In [ ]:

```
!pip install LightAutoML -q
```

```
 Preparing metadata (setup.py) ... done
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 412.5/412.5 kB 3.7 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 216.1/216.1 kB 4.1 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.2/99.2 MB 7.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 309.5/309.5 kB 9.0 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.8/9.8 MB 40.1 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 223.6/223.6 MB 6.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.5/64.5 kB 3.8 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 404.7/404.7 kB 22.3 MB/s eta 0:00:0
0
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 107.8/107.8 kB 6.9 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.7/7.7 MB 71.6 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 121.1/121.1 kB 7.6 MB/s eta 0:00:00
 Building wheel for json2html (setup.py) ... done
```

EDA

In [ ]:

```python
import gc
import warnings

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

warnings.filterwarnings("ignore")

# Загрузка данных
train_data = pd.read_csv("train.csv")
test_data = pd.read_csv("test.csv")

# Базовый анализ
print("\n Размеры датасетов ")
print(f"Train: {train_data.shape}")
print(f"Test: {test_data.shape}")

print("\n Типы данных ")
print(train_data.dtypes)

print("\n Пропущенные значения ")
print(train_data.isnull().sum())

print("\n Статистики числовых признаков")
print(train_data.describe())
```

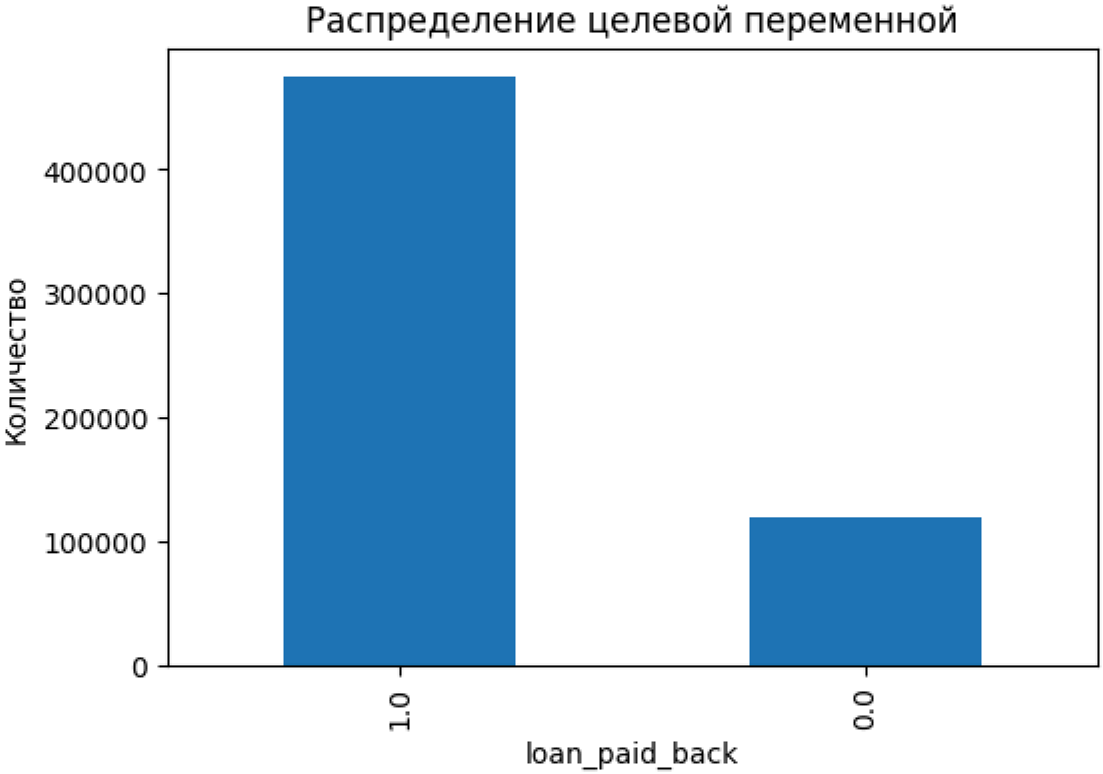
```
 Размеры датасетов
Train: (593994, 13)
Test: (254569, 12)

 Типы данных
id                        int64
annual_income           float64
debt_to_income_ratio    float64
credit_score              int64
loan_amount             float64
interest_rate           float64
gender                   object
marital_status           object
education_level          object
employment_status        object
loan_purpose             object
grade_subgrade           object
loan_paid_back          float64
dtype: object

 Пропущенные значения
id                      0
annual_income           0
debt_to_income_ratio    0
credit_score            0
loan_amount             0
interest_rate           0
gender                  0
marital_status          0
education_level         0
employment_status       0
loan_purpose            0
grade_subgrade          0
loan_paid_back          0
dtype: int64

 Статистики числовых признаков
                  id  annual_income  debt_to_income_ratio   credit_score  \
count  593994.000000  593994.000000         593994.000000  593994.000000
mean   296996.500000   48212.202976              0.120696     680.916009
std    171471.442235   26711.942078              0.068573      55.424956
min         0.000000    6002.430000              0.011000     395.000000
25%    148498.250000   27934.400000              0.072000     646.000000
50%    296996.500000   46557.680000              0.096000     682.000000
75%    445494.750000   60981.320000              0.156000     719.000000
max    593993.000000  393381.740000              0.627000     849.000000

         loan_amount  interest_rate  loan_paid_back
count  593994.000000  593994.000000   593994.000000
mean    15020.297629      12.356345        0.798820
std      6926.530568       2.008959        0.400883
min       500.090000       3.200000        0.000000
25%     10279.620000      10.990000        1.000000
50%     15000.220000      12.370000        1.000000
```

```
75%      18858.580000      13.680000        1.000000
max      48959.950000      20.990000        1.000000
```

In [ ]:

```
# Анализ целевой переменной
print("\n Распределение целевой переменной")
print(train_data["loan_paid_back"].value_counts(normalize=True))

plt.figure(figsize=(6, 4))
train_data["loan_paid_back"].value_counts().plot(kind="bar")
plt.title("Распределение целевой переменной")
plt.xlabel("loan_paid_back")
plt.ylabel("Количество")
plt.show()
```

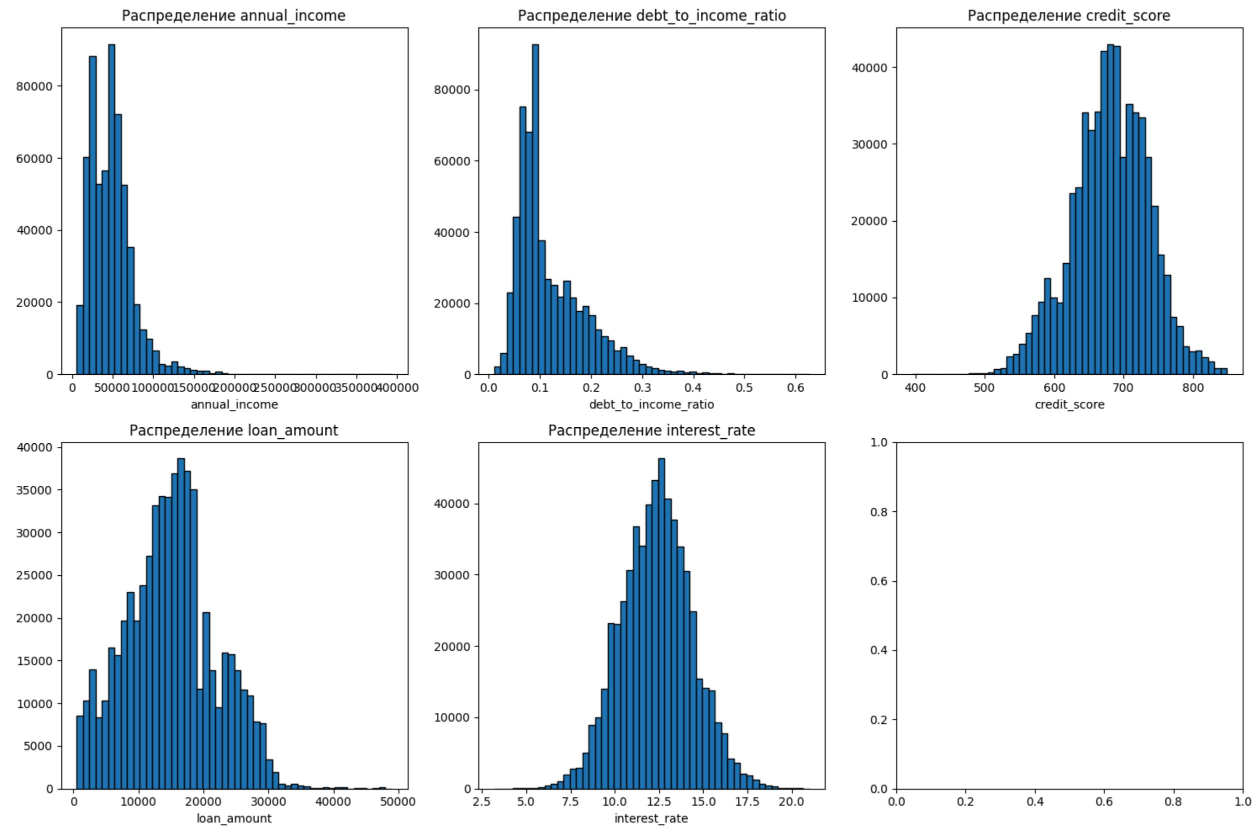
```
 Распределение целевой переменной
loan_paid_back
1.0    0.79882
0.0    0.20118
Name: proportion, dtype: float64
```



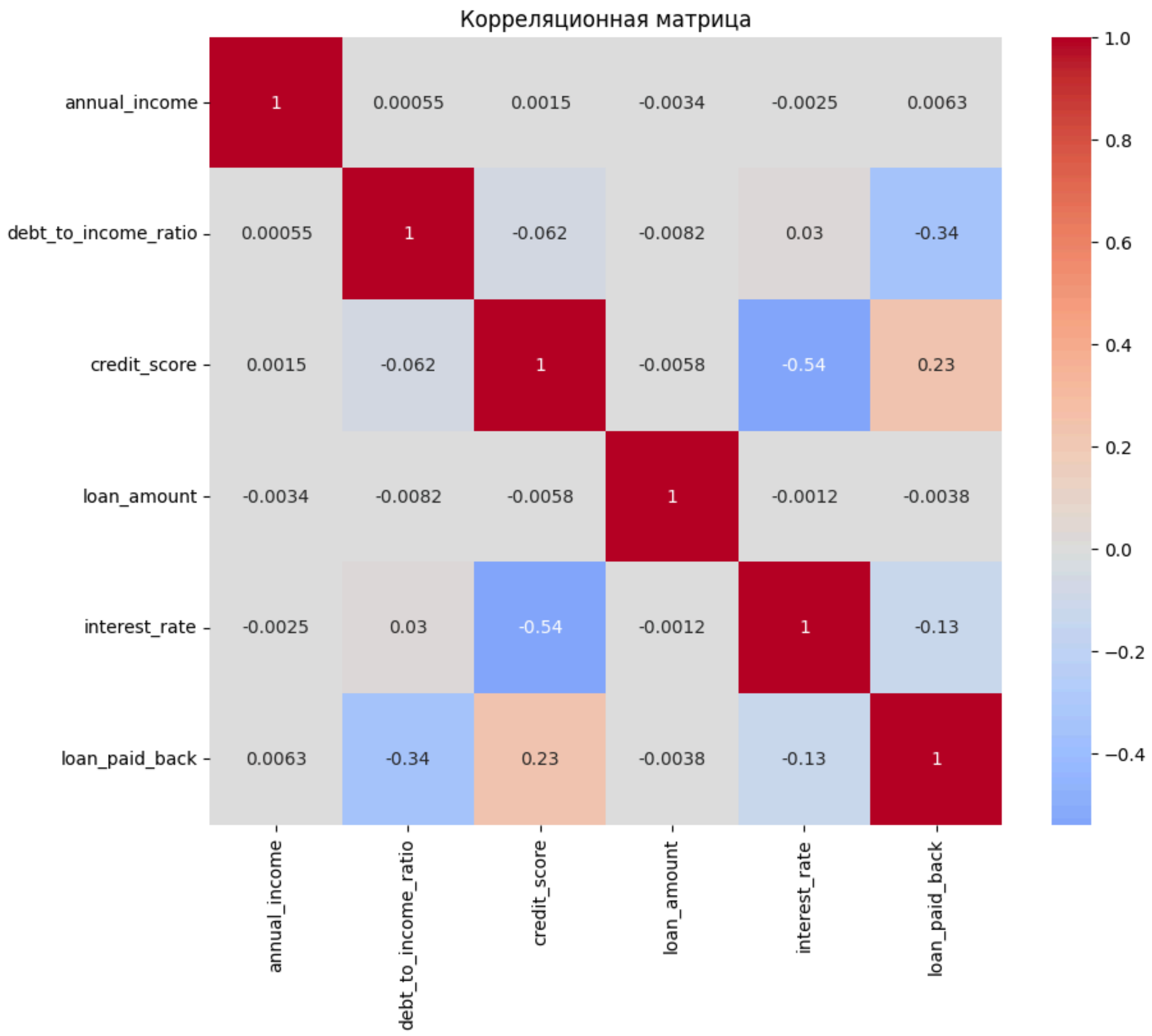
In [ ]:

```
# Определяем признаки
numeric_cols = [
    "annual_income",
    "debt_to_income_ratio",
    "credit_score",
    "loan_amount",
    "interest_rate",
]
categorical_cols = [
    "gender",
    "marital_status",
    "education_level",
    "employment_status",
    "loan_purpose",
    "grade_subgrade",
]

# Распределения числовых признаков
print("\n Анализ числовых признаков ")
fig, axes = plt.subplots(2, 3, figsize=(15, 10))
axes = axes.ravel()
for idx, col in enumerate(numeric_cols):
    axes[idx].hist(train_data[col], bins=50, edgecolor="black")
    axes[idx].set_title(f"Распределение {col}")
    axes[idx].set_xlabel(col)
plt.tight_layout()
plt.show()

# Корреляция с целевой переменной
print("\n Корреляция числовых признаков с целевой ")
correlations = (
    train_data[numeric_cols + ["loan_paid_back"]]
    .corr()["loan_paid_back"]
    .sort_values(ascending=False)
)
print(correlations)

# Тепловая карта корреляций
plt.figure(figsize=(10, 8))
sns.heatmap(
    train_data[numeric_cols + ["loan_paid_back"]].corr(), annot=True, cmap="co
)
plt.title("Корреляционная матрица")
plt.show()
```

```
Анализ числовых признаков
```



```
 Корреляция числовых признаков с целевой
loan_paid_back          1.000000
credit_score            0.234560
annual_income           0.006326
loan_amount            -0.003762
interest_rate          -0.131184
debt_to_income_ratio   -0.335680
Name: loan_paid_back, dtype: float64
```



Выводы

- debt_to_income_ratio: -0.336 (сильная отрицательная)
- credit_score: 0.235 (умеренная положительная)
- annual_income и loan_amount: слабая корреляция

In [ ]:

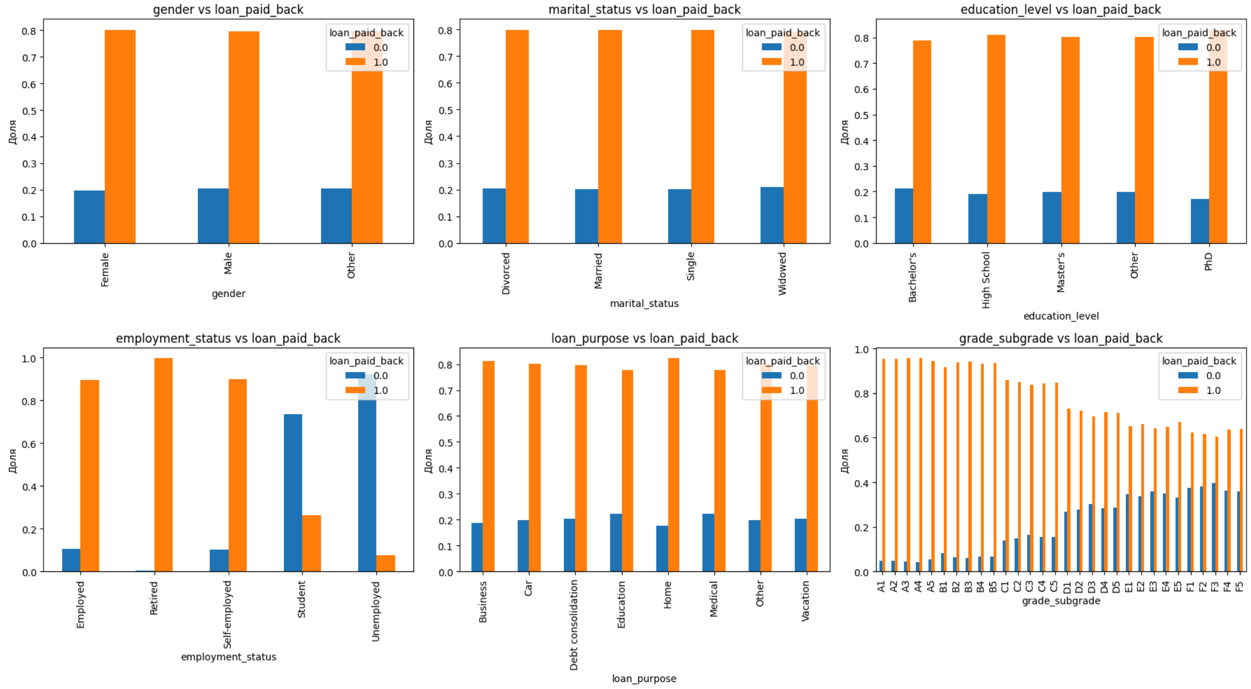
```
# Анализ категориальных признаков
print("\n Анализ категориальных признаков ")
for col in categorical_cols:
    print(f"\n--- {col} ---")
    print(train_data[col].value_counts())

    print(f"\nСредняя вероятность возврата кредита по {col}:")
    probs = train_data.groupby(col)["loan_paid_back"].mean().sort_values(ascer
    print(probs)

# Визуализация категориальных признаков
```

```
fig, axes = plt.subplots(2, 3, figsize=(18, 10))
axes = axes.ravel()
for idx, col in enumerate(categorical_cols):
    pd.crosstab(train_data[col], train_data["loan_paid_back"], normalize="inde
        kind="bar", ax=axes[idx]
    )
    axes[idx].set_title(f"{col} vs loan_paid_back")
    axes[idx].set_xlabel(col)
    axes[idx].set_ylabel("Доля")
    axes[idx].legend(title="loan_paid_back")
plt.tight_layout()
plt.show()
```

```
 Анализ категориальных признаков

--- gender ---
gender
Female    306175
Male      284091
Other       3728
Name: count, dtype: int64

Средняя вероятность возврата кредита по gender:
gender
Female    0.801708
Male      0.795752
Other     0.795333
Name: loan_paid_back, dtype: float64

--- marital_status ---
marital_status
Single      288843
Married     277239
Divorced     21312
Widowed       6600
Name: count, dtype: int64

Средняя вероятность возврата кредита по marital_status:
marital_status
Married     0.799144
Single      0.798873
Divorced    0.796640
Widowed     0.789848
Name: loan_paid_back, dtype: float64

--- education_level ---
education_level
Bachelor's     279606
High School    183592
Master's        93097
Other           26677
PhD             11022
Name: count, dtype: int64

Средняя вероятность возврата кредита по education_level:
education_level
PhD            0.830067
High School    0.809698
Other          0.802789
Master's       0.802346
Bachelor's     0.788892
Name: loan_paid_back, dtype: float64

--- employment_status ---
employment_status
Employed         450645
Unemployed        62485
```

```
Self-employed     52480
Retired           16453
Student           11931
Name: count, dtype: int64

Средняя вероятность возврата кредита по employment_status:
employment_status
Retired          0.997204
Self-employed    0.898457
Employed         0.894145
Student          0.263515
Unemployed       0.077619
Name: loan_paid_back, dtype: float64

--- loan_purpose ---
loan_purpose
Debt consolidation    324695
Other                  63874
Car                    58108
Home                   44118
Education              36641
Business               35303
Medical                22806
Vacation                8449
Name: count, dtype: int64

Средняя вероятность возврата кредита по loan_purpose:
loan_purpose
Home                  0.823224
Business              0.813104
Other                 0.802377
Car                   0.800630
Debt consolidation    0.796911
Vacation              0.796071
Medical               0.778085
Education             0.777053
Name: loan_paid_back, dtype: float64

--- grade_subgrade ---
grade_subgrade
C3    58695
C4    55957
C2    54443
C1    53363
C5    53317
D1    37029
D3    36694
D4    35097
D2    34432
D5    32101
B2    15167
B1    14344
B5    13937
B3    13926
```

```
B4    13877
E4     8036
E3     7075
E1     6891
E2     6372
E5     6084
F5     5947
F4     5535
F1     5534
F2     5203
F3     5082
A5     2471
A3     2066
A2     2018
A4     1701
A1     1600
Name: count, dtype: int64

Средняя вероятность возврата кредита по grade_subgrade:
grade_subgrade
A4    0.957084
A3    0.955470
A2    0.952924
A1    0.952500
A5    0.944962
B3    0.940040
B2    0.937430
B5    0.934204
B4    0.931758
B1    0.916341
C1    0.860090
C2    0.851165
C5    0.846259
C4    0.843987
C3    0.836000
D1    0.731886
D2    0.720957
D4    0.714733
D5    0.713000
D3    0.695972
E5    0.669461
E2    0.662743
E1    0.652010
E4    0.649577
E3    0.641837
F5    0.639314
F4    0.637037
F1    0.624503
F2    0.617721
F3    0.604093
Name: loan_paid_back, dtype: float64
```



Ключевые находки из EDA

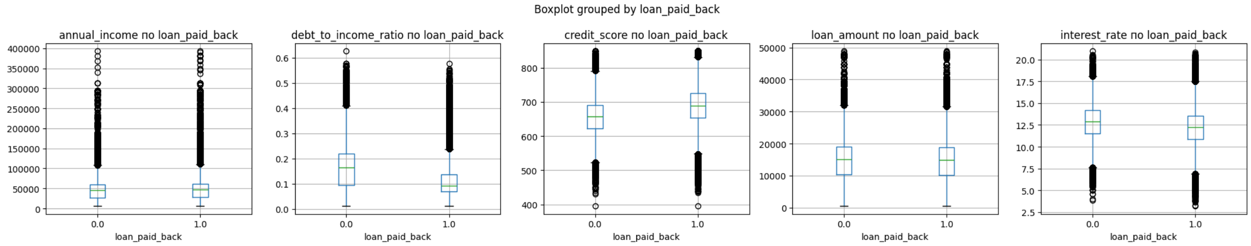
1. `employment_status` **— самый сильный признак**
   - Retired: **99.7%**
   - Unemployed: **7.8%** Огромная разница в вероятности возврата
2. `grade_subgrade` **— чёткая градация внутри буквенных классов**
   - Пример: A1/A5 (**95.2%/94.5%**)
   - Внутри каждой буквы есть выраженный паттерн по цифре.
3. **Числовые признаки слабокоррелированы**
   - Значит, модель выигрывает от *взаимодействий* (feature crosses).
4. `education_level`
   - PhD: **83%**
   - Bachelor’s: **78.9%** Признак информативный, но не доминирующий.
5. `loan_purpose`
   - Home: **82.3%**
   - Education: **77.7%** Существенные различия между

категориями.

In [ ]:

```
# Анализ выбросов
print("\n Выбросы (IQR метод) ")
for col in numeric_cols:
    Q1 = train_data[col].quantile(0.25)
    Q3 = train_data[col].quantile(0.75)
    IQR = Q3 - Q1
    outliers = ((train_data[col] < (Q1 - 1.5 * IQR)) | (train_data[col] > (Q3
    print(f"{col}: {outliers} выбросов ({outliers/len(train_data)*100:.2f}%)")

fig, axes = plt.subplots(1, 5, figsize=(20, 4))
for idx, col in enumerate(numeric_cols):
    train_data.boxplot(column=col, by="loan_paid_back", ax=axes[idx])
    axes[idx].set_title(f"{col} по loan_paid_back")
plt.tight_layout()
plt.show()
```

```
 Выбросы (IQR метод)
annual_income: 15917 выбросов (2.68%)
debt_to_income_ratio: 17556 выбросов (2.96%)
credit_score: 5901 выбросов (0.99%)
loan_amount: 2902 выбросов (0.49%)
interest_rate: 5136 выбросов (0.86%)
```



FEATURE ENGINEERING

Обоснования для создаваемых признаков

1. Извлечение `grade_digit` из `grade_subgrade`

- Внутри каждой буквенной категории (A-F) есть выраженная градация по цифре.
- Пример: A1/A5 соответствует снижению вероятности возврата (**95.2%/94.5%**).
- Цифра несёт собственную информативность и должна использоваться отдельно.

2. Взаимодействия числовых признаков

Корреляции между числовыми признаками низкие, поэтому отдельные фичи малоинформативны.
Комбинации улучшают модель за счёт доменных зависимостей.

- **income_to_loan_ratio** - способность заёмщика погасить займ.
- **loan_to_income_ratio** - относительный размер кредита.
- **debt_times_rate** - комплексная долговая нагрузка (долги x ставка).
- **income_times_credit** - интегральная «надёжность» (доход x кредитный рейтинг).
- **monthly_payment_estimate / payment_to_income** - реальная месячная нагрузка.

3. Усиление `employment_status` (ключевой признак)

- Разница между категориями экстремально высока:
    - Retired: **99.7%**
    - Unemployed: **7.8%**
- Создаются статистики по числовым признакам внутри каждой категории занятости:
    - Среднее
    - Отклонение
    - Процент отклонения
      Это позволяет выделить финансовые профили для каждой группы занятости.

4. Групповая статистика по категориальным признакам

Используется стандартная техника для извлечения скрытых паттернов внутри категорий.

По `grade_subgrade`: сильная связь (60–95%)

- mean
- std
- разница от среднего

По `loan_purpose`: (77–82%)

- mean и разница для ключевых числовых признаков

По `education_level` : (78–83%)

- mean для дохода и кредитного рейтинга

5. Комбинации категорий

Создаются взаимодействия между важными категориальными переменными.

- `employment_status + education_level`
- `employment_status + marital_status`
- `grade_subgrade + loan_purpose`
- `education_level + loan_purpose`

Цель - уловить различия внутри комбинаций (например, «Retired + PhD» не то же самое, что «Retired + High School»).

6. Бинарные флаги

- **high_credit_score ≥ 720** - высокий кредитный рейтинг.
- **low_debt_ratio ≤ 0.1** - низкая долговая нагрузка.
- **high_income ≥ 60,000** - устойчивый доход.
- **small_loan ≤ 10,000** - низкий размер кредита.
- **low_interest ≤ 11%** - выгодная ставка.

Флаги помогают модели выделять важные пороговые состояния.

In [ ]:

```
# Перезагружаем данные для FE
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")

# Сохраняем ID и target
test_ids = test["id"].copy()
target = train["loan_paid_back"].copy()

# Удаляем служебные колонки
train = train.drop(columns=["id", "loan_paid_back"])
test = test.drop(columns=["id"])

NUMS = ["annual_income", "debt_to_income_ratio", "credit_score", "loan_amount"
CATS = [
    "gender",
    "marital_status",
    "education_level",
    "employment_status",
    "loan_purpose",
    "grade_subgrade",
]

print("\nСоздаём признаки на основе инсайтов из EDA")

# ПРИЗНАК 1: grade_digit
print("\nПРИЗНАК 1: Извлечение цифры из grade_subgrade")
train["grade_digit"] = train["grade_subgrade"].str[1].astype("int8")
test["grade_digit"] = test["grade_subgrade"].str[1].astype("int8")

train["grade_letter"] = train["grade_subgrade"].str[0]
test["grade_letter"] = test["grade_subgrade"].str[0]

# ПРИЗНАК 2: Числовые взаимодействия
print("\nПРИЗНАК 2: Взаимодействия числовых признаков")

# Платёжеспособность
train["income_to_loan_ratio"] = train["annual_income"] / (train["loan_amount"]
test["income_to_loan_ratio"] = test["annual_income"] / (test["loan_amount"] +
print("- income_to_loan_ratio: способность погасить займ")

# Долговая нагрузка
train["loan_to_income_ratio"] = train["loan_amount"] / (train["annual_income"]
test["loan_to_income_ratio"] = test["loan_amount"] / (test["annual_income"] +
print("- loan_to_income_ratio: размер займа относительно дохода")

# Совокупный риск
train["debt_times_rate"] = train["debt_to_income_ratio"] * train["interest_rat
test["debt_times_rate"] = test["debt_to_income_ratio"] * test["interest_rate"]
print("- debt_times_rate: комплексная оценка долговой нагрузки")

# Кредитоспособность
train["income_times_credit"] = train["annual_income"] * train["credit_score"]
test["income_times_credit"] = test["annual_income"] * test["credit_score"] / 1
print("- income_times_credit: общая финансовая надёжность")

# Месячный платёж и нагрузка
train["monthly_payment_estimate"] = (train["loan_amount"] * train["interest_ra
test["monthly_payment_estimate"] = (test["loan_amount"] * test["interest_rate"

train["payment_to_income"] = train["monthly_payment_estimate"] / (train["annua
test["payment_to_income"] = test["monthly_payment_estimate"] / (test["annual_i
print("- payment_to_income: реальная месячная нагрузка на бюджет")

# ПРИЗНАК 3: employment_status
print("\nПРИЗНАК 3: Фокус на employment_status")

# Групповая статистика по employment
for num_col in NUMS:
    group_mean = train.groupby("employment_status")[num_col].mean()
    train[f"{num_col}_mean_by_employment"] = train["employment_status"].map(gr
    test[f"{num_col}_mean_by_employment"] = test["employment_status"].map(grou
    train[f"{num_col}_diff_from_employment"] = (
        train[num_col] - train[f"{num_col}_mean_by_employment"]
    )
    test[f"{num_col}_diff_from_employment"] = test[num_col] - test[f"{num_col}

    train[f"{num_col}_pct_from_employment"] = train[f"{num_col}_diff_from_empl
        train[f"{num_col}_mean_by_employment"] + 1
    )
    test[f"{num_col}_pct_from_employment"] = test[f"{num_col}_diff_from_employ
        test[f"{num_col}_mean_by_employment"] + 1
    )

print("   - Созданы среднее, отклонение и % отклонение для каждого числового г

# Комбинации employment с категориями
for cat_col in ["education_level", "marital_status", "loan_purpose"]:
    name = f"employment_{cat_col}"
    train[name] = train["employment_status"].astype(str) + "_" + train[cat_col
    test[name] = test["employment_status"].astype(str) + "_" + test[cat_col].a
print("   - Комбинации employment с education, marital_status, loan_purpose")

# ПРИЗНАК 4: Группировки по категориям
print("\nПРИЗНАК 4: Групповая статистика по категориям")
print("   Обоснование: стандартная практика для извлечения паттернов")

# По grade_subgrade (сильная связь: 60-95%)
for num_col in NUMS:
    group_mean = train.groupby("grade_subgrade")[num_col].mean()
    group_std = train.groupby("grade_subgrade")[num_col].std()

    train[f"{num_col}_mean_by_grade"] = train["grade_subgrade"].map(group_mean
    test[f"{num_col}_mean_by_grade"] = test["grade_subgrade"].map(group_mean)

    train[f"{num_col}_std_by_grade"] = train["grade_subgrade"].map(group_std)
    test[f"{num_col}_std_by_grade"] = test["grade_subgrade"].map(group_std)

    train[f"{num_col}_diff_from_grade"] = train[num_col] - train[f"{num_col}_m
    test[f"{num_col}_diff_from_grade"] = test[num_col] - test[f"{num_col}_mean

print("   - Статистика по grade_subgrade (A-F): mean, std, diff")

# По loan_purpose (77-82%)
for num_col in ["annual_income", "credit_score", "debt_to_income_ratio"]:
    group_mean = train.groupby("loan_purpose")[num_col].mean()
    train[f"{num_col}_mean_by_purpose"] = train["loan_purpose"].map(group_mean
    test[f"{num_col}_mean_by_purpose"] = test["loan_purpose"].map(group_mean)

    train[f"{num_col}_diff_from_purpose"] = train[num_col] - train[f"{num_col}
    test[f"{num_col}_diff_from_purpose"] = test[num_col] - test[f"{num_col}_me

print("- Статистика по loan_purpose")

# По education_level (78-83%)
```

```
for num_col in ["annual_income", "credit_score"]:
    group_mean = train.groupby("education_level")[num_col].mean()
    train[f"{num_col}_mean_by_education"] = train["education_level"].map(group
    test[f"{num_col}_mean_by_education"] = test["education_level"].map(group_m

print("- Статистика по education_level")

# ПРИЗНАК 5: Комбинации категорий
print("\nПРИЗНАК 5: Комбинации категориальных признаков")

important_cat_pairs = [
    ("employment_status", "education_level"),
    ("employment_status", "marital_status"),
    ("grade_subgrade", "loan_purpose"),
    ("education_level", "loan_purpose"),
]

for col1, col2 in important_cat_pairs:
    name = f"{col1}_{col2}_combo"
    train[name] = train[col1].astype(str) + "_" + train[col2].astype(str)
    test[name] = test[col1].astype(str) + "_" + test[col2].astype(str)

print(f"- Создано {len(important_cat_pairs)} комбинаций")


# ПРИЗНАК 6: Флаги
print("\nПРИЗНАК 6: Бинарные флаги")

train["high_credit_score"] = (train["credit_score"] >= 720).astype("int8")
test["high_credit_score"] = (test["credit_score"] >= 720).astype("int8")

train["low_debt_ratio"] = (train["debt_to_income_ratio"] <= 0.1).astype("int8"
test["low_debt_ratio"] = (test["debt_to_income_ratio"] <= 0.1).astype("int8")

train["high_income"] = (train["annual_income"] >= 60000).astype("int8")
test["high_income"] = (test["annual_income"] >= 60000).astype("int8")

train["small_loan"] = (train["loan_amount"] <= 10000).astype("int8")
test["small_loan"] = (test["loan_amount"] <= 10000).astype("int8")

train["low_interest"] = (train["interest_rate"] <= 11).astype("int8")
test["low_interest"] = (test["interest_rate"] <= 11).astype("int8")

print("\nFeature Engineering завершён!")
print("Было признаков: 11")
print(f"Стало признаков: {train.shape[1]}")
print(f"Создано новых: {train.shape[1] - 11}")

# Сохраняем для последующего использования
train_fe = train.copy()
test_fe = test.copy()
```

```
Создаём признаки на основе инсайтов из EDA

ПРИЗНАК 1: Извлечение цифры из grade_subgrade

ПРИЗНАК 2: Взаимодействия числовых признаков
- income_to_loan_ratio: способность погасить займ
- loan_to_income_ratio: размер займа относительно дохода
- debt_times_rate: комплексная оценка долговой нагрузки
- income_times_credit: общая финансовая надёжность
- payment_to_income: реальная месячная нагрузка на бюджет

ПРИЗНАК 3: Фокус на employment_status
   - Созданы среднее, отклонение и % отклонение для каждого числового признака
   - Комбинации employment с education, marital_status, loan_purpose

ПРИЗНАК 4: Групповая статистика по категориям
   Обоснование: стандартная практика для извлечения паттернов
   - Статистика по grade_subgrade (A-F): mean, std, diff
- Статистика по loan_purpose
- Статистика по education_level

ПРИЗНАК 5: Комбинации категориальных признаков
- Создано 4 комбинаций

ПРИЗНАК 6: Бинарные флаги

Feature Engineering завершён!
Было признаков: 11
Стало признаков: 69
Создано новых: 58
```

LightAutoML (baseline)

In [ ]:

```python
from lightautoml.automl.presets.tabular_presets import TabularAutoML
from lightautoml.tasks import Task

# Подготовка данных для AutoML
X_train_automl = pd.read_csv("train.csv")
X_test_automl = pd.read_csv("test.csv")

test_ids = X_test_automl["id"].copy()
target = X_train_automl["loan_paid_back"].copy()

# Удаляем служебные колонки
train = X_train_automl.drop(columns=["id", "loan_paid_back"])
test = X_test_automl.drop(columns=["id"])

# AutoML работает лучше с исходными данными
X_train_automl = X_train_automl.assign(loan_paid_back=target)

print("\nОбучение LightAutoML")
```

```python
task = Task("binary")
automl = TabularAutoML(
    task=task,
    timeout=3600,
    cpu_limit=4,
    reader_params={
        "n_jobs": 4,
        "cv": 5,
        "random_state": 42,
    },
)

# Обучение
oof_pred_automl = automl.fit_predict(X_train_automl, roles={"target": "loan_pa

# Предсказание
test_pred_automl = automl.predict(X_test_automl)
pred_proba_automl = test_pred_automl.data.flatten()

from sklearn.metrics import roc_auc_score

automl_cv_auc = roc_auc_score(target, oof_pred_automl.data.flatten())

print(f"\nLightAutoML CV AUC: {automl_cv_auc:.5f}")

# Сохранение submission
submission = pd.DataFrame({"id": test_ids, "loan_paid_back": pred_proba_automl

submission.to_csv("submission_automl.csv", index=False)

print("\nФайл submission.csv успешно сохранён!")
print(submission.head())
```

```
INFO:lightautoml.automl.presets.base:Stdout logging level is ERROR.
INFO:lightautoml.automl.presets.base:Task: binary

INFO:lightautoml.automl.presets.base:Start automl preset with listed constraint
s:
INFO:lightautoml.automl.presets.base:- time: 3600.00 seconds
INFO:lightautoml.automl.presets.base:- CPU: 4 cores
INFO:lightautoml.automl.presets.base:- memory: 16 GB

INFO:lightautoml.reader.base:Train data shape: (593994, 13)
```

```
Обучение LightAutoML
```

```
INFO3:lightautoml.reader.base:Feats was rejected during automatic roles guess: []
INFO:lightautoml.automl.base:Layer 1 train process start. Time left 3588.06 secs
INFO:lightautoml.ml_algo.base:Start fitting Lvl_0_Pipe_0_Mod_0_LinearL2 ...
DEBUG:lightautoml.ml_algo.base:Training params: {'tol': 1e-06, 'max_iter': 100, 'cs': [1e-05, 5e-05, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000], 'early_stopping': 2, 'categorical_idx': [0, 1, 2, 3], 'embed_sizes': array([ 6, 31,  9,  5], dtype=int32), 'data_size': 22}
INFO2:lightautoml.ml_algo.base:===== Start working with fold 0 for Lvl_0_Pipe_0_Mod_0_LinearL2 =====
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 1e-05 score = 0.920037407971417
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 5e-05 score = 0.9211852449075897
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0001 score = 0.9214482556460268
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0005 score = 0.9217867857838377
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.001 score = 0.9218322542958135
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.005 score = 0.9218013839950785
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.01 score = 0.9218013839950785
INFO2:lightautoml.ml_algo.base:===== Start working with fold 1 for Lvl_0_Pipe_0_Mod_0_LinearL2 =====
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 1e-05 score = 0.9207247879610919
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 5e-05 score = 0.9220889132030743
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0001 score = 0.9224374400954178
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0005 score = 0.9229468142765833
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.001 score = 0.9230411065523483
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.005 score = 0.9230171583874173
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.01 score = 0.9230171583874173
INFO2:lightautoml.ml_algo.base:===== Start working with fold 2 for Lvl_0_Pipe_0_Mod_0_LinearL2 =====
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 1e-05 score = 0.9182978714961483
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 5e-05 score = 0.9196443322852692
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0001 score = 0.9199907170631663
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0005 score = 0.9204561561838415
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.001 score = 0.9205127175286687
```

```
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.005 score = 0.9204828165473964
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.01 score = 0.9204828165473964
INFO2:lightautoml.ml_algo.base:===== Start working with fold 3 for Lvl_0_Pipe_0_Mod_0_LinearL2 =====
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 1e-05 score = 0.9195326092338381
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 5e-05 score = 0.9208902620583936
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0001 score = 0.9212238131083296
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0005 score = 0.9217062044073195
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.001 score = 0.9217552382601347
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.005 score = 0.9216539892378866
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.01 score = 0.9216539892378866
INFO2:lightautoml.ml_algo.base:===== Start working with fold 4 for Lvl_0_Pipe_0_Mod_0_LinearL2 =====
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 1e-05 score = 0.9188429177559592
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 5e-05 score = 0.9202061722557697
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0001 score = 0.9205520842858719
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.0005 score = 0.9209964989055415
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.001 score = 0.921022124525509
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.005 score = 0.9209038398506002
INFO3:lightautoml.ml_algo.torch_based.linear_model:Linear model: C = 0.01 score = 0.9209038398506002
INFO:lightautoml.ml_algo.base:Fitting Lvl_0_Pipe_0_Mod_0_LinearL2 finished. score = 0.9216299226519797
INFO:lightautoml.ml_algo.base:Lvl_0_Pipe_0_Mod_0_LinearL2 fitting and predicting completed
INFO:lightautoml.automl.base:Time left 3520.93 secs

INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't improve for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.805562
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.807143
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.806775
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[235]        valid's auc: 0.807322
INFO:lightautoml.ml_algo.base:Selector_LightGBM fitting and predicting completed
INFO:lightautoml.ml_algo.base:Start fitting Lvl_0_Pipe_1_Mod_0_LightGBM ...
DEBUG:lightautoml.ml_algo.base:Training params: {'task': 'train', 'learning_rate': 0.05, 'num_leaves': 244, 'feature_fraction': 0.7, 'bagging_fraction': 0.7,
'bagging_freq': 1, 'max_depth': -1, 'verbosity': -1, 'reg_alpha': 1, 'reg_lambd
a': 0.0, 'min_split_gain': 0.0, 'zero_as_missing': False, 'num_threads': 2, 'ma
x_bin': 255, 'min_data_in_bin': 3, 'num_trees': 2000, 'early_stopping_rounds':
100, 'random_state': 42, 'verbose_eval': 100}
INFO2:lightautoml.ml_algo.base:===== Start working with fold 0 for Lvl_0_Pip
e_1_Mod_0_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.811671
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.81231
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[182]        valid's auc: 0.812371
INFO2:lightautoml.ml_algo.base:===== Start working with fold 1 for Lvl_0_Pip
e_1_Mod_0_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.812746
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.813933
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.813684
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[203]        valid's auc: 0.81394
INFO2:lightautoml.ml_algo.base:===== Start working with fold 2 for Lvl_0_Pip
e_1_Mod_0_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.809592
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.810647
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[191]        valid's auc: 0.810662
INFO2:lightautoml.ml_algo.base:===== Start working with fold 3 for Lvl_0_Pip
e_1_Mod_0_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.809652
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.810758
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[195]        valid's auc: 0.810759
INFO2:lightautoml.ml_algo.base:===== Start working with fold 4 for Lvl_0_Pip
e_1_Mod_0_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.810926
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.81189
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[186]        valid's auc: 0.812032
INFO:lightautoml.ml_algo.base:Fitting Lvl_0_Pipe_1_Mod_0_LightGBM finished. sco
re = 0.8119440285853597
INFO:lightautoml.ml_algo.base:Lvl_0_Pipe_1_Mod_0_LightGBM fitting and predictin
g completed
INFO:lightautoml.ml_algo.tuning.optuna:Start hyperparameters optimization for L
vl_0_Pipe_1_Mod_1_Tuned_LightGBM ... Time budget is 251.21 secs
INFO:optuna.storages._in_memory:A new study created in memory with name: no-nam
e-a3898529-8856-4142-a42f-b32829b4fff9
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
```

```
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.811624
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.812913
INFO3:lightautoml.ml_algo.boost_lgbm:[300]	valid's auc: 0.812652
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[203]	valid's auc: 0.812952
INFO:optuna.study.study:Trial 0 finished with value: 0.812951918359713 and para
meters: {'feature_fraction': 0.6872700594236812, 'num_leaves': 244, 'bagging_fr
action': 0.8659969709057025, 'min_sum_hessian_in_leaf': 0.24810409748678125, 'r
eg_alpha': 2.5361081166471375e-07, 'reg_lambda': 2.5348407664333426e-07}. Best
is trial 0 with value: 0.812951918359713.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 1 with hyperparameters {'featur
e_fraction': 0.6872700594236812, 'num_leaves': 244, 'bagging_fraction': 0.86599
69709057025, 'min_sum_hessian_in_leaf': 0.24810409748678125, 'reg_alpha': 2.536
1081166471375e-07, 'reg_lambda': 2.5348407664333426e-07} scored 0.8129519183597
13 in 0:00:24.597746
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.812669
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.813831
INFO3:lightautoml.ml_algo.boost_lgbm:[300]	valid's auc: 0.813528
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[212]	valid's auc: 0.813878
INFO:optuna.study.study:Trial 1 finished with value: 0.8138779043705615 and par
ameters: {'feature_fraction': 0.5290418060840998, 'num_leaves': 223, 'bagging_f
raction': 0.8005575058716043, 'min_sum_hessian_in_leaf': 0.679657809075816, 're
g_alpha': 1.5320059381854043e-08, 'reg_lambda': 5.360294728728285}. Best is tri
al 1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 2 with hyperparameters {'featur
e_fraction': 0.5290418060840998, 'num_leaves': 223, 'bagging_fraction': 0.80055
75058716043, 'min_sum_hessian_in_leaf': 0.679657809075816, 'reg_alpha': 1.53200
59381854043e-08, 'reg_lambda': 5.360294728728285} scored 0.8138779043705615 in
0:00:28.053844
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.811058
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.812576
INFO3:lightautoml.ml_algo.boost_lgbm:[300]	valid's auc: 0.812811
INFO3:lightautoml.ml_algo.boost_lgbm:[400]	valid's auc: 0.812969
INFO3:lightautoml.ml_algo.boost_lgbm:[500]	valid's auc: 0.812925
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[460]	valid's auc: 0.813035
INFO:optuna.study.study:Trial 2 finished with value: 0.8130346458187808 and par
ameters: {'feature_fraction': 0.9162213204002109, 'num_leaves': 66, 'bagging_fr
action': 0.5909124836035503, 'min_sum_hessian_in_leaf': 0.00541524411940254, 'r
eg_alpha': 5.472429642032198e-06, 'reg_lambda': 0.00052821153945323}. Best is t
rial 1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 3 with hyperparameters {'featur
e_fraction': 0.9162213204002109, 'num_leaves': 66, 'bagging_fraction': 0.590912
4836035503, 'min_sum_hessian_in_leaf': 0.00541524411940254, 'reg_alpha': 5.4724
29642032198e-06, 'reg_lambda': 0.00052821153945323} scored 0.8130346458187808 i
n 0:00:45.809815
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
```

```
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.811469
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.81311
INFO3:lightautoml.ml_algo.boost_lgbm:[300]	valid's auc: 0.813202
INFO3:lightautoml.ml_algo.boost_lgbm:[400]	valid's auc: 0.81333
INFO3:lightautoml.ml_algo.boost_lgbm:[500]	valid's auc: 0.813254
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[419]	valid's auc: 0.813371
INFO:optuna.study.study:Trial 3 finished with value: 0.8133712304837104 and par
ameters: {'feature_fraction': 0.7159725093210578, 'num_leaves': 85, 'bagging_fr
action': 0.8059264473611898, 'min_sum_hessian_in_leaf': 0.003613894271216527,
'reg_alpha': 4.258943089524393e-06, 'reg_lambda': 1.9826980964985924e-05}. Best
is trial 1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 4 with hyperparameters {'featur
e_fraction': 0.7159725093210578, 'num_leaves': 85, 'bagging_fraction': 0.805926
4473611898, 'min_sum_hessian_in_leaf': 0.003613894271216527, 'reg_alpha': 4.258
943089524393e-06, 'reg_lambda': 1.9826980964985924e-05} scored 0.81337123048371
04 in 0:00:33.620984
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.811318
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.812069
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[191]	valid's auc: 0.812111
INFO:optuna.study.study:Trial 4 finished with value: 0.8121114171988444 and par
ameters: {'feature_fraction': 0.728034992108518, 'num_leaves': 204, 'bagging_fr
action': 0.5998368910791798, 'min_sum_hessian_in_leaf': 0.11400863701127326, 'r
eg_alpha': 0.0021465011216654484, 'reg_lambda': 2.6185068507773707e-08}. Best i
s trial 1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 5 with hyperparameters {'featur
e_fraction': 0.728034992108518, 'num_leaves': 204, 'bagging_fraction': 0.599836
8910791798, 'min_sum_hessian_in_leaf': 0.11400863701127326, 'reg_alpha': 0.0021
465011216654484, 'reg_lambda': 2.6185068507773707e-08} scored 0.812111417198844
4 in 0:00:24.772352
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.811052
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.81293
INFO3:lightautoml.ml_algo.boost_lgbm:[300]	valid's auc: 0.813323
INFO3:lightautoml.ml_algo.boost_lgbm:[400]	valid's auc: 0.813473
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[370]	valid's auc: 0.813529
INFO:optuna.study.study:Trial 5 finished with value: 0.8135290095468597 and par
ameters: {'feature_fraction': 0.8037724259507192, 'num_leaves': 56, 'bagging_fr
action': 0.5325257964926398, 'min_sum_hessian_in_leaf': 6.245139574743075, 're
g_alpha': 4.905556676028774, 'reg_lambda': 0.18861495878553936}. Best is trial
1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 6 with hyperparameters {'featur
e_fraction': 0.8037724259507192, 'num_leaves': 56, 'bagging_fraction': 0.532525
7964926398, 'min_sum_hessian_in_leaf': 6.245139574743075, 'reg_alpha': 4.905556
676028774, 'reg_lambda': 0.18861495878553936} scored 0.8135290095468597 in 0:0
0:38.785187
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.810956
```

```
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.812861
INFO3:lightautoml.ml_algo.boost_lgbm:[300]	valid's auc: 0.813266
INFO3:lightautoml.ml_algo.boost_lgbm:[400]	valid's auc: 0.813504
INFO3:lightautoml.ml_algo.boost_lgbm:[500]	valid's auc: 0.813539
INFO3:lightautoml.ml_algo.boost_lgbm:[600]	valid's auc: 0.813622
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[596]	valid's auc: 0.813634
INFO:optuna.study.study:Trial 6 finished with value: 0.8136342696602215 and par
ameters: {'feature_fraction': 0.6523068845866853, 'num_leaves': 39, 'bagging_fr
action': 0.8421165132560784, 'min_sum_hessian_in_leaf': 0.057624872164786026,
'reg_alpha': 1.254134495897175e-07, 'reg_lambda': 0.00028614897264046574}. Best
is trial 1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 7 with hyperparameters {'featur
e_fraction': 0.6523068845866853, 'num_leaves': 39, 'bagging_fraction': 0.842116
5132560784, 'min_sum_hessian_in_leaf': 0.057624872164786026, 'reg_alpha': 1.254
134495897175e-07, 'reg_lambda': 0.00028614897264046574} scored 0.81363426966022
15 in 0:00:42.086891
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]	valid's auc: 0.812353
INFO3:lightautoml.ml_algo.boost_lgbm:[200]	valid's auc: 0.812894
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[185]	valid's auc: 0.812987
INFO:optuna.study.study:Trial 7 finished with value: 0.8129874701405736 and par
ameters: {'feature_fraction': 0.5171942605576092, 'num_leaves': 234, 'bagging_f
raction': 0.6293899908000085, 'min_sum_hessian_in_leaf': 0.4467752817973907, 'r
eg_alpha': 6.388511557344611e-06, 'reg_lambda': 0.0004793052550782129}. Best is
trial 1 with value: 0.8138779043705615.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 8 with hyperparameters {'featur
e_fraction': 0.5171942605576092, 'num_leaves': 234, 'bagging_fraction': 0.62938
99908000085, 'min_sum_hessian_in_leaf': 0.4467752817973907, 'reg_alpha': 6.3885
11557344611e-06, 'reg_lambda': 0.0004793052550782129} scored 0.8129874701405736
in 0:00:24.078991
INFO:lightautoml.ml_algo.tuning.optuna:Hyperparameters optimization for Lvl_0_P
ipe_1_Mod_1_Tuned_LightGBM completed
INFO2:lightautoml.ml_algo.tuning.optuna:The set of hyperparameters {'feature_fr
action': 0.5290418060840998, 'num_leaves': 223, 'bagging_fraction': 0.800557505
8716043, 'min_sum_hessian_in_leaf': 0.679657809075816, 'reg_alpha': 1.532005938
1854043e-08, 'reg_lambda': 5.360294728728285}
 achieve 0.8139 auc
INFO:lightautoml.ml_algo.base:Start fitting Lvl_0_Pipe_1_Mod_1_Tuned_LightGBM
...
DEBUG:lightautoml.ml_algo.base:Training params: {'task': 'train', 'learning_rat
e': 0.05, 'num_leaves': 223, 'feature_fraction': 0.5290418060840998, 'bagging_f
raction': 0.8005575058716043, 'bagging_freq': 1, 'max_depth': -1, 'verbosity':
-1, 'reg_alpha': 1.5320059381854043e-08, 'reg_lambda': 5.360294728728285, 'mi
n_split_gain': 0.0, 'zero_as_missing': False, 'num_threads': 2, 'max_bin': 255,
'min_data_in_bin': 3, 'num_trees': 3000, 'early_stopping_rounds': 100, 'rando
m_state': 42, 'verbose_eval': 100, 'min_sum_hessian_in_leaf': 0.67965780907581
6}
INFO2:lightautoml.ml_algo.base:===== Start working with fold 0 for Lvl_0_Pip
e_1_Mod_1_Tuned_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
```

```
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.812669
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.813831
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.813528
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[212]        valid's auc: 0.813878
INFO2:lightautoml.ml_algo.base:===== Start working with fold 1 for Lvl_0_Pip
e_1_Mod_1_Tuned_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.813585
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.814681
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.814499
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[219]        valid's auc: 0.814731
INFO2:lightautoml.ml_algo.base:===== Start working with fold 2 for Lvl_0_Pip
e_1_Mod_1_Tuned_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.810676
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.811931
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.811858
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[221]        valid's auc: 0.812039
INFO2:lightautoml.ml_algo.base:===== Start working with fold 3 for Lvl_0_Pip
e_1_Mod_1_Tuned_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.810625
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.811917
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.81197
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[234]        valid's auc: 0.812049
INFO2:lightautoml.ml_algo.base:===== Start working with fold 4 for Lvl_0_Pip
e_1_Mod_1_Tuned_LightGBM =====
INFO3:lightautoml.ml_algo.boost_lgbm:Training until validation scores don't imp
rove for 100 rounds
INFO3:lightautoml.ml_algo.boost_lgbm:[100]        valid's auc: 0.811805
INFO3:lightautoml.ml_algo.boost_lgbm:[200]        valid's auc: 0.812813
INFO3:lightautoml.ml_algo.boost_lgbm:[300]        valid's auc: 0.812733
INFO3:lightautoml.ml_algo.boost_lgbm:Early stopping, best iteration is:
[259]        valid's auc: 0.812854
INFO:lightautoml.ml_algo.base:Fitting Lvl_0_Pipe_1_Mod_1_Tuned_LightGBM finishe
d. score = 0.8131012966677932
INFO:lightautoml.ml_algo.base:Lvl_0_Pipe_1_Mod_1_Tuned_LightGBM fitting and pre
dicting completed
INFO:lightautoml.ml_algo.base:Start fitting Lvl_0_Pipe_1_Mod_2_CatBoost ...
DEBUG:lightautoml.ml_algo.base:Training params: {'task_type': 'CPU', 'thread_co
unt': 2, 'random_seed': 42, 'num_trees': 3000, 'learning_rate': 0.05, 'l2_lea
f_reg': 0.01, 'bootstrap_type': 'Bernoulli', 'grow_policy': 'SymmetricTree', 'm
ax_depth': 5, 'min_data_in_leaf': 1, 'one_hot_max_size': 10, 'fold_permutatio
n_block': 1, 'boosting_type': 'Plain', 'boost_from_average': True, 'od_type':
'Iter', 'od_wait': 100, 'max_bin': 32, 'feature_border_type': 'GreedyLogSum',
'nan_mode': 'Min', 'verbose': 100, 'allow_writing_files': False, 'verbose_eva
l': 100}
```

```
INFO2:lightautoml.ml_algo.base:===== Start working with fold 0 for Lvl_0_Pip
e_1_Mod_2_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7785775        best: 0.778
5775 (0)        total: 173ms        remaining: 8m 40s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8071229        best: 0.8
071229 (100)        total: 9.33s        remaining: 4m 27s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8078928        best: 0.8
078928 (200)        total: 18.7s        remaining: 4m 20s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8083606        best: 0.8
083626 (299)        total: 28s        remaining: 4m 11s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8086368        best: 0.8
086369 (399)        total: 36.1s        remaining: 3m 53s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8087594        best: 0.8
087594 (500)        total: 45.4s        remaining: 3m 46s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8088730        best: 0.8
088730 (600)        total: 54.7s        remaining: 3m 38s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8089214        best: 0.8
089214 (700)        total: 1m 3s        remaining: 3m 28s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8089245        best: 0.8
089401 (767)        total: 1m 12s        remaining: 3m 18s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8089690        best: 0.8
089731 (890)        total: 1m 21s        remaining: 3m 9s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8089731428
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 890
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 891 iterations.
INFO2:lightautoml.ml_algo.base:===== Start working with fold 1 for Lvl_0_Pip
e_1_Mod_2_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7802450        best: 0.780
2450 (0)        total: 108ms        remaining: 5m 25s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8079578        best: 0.8
079578 (100)        total: 8.51s        remaining: 4m 4s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8088038        best: 0.8
088038 (200)        total: 18.1s        remaining: 4m 11s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8092254        best: 0.8
092254 (300)        total: 27.5s        remaining: 4m 6s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8094487        best: 0.8
094487 (400)        total: 36.5s        remaining: 3m 56s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8095900        best: 0.8
095916 (498)        total: 45s        remaining: 3m 44s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8096829        best: 0.8
096831 (594)        total: 54.4s        remaining: 3m 37s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8097384        best: 0.8
097455 (688)        total: 1m 3s        remaining: 3m 28s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8097678        best: 0.8
097723 (793)        total: 1m 11s        remaining: 3m 17s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8098020        best: 0.8
098043 (891)        total: 1m 21s        remaining: 3m 8s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8098186        best:
0.8098246 (975)        total: 1m 30s        remaining: 3m
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8098248        best:
0.8098321 (1020)        total: 1m 38s        remaining: 2m 50s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8098320527
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 1020
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 1021 iterations.
INFO2:lightautoml.ml_algo.base:===== Start working with fold 2 for Lvl_0_Pip
e_1_Mod_2_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7775127        best: 0.777
5127 (0)        total: 80.8ms        remaining: 4m 2s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8043840        best: 0.8
043840 (100)        total: 8.38s        remaining: 4m
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8053554        best: 0.8
053554 (200)        total: 17.8s        remaining: 4m 8s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8058415        best: 0.8
058415 (300)        total: 27.1s        remaining: 4m 2s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8061365        best: 0.8
061365 (400)        total: 36.3s        remaining: 3m 55s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8063301        best: 0.8
063301 (500)        total: 44.5s        remaining: 3m 42s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8064184        best: 0.8
064184 (600)        total: 53.9s        remaining: 3m 35s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8064577        best: 0.8
064701 (669)        total: 1m 3s        remaining: 3m 27s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8065048        best: 0.8
065048 (800)        total: 1m 11s        remaining: 3m 15s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8065237        best: 0.8
065256 (893)        total: 1m 20s        remaining: 3m 8s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8065724        best:
0.8065734 (998)        total: 1m 30s        remaining: 2m 59s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8065877        best:
0.8065890 (1096)        total: 1m 39s        remaining: 2m 51s
INFO3:lightautoml.ml_algo.boost_cb:1200:        test: 0.8065924        best:
0.8065936 (1199)        total: 1m 47s        remaining: 2m 40s
INFO3:lightautoml.ml_algo.boost_cb:1300:        test: 0.8065854        best:
0.8066028 (1255)        total: 1m 56s        remaining: 2m 32s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8066027784
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 1255
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 1256 iterations.
INFO2:lightautoml.ml_algo.base:===== Start working with fold 3 for Lvl_0_Pip
e_1_Mod_2_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7865672        best: 0.786
5672 (0)        total: 95.9ms        remaining: 4m 47s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8046093        best: 0.8
046093 (100)        total: 9.81s        remaining: 4m 41s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8054473        best: 0.8
054473 (200)        total: 19.2s        remaining: 4m 27s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8057701        best: 0.8
057701 (300)        total: 28.6s        remaining: 4m 16s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8059876        best: 0.8
059877 (397)        total: 36.6s        remaining: 3m 57s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8061071        best: 0.8
061076 (495)        total: 46s        remaining: 3m 49s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8061815        best: 0.8
```

```
061835 (589)        total: 55.3s        remaining: 3m 40s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8062390        best: 0.8
062441 (687)        total: 1m 3s        remaining: 3m 27s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8062743        best: 0.8
062743 (800)        total: 1m 12s        remaining: 3m 19s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8062280        best: 0.8
062743 (800)        total: 1m 21s        remaining: 3m 10s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8062742755
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 800
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 801 iterations.
INFO2:lightautoml.ml_algo.base:===== Start working with fold 4 for Lvl_0_Pip
e_1_Mod_2_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7818698        best: 0.781
8698 (0)        total: 135ms        remaining: 6m 44s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8061032        best: 0.8
061032 (100)        total: 9.86s        remaining: 4m 43s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8070111        best: 0.8
070111 (200)        total: 18s        remaining: 4m 10s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8074781        best: 0.8
074781 (300)        total: 27.3s        remaining: 4m 4s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8077167        best: 0.8
077167 (400)        total: 36.7s        remaining: 3m 57s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8078372        best: 0.8
078389 (499)        total: 45.1s        remaining: 3m 45s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8079458        best: 0.8
079458 (600)        total: 54.1s        remaining: 3m 35s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8079693        best: 0.8
079693 (700)        total: 1m 3s        remaining: 3m 28s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8080080        best: 0.8
080080 (800)        total: 1m 12s        remaining: 3m 20s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8080413        best: 0.8
080420 (898)        total: 1m 20s        remaining: 3m 8s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8080819        best:
0.8080835 (989)        total: 1m 30s        remaining: 3m
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8080775        best:
0.8080848 (1090)        total: 1m 39s        remaining: 2m 51s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8080847591
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 1090
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 1091 iterations.
INFO:lightautoml.ml_algo.base:Fitting Lvl_0_Pipe_1_Mod_2_CatBoost finished. sco
re = 0.8079388653577905
INFO:lightautoml.ml_algo.base:Lvl_0_Pipe_1_Mod_2_CatBoost fitting and predictin
g completed
INFO:lightautoml.ml_algo.tuning.optuna:Start hyperparameters optimization for L
vl_0_Pipe_1_Mod_3_Tuned_CatBoost ... Time budget is 1.00 secs
INFO:optuna.storages._in_memory:A new study created in memory with name: no-nam
e-d8f87b7e-7776-4e5c-bffe-c5ddfd3a6e12
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7770137        best: 0.777
0137 (0)        total: 195ms        remaining: 9m 43s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8067830        best: 0.8
```

```
067830 (100)        total: 9.18s        remaining: 4m 23s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8076886        best: 0.8
076886 (200)        total: 18.2s        remaining: 4m 13s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8081932        best: 0.8
081932 (300)        total: 26.7s        remaining: 3m 59s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8084563        best: 0.8
084564 (397)        total: 34.7s        remaining: 3m 44s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8086171        best: 0.8
086171 (499)        total: 43.6s        remaining: 3m 37s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8087574        best: 0.8
087574 (600)        total: 52.1s        remaining: 3m 27s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8088464        best: 0.8
088469 (699)        total: 59.9s        remaining: 3m 16s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8088985        best: 0.8
088985 (800)        total: 1m 8s        remaining: 3m 9s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8089484        best: 0.8
089486 (899)        total: 1m 17s        remaining: 3m 1s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8089735        best:
0.8089739 (999)        total: 1m 25s        remaining: 2m 50s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8089705        best:
0.8089801 (1013)        total: 1m 34s        remaining: 2m 42s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8089800777
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 1013
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 1014 iterations.
INFO:optuna.study.study:Trial 0 finished with value: 0.808980082590339 and para
meters: {'max_depth': 4, 'l2_leaf_reg': 3.6010467344475403, 'min_data_in_leaf':
15}. Best is trial 0 with value: 0.808980082590339.
INFO3:lightautoml.ml_algo.tuning.optuna:Trial 1 with hyperparameters {'max_dept
h': 4, 'l2_leaf_reg': 3.6010467344475403, 'min_data_in_leaf': 15} scored 0.8089
80082590339 in 0:01:35.798859
INFO:lightautoml.ml_algo.tuning.optuna:Hyperparameters optimization for Lvl_0_P
ipe_1_Mod_3_Tuned_CatBoost completed
INFO2:lightautoml.ml_algo.tuning.optuna:The set of hyperparameters {'max_dept
h': 4, 'l2_leaf_reg': 3.6010467344475403, 'min_data_in_leaf': 15}
 achieve 0.8090 auc
INFO:lightautoml.ml_algo.base:Start fitting Lvl_0_Pipe_1_Mod_3_Tuned_CatBoost
...
DEBUG:lightautoml.ml_algo.base:Training params: {'task_type': 'CPU', 'thread_co
unt': 2, 'random_seed': 42, 'num_trees': 3000, 'learning_rate': 0.03, 'l2_lea
f_reg': 3.6010467344475403, 'bootstrap_type': 'Bernoulli', 'grow_policy': 'Symm
etricTree', 'max_depth': 4, 'min_data_in_leaf': 15, 'one_hot_max_size': 10, 'fo
ld_permutation_block': 1, 'boosting_type': 'Plain', 'boost_from_average': True,
'od_type': 'Iter', 'od_wait': 100, 'max_bin': 32, 'feature_border_type': 'Greed
yLogSum', 'nan_mode': 'Min', 'verbose': 100, 'allow_writing_files': False, 'ver
bose_eval': 100}
INFO2:lightautoml.ml_algo.base:===== Start working with fold 0 for Lvl_0_Pip
e_1_Mod_3_Tuned_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7770137        best: 0.777
0137 (0)        total: 71.6ms        remaining: 3m 34s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8057664        best: 0.8
057664 (100)        total: 9.43s        remaining: 4m 30s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8070112        best: 0.8
```

```
070112 (200)        total: 17.3s        remaining: 4m 1s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8075662        best: 0.8
075662 (300)        total: 26.3s        remaining: 3m 55s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8079029        best: 0.8
079029 (400)        total: 35.3s        remaining: 3m 48s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8081949        best: 0.8
081949 (500)        total: 43.3s        remaining: 3m 36s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8083823        best: 0.8
083823 (600)        total: 51.8s        remaining: 3m 26s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8085214        best: 0.8
085214 (700)        total: 1m        remaining: 3m 19s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8086037        best: 0.8
086038 (799)        total: 1m 8s        remaining: 3m 9s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8086764        best: 0.8
086764 (900)        total: 1m 17s        remaining: 2m 59s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8087423        best:
0.8087423 (1000)        total: 1m 26s        remaining: 2m 51s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8087944        best:
0.8087944 (1100)        total: 1m 34s        remaining: 2m 42s
INFO3:lightautoml.ml_algo.boost_cb:1200:        test: 0.8088334        best:
0.8088334 (1200)        total: 1m 42s        remaining: 2m 33s
INFO3:lightautoml.ml_algo.boost_cb:1300:        test: 0.8088749        best:
0.8088749 (1300)        total: 1m 51s        remaining: 2m 25s
INFO3:lightautoml.ml_algo.boost_cb:1400:        test: 0.8089186        best:
0.8089186 (1400)        total: 2m        remaining: 2m 17s
INFO3:lightautoml.ml_algo.boost_cb:1500:        test: 0.8089564        best:
0.8089572 (1498)        total: 2m 7s        remaining: 2m 7s
INFO3:lightautoml.ml_algo.boost_cb:1600:        test: 0.8089781        best:
0.8089781 (1600)        total: 2m 16s        remaining: 1m 59s
INFO3:lightautoml.ml_algo.boost_cb:1700:        test: 0.8089883        best:
0.8089889 (1687)        total: 2m 25s        remaining: 1m 51s
INFO3:lightautoml.ml_algo.boost_cb:1800:        test: 0.8090037        best:
0.8090045 (1795)        total: 2m 33s        remaining: 1m 41s
INFO3:lightautoml.ml_algo.boost_cb:1900:        test: 0.8090207        best:
0.8090234 (1872)        total: 2m 41s        remaining: 1m 33s
INFO3:lightautoml.ml_algo.boost_cb:2000:        test: 0.8090350        best:
0.8090351 (1975)        total: 2m 50s        remaining: 1m 25s
INFO3:lightautoml.ml_algo.boost_cb:2100:        test: 0.8090371        best:
0.8090377 (2099)        total: 2m 58s        remaining: 1m 16s
INFO3:lightautoml.ml_algo.boost_cb:2200:        test: 0.8090429        best:
0.8090432 (2199)        total: 3m 7s        remaining: 1m 7s
INFO3:lightautoml.ml_algo.boost_cb:2300:        test: 0.8090424        best:
0.8090466 (2227)        total: 3m 15s        remaining: 59.5s
INFO3:lightautoml.ml_algo.boost_cb:2400:        test: 0.8090571        best:
0.8090579 (2399)        total: 3m 23s        remaining: 50.8s
INFO3:lightautoml.ml_algo.boost_cb:2500:        test: 0.8090586        best:
0.8090586 (2410)        total: 3m 32s        remaining: 42.4s
INFO3:lightautoml.ml_algo.boost_cb:2600:        test: 0.8090545        best:
0.8090591 (2501)        total: 3m 41s        remaining: 33.9s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8090591109
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 2501
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 2502 iterations.
```

```
INFO2:lightautoml.ml_algo.base:===== Start working with fold 1 for Lvl_0_Pipe_1_Mod_3_Tuned_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7788817        best: 0.7788817 (0)        total: 82.2ms        remaining: 4m 6s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8066997        best: 0.8066997 (100)        total: 8.05s        remaining: 3m 51s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8079476        best: 0.8079476 (200)        total: 17.1s        remaining: 3m 58s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8085041        best: 0.8085041 (300)        total: 26.1s        remaining: 3m 53s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8088239        best: 0.8088239 (400)        total: 33.8s        remaining: 3m 38s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8090701        best: 0.8090701 (500)        total: 42.7s        remaining: 3m 32s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8092424        best: 0.8092424 (600)        total: 51.6s        remaining: 3m 25s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8093603        best: 0.8093612 (699)        total: 59.2s        remaining: 3m 14s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8094531        best: 0.8094531 (800)        total: 1m 8s        remaining: 3m 6s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8095172        best: 0.8095172 (900)        total: 1m 16s        remaining: 2m 59s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8095766        best: 0.8095766 (1000)        total: 1m 24s        remaining: 2m 48s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8096242        best: 0.8096242 (1100)        total: 1m 33s        remaining: 2m 41s
INFO3:lightautoml.ml_algo.boost_cb:1200:        test: 0.8096627        best: 0.8096635 (1194)        total: 1m 42s        remaining: 2m 33s
INFO3:lightautoml.ml_algo.boost_cb:1300:        test: 0.8096968        best: 0.8096968 (1300)        total: 1m 50s        remaining: 2m 23s
INFO3:lightautoml.ml_algo.boost_cb:1400:        test: 0.8097292        best: 0.8097292 (1400)        total: 1m 58s        remaining: 2m 15s
INFO3:lightautoml.ml_algo.boost_cb:1500:        test: 0.8097490        best: 0.8097515 (1464)        total: 2m 7s        remaining: 2m 7s
INFO3:lightautoml.ml_algo.boost_cb:1600:        test: 0.8097658        best: 0.8097658 (1600)        total: 2m 16s        remaining: 1m 58s
INFO3:lightautoml.ml_algo.boost_cb:1700:        test: 0.8097740        best: 0.8097768 (1683)        total: 2m 24s        remaining: 1m 50s
INFO3:lightautoml.ml_algo.boost_cb:1800:        test: 0.8097954        best: 0.8097964 (1792)        total: 2m 33s        remaining: 1m 42s
INFO3:lightautoml.ml_algo.boost_cb:1900:        test: 0.8098102        best: 0.8098118 (1888)        total: 2m 41s        remaining: 1m 33s
INFO3:lightautoml.ml_algo.boost_cb:2000:        test: 0.8098342        best: 0.8098353 (1998)        total: 2m 49s        remaining: 1m 24s
INFO3:lightautoml.ml_algo.boost_cb:2100:        test: 0.8098359        best: 0.8098376 (2055)        total: 2m 58s        remaining: 1m 16s
INFO3:lightautoml.ml_algo.boost_cb:2200:        test: 0.8098479        best: 0.8098479 (2200)        total: 3m 6s        remaining: 1m 7s
INFO3:lightautoml.ml_algo.boost_cb:2300:        test: 0.8098473        best: 0.8098500 (2208)        total: 3m 14s        remaining: 59.2s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterations wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8098500218
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 2208
```

```
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 2209 iterations.
INFO2:lightautoml.ml_algo.base:===== Start working with fold 2 for Lvl_0_Pip
e_1_Mod_3_Tuned_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7762042        best: 0.776
2042 (0)        total: 89.5ms        remaining: 4m 28s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8030810        best: 0.8
030810 (100)        total: 9.36s        remaining: 4m 28s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8044345        best: 0.8
044345 (200)        total: 18.4s        remaining: 4m 16s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8050059        best: 0.8
050059 (300)        total: 26.1s        remaining: 3m 53s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8053871        best: 0.8
053871 (400)        total: 35s        remaining: 3m 46s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8056882        best: 0.8
056882 (500)        total: 43.9s        remaining: 3m 39s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8059279        best: 0.8
059279 (600)        total: 51.7s        remaining: 3m 26s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8060637        best: 0.8
060637 (700)        total: 1m        remaining: 3m 18s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8061939        best: 0.8
061939 (800)        total: 1m 9s        remaining: 3m 10s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8062956        best: 0.8
062956 (900)        total: 1m 17s        remaining: 2m 59s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8063903        best:
0.8063903 (1000)        total: 1m 25s        remaining: 2m 51s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8064445        best:
0.8064459 (1099)        total: 1m 34s        remaining: 2m 43s
INFO3:lightautoml.ml_algo.boost_cb:1200:        test: 0.8064839        best:
0.8064858 (1192)        total: 1m 42s        remaining: 2m 33s
INFO3:lightautoml.ml_algo.boost_cb:1300:        test: 0.8065307        best:
0.8065313 (1297)        total: 1m 51s        remaining: 2m 25s
INFO3:lightautoml.ml_algo.boost_cb:1400:        test: 0.8065682        best:
0.8065687 (1388)        total: 1m 59s        remaining: 2m 16s
INFO3:lightautoml.ml_algo.boost_cb:1500:        test: 0.8066052        best:
0.8066052 (1500)        total: 2m 7s        remaining: 2m 7s
INFO3:lightautoml.ml_algo.boost_cb:1600:        test: 0.8066373        best:
0.8066373 (1600)        total: 2m 16s        remaining: 1m 59s
INFO3:lightautoml.ml_algo.boost_cb:1700:        test: 0.8066674        best:
0.8066687 (1697)        total: 2m 25s        remaining: 1m 50s
INFO3:lightautoml.ml_algo.boost_cb:1800:        test: 0.8066983        best:
0.8066983 (1800)        total: 2m 32s        remaining: 1m 41s
INFO3:lightautoml.ml_algo.boost_cb:1900:        test: 0.8067109        best:
0.8067109 (1900)        total: 2m 41s        remaining: 1m 33s
INFO3:lightautoml.ml_algo.boost_cb:2000:        test: 0.8067194        best:
0.8067197 (1992)        total: 2m 50s        remaining: 1m 25s
INFO3:lightautoml.ml_algo.boost_cb:2100:        test: 0.8067346        best:
0.8067375 (2086)        total: 2m 58s        remaining: 1m 16s
INFO3:lightautoml.ml_algo.boost_cb:2200:        test: 0.8067382        best:
0.8067483 (2148)        total: 3m 7s        remaining: 1m 7s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.80674831
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 2148
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 2149 iterations.
```

```
INFO2:lightautoml.ml_algo.base:===== Start working with fold 3 for Lvl_0_Pip
e_1_Mod_3_Tuned_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7763464        best: 0.776
3464 (0)        total: 225ms        remaining: 11m 15s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8034457        best: 0.8
034457 (100)        total: 9.04s        remaining: 4m 19s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8045751        best: 0.8
045751 (200)        total: 18.1s        remaining: 4m 12s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8050887        best: 0.8
050887 (300)        total: 27.1s        remaining: 4m 2s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8054038        best: 0.8
054038 (400)        total: 34.6s        remaining: 3m 44s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8056522        best: 0.8
056522 (500)        total: 43.5s        remaining: 3m 36s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8058324        best: 0.8
058324 (600)        total: 52.3s        remaining: 3m 28s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8059506        best: 0.8
059506 (700)        total: 59.8s        remaining: 3m 16s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8060273        best: 0.8
060273 (800)        total: 1m 8s        remaining: 3m 8s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8060968        best: 0.8
060968 (900)        total: 1m 17s        remaining: 3m 1s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8061479        best:
0.8061479 (1000)        total: 1m 25s        remaining: 2m 50s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8062000        best:
0.8062028 (1094)        total: 1m 34s        remaining: 2m 42s
INFO3:lightautoml.ml_algo.boost_cb:1200:        test: 0.8062338        best:
0.8062338 (1200)        total: 1m 43s        remaining: 2m 34s
INFO3:lightautoml.ml_algo.boost_cb:1300:        test: 0.8062615        best:
0.8062644 (1295)        total: 1m 50s        remaining: 2m 24s
INFO3:lightautoml.ml_algo.boost_cb:1400:        test: 0.8062987        best:
0.8062987 (1400)        total: 1m 59s        remaining: 2m 16s
INFO3:lightautoml.ml_algo.boost_cb:1500:        test: 0.8063193        best:
0.8063193 (1500)        total: 2m 8s        remaining: 2m 8s
INFO3:lightautoml.ml_algo.boost_cb:1600:        test: 0.8063331        best:
0.8063336 (1593)        total: 2m 16s        remaining: 1m 59s
INFO3:lightautoml.ml_algo.boost_cb:1700:        test: 0.8063440        best:
0.8063472 (1682)        total: 2m 25s        remaining: 1m 50s
INFO3:lightautoml.ml_algo.boost_cb:1800:        test: 0.8063550        best:
0.8063554 (1798)        total: 2m 34s        remaining: 1m 42s
INFO3:lightautoml.ml_algo.boost_cb:1900:        test: 0.8063692        best:
0.8063698 (1889)        total: 2m 41s        remaining: 1m 33s
INFO3:lightautoml.ml_algo.boost_cb:2000:        test: 0.8063853        best:
0.8063853 (2000)        total: 2m 50s        remaining: 1m 25s
INFO3:lightautoml.ml_algo.boost_cb:2100:        test: 0.8063845        best:
0.8063932 (2029)        total: 2m 59s        remaining: 1m 16s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8063932379
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 2029
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 2030 iterations.
INFO2:lightautoml.ml_algo.base:===== Start working with fold 4 for Lvl_0_Pip
e_1_Mod_3_Tuned_CatBoost =====
INFO3:lightautoml.ml_algo.boost_cb:0:        test: 0.7810857        best: 0.781
```

```
0857 (0)        total: 95.4ms        remaining: 4m 46s
INFO3:lightautoml.ml_algo.boost_cb:100:        test: 0.8048998        best: 0.8
049024 (99)        total: 9.48s        remaining: 4m 32s
INFO3:lightautoml.ml_algo.boost_cb:200:        test: 0.8062583        best: 0.8
062583 (200)        total: 17.2s        remaining: 3m 59s
INFO3:lightautoml.ml_algo.boost_cb:300:        test: 0.8068273        best: 0.8
068273 (300)        total: 26.2s        remaining: 3m 54s
INFO3:lightautoml.ml_algo.boost_cb:400:        test: 0.8071119        best: 0.8
071119 (400)        total: 35.1s        remaining: 3m 47s
INFO3:lightautoml.ml_algo.boost_cb:500:        test: 0.8074052        best: 0.8
074052 (500)        total: 42.7s        remaining: 3m 33s
INFO3:lightautoml.ml_algo.boost_cb:600:        test: 0.8075923        best: 0.8
075923 (600)        total: 51.6s        remaining: 3m 26s
INFO3:lightautoml.ml_algo.boost_cb:700:        test: 0.8077365        best: 0.8
077365 (700)        total: 1m        remaining: 3m 18s
INFO3:lightautoml.ml_algo.boost_cb:800:        test: 0.8078289        best: 0.8
078289 (800)        total: 1m 8s        remaining: 3m 7s
INFO3:lightautoml.ml_algo.boost_cb:900:        test: 0.8078949        best: 0.8
078956 (899)        total: 1m 17s        remaining: 2m 59s
INFO3:lightautoml.ml_algo.boost_cb:1000:        test: 0.8079448        best:
0.8079448 (999)        total: 1m 25s        remaining: 2m 51s
INFO3:lightautoml.ml_algo.boost_cb:1100:        test: 0.8079841        best:
0.8079841 (1100)        total: 1m 33s        remaining: 2m 41s
INFO3:lightautoml.ml_algo.boost_cb:1200:        test: 0.8080231        best:
0.8080242 (1198)        total: 1m 42s        remaining: 2m 33s
INFO3:lightautoml.ml_algo.boost_cb:1300:        test: 0.8080645        best:
0.8080645 (1300)        total: 1m 51s        remaining: 2m 25s
INFO3:lightautoml.ml_algo.boost_cb:1400:        test: 0.8081011        best:
0.8081012 (1398)        total: 1m 58s        remaining: 2m 15s
INFO3:lightautoml.ml_algo.boost_cb:1500:        test: 0.8081331        best:
0.8081331 (1500)        total: 2m 7s        remaining: 2m 7s
INFO3:lightautoml.ml_algo.boost_cb:1600:        test: 0.8081506        best:
0.8081510 (1585)        total: 2m 16s        remaining: 1m 59s
INFO3:lightautoml.ml_algo.boost_cb:1700:        test: 0.8081703        best:
0.8081713 (1689)        total: 2m 24s        remaining: 1m 50s
INFO3:lightautoml.ml_algo.boost_cb:1800:        test: 0.8081834        best:
0.8081860 (1782)        total: 2m 32s        remaining: 1m 41s
INFO3:lightautoml.ml_algo.boost_cb:1900:        test: 0.8082000        best:
0.8082016 (1896)        total: 2m 41s        remaining: 1m 33s
INFO3:lightautoml.ml_algo.boost_cb:2000:        test: 0.8082161        best:
0.8082171 (1987)        total: 2m 49s        remaining: 1m 24s
INFO3:lightautoml.ml_algo.boost_cb:2100:        test: 0.8082178        best:
0.8082215 (2063)        total: 2m 57s        remaining: 1m 16s
INFO3:lightautoml.ml_algo.boost_cb:2200:        test: 0.8082229        best:
0.8082234 (2193)        total: 3m 6s        remaining: 1m 7s
INFO3:lightautoml.ml_algo.boost_cb:2300:        test: 0.8082330        best:
0.8082341 (2261)        total: 3m 15s        remaining: 59.3s
INFO3:lightautoml.ml_algo.boost_cb:2400:        test: 0.8082396        best:
0.8082406 (2393)        total: 3m 23s        remaining: 50.7s
INFO3:lightautoml.ml_algo.boost_cb:2500:        test: 0.8082412        best:
0.8082457 (2470)        total: 3m 31s        remaining: 42.3s
INFO3:lightautoml.ml_algo.boost_cb:Stopped by overfitting detector  (100 iterat
ions wait)
INFO3:lightautoml.ml_algo.boost_cb:bestTest = 0.8082456947
```

```
INFO3:lightautoml.ml_algo.boost_cb:bestIteration = 2470
INFO3:lightautoml.ml_algo.boost_cb:Shrink model to first 2471 iterations.
INFO:lightautoml.ml_algo.base:Fitting Lvl_0_Pipe_1_Mod_3_Tuned_CatBoost finishe
d. score = 0.8080471694462876
INFO:lightautoml.ml_algo.base:Lvl_0_Pipe_1_Mod_3_Tuned_CatBoost fitting and pre
dicting completed
INFO:lightautoml.automl.base:Time left 1342.58 secs

INFO:lightautoml.automl.base:Layer 1 training completed.

INFO:lightautoml.automl.blend:Blending: optimization starts with equal weights.
Score = 0.8867972
INFO:lightautoml.automl.blend:Blending: iteration 0: score = 0.9222639, weights
= [0.79122955 0.09529735 0.11347307 0.         0.        ]
INFO:lightautoml.automl.blend:Blending: iteration 1: score = 0.9223727, weights
= [0.8426707  0.06372625 0.09360307 0.         0.        ]
INFO:lightautoml.automl.blend:Blending: iteration 2: score = 0.9223746, weights
= [0.839163   0.06106354 0.09977347 0.         0.        ]
INFO:lightautoml.automl.blend:Blending: no improvements for score. Terminated.

INFO:lightautoml.automl.blend:Blending: best score = 0.9223746, best weights =
[0.839163   0.06106354 0.09977347 0.         0.        ]
INFO:lightautoml.automl.presets.base:Automl preset training completed in 2286.2
3 seconds

INFO:lightautoml.automl.presets.base:Model description:
Final prediction for new objects (level 0) =
         0.83916 * (5 averaged models Lvl_0_Pipe_0_Mod_0_LinearL2) +
         0.06106 * (5 averaged models Lvl_0_Pipe_1_Mod_0_LightGBM) +
         0.09977 * (5 averaged models Lvl_0_Pipe_1_Mod_1_Tuned_LightGBM)
```

```
LightAutoML CV AUC: 0.92237

Файл submission.csv успешно сохранён!
       id  loan_paid_back
0  593994        0.907680
1  593995        0.956158
2  593996        0.417844
3  593997        0.919785
4  593998        0.954089
```

Результаты

- Private Score 0.92584
- Public Score 0.92477

Catboost pipeline

Почему Pipeline избыточен

1. **CatBoost работает с Pool объектами** - это специальная структура данных CatBoost. Pipeline не умеет работать с Pool

2. **Двухэтапный процесс (Optuna + CV)** - сначала идёт оптимизация на hold-out, потом обучение с CV. Pipeline не добавит ценности ни на одном из этапов

3. **GPU-специфичные параметры** - `task_type`, `devices` передаются напрямую в конструктор CatBoost, Pipeline только усложнит их управление

In [ ]:

```
import warnings

import optuna
import pandas as pd
from catboost import CatBoostClassifier, Pool
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import StratifiedKFold, train_test_split

warnings.filterwarnings("ignore")

# Данные
train_data = train_fe.copy()
test_data = test_fe.copy()
y = target.copy().reset_index(drop=True)
test_ids = test_ids.copy()

# Категориальные признаки
cat_cols = train_data.select_dtypes(include=["object", "category"]).columns.tc
print(f"Категориальные признаки: {cat_cols}")

# Индексы категориальных колонок
cat_feature_indices = [train_data.columns.get_loc(c) for c in cat_cols]

# для воспроизводимости
RND = 42

# функция Optuna
def objective(trial):

    # Только поддерживаемые GPU лоссы
    loss = trial.suggest_categorical("loss_function", ["Logloss", "CrossEntrop
    params = {
        "iterations": trial.suggest_int("iterations", 200, 2000),
        "learning_rate": trial.suggest_loguniform("learning_rate", 0.001, 0.15
        "depth": trial.suggest_int("depth", 1, 8),
        "l2_leaf_reg": trial.suggest_loguniform("l2_leaf_reg", 0.1, 10.0),
        "bagging_temperature": trial.suggest_uniform("bagging_temperature", 0.
        # GPU
        "task_type": "GPU",
        "devices": "0",
        "random_seed": RND,
        "loss_function": loss,
        "eval_metric": "AUC",
        "verbose": False,
    }

    # Hold-out
    X_tr, X_val, y_tr, y_val = train_test_split(
        train_data, y, test_size=0.2, stratify=y, random_state=RND
    )

    train_pool = Pool(X_tr, label=y_tr, cat_features=cat_feature_indices)
    val_pool = Pool(X_val, label=y_val, cat_features=cat_feature_indices)

    model = CatBoostClassifier(**params)

    model.fit(
        train_pool, eval_set=val_pool, use_best_model=True, early_stopping_rou
    )

    pred = model.predict_proba(X_val)[:, 1]
    auc = roc_auc_score(y_val, pred)

    del model, X_tr, X_val, y_tr, y_val, train_pool, val_pool
    gc.collect()

    return auc


# Оптимизация через optuna

study = optuna.create_study(direction="maximize", study_name="catboost_gpu_opt
print("\nЗапуск Optuna (GPU)")
study.optimize(objective, n_trials=30, n_jobs=1)

print("\nOptuna завершён")
print(f"Лучший AUC (hold-out): {study.best_value:.5f}")
print("Лучшие параметры:")
print(study.best_params)

best_params = study.best_params.copy()

# GPU-настройки для финального обучения
best_params.update(
    {"task_type": "GPU", "devices": "0", "random_seed": RND, "verbose": 200, "
)


N_FOLDS = 3
skf = StratifiedKFold(n_splits=N_FOLDS, shuffle=True, random_state=RND)

oof_cat = np.zeros(len(train_data))
pred_cat = np.zeros(len(test_data))

for fold, (tr_idx, val_idx) in enumerate(skf.split(train_data, y)):
    print(f"\n=== Fold {fold + 1}/{N_FOLDS} ===")

    X_tr = train_data.iloc[tr_idx].reset_index(drop=True)
    X_val = train_data.iloc[val_idx].reset_index(drop=True)
    y_tr = y.iloc[tr_idx].reset_index(drop=True)
    y_val = y.iloc[val_idx].reset_index(drop=True)

    train_pool = Pool(X_tr, label=y_tr, cat_features=cat_feature_indices)
    val_pool = Pool(X_val, label=y_val, cat_features=cat_feature_indices)
    test_pool = Pool(test_data, cat_features=cat_feature_indices)

    model = CatBoostClassifier(**best_params)

    model.fit(
        train_pool, eval_set=val_pool, use_best_model=True, early_stopping_rou
    )

    oof_cat[val_idx] = model.predict_proba(X_val)[:, 1]
    pred_cat += model.predict_proba(test_data)[:, 1] / N_FOLDS

    fold_auc = roc_auc_score(y_val, oof_cat[val_idx])
    print(f"Fold AUC: {fold_auc:.5f}")

    del X_tr, X_val, y_tr, y_val, train_pool, val_pool, test_pool, model
    gc.collect()

# Полный CV AUC
final_auc = roc_auc_score(y, oof_cat)
print(f"\nИтоговый CV AUC CatBoost: {final_auc:.5f}")


# Сохранение submission
submission = pd.DataFrame({"id": test_ids, "loan_paid_back": pred_cat})

submission.to_csv("submission_catboost_gpu_optuna.csv", index=False)
print("\nSubmission сохранён: submission_catboost_gpu_optuna.csv")
```

```
[I 2025-12-11 20:39:22,023] A new study created in memory with name: catboost_g
pu_opt
```

```
Категориальные признаки: ['gender', 'marital_status', 'education_level', 'emplo
yment_status', 'loan_purpose', 'grade_subgrade', 'grade_letter', 'employment_ed
ucation_level', 'employment_marital_status', 'employment_loan_purpose', 'employ
ment_status_education_level_combo', 'employment_status_marital_status_combo',
'grade_subgrade_loan_purpose_combo', 'education_level_loan_purpose_combo']
CatBoost version: 1.2.8

Запуск Optuna (GPU)
```

```
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:42:28,401] Trial 0 finished with value: 0.9175501110826435 and
parameters: {'loss_function': 'Logloss', 'iterations': 1733, 'learning_rate':
0.01273514932958299, 'depth': 7, 'l2_leaf_reg': 0.46520561464476107, 'bagging_t
emperature': 0.2802152482103478}. Best is trial 0 with value: 0.917550111082643
5.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:45:19,876] Trial 1 finished with value: 0.9150484243080543 and
parameters: {'loss_function': 'Logloss', 'iterations': 1901, 'learning_rate':
0.003831509753263433, 'depth': 6, 'l2_leaf_reg': 1.1102045480486629, 'bagging_t
emperature': 0.4390568159500591}. Best is trial 0 with value: 0.917550111082643
5.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:46:20,520] Trial 2 finished with value: 0.9174818336922924 and
parameters: {'loss_function': 'Logloss', 'iterations': 737, 'learning_rate':
0.05779281152988618, 'depth': 5, 'l2_leaf_reg': 0.2374692418682117, 'bagging_te
mperature': 0.9864313192802572}. Best is trial 0 with value: 0.917550111082643
5.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:50:23,773] Trial 3 finished with value: 0.9132435600659075 and
parameters: {'loss_function': 'CrossEntropy', 'iterations': 1986, 'learning_rat
e': 0.0011261767848962528, 'depth': 8, 'l2_leaf_reg': 1.496022363886513, 'baggi
ng_temperature': 0.0051605707874324835}. Best is trial 0 with value: 0.91755011
10826435.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:51:29,009] Trial 4 finished with value: 0.9100722141897524 and
parameters: {'loss_function': 'Logloss', 'iterations': 1095, 'learning_rate':
0.0010600256097692915, 'depth': 4, 'l2_leaf_reg': 9.048757289463383, 'bagging_t
emperature': 0.4546120803571043}. Best is trial 0 with value: 0.917550111082643
5.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:51:50,782] Trial 5 finished with value: 0.913133839804406 and
parameters: {'loss_function': 'Logloss', 'iterations': 1072, 'learning_rate':
0.03649890832710231, 'depth': 1, 'l2_leaf_reg': 1.9393740339888896, 'bagging_te
mperature': 0.6603851239649985}. Best is trial 0 with value: 0.917550111082643
5.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:53:40,161] Trial 6 finished with value: 0.9146444127495865 and
parameters: {'loss_function': 'Logloss', 'iterations': 1837, 'learning_rate':
0.004867071499671273, 'depth': 4, 'l2_leaf_reg': 4.86128494622777, 'bagging_tem
perature': 0.03656277683975395}. Best is trial 0 with value: 0.917550111082643
5.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:55:52,000] Trial 7 finished with value: 0.9154455110853155 and
parameters: {'loss_function': 'CrossEntropy', 'iterations': 1207, 'learning_rat
e': 0.006216737562148128, 'depth': 7, 'l2_leaf_reg': 0.1204526205350981, 'baggi
ng_temperature': 0.010524628012682724}. Best is trial 0 with value: 0.917550111
0826435.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:56:42,333] Trial 8 finished with value: 0.910884617872311 and
parameters: {'loss_function': 'Logloss', 'iterations': 783, 'learning_rate':
0.002084923902901007, 'depth': 4, 'l2_leaf_reg': 0.6039475580108802, 'bagging_t
emperature': 0.26688350516042636}. Best is trial 0 with value: 0.91755011108264
35.
```

```
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:56:51,017] Trial 9 finished with value: 0.7689293510506501 and
parameters: {'loss_function': 'CrossEntropy', 'iterations': 1408, 'learning_rat
e': 0.01212059674723709, 'depth': 1, 'l2_leaf_reg': 0.17549992598412598, 'baggi
ng_temperature': 0.6765531119010415}. Best is trial 0 with value: 0.91755011108
26435.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:57:28,602] Trial 10 finished with value: 0.9152344042406503 an
d parameters: {'loss_function': 'CrossEntropy', 'iterations': 228, 'learning_ra
te': 0.026063446323456006, 'depth': 8, 'l2_leaf_reg': 0.41461739354881316, 'bag
ging_temperature': 0.2546398993560476}. Best is trial 0 with value: 0.917550111
0826435.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:58:24,270] Trial 11 finished with value: 0.9182448929077252 an
d parameters: {'loss_function': 'Logloss', 'iterations': 548, 'learning_rate':
0.1104240654650708, 'depth': 6, 'l2_leaf_reg': 0.2781912801033739, 'bagging_tem
perature': 0.9922036779392186}. Best is trial 11 with value: 0.918244892907725
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:59:08,109] Trial 12 finished with value: 0.9178307459315588 an
d parameters: {'loss_function': 'Logloss', 'iterations': 403, 'learning_rate':
0.1069574040722355, 'depth': 6, 'l2_leaf_reg': 0.35839692704913584, 'bagging_te
mperature': 0.9518489092689479}. Best is trial 11 with value: 0.918244892907725
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 20:59:45,372] Trial 13 finished with value: 0.9178302049468052 an
d parameters: {'loss_function': 'Logloss', 'iterations': 332, 'learning_rate':
0.1346463791066095, 'depth': 6, 'l2_leaf_reg': 0.27478816921730664, 'bagging_te
mperature': 0.9764843703443615}. Best is trial 11 with value: 0.918244892907725
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:00:15,841] Trial 14 finished with value: 0.9172308282300218 an
d parameters: {'loss_function': 'Logloss', 'iterations': 524, 'learning_rate':
0.11089386405040938, 'depth': 3, 'l2_leaf_reg': 0.7511765629338125, 'bagging_te
mperature': 0.815695359062523}. Best is trial 11 with value: 0.918244892907725
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:01:13,422] Trial 15 finished with value: 0.9180692633317581 an
d parameters: {'loss_function': 'Logloss', 'iterations': 566, 'learning_rate':
0.07808425389675355, 'depth': 6, 'l2_leaf_reg': 0.1059159856250044, 'bagging_te
mperature': 0.8340010081884629}. Best is trial 11 with value: 0.918244892907725
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:02:10,905] Trial 16 finished with value: 0.917597366784268 and
parameters: {'loss_function': 'Logloss', 'iterations': 697, 'learning_rate':
0.05433802736673219, 'depth': 5, 'l2_leaf_reg': 0.10698862785535782, 'bagging_t
emperature': 0.810372554937836}. Best is trial 11 with value: 0.918244892907725
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:02:56,548] Trial 17 finished with value: 0.9152640395794498 an
d parameters: {'loss_function': 'Logloss', 'iterations': 910, 'learning_rate':
0.021194158157023118, 'depth': 3, 'l2_leaf_reg': 0.19376149496545064, 'baggin
g_temperature': 0.8583591643305146}. Best is trial 11 with value: 0.91824489290
77252.
```

```
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:04:02,206] Trial 18 finished with value: 0.9180293719890089 an
d parameters: {'loss_function': 'CrossEntropy', 'iterations': 551, 'learning_ra
te': 0.06776428310186453, 'depth': 7, 'l2_leaf_reg': 0.13734734391909623, 'bagg
ing_temperature': 0.6777948963468968}. Best is trial 11 with value: 0.918244892
9077252.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:06:15,704] Trial 19 finished with value: 0.9197395548608142 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1437, 'learning_rate':
0.07966798053955434, 'depth': 6, 'l2_leaf_reg': 3.2529943282619826, 'bagging_te
mperature': 0.5666654009465922}. Best is trial 19 with value: 0.919739554860814
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:08:14,409] Trial 20 finished with value: 0.9186511266040561 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1575, 'learning_rate':
0.037861941898056146, 'depth': 5, 'l2_leaf_reg': 2.8868117828712703, 'bagging_t
emperature': 0.5180320484734986}. Best is trial 19 with value: 0.91973955486081
42.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:10:07,441] Trial 21 finished with value: 0.9189116497826075 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1493, 'learning_rate':
0.04552061677848098, 'depth': 5, 'l2_leaf_reg': 3.308674368005426, 'bagging_tem
perature': 0.5713219560117918}. Best is trial 19 with value: 0.919739554860814
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:12:05,288] Trial 22 finished with value: 0.9187094824133881 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1553, 'learning_rate':
0.03800728918006622, 'depth': 5, 'l2_leaf_reg': 3.0938141291979804, 'bagging_te
mperature': 0.54692026685245}. Best is trial 19 with value: 0.9197395548608142.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:13:10,125] Trial 23 finished with value: 0.9155503267711045 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1417, 'learning_rate':
0.015415615888857355, 'depth': 3, 'l2_leaf_reg': 3.8006401737652364, 'bagging_t
emperature': 0.5510451747666425}. Best is trial 19 with value: 0.91973955486081
42.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:15:07,998] Trial 24 finished with value: 0.9183851455198284 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1558, 'learning_rate':
0.03355025842145218, 'depth': 5, 'l2_leaf_reg': 6.186281433429206, 'bagging_tem
perature': 0.6016230473681886}. Best is trial 19 with value: 0.919739554860814
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:15:51,117] Trial 25 finished with value: 0.9164884895683633 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1335, 'learning_rate':
0.04857753705210525, 'depth': 2, 'l2_leaf_reg': 2.675476895362044, 'bagging_tem
perature': 0.35635897605622624}. Best is trial 19 with value: 0.919739554860814
2.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:17:26,087] Trial 26 finished with value: 0.9170404787543118 an
d parameters: {'loss_function': 'CrossEntropy', 'iterations': 1609, 'learning_r
ate': 0.02096482577172055, 'depth': 4, 'l2_leaf_reg': 8.492700102280256, 'baggi
ng_temperature': 0.6128006409095074}. Best is trial 19 with value: 0.9197395548
608142.
Default metric period is 5 because AUC is/are not implemented for GPU
```

```
[I 2025-12-11 21:19:53,246] Trial 27 finished with value: 0.919942641066404 and
parameters: {'loss_function': 'Logloss', 'iterations': 1316, 'learning_rate':
0.08005052780883877, 'depth': 7, 'l2_leaf_reg': 1.8510863077392115, 'bagging_te
mperature': 0.39530782216742166}. Best is trial 27 with value: 0.91994264106640
4.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:22:14,703] Trial 28 finished with value: 0.9198128915829078 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1266, 'learning_rate':
0.08444892662750061, 'depth': 7, 'l2_leaf_reg': 1.6122264262519133, 'bagging_te
mperature': 0.36453317561803594}. Best is trial 27 with value: 0.91994264106640
4.
Default metric period is 5 because AUC is/are not implemented for GPU
[I 2025-12-11 21:24:31,001] Trial 29 finished with value: 0.9199865626794326 an
d parameters: {'loss_function': 'Logloss', 'iterations': 1234, 'learning_rate':
0.08239100882470687, 'depth': 7, 'l2_leaf_reg': 1.831066651742579, 'bagging_tem
perature': 0.18603106703265415}. Best is trial 29 with value: 0.919986562679432
6.
Optuna завершён
Лучший AUC (hold-out): 0.91999
Лучшие параметры:
{'loss_function': 'Logloss', 'iterations': 1234, 'learning_rate': 0.08239100882
470687, 'depth': 7, 'l2_leaf_reg': 1.831066651742579, 'bagging_temperature':
0.18603106703265415}

=== Fold 1/3 ===
Default metric period is 5 because AUC is/are not implemented for GPU
0:        test: 0.9049505        best: 0.9049505 (0)        total: 112ms
remaining: 2m 18s
200:        test: 0.9181073        best: 0.9181073 (200)        total: 17.3s
remaining: 1m 28s
400:        test: 0.9195079        best: 0.9195079 (400)        total: 33.9s
remaining: 1m 10s
600:        test: 0.9201159        best: 0.9201159 (600)        total: 51s
remaining: 53.8s
800:        test: 0.9204982        best: 0.9205081 (787)        total: 1m 8s
remaining: 36.8s
1000:        test: 0.9208233        best: 0.9208252 (986)        total: 1m 24s
remaining: 19.7s
1200:        test: 0.9209250        best: 0.9209278 (1199)        total: 1m 42s
remaining: 2.81s
1233:        test: 0.9209318        best: 0.9209355 (1227)        total: 1m 45s
remaining: 0us
bestTest = 0.9209355116
bestIteration = 1227
Shrink model to first 1228 iterations.
Fold AUC: 0.92094

=== Fold 2/3 ===
Default metric period is 5 because AUC is/are not implemented for GPU
```

```
0:        test: 0.9020538        best: 0.9020538 (0)        total: 112ms
remaining: 2m 18s
200:        test: 0.9163269        best: 0.9163269 (200)        total: 16.7s
remaining: 1m 26s
400:        test: 0.9178674        best: 0.9178686 (399)        total: 33.7s
remaining: 1m 9s
600:        test: 0.9184692        best: 0.9184768 (593)        total: 51.7s
remaining: 54.4s
800:        test: 0.9187582        best: 0.9187597 (795)        total: 1m 8s
remaining: 37s
1000:        test: 0.9189578        best: 0.9189625 (998)        total: 1m 26s
remaining: 20s
bestTest = 0.9191297293
bestIteration = 1113
Shrink model to first 1114 iterations.
Fold AUC: 0.91913

=== Fold 3/3 ===
Default metric period is 5 because AUC is/are not implemented for GPU
0:        test: 0.9051097        best: 0.9051097 (0)        total: 156ms
remaining: 3m 11s
200:        test: 0.9172566        best: 0.9172566 (200)        total: 16.8s
remaining: 1m 26s
400:        test: 0.9186828        best: 0.9186902 (398)        total: 33.4s
remaining: 1m 9s
600:        test: 0.9191728        best: 0.9191792 (597)        total: 50.9s
remaining: 53.6s
800:        test: 0.9195902        best: 0.9195902 (800)        total: 1m 7s
remaining: 36.5s
1000:        test: 0.9198743        best: 0.9198826 (994)        total: 1m 24s
remaining: 19.8s
bestTest = 0.919968605
bestIteration = 1063
Shrink model to first 1064 iterations.
Fold AUC: 0.91997

Итоговый CV AUC CatBoost: 0.92001

Submission сохранён: submission_catboost_gpu_optuna.csv
```

Результаты

- Private Score: 0.92156
- Public Score: 0.92087

CatBoost + LightGBM

Почему Pipeline избыточен

1. **StackingClassifier уже является своего рода pipeline** - он сам управляет потоком данных через базовые модели к мета-модели

2. **Категориальные признаки обрабатываются напрямую** - CatBoost и LightGBM работают с категориями нативно через параметры `cat_features` и `categorical_feature`, которые передаются в конструктор моделей

3. **Нет препроцессинга, который нужно применять последовательно** - данные уже прошли feature engineering (`train_fe`, `test_fe`), остаётся только преобразование типов в `category`, что делается один раз

4. **Pipeline не упростит код** - пришлось бы создавать обёртки для передачи `cat_features` и `categorical_feature`, что только усложнит структуру

5. **StackingClassifier + cross_val_predict уже обеспечивают всю нужную логику** - управление CV-фолдами, OOF предсказаниями и финальным обучением

In [ ]:
```python
import gc
import warnings

import lightgbm as lgb
import pandas as pd
from catboost import CatBoostClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import cross_val_predict

warnings.filterwarnings("ignore")

train_data = train_fe.copy()
test_data = test_fe.copy()
y = target.copy().reset_index(drop=True)
test_ids = test_ids.copy()

# Категориальные признаки
cat_cols = train_data.select_dtypes(include=["object", "category"]).columns.to
```

```python
# Привести object в category
for c in cat_cols:
    train_data[c] = train_data[c].astype("category")
    test_data[c] = test_data[c].astype("category")

cat_feature_indices = [train_data.columns.get_loc(c) for c in cat_cols]

print(f"Категориальных признаков: {len(cat_cols)}")
print(f"Размер train: {train_data.shape}, test: {test_data.shape}")

RND = 42

# CatBoost (GPU)
catboost_model = CatBoostClassifier(
    loss_function="Logloss",
    iterations=1500,
    learning_rate=0.08,
    depth=7,
    eval_metric="AUC",
    verbose=0,
    random_seed=RND,
    cat_features=cat_feature_indices,
    early_stopping_rounds=50,
    task_type="GPU",
    devices="0",
)

# LightGBM
lightgbm_model = lgb.LGBMClassifier(
    objective="binary",
    metric="auc",
    n_estimators=2000,
    learning_rate=0.05,
    verbosity=-1,
    random_state=RND,
    n_jobs=-1,
    categorical_feature=cat_cols,
    device_type="gpu",
    gpu_platform_id=0,
    gpu_device_id=0,
)

base_estimators = [
    ("catboost", catboost_model),
    ("lightgbm", lightgbm_model),
]

# Мета-модель
meta_model = LogisticRegression(
    C=1.0,
    random_state=RND,
    max_iter=1000,
    solver="lbfgs",
)

stacking_clf = StackingClassifier(
    estimators=base_estimators,
    final_estimator=meta_model,
    cv=3,
    stack_method="predict_proba",
    n_jobs=1,
    verbose=2,
    passthrough=False,
)

# Обучение стэкинга
stacking_clf.fit(train_data, y)

# Предсказание на test
final_pred = stacking_clf.predict_proba(test_data)[:, 1]

# Сохранение submission
submission = pd.DataFrame({"id": test_ids, "loan_paid_back": final_pred})
submission.to_csv("submission_sklearn_stacking_gpu.csv", index=False)
print("Saved: submission_sklearn_stacking_gpu.csv")

# Оценка качества через OOF предсказания
print("Оценка качества (OOF)")

oof_predictions = cross_val_predict(
    stacking_clf,
    train_data,
    y,
    cv=5,
    method="predict_proba",
    n_jobs=1,
    verbose=1,
)[:, 1]

oof_auc = roc_auc_score(y, oof_predictions)
print(f"OOF AUC: {oof_auc:.5f}")

gc.collect()
```

```
Категориальных признаков: 14
Размер train: (593994, 69), test: (254569, 69)
```

```
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  7.0min finished
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  3.9min finished
```

```
Saved: submission_sklearn_stacking_gpu.csv
Оценка качества (OOF)
```

```
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  6.1min finished
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  3.2min finished
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  5.8min finished
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  3.2min finished
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  5.9min finished
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  3.2min finished
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  5.9min finished
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  3.5min finished
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
Default metric period is 5 because AUC is/are not implemented for GPU
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  5.8min finished
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:  3.2min finished
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed: 65.6min finished
OOF AUC: 0.92009
```

```
Out[ ]: 371
```

- Private score: 0.92184
- Public score: 0.92104

LGBM

Почему Pipeline избыточен

1. **Кодирование делается один раз на train+test вместе** - это нельзя поместить в Pipeline, так как Pipeline применяет трансформации отдельно на каждом фолде CV

2. **LightGBM принимает `categorical_feature` как параметр fit()** -

Pipeline не умеет пробрасывать такие специфичные параметры через стандартный `.fit(X, y)`

3. **Для работы с Pipeline пришлось бы создавать обёртки-классы** - это усложняет код без реальной пользы

In [ ]:

```
import gc

import lightgbm as lgbm
import numpy as np
import optuna
import pandas as pd
from optuna.samplers import TPESampler
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import StratifiedKFold

# Подготовка данных после FE
X = train_fe.copy()
y = target.copy().reset_index(drop=True)
test_final = test_fe.copy()

# Кодирование категориальных признаков для LightGBM
cat_cols = X.select_dtypes(include=["object"]).columns.tolist()
print(f"Найдено категориальных признаков: {len(cat_cols)}")
print(f"Категории: {cat_cols}")

# объединяем train + test
for col in cat_cols:
    # Собираем все уникальные значения из train и test
    all_values = pd.concat([X[col], test_final[col]]).astype(str).unique()
    # Создаём маппинг: строка -> целое число
    label_to_id = {v: i for i, v in enumerate(all_values)}
    # Применяем маппинг
    X[col] = X[col].astype(str).map(label_to_id).astype("int32")
    test_final[col] = test_final[col].astype(str).map(label_to_id).astype("int

# индексы категориальных признаков
cat_col_indices = [X.columns.get_loc(c) for c in cat_cols]

# Функция для оптимизации
def objective(trial):
    params = {
        "objective": "binary",
        "metric": "auc",
        "verbosity": -1,
        "random_state": 42,
        "n_estimators": trial.suggest_int("n_estimators", 500, 5000, step=500)
        "learning_rate": trial.suggest_float("learning_rate", 0.001, 0.05, log
        "max_depth": trial.suggest_int("max_depth", 3, 8),
        "num_leaves": trial.suggest_int("num_leaves", 15, 128),
    }
    # CV
    num_folds = 3
    skf = StratifiedKFold(n_splits=num_folds, shuffle=True, random_state=42)

    oof_preds = np.zeros(len(X))
    fold_scores = []

    for fold, (train_idx, val_idx) in enumerate(skf.split(X, y)):
        X_train, X_val = X.iloc[train_idx], X.iloc[val_idx]
        y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]

        model = lgbm.LGBMClassifier(**params)

        model.fit(
            X_train,
            y_train,
            eval_set=[(X_val, y_val)],
            categorical_feature=cat_col_indices,
            callbacks=[lgbm.early_stopping(stopping_rounds=100, verbose=False)
        )

        val_pred = model.predict_proba(X_val)[:, 1]
        oof_preds[val_idx] = val_pred

        score = roc_auc_score(y_val, val_pred)
        fold_scores.append(score)

        del model, X_train, X_val, y_train, y_val
        gc.collect()

    cv_score = np.mean(fold_scores)

    # Логируем промежуточные результаты
    trial.set_user_attr("cv_std", np.std(fold_scores))
    trial.set_user_attr("fold_scores", fold_scores)

    return cv_score


study = optuna.create_study(
    direction="maximize", sampler=TPESampler(seed=42), study_name="lgbm_optimi
)

# Оптимизация
study.optimize(objective, n_trials=5, show_progress_bar=True)

# Результаты оптимизации
print(f"Лучший AUC: {study.best_value:.5f}")
print("Лучшие параметры:")
for key, value in study.best_params.items():
    print(f"  {key}: {value}")
```

```python
best_trial = study.best_trial
print(f"\nCV std: {best_trial.user_attrs['cv_std']:.5f}")
print(f"Fold scores: {[f'{s:.5f}' for s in best_trial.user_attrs['fold_scores'

# Обучение финальной модели с лучшими параметрами
best_params = study.best_params.copy()
best_params.update(
    {
        "objective": "binary",
        "metric": "auc",
        "verbosity": -1,
        "random_state": 42,
    }
)

num_folds = 3
skf = StratifiedKFold(n_splits=num_folds, shuffle=True, random_state=42)

oof_preds = np.zeros(len(X))
test_preds = np.zeros((len(test_final), num_folds))
scores = []


for fold, (train_idx, val_idx) in enumerate(skf.split(X, y)):
    print(f"\n--- Fold {fold + 1}/{num_folds} ---")

    X_train, X_val = X.iloc[train_idx], X.iloc[val_idx]
    y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]

    model = lgbm.LGBMClassifier(**best_params)

    model.fit(
        X_train,
        y_train,
        eval_set=[(X_val, y_val)],
        categorical_feature=cat_col_indices,
        callbacks=[
            lgbm.log_evaluation(500),
            lgbm.early_stopping(stopping_rounds=100, verbose=False),
        ],
    )

    # валидационные предсказания
    val_pred = model.predict_proba(X_val)[:, 1]
    oof_preds[val_idx] = val_pred

    # Тестовые предсказания
    test_pred = model.predict_proba(test_final)[:, 1]
    test_preds[:, fold] = test_pred

    score = roc_auc_score(y_val, val_pred)
    scores.append(score)
    print(f"Fold {fold + 1} AUC: {score:.5f}")
    del model, X_train, X_val, y_train, y_val
    gc.collect()


# Итоги
cv_mean = np.mean(scores)
cv_std = np.std(scores)
oof_auc = roc_auc_score(y, oof_preds)

print(f"CV AUC: {cv_mean:.5f} ± {cv_std:.5f}")
print(f"OOF AUC: {oof_auc:.5f}")

# Submission
submission = pd.DataFrame({"id": test_ids, "loan_paid_back": test_preds.mean(a
submission.to_csv("submission_lightgbm_optuna.csv", index=False)
print("Submission сохранён как 'submission_lightgbm_optuna.csv'")
```

```
Найдено категориальных признаков: 14
Категории: ['gender', 'marital_status', 'education_level', 'employment_status',
'loan_purpose', 'grade_subgrade', 'grade_letter', 'employment_education_level',
'employment_marital_status', 'employment_loan_purpose', 'employment_status_educ
ation_level_combo', 'employment_status_marital_status_combo', 'grade_subgrade_l
oan_purpose_combo', 'education_level_loan_purpose_combo']
```

```
[I 2025-12-26 11:42:47,673] A new study created in memory with name: lgbm_optim
ization
```

```
  0%|          | 0/5 [00:00<?, ?it/s]
```

```
[I 2025-12-26 11:48:15,493] Trial 0 finished with value: 0.9178164268892992 and
parameters: {'n_estimators': 2000, 'learning_rate': 0.04123206532618727, 'max_d
epth': 7, 'num_leaves': 83}. Best is trial 0 with value: 0.9178164268892992.
[I 2025-12-26 11:54:50,668] Trial 1 finished with value: 0.9073471513215755 and
parameters: {'n_estimators': 1000, 'learning_rate': 0.0018408992080552514, 'ma
x_depth': 3, 'num_leaves': 113}. Best is trial 0 with value: 0.917816426889299
2.
[I 2025-12-26 12:17:29,239] Trial 2 finished with value: 0.9182019456931712 and
parameters: {'n_estimators': 3500, 'learning_rate': 0.01595857358814127, 'max_d
epth': 3, 'num_leaves': 125}. Best is trial 2 with value: 0.9182019456931712.
[I 2025-12-26 12:51:11,296] Trial 3 finished with value: 0.9161938959416641 and
parameters: {'n_estimators': 4500, 'learning_rate': 0.002294868368113055, 'ma
x_depth': 4, 'num_leaves': 35}. Best is trial 2 with value: 0.9182019456931712.
[I 2025-12-26 13:08:08,534] Trial 4 finished with value: 0.917269217234773 and
parameters: {'n_estimators': 2000, 'learning_rate': 0.0077901431262762414, 'ma
x_depth': 5, 'num_leaves': 48}. Best is trial 2 with value: 0.9182019456931712.
Лучший AUC: 0.91820
Лучшие параметры:
  n_estimators: 3500
  learning_rate: 0.01595857358814127
  max_depth: 3
  num_leaves: 125

CV std: 0.00074
Fold scores: ['0.91874', '0.91716', '0.91871']

--- Fold 1/3 ---
[500]	valid_0's auc: 0.915186
[1000]	valid_0's auc: 0.916587
[1500]	valid_0's auc: 0.917085
[2000]	valid_0's auc: 0.917765
[2500]	valid_0's auc: 0.918136
[3000]	valid_0's auc: 0.91849
[3500]	valid_0's auc: 0.918734
Fold 1 AUC: 0.91874

--- Fold 2/3 ---
[500]	valid_0's auc: 0.913307
[1000]	valid_0's auc: 0.914701
[1500]	valid_0's auc: 0.915324
[2000]	valid_0's auc: 0.916082
[2500]	valid_0's auc: 0.916691
[3000]	valid_0's auc: 0.91701
[3500]	valid_0's auc: 0.917153
Fold 2 AUC: 0.91716

--- Fold 3/3 ---
[500]	valid_0's auc: 0.914542
[1000]	valid_0's auc: 0.915989
[1500]	valid_0's auc: 0.916582
[2000]	valid_0's auc: 0.917223
[2500]	valid_0's auc: 0.917844
[3000]	valid_0's auc: 0.918299
[3500]	valid_0's auc: 0.91871
```

```
Fold 3 AUC: 0.91871
CV AUC: 0.91820 ± 0.00074
OOF AUC: 0.91820
Submission сохранён как 'submission_lightgbm_optuna.csv'
```

- Public Score: 0.91943
- Private Score: 0.91998