Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа информационных технологий и робототехники

Отделение автоматизации и робототехники

Направление подготовки 15.04.06 Мехатроника и робототехника

ОТЧЁТ

по лабораторной работе №1

«Методы искусственного интеллекта. EDA. Линейная регрессия. Дерево решений. CatBoost. XGBoost. Нейронные сети (MLP)»

по дисциплине Методы искусственного интеллекта в мехатронике и робототехнике

Вариант - 7

Выполнила: студентка гр. 8ЕМ42 _____ _____ Игнатьева А.В.

Подпись Дата Фамилия И.О.

Проверил: ассистент ОАР _____ _____ Куренко В.А.

Подпись Дата Фамилия И.О.

Томск – 2025 г.

Цель работы: получение навыков анализа первичных данных и определение признаков взаимосвязи (EDA), понимания моделей: линейная регрессия, дерево решений, CatBoost, XGBoost, нейронные сети (MLP) и умения разрабатывать программу на языке Python для реализации представленных моделей.

Задание

1) Описать датасет, определить влияние признаков и выбрать признаки, которые наиболее подходят для поставленной задачи предсказания (EDA).
2) Построить пайплайн (DVC) исходя из результатов EDA.
3) Реализовать линейную регрессию, определить весы, метрики и ошибки.
4) Реализовать дерево решений, определить метрики и ошибки. Привести рисунок первых узлов дерева решений.
5) Реализовать CatBoost, определить метрики и ошибки. Выгрузить Feature Importance.
6) Реализовать XGBoost, определить метрики и ошибки. Выгрузить Feature Importance.
7) Реализовать нейронную сеть, определить метрики, ошибки, кривые обучения, гистограммы весов с интерпретацией и график из Tensorboard.
8) Выгрузить конечный вычислительный граф DVC.
9) Построить сводную таблицу с метриками и сделать вывод какая модель отработала лучше и почему.

Основная часть

1. Описание датасета

Набор данных, который был выбран для применения методов искусственного интеллекта в данной лабораторной работе, содержит информацию о погодных условиях, зарегистрированных каждый день с 1940 по 1945 год на различных метеостанциях по всему миру. Информация

включает осадки, снегопады, температуру, скорость ветра и то, были ли в этот день грозы или другие плохие погодные условия.

Выбранный датасет содержит два файла, включающими следующие признаки.

Summary of Weather.csv

- STA: метеорологическая станция;
- Date: дата;
- Precip: осадки в мм;
- WindGustSpd: пиковая скорость порыва ветра в км/ч;
- MaxTemp: Максимальная температура в градусах Цельсия;
- MinTemp: минимальная температура в градусах Цельсия;
- MeanTemp: средняя температура в градусах Цельсия;
- Snowfall: Количество выпавшего снега и ледяных гранул в мм;
- PoorWeather: Повторение колонки TSHDSBRSGF;
- YR: Год наблюдения;
- MO: месяц наблюдения;
- DA: день наблюдения;
- PRCP: Осадки в дюймах и сотых долях;
- DR: Направление пикового порыва ветра в десятках градусов;
- SPD: Пиковая скорость порыва ветра в узлах;
- MAX: Максимальная температура в градусах по Фаренгейту;
- MIN: минимальная температура в градусах по Фаренгейту;
- MEA: средняя температура в градусах по Фаренгейту;
- SNF: Количество выпавшего снега в дюймах и десятых;
- SND: Высота снежного покрова (включая ледяные гранулы), зарегистрированная в 1200 GMT кроме 0000 GMT в районе Дальней Восточной Азии в дюймах и десятых долях;
- FT: Верхняя часть промерзшего грунта (глубина в дюймах);
- FB: Замерзшее основание (глубина в дюймах);

- FTI: Толщина промерзшего грунта (толщина в дюймах);
- ITH: Толщина льда на воде (дюймы и десятые);
- PGT: Время пикового порыва ветра (часы и десятые);
- TSHDSBRSGF: День с: грозой (Thunder); мокрым снегом (Sleet); градом (Hail); пылью или песком (Dust or Sand); смогом (Smoke or Haze); метелью (Blowing Snow); дождем (Rain); снегом (Snow); гололедицей (Glaze); туманом (Fog); 0 = нет, 1 = да;
- SD3: Высота снежного покрова на 0030 GMT включает гранулы льда в дюймах и десятых долях;
- RHX: 24-часовая максимальная относительная влажность, в целых процентах;
- RHN: Минимальная относительная влажность воздуха за 24 часа, в целых процентах;
- RVG: Речная шкала в футах и десятых;
- WTE: Водный эквивалент снега и льда на земле в дюймах и сотых долях.

Summary of Weather.csv

- WBAN: Номер метеорологической станции;
- NAME: Название метеостанции;
- STATE/COUNTRY ID: Местонахождение;
- LAT: Широта как строка;
- LON: Долгота как строка;
- ELEV: высота, обратите внимание, что значение высоты 9999 означает неизвестно;
- Latitude: Широта как числовая переменная;
- Longitude: Долгота как числовая переменная.

На основе данного датасета требуется предсказать среднюю температуру (без использования минимальной и максимальной температуры напрямую) с применением методов искусственного интеллекта.

2. *Анализ данных*

Для выявления признаков, которые в дальнейшем будут использованы для предсказания средней температуры, требуется провести EDA (Exploratory Data Analysis). Основные цели EDA – это понять структуру данных, определить типы переменных, их распределение и взаимосвязи между ними.

Для начала было решено посмотреть на разнообразие стран и количество метеостанций в них (рисунок 1).

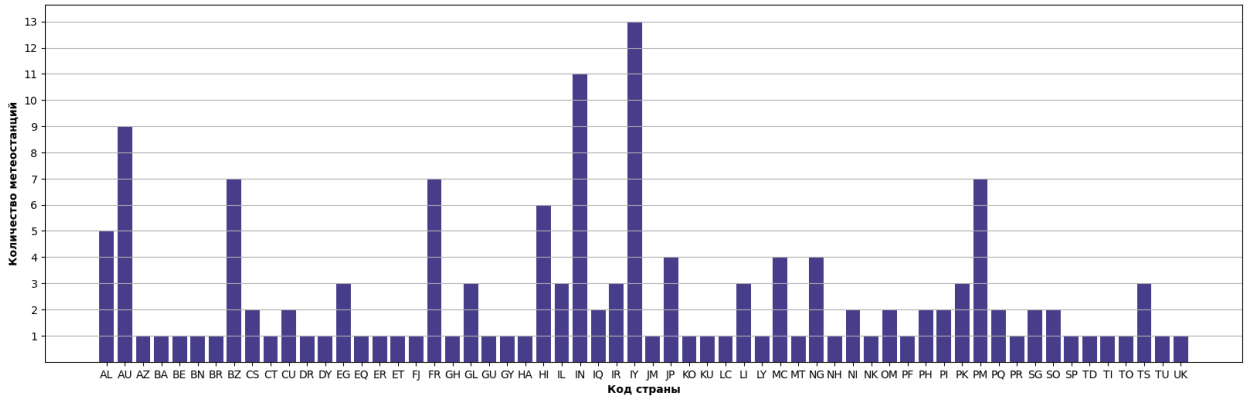

Рисунок 1 – Распределение метеостанций по странам

Далее для наглядности географического расположения метеорологических станций был получен график с долготой по оси абсцисс и широтой по оси ординат (рисунок 2).

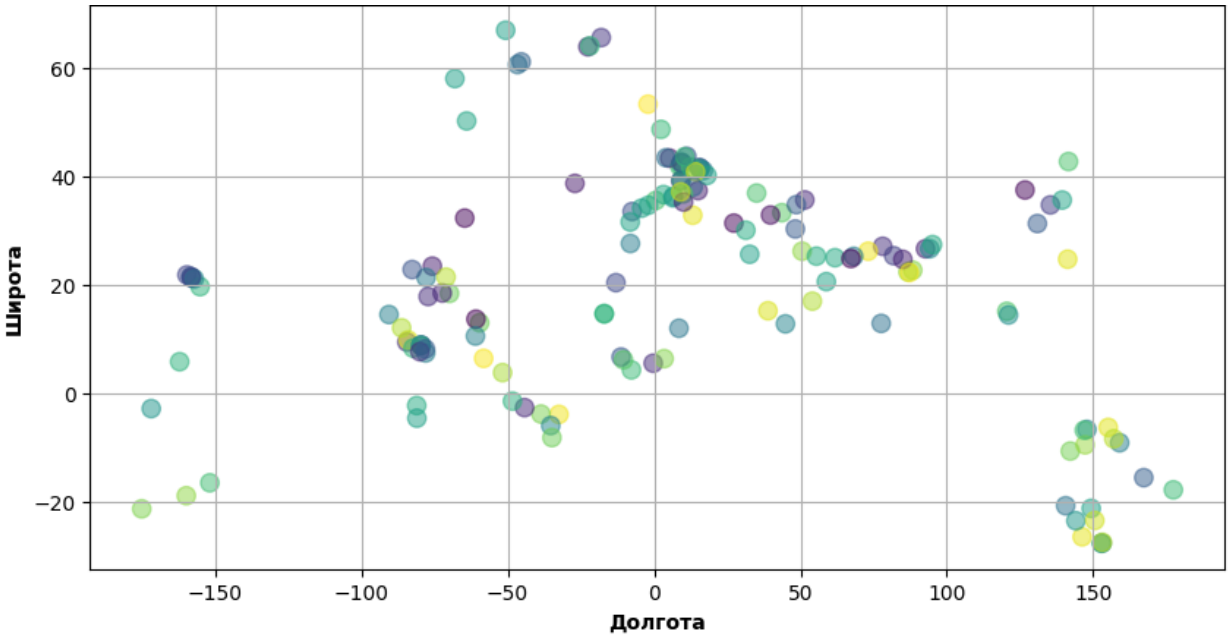

Рисунок 2 – Географическое расположение метеостанций

Затем были рассмотрены зависимости средней температуры от долготы, широты и высоты географического расположения метеостанции (рисунки 3-5).

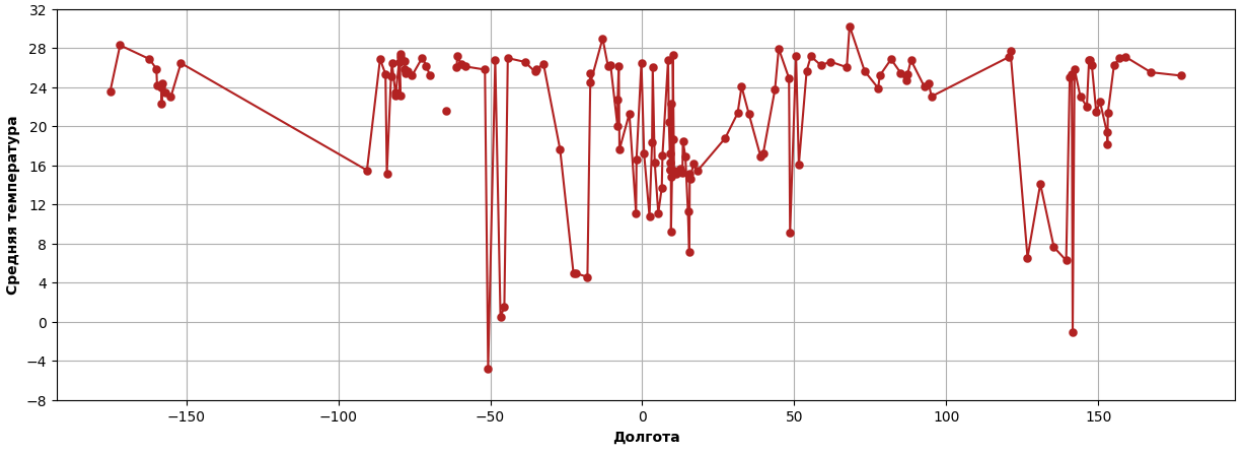

Рисунок 3 – Зависимость средней температуры от долготы географического расположения метеостанции

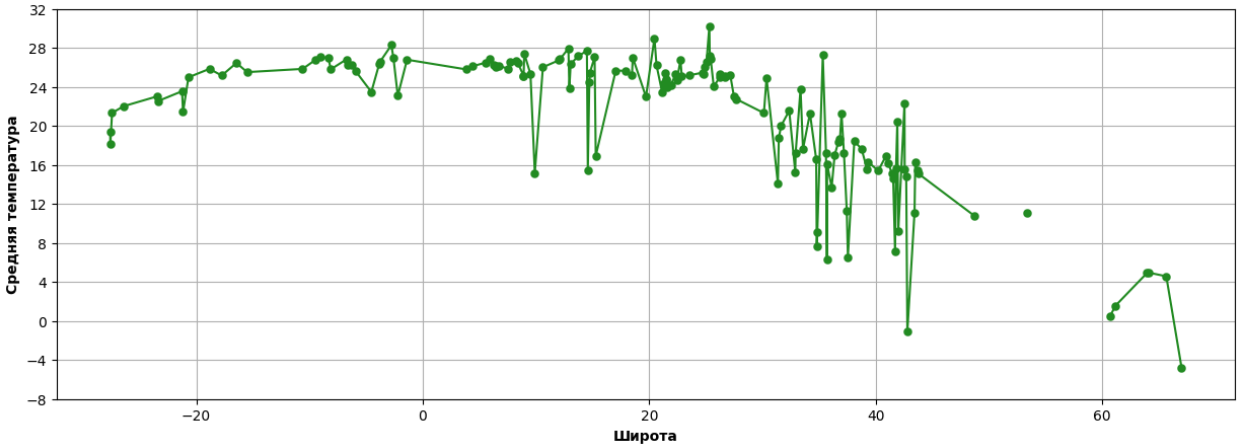

Рисунок 4 – Зависимость средней температуры от широты географического расположения метеостанции

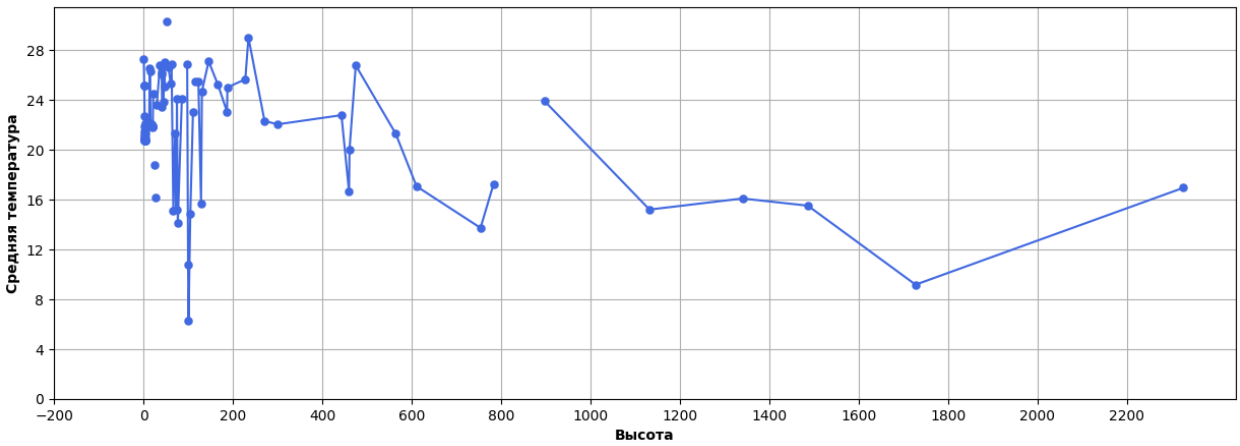

Рисунок 5 – Зависимость средней температуры от высоты географического расположения метеостанции

В результате построения графиков была выявлена закономерность в изменении средней температуры в зависимости от широты. При анализе зависимости от других двух параметров географического расположения закономерности, которая может пригодится для предсказания средней температуры, обнаружено не было.

Далее были рассмотрены даты, в которые были зафиксированы погодные условия, а именно год и месяц. Построены графики зависимости (рисунки 6-7).

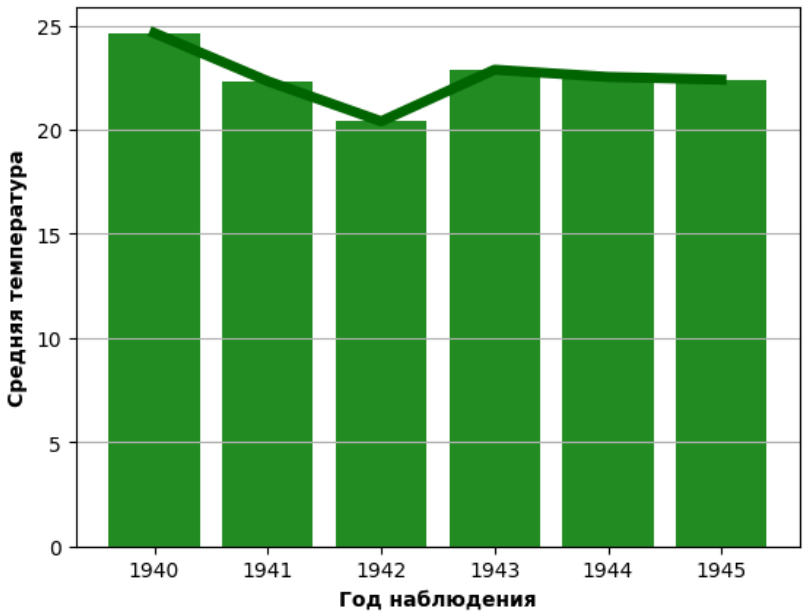

Рисунок 6 – График зависимости средней температуры за год от года наблюдения

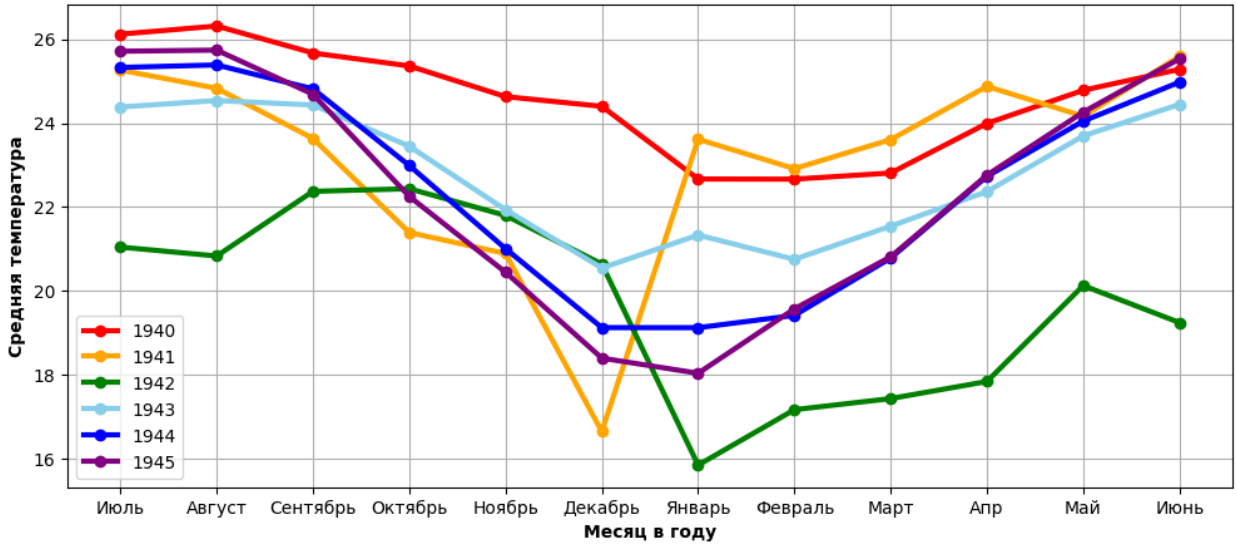

Рисунок 7 – График зависимости средней температуры за месяц от месяца наблюдения

Таким образом, логическая закономерность наблюдается при изменении средней температуры в зависимости от месяца: как правило, более холодные месяцы – зимние, более жаркие – летние. Построение графика зависимости температуры от года наблюдения для выявления закономерности результата не дало.

Для упрощения поиска зависимости средней температуры от погодных условий была построена матрица корреляции (рисунок 8).

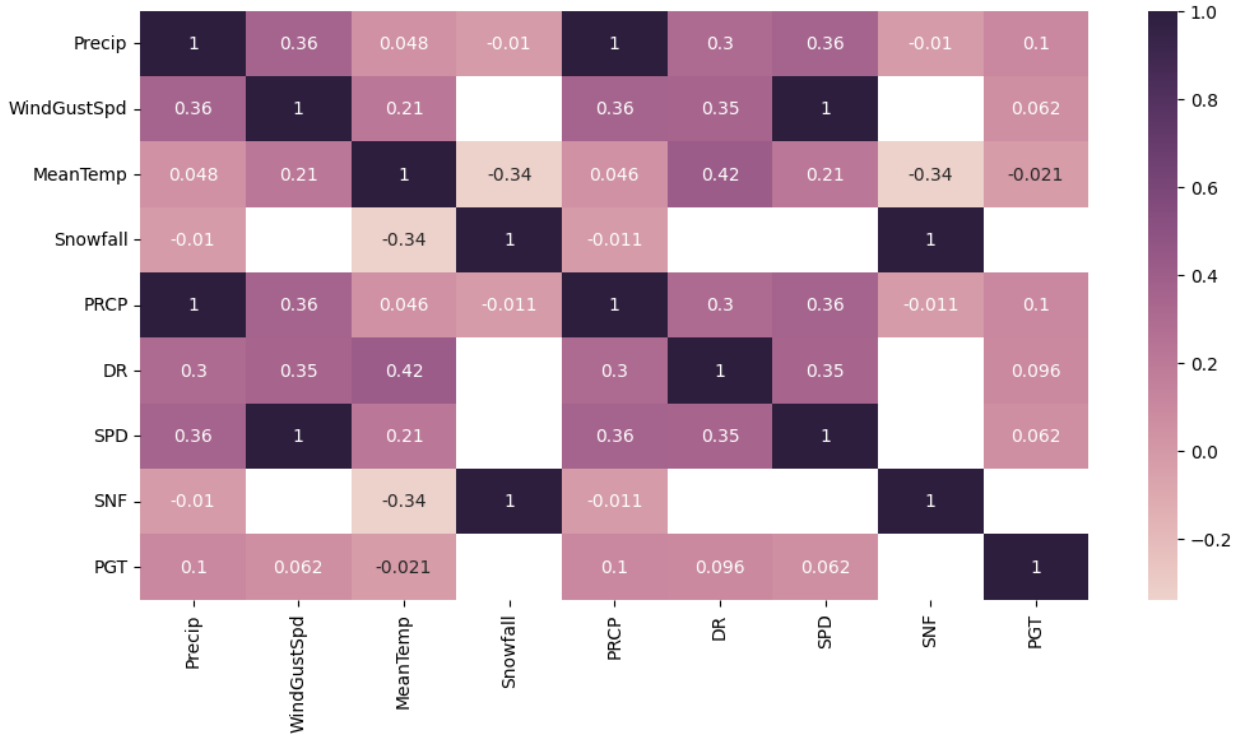

Рисунок 8 – Матрица корреляции погодных условий

При анализе матрицы значения корреляции средней температуры и различных погодных условий не превысили даже 0,5, что указывает на отсутствие явных закономерностей.

Таким образом, в результате проведения EDA для предсказания средней температуры были выбраны следующие признаки: Latitude (широта географического расположения метеостанции) и MO (месяц наблюдения средней температуры).

3. Построение пайплайна DVC

После проведения EDA была создана последовательная система обработки данных и обучения моделей – пайплайн DVC. Этот процесс основан на результатах EDA, где были выделены ключевые признаки, влияющие на среднюю температуру.

DVC (Data Version Control) – это инструмент, который создан для управления версиями моделей и данных в ML-проектах. Помимо контроля версий есть возможность создавать пайплайны – цепочки вычислений с зависимостями. При построении пайплайна каждая стадия обработки

оформляется в виде отдельного шага, зарегистрированного в DVC. Этот инструмент упрощает проведение экспериментов, так как помогает упорядочить потоки данных и шаги обучения.

Построение пайплайна начинается с настройки обработки входных данных из датасета. Был создан скрипт для подготовки данных к машинному обучению: из исходного датасета были выделены признаки, выбранные в ходе анализа, категориальные данные переведены в числовой формат. После создания скрипта его запуск был добавлен в виде первого шага пайплайна DVC. Затем аналогичным образом были добавлены шаги обучения различных моделей.

4. Линейная регрессия

Первой моделью машинного обучения для решения задачи регрессии – прогноза средней температуры на основе выборки объектов с различными признаками – стала линейная регрессия. Данная модель машинного обучения на основе данных строит регрессионную прямую методом наименьших квадратов.

Для реализации была использована библиотека scikit-learn. Данные были заранее разделены на тренировочную и тестовую выборки для обучения и дальнейшей оценки обученной модели. В качестве бейзлайна – исходной модели, с которой проводится сравнение при оценке обучения – было выбрано нормальное распределение.

Метрики и коэффициенты линейной регрессии вынесены в таблицу 1. Веса (коэффициенты) дополнительно представлены в виде графика (рисунок 9).

Сравнив среднюю абсолютную ошибку (MAE) модели линейной регрессии с ошибкой нормального распределения, был сделан вывод, что модель после обучения показывает сравнительно лучший результат в прогнозировании средней температуры. Однако величина коэффициента детерминации (R^2) говорит о низкой предсказательной способности модели.

Таблица 1 – Результаты обучения модели линейной регрессии

Модель	***random.normal***	***LinearRegression***
Описание	Нормальное распределение	Линейная регрессия методом наименьших квадратов
Метрики	MAE: 9,149700380292549	MAE: 4,90980670806521 R^2: 0,3191816352444038
Коэффициенты	–	1: 0,136869 2: -0,231829 intercept: 25,486328

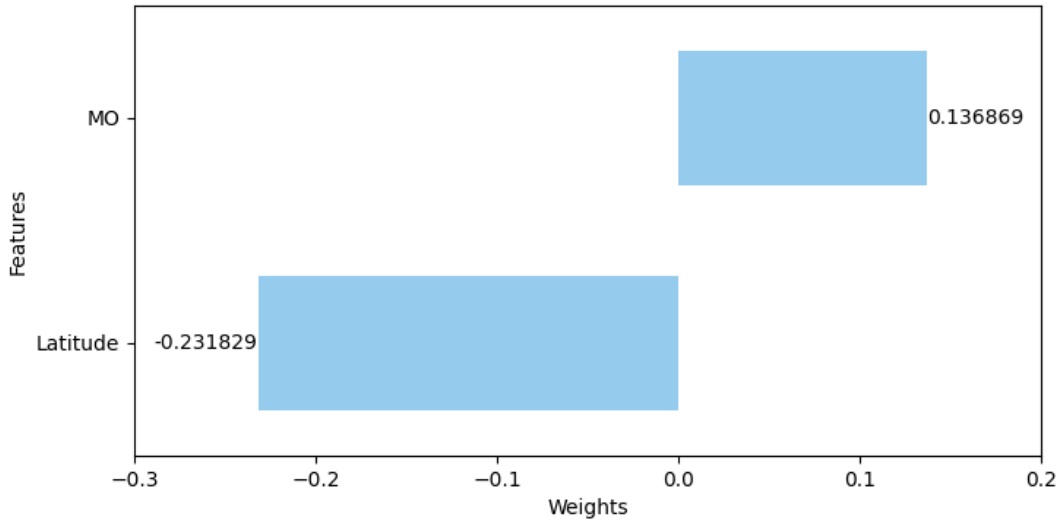

Рисунок 9 – График распределения весов

5. Дерево решений

Следующая модель, которая была использована для прогнозирования средней температуры – это дерево решений. Оно представляет собой древовидную структуру, состоящую из узлов и ветвей. В каждом узле происходит разделение данных на две или более группы на основе определённого критерия. Ветви представляют собой возможные исходы или результаты разделения.

Для реализации была использована аналогичная библиотека scikit-learn и подготовленные тренировочная и тестовая выборки. В качестве бейзлайна была использована модель линейной регрессии, обученная в предыдущем

шаге. Так как деревья решений имеют ряд гиперпараметров, было решено использовать инструмент GridSearchCV для автоматического перебора гиперпараметров с выбором лучших из них.

После обучения модели был получен рисунок первых узлов дерева решений, что позволило наглядно понаблюдать за процессом обучения (рисунок 10).

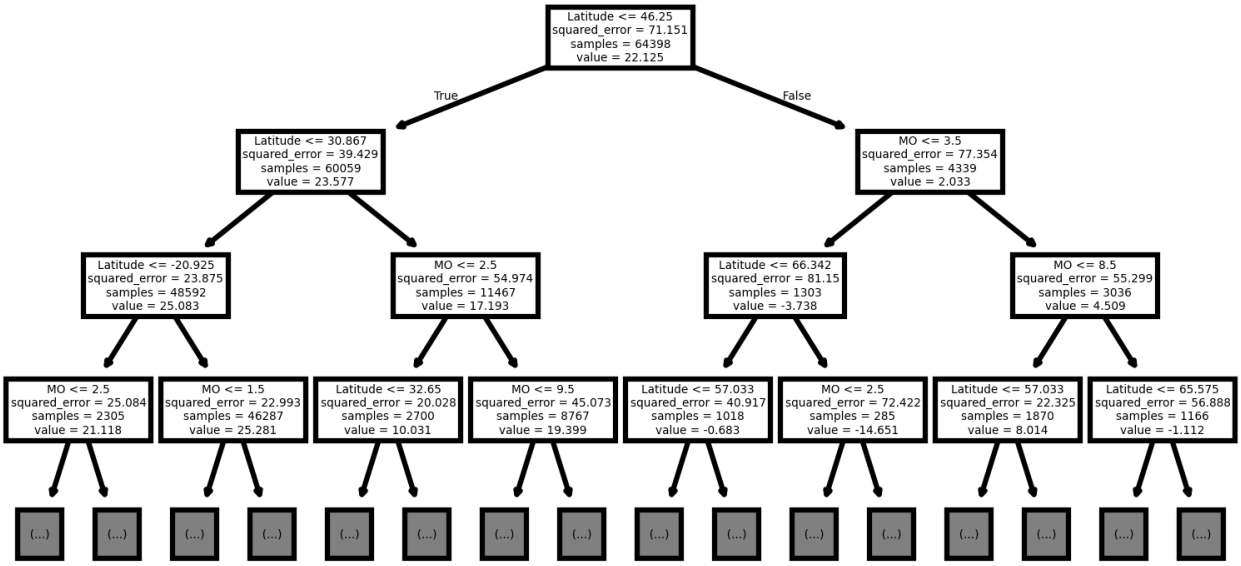

Рисунок 10 – Визуализация первых узлов дерева решений

Метрики и лучшие параметры дерева решений вынесены в таблицу 2.

При сравнении метрик дерева решений и линейной регрессии видно, что средняя абсолютная ошибка дерева решений оказалась в два раза меньше, что значит, что дерево решений имеет сравнительно лучшую предсказательную способность. Величина коэффициента детерминации (R^2) также говорит о хорошем качестве модели.

Таблица 2 – Результаты обучения модели дерева решений

Модель	***LinearRegression***	***DecisionTreeRegressor***
Описание	Линейная регрессия методом наименьших квадратов	Дерево принятия решений
Метрики	MAE: 4,90980670806521 R^2: 0,3191816352444038	MAE: 2,372253479906388 R^2: 0,7678104905701086

Лучшие параметры	–	max_depth: 7 min_samples_leaf: 1 min_samples_split: 2 splitter: best

6. XGBoost

Третьей моделью, которая была использована для прогнозирования средней температуры, стала XGBoost. Данный алгоритм является одной из наиболее популярных реализаций алгоритма градиентного бустинга.

Бустинг – это ансамблевый метод машинного обучения, целью которого является объединение нескольких слабых моделей предсказания для создания одной сильной. Бустинг работает путём последовательного добавления моделей в ансамбль. Каждая следующая модель строится таким образом, чтобы исправлять ошибки, сделанные предыдущими моделями.

Для реализации использованы тренировочная и тестовая выборки данных и инструмент GridSearchCV для автоматического перебора гиперпараметров с выбором лучших из них. В качестве бейзлайна была использована модель линейной регрессии.

Полученные метрики и лучшие параметры XGBoost вынесены в таблицу 3. Также был получен график Feature Importance (важности признаков), представленный на рисунке 11.

При сравнении метрик XGBoost и линейной регрессии видно, что средняя абсолютная ошибка алгоритма градиентного бустинга оказалась меньше более чем в два раза, а это означает, что модель справляется с предсказыванием среднесуточной температуры лучше модели линейной регрессии, а также дерева решений. Величина коэффициента детерминации (R^2) также говорит об очень хорошем качестве модели.

Таблица 3 – Результаты обучения модели XGBoost

Модель	***LinearRegression***	***XGBRegressor***
Описание	Линейная регрессия методом наименьших квадратов	Алгоритм градиентного бустинга Extreme Gradient Boosting
Метрики	MAE: 4,90980670806521 R^2: 0,3191816352444038	MAE: 1,810006856918335 R^2: 0,840586245059967
Лучшие параметры	–	eta: 0,3 max_depth: 7 n_estimators: 15 subsample: 0,9

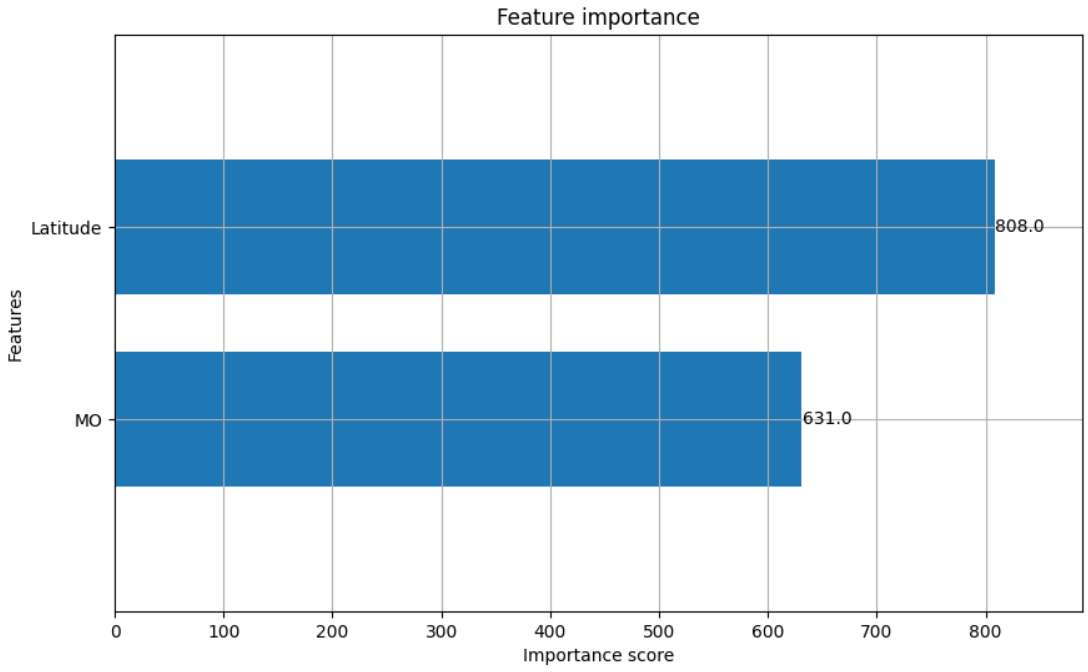

Рисунок 11 – График Feature Importance модели XGBoost

7. CatBoost

Четвертая модель для предсказания средней температуры – CatBoost. Данный алгоритм был разработан компанией «Яндекс» и является алгоритмом градиентного бустинга.

Особенностями данного алгоритма являются высокая скорость и эффективность, минимизация переобучения и самостоятельная обработка категориальных данных.

Для реализации, как и в предыдущие шаги, были использованы тренировочная и тестовая выборки данных и инструмент GridSearchCV для автоматического перебора гиперпараметров с выбором лучших из них. В качестве бейзлайна была использована модель линейной регрессии.

Полученные метрики и лучшие параметры CatBoost вынесены в таблицу 4. Также был получен график Feature Importance (важности признаков), представленный на рисунке 12.

При сравнении метрик CatBoost и линейной регрессии видно, что средняя абсолютная ошибка алгоритма градиентного бустинга оказалась меньше приблизительно в два раза, а это означает, что модель справляется с прогнозированием лучше модели линейной регрессии, но ненамного лучше дерева решений. Величина коэффициента детерминации (R^2) говорит о хорошем качестве модели.

Таблица 4 – Результаты обучения модели CatBoost

Модель	***LinearRegression***	***CatBoostRegressor***
Описание	Линейная регрессия методом наименьших квадратов	Алгоритм градиентного бустинга Category Boosting
Метрики	MAE: 4,90980670806521 R^2: 0,3191816352444038	MAE: 2,217935753319761 R^2: 0,7988047599821464
Лучшие параметры	–	eta: 0,3 max_depth: 7 n_estimators: 15 subsample: 1

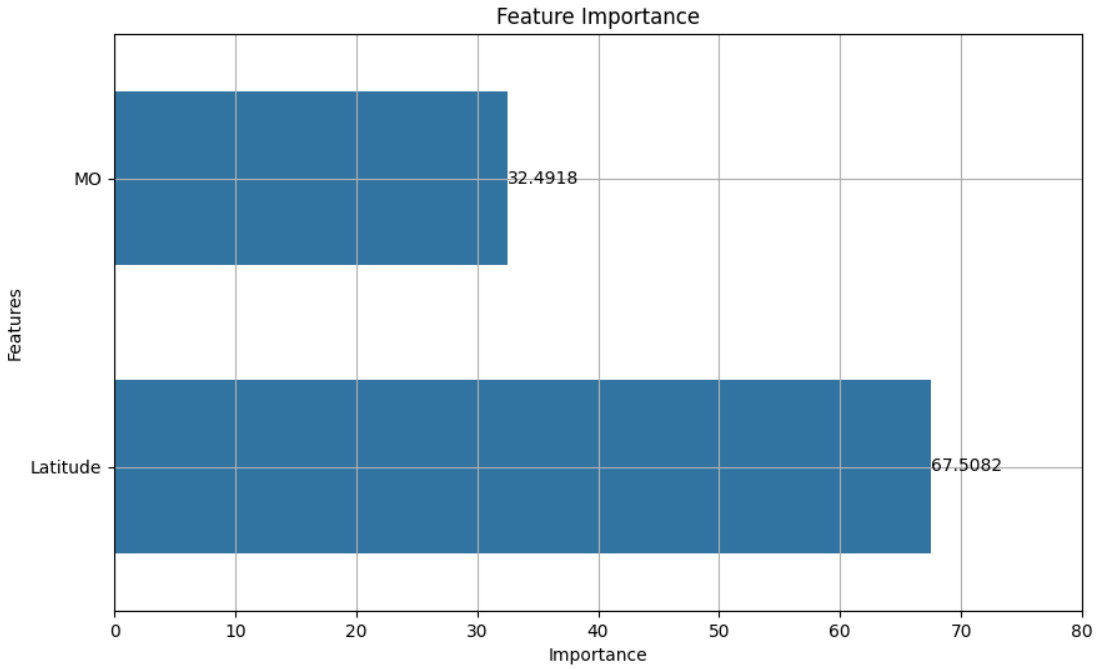

Рисунок 12 – График Feature Importance модели CatBoost

8. Нейронная сеть

Заключительная модель, которая была использована для предсказания средней температуры – нейронная сеть.

Искусственные нейронные сети представляют собой систему взаимодействующих искусственных нейронов, соединенных синапсами, имеющими свой вес. Именно благодаря этим весам, входная информация обрабатывается и превращается в результат.

Для реализации была использована библиотека TensorFlow. Структура нейронной сети была определена с помощью класса Model и представляет собой нейронную сеть с одним скрытым слоем. Количество нейронов во входном и выходном слое составляет два (по числу признаков) и один соответственно. В качестве функций активации были выбраны ‘relu’ для входных узлов и ‘sigmoid’ для узлов скрытого слоя.

Во время обучения нейронной сети изменялось количество нейронов скрытого слоя, а также размер пакета данных во время итерации процесса обучения с целью подбора лучших параметров.

Полученные результаты обучения нейронной сети были вынесены в таблицу 5. Также с помощью инструмента TensorBoard для модели с лучшими параметрами был получены кривые обучения и гистограммы весов, представленные на рисунках 13-17.

При сравнении метрик нейронной сети с предыдущими моделями видно, что средняя абсолютная ошибка больше аналогичной ошибки модели дерева решений и алгоритмов градиентного бустинга, но меньше ошибки модели линейной регрессии, а это означает, что нейронные сети в данном случае уступают деревьям решений и алгоритмам градиентного бустинга.

Таблица 5 – Результаты обучения нейронной сети

Модель	*Описание*	*Метрики*	*Лучшие параметры*
Нейронная сеть, реализованная с помощью TensorFlow	Нейронная сеть с одним скрытым слоем, двумя входными и одним выходным нейроном	MAE: 3,7349482	neurons_cnt: 64 batch_size: 128

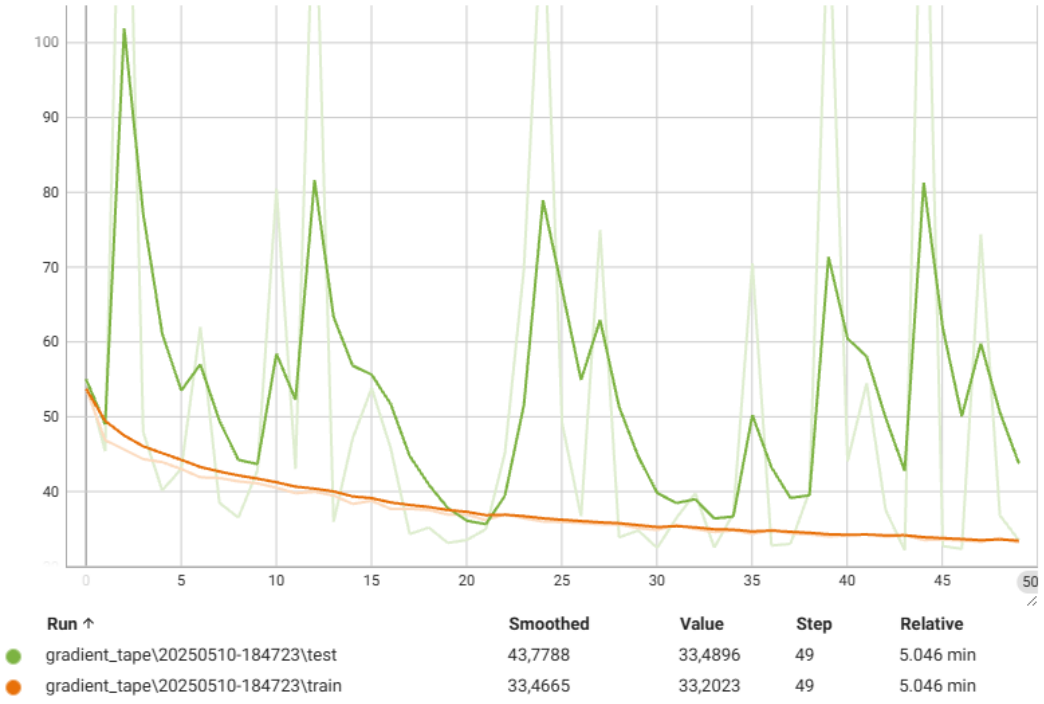

Рисунок 13 – График функции потерь нейронной сети с лучшими параметрами

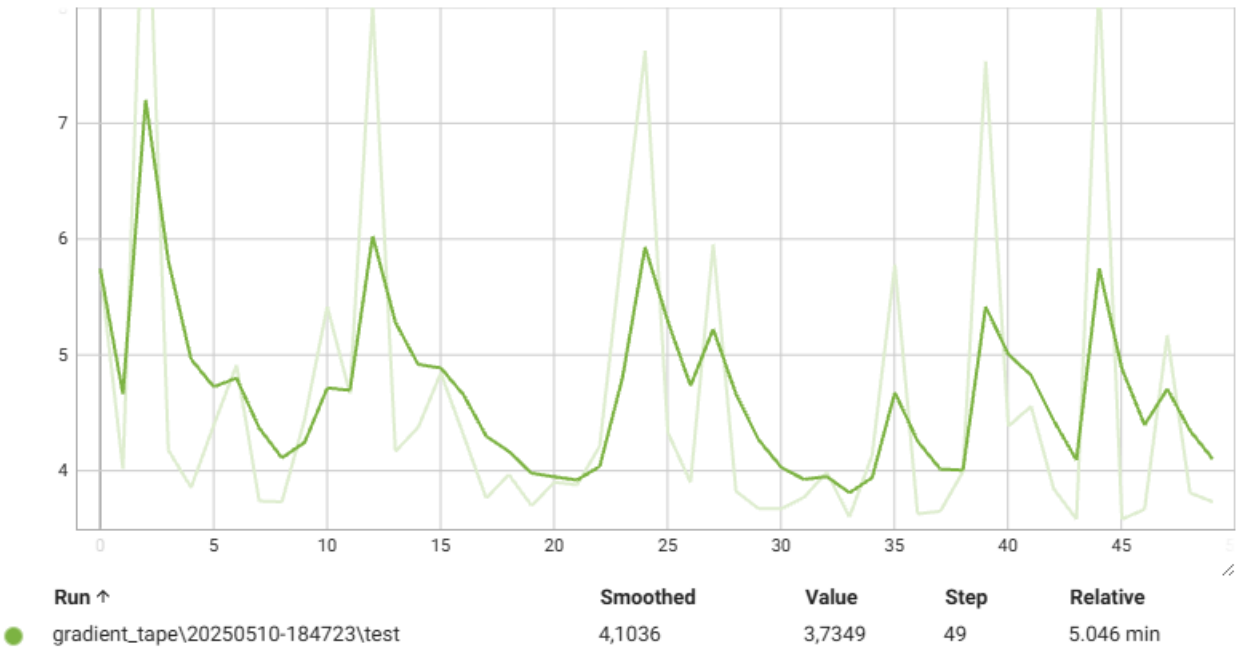

Рисунок 14 – График изменения ошибки нейронной сети с лучшими параметрами

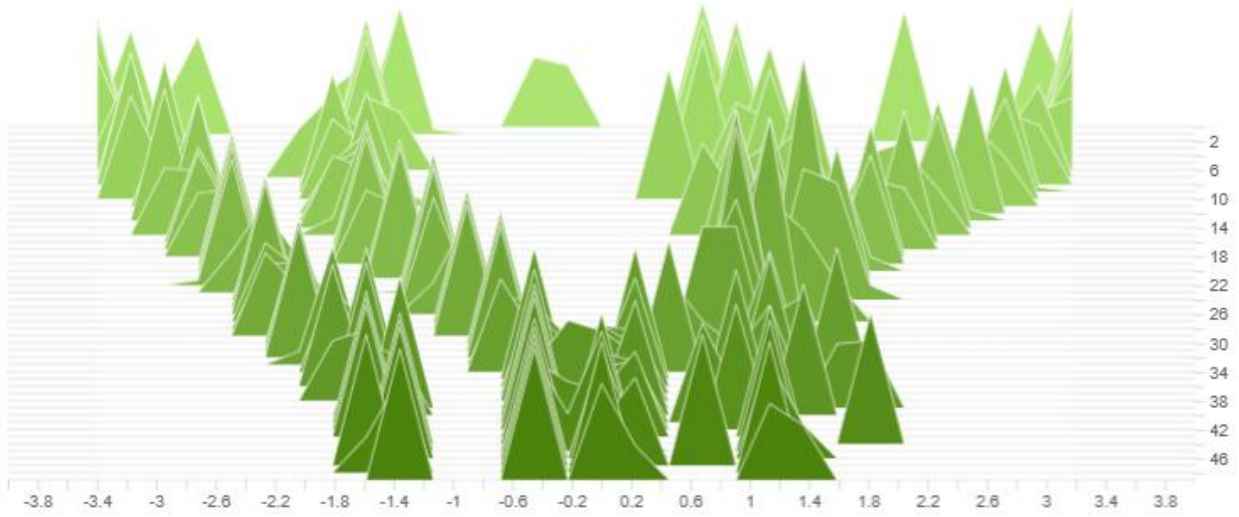

Рисунок 15 – Гистограмма весов входного слоя нейронной сети с лучшими параметрами

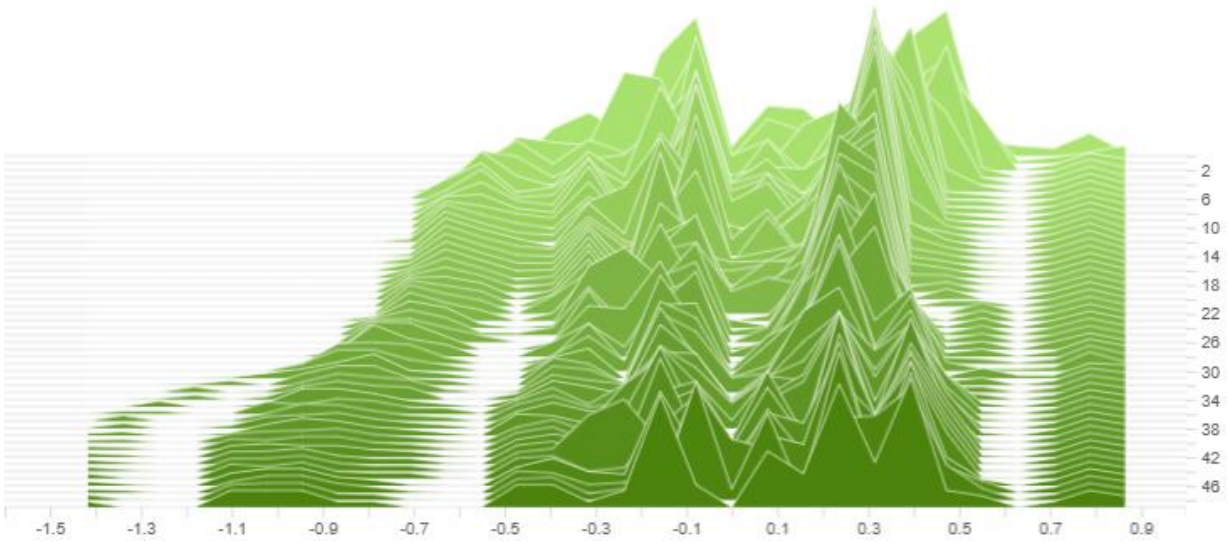

Рисунок 16 – Гистограмма весов скрытого слоя нейронной сети с лучшими параметрами

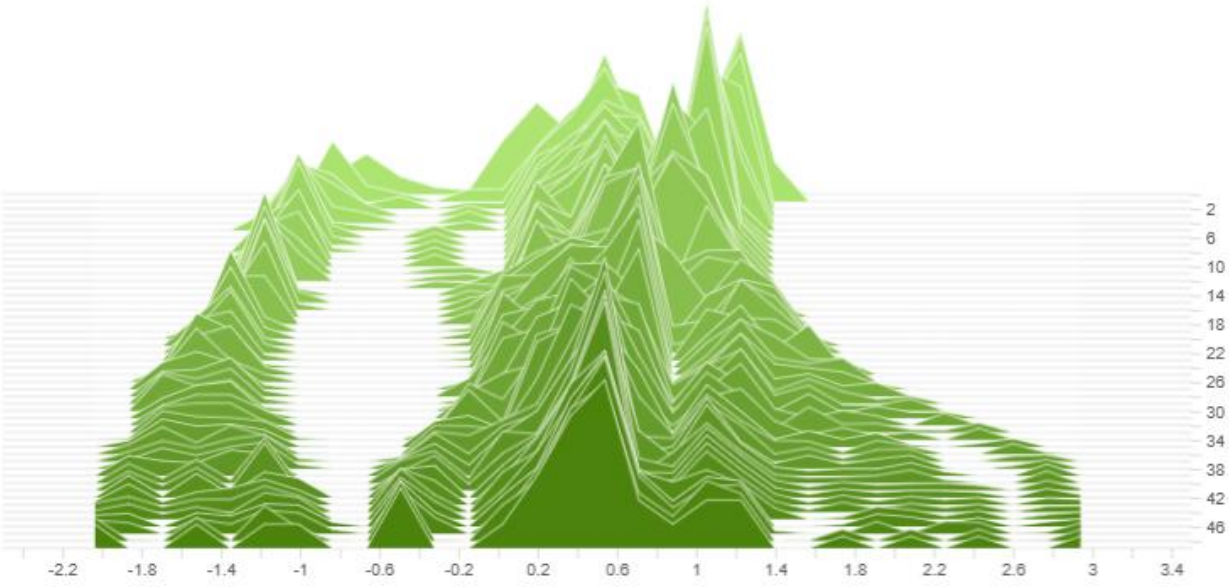

Рисунок 17 – Гистограмма весов выходного слоя нейронной сети с лучшими параметрами

9. Сравнительный анализ моделей

После окончания построения пайплайна DVC и реализации всех шагов, можно запустить последовательный процесс подготовки данных и обучение моделей (рисунок 18). Получившийся конечный вычислительный граф DVC представлен на рисунке 19.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER

>>>> DVC Terminal >>>>

Running: dvc exp run

Reproducing experiment 'edged-many'
WARNING: No file hash info found for 'C:\Users\ignat\source\repos\ML\data\models\neur_network.keras'. It won't be created.
Stage 'data_preparation' didn't change, skipping
Running stage 'linear_regression':
> python linear_regression.py -id data/prepared -od data/models -mn LinearRegression
0.3191816352444038
Baseline MAE:  9.153231844973405
Model MAE:  4.90980670806521
In range: [ -34.44444444 ; 40.0 ]
MAE in percents:  6.5952627425708945 %
intercept: 0    25.486328
Name: intercept, dtype: float64
list of coefficients: 0    0.136869
1   -0.231829
Name: coefficients, dtype: float64
Updating lock file 'dvc.lock'

Running stage 'decision_tree':
> python decision_tree.py -id data/prepared -od data/models -bm data/models/LinearRegression.joblib -mn DecisionTree
0.7678104905701086
Best params:  {'max_depth': 7, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
Baseline MAE:  4.90980670806521
Model MAE:  2.3722534799063846
In range: [ -35.0 ; 38.88888889 ]
MAE in percents:  3.210568619373895 %
Updating lock file 'dvc.lock'

Running stage 'xgboost':
> python xgboosting.py -id data/prepared -od data/models -bm data/models/LinearRegression.joblib -mn XGBoost
```

Рисунок 18 – Эксперимент, запущенный по пайплайну DVC

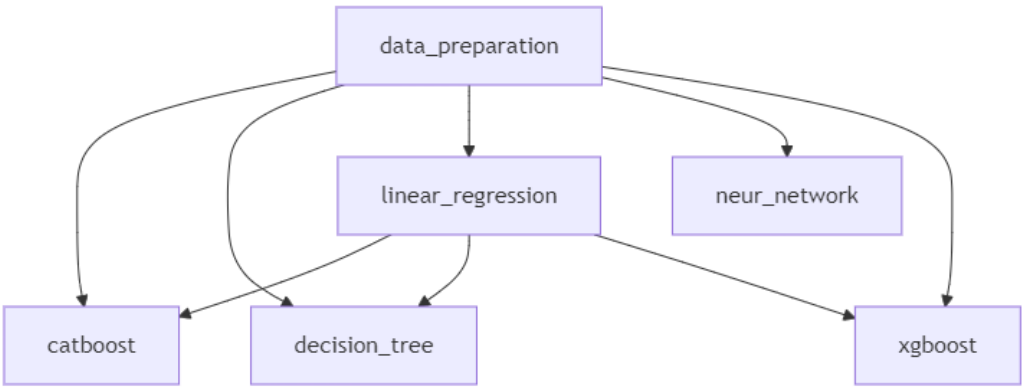

Рисунок 19 – Конечный вычислительный граф DVC

Для наглядности все метрики созданных и обученных моделей в ходе лабораторной работы были вынесены в одну таблицу (таблица 6) для проведения сравнительного анализа.

Таблица 6 – Сравнительная таблица метрик использованных моделей

Модель	***Метрики***	
	MAE	R^2
LinearRegression	4,90980670806521	0,3191816352444038
DecisionTreeRegressor	2,372253479906388	0,7678104905701086
XGBRegressor	1,810006856918335	0,840586245059967
CatBoostRegressor	2,217935753319761	0,7988047599821464
Нейронная сеть	3,7349482	0,5857395

Сравнивая метрики, можно увидеть, что самым эффективным в предсказывании показал себя один из алгоритмов градиентного бустинга, а именно XGBoost, коэффициент детерминации которого превысил 0,8. Наихудший результат показала модель линейной регрессии, чей коэффициент детерминации оказался менее 0,5.

Нейронная сеть потребовала долгого перебора параметров и больших вычислительных мощностей, однако среди всех использованных моделей она оказалась лучше в прогнозировании среднесуточной температуры только модели линейной регрессии. Предполагается, что увеличение количества эпох обучения положительно отразилось бы на точности нейронной сети, но увеличило бы и так довольно продолжительное время обучения.

Хорошие результаты показали модели дерева решений и CatBoost. Их коэффициенты детерминации оказались меньше аналогичного коэффициента XGBoost на 0,07 и 0,04 соответственно.

Таким образом, можно сделать вывод, что при наличии малого количества признаков во входных данных наиболее точный результат в прогнозировании показывает алгоритм градиентного бустинга XGBoost, а модель линейной регрессии справляется хуже всего в связи со слабой линейной корреляцией между входными и выходными данными.

Заключение

В ходе выполнения лабораторной работы были получены навыки анализа первичных данных и определения признаков взаимосвязи (EDA), понимания моделей: линейная регрессия, дерево решений, CatBoost, XGBoost, нейронные сети (MLP) и умения разрабатывать программу на языке Python для реализации представленных моделей.

Также были получены навыки управления версиями проектов и построения пайплайна DVC для упрощения процесса обучения и управления потоками данных.