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:** «**Анализ и прогнозирование параметров лазерной фрезеровки нержавеющей стали»**

Слушатель Штамм Елена Васильевна

Москва, 2024

# Содержание

[Содержание 2](#_Toc157779734)

[Введение 3](#_Toc157779735)

[1. Аналитическая часть 4](#_Toc157779736)

[1.1. Постановка задачи 4](#_Toc157779737)

[1.2. Описание используемых методов 5](#_Toc157779738)

[1.3 Разведочный анализ данных 9](#_Toc157779739)

[2. Практическая часть 14](#_Toc157779740)

[2.1. Предобработка данных 14](#_Toc157779741)

[2.2. Разработка и обучение модели. 15](#_Toc157779742)

[2.3. Тестирование модели. 17](#_Toc157779743)

[2.4. Написать нейронную сеть, которая будет выдавать параметр шероховатости Ra 18](#_Toc157779744)

[2.5. Разработка приложения 20](#_Toc157779745)

[2.6. Создание удаленного репозитория и загрузка результатов работы в него. 21](#_Toc157779746)

[Заключение 22](#_Toc157779747)

[Список используемой литературы и веб ресурсы 23](#_Toc157779748)

# Введение

Лазерная фрезеровка – новый и перспективный способ обработки металла. Данный метод позволяет получить на поверхности металла рельеф(форму), заданную 3D моделью с очень высокой точностью – размер «инструмента» составляет порядка 25-30 мкм, что недостижимо при механической обработке.

Основная проблема процесса фрезеровки – очень узкие диапазоны технологических параметров обработки для получения оптимальных результатов. Таким образом, при подборе режимов необходимо проведение огромного количества экспериментов. Для ограничения числа тестов, необходимо использовать анализ с использованием методов машинного обучения.

Для проведения аналитической работы были предоставлены данные экспериментов отдела Разработки лазерных технологий компании-производителя оборудования.

Объектом исследования является нержавеющая сталь 12Х18Н9, которая используются в промышленности

Входные параметры обработки, задаваемые оператором: мощность излучения, частота и длительность импульсов, скорость сканирования луча, количество проходов. Выходные (измеряемые после обработки) – глубина фрезеровки, шероховатости поверхности, выраженные через Ra и Rz.

В процессе исследовательской работы были разработаны несколько моделей. На основе одного из алгоритмов машинного обучения было создано доступное пользовательское веб - приложение на фреймворке Flask.

# Аналитическая часть

## Постановка задачи

Для исследовательской работы был предоставлен технологический отчет о подборе режимов лазерной фрезеровки. Выполнено 480 экспериментов, зафиксированы 14 показателей по каждому.

(DATA\_F.xlsx)

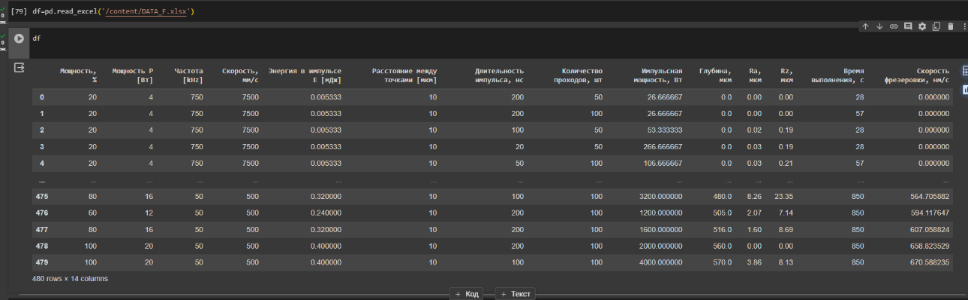


Рисунок Пример начала работы с файлом DATA\_F.xlsx

Необходимо провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек.

Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; пред обработать данные: удалить шумы и выбросы, сделать нормализацию и стандартизацию. Обучить несколько моделей для прогноза выходных параметров. Написать нейронную сеть. Разработать приложение с графическим интерфейсом. Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования. Оформить файл README.

## 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:

* Ансамблевый метод суммарное обобщение - [StackingRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingRegressor.html#sklearn.ensemble.StackingRegressor)
* линейная регрессия;
* К-ближайших соседей;
* метод опорных векторов;
* случайный лес;
* многослойный перцептрон;

1. Ансамблевый метод [StackingRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingRegressor.html#sklearn.ensemble.StackingRegressor) - это метод комбинирования оценок для уменьшения их ошибок. Точнее, прогнозы каждого отдельного оценщика складываются вместе и используются в качестве входных данных для окончательного оценщика для вычисления прогноза. Этот окончательный оценщик обучается посредством перекрестной проверки.
2. 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. R2 , или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат равен 0,5, модель объясняет лишь 50 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R2 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.

1. 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.

1. 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 использование 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 p-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, необходима 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.

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

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

1. Многослойный персептрон (MLPRegressor) — это алгоритм обучения с учителем, который изучает функцию f(⋅):Rm→Ro 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.

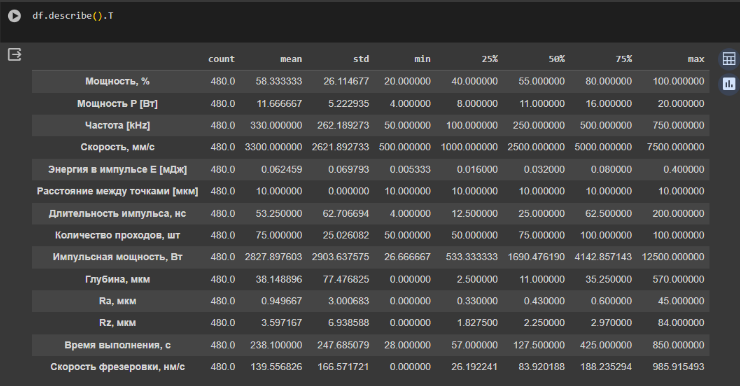
Недостатки метода: имеет невыпуклую функцию потерь; разные инициализации случайных весов могут привести к разной точности проверки; требует настройки ряда гиперпараметров; чувствителен к масштабированию функций.

## Разведочный анализ данных

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.

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной (несколько различных вариантов); диаграммы ящика с усами (несколько интерактивных вариантов); попарные графики рассеяния точек (несколько вариантов); график «квантиль-квантиль»; тепловая карта (несколько вариантов); описательная статистика для каждой переменной; анализ и полное исключение выбросов (5 повторных итераций); проверка наличия пропусков и дубликатов; ранговая корреляция Кендалла и Пирсона.

Описательная статистика дала информацию о том, что параметр Расстояние между точками [мкм] имеет нулевую дисперсию, поскольку задан константой. Удаляем этот столбец.

 Рисунок Описательная статистика датасета

Парные графики позволяют судить о характере распределения данных.

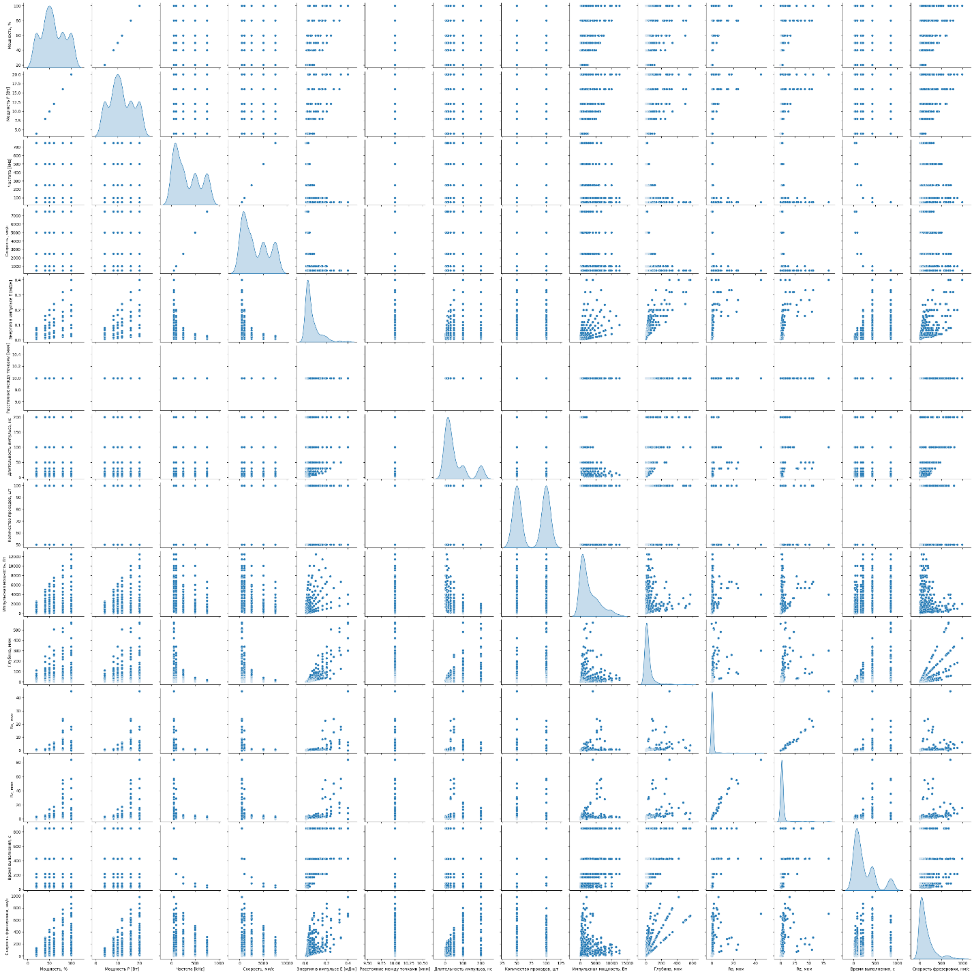


Рисунок Парные графики

При нулевых значениях показателей скорости фрезеровки, глубины, Ra и Rz эксперимент считается неудачным, поскольку процесс фрезеровки не произошел.

Следует удалить строки с нулевыми значениями указанных параметров. Данные строки не несут аналитической информации, а с технологической точки зрения такой режим не имеет смысла.

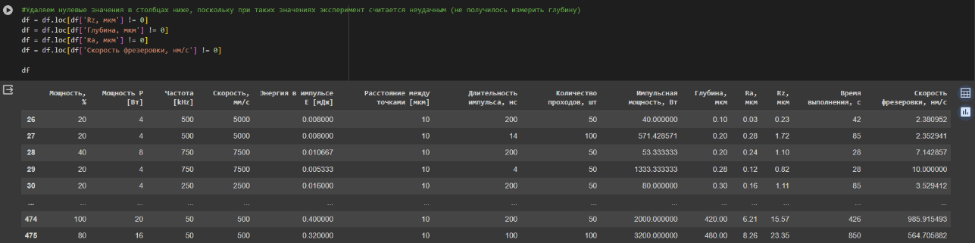
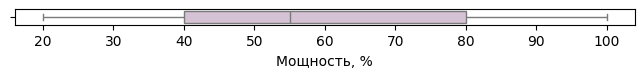


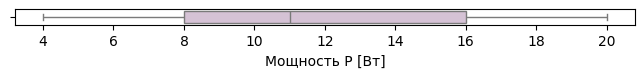
Рисунок Удаляем нулевые значения

Анализируя диаграммы «ящик с усами», на первый взгляд, можно сделать вывод о большом количестве выбросов. Проконсультировавшись с исполнителем экспериментов, удалось выяснить, что такие данные нельзя считать выбросами. Напротив, это могут быть лучшие режимы обработки (производительность, качество поверхности), но из-за узости оптимальных диапазонов параметров, их (режимов) мало, поэтому они могут казаться выбросами.

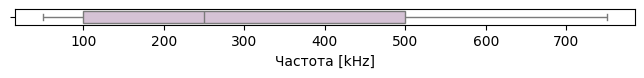
Мощность, %



Мощность P [Вт]



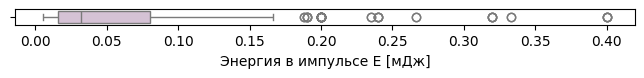
Частота [kHz]



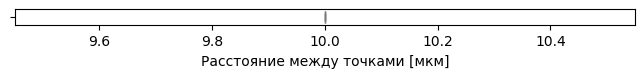
Скорость, мм/с



Энергия в импульсе E [мДж]



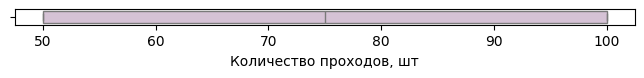
Расстояние между точками [мкм]



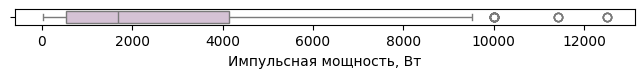
Длительность импульса, нс



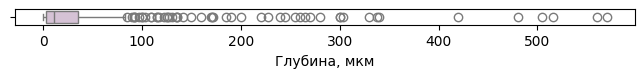
Количество проходов, шт



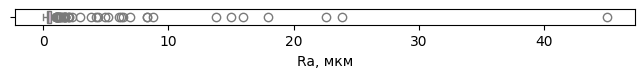
Импульсная мощность, Вт



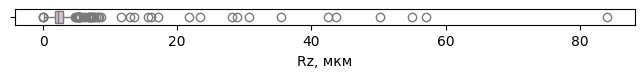
Глубина, мкм



Ra, мкм



Rz, мкм



Время выполнения, с



Скорость фрезеровки, нм/с

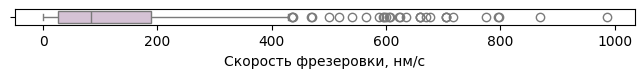


Рисунок Диаграммы "Ящик с усами"

Также строим тепловую карту корреляций, чтобы обнаружить зависимости с выходными переменными Rа и Rz.

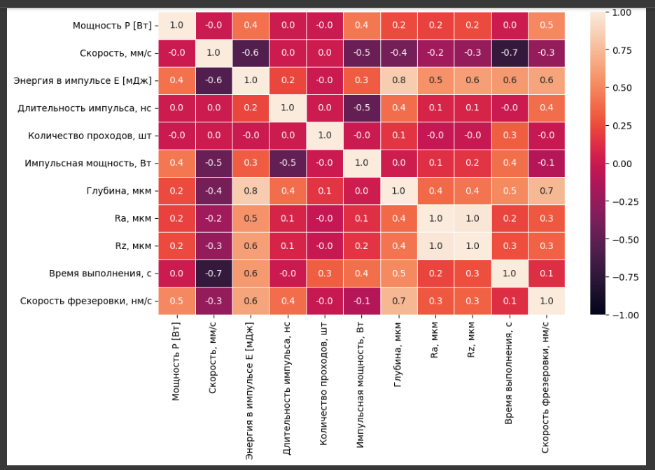


Рисунок Тепловая карта корреляций

Максимальную корреляцию мы наблюдаем между выходными параметрами. Можно сделать вывод о том, что один показатель рассчитывается с использованием другого. Для дальнейшего анализа выходной переменной считаем только Ra. Чтобы избежать искажения результата, получение завышенных показателей оценки, удаляем из выборки столбец Rz.

# Практическая часть

## Предобработка данных

Категориальных данных в нашем датасете нет, поэтому деление датасета на категориальные и числовые для их отдельной обработки не требуется. Предобработка данных будет заключаться в применении StandatdScaller как первого шага в Пайплайне.

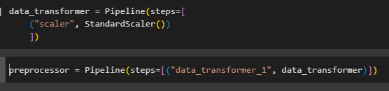


Рисунок Предобработка данных

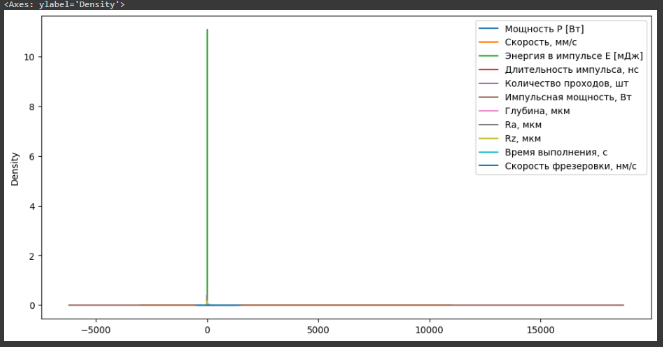


Рисунок Рапредление данных до стандартизации

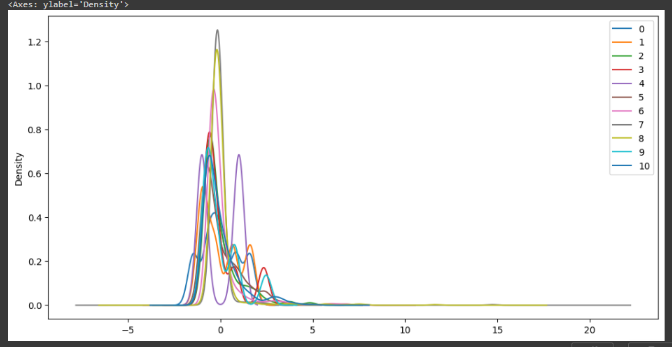


Рисунок Распределение данных после стандартизации

## Разработка и обучение модели.

Разработка и обучение моделей машинного обучения осуществлялась для параметра Ra. Для решения применим все методы, описанные выше.

Порядок разработки модели можно разделить на следующие этапы: разделение нормализованных данных на обучающую и тестовую выборки (в соотношении 70 на 30%); формирование пайплайна для алоритмов и ансамбля алгоритмов, проверка моделей при стандартных значениях; сравнение моделей по метрике МАЕ и R2; поиск сетки гиперпараметров, по которым будет происходить оптимизация модели. В качестве параметра оценки выбран коэффициент детерминации (R2); 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.

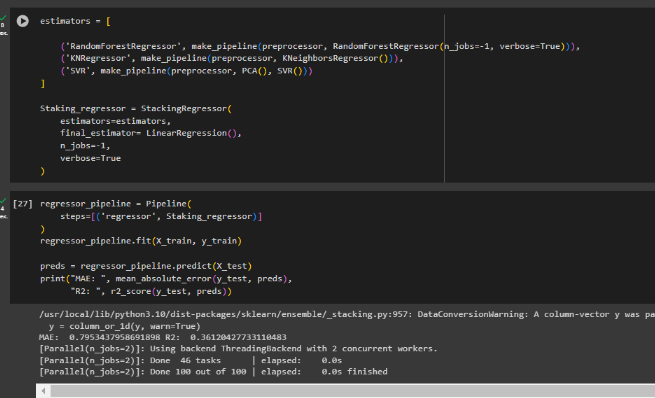


Рисунок Разработка алгоритма StackingRegressor

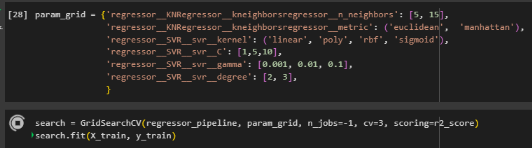


Рисунок Настройка гиперпараметров

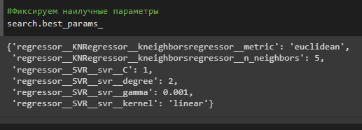
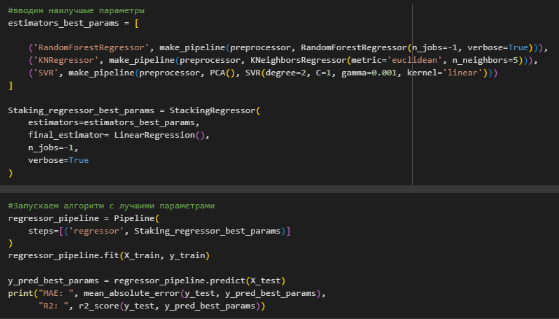


Рисунок Наилучшие гиперпараметры

После настройки гиперпараметров модель показала результат лучше, чем базовая. Коэффициент детерминации (R2) равен 0,93



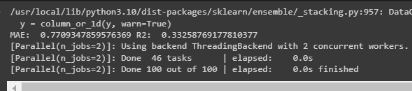


Рисунок StackingRegressor с наилучшими параметрами

## Тестирование модели.

После обучения моделей была проведена оценка точности этих моделей на обучающей и тестовых выборках. В качестве параметра оценки модели использовался коэффициент детерминации R2 и средняя абсолютная ошибка(MAE).

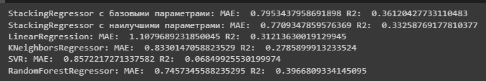
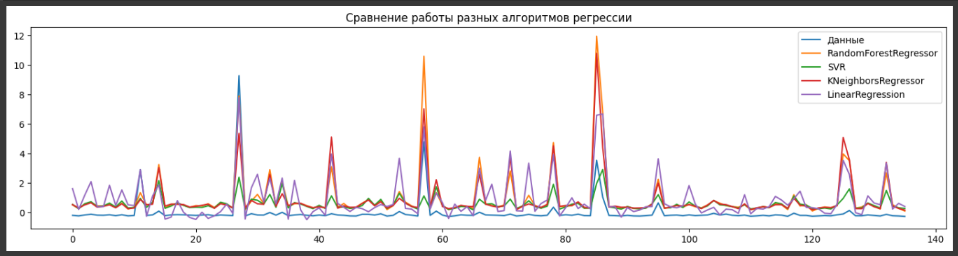


Рисунок Сравнение метрик разных алгоритмов



Обобщая итоги работы алгоритмов, можно сделать вывод, что ни один алгоритм не дал удовлетворительных результатов. Коэффициент детерминации порядка 0,3-0,36

## **Написать нейронную сеть, которая будет выдавать параметр шероховатости Ra**

Обучение нейронной сети — это такой процесс, при котором происходит подбор оптимальных параметров модели, с точки зрения минимизации ошибки. Начнём стоить нейронную сеть с помощью класса keras.Sequential.

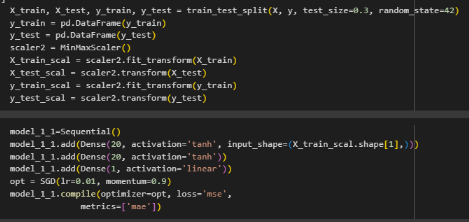


Рисунок Создание нейронной сети

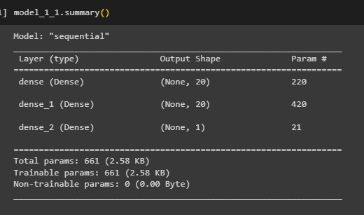


Рисунок Первая нейронная сеть

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

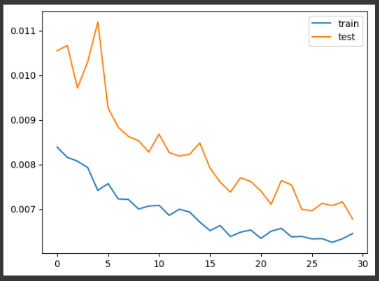


Рисунок График ошибки первой нейросети

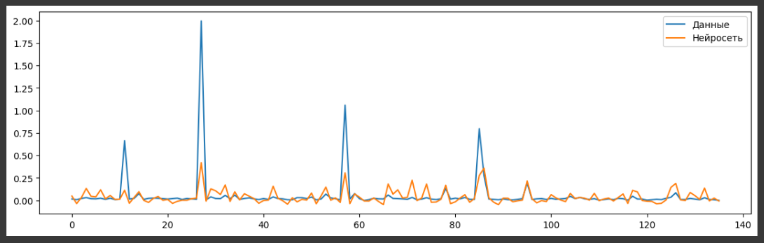


Рисунок Тестовые и прогнозные значения

## Разработка приложения

Приложение успешно работает и показывает результат прогноза для соотношения «матрица – наполнитель».

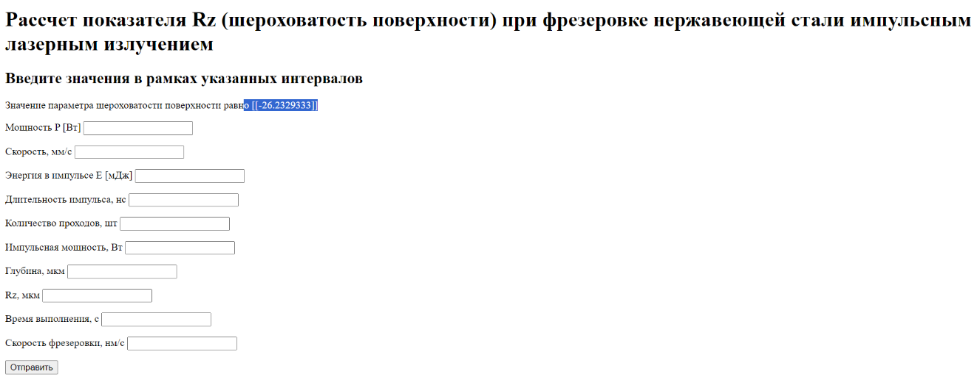


Рисунок Пример работы Приложения

Данное приложение — это основной файл Flask, папка templates, с шаблоном html - страницы, папка model c сохранённой моделью для данных.

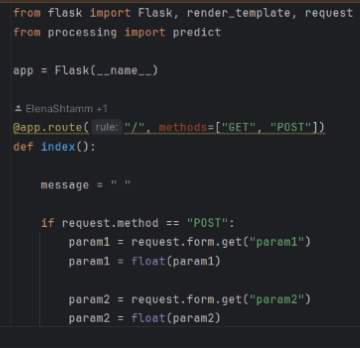


Рисунок Часть кода приложения

При запуске приложения, пользователь переходит на: [https://frezerovka.onrender.com](https://frezerovka.onrender.com/).

## Создание удаленного репозитория и загрузка результатов работы в него.

Репозиторий был создан на github.com по адресу: https://github.com/ElenaShtamm/SHTAMM\_ELENA\_VKR

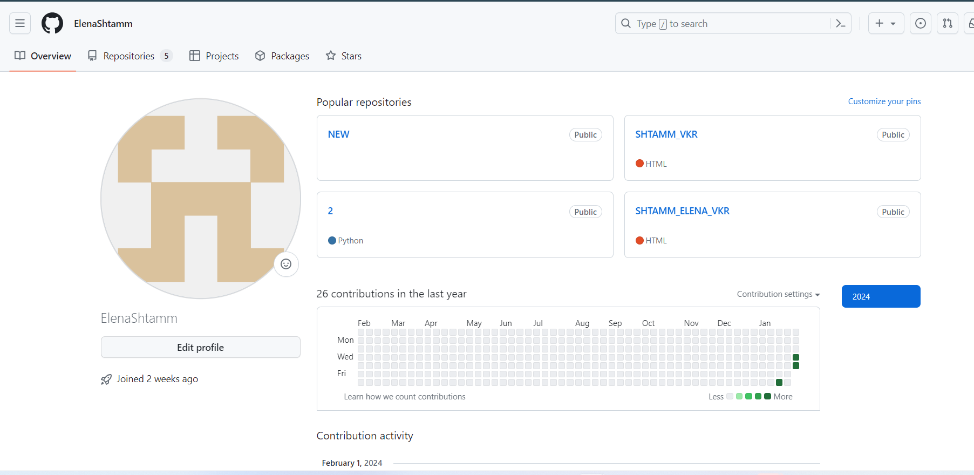


Рисунок Страница GitHUB

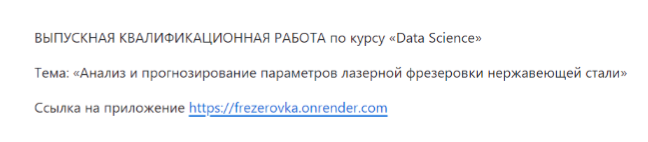


Рисунок файл README

# Заключение

Данная исследовательская работа позволяет сделать некоторые основные выводы по теме. Примененные модели регрессии показали неудовлетворительные результаты в предсказании параметра шероховатости поверхности после лазерной фрезеровки. Коэффициент детерминации ~0.35 говорит о том, что предсказательная способность модели ниже той, которая предсказывает всегда средние значения.

Был сделан вывод о том, что представленная выборка содержит слишком мало значений для эффективного прогнозирования, а также сильно смещена относительно средних значений. Необходимо дополнить базу результатами новых экспериментов, используя такие технологические режимы обработки, которые нивелируют смещение относительно текущих средних значений.

Поскольку процесс обработки материала лазерным излучением является физическим, то его можно описать формулами/уравнениями теплопроводности, поглощения/отражения излучения, распределением тепловых полей и тд. Таким образом эксперименты хорошо подвергаются математическому моделированию (например, в среде COMSOL Multiphysics), но оно плохо предсказывает конкретные физические величины, а лишь позволяет понять качественные зависимости между параметрами.

Исходя из вышесказанного, можно сделать вывод, что технологические параметры и результаты обработки должны хорошо прогнозироваться с помощью методов машинного обучения при увеличении выборки. Также, вероятно, стоит протестировать другие алгоритмы машинного обучения в конкретной задаче.

# Список используемой литературы и веб ресурсы

1. Alex Maszański. Метод k-ближайших соседей (k-nearest neighbour): – Режим доступа: <https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19>. (дата обращения: 07.06.2022)
2. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам сле-дует знать: – Режим доступа: https://habr.com/ru/company/vk/blog/513842/(дата обращения: 01.06.2022).
3. Devpractice Team. Python. Визуализация данных. Matplotlib. Seaborn. Mayavi. - devpractice.ru. 2020. - 412 с.: ил.
4. Бизли Д. Python. Подробный справочник: учебное пособие. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 864 с., ил.
5. Гафаров, Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Издательство Казанского университета, 2018. – 121 с.
6. Грас Д. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербурr, 2021. - 416 с.: ил.
7. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>.(дата обращения: 08.06.2022).
8. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>. (дата обращения: 10.06.2022)
9. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>. (дата обращения: 03.06.2022).
10. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>. (дата обращения: 04.06.2022).
11. Документация по библиотеке scikit-learn: – Режим доступа: <https://scikit-learn.org/stable/user_guide.html>. (дата обращения: 05.06.2022).
12. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>. (дата обращения: 06.06.2022).
13. Документация по библиотеке Tensorflow: – Режим доступа: <https://www.tensorflow.org/overview> (дата обращения: 10.06.2022).
14. Документация по языку программирования python: – Режим доступа:  <https://docs.python.org/3.8/index.html>. (дата обращения: 02.06.2022).
15. Краткий обзор алгоритма машинного обучения Метод Опорных Векторов (SVM) – Режим доступа: <https://habr.com/ru/post/428503/> (дата обращения 07.06.2022)
16. Миронов А.А. Машинное обучение часть I ст.9 – Режим доступа: <http://is.ifmo.ru/verification/machine-learning-mironov.pdf>. (дата обращения 08.06.2022)
17. Плас Дж. Вандер, Python для сложных задач: наука о данных и машинное обучение. Санкт-Петербург: Питер, 2018, 576 с.
18. Роббинс, Дженнифер. HTML5: карманный справочник, 5-е издание.: Пер. с англ. - М.: ООО «И.Д. Вильямс»: 2015. - 192 с.: ил.
19. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>. (дата обращения: 09.06.2022)
20. Траск Эндрю. Грокаем глубокое обучение. – СПб.: Питер, 2019. – 352 с.: ил.
21. //habr.com