Курс «Поступашки ML-старт»

Отчёт по «Реализации моделей машинного обучения на основе данных Wild-Blueberry-Yield-Dataset»

Подготовил: Даниил Михайленко
Студент 4 курса НИУ ВШЭ Нижний Новгород

Нижний Новгород 2025

Анализ данных

В данном проекте используется синтетический набор данных об урожайности дикой голубики, подготовленный командой Kaggle на основе реальных данных с помощью моделей глубокого обучения. https://data.mendeley.com/datasets/p5hvjzsvn8/1

Признаки	Единица измерения	Описание
Clonesize	м²	Средний размер клона голубики на поле
Honeybee	пчёлы/м²/мин	Плотность медоносных пчёл на поле
Bumbles	пчёлы/м²/мин	Плотность шмелей на поле
Andrena	пчёлы/м²/мин	Плотность пчёл рода Andrena на поле
Osmia	пчёлы/м²/мин	Плотность пчёл рода Osmia на поле |
MaxOfUpperTRange	°C	Максимальное значение температуры верхнего диапазона воздуха за день в течение сезона цветения
MinOfUpperTRange	°C	Минимальное значение температуры верхнего диапазона воздуха за день
AverageOfUpperTRange	°C	Среднее значение температуры верхнего диапазона воздуха за день
MaxOfLowerTRange	°C	Максимальное значение температуры нижнего диапазона воздуха за день
MinOfLowerTRange	°C	Минимальное значение температуры нижнего диапазона воздуха за день
AverageOfLowerTRange	°C	Среднее значение температуры нижнего диапазона воздуха за день
RainingDays	День	Общее количество дней с осадками (больше нуля) в течение сезона цветения
AverageRainingDays	День	Среднее количество дождливых дней за весь сезон цветения
fruitset	%	Доля цветов превратившихся в завязи
fruitmass	грамм	Средний вес плода
seeds	шт	Среднее число семян в каждом плоде

Целевая переменная – yield кг/гектар – урожайность дикой голубики на один гектар.
Пропущенных значений в данном датасете не обнаружено.

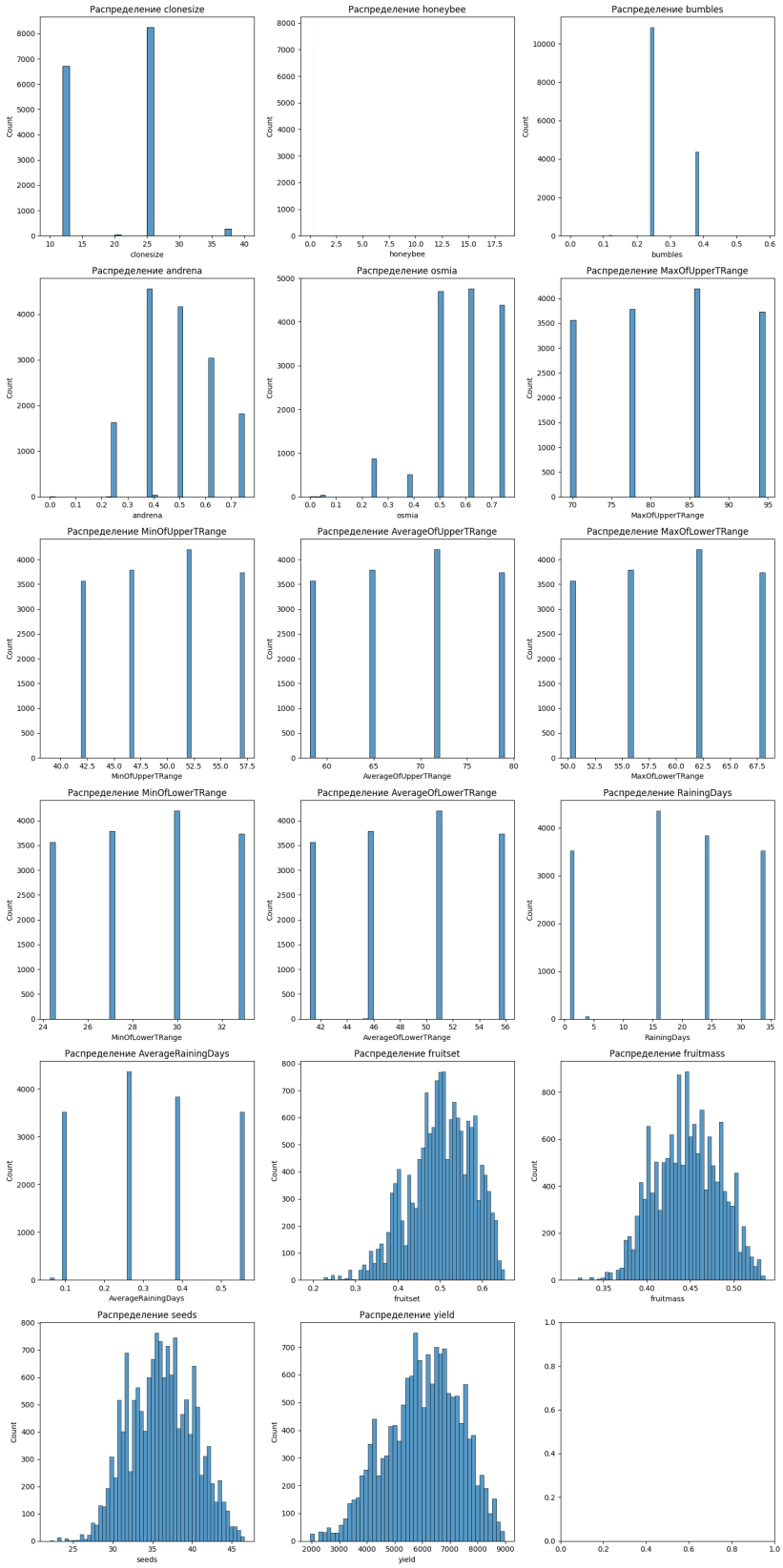

Рисунок 1

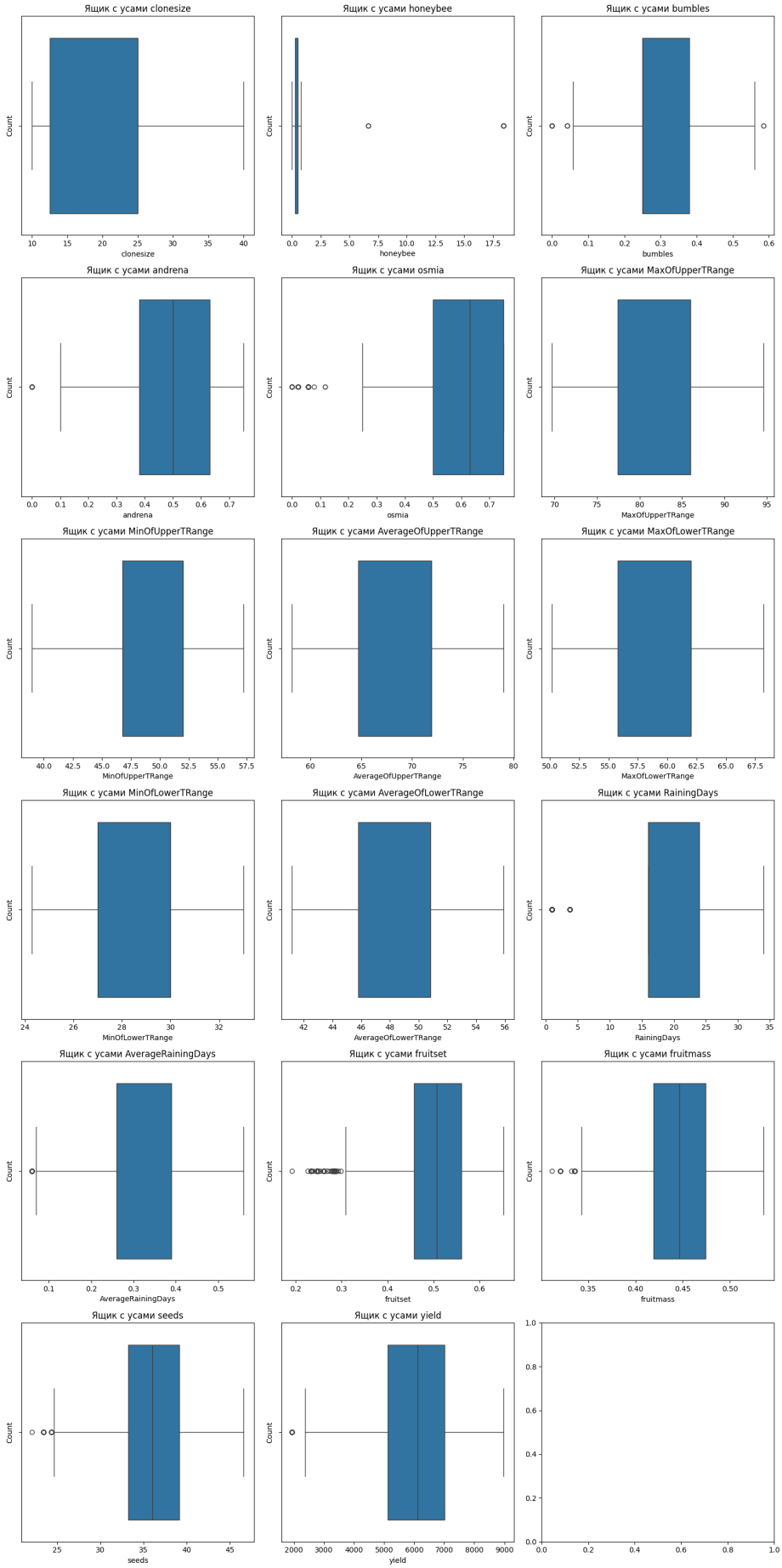
Ящик с усами clonesize
Count
clonesize
Ящик с усами honeybee
honeybee
Ящик с усами bumbles
bumbles
Ящик с усами andrena
andrena
Ящик с усами osmia
osmia
Ящик с усами MaxOfUpperTRange
MaxOfUpperTRange
Ящик с усами MinOfUpperTRange
MinOfUpperTRange
Ящик с усами AverageOfUpperTRange
AverageOfUpperTRange
Ящик с усами MaxOfLowerTRange
MaxOfLowerTRange
Ящик с усами MinOfLowerTRange
MinOfLowerTRange
Ящик с усами AverageOfLowerTRange
AverageOfLowerTRange
Ящик с усами RainingDays
RainingDays
Ящик с усами AverageRainingDays
AverageRainingDays
Ящик с усами fruitset
fruitset
Ящик с усами fruitmass
fruitmass
Ящик с усами seeds
seeds
Ящик с усами yield
yield

Рисунок 2

Из рисунка 1 можно заметить, что признаки: Clonesize, honeybee, bumbles, andrena, osmia, MaxOfUpperTRange, MinOfUpperTRange, AverageOfUpperTRange, MaxOfLowerTRange, MinOfLowerTRange, AverageOfLowerTRange, RainingDays, AverageRainingDays – демонстрируют низкую вариацию в данных и данные больше схожи на дискретные признаки. Для переменных fruirset, fruitset, seeds – распределение данных визуально похоже на нормальное, поэтому дополнительной предобработки данных переменных такой как логарифмирование не требуется для приведения наблюдений к нормальному виду, однако из рисунка 2 видно, что данные имеют различный масштаб, поэтому для моделей чувствительных к масштабу придётся применить скалирование признаков, чтобы улучшить перфоманс модели. Также можно заметить, что для некоторых признаков имеются наблюдения, которые лежат ниже Q1 - IQR*1.5, что говорит о наличии выбросов в наблюдениях. Однако было принято решение не работать с выбросами в данных, т. к. они характеризуют поведение отдельных кустов черники и данные наблюдения могут статистически являться выбросами в виду особых климатических условий, поэтому удаление данных приведёт к обеднению выборки и возможному снижению обобщающей способности моделей. Данные были нормализованы с помощью sklearn.Standard Scalar нормализация необходима для линейных моделей, которые чувствительны к масштабу данных Нормализованные данные использовались только для линейных: OLS, Ridge, Lasso, и MLP, ансамблевые и tree-based модели обучались на оригинальных не масштабированных данных

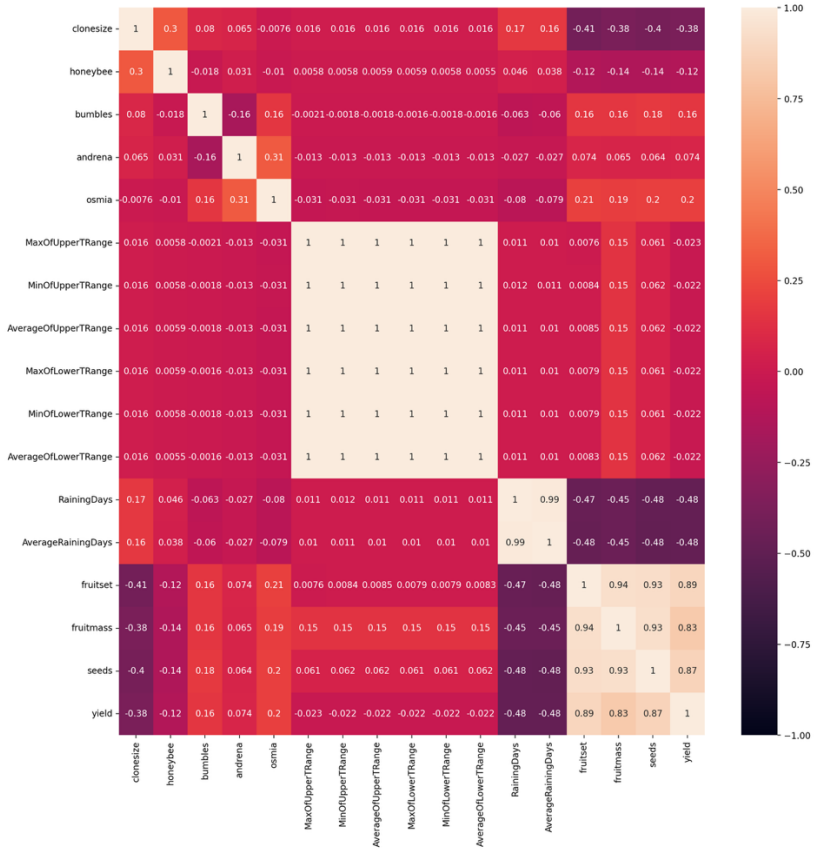

Рисунок 3

При анализе матрицы корреляций (рисунок 3) было установлено, что переменные отвечающие за колебания температуры демонстрируют чистую мультиколлинерность: фактически функциональная зависимость: поэтому придётся избавиться от признаков между которыми корреляция равна 1, т.к при использовании алгоритма МНК матрица признаков будет вырожденной, что приведёт к невозможности подобрать коэффициенты для модели, ну и также для других алгоритмов, которые менее чувствительны к проблемам мултиколлинерности: основанные на деревьях данные признаки будут источниками избыточной информации. Всё сказанное ранее также относится к признакам RainingDays и AverageRainingDays

Для температурных признаков можно создать дополнительную переменную, которая будет среднем высоких температур и низких температур, что можно трактовать как среднюю температуру в исследуемом сезоне (AveargeSeasonalTemp), таким образом информация о температурных условиях будет сохранена, но будет решена проблема мультиколлинеарнсти

Данные были разбиты в пропорции 80/20. 80% данных используются для обучения моделей. 20% это отложенная выборка, на которой проводилось тестирование.

Оценка качества работы моделей производилась по метрикам: R2, MAE, RMSE, MAPE

Используемые модели

Классические ML методы:
При работе с классическими моделями машинного обучения были использованы следующие модели

- Линейные модели
    - OLS (Метод наименьших квадратов) – без подбора гиперпарметров т.к подбирать особо и нечего
    - *Ridge регрессия с подбором гиперпарметров*
    - *Lasso регрессия с подбором гиперпарметров*

Для Ridge и Lasso регрессии гиперпарметры были подобраны с помощью поиска по сетке (RidgeCV, LassoCV) – были подобраны оптимальные параметры регуляризации и оптимизационная метрика – отрицательный MAE т. к. Kaggle соревнования проводит оценку по MAE

Все три линейные модели показали одинаковое качество работы различия составляют тысячные единицы. (рисунок
Модели, основанные на деревьях решений

- Decision Tree Regressor

Модель решающего дерева построена на основе деревьев, внутри которых происходит разделение объектов по некоторому признаков. В отличие от

ансамблевых методов здесь используется лишь одно решающее дерево. При работе с данным алгоритмом были подобраны оптимальные гиперпарметры: splitter, max_depth, max_features, min_samples_split:
После поиска по сетке оптимальные параметры выглядит следующим образом:
(criterion='absolute_error', max_depth=300,
max_features='log2', min_samples_split=5, splitter='random')

- Random Forest Regressor

Затем была применён ансамблевый метод – модель случайного леса. Метод случайного леса основан на множестве решающих деревьев внутри, которых происходит разделение, затем путём усреднения предсказаний в каждом из решающих деревьев. Данный алгоритм более устойчив к переобучению за счёт того, что деревья формируются на основе случайных выборок, а также случайного отбора признаков, по которым происходит разбиение. Для данного алгоритма с помощью поиска по сетке были подобраны оптимальные гиперпараметры: criterion='absolute_error', max_features='sqrt', min_samples_split=5

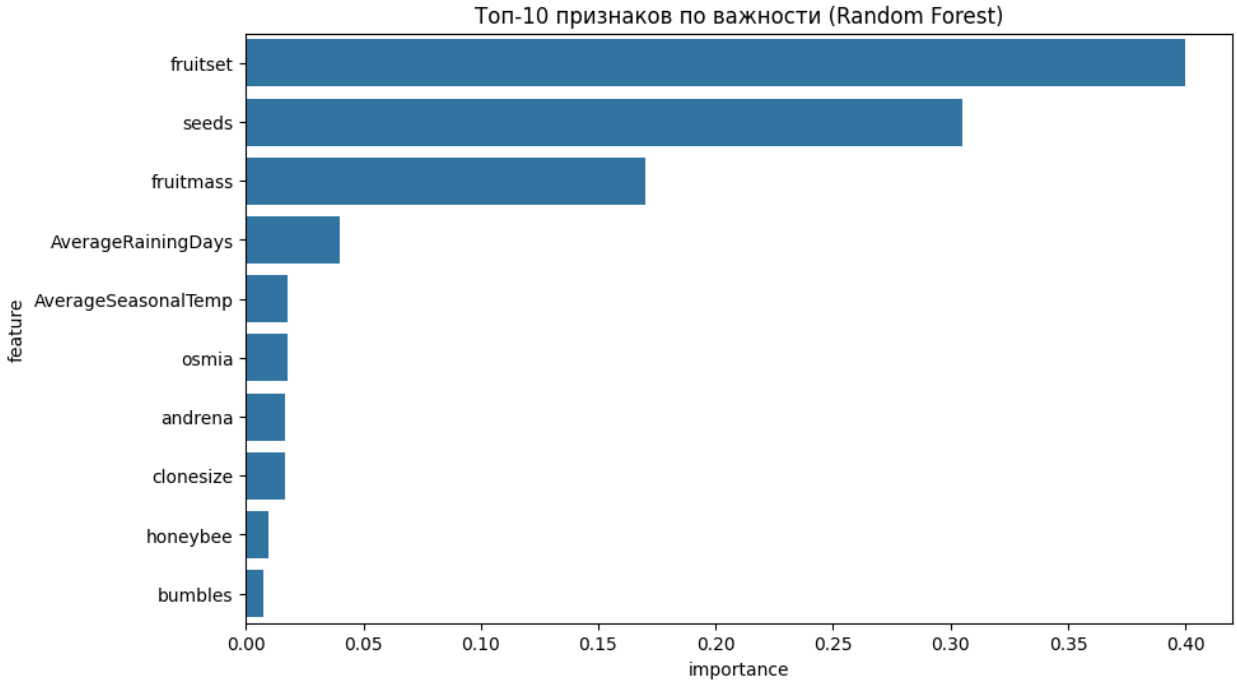

Рисунок 4

На графике выше представлена важность признаков для алгоритма случайного леса: наибольший вклад в предсказательную способность вносит признак fruitset и seeds, fruitmass на данные признаки приходится порядка 95% обобщающей способности модели.

- **XGBoost и LightGBM**

Затем были использованы для модели основанные на градиентных бустингах. Была использована модель Extreme Gradient Boosting и LightGradientBoostingModel. Данные модели основаны на «решающих» пнях - неглубокие деревья, у которых низкая предсказательная способность и каждое следующее дерево исправляет ошибки предыдущего дерева. Для поиска оптимальных гиперпарметров для модели XGBoost был использован RandomizedGridSearchCV так как обычный поиск по сетке занимал значительное число времени. Гиперпараметры, которые были оптимизированы для данной модели:

- learning_rate
- max_depth
- subsample
- n_estimators

для модели LightGBM был использован поиск по сетке, были подобраны следующие гиперпарметры:

- num_leaves
- max_depth
- learning_rate
- n_estimators
- min_child_sample
- subsample
- colsample_bytree
- reg_alpha
- reg_lambda

MLP

Архитектура модели

Многослойный персептрон (MLP), используемый в проекте, представляет собой полносвязную нейронную сеть для решения задачи регрессии. Его архитектура состоит из нескольких последовательных слоев:

Входной слой: Размерность входного слоя соответствует количеству признаков в предобработанном наборе данных. На вход подаются нормализованные данные о пространственных характеристиках растений, пчёлах и погоде.

Скрытые слои: Архитектура включает несколько скрытых слоев, каждый из которых состоит из полносвязных нейронов (nn.Linear). Между полносвязными слоями применяются слои BatchNorm1d для стабилизации обучения и ReLU в качестве активационной функции, которая вводит нелинейность. Также используется Dropout для регуляризации.

Выходной слой: состоит из одного нейрона (nn.ReLU), который выдает одно число — прогнозируемое значение урожайности (Yield). Важно отметить, что на последнем слое используется активационная функция, чтобы гарантировать предсказание только положительных значений ввиду того, что урожайность не может быть отрицательной

2. Обоснование выбора ключевых компонентов

Активационные функции:ReLU (nn.ReLU) выбрана для скрытых слоев. Её основное преимущество — простота вычислений (выдает 0 для отрицательных значений и само значение для положительных), что значительно ускоряет обучение. Кроме того, сигмоида и гиперболичнский тангес лежат в пределах от 0 до 1, что больше подходить для задач классификации нежели задачи регрессии.

Функция потерь:Средняя квадратичная ошибка выбрана в качестве функции потерь. MSE является стандартным выбором для задач регрессии. Она вычисляет сумму квадратов разностей между предсказанными и истинными значениями.

Оптимизатор:Адам выбран в качестве оптимизатора. Adam сочетает в себе преимущества двух других популярных алгоритзамов — Adagrad и RMSprop. Он эффективно адаптирует скорость обучения для каждого веса в сети, что делает его надёжным и быстрым выбором для большинства задач. Параметр weight_decay был добавлен для L2-регуляризации, что помогает предотвратить переобучение.

3. Гиперпараметры

Количество слоев и их размерность: Архитектура модели имеет несколько скрытых слоев, размерность которых уменьшается: hidden_sizes=[512, 256, 128, 64, 32 8, 4, 2]. Такой подход (так называемый "воронкообразный") позволяет модели сначала уловить общие, высокоуровневые закономерности, а затем детализировать их, сжимая информацию.

Размер батча (batch_size): Размер батча определяет количество образцов, которые используются для одного шага обновления весов. В данном случае это число не указано в коде, но оно задаётся при создании DataLoader.

Количество эпох (num_epochs):500 эпох выбрано для обучения. Эпоха — это один полный проход по всему обучающему набору данных. Такое количество обеспечивает достаточное время для сходимости модели, при этом не приводя к избыточному времени обучения. При необходимости можно использовать механизм ранней остановки для предотвращения переобучения, если производительность на валидационном наборе перестанет улучшаться.

Скорость обучения (learning_rate): 0.05 была выбрана как начальная скорость обучения. Это довольно высокое значение, которое позволяет модели быстро сходиться на ранних этапах обучения. Для дальнейшего улучшения сходимости используется StepLR — планировщик скорости обучения, который уменьшает learning_rate в 10 раз (gamma=0.1) каждые 30 эпох

(step_size=30). Это позволяет модели "тонко настраивать" веса на более поздних этапах обучения.

Dropout Rate: 0.25 была выбрана как вероятность отключения нейрона. Это означает, что 25% нейронов будут временно отключены во время обучения, что предотвращает их созависимость и улучшает обобщающую способность модели.

Анализ времени обучения каждого алгоритма.

	время_обучения
OLS	0.0252
xgboost_best	0.1567
RidgeCV	0.2311
LassoCV	0.2613
Decision_tree_best	0.9361
rfc_best	49.6185
LightGBM_best	88.5603
model_xgb_cv	106.0636
DecisionTreeLearningCV	139.5042
MLP	212.6239
RandomForestCV	3369.0597

Рисунок 5

Линейные алгоритмы показали наиболее высокую скорость обучения менее одной секунды. Модель случайного леса оказалась наиболее трудозатратной подбор оптимальных гиперпарметров занял порядка 60 минут, а также обучение модели с лучшими параметрами порядка одной минуты. Подбор гиперпарметров для бустингах занял примерно 2 минут, что можно считать наилучшим результатом, модели, основанные на бустингах демонтируют баланс между скоростью обучения и качеством работы модели. Многослойный персептрон обучался порядка 4 минут

Сравнение перфоманса используемых моделей

	MAE	RMSE	R2	MAPE
XGBoost	352.405	560.684	0.815	0.062
LGBM	354.092	560.219	0.816	0.062
RFR	355.577	566.455	0.812	0.063
OLS	370.743	582.079	0.801	0.066
Ridge	370.743	582.079	0.801	0.066
Lasso	370.743	582.079	0.801	0.066
Tree_regressor	481.611	718.717	0.697	0.085
mlp	1076.353	1250.121	0.083	0.174

Рисунок 6

На графике выше представлена сравнительная таблица производительности каждого из алгоритмов, как можно видеть наилучшие результаты были продемонстрированы градиентными бустингами и моделью случайного леса. В целом простые линейные модели, показали результаты не сильно хуже, чем более продвинутые модели, основанные на бустингах. Отличие составляет примерно 0.001 на метрике MAPE (Mean Absolute Percentage Error). В среднем лучшая модель допускает ошибку в предсказании на 6.2%, что можно считать приемлемым результатом, особенно если говорить о предсказании в сельскохозяйственном секторе.

Наихудший результат показала нейронная модель метрика R2 составляет 0.17, что говорит о том, что MLP объясняет лишь 17% вариации в данных. MLP работает намного хуже простая линейная регрессия.

На графике обучения MLP можно заметить, что модель сошлась достаточно быстро, менее чем за 100 эпох.

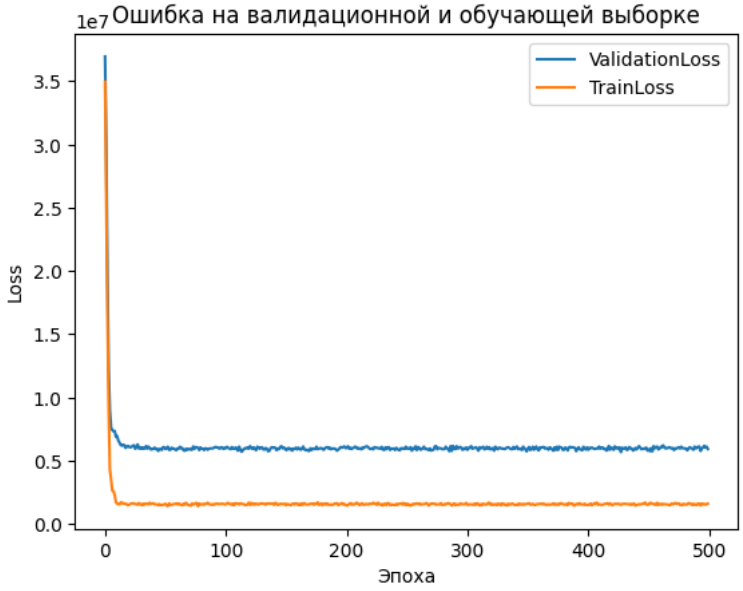

Рисунок 7

Вывод:

Таким образом, можно сделать вывод, что модель XGBoost с подбором гиперпарметров показала наиболее высокий результат со значением ошибки MAE на тестовой выборке – 352.4.

При загрузке предсказаний на Kaggle MAE на неразмеченных данных составил (Private Score) – 345.99. Лучший результат на соревновании составил 327. 1100 позиция из 1875.