ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет МИЭМ

Департамент прикладной математики

Отчёт по выполнению проекта
по дисциплине «Профориентационный семинар»
для направления 01.03.04 ПРИКЛАДНАЯ МАТЕМАТИКА

«Решение задач регрессии с помощью нейронных сетей»

Выполнил студент группы БПМ234
Нефедов Родион Игоревич

Преподаватель
Попов Виктор Юрьевич

Москва 2025г.

Введение

В ходе данного задания необходимо было:

- Изучить в чём заключаются задачи регрессии в контексте нейронных сетей
- Определить свою задачу регресии.
- Найти датасет, пригодный для обучения нейронной сети, и обучить нейросеть.
- Исследовать разные конфигурации архитектуры нейронной сети для выбора оптимального варианта в контексте выбранной задачи.

Все задания выполнялись в Jupyter Notebook с помощью окружений, созданных в Anaconda.

Содержание

1 Аннотация

В данном отчете рассматривается применение нейронных сетей для решения двух задач регрессии: прогнозирования прочности бетона на основе его состава и возраста, а также описания погодных условий на естественном языке на основе различных погодных параметров.

Первоначально требовалось определить задачи исследования и найти подходяще датасеты, что обеспечило основу для дальнейшей работы.

Первая исследовательская задача направлена на прогнозирование прочности бетона в зависимости от его состава и возраста. Для решения данной задачи использован многослойный персептрон (MLP), являющийся одной из разновидностей искусственных нейронных сетей. Данный метод позволяет эффективно моделировать сложные нелинейные зависимости в данных. В процессе работы использованы различные библиотеки Python, такие как pandas, numpy, matplotlib, seaborn и tensorflow, что позволило обрабатывать и визуадизировать данные, а также обучать и оценивать модели. Проведены эксперименты с архитектурой нейронной сети, включающие настройку гиперпараметров для достижения наилучших результатов. Предварительные итоги показывают, что многослойный персептрон успешно справляется с задачей прогнозирования прочности бетона, демонстрируя относительно высокую точность предсказаний.

Вторая исследовательская задача посвящена описанию погодных условий на естественном языке на основе различных параметров погоды. Для этой цели применена модель случайного леса (Random Forest), которая является мощным инструментом для задач классификации и регрессии. Использование данной модели позволяет учитывать множество факторов и их взаимодействий, что повышает качество прогнозов. Благодаря специфике выбранной модели, данные не требовали особой обработки, вроде нормализации и векторизации. Предварительные результаты показывают, что модель случайного леса успешно словесно описывает погоду, демонстрируя адекватность и точность описаний.

Таким образом, в данном отчёте демонстрируются успешные применения нейронных сетей для решения прикладных задач в области строительства и метеорологии. Полученные результаты подчеркивают важность правильной подготовки данных и выбора соответствующих моделей для достижения высокой точности предсказаний.

1.1 Принципы работы многослойного персептрона (MLP)

Многослойный персептрон (MLP) является разновидностью искусственной нейронной сети, предназначенной для моделирования сложных зависимостей в данных. Основные принципы работы MLP:

Архитектура сети:

- Входной слой: Получает исходные данные. Количество нейронов во входном

слое соответствует числу признаков в данных.

- Скрытые слои: Один или несколько слоев между входным и выходным слоями. Каждый нейрон в скрытом слое соединен со всеми нейронами предыдущего слоя. Количество нейронов и слоев выбирается эмпирически.

- Выходной слой: Представляет результат работы сети. Количество нейронов в выходном слое зависит от задачи (например, один нейрон для регрессии или несколько для многоклассовой классификации).

Активационные функции:

Используются для введения нелинейности в модель. Популярные функции активации включают ReLU (Rectified Linear Unit), сигмоидную функцию и гиперболический тангенс. Активационные функции применяются к выходам нейронов каждого слоя.

Функция активации ReLU:

$$f(x) = x^+ = max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if} x > 0, \\ 0 & \text{otherwise}, \end{cases}$$

где x это входные данные нейрона.

Компоненты обучения:

- Прямое распространение: Входные данные проходят через слои сети, и на выходе генерируется предсказание.

- Функция потерь: Оценивает расхождение между предсказанными и реальными значениями. В задачах регрессии часто используется среднеквадратичная ошибка.

- Обратное распространение: Процесс, в котором градиенты функции потерь вычисляются и распространяются назад по сети для обновления весов с использованием алгоритма градиентного спуска или его модификаций.

Обучение модели:

Модель обучается на тренировочных данных, постепенно обновляя веса сети. Процесс повторяется на протяжении множества эпох до достижения приемлемого уровня ошибки на тестовых данных.

1.2 Принципы работы модели случайного леса (Random Forest)

Случайный лес (Random Forest) представляет собой ансамблевый метод машинного обучения, который объединяет несколько деревьев решений для улучшения общей производительности и устойчивости модели. Основные принципы работы случайного леса включают:

Ансамблирование:

Бэггинг (Bootstrap Aggregating): Метод заключается в создании нескольких подвыборок исходного набора данных с повторением (bootstrap-выборки). Для каждой

подвыборки строится отдельное дерево решений.

Обучение деревьев: Каждое дерево обучается на своей bootstrap-выборке. При построении деревьев случайным образом выбирается подмножество признаков для каждого разбиения узлов, что добавляет дополнительную случайность и уменьшает корреляцию между деревьями.

Деревья могут быть построены до максимальной глубины или до тех пор, пока каждый лист не будет содержать минимальное количество образцов. В случайном лесе деревья часто строят до максимальной глубины.

Для задачи регрессии каждое дерево предсказывает числовое значение. Окончательный прогноз получается путем усреднения предсказаний всех деревьев.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

Где:

- $\hat{f}$ - прогноз случайного леса.
- B - количество раз, которое берётся случайная выборка данных.
- f_b - дерево, обученное на обучающей выборке X_b с ответами Y_b.
- x' - «невидимая» выборка.

Случайный лес уменьшает вероятность переобучения по сравнению с отдельными деревьями решений за счет усреднения. Оценка неопределенности прогноза может быть сделана как стандартное отклонение интерполяции от всех отдельных деревьев регрессии на x':

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}$$

Где:

- $\hat{f}$ - прогноз случайного леса.
- B - количество раз, которое берётся случайная выборка данных.
- f_b - дерево, обученное на обучающей выборке X_b с ответами Y_b.
- x' - «невидимая» выборка.
- σ - стандартное отклонение интерполяции от всех отдельных деревьев регрессии

2 Начало работы

Было решено проводить работу в Jupyter Notebook, так как это позволяет работать оффлайн и сохранять результаты работы локально.

Для обучения нейронных сетей использовалась библиотека Tensorflow версии 2.10.1, что позволяло использовать GPU для ускорения обучения.

3 Определение задачи и поиск датасета

Задача регрессии была определена так: на вход подаётся ряд некоторых параметров, которые после обработки некоторой функцией передают на выход целевое значение.

Под условие задачи в социальной сети Kaggle было найденно несколько датасетов, из который были выбраны:

- Данные о прочности бетона, его составе и возрасте.
- Данные о погоде в Сиэтле с небольшим количеством параметров и описанием погоды естественным языком.

4 Прогноз прочности бетона в зависимости от его состава и возраста

Прочность бетона измерялась в давлениии (МПа), которое способен выдержать бетон на сжатие.

4.1 Использованные библиотеки Python

Выбранные библиотеки можно разделить на две части:

- Необходимые для обработки датасета и визуализации результатов
    1. numpy
    2. pandas
    3. matplotlib
    4. seaborn
- Необходимые непосредственно для работы с нейронной сетью
    1. tensorflow

4.2 Обработка датасета

Изначально датасет был представлен в виде *csv* файла

cement,slag,flyash,water,superplasticizer,coarseaggregate,fineaggregate,age,csMPa
540,0,0,162,2.5,1040,676,28,79.99
540,0,0,162,2.5,1055,676,28,61.89
332.5,142.5,0,228,0,932,594,270,40.27
332.5,142.5,0,228,0,932,594,365,41.05
198.6,132.4,0,192,0,978.4,825.5,360,44.3
266,114,0,228,0,932,670,90,47.03
380,95,0,228,0,932,594,365,43.7
380,95,0,228,0,932,594,28,36.45
266,114,0,228,0,932,670,28,45.85

Рис. 1: *csv* файл с данными. Видно, что разделителем колонок является запятая

Для его преобразования в пригодный для использования вид использовалась библиотека pandas.

	cement	slag	flyash	water	superplasticizer	coarseaggregate	fineaggregate	age	csMPa
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28	44.28
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28	31.18
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28	23.70
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28	32.77
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.40

Рис. 2: Датасет, преобразованный в dataframe с помощью pandas

Затем данные были разделены на тренировочную и тестовую выборки в соотношении 4:1, а также была задана функция нормализации.

4.3 Использование нейронной сети для предсказания прочности бетона

Для выполнения данной задачи была выбрана нейронная сеть являющаяся многослойным персептроном (MLP) с двумя скрытыми слоями, использующими функцию активации ReLU, и одним выходным слоем, предназначенным для регрессионных задач.

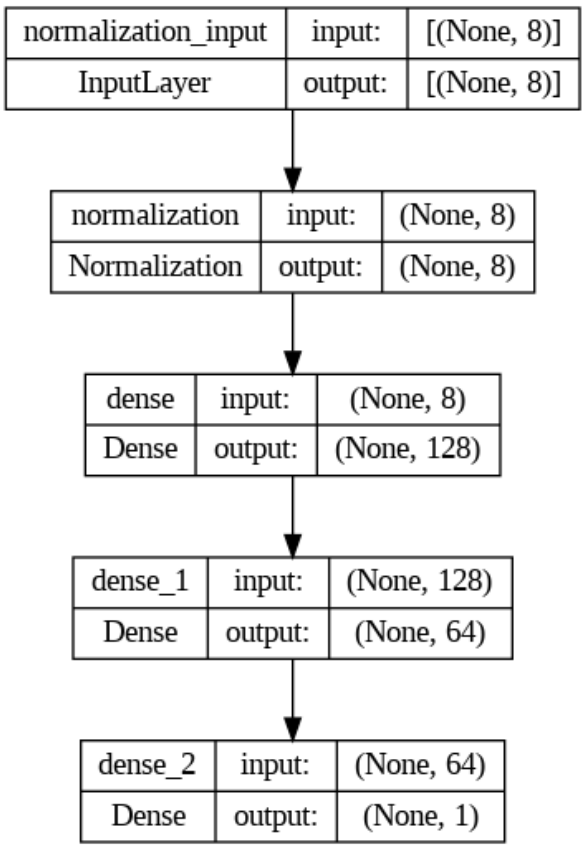

Рис. 3: Схема архитектуры используемой нейронной сети

4.4 Эксперимент с архитектурой нейросети

После первого обучения нейросети был поставлен вопрос поиска её оптимальной архитектуры. Для этого был создан цикл, в каждой итерации которого обучалась нейронная сеть с немного другими параметрами:

- Количество нейронов первого слоя (от 16 до 128 включительно).
- Количество нейронов второго слоя (от 16 до 128 включительно).
- Число эпох обучения (от 100 до 1000 включительно).

После каждой итерации в заранее созданный массив сохранялись такие результаты как: время обучения и средняя ошибка по абсолютному значению.

В рамках исследования было проанализировано 640 архитектур нейронных сетей, на что было затрачено приблизительно 5 часов. Полученные данные были обработанны и исследованны.

Результаты представлены ниже.

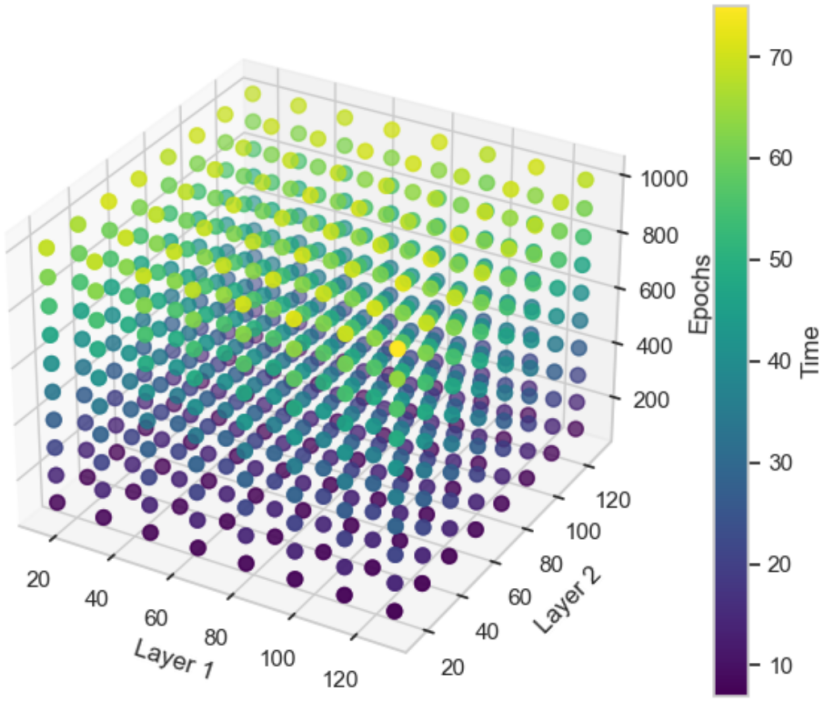

Рис. 4: Зависимость времени обучения от количиства нейронов на 1-ом и 2-ом слоях нейронной сети и числа эпох.

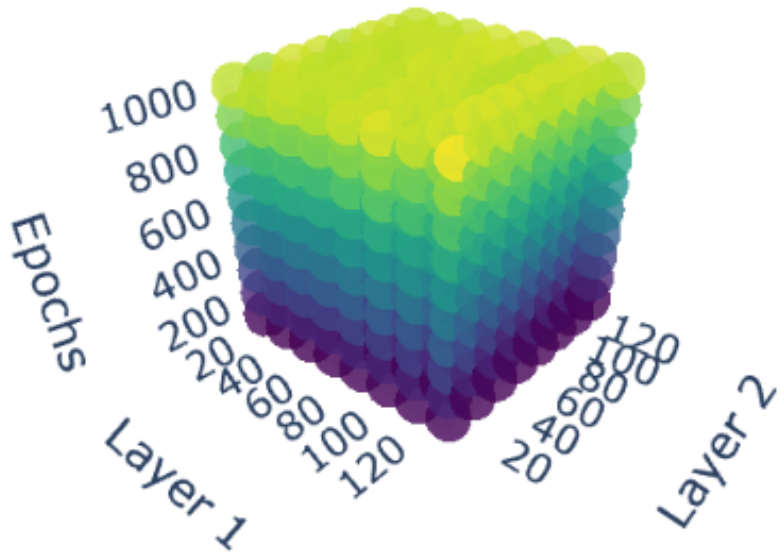

Рис. 5: Зависимость времени обучения от количиства нейронов на 1-ом и 2-ом слоях нейронной сети и числа эпох.

Исходя из анализа данных графиков, можно заключить, что влияние количества нейронов на продолжительность обучения нейронной сети носит заметно меньший характер по сравнению с воздействием числа эпох обучения.

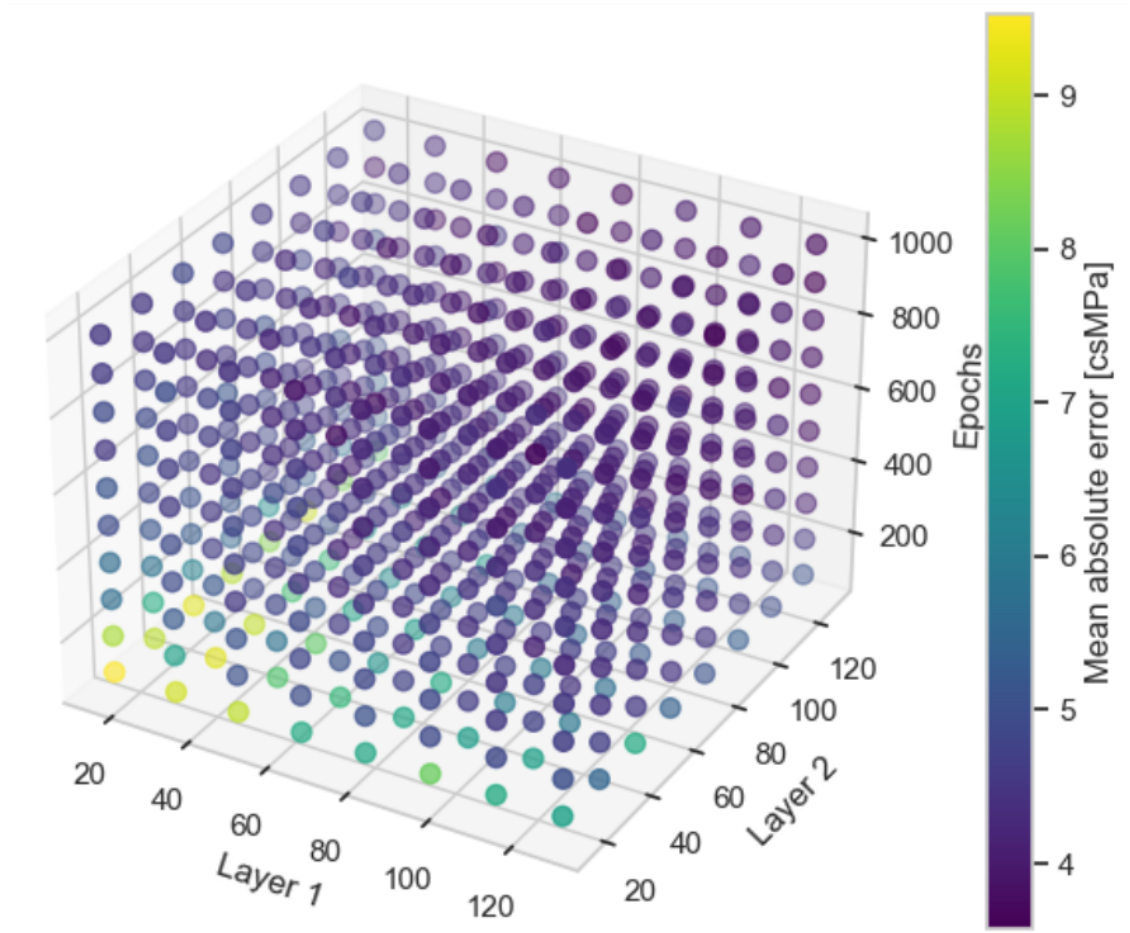

Рис. 6: Зависимость средней абсолютной ошибки от количиства нейронов на 1-ом и 2-ом слоях нейронной сети и числа эпох.

Исходя из анализа данного графика, можно заключить, что влияние количества нейронов на среднюю абсолютню ошибку нейронной сети носит заметный характер наряду с воздействием числа эпох обучения. Так, моделям с бо̂льшим количеством

нейронов на первом слое требовалось гораздо меньше эпох для достижения результата аналогичного моделям с меньшим количеством нейронов. Похожая ситуация вознкиает и с нейронами второго слоя, пусть и менее выраженная. Также можно заметить, что с какого-то момента увеличение числа эпох перестаёт давать прирост к точности результатов нейросети.

Согласно результатам исследования лучший результат показала нейросеть с такими характеристиками:

- Количество нейронов первого слоя: 128.
- Количество нейронов второго слоя: 128.
- Число эпох обучения: 900.

Со средней абсолютной ошибкой: 3.576234, что составляет *9.98%* от среднего значения целевого параметра.

5 Описание погоды естественным языком на основе её параметров

Для выполнения этой задачи я решил использовать модель случайного леса, поскольку она позволяет обрабатывать данные в текстовом формате без необходимости их предварительной векторизации.

5.1 Использованные библиотеки Python

Выбранные библиотеки можно разделить на две части:

- Необходимые для обработки датасета и визуализации результатов
    1. numpy
    2. pandas
- Необходимые непосредственно для работы с нейронной сетью
    1. tensorflow_decision_forests
    2. tensorflow
    3. tf_keras
    4. math

5.2 Обработка датасета

Датасет был разделён на тренировочную и тестовую части, а также была выбрана категориальная метка для кодирования в виде целых чисел.

5.3 Использование нейронной сети для предсказания погоды

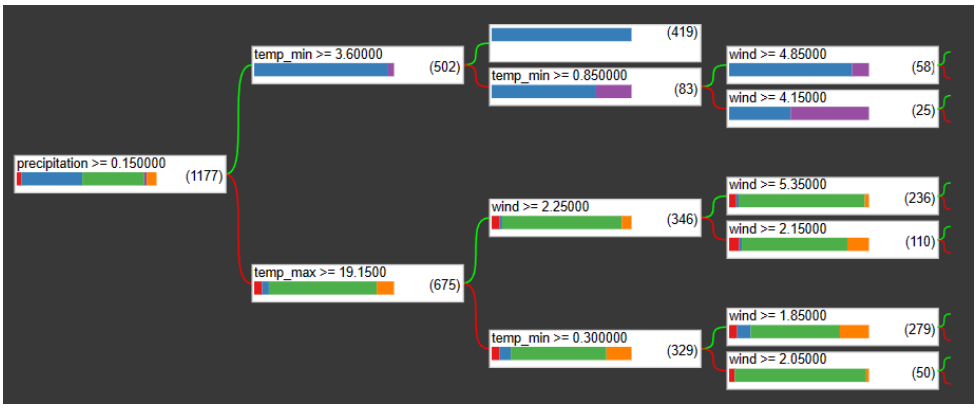

Рис. 7: Дерево решений используемой нейронной сети

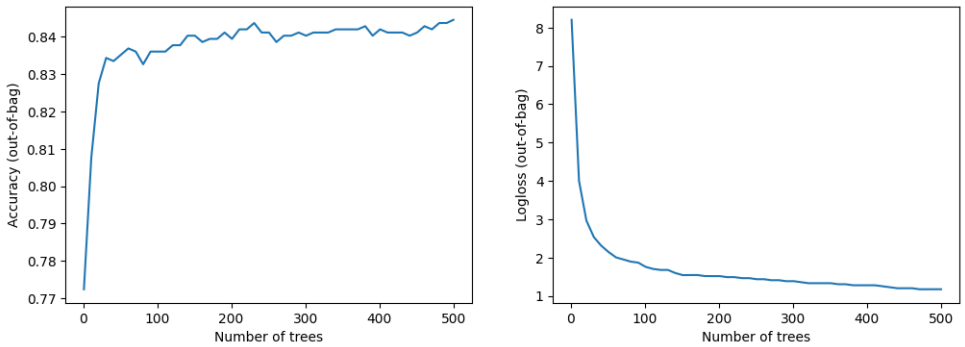

Рис. 8: Зависимость точности и логарифмической потери от количества деревьев)

При увеличении количества деревьев более 400 точность нейросети не менятеся.

Точность достигла: 0.8415

Проверим нейросеть на свежих реальных данных:

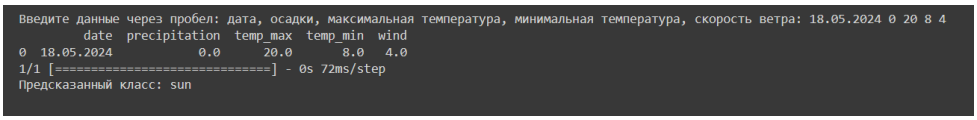

Рис. 9: Проверка нейросети на реальных данных)

И дейсвительно в этот день была солнечнная погода. Можно сделать вывод, что нейросеть хорошо справляется с реальными задачами.

Использованные материалы

[1] Документация Pandas ссылка на сайт

[2] Обучающие материалы Tensorflow для задач регрессии. ссылка на статью

[3] Документация Tensorflow для «моделей леса решений» ссылка на сайт

[4] Урок по созданию трёхмерных графиков в python с помощью matplotlib ссылка на сайт

[5] Документация plotly ссылка на сайт

[6] Статья про многослойный персептрон на Wikipedia ссылка на статью

[7] Статья про «Random Forest» на Wikipedia ссылка на статью

[8] ChatGPT ссылка на сайт