# 

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 Бутакова Наталья Валерьевна

Екатеринбург, 2023

Содержание

[Введение 1](#_Toc133155923)

[1. Аналитическая часть 2](#_Toc133155924)

[1.1 Постановка задачи 2](#_Toc133155925)

[1.2 Описание используемых методов 3](#_Toc133155926)

[1.2.1 Линейная регрессия (Linear regression) 3](#_Toc133155927)

[1.2.2 Полиномиальная регрессия 4](#_Toc133155928)

[1.2.3 Лассо регрессия 5](#_Toc133155929)

Введение

Тема данной работы - прогнозирование конечных свойств новых материалов (композиционных материалов).

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).

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

* + - 1. Аналитическая часть
  1. Постановка задачи

Для исследовательской работы были даны 2 файла: X\_bp.xlsx (с данными о параметрах базальтопластика, состоящий из 1023 строк и 10 столбцов данных) и X\_nup.xlsx (данными углепластика, состоящий из 1040 строк и 3 столбцов данных). Для разработки моделей по прогнозу модуля упругости при растяжении, прочности при растяжении и соотношения матрица-наполнитель нужно объединить 2 файла. Объединение по типу INNER, поэтому часть информации (17 строк таблицы X\_nup.xlsx) не имеет соответствующих строк в таблице X\_bp.xlsx и будет удалена. Также необходимо провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы boxplot (ящик с усами), попарные графики рассеяния точек. Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; сделать предобработку: удалить шумы и выбросы, сделать нормализацию и стандартизацию. Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения «матрица-наполнитель». Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования. Оформить файл README.

* 1. Описание используемых методов

Данная задача в рамках классификации методов машинного обучения относится к машинному обучению с учителем, так как в предоставленном наборе данных имеются значения целевых параметров.

Так как перед нами стоит задача предсказания значений вещественной переменной — это задача регрессии.

В настоящее время разработано много методов регрессионного анализа. В данной работе были исследованы (и некоторые из них применены) следующие методы:

1. линейная регрессия (Linear regression);
2. полиномиальная регрессия (Polynomial regression);
3. лассо регрессия (Lasso regression);
4. эластичная сеть (Elastic Net);
5. обобщенная линейная модель с распределением Tweedie (GLM with a Tweedie distribution);
6. дерево решений (Decision Tree Regressor);
7. случайный лес (Random Forest);
8. К-ближайших соседей (KNeighbors Regressor);
9. градиентный бустинг (Gradient Boosting Regressor);
   * 1. Линейная регрессия

Простая линейная регрессия имеет место, если рассматривается зависимость между одной входной и одной выходной переменными. Для этого определяется уравнение регрессии (1) и строится соответствующая прямая, известная как линия регрессии.

(1)

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

Если ищется зависимость между несколькими входными и одной выходной переменными, то имеет место множественная линейная регрессия. Соответствующее уравнение имеет вид (2).

(2)

где n - число входных переменных.

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

Линейная регрессия — первый тщательно изученный метод регрессионного анализа. Его главное достоинство — простота. Такую модель можно построить и рассчитать даже без мощных вычислительных средств. Простота является и главным недостатком этого метода. Тем не менее, именно с линейной регрессии целесообразно начать подбор подходящей модели.

* + 1. Полиномиальная регрессия

Полиномиальная регрессия – это алгоритм машинного обучения, который используется для обучения линейной модели на нелинейных данных. Довольно часто данные намного сложнее, чем прямая линия, и в таких случаях обучение на основе алгоритма линейной регрессии не даст хороших результатов. Однако можно использовать алгоритм полиномиальной регрессии, чтобы добавить производительности каждой функции, а затем обучить линейную модель на расширенном наборе функций. Этот подход поддерживает в целом высокую производительность линейных методов, позволяя им соответствовать гораздо более широкому диапазону данных.

Например, простую линейную регрессию можно расширить, построив полиномиальные признаки из коэффициентов. В случае стандартной линейной регрессии у нас может быть модель, которая выглядит следующим образом (для двумерных данных):

 (3)

Если мы хотим подогнать к данным параболоид вместо плоскости, мы можем объединить признаки в полиномы второго порядка, чтобы модель выглядела так:

 (4)

Мы по-прежнему видим линейную модель, но набор признаков теперь такой:

 (5)

И мы можем представить модель в следующем виде:

 (6)

Аналогичным образом можно работать с полиномами любых порядков.

Мы видим, что результирующая полиномиальная регрессия принадлежит к тому же классу линейных моделей, который мы рассматривали выше (т. е. модель линейна по *w*) и может быть решена теми же методами.

* + 1. Лассо регрессия

Метод регрессии лассо (LASSO, Least Absolute Shrinkage and Selection Operator) — это вариация линейной регрессии, специально адаптированная для данных, которые имеют сильную корреляцию признаков друг с другом.

Лассо-регрессия использует сжатие коэффициентов (shrinkage) и этим пытается уменьшить сложность данных, искривляя пространство, на котором они лежат. В этом процессе лассо автоматически помогает устранить или исказить сильно коррелированные и избыточные функции в методе с низкой дисперсией.

Регрессия лассо использует регуляризацию L1, то есть взвешивает ошибки по их абсолютному значению.

Регуляризация позволяет интерпретировать модели. Если коэффициент стал 0, значит данный входной признак не является значимым.

* + 1. Эластичная сеть

Elastic-Net — это модель линейной регрессии с двумя регуляризаторами, L1 и L2. Эта комбинация позволяет изучать разреженную модель, в которой несколько весов не равны нулю, как у Лассо, при этом сохраняя свойства гребневой модели (Ridge). Модель Лассо и гребневая регрессия являются частными случаями Эластичной сети.

Эластичная сеть полезна, когда есть несколько признаков, которые коррелируют друг с другом. Лассо-регрессия, скорее всего, выберет один из них случайным образом, в то время как эластичная сеть, скорее всего, выберет оба.

* + 1. Обобщенная линейная модель с распределением Tweedie

Обобщённые линейные модели (Generalized Linear Models) – универсальный метод построения регрессионных моделей, позволяющий учитывать взаимодействие между факторами, вид распределения зависимой переменной и предположения о характере регрессионной зависимости.

Обобщенные линейные модели (GLM) расширяют линейные модели двумя способами.

Во-первых, прогнозируемые значения  связаны с линейной комбинацией входных переменных *Х* через функцию обратной связи *h* как:

 (7)

Во-вторых, квадратичная функции ошибки заменяется единичным отклонением распределения *d* в экспоненциальном семействе (точнее, моделью репродуктивной экспоненциальной дисперсии (EDM)).

 (8)

где  - штраф регуляризации L2.

Если указаны веса выборки, среднее значение становится средневзвешенным.

Ниже перечислены некоторые конкретные EDM и их единичное отклонение *d* (3я колонка):



Все они являются экземплярами семейства Tweedie)

* + 1. Дерево решений

Деревья решений (Decision Trees) - непараметрический метод, применяемый и для классификации, и для регрессии. Деревья решений используются в самых разных областях человеческой деятельности и представляют собой иерархические древовидные структуры, состоящие из правил вида «Если ..., то ...».

Решающие правила автоматически генерируются в процессе обучения на обучающем множестве путем обобщения обучающих примеров. Поэтому их называют индуктивными правилами, а сам процесс обучения — индукцией деревьев решений.

Дерево состоит из элементов двух типов: узлов (node) и листьев (leaf).

В узлах находятся решающие правила и производится проверка соответствия примеров этому правилу. В результате проверки множество примеров, попавших в узел, разбивается на два подмножества: удовлетворяющие правилу и не удовлетворяющие ему. Затем к каждому подмножеству вновь применяется правило и процедура рекурсивно повторяется пока не будет достигнуто некоторое условие остановки алгоритма. В последнем узле проверка и разбиение не производятся, и он объявляется листом.

В листе содержится не правило, а подмножество объектов, удовлетворяющих всем правилам ветви, которая заканчивается данным листом. Для классификации — это класс, ассоциируемый с узлом, а для регрессии — соответствующий листу интервал целевой переменной.

При формировании правила для разбиения в очередном узле дерева необходимо выбрать атрибут, по которому это будет сделано. Для регрессии критерием является дисперсия от среднего значения.

Огромное преимущество деревьев решений в том, что они легко интерпретируемы, понятны человеку. Они могут использоваться для извлечения правил на естественном языке. Еще преимущества — высокая точность работы, нетребовательность к подготовке данных.

Недостаток деревьев решений - склонность переобучаться. Переобучение в случае дерева решений - это ситуация, когда происходит точное распознавание примеров, участвующих в обучении, и полная несостоятельность на новых данных. В худшем случае, дерево будет большой глубины и сложной структуры, а в каждом листе будет только один объект. Для решения этой проблемы используют разные критерии остановки алгоритма.

* + 1. Случайный лес

Случайный лес (RandomForest) — представитель ансамблевых методов.

Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив. Формула итогового решателя (9) — это усреднение предсказаний отдельных деревьев.

 (9)

где

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

Для определения входных данных каждому дереву используется метод случайных подпространств. Базовые алгоритмы обучаются на различных подмножествах признаков, которые выделяются случайным образом.

Преимущества случайного леса:

* высокая точность предсказания;
* редко переобучается;
* практически не чувствителен к выбросам в данных;
* одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки, данные с большим числом признаков;
* высокая параллелизуемость и масштабируемость.

Из недостатков можно отметить, что его построение занимает больше времени. Так же теряется интерпретируемость.

* + 1. Метод К-случайных соседей

Метод К-ближайших соседей (k Nearest Neighbors) – это метод классификации, который адаптирован для регрессии. На интуитивном уровне суть метода проста: посмотри на соседей вокруг, какие из них преобладают, таковым ты и являешься.

В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Для реализации метода необходима метрика расстояния между объектами. Используется, например, эвклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных.

Этот метод — пример непараметрической регрессии.

* + 1. Градиентный бустинг

Градиентный бустинг (GradientBoosting) — еще один представитель ансамблевых методов.

В отличие от случайного леса, где каждый базовый алгоритм строится независимо от остальных, бустинг воплощает идею последовательного построения линейной комбинации алгоритмов. Каждый следующий алгоритм старается уменьшить ошибку предыдущего.

Чтобы построить алгоритм градиентного бустинга, нам необходимо выбрать базовый алгоритм и функцию ошибки (loss). Loss-функция – это мера, которая показывает насколько хорошо предсказание модели соответствуют данным. Используя градиентный спуск и обновляя предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых функция ошибки минимальна.

Бустинг, использующий деревья решений в качестве базовых алгоритмов, называется градиентным бустингом над решающими деревьями. Он отлично работает на выборках с «табличными», неоднородными данными и способен эффективно находить нелинейные зависимости в данных различной природы. На настоящий момент это один из самых эффективных алгоритмов машинного обучения. Благодаря этому он широко применяется во многих конкурсах и промышленных задачах. Он проигрывает только нейросетям на однородных данных (изображения, звук и т. д.).

Из недостатков алгоритма можно отметить только затраты времени на вычисления и необходимость грамотного подбора гиперпараметров.

* + 1. Нейронная сеть

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Структура нейронной сети пришла в мир программирования из биологии. Вычислительная единица нейронной сети — нейрон или персептрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа.

Смещение – это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения.

Так же у нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть. Примеры активационных функций: relu, сигмоида, гиперболический тангенс.

У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

* входной слой — его размер соответствует входным параметрам;
* скрытые слои — их количество и размерность определяем специалист;
* выходной слой — его размер соответствует выходным параметрам.

Прямое распространение – это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением.

Прогнозируемое значение сравниваем с фактическим с помощью функции потери. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась.

Для обновления весов в модели используются различные оптимизаторы.

Количество эпох показывает, сколько раз выполнялся проход для всех примеров обучения.

Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модели для прогнозирования Модуля упругости при растяжении | | | | |
| Название модели | Метрика R2 (коэффициент детерминации) | Средняя квадратичная ошибка MSE | Средняя абсолютная ошибка MAE | Точность  (1-MAE/y\_test.mean)\*100 |
| Линейная регрессия (sklearn) | -0,019 | 0.027 | 0.137 | 72.31 |
| Полиномиальная регрессия 2го порядка | -0,238 | 0.033 | 0.148 | 69.944 |
| Полиномиальная регрессия 3го порядка | -2.314 | 0.089 | 0.227 | 53.961 |
| Полиномиальная регрессия 4го порядка | -18.089 | 0.515 | 0.525 | -6.374 |
| Полиномиальная регрессия 6го порядка | -16.479 | 0.471 | 0.496 | -0.452 |
| Случайный лес | -0.048 | 0.028 | 0.137 | 72.242 |
| Метод К-ближайших соседей | -0.001 | 0.027 | 0.136 | 72.43 |
| Градиентный бустинг | -0.001 | 0.027 | 0.135 | 72.559 |
| Дерево решений | -0,131 | 0,03 | 0,141 | 41,404 |
| Лассо | -0,001 | 0,027 | 0.135 | 72,561 |
| Эластичная сеть | -0,001 | 0,027 | 0.135 | 72,561 |

Для 4х лучших моделей выполняем автоматический подбор гиперпараметров. Записываем для этих методов лучший результат (с самыми оптимальными из перебранных параметрами):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | R2 | MSE | MAE | Точность, % |
| Метод К-ближайших соседей  KNeighborsRegressor (n\_neighbors=106) | 0.001 | 0.027 | 0.136 | 72.424 |
| Градиентный бустинг  GradientBoostingRegressor (n\_estimators=30, learning\_rate=0.0001) | -0.001 | 0.027 | 0.135 | 72.559 |
| Лассо  Lasso(alpha=0.1) | -0,001 | 0,027 | 0.135 | 72,561 |
| Эластичная сеть  ElasticNet(alpha=0.1) | -0,001 | 0,027 | 0.135 | 72,561 |

Методы указаны со значимыми параметрами, значения которых отличаются от значений по умолчанию.

Сравнительная таблица моделей, полученная с помощью библиотеки lazypredict:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Adjusted  R-Squared | R-Squared | RMSE | Time Taken |
| DummyRegressor | -0.04 | -0.00 | 0.16 | 0.01 |
| ElasticNet | -0.04 | -0.00 | 0.16 | 0.03 |
| LassoLars | -0.04 | -0.00 | 0.16 | 0.04 |
| Lasso | -0.04 | -0.00 | 0.16 | 0.02 |
| BayesianRidge | -0.04 | -0.00 | 0.16 | 0.06 |
| QuantileRegressor | -0.04 | -0.00 | 0.16 | 13.52 |
| PoissonRegressor | -0.04 | -0.00 | 0.16 | 0.02 |
| TweedieRegressor | -0.05 | -0.00 | 0.16 | 0.02 |
| GammaRegressor | -0.05 | -0.01 | 0.16 | 0.08 |
| ElasticNetCV | -0.05 | -0.01 | 0.16 | 0.13 |
| LassoCV | -0.05 | -0.01 | 0.16 | 0.18 |
| LarsCV | -0.05 | -0.01 | 0.16 | 0.06 |
| LassoLarsCV | -0.05 | -0.01 | 0.16 | 0.05 |
| OrthogonalMatchingPursuit | -0.05 | -0.01 | 0.16 | 0.01 |
| LassoLarsIC | -0.05 | -0.01 | 0.17 | 0.07 |
| RidgeCV | -0.06 | -0.02 | 0.17 | 0.05 |
| Ridge | -0.06 | -0.02 | 0.17 | 0.06 |
| Lars | -0.06 | -0.02 | 0.17 | 0.05 |
| TransformedTargetRegressor | -0.06 | -0.02 | 0.17 | 0.02 |
| LinearRegression | -0.06 | -0.02 | 0.17 | 0.03 |
| OrthogonalMatchingPursuitCV | -0.06 | -0.02 | 0.17 | 0.03 |
| SGDRegressor | -0.07 | -0.02 | 0.17 | 0.03 |
| LinearSVR | -0.07 | -0.03 | 0.17 | 0.03 |
| HuberRegressor | -0.07 | -0.03 | 0.17 | 0.03 |
| AdaBoostRegressor | -0.08 | -0.04 | 0.17 | 0.31 |
| ExtraTreesRegressor | -0.10 | -0.06 | 0.17 | 0.43 |
| RandomForestRegressor | -0.14 | -0.10 | 0.17 | 1.33 |
| GradientBoostingRegressor | -0.21 | -0.16 | 0.18 | 0.47 |
| BaggingRegressor | -0.27 | -0.22 | 0.18 | 0.15 |
| HistGradientBoostingRegressor | -0.30 | -0.25 | 0.18 | 0.54 |
| LGBMRegressor | -0.34 | -0.29 | 0.19 | 0.12 |
| KNeighborsRegressor | -0.34 | -0.29 | 0.19 | 0.05 |
| SVR | -0.41 | -0.35 | 0.19 | 0.09 |
| XGBRegressor | -0.43 | -0.37 | 0.19 | 0.21 |
| NuSVR | -0.48 | -0.43 | 0.20 | 0.11 |
| MLPRegressor | -0.51 | -0.45 | 0.20 | 0.29 |
| PassiveAggressiveRegressor | -0.56 | -0.50 | 0.20 | 0.02 |
| ExtraTreeRegressor | -0.92 | -0.85 | 0.22 | 0.02 |
| DecisionTreeRegressor | -1.33 | -1.24 | 0.25 | 0.03 |
| RANSACRegressor | -1.73 | -1.62 | 0.27 | 0.17 |
| GaussianProcessRegressor | -3.75 | -3.56 | 0.35 | 0.14 |
| KernelRidge | -9.49 | -9.07 | 0.52 | 0.09 |

Переходим к прогнозу Прочности при растяжении.

Сравнительная таблица моделей, полученная с помощью библиотеки lazypredict:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Adjusted R-Squared | R-Squared | RMSE | Time Taken |
| TweedieRegressor | -0.04 | -0.00 | 0.17 | 0.03 |
| GammaRegressor | -0.04 | -0.00 | 0.17 | 0.12 |
| PoissonRegressor | -0.04 | -0.00 | 0.17 | 0.02 |
| BayesianRidge | -0.04 | -0.00 | 0.17 | 0.07 |
| LassoLarsIC | -0.04 | -0.00 | 0.17 | 0.05 |
| LassoLarsCV | -0.05 | -0.01 | 0.17 | 0.04 |
| LassoCV | -0.05 | -0.01 | 0.17 | 0.09 |
| DummyRegressor | -0.05 | -0.01 | 0.17 | 0.02 |
| ElasticNet | -0.05 | -0.01 | 0.17 | 0.03 |
| ElasticNetCV | -0.05 | -0.01 | 0.17 | 0.14 |
| LarsCV | -0.05 | -0.01 | 0.17 | 0.05 |
| Lasso | -0.05 | -0.01 | 0.17 | 0.01 |
| LassoLars | -0.05 | -0.01 | 0.17 | 0.01 |
| OrthogonalMatchingPursuit | -0.05 | -0.01 | 0.17 | 0.02 |
| RidgeCV | -0.05 | -0.01 | 0.17 | 0.04 |
| Ridge | -0.05 | -0.01 | 0.17 | 0.02 |
| Lars | -0.05 | -0.01 | 0.17 | 0.04 |
| LinearRegression | -0.05 | -0.01 | 0.17 | 0.02 |
| TransformedTargetRegressor | -0.05 | -0.01 | 0.17 | 0.03 |
| SGDRegressor | -0.05 | -0.01 | 0.17 | 0.05 |
| QuantileRegressor | -0.05 | -0.01 | 0.17 | 13.99 |
| OrthogonalMatchingPursuitCV | -0.05 | -0.01 | 0.17 | 0.03 |
| LinearSVR | -0.06 | -0.02 | 0.17 | 0.05 |
| HuberRegressor | -0.06 | -0.02 | 0.17 | 0.05 |
| AdaBoostRegressor | -0.07 | -0.03 | 0.17 | 0.36 |
| RandomForestRegressor | -0.09 | -0.05 | 0.17 | 1.67 |
| ExtraTreesRegressor | -0.09 | -0.05 | 0.17 | 0.44 |
| GradientBoostingRegressor | -0.19 | -0.14 | 0.18 | 0.48 |
| BaggingRegressor | -0.21 | -0.16 | 0.18 | 0.20 |
| KNeighborsRegressor | -0.21 | -0.16 | 0.18 | 0.03 |
| SVR | -0.21 | -0.16 | 0.18 | 0.12 |
| LGBMRegressor | -0.21 | -0.17 | 0.18 | 0.14 |
| HistGradientBoostingRegressor | -0.22 | -0.17 | 0.18 | 0.55 |
| NuSVR | -0.31 | -0.26 | 0.19 | 0.10 |
| XGBRegressor | -0.34 | -0.29 | 0.19 | 0.44 |
| MLPRegressor | -0.41 | -0.35 | 0.20 | 0.33 |
| PassiveAggressiveRegressor | -0.42 | -0.36 | 0.20 | 0.03 |
| RANSACRegressor | -0.64 | -0.57 | 0.21 | 0.20 |
| DecisionTreeRegressor | -1.02 | -0.94 | 0.24 | 0.03 |
| ExtraTreeRegressor | -1.29 | -1.20 | 0.25 | 0.03 |
| GaussianProcessRegressor | -3.82 | -3.63 | 0.36 | 0.20 |
| KernelRidge | -9.92 | -9.49 | 0.55 | 0.07 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модели прогнозирования Прочности при растяжении | | | | |
| Модель | R2 | MSE | MAE | Точность, % |
| Линейная регрессия | -0,008 | 0,029 | 0,136 | 73,686 |
| Полиномиальная регрессия 2го порядка | -0.062 | 0.029 | 0.137 | 72.339 |
| Полиномиальная регрессия 3го порядка | -1.859 | 0.077 | 0.216 | 56.299 |
| Полиномиальная регрессия 4го порядка | -26.868 | 0.751 | 0.552 | -11.795 |
| Метод К-ближайших соседей | 0.002 | 0.028 | 0.136 | 73.806 |
| Градиентный бустинг | -0.006 | 0.029 | 0.136 | 73.73 |
| Обобщенная линейная модель с нормальным распределением | -0.006 | 0.029 | 0.136 | 73.735 |
| Обобщенная линейная модель с распределением Пуассона | -0.006 | 0.029 | 0.136 | 73.732 |
| Обобщенная линейная модель с составным распределением Гамма-Пуассона | -0.006 | 0.029 | 0.136 | 73.734 |
| Обобщенная линейная модель с Гамма-распределением | -0.005 | 0.029 | 0.136 | 73.735 |
| Обобщенная линейная модель с обратным распределением Гаусса | -0.005 | 0.029 | 0.136 | 73.74 |

Для 3х лучших моделей выполняем автоматический подбор гиперпараметров. Записываем для этих методов лучший результат (с самыми оптимальными из перебранных параметрами):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | R2 | MSE | MAE | Точность, % |
| Метод К-ближайших соседей  KNeighborsRegressor (n\_neighbors=145) | 0.006 | 0.028 | 0.135 | 73.86 |
| Градиентный бустинг  GradientBoostingRegressor (learning\_rate=0.01, n\_estimators=3) | -0.006 | 0.029 | 0.136 | 73.73 |
| Обобщенная линейная модель TweedieRegressor(alpha=100, max\_iter=10, power=1, verbose=1) | -0.006 | 0.029 | 0.136 | 73.729 |

Методы указаны со значимыми параметрами, значения которых отличаются от значений по умолчанию.