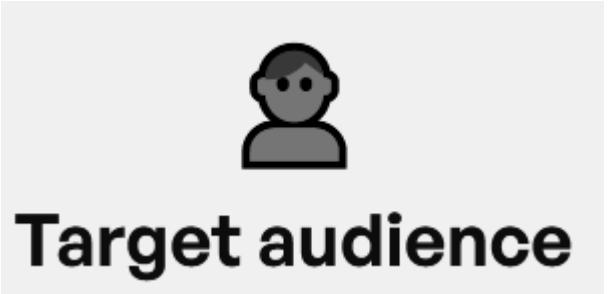
ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

Итоговая аттестация

по программе повышения квалификации
«Архитектор данных»
19.09.2023 – 01.11.2023

Слушатель: Наумова Елена Анатольевна

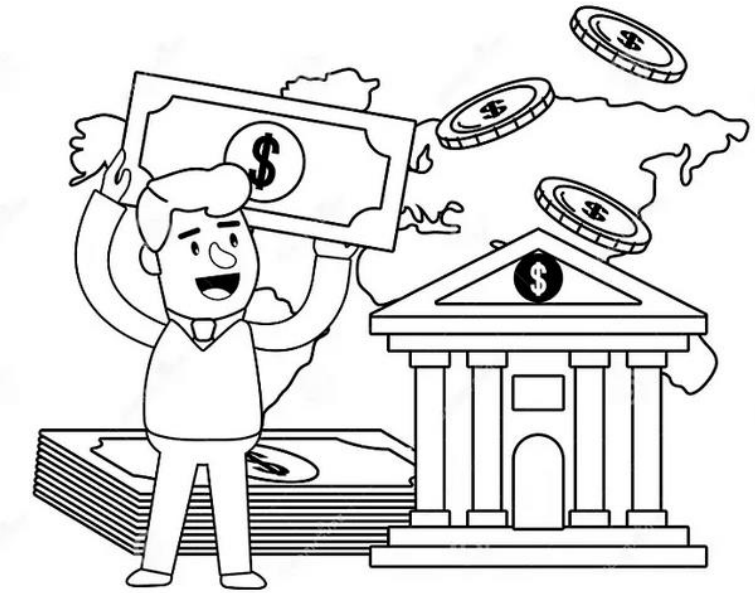
Контекст



Банк привлекает клиентов по различным каналам и собирает сведения о клиентах в базу данных.

Задачи банка:

- определить клиентов с высокой вероятностью конверсии,
- увеличить количество привлеченных квалифицированных клиентов, т.е. увеличить число потенциальных клиентов, попадающих в воронку конверсий.



<u>Текущая бизнес-задача:</u>

- определить, какая техника обработки данных приведет к большей результативности модели: undersampling, oversampling или SMOTE.

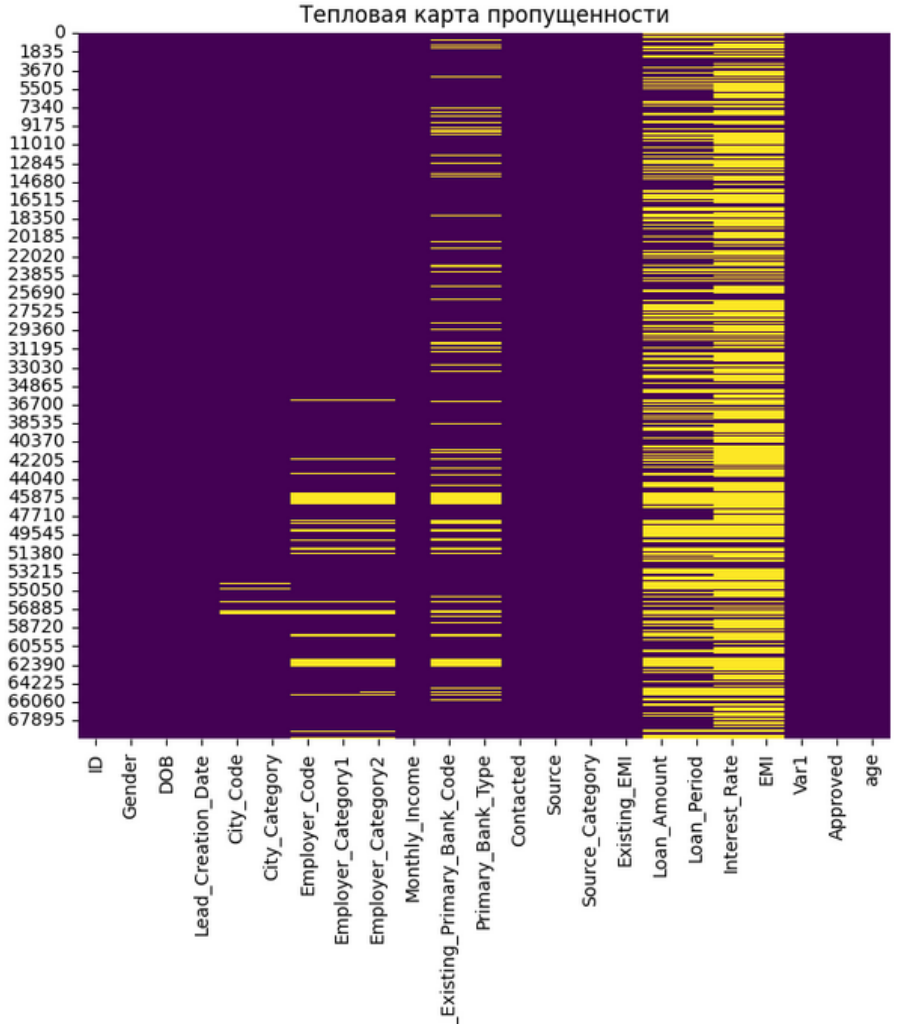
Данные

ID	Уникальный идентификатор клиента
Gender	Пол заявителя
DOB	Дата рождения заявителя
Lead_Creation_Date	Дата создания заявки
City_Code	Анонимизированный код города
City_Category	Анонимизированная характеристика города
Employer_Code	Анонимизированный код работодателя
Employer_Category1	Анонимизированная характеристика работодателя 1
Employer_Category2	Анонимизированная характеристика работодателя 2
Monthly_Income	Ежемесячный доход в долларах
Customer_Existing_Primary_Bank_Code	Анонимизированный код банка клиента
Primary_Bank_Type	Анонимизированная характеристика банка
Contacted	Проверка контакта (Y/N)
Source	Категориальная переменная, представляющая источник заявки
Source_Category	Тип источника
Existing_EMI	Ежемесячные платежи по существующим кредитам в долларах
Loan_Amount	Запрошенная сумма кредита
Loan_Period	Срок кредита (в годах)
Interest_Rate	Процентная ставка по запрошенной сумме кредита
EMI	Ежемесячный платеж по запрошенной сумме кредита в долларах
Var1	Анонимизированная категориальная переменная с несколькими уровнями
Approved	(Цель) Одобрен ли кредит или нет.(1-0) Клиент является квалифицированным клиентом или нет (1-0)

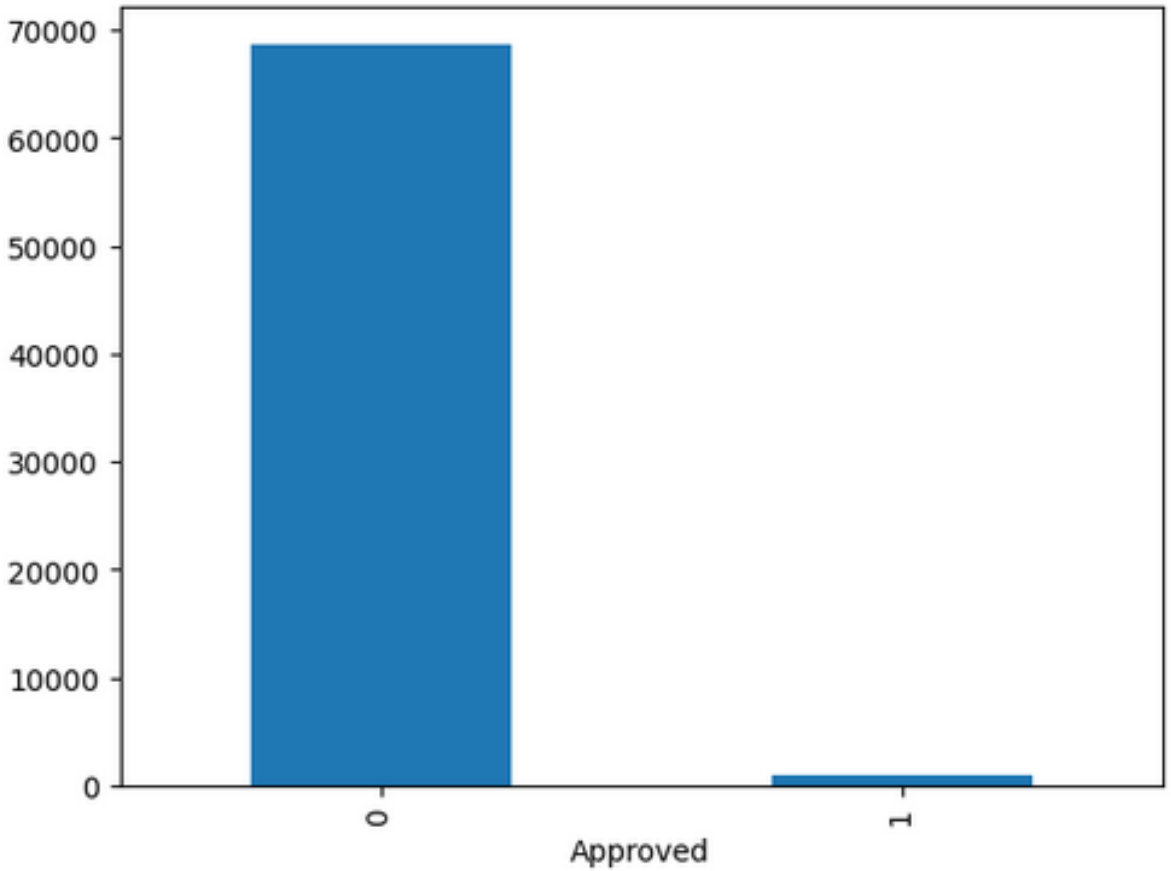
- Преобразование типов DOB и Lead_Creation_Data
- Создание нового признака age
- Подсчет пропущенных значений, дубликатов
- Исследование данных EDA
- Преобразование категориальных признаков в числовые с помощью LabelEncoder()
- Удаление признаков, где каждое значение уникально

Данные

Пропущено 11.42 % данных



Данные не сбалансированы.
«Approved» percentage = 1.46%.



Модель машинного обучения

Используем **логистическую регрессию** из библиотеки **sklearn**


```python
# Функция для обучения и проверки модели
def log_reg_model(X, y):
  # Разделение на обучающую и тестовую выборки
  X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                      test_size=0.2,
                                                      random_state=42)
  log_reg = LogisticRegression()
  log_reg.fit(X_train, y_train)
  y_pred = log_reg.predict(X_test)
  prediction=classification_report(y_test, y_pred)
  return prediction
```

Обработка пропусков

Удаление

```
# Оценка модели на данных после удаления пропусков
X_drop, y_drop = split_data(train_data_drop)
print(log_reg_model(X_drop, y_drop))
```

```
balanced_accuracy: 0.4996531791907514
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      4325
           1       0.00      0.00      0.00        92

    accuracy                           0.98      4417
   macro avg       0.49      0.50      0.49      4417
weighted avg       0.96      0.98      0.97      4417
```

Общие показатели высокие, но на классе 1 признака Approved нерезультативно.

Balanced_accuracy_score низкий.

Заполнение MICE

```
# Оценка модели на данных после заполнения пропусков MICE
train_data_mice = fit_encoded.data.data
X_mice, y_mice = split_data(train_data_mice)
print(log_reg_model(X_mice, y_mice))
```

```
balanced_accuracy: 0.5023907864169549
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     13737
           1       0.50      0.00      0.01       206

    accuracy                           0.99     13943
   macro avg       0.74      0.50      0.50     13943
weighted avg       0.98      0.99      0.98     13943
```

Показатели precision и f1-score класса 1 признака Approved улучшились.

Balanced_accuracy_score увеличился.

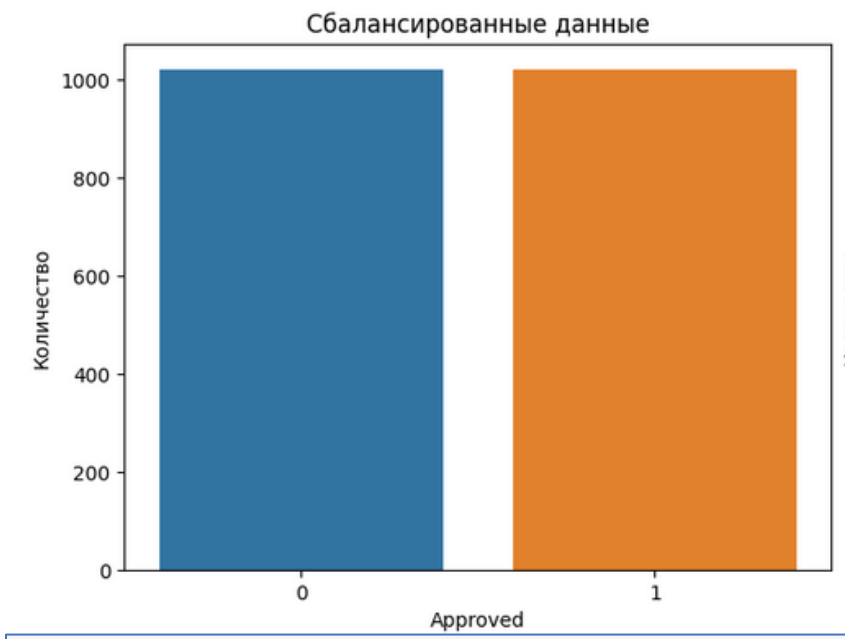
Модель работает лучше при заполнении mice.

Работа с несбалансированными данными

Undersampling reduces instances to balance classes

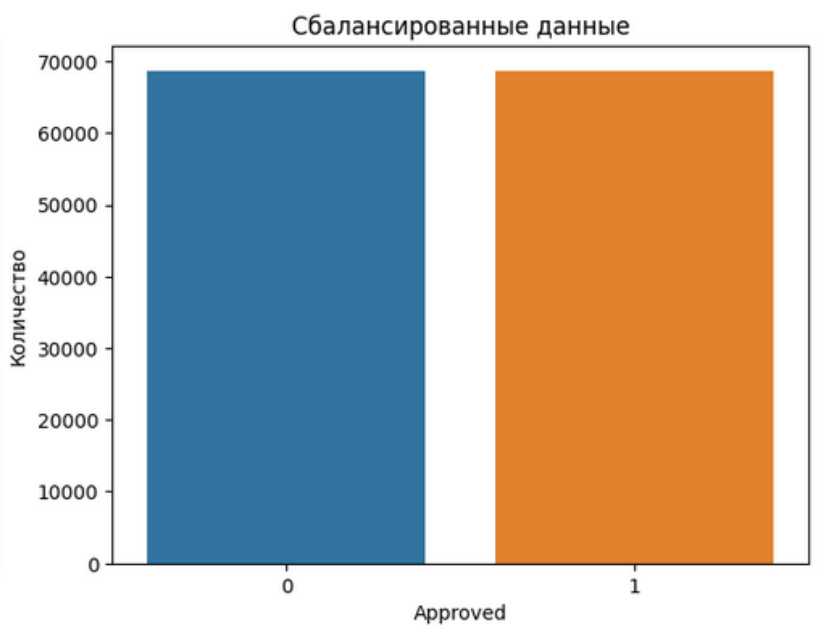
Oversampling replicates minority class instances

SMOTE synthetically generates minority class instances

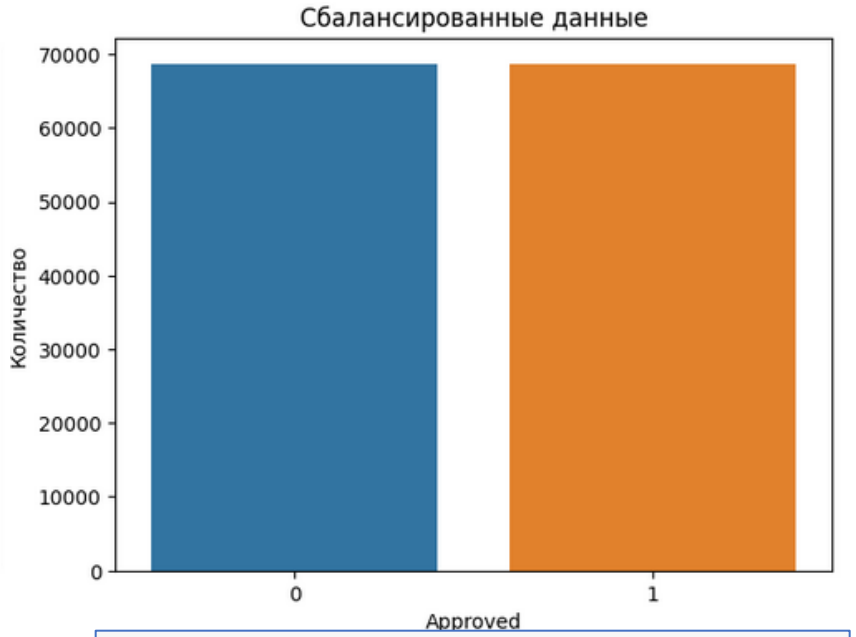



```
undersampler = RandomUnderSampler(random_state=42)
X_train_undersampled, y_train_undersampled = undersampler.fit_resample(X_mice,
                                                                       y_mice)

prediction_undersampled = log_reg_model(X_train_undersampled,
                                        y_train_undersampled)
print(prediction_undersampled)
```




```
oversampler = RandomOverSampler(random_state=42)
X_train_oversampled, y_train_oversampled = oversampler.fit_resample(X_mice,
                                                                    y_mice)
prediction_oversampled = log_reg_model(X_train_oversampled,
                                       y_train_oversampled)
print(prediction_oversampled)
```




```
smote = SMOTE(random_state=42)
X_smote, y_smote = smote.fit_resample(X_mice, y_mice)
prediction_smote = log_reg_model(X_smote, y_smote)
print(prediction_smote)
```

Сравнение результатов модели

Undersampling reduces instances to balance classes

Oversampling replicates minority class instances

SMOTE synthetically generates minority class instances

	precision	recall	f1-score	support
0	0.72	0.70	0.71	209
1	0.69	0.71	0.70	199
accuracy			0.71	408
macro avg	0.71	0.71	0.71	408
weighted avg	0.71	0.71	0.71	408

	precision	recall	f1-score	support
0	0.68	0.74	0.71	13823
1	0.71	0.65	0.68	13655
accuracy			0.69	27478
macro avg	0.69	0.69	0.69	27478
weighted avg	0.69	0.69	0.69	27478

	precision	recall	f1-score	support
0	0.69	0.72	0.70	13823
1	0.70	0.68	0.69	13655
accuracy			0.70	27478
macro avg	0.70	0.70	0.70	27478
weighted avg	0.70	0.70	0.70	27478

	precision	recall	f1-score	support
0	0.71	0.70	0.71	209
1	0.69	0.70	0.70	199
accuracy			0.70	408
macro avg	0.70	0.70	0.70	408
weighted avg	0.70	0.70	0.70	408

	precision	recall	f1-score	support
0	0.66	0.74	0.70	13823
1	0.70	0.61	0.65	13655
accuracy			0.68	27478
macro avg	0.68	0.67	0.67	27478
weighted avg	0.68	0.68	0.67	27478

	precision	recall	f1-score	support
0	0.71	0.71	0.71	13823
1	0.70	0.70	0.70	13655
accuracy			0.71	27478
macro avg	0.71	0.71	0.71	27478
weighted avg	0.71	0.71	0.71	27478

	precision	recall	f1-score	support
0	0.72	0.71	0.72	209
1	0.70	0.71	0.71	199
accuracy			0.71	408
macro avg	0.71	0.71	0.71	408
weighted avg	0.71	0.71	0.71	408

	precision	recall	f1-score	support
0	0.69	0.74	0.71	13823
1	0.71	0.67	0.69	13655
accuracy			0.70	27478
macro avg	0.70	0.70	0.70	27478
weighted avg	0.70	0.70	0.70	27478

	precision	recall	f1-score	support
0	0.70	0.70	0.70	13823
1	0.69	0.70	0.70	13655
accuracy			0.70	27478
macro avg	0.70	0.70	0.70	27478
weighted avg	0.70	0.70	0.70	27478

Вывод

- **Undersampling** и **SMOTE** оказали более положительное влияние на результат работы модели для класса 1 по сравнению с **oversampling.**
- Метрики в совокупности при сравнении нескольких итераций модели указывают на более сбалансированный результат работы модели для класса 1 после применения **undersampling.**
- Однако, разница в метриках между этими методами не является значительной.