



ОБРАЗОВАНИЕ И ТРУДОУСТРОЙСТВО

АНАЛИЗ ОКУПАЕМОСТИ ИНВЕСТИЦИЙ
В ВЫСШЕЕ ОБРАЗОВАНИЕ В США
НА ОСНОВЕ ДАННЫХ COLLEGE SCORECARD

НПМ-01-23

Гозенко Артём

Зайцева Пелагея

Кузнецова Елизавета

Цатурьян Лев

АКТУАЛЬНОСТЬ

- Высшее образование в США – дорогостоящая инвестиция, часто финансируемая за счет кредитов.
- Студенты заканчивают вузы с большими долгами.

Выбор колледжа становится **финансовым риском**.

Необходимо сделать осознанный финансовый выбор.

Инициатива Министерства образования США:
Создание набора данных College Scorecard для предоставления объективной информации.

КЛЮЧЕВОЙ ВОПРОС

«В каком колледже каждый доллар, вложенный в обучение, с наибольшей вероятностью конвертируется в высокий заработок выпускника?»

Наша задача **оценить финансовую выгодность** вложений в конкретный колледж и создать **рейтинг** колледжей США

Библиотеки для анализа и загрузка данных

```
import pandas as pd
import numpy as np

from catboost import CatBoostRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.impute import SimpleImputer
from sklearn.ensemble import HistGradientBoostingRegressor

import zipfile
import os
```

```
zip_path = "/content/College_Scorecard_Raw_Data_10032025.zip"
extract_path = "/content/scorecard_data"

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

os.listdir(extract_path)
```

1. Предобработка данных (pandas, SimpleImputer)
2. Построение моделей ML (CatBoost, HistGradientBoostingRegressor)
3. Оценка качества (метрики MAE и R²)
4. Работа с файлами (распаковка датасета)

College Scorecard

ВЫБОР ПРИЗНАКОВ

`TARGET = 'MD_EARN_WNE_P10'` Определение целевой переменной

<i>числовые признаки</i>	<i>Описание</i>	<i>Роль в прогнозировании дохода</i>
COSTT4_A	Стоимость обучения	Прямой показатель инвестиций
DEBT_MDN	Медианный долг	Финансовая нагрузка выпускников
SAT_AVG	Средний SAT	Академический уровень студентов
ADM_RATE	Процент принятых	Селективность колледжа
PCTPELL	Доля нуждающихся студентов	Социально-экономический состав
UGDS	Количество студентов	Размер учреждения

категориальные признаки (3 группы)	Что означает	Влияние на доход
CONTROL	Тип контроля (1- публичный, 2-частный неком, 3-частный ком)	Частные обычно дают более высокие доходы
LOCALE	Местоположение (город/пригород/ сельская местность)	Географические различия в зарплатах
REGION	Регион США (Новая Англия, Юг и т.д.)	Региональные экономические различия

- Из 2100+ столбцов остаются только 10 наиболее релевантных
- Размерность: (≈ 7800 строк \times 10 столбцов)
- Эффективность: Сокращение данных в 210 раз при сохранении информативности

Такой отбор эффективен

- **Избегаем переобучения.** Меньше признаков = более устойчивая модель
- **Убираем шум.** Многие из 2100 столбцов нерелевантны или дублируются
- **Интерпретируемость.** Можно понять, какие факторы действительно влияют на доход
- **Вычислительная эффективность.** Модель обучается быстрее

Какие признаки НЕ вошли и почему:

- Названия колледжей , которые не несут предсказательной силы для новых колледжей
- Конкретные программы. Это слишком детализировано для общей модели
- Разбивка по полу/расе.
- Исторические данные, т.к. мы фокусируемся на текущих показателях

ПОДГОТОВКА ДАННЫХ

```
df_model = df_model.replace('PS', np.nan)
```

В данных College Scorecard значения 'PS' означают, что данные скрыты для защиты приватности.
Заменяем на NaN для последующей обработки.

```
for col in numeric_features + [TARGET]:  
    df_model[col] = pd.to_numeric(df_model[col], errors='coerce')
```

Приведение числовых признаков
к типу float.

```
df_model = df_model.dropna(subset=[TARGET])  
print("Размер датасета после очистки:", df_model.shape)
```

Удаление строк без целевой переменной.
Удаление ≈1,300 строк, остается ≈6,500 вузов
с известными доходами выпускников.

```
numeric_imputer = SimpleImputer(strategy='median')  
df_model[numeric_features] = numeric_imputer.fit_transform(df_model[numeric_features])
```

Замена NaN на медианное значение по столбцу (Устойчива к выбросам)

```
categorical_imputer = SimpleImputer(strategy='most_frequent')  
df_model[categorical_features] = categorical_imputer.fit_transform(df_model[categorical_features])
```

Замена NaN на наиболее частую категорию (моду)

РЕЗУЛЬТАТЫ ОБРАБОТКИ ДАННЫХ

До обработки:

Размер: (7800, 10)

Пропуски: ~15-20% данных

Типы данных: Смешанные (строки, числа, 'PS')

После обработки:

Размер: (~6500, 10) # Удалено ~1300 строк без целевой переменной

Пропуски: 0% # Все NaN заполнены

Типы данных: Все числовые (float) и категориальные (int)

Готовность: 100% для машинного обучения

Gradient Boosting

*Цель: Создать модель для прогнозирования
доходов выпускников через 10 лет*

1. Разделяем на признаки (9 характеристик колледжа) и целевую переменную (Медианный доход выпускников через 10 лет)
2. Разделяем на обучающую и тестовую выборки. 80% и 20% соответственно.
Random state = 42: Гарантия одинаковых результатов при повторных запусках
3. Модель: HistGradientBoostingRegressor.

```
gbr = HistGradientBoostingRegressor(  
    max_depth=6,  
    learning_rate=0.05,  
    max_iter=300,  
    random_state=42)
```

4. Обучение и оценка модели.

```
gbr.fit(X_train, y_train)  
y_pred = gbr.predict(X_test)
```

R²: 0.732
MAE (\$): 5560.0

Модель объясняет 73.2% вариативности
доходов выпускников.
Средняя ошибка прогноза всего \$5,560

Создание финального рейтинга

```
df_rank = df_model.copy()
```

```
# Предсказанный доход
```

```
df_rank['PRED_EARN'] = gbr.predict(  
    df_rank[numeric_features + categorical_features])
```

- Модель gbr применяется ко всем 6,500+ колледжам
- Для каждого колледжа рассчитывается прогнозируемый доход через 10 лет
- Результат: Новый столбец PRED_EARN с предсказанными доходами

```
df_rank['ROI'] = df_rank['PRED_EARN'] / df_rank['COSTT4_A']
```

Расчет Return on Investment

```
# Убираем экстремальные значения
```

```
df_rank = df_rank[df_rank['COSTT4_A'] > 0]
```

ROI = Прогнозируемый годовой доход через 10 лет / Годовая стоимость обучения

```
#топ-20 по ROI  
top_20 = df_rank.sort_values('ROI', ascending=False).head(20)  
# добавляем названия университетов из исходного df  
top_20['INSTNM'] = df.loc[top_20.index, 'INSTNM']  
top_20[['INSTNM', 'COSTT4_A', 'DEBT_MDN', 'PRED_EARN', 'ROI']]
```

- Сортируем все колледжи по ROI от большего к меньшему
- Берем первые 20 - лучшие по соотношению "доход/стоимость"
- Добавляем названия для читаемости результатов

Финальный рейтинг

		INSTNM	COSTT4_A	DEBT_MDN	PRED_EARN	ROI
...	2202	United States Merchant Marine Academy	9547.0	6500.0	78236.647574	8.194893
	3542	Instituto Tecnologico de Puerto Rico-Recinto d...	4274.0	9833.0	24233.652024	5.670017
	2244	Cleveland Community College	5477.0	9833.0	30073.300810	5.490835
	2492	Ohio University-Eastern Campus	10263.0	15332.0	54500.789016	5.310415
	2495	Ohio University-Lancaster Campus	10267.0	15332.0	54198.011097	5.278856
	2011	CUNY Bernard M Baruch College	13521.0	10000.0	71033.654455	5.253580
	222	Canada College	9514.0	9272.0	49596.973509	5.213052
	3906	Carroll Community College	8611.0	7769.0	44766.527141	5.198761
	3942	South Texas College	7505.0	9833.0	38788.373171	5.168338
	1313	Frederick Community College	8886.0	5500.0	45836.993682	5.158338
	3119	Lamar State College-Orange	7609.0	8125.0	38644.054678	5.078730
	232	Cerritos College	8409.0	8350.0	42394.599357	5.041574
	2016	CUNY City College	13514.0	9775.0	68107.848771	5.039799
	353	Moorpark College	9685.0	7161.0	47321.394907	4.886050
	3127	College of the Mainland	8014.0	5156.0	38886.395062	4.852308
	435	Skyline College	10461.0	7500.0	50629.460275	4.839830
	2497	Ohio University-Zanesville Campus	10960.0	15332.0	52925.394688	4.828959
	722	Beulah Heights University	9992.0	38980.0	48040.251328	4.807871
	2494	Ohio University-Southern Campus	10397.0	15332.0	49918.935047	4.801283
	2431	Eastern Gateway Community College	8388.0	3500.0	39915.365680	4.758627

Реализация через Catboost и сравнение с Gradient Boosting

Создание модели CatBoostRegressor

```
cat_model = CatBoostRegressor(  
    iterations=500,  
    depth=6,  
    learning_rate=0.05,  
    loss_function='MAE',  
    random_seed=42,  
    verbose=False  
)
```

Обучение модели:
автоматически кодирует категории

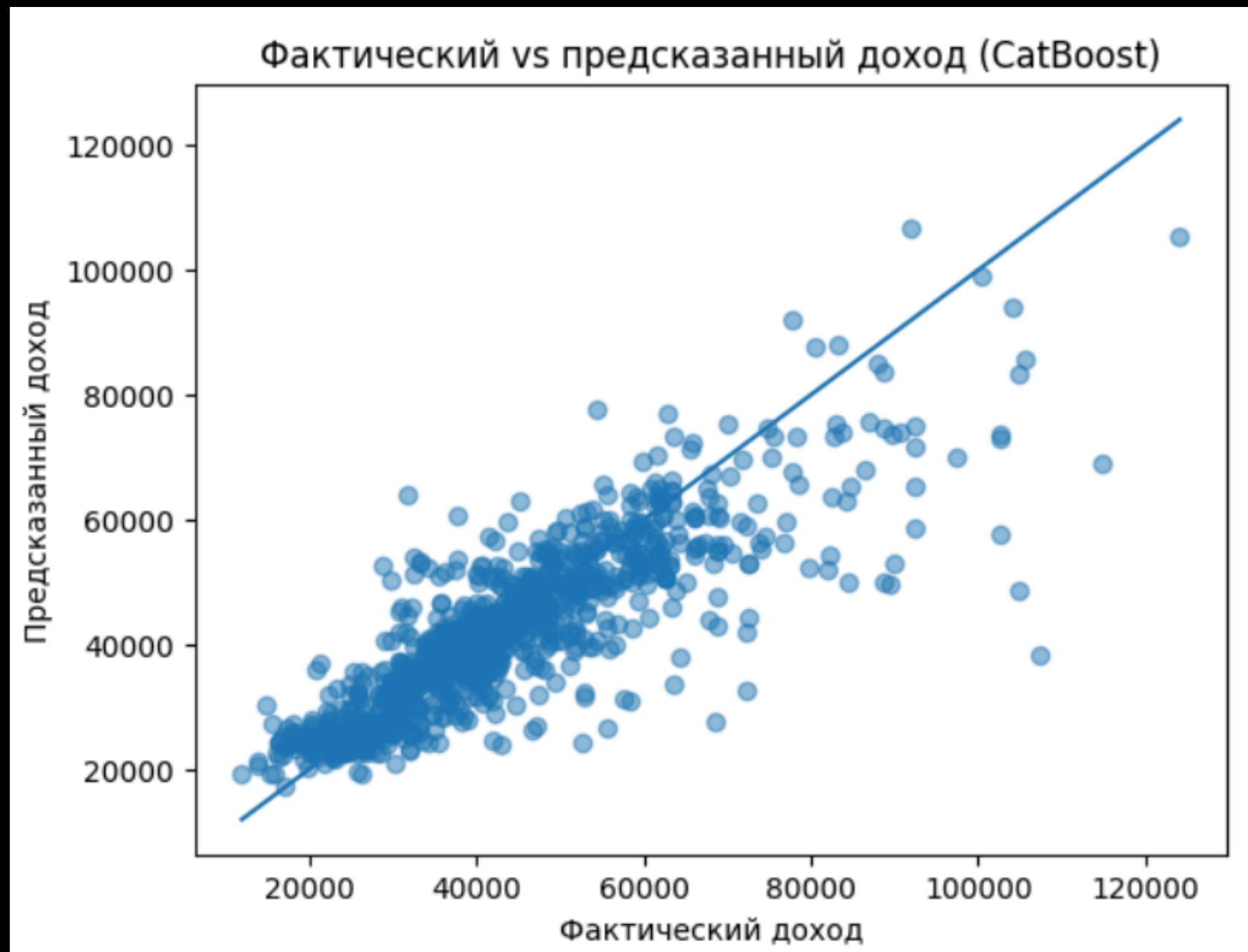
использует target statistics
строит ансамбль деревьев

```
cat_model.fit(  
    X_train, y_train,  
    cat_features=cat_features_idx  
)
```

...	Model	MAE	R2
0	HistGradientBoosting	5559.847719	0.731573
1	CatBoost	5671.125719	0.709588

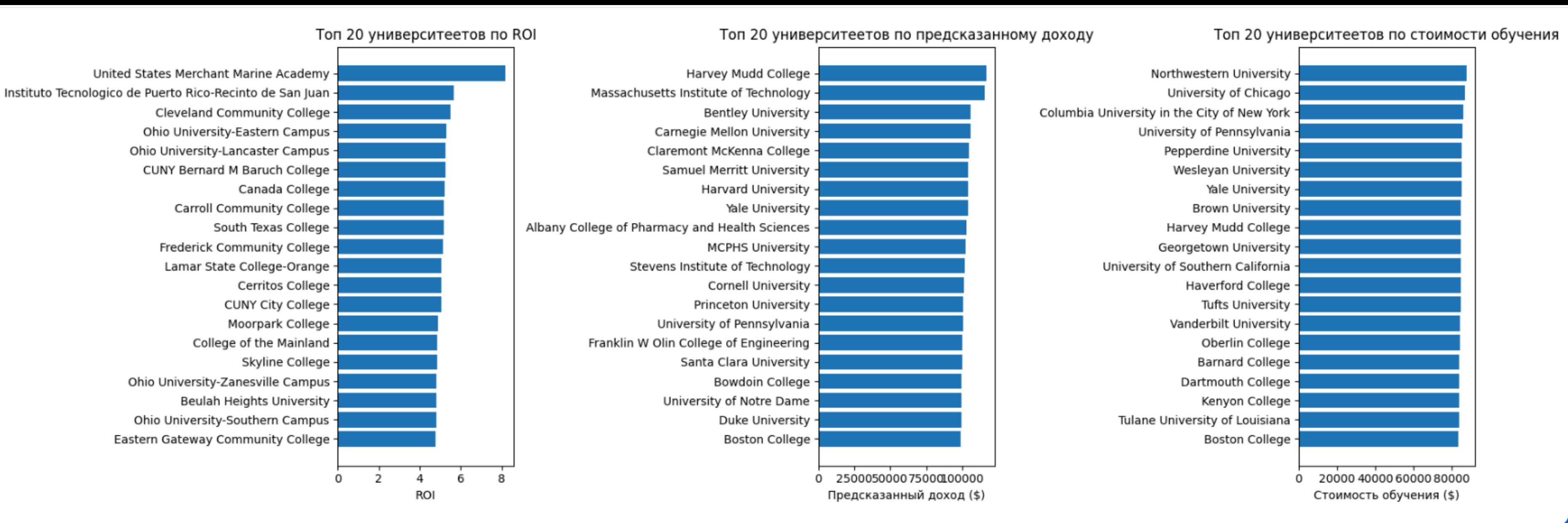
По результатам оценки на тестовой выборке модель HistGradientBoosting показала более высокое качество прогнозирования по сравнению с CatBoost (MAE ниже, R² выше).

Визуализации качества моделей



Формирование трёх рейтингов (Топ-20)

Построены рейтинги университетов по показателям окупаемости инвестиций (ROI), прогнозируемому доходу выпускников и стоимости обучения. Это позволяет комплексно оценить эффективность и экономическую целесообразность получения образования.



Кластеризация вузов по ROI и стоимости

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Берем признаки для кластеризации
cluster_features = df_rank[['ROI', 'COSTT4_A']].copy()

# Масштабирование
scaler = StandardScaler()
cluster_scaled = scaler.fit_transform(cluster_features)

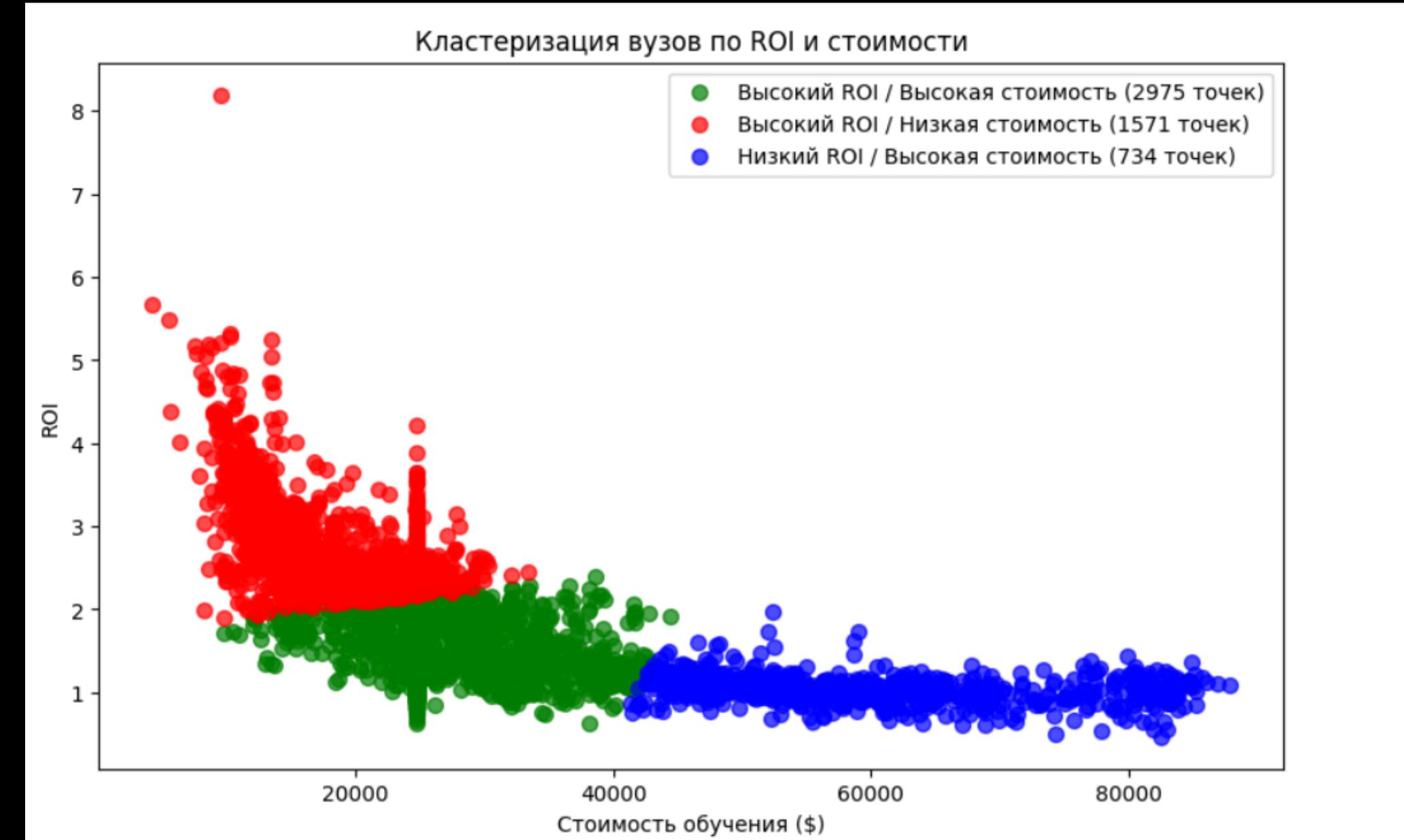
# KMeans с 3 кластерами
kmeans = KMeans(n_clusters=3, random_state=42)
df_rank['cluster'] = kmeans.fit_predict(cluster_scaled)

# Средние значения ROI и COST
cluster_means = df_rank.groupby('cluster')[['ROI', 'COSTT4_A']].mean()
```

ROI и COSTT4_A в разных единицах
KMeans использует евклидово расстояние
Без масштабирования стоимость перетянула бы всё на себя

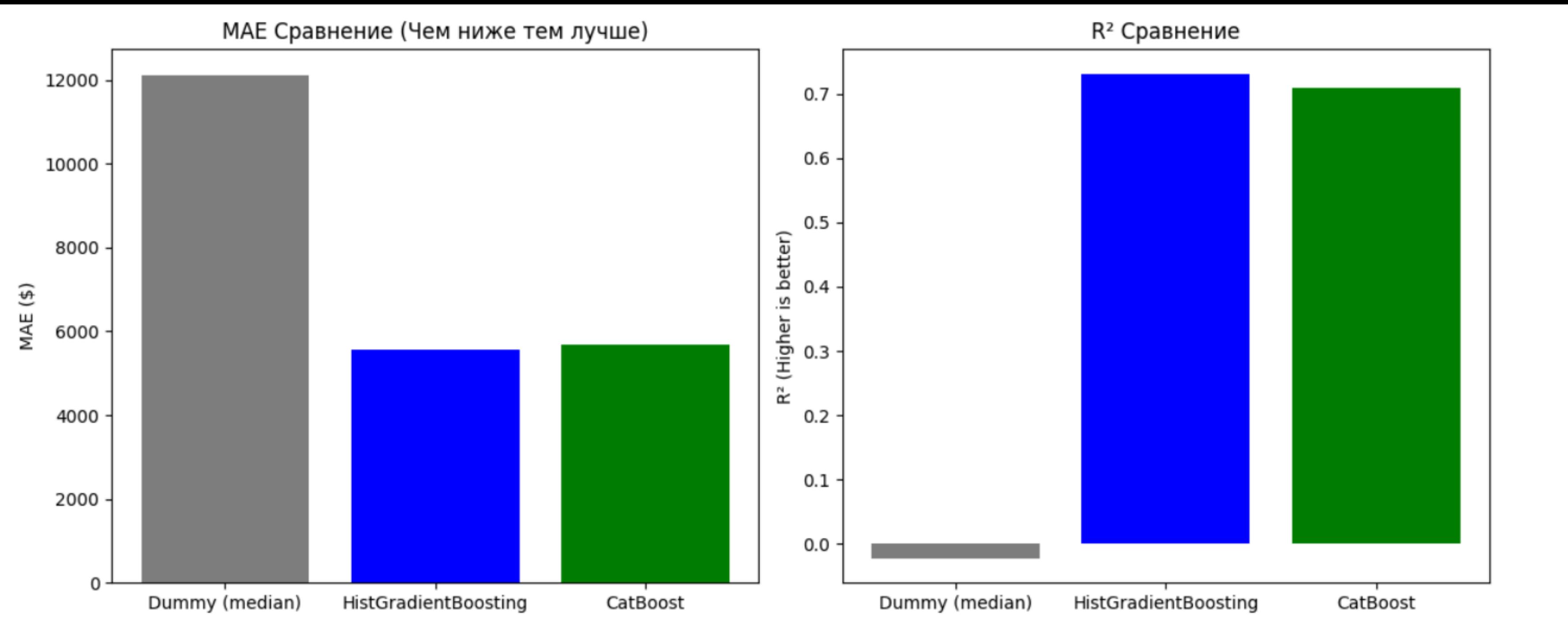
Теперь оба признака имеют:
среднее = 0
стандартное отклонение = 1

Каждый университет получает номер кластера: 0, 1 или 2

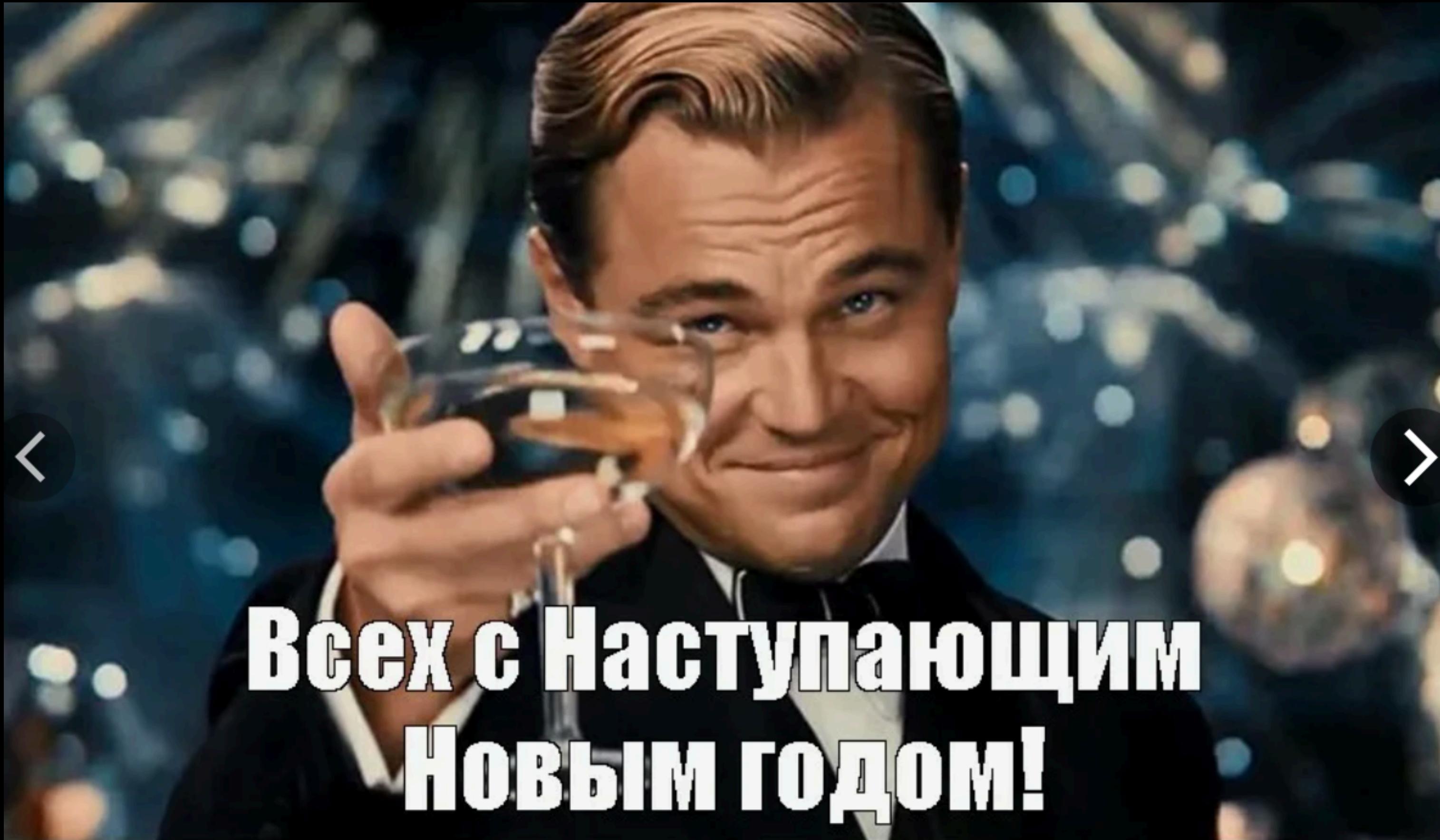


С помощью алгоритма KMeans выполнена кластеризация университетов по показателям стоимости обучения и окупаемости инвестиций. В результате выделены три группы вузов с различной экономической эффективностью образования.

Сравнение с медианным предсказанием



Сравнение моделей показало, что обе градиентные модели существенно превосходят базовый DummyRegressor. Наилучшие результаты по метрикам MAE и R² продемонстрировала модель HistGradientBoosting.



**Всех с Наступающим
Новым годом!**