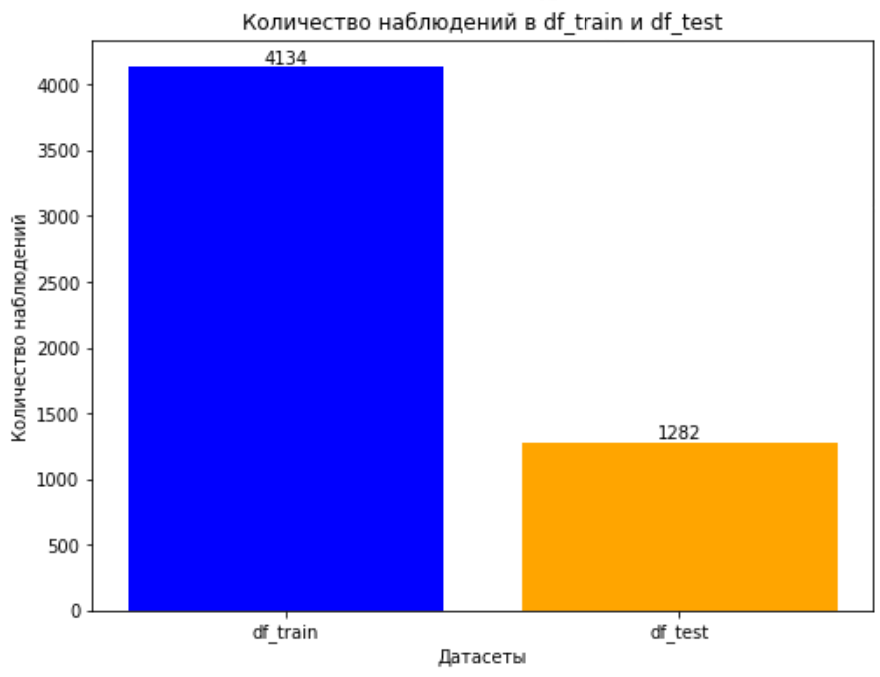
Ильин Матвей

Тестовое задание

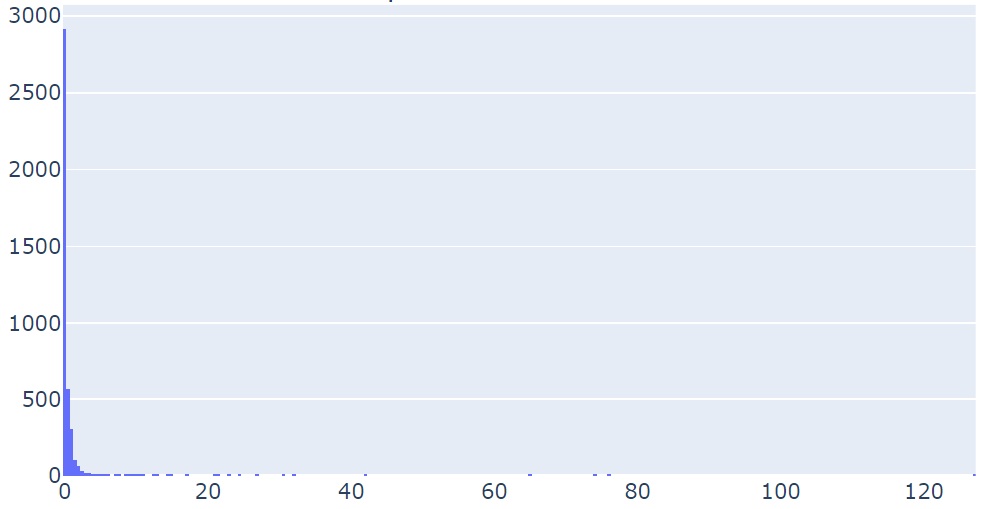
1. **Предобработка и первичный анализ данных**
2. Необходимо преобразовать тип данных в столбце **start\_time** в формат datetime для более удобного разделения на тренировочную и тестовую выборку.
3. Определение пропущенных значений в датафрейме: в столбце **max\_eth\_1st** было найдено **6** пропущенных значений. В данном столбце содержатся вещественные числа, и пропуски в данном случае следовало заменить средним значением в столбце. Однако, в силу очень малого количества пропусков мы можем просто удалить данные строки из фрейма.
4. Далее данные были разделены на тренировочную и тестовую выборку по указанной дате. В тренировочном наборе содержится **76.3 %** данных**,** в тестовом **23.7 %.**



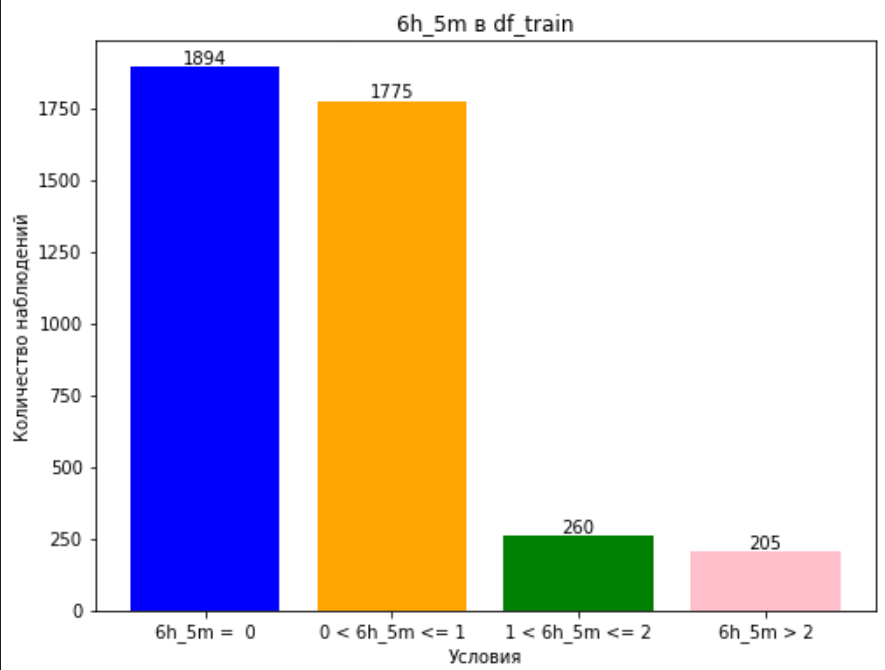
Стоить отметить, что в данном случае df\_test никак не участвует в выборе и обучении модели, и служит только для тестирования.

1. Изучение переменной **6h\_5m** в тренировочном наборе данных**:**

Для начала построим график распределения переменной **6h\_5m.**

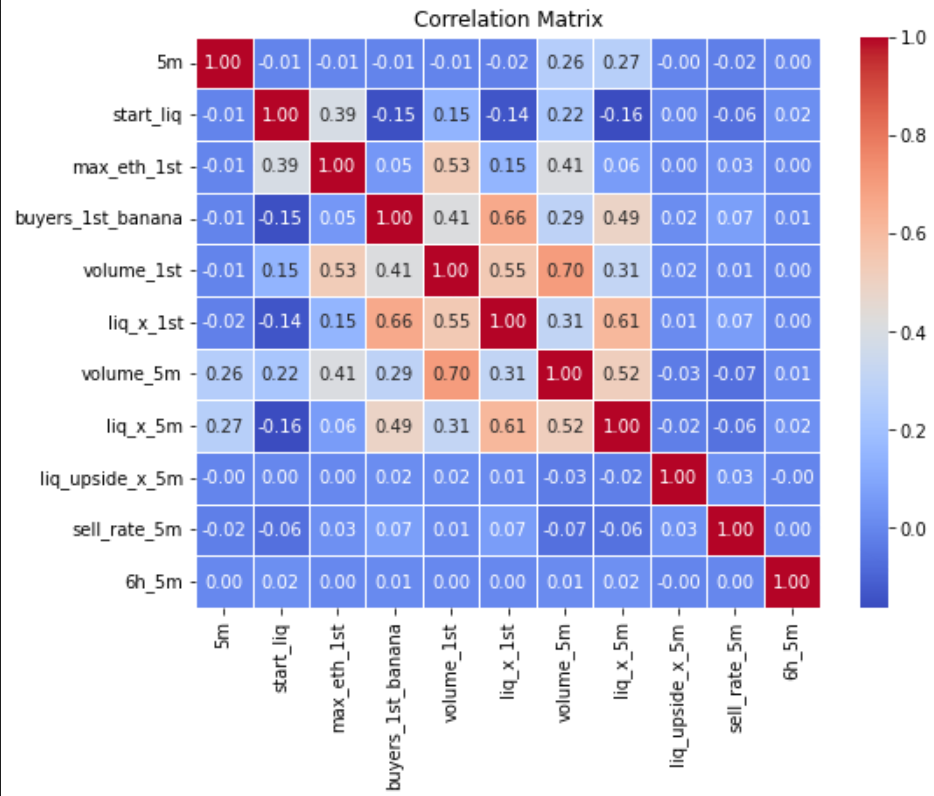


Как видно из графика, почти все значения находятся в промежутке от 0 до 1, однако наблюдаются сильные выбросы. Построим гистограмму с 4 столбцами для более подробного изучения данных:



Как видно из гистограммы **45.8 %** монет являются “скамом”, на котором мы теряем 100% своих вложений.

1. Также полезной информацией для нас будет изучение корреляционной матрицы регрессоров и зависимой переменной. После очистки от сильно коррелирующих между собой регрессоров (> 0.75) мы получаем следующую корреляционную матрицу:



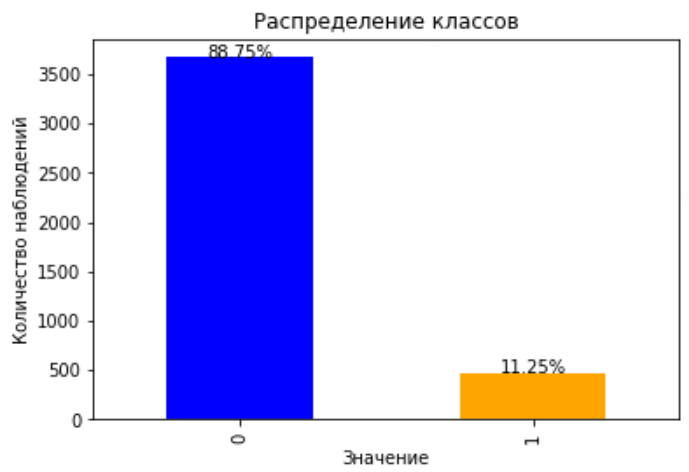
Как видно из матрицы, оставшиеся регрессоры очень слабо коррелируют с переменной 6h\_5m, что в целом является логичным фактом, т.к. на итоговую стоимость монет могут влиять не только внутренние, но и внешние факторы, и интервал в 6 часов является довольно большим относительно первых 5 минут.

Оставшиеся после устранения мультиколлинеарности регрессоры будут использоваться для построения моделей

6) Изучение распределения классов в обучающей выборке:

В основе алгоритма отбора монет для покупки будет использоваться модель классификации. Для начала необходимо создать фиктивную переменную на основе значений “6h\_5m”.

* Значения 1: монеты, у которых есть профит (6h\_5m > 1)
* Значения 0: монеты, у которых нет профита (6h\_5m <= 1)



Как видно из графика, в обучающем наборе данных наблюдается сильный дисбаланс классов, что требует особого подхода обучения и оценки качества модели классификации.

1. **Построение и обучение модели классификации**

В качестве итоговой модели классификации будет использован алгоритм **Random Forest**. Для подбора оптимальных гиперпараметров будет использована библиотека skopt.

Изменяемые параметры модели:

* ***n\_estimators***: количество деревьев в случайном лесу.
* ***max\_features***: максимальное количество признаков, которые рассматриваются при поиске лучшего разделения в деревьях.
* ***max\_depth***: максимальная глубина каждого дерева в лесу.

Неизменяемые параметры модели:

* ***class\_weight***: в модели используется значение “balanced”. Использование 'balanced' особенно полезно для несбалансированных классов, чтобы учесть разное количество наблюдений в каждом классе.

Поскольку в тренировочном наборе данных наблюдается сильный дисбаланс классов, то для обучения модели необходимо использовать кросс-валидацию со стратифицированным разбиением данных на подвыборки.

Стратифицированное разбиение необходимо для того, чтобы распределение классов в каждом фолде оставалось примерно таким же, как в исходном наборе данных, что существенно улучшает качество обучения модели в случае сильного дисбаланса классов.

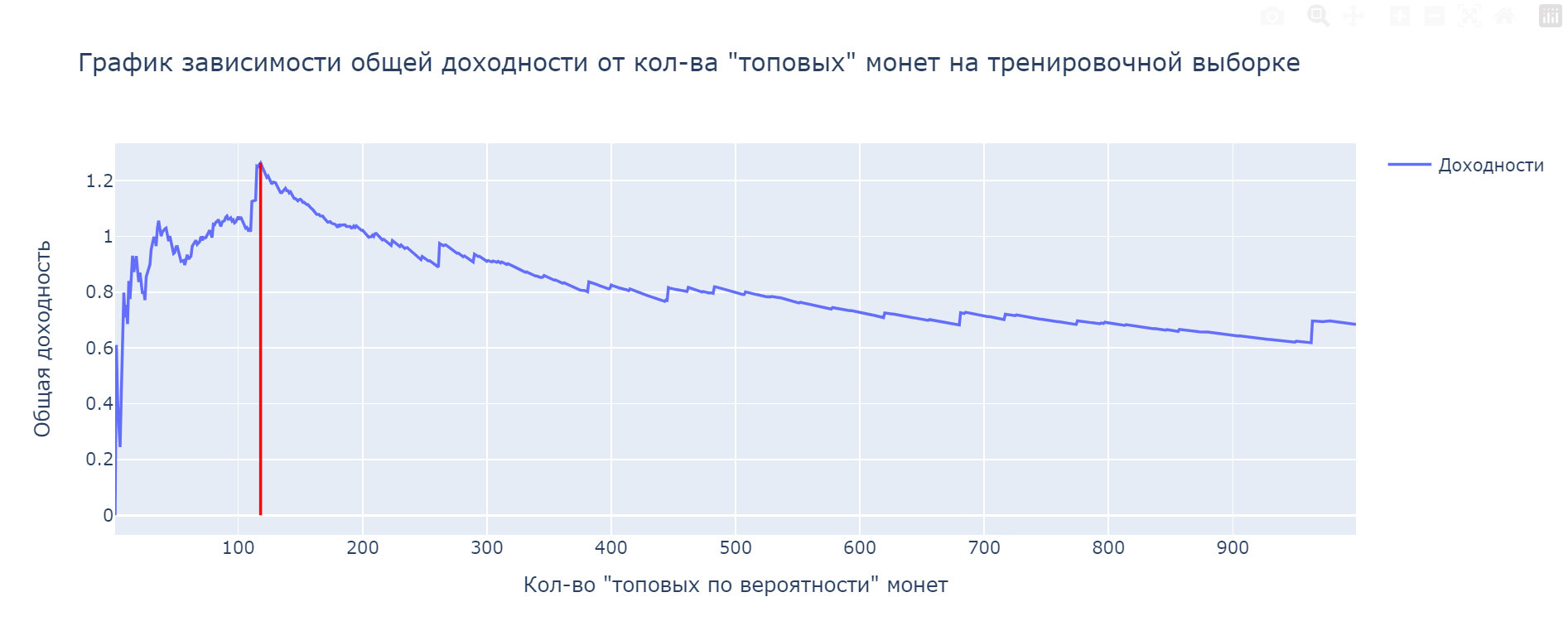
В качестве метрики качества обучения модели будет использоваться Precision-Recall (PR) AUC - метрика, используемая для оценки качества модели в задачах бинарной классификации, особенно в случаях, когда классы несбалансированы.

В случаях сильного дисбаланса классов, где классы 0 и 1 представлены неравномерно, PR AUC может быть более информативным показателем эффективности модели по сравнению с ROC AUC, потому что PR AUC фокусируется на способности модели правильно предсказывать положительные примеры.

Оптимальные гиперпараметры модели и значение метрики вы можете увидеть в таблице ниже:

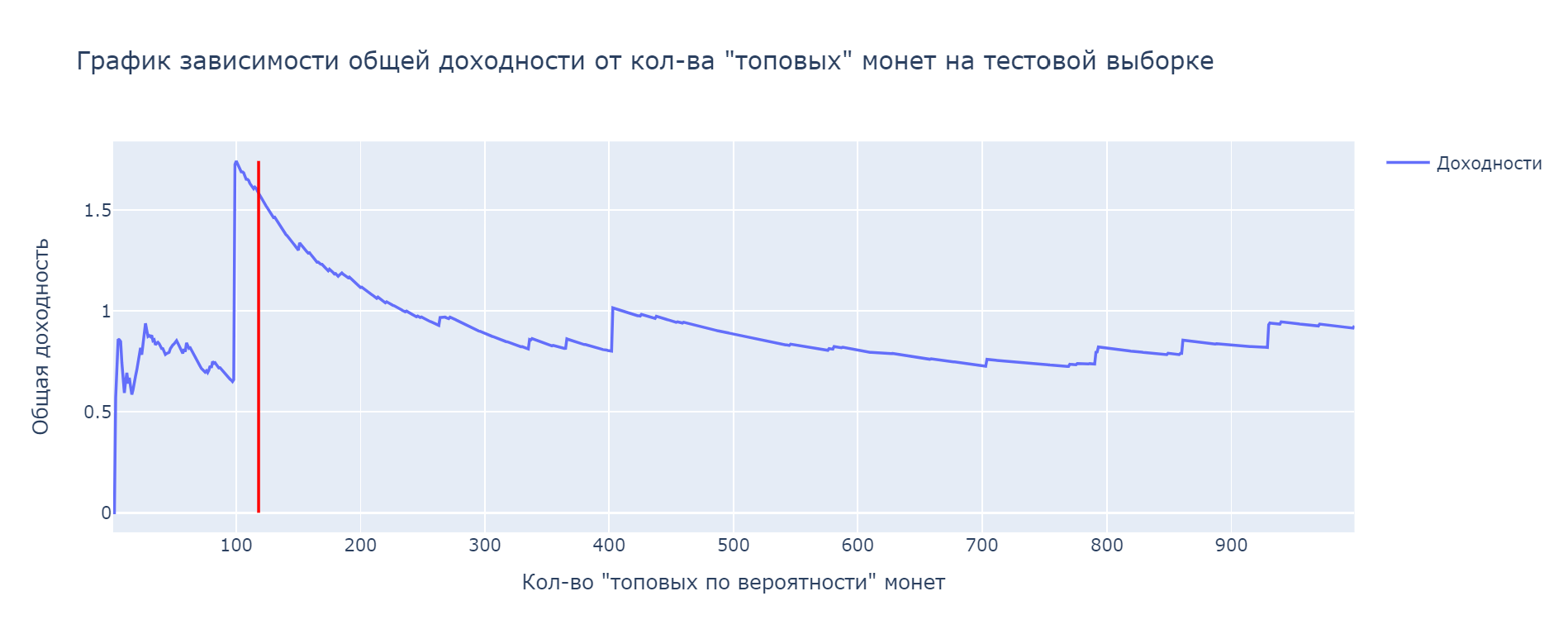
|  |  |
| --- | --- |
| Precision-Recall (PR) AUC | 0.19 |
| n\_estimators | 300 |
| max\_features | 16 |
| max\_depth | 1 |

1. **Создание алгоритма отбора монет.**
2. Получаем вероятностные прогнозы классов наблюдений из тренировочной выборки
3. Отсортировываем наблюдения по убыванию предсказанных вероятностей принять значение 1 в тренировочном наборе данных
4. Строим график общих доходностей от покупки различного количества монет с максимальными вероятностями принять значения 1.



Из графика видно, что максимальная прибыль достигается при покупке 118 "топовых по вероятности" монет и составляет **26.36** %. Таким образом 118 монет будет являться ориентировочным значением, которое максимизирует общую доходность “портфеля” монет.

1. **Проверка алгоритма на тестовом наборе данных**
2. Получаем вероятностные прогнозы классов наблюдений из тестовой выборки
3. Отсортировываем наблюдения по убыванию предсказанных вероятностей принять значение 1 в тестовом наборе данных
4. Строим график общих доходностей от покупки различного количества монет с максимальными вероятностями принять значения 1.

****

На тестовом наборе данных использование **118** "топовых" монет показывает почти максимальный результат доходности. Прибыль составляет **58.29%.**

**Проверка других гипотез:**

В ходе выполнения тестового задания мной были проверены другие подходы и методы решения поставленной задачи, такие как:

1. Построение модели линейной регрессии: c удалением выбросов и без удаления
2. Использование других алгоритмов классификации: логистическая регрессия, XGBClassifier

Однако все модели показали отрицательную доходность на тестовых данных.