля примерно на четверть, а значит повысить эффективность двигателя и уменьшить расход горючего.

По мере развития науки композитные материалы становятся разнообразнее и дешевле. Всего столетие назад европейские монархи заказывали алюминиевые короны и сервировали столы алюминиевой посудой – тогда алюминий стоил дороже золота. Позже алюминий научились изготавливать в промышленных масштабах, и сегодня любой может купить газировку в алюминиевой банке   
и использовать фольгу. Композиты проходят тот же путь, и их развитие способно предоставить промышленности совершенно новые технологические возможности.[[4]](#footnote-4)

Разработка и создание деталей из композиционных материалов, а также разработка и исследование характеристик нового материала являются комплексом сложных и связанных задач. Подход к решению таких задач заключается в выборе оптимального сочетания технологий и материалов; проведении компьютерного моделирования напряженно-деформированного состояния конструкции и технологических процессов производства; организации экспериментальных исследований для определения исходных характеристик материалов. Одна из ведущих российских авиастроительных компаний недавно столкнулась с трудностями при попытке перед изготовлением точно смоделировать рабочие характеристики деталей композитной конструкции. Их трудности были связаны с более сложной природой нового материала, из-за чего в результате у них случился неожиданный отказ.

Способность точно предсказывать поведение композитных материалов   
во время производства, а также прогнозировать свойства уже готового изделия имеет решающее значение для обеспечения уверенности в конструкции, поддержки изначально правильного подхода, увеличения доли использования композитов и ускорения их внедрения.

Задача точного моделирования композитного материала заключается   
в получении точного описания его состава. В отличие от обычного материала (например, стали), композит, как правило, представляет собой смесь волокнистых материалов различной толщины, которые могут быть наслоены друг на друга для создания единого материала. Ориентация волокон в каждом слое (направление основного волокна) отличается с точки зрения угла расположения, толщины и материала, расположенного слоем выше и ниже.   
Эта сложность компенсируется гибкостью, позволяющей локально настраивать свойства жесткости и прочности в каждой детали.[[5]](#footnote-5)

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

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). Успешное создание модели позволит сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Представлены «датасеты» (datasets) с необходимыми числовыми параметрами, а именно свойствами композитов для проведения операций.

В первой входной таблице «X\_bp» представлены следующие параметры:

Таблица 1 – Параметры с таблицы «X\_bp».

|  |
| --- |
| Соотношение матрица-наполнитель |
| Плотность, кг/м3 |
| модуль упругости, ГПа |
| Количество отвердителя, м.% |
| Содержание эпоксидных групп,%\_2 |
| Температура вспышки, С\_2 |
| Поверхностная плотность, г/м2 |
| Модуль упругости при растяжении, ГПа |
| Прочность при растяжении, МПа |
| Потребление смолы, г/м2 |

Во второй входной таблице «X\_nup» представлены следующие параметры:

Таблица 2 – Параметры с таблицы «X\_nup».

|  |
| --- |
| Угол нашивки, град |
| Шаг нашивки |
| Плотность нашивки |

В зависимости от решаемой задачи, выделяются следующие выходные переменные:

* Соотношение матрица-наполнитель;
* Модуль упругости при растяжении, Гпа;
* Прочность при растяжении, МПа;

При это соответствующе переменные исключаются из входных данных.

Для решения задач датасеты были предварительно объединены   
и приведены к одной длине

Объединение проводилось по индексу, в результате выполнения операции были отрезаны 17 строк массива, однако это составляет 1,6% данных   
и не является существенным.

Далее было установлено отсутствие пропусков. Проанализировано количество выбросов и ввиду их незначительности – удаление из общего массива данных.

При проведении анализа данных было выявлено отсутствие пропусков   
в датасете и незначительное количество выбросов, процент которых   
был достаточно мал и удален из выборки путем удаления строк с этими данными.

* 1. **Описание используемых методов.**

Для выполнения данной задачи были использованы методы машинного обучения. Машинное обучение (ML) — это использование математических моделей данных, которые помогают компьютеру обучаться   
без непосредственных инструкций. Оно считается одной из форм искусственного интеллекта (ИИ). При машинном обучении с помощью алгоритмов выявляются закономерности в данных. На основе этих закономерностей создается модель данных для прогнозирования. Чем больше данных обрабатывает такая модель и чем дольше она используется, тем точнее становятся результаты. Это очень похоже на то, как человек оттачивает навыки на практике.[[6]](#footnote-6)

К основным методам машинного обучения относятся:

* ***Обучение с учителем*** — когда у машины есть некий учитель, который знает, какой ответ правильный. Это значит, что исходные данные   
  уже размечены (отсортированы) нужным образом, и машине остается лишь определить объект с нужным признаком или вычислить результат[[7]](#footnote-7), для каждого прецедента задаётся пара «ситуация, требуемое решение».
* ***Обучение без учителя*** — когда машина сама должна найти среди хаотичных данных верное решение и отсортировать объекты по неизвестным признакам. Например, определить, где на фото собака. (для каждого прецедента задаётся только «ситуация», требуется сгруппировать объекты в [кластеры](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), используя данные о попарном сходстве объектов, и/или понизить размерность данных).
* [***Обучение с подкреплением***](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D0%BF%D0%BE%D0%B4%D0%BA%D1%80%D0%B5%D0%BF%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%D0%BC) — для каждого прецедента имеется пара «ситуация, принятое решение»;

Основные алгоритмы моделей машинного обучения:

1. ***Дерево принятия решений***

Это метод поддержки принятия решений, основанный на использовании древовидного графа: модели принятия решений, которая учитывает   
их потенциальные последствия (с расчётом вероятности наступления того   
или иного события), эффективность, ресурсозатратность.

Для бизнес-процессов это дерево складывается из минимального числа вопросов, предполагающих однозначный ответ — «да» или «нет». Последовательно дав ответы на все эти вопросы, мы приходим к правильному выбору. Методологические преимущества дерева принятия решений – в том,   
что оно структурирует и систематизирует проблему, а итоговое решение принимается на основе логических выводов.

2. ***Наивная байесовская классификация***

Наивные байесовские классификаторы относятся к семейству простых вероятностных классификаторов и берут начало из теоремы Байеса, которая применительно к данному случаю рассматривает функции как независимые   
(это называется строгим, или наивным, предположением). На практике используется в следующих областях машинного обучения:

* определение спама, приходящего на электронную почту;
* автоматическая привязка новостных статей к тематическим рубрикам;
* выявление эмоциональной окраски текста;
* распознавание лиц и других паттернов на изображениях.

3. ***Метод наименьших квадратов***

К вариантам реализации линейной регрессии относятся наименьшие квадраты. Обычно с помощью линейной регрессии решают задачи по подгонке прямой, которая проходит через множество точек. Вот как это делается   
с помощью метода наименьших квадратов: провести прямую, измерить расстояние от неё до каждой из точек (точки и линию соединяют вертикальными отрезками), получившуюся сумму перенести наверх. В результате та кривая,   
в которой сумма расстояний будет наименьшей, и есть искомая (эта линия пройдёт через точки с нормально распределённым отклонением от истинного значения).

Линейная функция обычно используется при подборе данных   
для машинного обучения, а метод наименьших квадратов – для сведения   
к минимуму погрешностей путем создания метрики ошибок.

4***. Логистическая регрессия***

Логистическая регрессия – это способ определения зависимости между переменными, одна из которых категориально зависима, а другие независимы. Для этого применяется логистическая функция (аккумулятивное логистическое распределение). Практическое значение логистической регрессии заключается  
 в том, что она является мощным статистическим методом предсказания событий, который включает в себя одну или несколько независимых переменных. Это востребовано в следующих ситуациях:

* кредитный скоринг;
* замеры успешности проводимых рекламных кампаний;
* прогноз прибыли с определённого товара;
* оценка вероятности землетрясения в конкретную дату.

5. ***Метод опорных векторов (SVM)***

Это целый набор алгоритмов, необходимых для решения задач   
на классификацию и регрессионный анализ. Исходя из того, что объект, находящийся в N-мерном пространстве, относится к одному из двух классов, метод опорных векторов строит гиперплоскость с мерностью (N – 1), чтобы   
все объекты оказались в одной из двух групп. На бумаге это можно изобразить так: есть точки двух разных видов, и их можно линейно разделить. Кроме сепарации точек, данный метод генерирует гиперплоскость таким образом, чтобы она была максимально удалена от самой близкой точки каждой группы.

SVM и его модификации помогают решать такие сложные задачи машинного обучения, как сплайсинг ДНК, определение пола человека   
по фотографии, вывод рекламных баннеров на сайты.

6***. Метод ансамблей***

Он базируется на алгоритмах машинного обучения, генерирующих множество классификаторов и разделяющих все объекты из вновь поступающих данных на основе их усреднения или итогов голосования. Изначально метод ансамблей был частным случаем байесовского усреднения, но затем усложнился и оброс дополнительными алгоритмами:

* бустинг (boosting) – преобразует слабые модели в сильные посредством формирования ансамбля классификаторов (с математической точки зрения это является улучшающим пересечением);
* бэггинг (bagging) – собирает усложнённые классификаторы,   
  при этом параллельно обучая базовые (улучшающее объединение);
* корректирование ошибок выходного кодирования.

Метод ансамблей – более мощный инструмент по сравнению с отдельно стоящими моделями прогнозирования, поскольку:

* он сводит к минимуму влияние случайностей, усредняя ошибки каждого базового классификатора;
* уменьшает дисперсию, поскольку несколько разных моделей, исходящих из разных гипотез, имеют больше шансов прийти к правильному результату, чем одна отдельно взятая;
* исключает выход за рамки множества: если агрегированная гипотеза оказывается вне множества базовых гипотез, то на этапе формирования комбинированной гипотезы оно расширяется при помощи того или иного способа, и гипотеза уже входит в него.

7. ***Алгоритмы кластеризации***

Кластеризация заключается в распределении множества объектов   
по категориям так, чтобы в каждой категории – кластере – оказались наиболее схожие между собой элементы.

Кластеризировать объекты можно по разным алгоритмам. Чаще всего используют следующие:

* на основе центра тяжести треугольника;
* на базе подключения;
* сокращения размерности;
* плотности (основанные на пространственной кластеризации);
* вероятностные;
* машинное обучение, в том числе нейронные сети.

Алгоритмы кластеризации используются в биологии (исследование взаимодействия генов в геноме, насчитывающем до нескольких тысяч элементов), социологии (обработка результатов социологических исследований методом Уорда, на выходе дающим кластеры с минимальной дисперсией   
и примерно одинакового размера) и информационных технологиях.

8. ***Метод главных компонент (PCA)***

Метод главных компонент, или PCA, представляет собой статистическую операцию по ортогональному преобразованию, которая имеет своей целью перевод наблюдений за переменными, которые могут быть как-то взаимосвязаны между собой, в набор главных компонент – значений, которые линейно   
не коррелированы.

Практические задачи, в которых применяется PCA, – визуализация   
и большинство процедур сжатия, упрощения, минимизации данных для того, чтобы облегчить процесс обучения. Однако метод главных компонент не годится для ситуаций, когда исходные данные слабо упорядочены (то есть   
все компоненты метода характеризуются высокой дисперсией). Так что   
его применимость определяется тем, насколько хорошо изучена и описана предметная область.

***9. Сингулярное разложение***

В линейной алгебре сингулярное разложение, или SVD, определяется как разложение прямоугольной матрицы, состоящей из комплексных   
или вещественных чисел. Так, матрицу M размерностью [m\*n] можно разложить таким образом, что M = UΣV, где U и V будут унитарными матрицами,   
а Σ – диагональной.

Одним из частных случаев сингулярного разложения является метод главных компонент. Самые первые технологии компьютерного зрения разрабатывались на основе SVD и PCA и работали следующим образом: вначале лица (или другие паттерны, которые предстояло найти) представляли в виде суммы базисных компонент, затем уменьшали их размерность, после чего производили их сопоставление с изображениями из выборки. Современные алгоритмы сингулярного разложения в машинном обучении, конечно, значительно сложнее и изощрённые, чем их предшественники, но суть   
их в целом нем изменилась.[[8]](#footnote-8)

Заказчиком было поставлено три задачи для решения: две задачи регрессии , атак же задача на создание рекомендательной системы с использованием обучения модели с помощью нейронной сети.

Для решения поставленных задач были использованы следующие методы машинного обучения:

1. ***Линейная регрессия*** – используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной «y» от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) «x» с линейной функцией зависимости.[[9]](#footnote-9)
2. ***Случайный лес*** – алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана, и метод случайных подпространств, предложенный Тин Кам Хо. Алгоритм применяется для задач классификации, регрессии   
   и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.[[10]](#footnote-10)

Для решения задачи создания рекомендательной системы «Соотношение матрица – наполнитель» был использован ***метод многослойного персептрона***:

Рекомендательная система – это комплекс сервисов, программ, алгоритмов, который анализирует предпочтения пользователей и пытается предсказать, что может их заинтересовать.[[11]](#footnote-11)

***Многослойный персептрон (MLP)*** — это алгоритм обучения с учителем, который изучает функцию:



Рисунок 3 – Функция, которую изучает MLP

Обучение на наборе данных, где m — количество измерений для ввода и o- количество размеров для вывода.

Учитывая набор функций «X =*x1, x2,…,xm»* и цель «у» , 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 1 показан MLP с одним скрытым слоем   
со скалярным выходом.[[12]](#footnote-12)

Особенностью является наличие более чем одного обучаемого слоя   
(как правило — два или три). Необходимость в большом количестве обучаемых слоёв отпадает, так как теоретически единственного скрытого слоя достаточно, чтобы перекодировать входное представление таким образом, чтобы получить линейную разделимость для выходного представления.

Существует предположение, что, используя большее число слоёв, можно уменьшить число элементов в них, то есть суммарное число элементов в слоях будет меньше, чем если использовать один скрытый слой. Это предположение успешно используется в технологиях глубокого обучения и имеет обоснование.[[13]](#footnote-13)

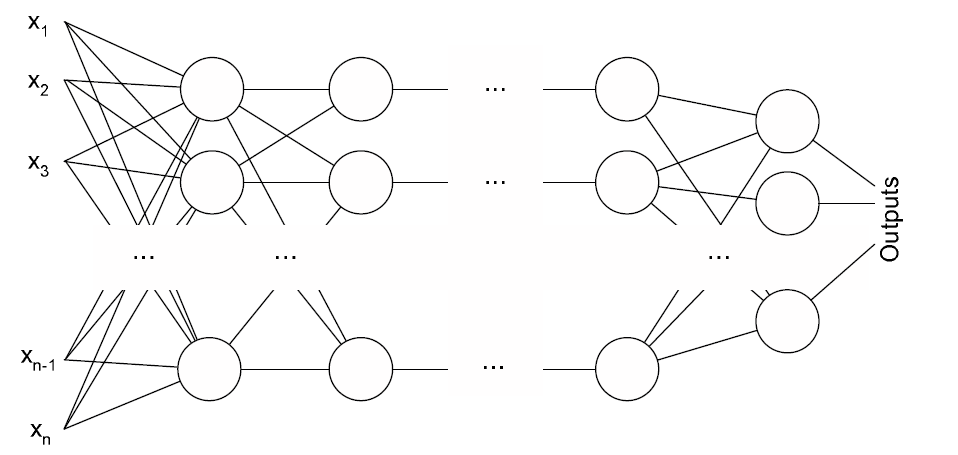


Рисунок 4 – Архитектура многослойного перцептрона

Задачи, предоставленные для исследования, исследуются   
с использованием следующих методов и библиотек Python:

1. ***NumPy*** - это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами. Основным объектом NumPy является однородный многомерный массив.   
   Это таблица элементов (обычно чисел), всех одного типа, индексированных последовательностями натуральных чисел.[[14]](#footnote-14)
2. ***Matplotlib*** – это обширная библиотека для создания статических, анимированных и интерактивных визуализаций в Python.[[15]](#footnote-15)
3. ***Pandas*** – программная библиотека на языке Python для обработки   
   и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.[[16]](#footnote-16)
4. ***Scikit-learn (Sklearn)*** - полезная и надежная библиотека   
   для машинного обучения на Python. Он предоставляет ряд эффективных инструментов для машинного обучения и статистического моделирования, включая классификацию, регрессию, кластеризацию и уменьшение размерности с помощью интерфейса согласованности на языке Python.[[17]](#footnote-17)
5. ***Seaborn*** — популярная библиотека готовых шаблонов   
   для статистической визуализации, написанная на бэкенде matplotlib.[[18]](#footnote-18)
6. ***Tensorflow*** – открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия. Имеет комплексную и гибкую экосистему инструментов, библиотек и ресурсов сообщества, которая позволяет исследователям внедрять самые современные технологии машинного обучения.[[19]](#footnote-19)
7. ***Math*** - библиотека содержит наиболее применяемые математические функции и константы. Все вычисления происходят на множестве вещественных чисел.[[20]](#footnote-20)
   1. **Разведочный анализ данных.**

Перед началом составление моделей необходимом провести разведочный анализ данных (Exploratory Data Analysis (EDA)). Является одним из первых   
и определяющих шагов проекта науки о данных, который приводит в движение весь проект.[[21]](#footnote-21)

Используем следующие методы Python и описательной статистики:

df = pd.read\_excel – чтение и загрузка данных в DataFrame;



Рисунок 5 – Загрузка исходных данных

df = df\_nup.join – объединение столбцов другого фрейма данных;[[22]](#footnote-22)



Рисунок 6 – объединение двух таблиц

df.info() – данный метод отображает информацию о фрейме данных, включая индекс, тип дпнных и столбцы, ненулевые значения и использование памяти.

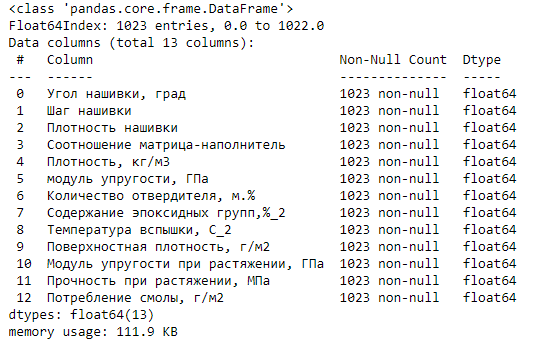


Рисунок 7– результат выполнения df.info()

sorted(df[column].unique() – определение уникальности значений;

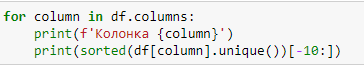


Рисунок 8 – вывод уникальных 10 последних значений

df.isna().sum() – суммарное количество пропусков;

df.isna().mean() – процентное соотношение пропусков;

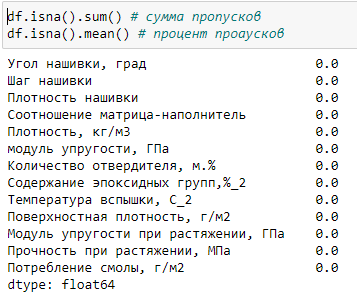


Рисунок 9 – результат исполнения df.isna().sum()., df.isna().mean()

df.duplicated().sum() – возвращает логические ряды, обозначающие повторяющиеся строки (проверка на дубликаты);

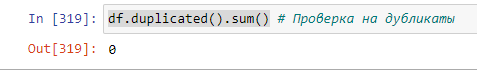


Рисунок 10 – Проверка на дубликаты

df.describe() – оценивает столбцы с числовым типом данных (int или float) и возвращает статистические данные, которые дают представление   
о распределении значений;

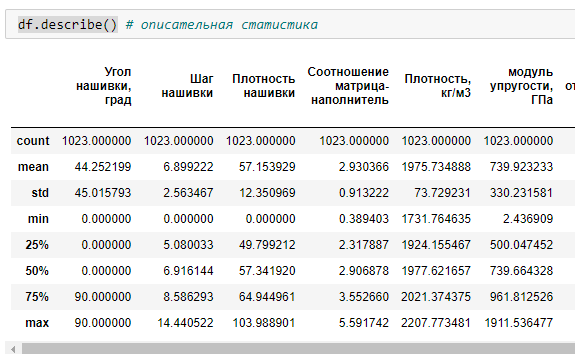


Рисунок 11 – результат выполнения df.describe()

Следующим шагом были построены гистограммы плотности распределения значений каждой переменной, а именно 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.[[23]](#footnote-23)

По результатам выполнения данной операции наблюдается факт нормального распределения для большинства характеристик, однако наблюдаются также и исключения.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Рисунок 12 – Примеры построения гистограмм для некоторых характеристик, где распределение близко к нормальному.

Выявленные исключения:

* Поверхностная плотность, г/м2 - нормальное распределение   
  со смещением

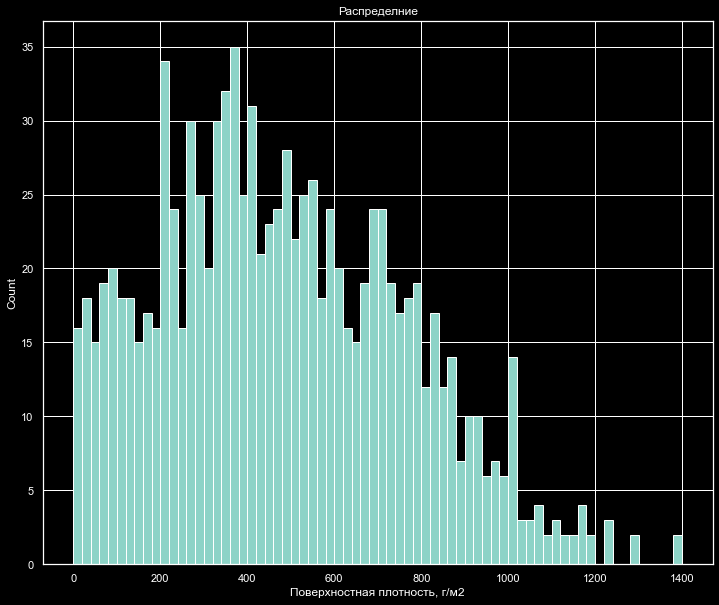
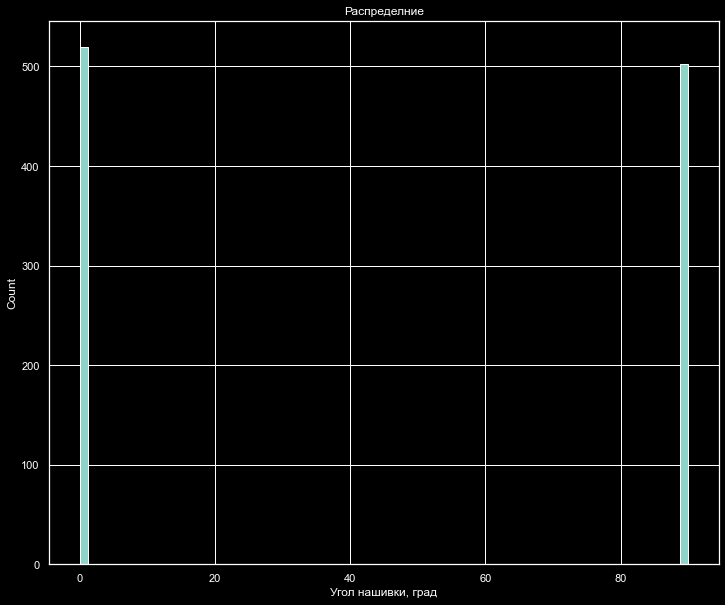


Рисунок 13. Гистограмма параметра   
(поверхностная плотность, г/м2)

* Угол нашивки, град – используется как вещественная переменная дискретное распределение (дискретное распределение данных по параметру), колонка содержит два уникальных значения.

  
Рисунок 14. Гистограмма параметра   
(Угол нашивки, град)

Далее для поиска выбросов построены диаграммы распределения «Ящики с усами», они позволят разбить данные на квантили (Отсортированная последовательность состоит из семи чисел, поэтому медианой является четвертое число в ряду. Его также называют 0.5-квантилем[[24]](#footnote-24)):25, 50, 75, 100, что в свою очередь располагает использование статистического метода трех сигм - при нормальном распределении практически все значения величины   
с вероятностью 0,9973 лежат не далее трех сигм в любую сторону   
от математического ожидания, то есть находятся в диапазоне :  
*[М(х) - 3σ ... М(х) + 3σ,* где M(x) – математическое ожидание (среднее значение случайной величины), а «*σ*» - стандартное   
или среднеквадратичное отклонение (наиболее частый показатель рассеивания значений величины относительно математического ожидания. Обозначается символом σ, который произносится как «сигма».)[[25]](#footnote-25)

|  |  |
| --- | --- |
|  |  |
| Рисунок 15 – диаграмма рассеивания,  демонстрирующая выбросы с двух сторон значений | |
|  |  |
| Рисунок 16 – диаграмма рассеивания,  демонстрирующая выбросы со стороны юольших значений | |

Далее для исследования выстраиваем график рассеивания точек,   
в котором наблюдается слабая зависимость между переменными, однако можно предположить, что их связывает сложная нелинейная связь.

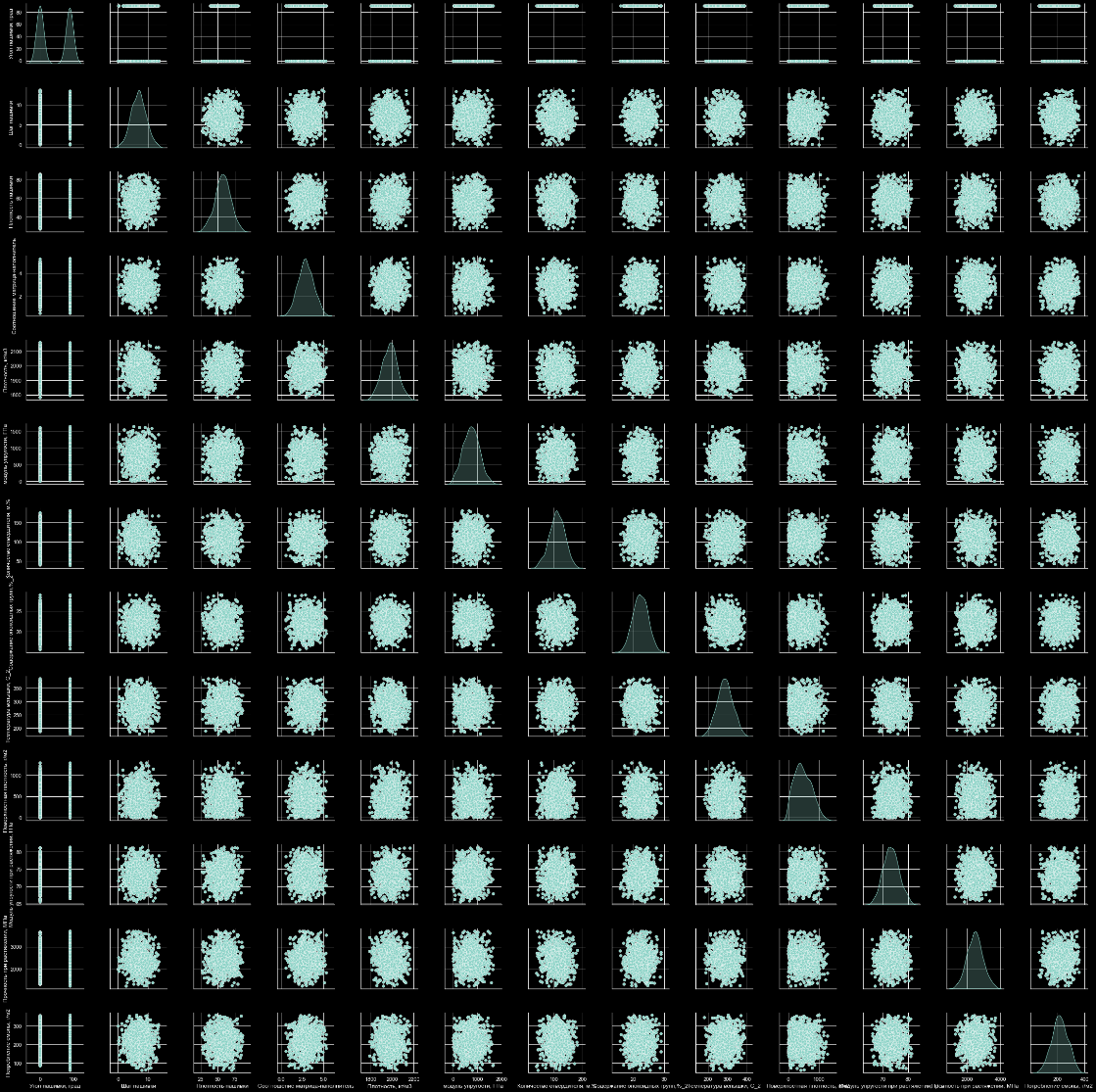


Рисунок 17 – попарные графики рассеивания

Рассчитываем коэффициент корреляции (статистическая взаимосвязь двух или более случайных величин[[26]](#footnote-26)). При этом изменения значений одной или нескольких из этих величин сопутствуют систематическому изменению значений другой или других величин. для каждого   
из параметров, строя после тепловую карту коэффициентов корреляции.

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.[[27]](#footnote-27)



Рисунок 18 – Расчет коэффициентов корреляции

Строим корреляционную матрицу с помощью модуля «seaborn». Тепловая карта используется для графического представления значений матрицы с разными цветовыми оттенками для разных значений. Он очень четко визуализирует общую матрицу. Тепловая карта хорошо демонстрирует отсутствие корреляция у переменных предоставленного дата фрейма. Большинство коэффициентов корреляции находятся   
в помежутке значений от - 0,25 до 0,25. Вывод: входные данные подобраны таким образом, что каждая из них несет свою функцию при воздействии   
на композит.

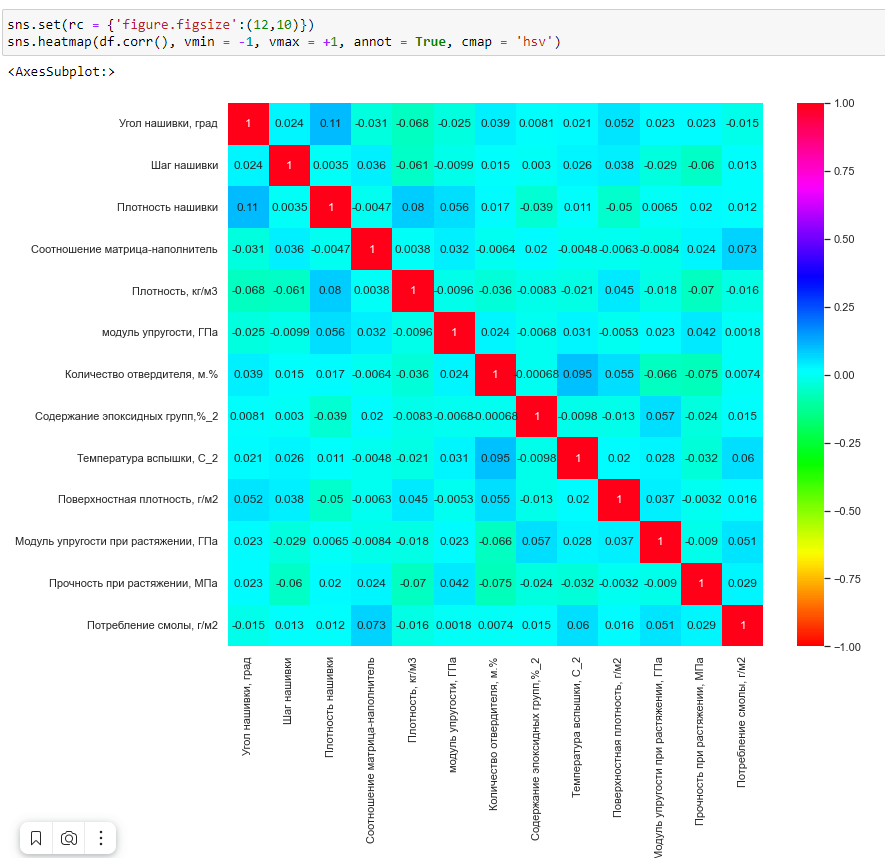


Рисунок 19 – пример построения   
тепловой карты коэффициента корреляции.

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

Обрабатываем данные. Для этого отчищаем данные от выбросов, используя:

* функцию расчета значения в квантилях;
* замена выбросов на нулевые значения;

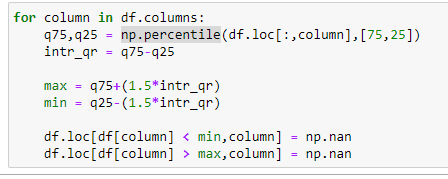


Рисунок 20 – Функция расчета значения в квантилях.

* выявление количества выбросов (df.isnull().sum());
* удаление выбросов (df\_used = df.dropna(axis = 0));
* проверка на отсутствие выбросов (df\_used.isna().sum());

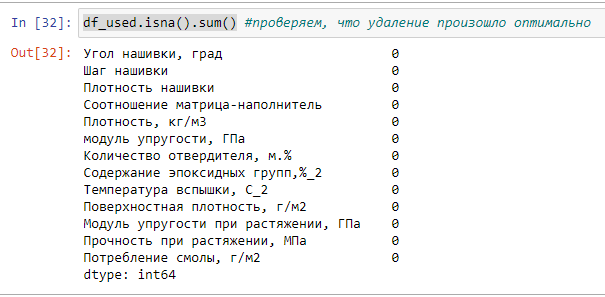


Рисунок 21 – Результат проведенной операции  
 (выбросы отсутствуют)

Далее с помощью графического исполнения убеждаемся, что выбросы отсутствуют:

|  |  |
| --- | --- |
|  |  |
|  |  |

Рисунок 22 – выборочные графики «catplot»   
после предобработки данных

Нормализуем значение с применением «MinMaxScaler» - преобразование объектов путем масштабирования каждого объекта   
до заданного диапазона. Этот оценщик масштабирует и переводит каждую функцию индивидуально так, чтобы она находилась в заданном диапазоне на обучающем наборе, например, между нулем и единицей.[[28]](#footnote-28)

Также стандартизируем датасет с помощью функции «preprocessing.scale» - помещает ваши данные в одну шкалу. Это полезно для очень разреженных наборов данных. Проблема с разреженностью заключается в том, что она очень предвзята или в статистических терминах искажена. Таким образом, масштабирование данных переносит все ваши значения в одну шкалу, устраняя разреженность. Что касается того, как   
это работает в математических деталях, это следует той же концепции нормализации и стандартизации.[[29]](#footnote-29)

Для оценки факторов, влияющих на обработанный массив информации, представленным для исследования используем метод главных компонент. Анализ главных компонентов (Principal Component Analysis – PCA) представляет собой алгебраический метод преобразования набора наблюдений возможно коррелированных переменных в набор значений линейных некоррелированных переменных.

Все главные компоненты выбраны для описания большей части доступной дисперсии переменной, и все главные компоненты ортогональны друг другу. В наборах главного компонента первый главный компонент всегда будет иметь максимальную дисперсию.[[30]](#footnote-30)

Одним из инструментов метода главных компонент является факторный анализ данных. Главными целями факторного анализа являются: (1) сокращение числа переменных (редукция данных) и (2) определение структуры взаимосвязей между переменными, т.е. классификация переменных. Поэтому факторный анализ используется или как метод сокращения данных или как метод классификации.[[31]](#footnote-31) Для проведения факторного анализа данные были стандартизированы.

В процессе стандартизации происходит формирование стандартизированных шкал. Стандартизация позволяет устранить возможное влияние отклонений по какому-либо признаку. Стандартизация приводит все исходные значения набора данных, независимо от их начальных распределений и единиц измерения, к набору значений из распределения   
с нулевым средним и стандартным отклонением, равным 1. В результате формируется так называемая стандартизированная шкала, которая определяет место каждого значения в наборе данных, измеряя его отклонение от среднего в единицах стандартного отклонения. Значения стандартизированной шкалы определяются следующим образом:[[32]](#footnote-32)



Рисунок 23 – Значение стандартизированной шкалы

Таблица 3 – расшифровка значения стандартизированной шкалы.

|  |  |
| --- | --- |
|  | исходное значение признака |
|  | среднее значение и стандартное отклонение признака, оцененные по набору данных. |

Проведено преобразование данных в вектора от двух до пяти факторов.   
После выполнения данной операции можно предположить, что такие характеристики, как: *количество отвердителя, плотность нашивки, плотность, угол нашивки, потребление смолы* - являются наиболее подходящими параметрами для построения моделей.

* 1. **Разработка и обучение модели.**

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

* Модель линейной регрессии
* Модель «Случайный лес»

Для обучения моделей линейной регрессии была использована функция: LinearRegression(), а в качестве метрик оценки качества:

R2 - (коэффициент детерминации) представляет собой коэффициент того, насколько хорошо совпадают значения по сравнению с исходными значениями. Значение от 0 до 1 интерпретируется как проценты. Чем выше значение, тем лучше модель.

MAE - (средняя абсолютная ошибка) представляет собой разницу между исходными и прогнозируемыми значениями, извлеченными путем усреднения абсолютной разницы по набору данных.[[33]](#footnote-33)

Для обучения моделей случайный лес были настроены следуюие параметры:

* Количество деревьев ('n\_estimators': [20, 40, 60]); использование множества деревьев для соответствия модели поможет нам получить более обобщенный результат, но это не всегда так. Однако это не приведет   
  к переобучению, но, безусловно, может увеличить временную сложность модели. В scikit-learn количество оценщиков по умолчанию равно 100.[[34]](#footnote-34)
* Число признаков для выбора расщепления: ('max\_features': ['auto', 'sqrt', 'log2']); случайный лес выбирает случайные подмножества функций   
  и пытается найти лучшее разделение. max\_features помогает найти количество функций, которые следует учитывать, чтобы сделать наилучшее разбиение. Может принимать четыре значения: « авто », « sqrt », « log2 » и « Нет» .
* Максимальная глубина деревьев (количество ветвлений или уровней деревьев): ('max\_depth' : [3,4,5,6]); когда дело доходит до повышения точности модели, поскольку мы увеличиваем глубину дерева, точность модели увеличивается до определенного предела, но затем она начнет постепенно уменьшаться из-за переобучения модели. Важно правильно установить   
  его значение, чтобы избежать переобучения. Значение по умолчанию установлено на None, None указывает, что узлы внутри дерева будут продолжать расти, пока все листья не станут чистыми или все листья не будут содержать меньше min\_samples\_split (другой гиперпараметр).
* Функцией GridSearchCV (инструмент для автоматического подбирания параметров для моделей машинного обучения. GridSearchCV находит наилучшие параметры, путем обычного перебора: он создает модель для каждой возможной комбинации параметров. Важно отметить, что такой подход может быть весьма времязатратным.[[35]](#footnote-35)) и параметром cv (сколько разрезов кросс-валидации мы ходим сделать[[36]](#footnote-36)) = 10 задано количество блоков.
  1. **Тестирование модели.**

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | | Средняя абсолютная ошибка | Коэффициент детерминации |
| Модуль упругости при растяжении | Линейная регрессия | 0.148908 | -37.247571 |
| Случайный лес | 0.149932 | -47.374302 |
| Прочность при растяжении | Линейная регрессия | 0.150917 | -32.786002 |
| Случайный лес | 0.149897 | -107.726809 |

Рисунок 24 – результат использования  
 выбранных моделей

Вывод – модели демонстрирую скромные результаты обучения, что подводит нас к началу исследования моделей для устранения возможных ошибок и реализации удовлетворительных результатов обучения.

* 1. **Построение нейронной сети.**

Для построения модели для рекомендательной системы был использован многослойный персептрон (нейронная сеть прямого распространения сигнала (без обратных связей), в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько слоев. Первый из таких слоев называют входным, последний - выходным[[37]](#footnote-37)). На вход подавались нормализованные данные, модель включала один входной слой с активационной функцией Relu и 256 нейронов. Далее были настроены два скрытых слоя:

* Первый – 256 нейронов, активационная функция Relu;
* Второй - 64 нейрона, активационная функция – линейная регрессия;

Выходной слой с 1 нейроном, так как на выходе выводится одно значение для введенных данных. Для оптимизации был применен метод Adam (достаточно часто применяется при обучении нейронных сетей и хорошо себя зарекомендовал для поиска минимума функций потерь. Фактически, этот алгоритм является очередной модификацией алгоритма Adagrad, использующий сглаженные версии среднего и среднеквадратического градиентов.   
Для его применения в Tensorflow достаточно создать такой оптимизатор и указать требуемые параметры, например, так (opt = tf.optimizers.Adam  
(learning\_rate=0.1)).[[38]](#footnote-38)

Для оценки качества модели был применена метрика MSE – среднеквадратическая ошибка. Среднеквадратичная ошибка (Mean Squared Error) – Среднее арифметическое (Mean) квадратов разностей между предсказанными и реальными значениями Модели (Model) Машинного обучения (ML). MSE практически никогда не равен нулю, и происходит   
это из-за элемента случайности в данных или неучитывания Оценочной функцией (Estimator) всех факторов, которые могли бы улучшить предсказательную способность.[[39]](#footnote-39)

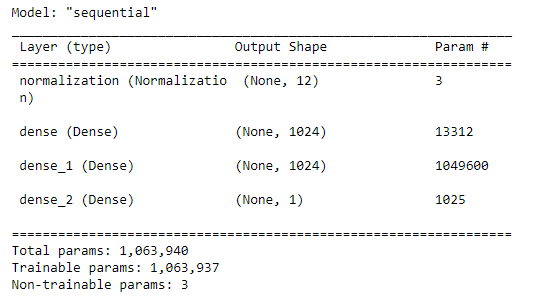


Рисунок 25. Архитектура нейронной сети

Модель обучалась на 100 эпохах. Такая продолжительность циклов не дала лучшего результата, соответственно модель может обучаться на меньшем количестве циклов. Результаты обучения модели характеризуются изменением среднеквадратической ошибки.

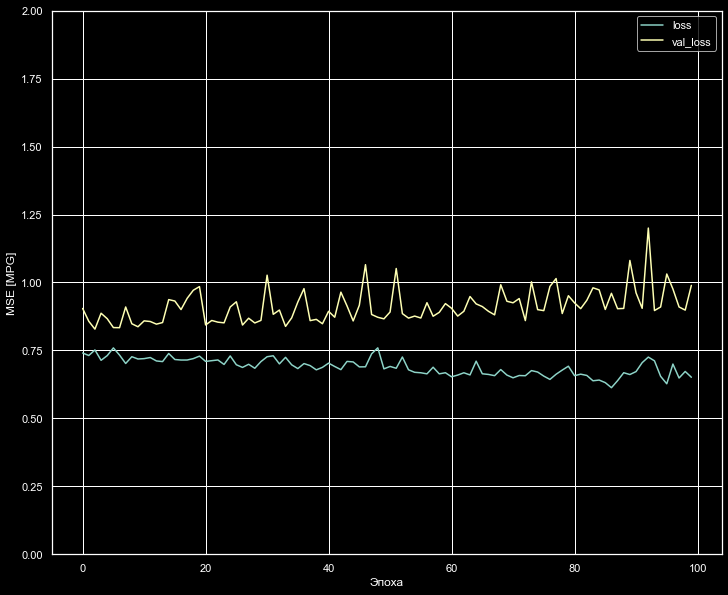


Рисунок 26. Результат изменения среднеквадратической ошибки

* 1. **Разработка приложения**

Разработано вэб-приложение для рекомендательной системы «Соотношение матрица-наполнитель». Приложение разработано в среде разработки VS Code. Для разработки приложения был использован интерпретатор Python для запуска веб-приложения Flask.

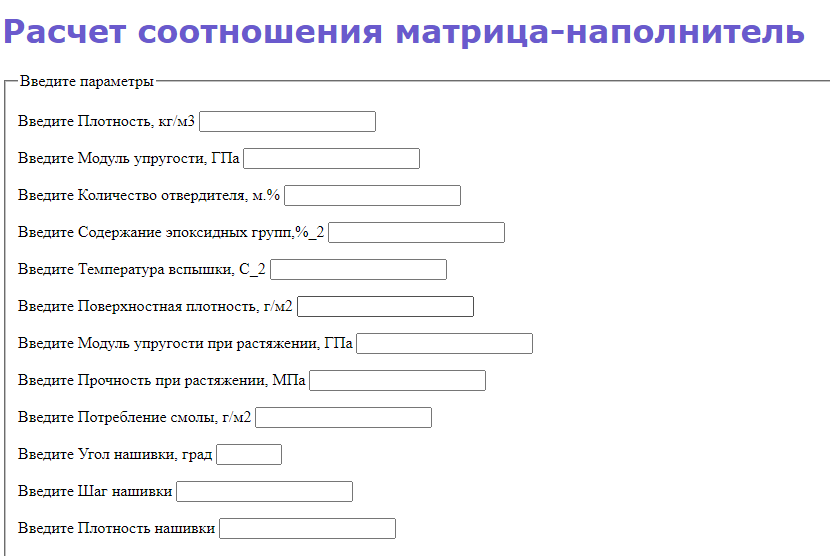


Рисунок 27 – Инициализация работы программы

* 1. **Репозиторий**

Созданный репозиторий с результатами работы и комплектом документов находится по адресу https://github.com/hmoistramp/VKR\_project

# **Заключение**

По результатам исследования были изучены теоретические основы методов машинного обучения, изучены основные библиотеки Python, их применение

Были проведены мероприятия по стандартизации данных, построению моделей и их обучению, однако достигнут не внушительные результаты – направляющие на повторное вникание в суть вопроса и разбор применения методов машинного обучения, создания нейронных сетей. Данные курса позволили определить вектор дальнейшего обучения.

# **Список литературы**

1. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>.
2. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>
3. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>.
4. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>.
5. Документация по библиотеке sklearn: – Режим доступа: <https://scikit-learn.org/stable/user_guide.html>.
6. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>.
7. Язык программирования Python- Режим доступа: https://www.python.org/. (дата обращения 15.04.2022)
8. Среда разработки Jupyter Notebook- Режим доступа: https://jupyter.org/. (дата обращения 15.04.2022)
9. Библиотека Tensorflow: Режим доступа: https://www.tensorflow.org/. (дата обращения 15.04.2022)
10. Метрики и оценки: количественная оценка качества прогнозов: - Режим доступа: https://scikit-learn.ru/3-3-metrics-and-scoring-quantifying-the-quality-of-predictions/

1. Классификация композиционных материалов и их роль в современном машиностроении УДК 621.43.068.5 [↑](#footnote-ref-1)
2. https://fishki.net/4001985-likbez-po-kompozitnym-materialam.html [↑](#footnote-ref-2)
3. https://habr.com/ru/post/362189/ [↑](#footnote-ref-3)
4. https://rostec.ru/news/kompozitnaya-istoriya/ [↑](#footnote-ref-4)
5. https://compositeworld.ru/articles/market/id60535f4f937fe40013900a95 [↑](#footnote-ref-5)
6. https://azure.microsoft.com/ru-ru/overview/what-is-machine-learning-platform/ [↑](#footnote-ref-6)
7. https://trends.rbc.ru/trends/industry/60c85c599a7947f5776ad409 [↑](#footnote-ref-7)
8. https://habr.com/ru/post/448892/ [↑](#footnote-ref-8)
9. https://ru.wikipedia.org/wiki/Линейная\_регрессия [↑](#footnote-ref-9)
10. https://ru.wikipedia.org/wiki/Random\_forest [↑](#footnote-ref-10)
11. https://cloud.yandex.ru/blog/posts/2022/05/recommendation-system-instruction [↑](#footnote-ref-11)
12. https://scikit-learn.ru/1-17-neural-network-models-supervised/ [↑](#footnote-ref-12)
13. https://ru.wikipedia.org/wiki/Многослойный\_перцептрон\_Румельхарта [↑](#footnote-ref-13)
14. https://habr.com/ru/post/121031/ [↑](#footnote-ref-14)
15. https://matplotlib.org/ [↑](#footnote-ref-15)
16. https://pandas.pydata.org/ [↑](#footnote-ref-16)
17. https://www.tutorialspoint.com/scikit\_learn/scikit\_learn\_quick\_guide.htm#:~:text=Scikit-learn%20(Sklearn)%20-%20самая%20полезная,интерфейса%20согласованности%20на%20языке%20Python [↑](#footnote-ref-17)
18. https://mipt-stats.gitlab.io/courses/python/09\_seaborn.html [↑](#footnote-ref-18)
19. https://www.tensorflow.org/ [↑](#footnote-ref-19)
20. https://pythonchik.ru/matematika/modul-math-v-python [↑](#footnote-ref-20)
21. https://waksoft.susu.ru/2021/07/23/issledovatelskij-analiz-dannyh-v-python-rukovodstvo-dlya-novichkov-na-2021-god/ [↑](#footnote-ref-21)
22. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.join.html [↑](#footnote-ref-22)
23. https://studref.com/438270/menedzhment/grafik\_protekaniya\_protsessa [↑](#footnote-ref-23)
24. https://habr.com/ru/post/556040/ [↑](#footnote-ref-24)
25. https://wiki.fenix.help/matematika/pravilo-trekh-sigm [↑](#footnote-ref-25)
26. https://ru.wikipedia.org/wiki/Корреляция [↑](#footnote-ref-26)
27. https://www.delftstack.com/ru/howto/seaborn/correlation-heatplot-seaborn-python/ [↑](#footnote-ref-27)
28. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html [↑](#footnote-ref-28)
29. https://stackoverflow.com/questions/42325212/what-preprocessing-scale-do-how-does-it-work [↑](#footnote-ref-29)
30. https://pythonpip.ru/osnovy/analiz-glavnyh-komponentov-pca-na-python [↑](#footnote-ref-30)
31. http://statsoft.ru/home/textbook/modules/stfacan.html [↑](#footnote-ref-31)
32. https://wiki.loginom.ru/articles/data-standartization.html [↑](#footnote-ref-32)
33. https://www.datatechnotes.com/2019/10/accuracy-check-in-python-mae-mse-rmse-r.html [↑](#footnote-ref-33)
34. https://progler.ru/blog/giperparametry-klassifikatora-sluchaynogo-lesa [↑](#footnote-ref-34)
35. https://vc.ru/ml/147132-kak-avtomaticheski-podobrat-parametry-dlya-modeli-mashinnogo-obucheniya-ispolzuem-gridsearchcv#:~:text=GridSearchCV%20–%20это%20очень%20мощный,подход%20может%20быть%20весьма%20времязатратным [↑](#footnote-ref-35)
36. https://newtechaudit.ru/gridsearchcv-pomoshhnik-v-vybore/ [↑](#footnote-ref-36)
37. https://intuit.ru/studies/courses/6/6/lecture/178?page=5#:~:text=Многослойный%20персептрон%20(MLP)%20-%20нейронная,называют%20входным%2C%20последний%20-%20выходным [↑](#footnote-ref-37)
38. https://proproprogs.ru/tensorflow/tf-stroim-gradientnye-algoritmy-optimizacii-adam-rmsprop-adagrad-adadelta [↑](#footnote-ref-38)
39. https://www.helenkapatsa.ru/sriedniekvadratichieskaia-oshibka/ [↑](#footnote-ref-39)