

Выпускная квалификационная работа

Разработка модели по прогнозу загрузки денежных средств для банкоматной сети

Выполнил: Шишкин Армен Владимирович

Руководитель: к.т.н. Семендяев Родион Юрьевич

Анализ данных на языке Python

Актуальность темы

Оптимизация управления неработающими активами представляет собой приоритетную цель для любого банка.

В контексте этой цели, одной из важных подзадач является эффективное управление остатками наличных денежных средств в сети банкоматов.

Для решения данной проблемы широко используется прогнозирование потоков денежных средств в банкоматной сети, направленное на поддержание оптимальных остатков при инкассации банкоматов.

Установка и поддержание оптимальных остатков в банкоматах направлена на достижение нескольких важных целей:

- Максимизация прибыли банка за счет эффективного управления неработающими активами.
- Снижение риска ликвидности банка.
- Минимизация репутационного риска.

Цели и задачи

Цель работы

Разработка модели для прогнозирования загрузки денежных средств с использованием языка программирования Python на основании имеющихся данных.

Задачи

- Анализ исходных данных по выдаче денежных средств через банкоматы;
- Обработка данных и добавление дополнительных признаков;
- Построение и обучение модели машинного обучения;
- Оценка полученных результатов.

Декомпозиция задачи

- Результатом работы модели по загрузке денежных средств для банкоматной сети является определение резерва наличности, необходимого для подкрепления банкоматов в течение определенного календарного времени;
- Модель должна дать возможность прогнозировать расход денег за заданный интервал времени с целью исключения дефицита наличности, необходимой для обеспечения банкоматных операций;
- Таким образом решаемая задача сводится к прогнозированию целевого значения, зависящего от потока клиентов, демонстрирующих повышенный либо пониженный спрос на наличные денежные средства, который, в свою очередь, определяется днем недели, днем месяца (например, дни получения заработной платы или социальных пособий), временем суток, выходными и праздничными днями, сезоном и другими факторами;

Решаемая задача – прогноз объема выдач наличных денежных средств через банкоматы.

Исходные данные

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2468572 entries, 0 to 2468571
Data columns (total 34 columns):
#   Column                Dtype
---  -
0   year                  int64
1   month                 object
2   day                   int64
3   weekday               object
4   hour                  int64
5   atm_status            object
6   atm_id                int64
7   atm_manufacturer      object
8   atm_location          object
9   atm_streetname        object
10  atm_street_number     int64
11  atm_zipcode            int64
12  atm_lat                float64
13  atm_lon                float64
14  currency               object
15  card_type              object
16  service                object
17  message_code           float64
18  message_text           object
19  weather_lat            float64
20  weather_lon            float64
21  weather_city_id        int64
22  weather_city_name      object
23  temp                   float64
24  pressure               int64
25  humidity               int64
26  wind_speed             int64
27  wind_deg               int64
28  rain_3h                float64
29  clouds_all             int64
30  weather_id             int64
31  weather_main           object
32  weather_description    object
33  amount                 int64
dtypes: float64(7), int64(14), object(13)
memory usage: 640.3+ MB
```

В качестве исходных данных выступает массив операций по выдаче денежных средств через банкоматы банка Spar Nord Bank (Дания).

Источник данных: портал Kaggle.

Датасет содержит **2.5 миллиона транзакций**, осуществленных в **113 банкоматах**.

Период данных: с **01.01.2017 по 31.12.2017**.

Основные признаки, доступные в исходных данных:

- Дата и время осуществления транзакции;
- Характеристики банкоматов (адрес местонахождения с указанием координат, описание расположения)
- Характеристики банковской карты (платежная система, разделение клиентов на своих и чужих);
- Параметры погоды в момент осуществления транзакции (температура, давление, влажность, облачность).



Показатели региона

Данные о наличии и количестве организаций сферы услуг в районе расположения банкоматов были получены с портала Openstreetmap - импорт данных осуществлен с использованием библиотеки [Overpy](#) и координат банкоматов (долгота, широта).



OpenStreetMap

Метеорологические данные

Данные о погоде в районе местонахождения банкоматов были получены из портала [Worldweatheronline.com](#) – импорт данных осуществлен с использованием библиотек [Request](#), [Json](#) и координат банкоматов (долгота, широта).



Производственный календарь Дании

В качестве сведений о рабочих, выходных и праздничных днях были использованы данные портала [Workingdays.org](#)



В исходный датасет внесены следующие изменения:

- Транзакции сгруппированы до уровня дня;
- Исключены неотработанные операции (операции с ошибкой);
- Добавлены признаки рабочих и нерабочих дней для отчетного дня, а также для предшествующих и будущих дней;
- Добавлены скользящие средние по количеству и объему транзакций по каждой категории за 7,14,30,90 дней;
- Добавлены поля с временными сдвигами (лагами) 1-7 дней;
- Добавлены сведения о погоде за каждый день;
- Добавлены сведения о количестве организаций определенных видов деятельности в радиусе 300м от банкоматов (общественное питание, магазины и аптеки, отели, бары, клубы и др.)

Выбор модели для прогнозирования

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
Light Gradient Boosting Machine	5252.6921	83412289.1175	8902.5979	0.8413	0.4001	0.4557
Extreme Gradient Boosting	5397.4337	84766209.3247	8919.5767	0.8430	0.4163	0.4531
Random Forest Regressor	5808.9595	100536608.3662	9789.1358	0.8097	0.4175	0.4599
Gradient Boosting Regressor	5849.9973	100941744.1371	9838.6728	0.8063	0.4197	0.4729
Extra Trees Regressor	5824.7966	102803892.2350	9874.6435	0.8043	0.4220	0.4793
K Neighbors Regressor	7730.7494	172757202.7273	12914.1724	0.6682	0.4794	0.5881
Orthogonal Matching Pursuit	8129.0053	179529080.7626	13113.5047	0.6571	0.5573	0.8568
Decision Tree Regressor	8377.4940	186612767.0109	13451.7500	0.6385	0.6069	0.5623
AdaBoost Regressor	10696.4801	245709897.8380	15582.2650	0.4949	0.6118	0.8155
Linear Regression	5442.7706	351304717.9125	14939.6672	0.3870	0.4715	0.6387
Ridge Regression	5442.4118	351383274.3866	14940.7807	0.3869	0.4715	0.6387
Bayesian Ridge	5419.2423	358132342.2847	15029.7130	0.3751	0.4710	0.6377
Elastic Net	5186.4032	413316126.0457	15839.5541	0.2766	0.4623	0.6154
Lasso Regression	5181.5075	424804539.3536	15957.6616	0.2585	0.4628	0.6164
Lasso Least Angle Regression	5186.1935	497121272.6792	16702.4123	0.1338	0.4630	0.6178
Dummy Regressor	18768.6513	584222734.2111	23970.6443	-0.1646	0.9844	2.5213
Huber Regressor	5056.3629	1462514846.2077	26307.7710	-1.5203	0.4943	0.7007
Passive Aggressive Regressor	11341.0274	3879546733.5719	44174.5607	-5.7109	0.4607	0.5453
Least Angle Regression	48242081630.2200	198754127620961527136256.0000	199376102457.2481	-339176606292085.3125	2.8329	21625388.1344

Предварительный выбор модели для прогнозирования осуществлен с использованием библиотеки **PyCaret**.

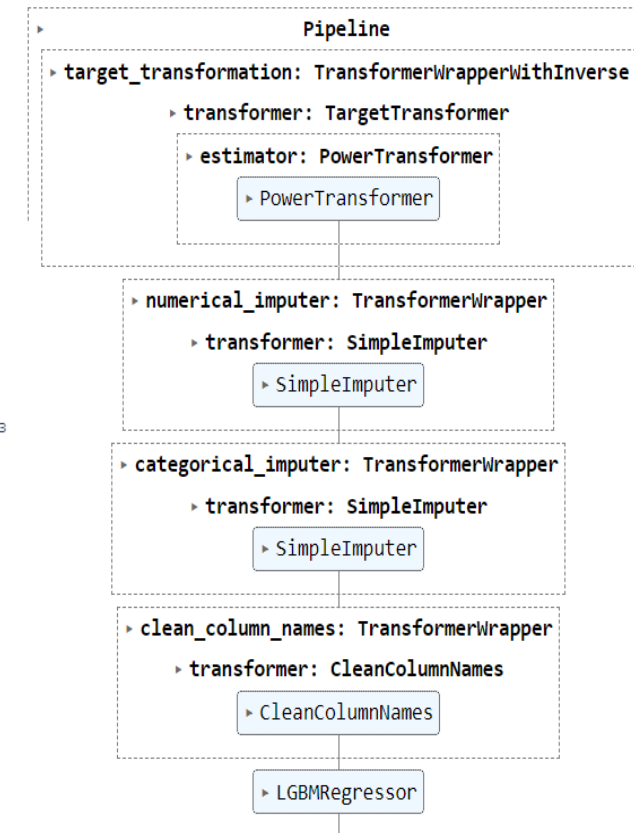
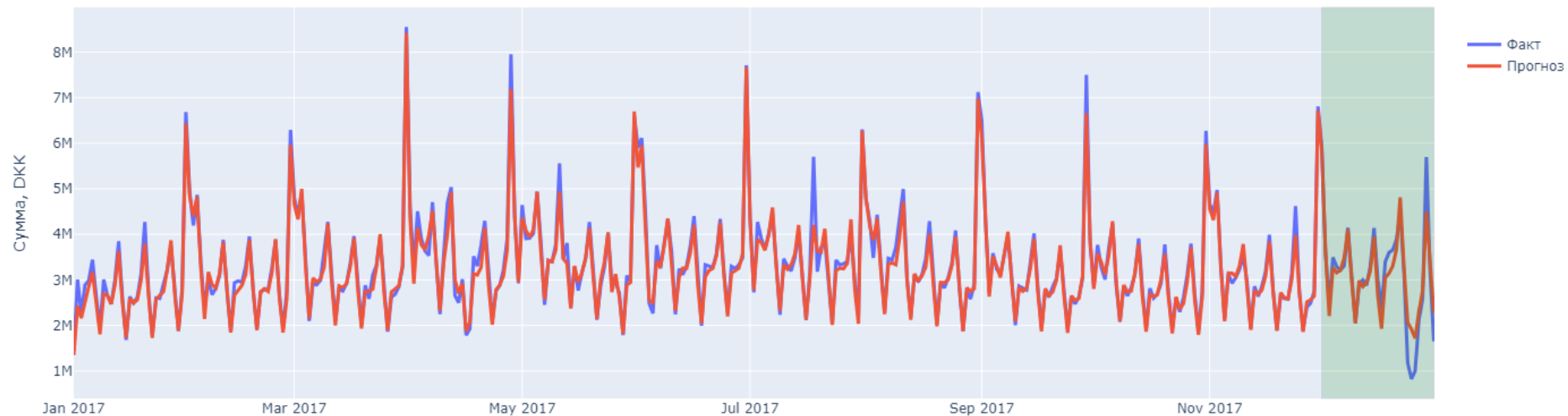
```
train = df[df['month'] < 12]  
test = df[df['month'] == 12]
```

Выбор модели для прогнозирования

Предварительные наилучшие результаты показала модель градиентного бустинга деревьев решений с использованием **LightGBM**.

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
Light Gradient Boosting Machine	5252.6921	83412289.1175	8902.5979	0.8413	0.4001	0.4557

Объем выдач денежных средств через банкоматы (прогноз/факт)



Обзор решения

В качестве решения для прогнозирования объема выдач денежных средств выбрана модель на основе **градиентного бустинга деревьев решений** на базе библиотеки **LightGBM**.



Принцип работы:

LightGBM использует метод градиентного бустинга, при котором модель последовательно обучается на ошибках предыдущих шагов.

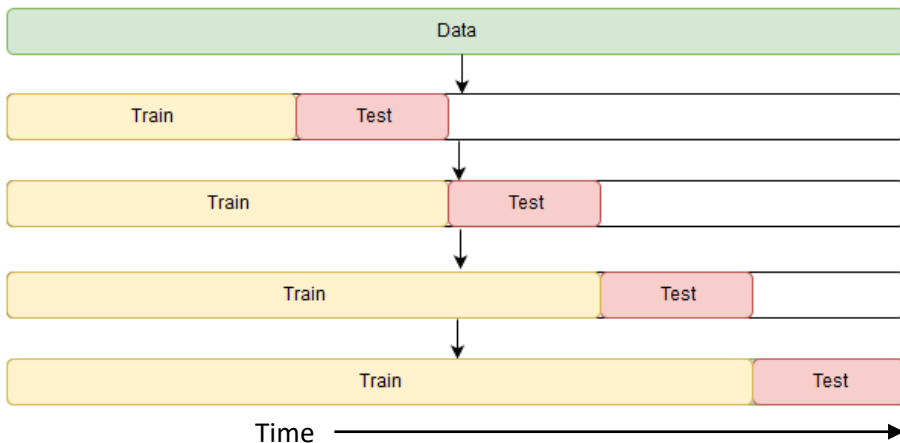
Сравнивая LightGBM с другими библиотеками, такими как XGBoost и CatBoost, можно выделить его уникальность, которая заключается в использовании легковесных деревьев решений. Вместо того чтобы строить дерево по уровням (Level-wise tree growth), LightGBM строит его вертикально (Leaf-wise tree growth), начиная с самых информативных узлов. Этот метод не только обеспечивает более быстрое обучение модели, но и позволяет создавать более компактные и информативные деревья, что особенно важно при работе с большими объемами данных и задачами временных рядов.

Преимущества:

- Эффективность: Быстрое обучение благодаря вертикальному строению деревьев решений.
- Обработка больших данных: Эффективная обработка больших объемов данных без потери качества модели.
- Гибкость настройки: Широкий выбор параметров для оптимальной настройки модели.

Построение модели

```
params = {
    'objective': 'regression',
    'metric': 'mae',
    'boosting_type': 'gbdt',
    'num_leaves': trial.suggest_int('num_leaves', 2, 800),
    'learning_rate': trial.suggest_loguniform('learning_rate', 0.001, 0.75),
    'max_depth': trial.suggest_int('max_depth', 3, 30),
    'n_estimators': trial.suggest_int('n_estimators', 10, 300),
    'feature_fraction': trial.suggest_uniform('feature_fraction', 0.1, 1.0),
    'bagging_fraction': trial.suggest_uniform('bagging_fraction', 0.1, 1.0),
    'bagging_freq': trial.suggest_int('bagging_freq', 1, 10),
    'min_child_samples': trial.suggest_int('min_child_samples', 5, 200),
    'min_child_weight': trial.suggest_loguniform('min_child_weight', 0.1, 10),
    'subsample': trial.suggest_uniform('subsample', 0.1, 1.0),
    'colsample_bytree': trial.suggest_uniform('colsample_bytree', 0.1, 1.0),
    'reg_alpha': trial.suggest_loguniform('reg_alpha', 1e-9, 10.0),
    'reg_lambda': trial.suggest_loguniform('reg_lambda', 1e-9, 10.0),
    'bagging_seed': trial.suggest_int('bagging_seed', 1, 100),
    'feature_fraction_seed': trial.suggest_int('feature_fraction_seed', 1, 100),
    'min_split_gain': trial.suggest_loguniform('min_split_gain', 1e-9, 1.0),
    'min_data_in_leaf': trial.suggest_int('min_data_in_leaf', 1, 100),
    'max_bin': trial.suggest_int('max_bin', 32, 512),
    'scale_pos_weight': trial.suggest_uniform('scale_pos_weight', 0.1, 10.0),
    'min_sum_hessian_in_leaf': trial.suggest_loguniform('min_sum_hessian_in_leaf', 1e-5, 1e-2),
    'verbose': -1
}
```



- Подбор гиперпараметров осуществлен при помощи библиотеки [Optuna](#).
 - `num_leaves` - количество листьев в дереве (от 2 до 800);
 - `learning_rate` - скорость обучения (от 0.001 до 0.75).
 - `max_depth` - максимальная глубина дерева (от 3 до 30)
 - `n_estimators` - количество деревьев в модели (от 10 до 300)
 - `feature_fraction` - доля признаков, используемых для обучения каждого дерева (от 0.1 до 1.0)
- Кросс-валидация осуществлена с помощью класса [TimeSeriesSplit](#) библиотеки [Scikit-learn](#).

Оценка работы модели: метрики

Объем выдач денежных средств через банкоматы (прогноз/факт)

```
train = df[df['month'] < 12]
test = df[df['month'] == 12]
```



До подбора гиперпараметров

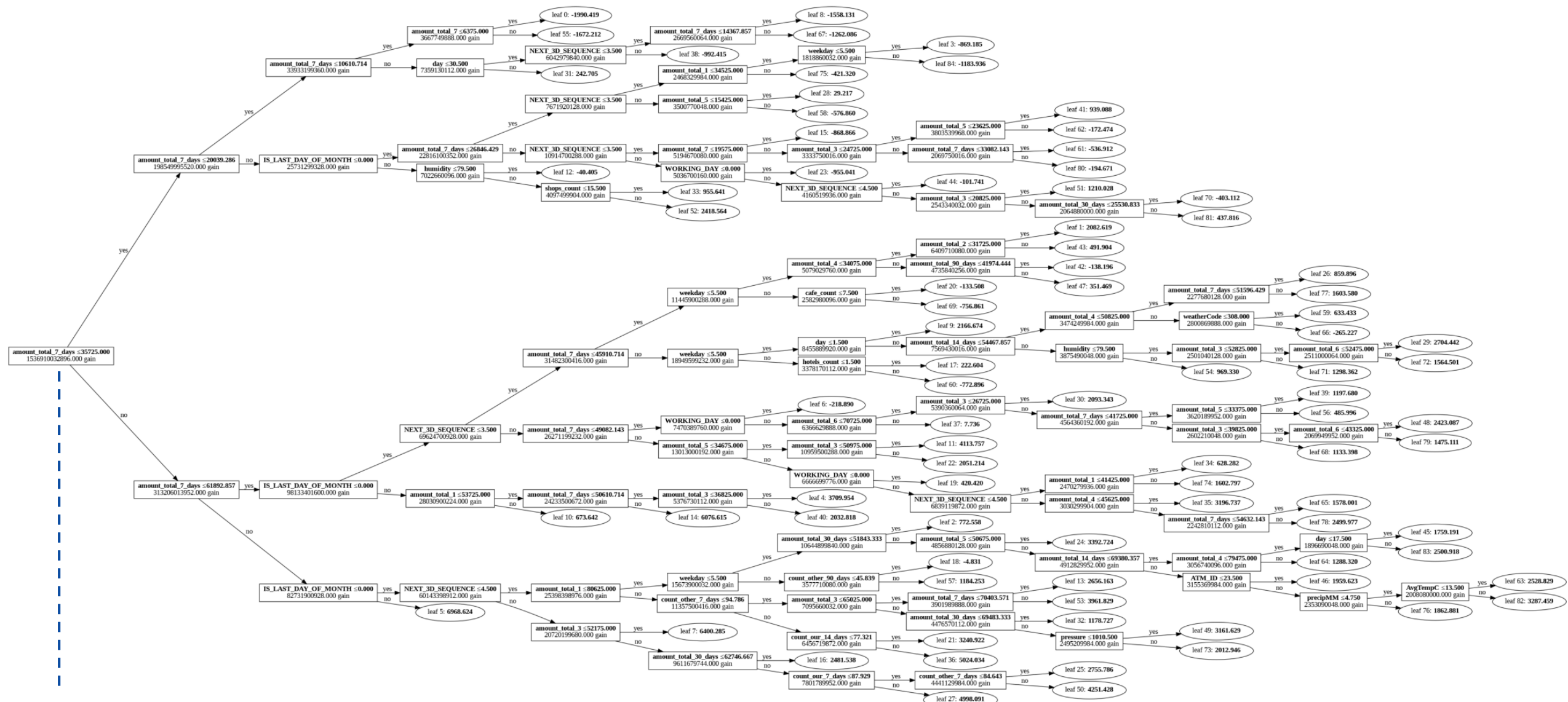
MAE on Test Set: 5 253
MSE on Test Set: 83 412 289
R2 on Test Set: 0,8413
MAPE on Test Set: 45,57%

После подбора гиперпараметров

4 923
69 980 682
0.8870
38,57%

```
LGBMRegressor
LGBMRegressor(bagging_fraction=0.7860989964429168, bagging_freq=1,
               bagging_seed=97, colsample_bytree=0.7365098502521354,
               feature_fraction=0.9767241244471809, feature_fraction_seed=94,
               learning_rate=0.1419152614178975, max_bin=486, max_depth=27,
               min_child_samples=113, min_child_weight=0.3884994278962764,
               min_data_in_leaf=61, min_split_gain=4.374733307142097e-09,
               min_sum_hessian_in_leaf=7.767940168247641e-05, n_estimators=295,
               num_leaves=85, reg_alpha=6.01135913516868,
               reg_lambda=3.9418432380843385e-08,
               scale_pos_weight=7.610046719097241,
               subsample=0.15449378919485532)
```


Оценка работы модели: пример дерева решений



amount_total_7_days – средний объем выдач денежных средств за последние 7 дней

Заключение

1. Разработанная модель показала следующие метрики, что для данной задачи является высоким результатом работы:
 - MAE: 4 923
 - R2: 0.89
 - MAPE: 38,6%
2. Модель позволяет спрогнозировать объем выдачи наличных денежных средства и, как следствие, определить требуемый резерв средств для подкрепления банкоматной сети.
3. Все поставленные задачи и цели в рамках данной выпускной квалификационной работы были выполнены.
 - Произведены анализ и обработка исходных данных по выдаче денежных средств через банкоматы;
 - Осуществлен отбор наилучшей модели прогноза;
 - Построена модель градиентного бустинга деревьев решений с использованием библиотеки LightGBM;
 - Осуществлены настройка и подбор наилучших гиперпараметров модели;
 - Произведена оценка полученных результатов.

Перспективы для дальнейшей работы:

1. В рамках ВКР был использован ограниченный набор исходных данных, что не позволяет в полной мере раскрыть весь потенциал модели в определении необходимого объема и частоты подкрепления банкоматной сети с учетом транспортных расходов и режима работы инкассаторских служб. В текущем виде разработанную модель можно использовать при подкреплении банкоматов, расположенных в банковских отделениях, где инкассация осуществляется силами операционных работников без привлечения инкассаторской службы.
2. С целью дальнейшего развития модели можно выделить несколько аспектов, которые могут быть дополнительно учтены для улучшения функционала и точности модели: учет транспортных расходов, интеграция финансовых и клиентских данных (стоимость фондирования, информация о зарплатных проектах и клиентской базе, сведения о темпах кредитования и привлечения средств), учет данных о фактических остатках денежных средств, учет сведений об отказах в выдаче денежных средств по причине недостатка средств в банкомате, доработка модели с целью возможного применения для банкоматов с технологией Cash-Recycling.