

**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

Мавлютова Альфия Галиановна

Москва, 2022

Содержание

Содержание.....	2
Введение	3
1. Аналитическая часть	5
1.1. Постановка задачи	5
1.2. Описание используемых методов	7
1.3. Разведочный анализ данных.....	13
2. Практическая часть.....	15
2.1. Предобработка данных.....	15
2.2. Разработка и обучение модели.....	15
2.3. Тестирование модели.....	16
2.4. Написать нейронную сеть, которая будет рекомендовать.....	17
2.6. Создание удалённого репозитория и загрузка.....	17
2.7. Заключение.....	18
2.8. Список используемой литературы и веб ресурсы.....	18

Введение

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. Объединение делать по индексу тип объединения INNER

https://drive.google.com/file/d/1B1s5gBlvgU81H9GGolLQVw_SOivyNf2/view?usp=sharing

Требуется:

1. Изучить теоретические основы и методы решения поставленной задачи.
2. Провести разведочный анализ предложенных данных. Необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами,

попарные графики рассеяния точек. Необходимо также для каждой колонке получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков.

3. Провести предобработку данных (удаление шумов, нормализация и т.д.).

4. Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. При построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.

5. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.

6. Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5 (один или два прогноза, на выбор учащегося).

7. Оценить точность модели на тренировочном и тестовом датасете.

8. Создать репозиторий в GitHub / GitLab и разместить там код исследования. Оформить файл README.

1. Аналитическая часть

1.1. Постановка задачи

Исходные данные представляют собой два файла в формате .xlsx (1023,11) и (1040,4). Объединив их и удалив неинформативный столбец с индексом, получили (1023, 13)

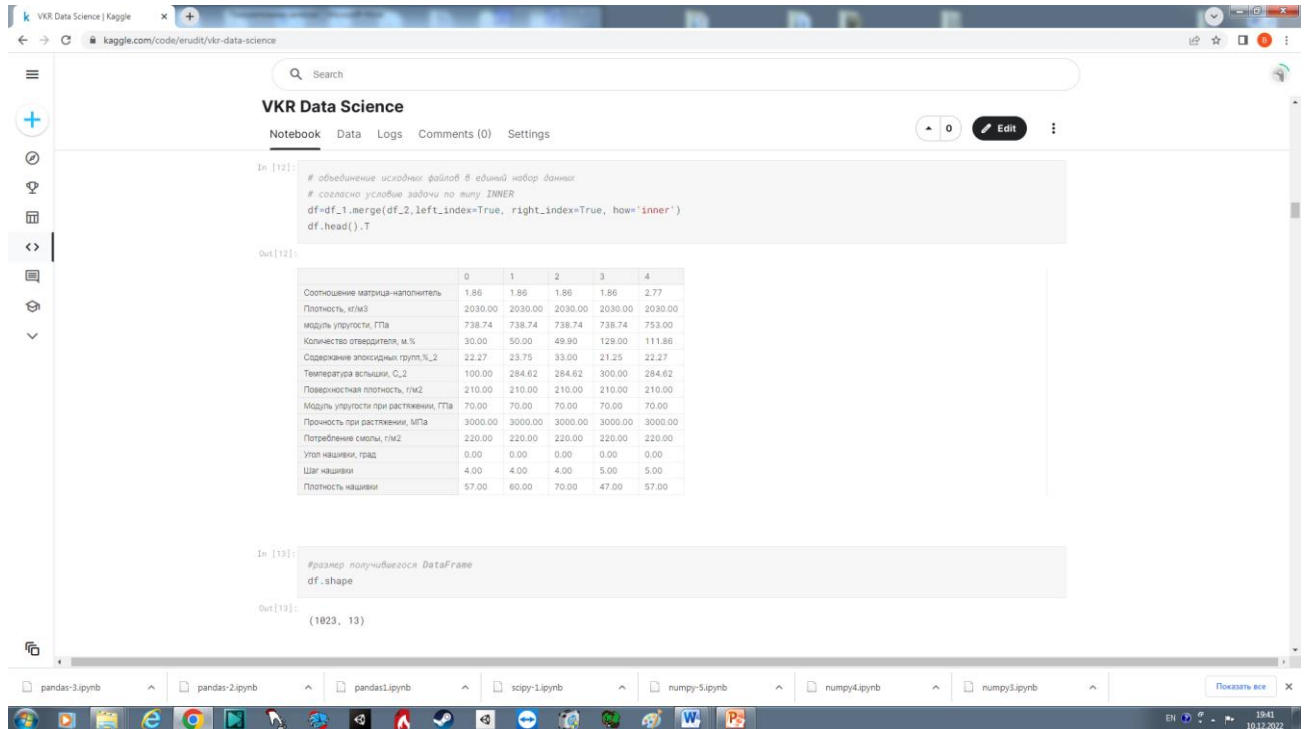


Рисунок 1

*В итоговом датасете 17 строк второго датасета не рассматриваем.

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

Затем проводим разведочный анализ данных:

- просмотр начальных и конечных строк датасета
- округление с указанием количества цифр, вводимых после запятой
- удаление первого столбца с порядковыми номерами (неинформативный)
- изучение информации о датасете
- проверка типов данных в столбцах
- проверка на наличие пропусков

- проверка уникальных значений (функция `nunique`)

Изучаем максимальные минимальные значения, квантили, медианы и т.д.

Визуализируем данные посредством гистограммы распределения каждой переменной, диаграммы «ящик с усами» и тепловой карты.

Исходя из графиков видим, что параметры в большинстве близки к нормальному распределению. Исключение составляет «Угол нашивки», т.к. значения равны 0 или 1. Выбросы наблюдаются в каждом столбце, кроме «Угол нашивки»

Корреляция входных переменных очень слабая.



Рисунок 2

Проводим анализ и исключение выбросов, проверяем на наличие пропусков, проводим операции по нормализации и стандартизации.

Обучаем алгоритм машинного обучения для прогноза модуля упругости при растяжении и прочности при растяжении. Пишем нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разрабатываем прило-

жение, которое будет выдавать прогноз, полученный в предыдущей поставленной задаче. Оцениваем точность модели на тренировочном и тестовом датасете. Создаем репозиторий в GitHub и размещаем полученные результаты исследования.

1.2. 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:

- метод опорных векторов;
- случайный лес;
- линейная регрессия;
- градиентный бустинг;
- К-ближайших соседей;
- дерево решений;
- стохастический градиентный спуск;
- многослойный перцептрон;
- Лассо;

Метод опорных векторов (Support Vector Regression) – этот бинарный линейный классификатор был выбран, потому что он хорошо работает на небольших датасетах. Данный алгоритм – это алгоритм обучения с учителем, использующихся для задач классификации и регрессионного анализа, это контролируемое обучение моделей с использованием схожих алгоритмов для анализа данных и распознавания шаблонов. Учитывая обучающую выборку, где алгоритм помечает каждый объект, как принадлежащий к одной из двух категорий, строит модель, которая определяет новые наблюдения в одну из категорий.

Модель метода опорных векторов – отображение данных точками в пространстве, так что между наблюдениями отдельных категорий имеется разрыв, и он максимален.

Каждый объект данных представляется как вектор (точка) в r -мерном пространстве. Он создаёт линию или гиперплоскость, которая разделяет данные на классы.

Достоинства метода: для классификации достаточно небольшого набора данных. При правильной работе модели, построенной на тестовом множестве, вполне возможно применение данного метода на реальных данных. Эффективен при большом количестве гиперпараметров. Способен обрабатывать случаи, когда гиперпараметров больше, чем количество наблюдений. Существует возможность гибко настраивать разделяющую функцию. Алгоритм максимизирует разделяющую полосу, которая, как подушка безопасности, позволяет уменьшить количество ошибок классификации.

Недостатки метода: неустойчивость к шуму, поэтому в работе была проведена тщательнейшая работа с выбросами, иначе в обучающих данных шумы становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости; для больших наборов данных требуется долгое время обучения; достаточно сложно подбирать полезные преобразования данных; параметры модели сложно интерпретировать, поэтому были рассмотрены и другие методы.

Случайный лес (RandomForest) — это множество решающих деревьев. Универсальный алгоритм машинного обучения с учителем, представитель ансамблевых методов. Если точность дерева решений оказалась недостаточной, мы можем множество моделей собрать в коллектив.

Достоинства метода: не переобучается; не требует предобработки входных данных; эффективно обрабатывает пропущенные данные, данные с большим чис-

лом классов и признаков; имеет высокую точность предсказания и внутреннюю оценку обобщающей способности модели, а также высокую параллелизуемость и масштабируемость.

Недостатки метода: построение занимает много времени; сложно интерпретируемый; не обладает возможностью экстраполяции; может недо обучаться; трудоёмко прогнозируемый; иногда работает хуже, чем линейные методы.

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными. Это один из самых простых и эффективных инструментов статистического моделирования. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R^2 , или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если R-квадрат равен 0,5, модель объясняет лишь 50 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R^2 к единице, тем лучше.

Достоинства метода: быстр и прост в реализации; легко интерпретируем; имеет меньшую сложность по сравнению с другими алгоритмами;

Недостатки метода: моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

Градиентный бустинг (Gradient Boosting) — это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компен-

сирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов.

Достоинства метода: новые алгоритмы учатся на ошибках предыдущих; требуется меньше итераций, чтобы приблизиться к фактическим прогнозам; наблюдения выбираются на основе ошибки; прост в настройке темпа обучения и применения; легко интерпретируем.

Недостатки метода: необходимо тщательно выбирать критерии остановки, иначе это может привести к переобучению; наблюдения с наибольшей ошибкой появляются чаще; слабее и менее гибок чем нейронные сети.

Метод ближайших соседей - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значения целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами. Алгоритм находит расстояния между запросом и всеми примерами в данных, выбирая определенное количество примеров (k), наиболее близких к запросу, затем голосует за наиболее часто встречающуюся метку (в случае задачи классификации) или усредняет метки (в случае задачи регрессии).

Достоинства метода: прост в реализации и понимании полученных результатов; имеет низкую чувствительность к выбросам; не требует построения модели; допускает настройку нескольких параметров; позволяет делать дополнительные допущения; универсален; находит лучшее решение из возможных; решает задачи небольшой размерности.

Недостатки метода: замедляется с ростом объёма данных; не создаёт правил; не обобщает предыдущий опыт; основывается на всем массиве доступных исторических данных; невозможно сказать, на каком основании строятся ответы; сложно выбрать близость метрики; имеет высокую зависимость результатов классификации от выбранной метрики; полностью перебирает всю обучающую выборку при распознавании; имеет вычислительную трудоёмкость.

Дерево принятия решений (DecisionTreeRegressor) – метод автоматического анализа больших массивов данных. Это инструмент принятия решений, в котором используется древовидная структура, подобная блок-схеме, или модель решений и всех их возможных результатов, включая результаты, затраты и полезность. Дерево принятия решений - эффективный инструмент интеллектуального анализа данных и предсказательной аналитики. Алгоритм дерева решений подпадает под категорию контролируемых алгоритмов обучения. Он работает как для непрерывных, так и для категориальных выходных переменных. Правила генерируются за счёт обобщения множества отдельных наблюдений (обучающих примеров), описывающих предметную область. Регрессия дерева решений отслеживает особенности объекта и обучает модель в структуре дерева прогнозированию данных в будущем для получения значимого непрерывного вывода. Дерево решений один из вариантов решения регрессионной задачи, в случае если зависимость в данных не имеет очевидной корреляции.

Достоинства метода: помогают визуализировать процесс принятия решения и сделать правильный выбор в ситуациях, когда результаты одного решения влияют на результаты следующих решений; создаются по понятным правилам; просты в применении и интерпретации; заполняют пропуски в данных наиболее вероятным решением; работают с разными переменными; выделяют наиболее важные поля для прогнозирования;

Недостатки метода: ошибаются при классификации с большим количеством классов и небольшой обучающей выборкой; имеют нестабильный процесс (изменение в одном узле может привести к построению совсем другого дерева); имеет затратные вычисления; необходимо обращать внимание на размер; ограниченное число вариантов решения проблемы.

Стохастический градиентный спуск (SGDRegressor) — это простой, но очень эффективный подход к подгонке линейных классификаторов и регрессоров под выпуклые функции потерь. Этот подход подразумевает корректировку весов нейронной сети, используя аппроксимацию градиента функционала, вычисленную только на одном случайном обучающем примере из выборки.

Достоинства метода: эффективен; прост в реализации; имеет множество возможностей для настройки кода; способен обучаться на избыточно больших выборках.

Недостатки метода: требует ряд гиперпараметров; чувствителен к масштабированию функций; может не сходиться или сходиться слишком медленно; функционал многоэкстремален; процесс может "застрять" в одном из локальных минимумов; возможно переобучение.

Многослойный персептрон (MLPRegressor) — это алгоритм обучения с учителем, который изучает функцию $f(\cdot): R_m \rightarrow R_o$ обучением на наборе данных, где m — количество измерений для ввода и o — количество размеров для вывода. Это искусственная нейронная сеть, имеющая 3 или более слоёв персептронов. Эти слои - один входной слой, 1 или более скрытых слоёв и один выходной слой персептронов.

Достоинства метода: построение сложных разделяющих поверхностей; возможность осуществления любого отображения входных векторов в выходные; легко обобщает входные данные; не требует распределения входных векторов; изучает нелинейные модели.

Недостатки метода: имеет невыпуклую функцию потерь; разные инициализации случайных весов могут привести к разной точности проверки; требует настройки ряда гиперпараметров; чувствителен к масштабированию функций.

Лассо регрессия (Lasso) — это линейная модель, которая оценивает разреженные коэффициенты. Это простой метод, позволяющий уменьшить сложность модели и предотвратить переопределение, которое может возникнуть в результате простой линейной регрессии. Данный метод вводит дополнительное слагаемое регуляризации в оптимизацию модели. Это даёт более устойчивое решение. В регрессии лассо добавляется условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, используется смещение абсолютного значения. Лассо регрессия хорошо прогнозирует модели временных рядов на основе регрессии, таким как авторегрессии.

Достоинства метода: легко полностью избавляется от шумов в данных; быстро работает; не очень энергоёмко; способно полностью убрать признак из датасета; доступно обнуляет значения коэффициентов.

Недостатки метода: выбор модели не помогает и обычно вредит; часто страдает качество прогнозирования; выдаёт ложное срабатывание результата; случайным образом выбирает одну из коллинеарных переменных; не оценивает правильность формы взаимосвязи между независимой и зависимой переменными; не всегда лучше, чем пошаговая регрессия.

1.3. Разведочный анализ данных

Перед обучением модели, необходимо данные обработать и очистить, т.к. это может привести к неверным результатам. Некоторые выбросы должны быть исключены из набора данных, так как их причинами являются ошибки и технические неполадки, другие выбросы необходимо оставить в наборе данных. Например, выброс не является результатом ошибки и дает новое понимание тестируемого

явления. Научные эксперименты особенно чувствительны к выбросам- исключив выброс по ошибке, можно пропустить новую тенденцию или открытие.

Поиск выбросов с помощью кластеризации, где выбросы- это элементы малых групп, с помощью моделей предсказания (метод опорных векторов и вариации решающих деревьев)

Неинформативные признаки: если признак несет слишком много строк с одинаковыми значениями, он не несет полезной информации.

Цель разведочного анализа - получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

Инструменты разведочного анализа:

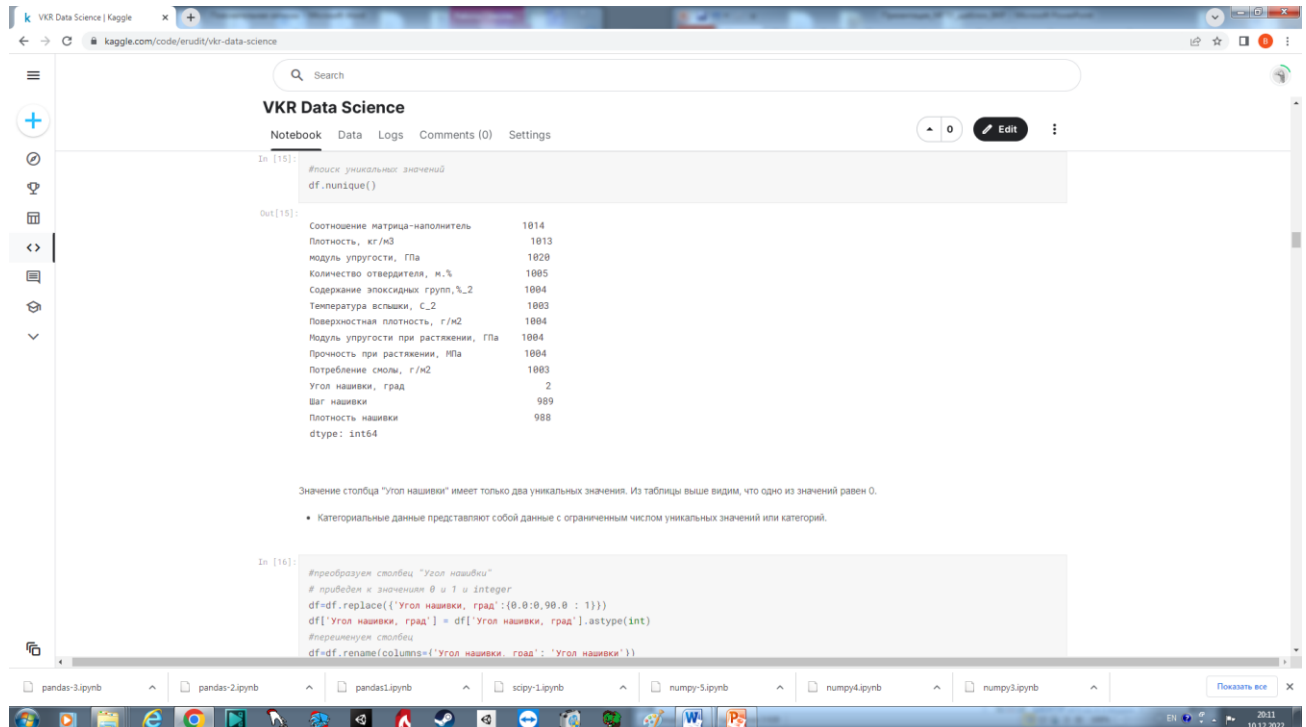
- оценка статистических характеристик датасета;
- гистограммы распределения каждой из переменной;
- диаграммы ящика с усами
- попарные графики рассеяния точек;
- описательная статистика для каждой переменной;
- анализ и полное исключение выбросов;
- проверка наличия пропусков и дубликатов;
- ранговая корреляция Пирсона.

После обнаружения выбросов данные, которые заметно отличаются от выборки будут полностью удалены. Для расчёта этих данных используется метод трех сигм и межквартильного расстояния.

2. Практическая часть

2.1. Предобработка данных

Значение столбца "Угол нашивки" имеет только два уникальных значения. Из первоначальной таблицы видим, что одно из значений равно 0. Преобразуем данный столбец и приведем к значениям «0» и «1».



The screenshot shows a Kaggle notebook with the following content:

```
In [15]: #поиск уникальных значений
df.unique()
```

Out[15]:

Соотношение матрица-наполнитель	1014
Плотность, кг/м3	1013
модуль упругости, ГПа	1029
Количество отвердителя, %	1005
Содержание эпоксидных групп, %_2	1004
Температура вспышки, C_2	1003
Поверхностная плотность, г/м2	1004
Модуль упругости при растяжении, ГПа	1004
Прочность при растяжении, МПа	1004
Потребление смолы, г/м2	1003
Угол нашивки, град	2
Шаг нашивки	989
Плотность нашивки	988
dtype:	int64

Значение столбца "Угол нашивки" имеет только два уникальных значения. Из таблицы выше видим, что одно из значений равно 0.

- Категориальные данные представляют собой данные с ограниченным числом уникальных значений или категорий.

```
In [16]: #преобразуем столбец "Угол нашивки"
# заменяем на значения 0 и 1 = integer
df=df.replace({'Угол нашивки, град':(0,0,90,0 : 1)})
df['Угол нашивки, град'] = df['Угол нашивки, град'].astype(int)
#переименовываем столбец
df=df.rename(columns={'Угол нашивки, град': 'Угол нашивки'})
```

По условиям задания нормализуем значения.

Для этого применим MinMaxScaler() и Normalizer().

2.2. Разработка и обучение модели

Разработка и обучение моделей машинного обучения должна быть выполнена для двух параметров: «Прочность при растяжении» и «Модуль упругости при растяжении».

Порядок разработки модели для каждого параметра:

- разделение нормализованных данных на обучающую и тестовую выборки (в соотношении 70 на 30%);
- проверка моделей при стандартных значениях;
- сравнение с результатами модели, выдающей среднее значение;
- создание графика;
- сравнение моделей по метрике MAE;
- поиск сетки гиперпараметров, по которым будет происходить оптимизация модели.
- оптимизация подбора гиперпараметров модели с помощью выбора по сетке и перекрёстной проверки;
- подстановка оптимальных гиперпараметров в модель и обучение модели на тренировочных данных;
- оценка полученных данных;
- сравнение со стандартными значениями.

2.3. Тестирование модели

После обучения моделей необходимо провести оценка точности моделей на обучающей и тестовых выборках.

2.4. Написать нейронную сеть, которая будет рекомендовать соотношение «матрица – наполнитель».

Обучение нейронной сети— процесс, при котором происходит подбор оптимальных параметров модели, с точки зрения минимизации функционала ошибки.

Построение нейронной сети осуществим с помощью класса `keras.Sequential`.

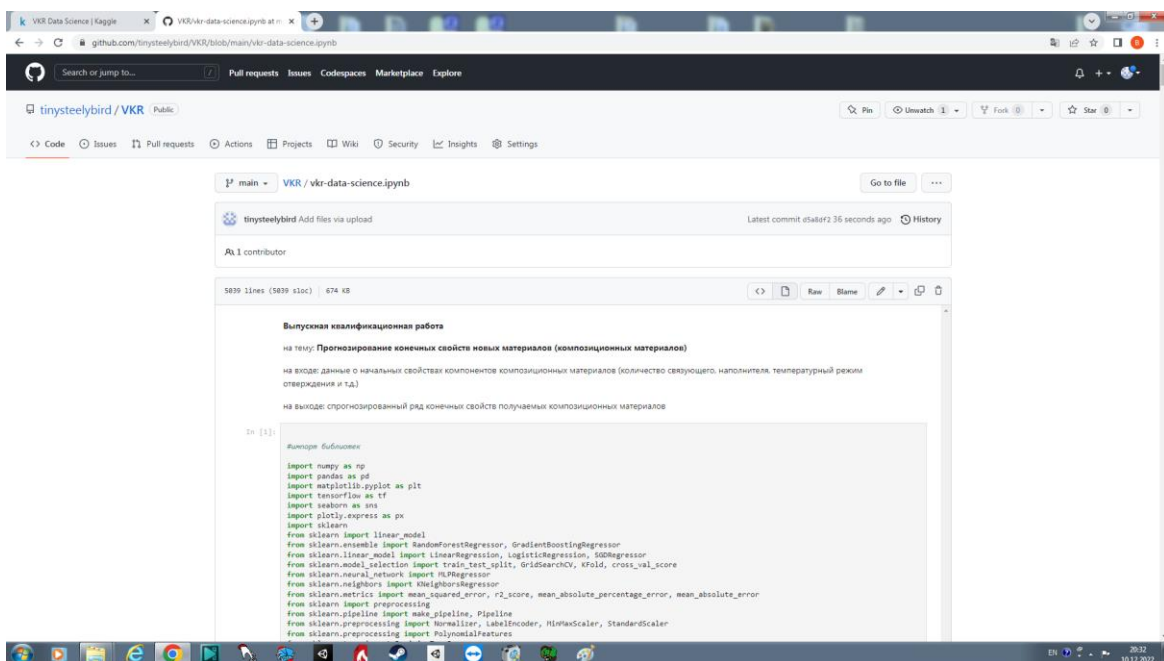
Обучим и оценим модель, посмотрим на потери, зададим функцию для визуализации факт/прогноз для результатов моделей.

2.5. Разработка приложения

Полученное приложение должно показывать результат прогноза для соотношения «матрица – наполнитель».

2.6. Создание удалённого репозитория и загрузка

Репозиторий был создан на github.com по адресу:
<https://github.com/tinysteelybird/VKR>



2.7. Заключение

В ходе обучения и проведения исследовательской работы по поставленной задаче уровень знаний и навыков в изучаемой области существенно возрос. Появилось понимание нейронных сетей, прогнозирования, анализа данных. Поставлена задача самостоятельно углубить знания в области статистики, программировании и машинному обучению, чтобы достичь лучших результатов.

2.8. Список используемой литературы и веб ресурсы.

1. Лекционный материал
2. <https://github.com/>
3. <https://www.kaggle.com/>