**Лабораторная работа №1.**

# Обучение нейросетевых регрессора и классификаторов.

Вариант 2.

Подготовил: Боровских В.Б.

Группа: 932003

1. **Описание наборов данных.**

Существует три основных набора данных, использованных в данной работе:

1. Общий набор данных для банка, который содержит информацию об клиентах банка. Это включает такие данные, как возраст, профессия, образование, семейное положение, наличие кредита и другие.

bank-additional-full.csv - размеченный набор данных для бинарной классификации. Данные связаны с прямыми маркетинговыми кампаниями португальского банковского учреждения. Цель бинарной классификации состоит в том, чтобы предсказать, подпишется ли клиент на срочный банковский депозит (переменная y)

1. Набор данных о здоровье плода, который содержит данные по различным показателям, связанным со здоровьем плода в период беременности.

fetal\_health.csv - набор данных для многоклассовой классификации. Данный набор данных составлен из результатов кардиотокографии. Для этого мы создаем многоклассовую модель, чтобы классифицировать функции КТГ по трем состояниям здоровья плода: нормальное (1), подозрительное (2) и паталогическое (3).

1. Набор данных об энергетическом потреблении, который содержит информацию о потреблении энергии в разных областях.

DS\_2019\_public.csv - набор данных для регрессии. Данный набор данных содержит информацию о потреблении энергии в зданиях. В этом наборе данных мы будем классифицировать по TOTALBTUCOL - общее потребление энергии для кондиционирования воздуха, в тысячах BTU. BTU - Британская тепловая единица.

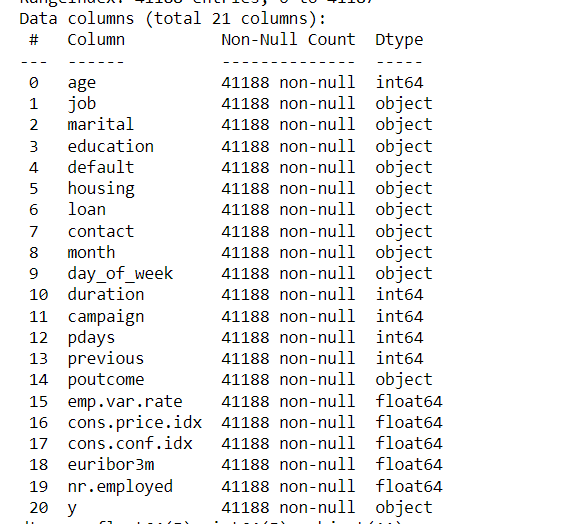
1. **Признаки, которые были использованы для анализа.**

Признаки, используемые для анализа, преобразовывались и масштабировались в процессе предварительной обработки данных. В этом процессе категориальные признаки были преобразованы в числовые, а все признаки были приведены к одному масштабу.

**В пункте A:**

Признаки, использованные для анализа:

1. marital (семейное положение) был перекодирован в Marital Binary, где женатые/замужем получают 1, а все остальные получают 0.
2. default (есть ли у субъекта непогашенные кредиты) был перекодирован в Default Binary, где заемщики с кредитами в просрочке получают 1, а все остальные получают 0.
3. housing (имеет ли субъект ипотеку) был перекодирован в Housing Binary, где те, кто имеет ипотеку, получают 1, а все остальные получают 0.
4. loan (есть ли у субъекта личные заемные средства) был перекодирован в Loan Binary, где те, у кого есть личные заемные средства, получают 1, а все остальные получают 0.
5. poutcome (результат прошлой маркетинговой кампании) был перекодирован в Poutcome Binary, где успешные результаты получают 2, несуществующие получают 1, а все другие получают 0.
6. И остальные признаки которые не требовали перекодировки:

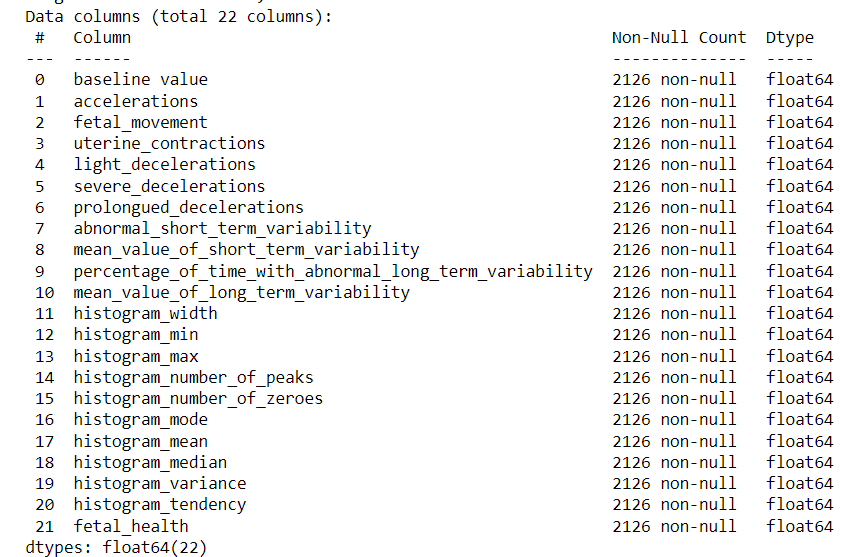


Следующие признаки были исключены из анализа:

1. y - Исключен, поскольку этот столбец был перекодирован в Target Binary и использовался в качестве целевого столбца для нашей задачи классификации.
2. job - Работа клиента, исключен, это связано с тем, что этот столбец является номинальным и требует дополнительной предварительной обработки, чтобы быть полезным для модели.
3. education - Образование клиента. Исключено теми же причинами, что и job.
4. contact - Способ связи с клиентом исключен по той же причине, что и job и education.
5. month, day\_of\_week - Месяц и день недели последнего контакта. Исключены из-за незначительного влияния на вероятность подписания депозита.

**В пункте B:**

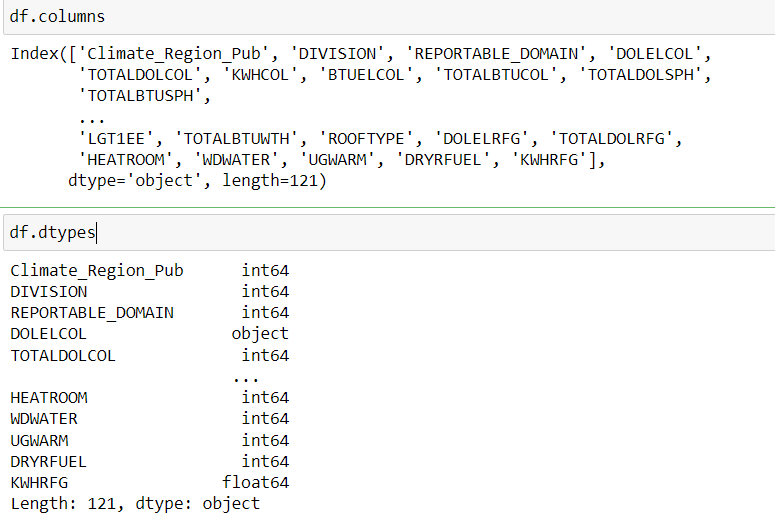
Были использованы все признаки, кроме fetal\_health, поскольку этот столбец был перекодирован в Target Multi и использовался в качестве целевого столбца для нашей задачи классификации. Эти данные представляют особенности здоровья плода, полученные на основе кардиографии плода



**В пункте C:**

Для анализа использовались все столбцы набора данных "DS\_2019\_public.csv", кроме 'TOTALBTUCOL'. Этот столбец был выделен как целевая переменная (y), которую необходимо было предсказать, в то время как все остальные столбцы были использованы как признаки (X) для модели.

Остальные данные были предварительно обработаны (были удалены строки с некорректными значениями), затем прошли масштабирование при помощи MinMaxScaler (все признаки были приведены к диапазону между 0 и 1).



1. **Параметры архитектур и обучения нейронных сетей, использованные для обучения.**

Построены три различные модели: для бинарной классификации, многоклассовой классификации и регрессии.

Для каждой из них были использованы полносвязные нейронные сети. В модели бинарной классификации на последнем слое использовалась сигмоидная функция активации, в многоклассовой классификации - softmax, а в модели регрессии - линейная функция активации. Количество нейронов в скрытых слоях варьировалось в зависимости от модели.

A)

**Архитектура сети**: Сеть является бинарным классификатором и составлена из последовательности слоев (Sequential). Сначала добавляется Dense слой, который имеет 4 нейрона, с функцией активации ReLU, и принимающий на вход данные с 15 признаками. Затем следует еще один Dense слой с одним нейроном и функцией активации 'sigmoid', предназначенный для предоставления выходного значения.

**Компиляция модели**: Функция потерь, использованная в этой модели, - это бинарная кросс-энтропия, которая часто используется для задач бинарной классификации. Оптимизатор 'Adam' использовался для настройки весов модели. Метрика, используемая для оценки производительности модели в процессе обучения, - это точность(accuracy).

**Обучение модели**: Модель была обучена в течение 25 эпох, с размером пакета 10. Был использована валидационная выборка и коллбеки для ранней остановки и сохранения лучших весов модели. Режим валидации был поставлен на 'max' и показатель для проверки - 'val\_accuracy'. Остановка обучения осуществляется, если нет улучшений в показателе 'val\_accuracy' в течение 15 эпох.

B)

**Архитектура сети**: Аналогично предыдущему примеру, эта сеть создана с использованием модели последовательных слоев в Keras. Она также содержит два плотных слоя. Первый плотный слой имеет 8 нейронов с функцией активации ReLU. Второй - 4 нейрона с функцией активации "softmax". Функция активации "softmax" используется в задачах многоклассовой классификации, так как она предоставляет вероятностные оценки, которые суммируются в 1.

**Компиляция модели**: Сеть компилируется с функцией потерь "categorical\_crossentropy", которая используется в задачах многоклассовой классификации. В качестве оптимизатора используется 'adam'. Метрика, используемая для оценки производительности модели, - 'accuracy'.

**Обучение модели**: Модель обучается в течение 50 эпох с размером пакета 20. Используются валидационные данные, а также коллбеки для ранней остановки и сохранения лучших весов модели. Показателем валидации является 'val\_accuracy'. Мониторинг прекращается, если нет улучшения указанного показателя в течение 15 эпох.

C)

**Архитектура нейронной сети**: Создается последовательная модель в Keras, состоящающая из трех полносвязных слоев. Первый слой содержит 120 нейронов с функцией активации ReLU, второй слой состоит из 60 нейронов с функцией активации ReLU и на выходе один слой с линейной активацией. Функция активации ReLU используется для предотвращения проблемы затухания градиента, а линейная активация выбирается для задачи регрессии.

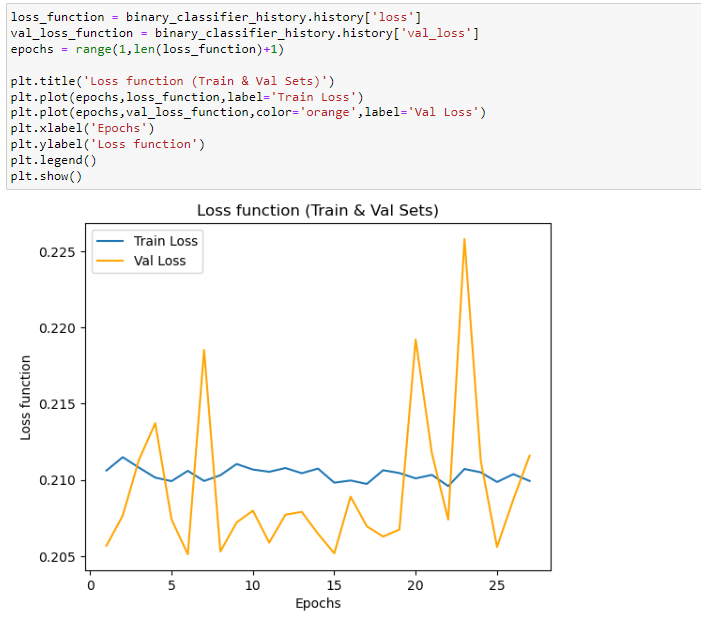
**Компиляция модели**: Модель компилируется с функцией потерь MSE (среднеквадратичная ошибка), использующуюся для задач регрессии, и оптимизатором Adam. Также отслеживается метрика MAE (средняя абсолютная ошибка).

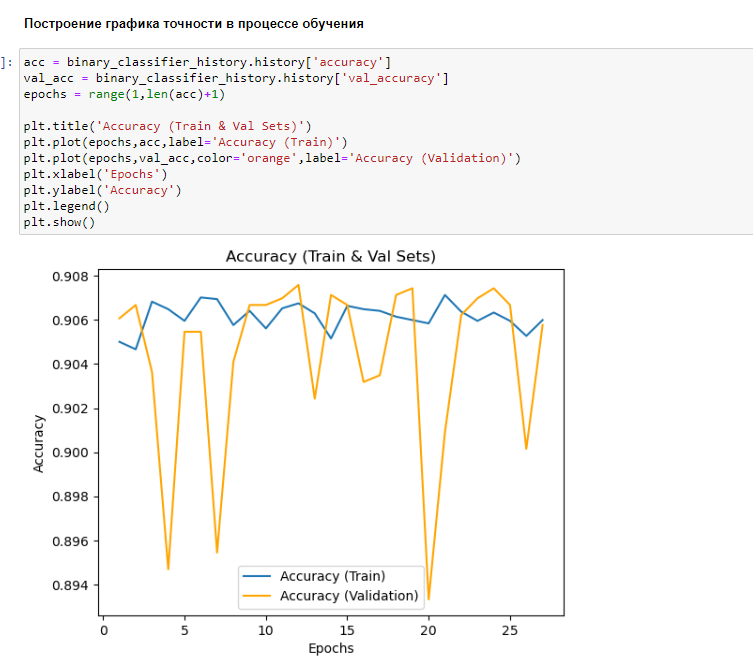
**Обучение модели**: Модель обучается в течение 350 эпох с размером пакета 40. Добавляются функции обратного вызова для ранней остановки обучения, если модель перестает улучшаться (с терпением в 20 эпох), и сохранения лучших весов модели. Валидационные данные используются в процессе обучения для оценки производительности модели на данных, которые она не видела.

1. **Графики обучения для архитектур нейронных сетей с лучшими характеристиками эффективности**

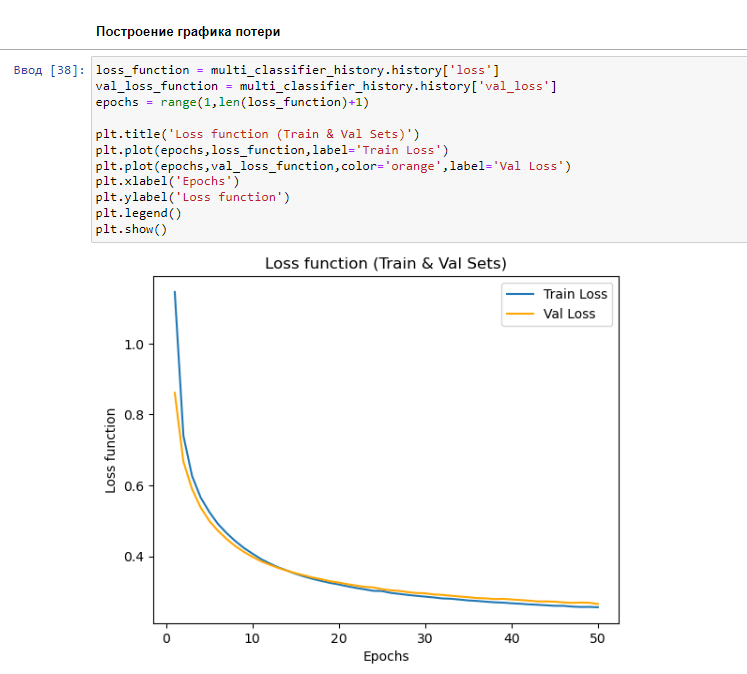
Графики обучения, представленные в коде, показывают как потери, так и точность модели в процессе обучения для тренировочного и валидационного набора данных.

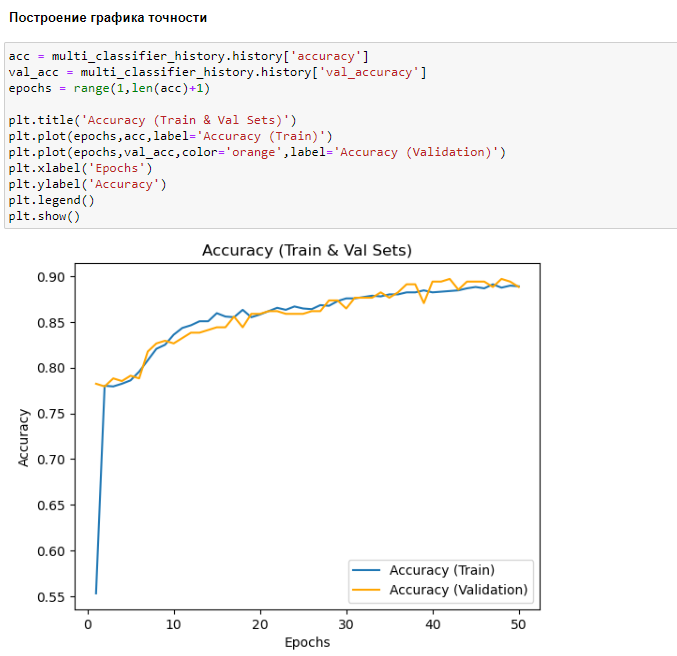
A)



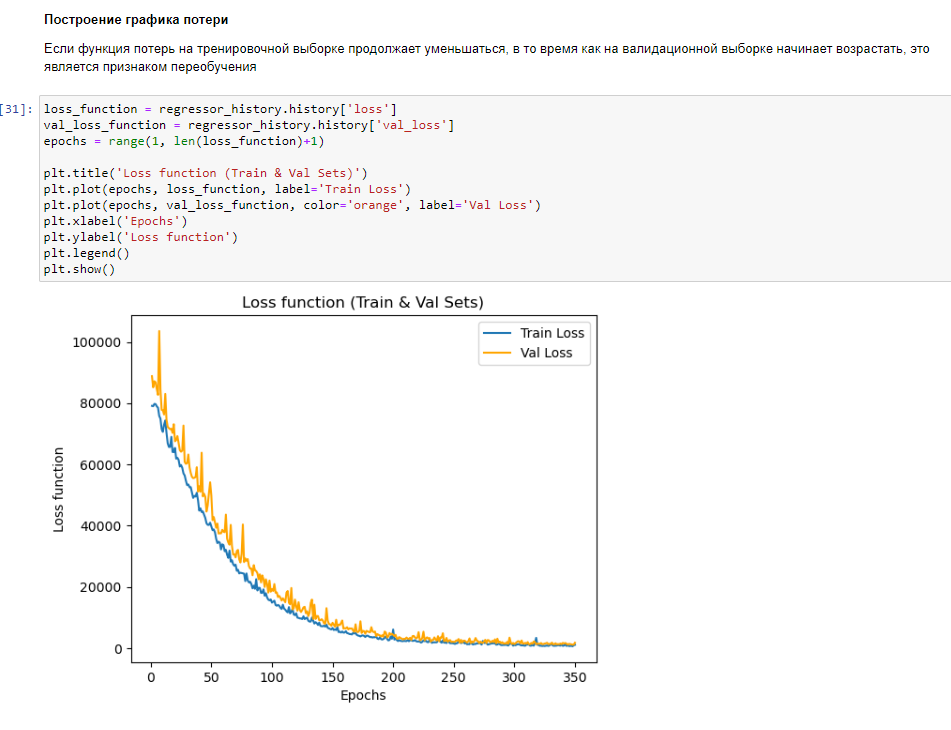


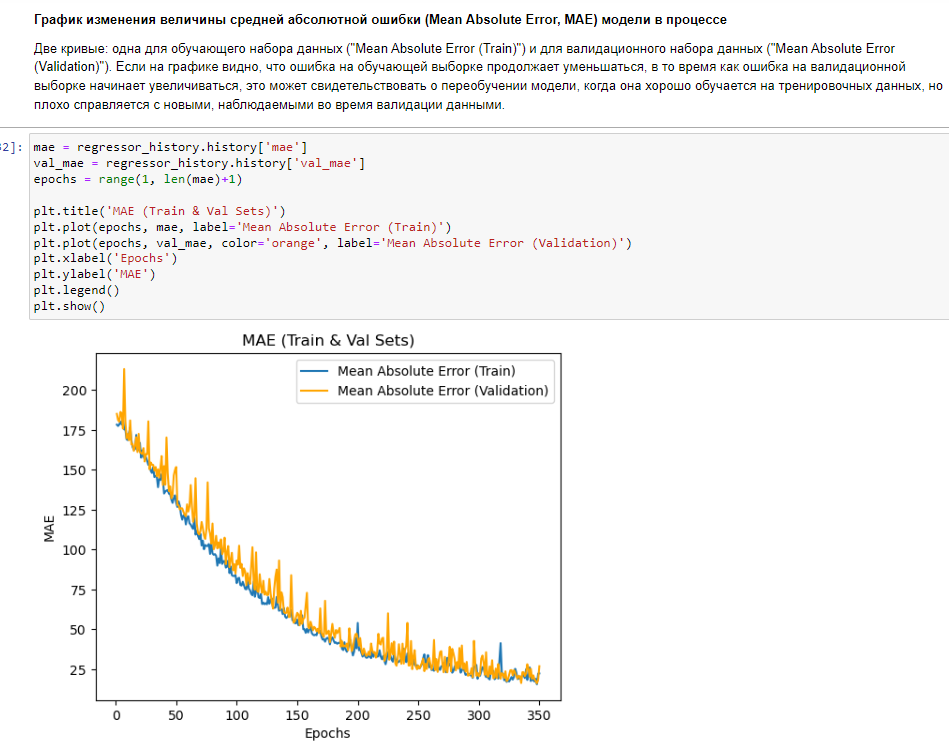
B)





C)

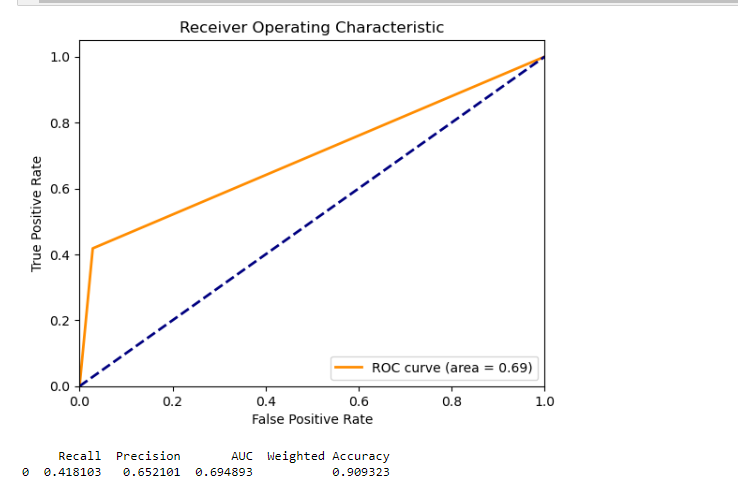




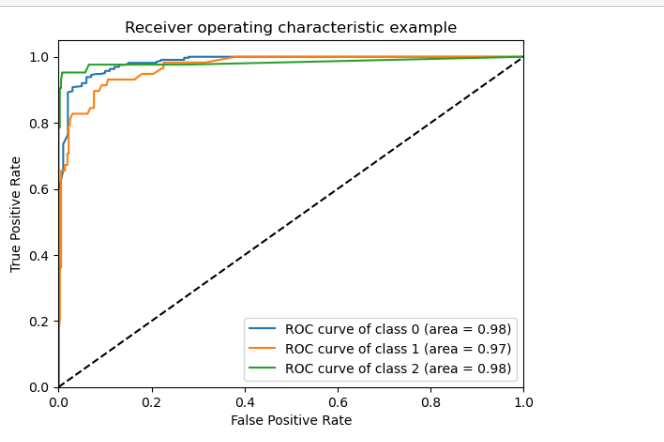
1. **ROC-кривые классов для лучших классификаторов.**

ROC-кривые, представленные в коде, дают визуализацию того, как хорошо наши бинарные и многоклассовые классификаторы справляются с обобщением на новые данные.

A)



B)



1. **Оценки моделей на тестовых выборках в виде таблиц/ диаграмм, отображающих метрики качества.**

Оценки моделей, показанные в коде, включают такие метрики, как точность, полноту, MAE, MSE и коэффициент детерминации (R2).

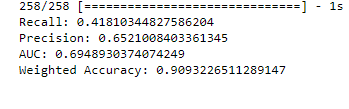
Все указанные оценки были произведены на разделенных тестовых выборках, чтобы обеспечить корректное измерение эффективности модели. Графики позволяют визуально анализировать процесс обучения машины, а таблицы и диаграммы предоставляют числовые оценки метрик качества моделей.

Дополнительно вы можете обратить внимание на использование EarlyStopping и ModelCheckpoint, это обеспечивает сохранение лучшей модели и прекращение обучения в случае, если результаты не улучшаются на протяжении фиксированного количества эпох.

A)

1. Recall (полнота) равен 0.418, что означает, что модель правильно классифицировала только 41.8% положительных результатов из всех истинных положительных результатов. Recall важен, когда ложно отрицательные результаты являются недопустимыми, и мы хотим минимизировать их количество.
2. Precision (точность) составляет 0.652, что означает, что модель верно классифицировала 65.2% от всех предсказанных положительных результатов. Precision важна, когда ложно положительные результаты являются недопустимыми, и мы хотим минимизировать их количество.
3. AUC (площадь под ROC-кривой) равна 0.695. ROC-кривая показывает зависимость между чувствительностью (True Positive Rate) и специфичностью (1 - False Positive Rate) модели при изменении порога классификации. AUC меряет общую производительность модели и должен быть максимально близким к 1.
4. Weighted Accuracy (взвешенная точность) составляет 0.909. Взвешенная точность учитывает дисбаланс классов в данных, присваивая вес каждому классу в зависимости от его доли в выборке.

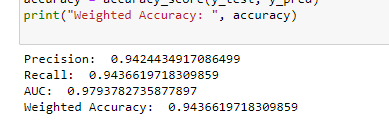
Вывод: Модель имеет достаточно высокую точность и AUC, что может говорить о ее хорошей способности классифицировать данные. Однако полнота (recall) невысока, что означает, что модель может пропускать много истинно положительных результатов.



B)

1. Precision (точность): это метрика, показывающая, какая доля объектов, отнесённых моделью к классу 1, действительно относится к классу 1. Модель имеет точность 0.942, что свидетельствует о том, что почти 94.2% всех положительных предсказаний модели действительно верны.
2. Recall (полнота): это метрика, показывающая, какую часть объектов класса 1 из всех объектов класса 1 модель смогла обнаружить. Величина полноты в 0.9436 говорит о том, что модель идентифицировала почти 94.36% всех истинных положительных случаев.
3. AUC-ROC: AUC равен 0.979. Эта метрика говорит о том, насколько модель способна отличить положительные и отрицательные данные – чем ближе значение к 1, тем лучше. 0.979 – высокий показатель.
4. Weighted Accuracy также равно 0.944, что является высокой оценкой.

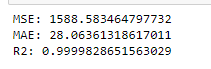
Вывод: Основываясь на этих данных, можно сказать, что мультиклассовый классификатор в рамках существующего набора данных работает отлично.



C)

1. MSE (Mean Squared Error, Средняя квадратическая ошибка) составляет 1588.58. MSE является общепризнанной мерой качества регрессионных предсказаний и чем меньше её значение, тем качественнее работает модель.
2. MAE (Mean Absolute Error, Средняя абсолютная ошибка) равна 28.06. Это абсолютное значение разности между реальными и прогнозными значениями. Чем меньше это число, тем лучше качество модели.
3. Коэффициент детерминации R^2 составляет почти 1 (0.99998), что является отличным значением и означает, что модель объясняет почти 100% вариации зависимой переменной.

Вывод: по представленным метрикам, модель показывает отличные результаты, причем коэффициент детерминации превосходно близок к 1.



1. **Программный код.**