МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)

КУРСОВАЯ РАБОТА

по дисциплине «Прогнозно-аналитические системы»

Тема курсовой работы: «Разработка сценария анализа и обработки данных на примере задачи оттока клиентов банка с применением логистической модели»

Студент группы ИМБО-02-22	Ким Кирилл Сергеевич	(подпись)
Руководитель курсовой работы	старший преподаватель кафедры ПМ, Юрченков И.А.	(подпись)
Работа представлена к защите	«___»_____2025 г.	
Допущен к защите	«___»_____2025 г.	

Москва 2025 г.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)

Утверждаю
Заведующий кафедрой ПМ
_____ Смоленцева Т.Е.
(подпись)
«22» сентября 2025 г.

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине «Прогнозно-аналитические системы»

Студент Ким Кирилл Сергеевич Группа ИМБО-02-22

Тема «Разработка сценария анализа и обработки данных на примере задачи оттока клиентов банка с применением логистической модели»

Исходные данные: собранный студентом набор данных по теме работы
Перечень вопросов, подлежащих разработке, и обязательного графического материала:
Описание исследуемой предметной области, применяемого алгоритма и набора данных (включает анализ текущего состояния изучаемой области, определение перспективных направлений исследований, оценку применимости алгоритмов анализа и обработки данных, а также характеристику полей набора данных). Математическая формулировка предлагаемого метода анализа и обработки данных (классическая постановка задачи, формулировка задачи статистической обработки данных, описание параметров, описание критерия качества решения конечной задачи).
Анализ собранных данных с использованием рассматриваемых в работе методов анализа и обработки данных (описание последовательности действий или сценария обработки данных, численные метрики)
Построение визуализаций и качественных выводов по проделанной работе
Срок представления к защите курсовой работы: **до** «19» декабря 2025 г.

Задание на курсовую работу выдал		Юрченков И.А.
	Подпись руководителя	*(ФИО руководителя)*
		«19» сентября 2025 г.
Задание на курсовую работу получил		Ким К.С.
	Подпись обучающегося	*(ФИО обучающегося)*
		«19» сентября 2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современных условиях высокой конкуренции на финансовом рынке проблема оттока клиентов является одной из наиболее актуальных для банков. Потеря клиентов ведет к прямым убыткам, увеличению затрат на привлечение новых клиентов и снижению доходности бизнеса. Согласно исследованиям, привлечение нового клиента обходится в 5-7 раз дороже, чем удержание существующего.

Актуальность проблемы обусловлена необходимостью разработки эффективных методов прогнозирования оттока клиентов, что позволяет банкам своевременно принимать превентивные меры по удержанию ценных клиентов и оптимизировать маркетинговые стратегии.

Целью данной работы является разработка сценария анализа и обработки данных для прогнозирования оттока клиентов банка с использованием логистической регрессии.

Задачи работы:

- Исследовать данные о клиентах банка и выявить факторы, влияющие на отток
- Провести предобработку данных и создать новые признаки
- Построить и оценить модель логистической регрессии
- Проанализировать важность признаков и сделать выводы
- Разработать систему скоринга для сегментации клиентов по риску оттока
- Предложить практические рекомендации для банка

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Описание предметной области и постановка задачи

Отток клиентов — это явление, когда клиенты прекращают пользоваться услугами компании. В банковской сфере отток может проявляться в различных формах:

- Закрытие счетов и перевод средств в другие банки
- Прекращение использования банковских продуктов
- Снижение активности по имеющимся продуктам
- Отказ от продления договоров на обслуживание

Математическая постановка задачи:

Задача прогнозирования оттока является задачей бинарной классификации, где:

- y=1 — клиент уйдет,
- y=0 — клиент останется.

Критерий качества:

Оценка качества модели осуществляется с помощью метрик:

- Accuracy (точность) — общая доля правильных прогнозов
- Precision (точность) — доля истинных отток клиентов среди спрогнозированных
- Recall (полнота) — доля обнаруженных реальных отток клиентов
- F1-score — гармоническое среднее между precision и recall
- ROC-AUC — площадь под кривой ошибок, характеризующая общее качество классификатора
- PR-AUC — площадь под Precision-Recall кривой, важная для несбалансированных данных

1.2 Логистическая регрессия

Логистическая регрессия является одним из наиболее широко применяемых алгоритмов для решения задач бинарной классификации в различных областях, включая метеорологию. Несмотря на название "регрессия", этот метод решает именно задачу классификации, оценивая вероятность принадлежности объекта к определенному классу.

Математические основы логистической регрессии:

Основой алгоритма является логистическая функция (сигмоида), которая имеет вид:

$$f(z) = \frac{1}{1 + e^{-z}},$$

где $z = \theta^T x,$

θ — вектор-столбец параметров (весов) логистической регрессии,

x — вектор-столбец независимых переменных.

Логистическая функция преобразует линейную комбинацию в вероятность принадлежности к целевому классу, принимающую значения в диапазоне от 0 до 1.

Функция потерь и оптимизация:

Для обучения модели логистической регрессии используется функция log loss (логистические потери):

$$\mathrm{L(X, y)} = -\frac{1}{n}\sum(y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)) \to min,$$

где p_i — вероятность принадлежности к целевому классу i-ого объекта,

y_i — целевое значение i-ого объекта,

n — количество объектов.

Минимизация функции потерь осуществляется с помощью методов оптимизации, таких как градиентный спуск:

$$\theta^t = \theta^{t-1} + \varepsilon \nabla_\theta \tilde{Q}(\theta, X),$$

где $\nabla_\theta \tilde{Q}(\theta, X)$ — градиент для функции потерь Log Loss,

ε — шаг спуска.

На практике чтобы не возникло явление переобучения, когда модель хорошо справляется с данными из обучающей выборки, но плохо работает на новых данных. Это происходит из-за того, что модель запоминает ненужные детали обучающих данных, вместо того чтобы обобщать общие закономерности. Переобучение в большинстве случаев проявляется в том, что в получающихся полиномах слишком большие коэффициенты.

Для улучшения обобщающей способности получающейся модели, то есть уменьшения эффекта переобучения, часто рассматривается логистическая регрессия с регуляризацией.

Регуляризация — метод управления сложностью модели путем добавления дополнительных ограничений к условию задачи. Это помогает избежать переобучения и решить некорректно поставленные задачи.

В случае логистической регрессии вектор параметров θ рассматривается как случайный вектор с некоторой заданной априорной плотностью распределения p(θ). В байесовском статистическом выводе априорное распределение вероятностей неопределённой величины p — распределение вероятностей, которое выражает предположения о p до учёта экспериментальных данных. Апостериорная вероятность — условная вероятность случайного события при условии того, что известны апостериорные данные. Для обучения модели вместо метода наибольшего правдоподобия используется метод максимизации апостериорной оценки, то есть ищутся параметры θ, максимизирующие величину на основе метода максимального правдоподобия, но дополнительно при оптимизации использует априорное распределение величины, которую оценивает:

$$L(X, y) = \prod p\{y^i | x^i, \theta\} * p(\theta),$$

где x_i — признаковое описание i-ого объекта,

θ — вектор-столбец весов,

y_i — целевое значение i-ого объекта,

$p(\theta)$ — априорной плотностью распределения.

В качестве априорного распределения выступает многомерное нормальное распределение $N(0, \sigma^2 I)$ с нулевым средним и матрицей ковариации $\sigma^2 I$ соответствующее априорному убеждению о том, что все коэффициенты регрессии должны быть небольшими числами, идеально — многие малозначимые коэффициенты должны быть нулями. Подставив плотность этого априорного распределения в формулу выше, и прологарифмировав, получим следующую оптимизационную задачу:

$$L(X, y) = \sum \log\left(p\{y^i | x^i, \theta\} - \alpha\|\theta\|^2\right) \to max$$

где x_i — признаковое описание i-ого объекта,

θ — вектор-столбец весов,

y_i — целевое значение i-ого объекта,

α — параметр регуляризации.

Этот метод известен как L2-регуляризованная логистическая регрессия, так как в целевую функцию входит L2-норма вектора параметров для регуляризации. При L2 регуляризации признаки сглаживаются, а не отбрасываются вовсе, как при L1 регуляризации.

А теперь перейдем к применению рассмотренного метода для разработки программы классификации данных на базе алгоритмов логистической регрессии и анализа весовых коэффициентов модели на примере данных физических характеристик для оценки качества яблок

С помощью данных показателей, можно узнать точность модели.

Меткость измерений (Accuracy) — это показатель, используемый для оценки производительности модели на основе предсказанных меток классов.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)},$$

где TP — True Positives (истинно положительные),

TN — True Negatives (истинно отрицательные),

FP — False Positives (ложноположительные),

FN — False Negatives (ложноотрицательные).

Несмотря на то, что эта мера хорошо интерпретируется, на практике она используется достаточно редко, поскольку плохо работает в случае дисбаланса классов в обучающей выборке.

Точность равна доле истинно положительных классификаций к общему числу положительных классификаций. Данная величина часто упоминается как положительное прогностическое значение (PPV).

$$PPV = \frac{TP}{TP + FP},$$

где TP — True Positives (истинно положительные),

FP — False Positives (ложноположительные).

Полнота — доля истинно положительных примеров (TPR), определяется как число истинно положительных классификаций относительно общего числа положительных наблюдений:

$$TPR = \frac{TP}{TP + FN},$$

где TP — True Positives (истинно положительные),

FN — False Negatives (ложноотрицательные).

Для каждого класса легко вычислить точность и полноту с помощью матрицы ошибок. Точность определяется как отношение верно предсказанных положительных случаев к общему количеству предсказанных положительных

случаев, а полнота — как отношение верно предсказанных положительных случаев к общему количеству истинных положительных случаев.

Точность и полнота не зависят от соотношения классов и могут быть использованы даже при несбалансированных выборках. Высокие значения точности и полноты указывают на хорошую модель, но обычно нельзя максимизировать обе метрики одновременно. Для нахождения баланса между ними используется F1-мера, которая учитывает и точность, и полноту. Она вычисляется как гармоническое среднее между точностью и полнотой.

$$F1 = \frac{TP + TP}{TP + TP + FP + FN},$$

где TP — True Positives (истинно положительные),

FP — False Positives (ложноположительные),

FN — False Negatives (ложноотрицательные).

Преимущества логистической регрессии для прогнозирования отток клиентов:

- Интерпретируемость — коэффициенты модели имеют четкую вероятностную интерпретацию
- Вычислительная эффективность — быстрое обучение и прогнозирование
- Устойчивость к шуму — хорошо работает с зашумленными данными
- Калиброванность вероятностей — выходные значения являются истинными вероятностями
- Возможность регуляризации — позволяет бороться с переобучением

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Подготовка данных

Набор данных с Kaggle: Churn for Bank Customers. Содержит информацию о клиентах банка (https://www.kaggle.com/datasets/mathchi/churn-for-bank-customers/data).

Основные поля:

- RowNumber, CustomerId, Surname — идентификаторы клиентов.
- CreditScore — кредитный рейтинг.
- Geography — страна клиента.
- Gender — пол.
- Age — возраст.
- Tenure — сколько лет клиент с банком.
- Balance — баланс счета.
- NumOfProducts — число продуктов.
- HasCrCard — наличие кредитной карты (0/1).
- IsActiveMember — активность (0/1).
- EstimatedSalary — предполагаемая зарплата.
- Exited — целевая переменная (0 — остался, 1 — ушел).

На Рисунке 2.1 представлены первые 5 записей импортированного набора данных с целевой переменной.

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Рисунок 1.1 — Фрагмент набора данных

В первую очередь проанализируем данные для построения модели логистической регрессии.

На Рисунке 2.2 представлены распределение клиентов по оттоку.

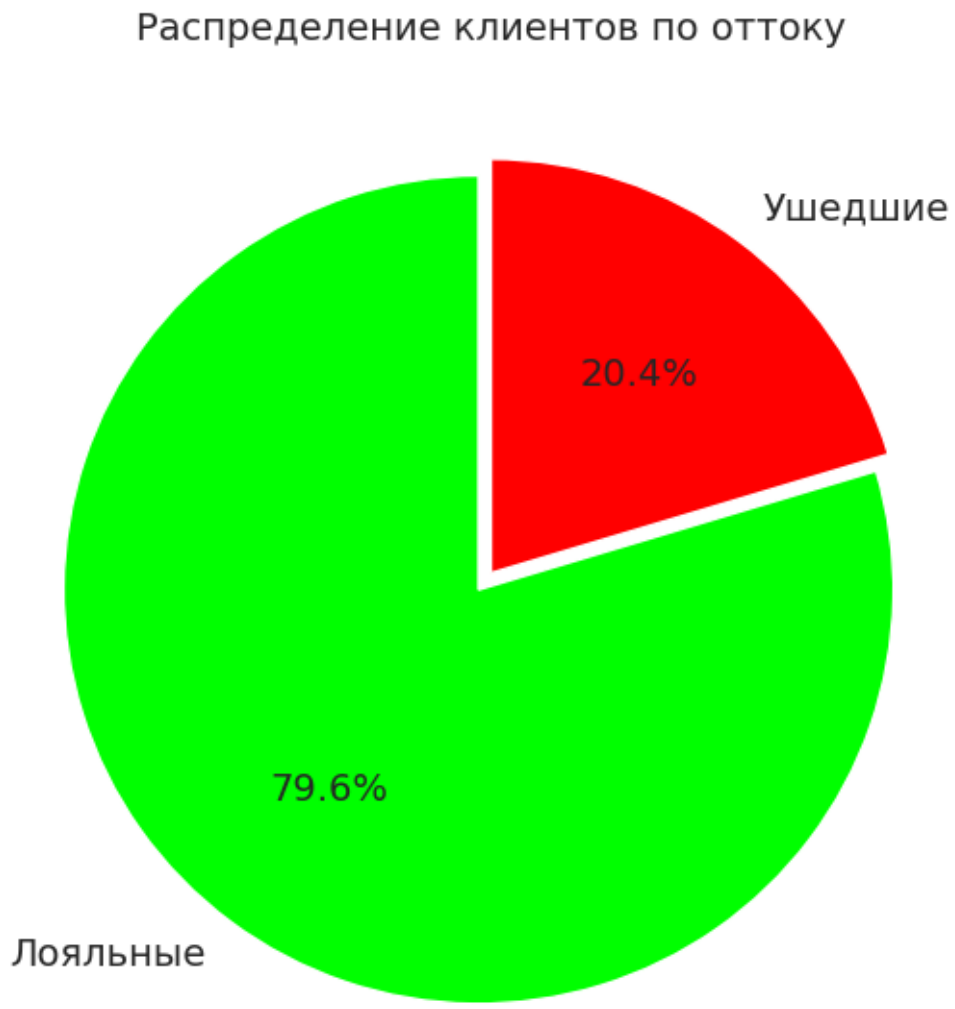

Рисунок 2.2 — Круговая диаграмма распределения клиентов по оттоку

Посмотрим внимательнее на распределения оттоков клиентов по странам, где больше. И в какой стране больше буду анализировать. Наибольший отток наблюдается в Германии (32.4%), затем идет Франция (16.1%) и Испания (16.7%). Результат представлен на Рисунке 2.3.

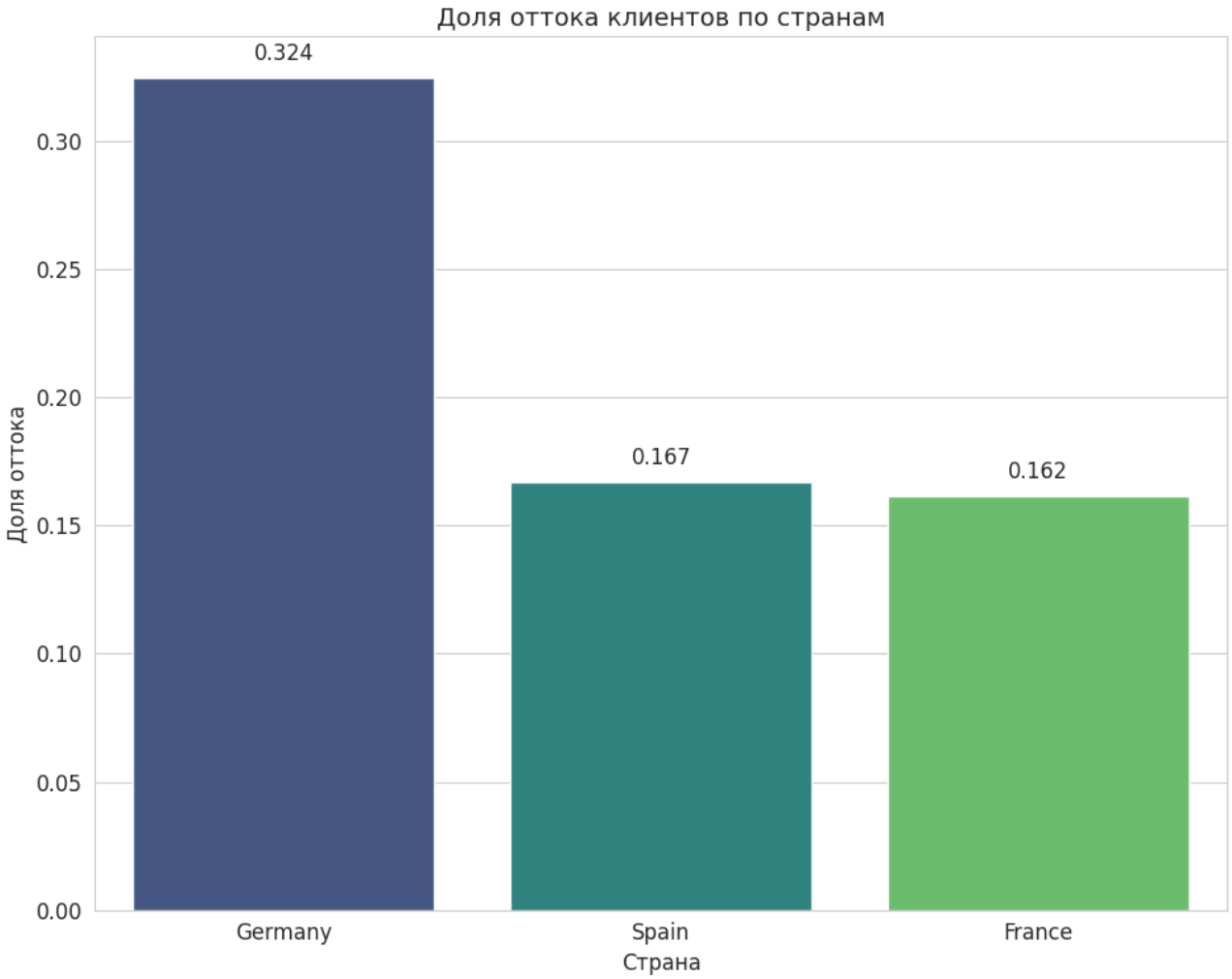

Рисунок 2.3 — Доля оттока клиентов по странам

Предобработка данных, фильтруем данные, оставляя только клиентов из Германии. Удаляем ненужные столбцы и преобразуем из текстовых значений в числа (Male – 1, Female – 0) с помощью LabelEncoder.

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7	376	0	29	4	115046.74	4	1	0	119346.88	1
15	616	1	45	3	143129.41	2	0	1	64327.26	0
16	653	1	58	1	132602.88	1	1	0	5097.67	1
26	756	1	36	2	136815.64	1	1	1	170041.95	0
28	574	0	43	3	141349.43	1	1	1	100187.43	0

Рисунок 2.4 — Предобработанные данные

Теперь, когда данные предобработаны и очищены, мы можем приступить к разработке программы классификации и оценке качества модели.

2.2 Разработка программы логистической модели

Для начала разделим наш набор данных на обучающий и тестовый для независимой оценки. На обучающий отведем 70% данных, на тестовый — 30%.

Нормализация числовых признаков с использованием StandardScaler.

Для обучения модели использовалась логистическая регрессия с L2-регуляризацией

После обучения и тестирования модели рассчитаем метрики качества (Рисунок 2.5).

```
Accuracy: 0.6706507304116865
Precision: 0.49363057324840764
Recall: 0.6352459016393442
F1: 0.5555555555555556
ROC-AUC: 0.7198460497922639
PR-AUC: 0.5329887037693996
Логарифмическая функция потерь: 0.613944144441652
```

Рисунок 2.5 — Метрики модели

Получились неплохие показатели точность модели (67%), ROC-AUC (55%), посмотрим, как выглядит матрица ошибок, которая покажет, где модель сработала неправильно. Матрица ошибок представлена на Рисунке 2.6.

Рисунок 2.6 — Матрица ошибок

Построил ROC-кривую представлено на Рисунке 2.7.

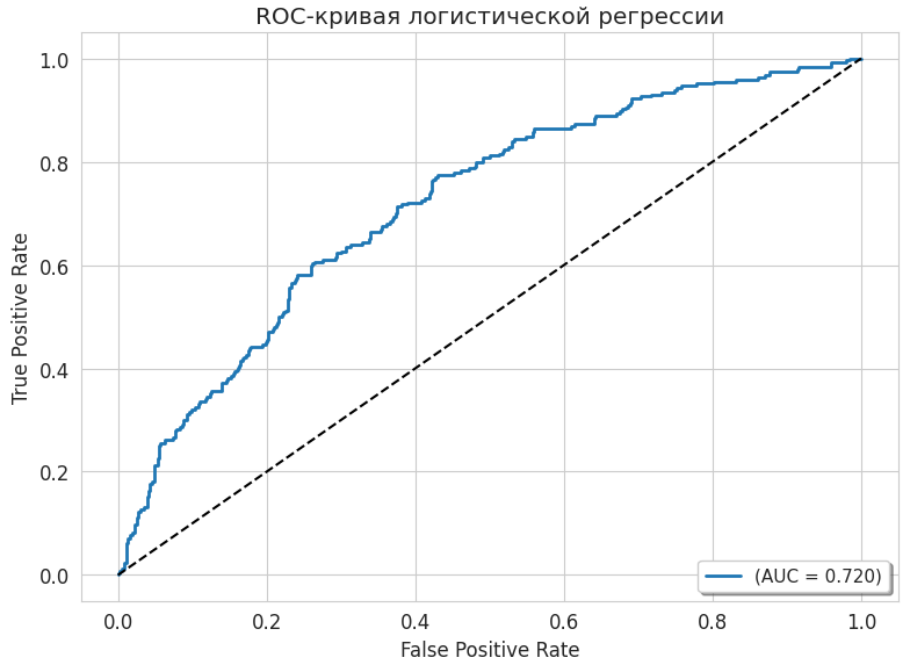

Рисунок 2.7 — ROC-кривая

Построил PR-AUC-кривую представлено на Рисунке 2.8.

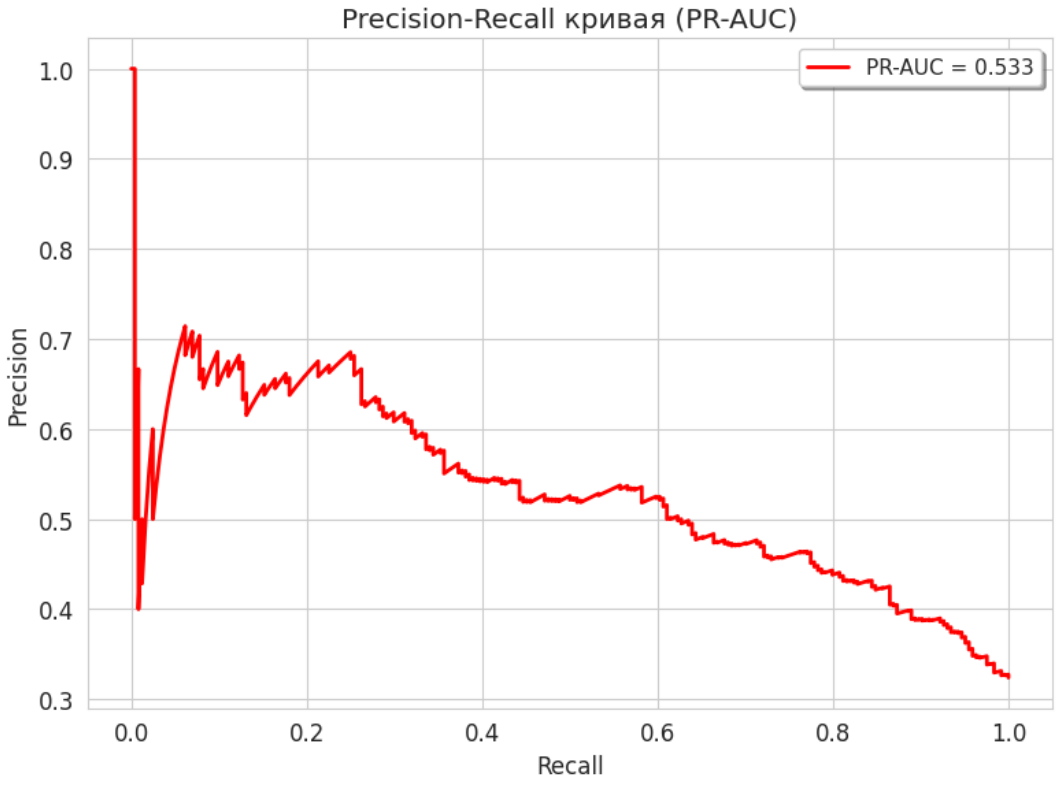

Рисунок 2.8 — PR-AUC-кривая

Разработана система скоринга, разделяющая клиентов на 5 групп риска на основе предсказанных вероятностей оттока.

```
1. Распределение клиентов по группам риска:
Группа 1: 151 клиентов  20.1%
Группа 2: 150 клиентов  19.9%
Группа 3: 151 клиентов  20.1%
Группа 4: 150 клиентов  19.9%
Группа 5: 151 клиентов  20.1%
2. Доля оттока по группам риска:
Группа 1: 10.6% оттока (16 из 151 клиентов)
Группа 2: 20.7% оттока (31 из 150 клиентов)
Группа 3: 30.5% оттока (46 из 151 клиентов)
Группа 4: 42.7% оттока (64 из 150 клиентов)
Группа 5: 57.6% оттока (86 из 151 клиентов)
3. Пороги вероятностей для групп риска:
Группа 1: 0.000 - 0.303
Группа 2: 0.303 - 0.396
Группа 3: 0.396 - 0.512
Группа 4: 0.512 - 0.631
Группа 5: 0.631 - 1.000
```

Рисунок 2.9 — Распределение клиентов

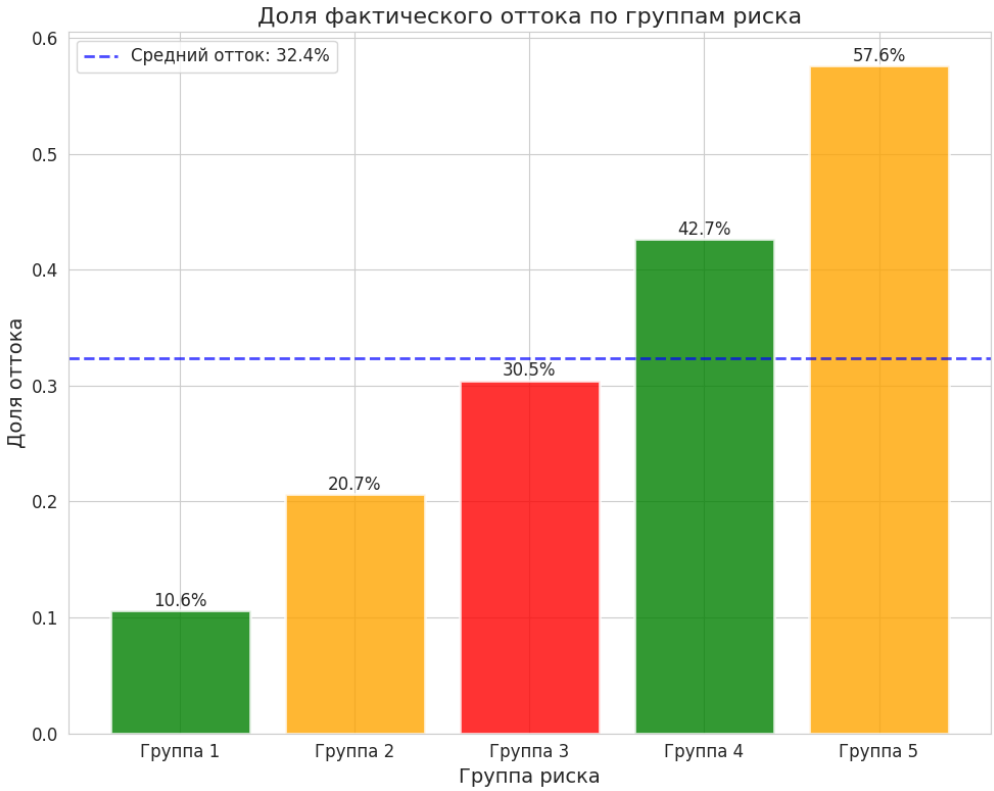

Рисунок 2.10 — Распределение по группам

График калибровки визуально оценивает, насколько хорошо предсказанные вероятности вашей модели соответствуют истинным вероятностям. Идеально калиброванная модель должна следовать диагональной линии, где предсказанная вероятность (ось X) совпадает с фактической долей

положительных исходов (ось Y). Отклонения от этой линии указывают на недооценку или переоценку вероятностей моделью показано на Рисунке 2.11.

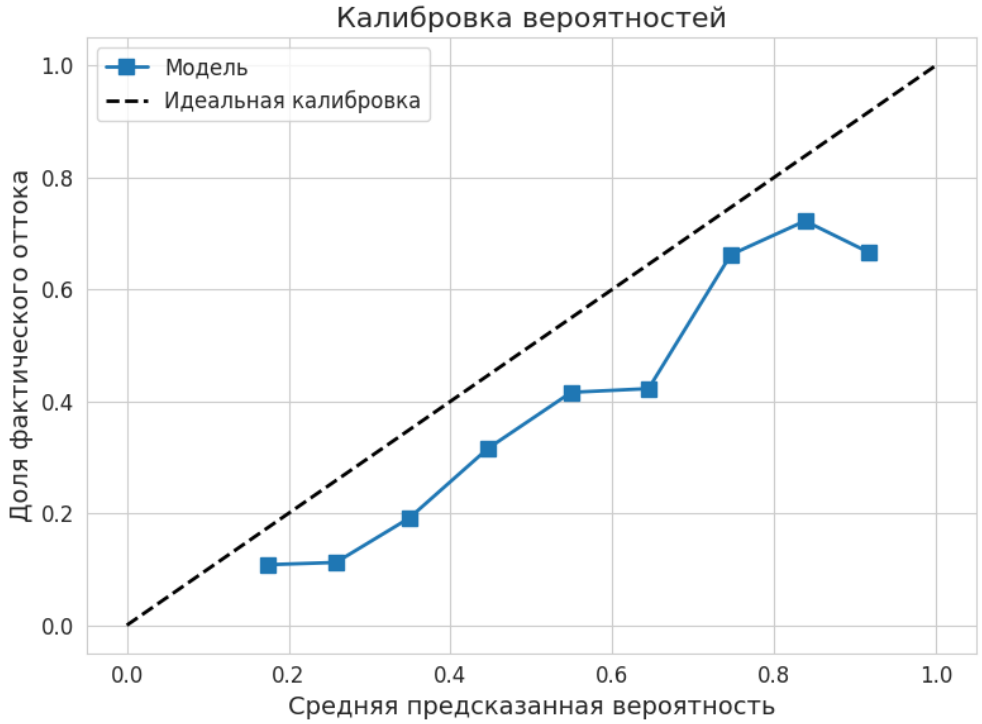

Рисунок 2.11 — Калибровка вероятностей

График важности признаков — наибольшее влияние: Age, Balance показано на Рисунке 2.12.

Рисунок 2.12 — Важность признаков

Разработана функция для детального анализа конкретных клиентов с учетом их характеристик и вклада каждого признака в итоговый прогноз показано на Рисунке 2.13.

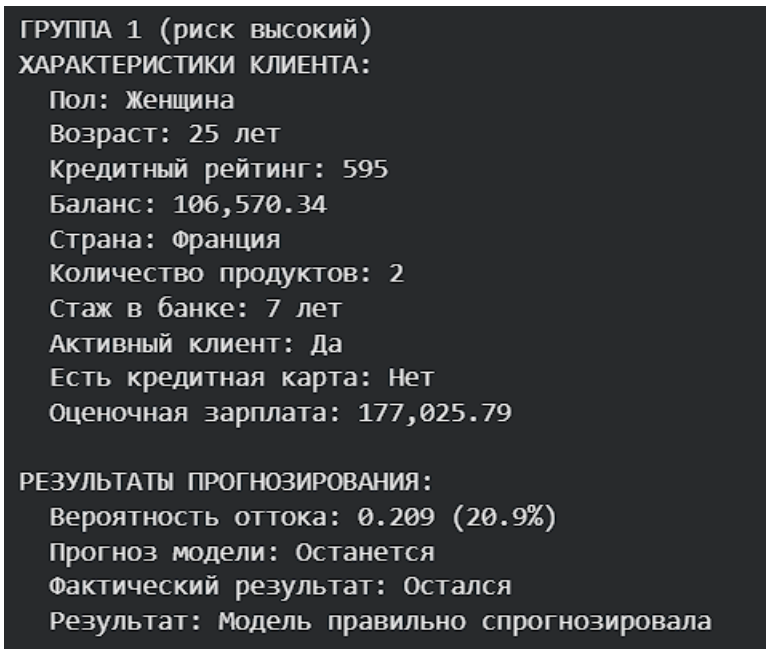

Рисунок 2.13 — Анализ клиента

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы достигнута поставленная цель — разработан и протестирован сценарий анализа и обработки данных для прогнозирования оттока клиентов банка с использованием логистической регрессии.

Основные результаты показывают:

- Анализ данных показал, что наибольший отток наблюдается в Германии, а также среди клиентов старшего возраста с высоким балансом.
- Построена и оценена модель логистической регрессии с L2-регуляризацией, показала хорошие результаты (F1 = 0.56, ROC-AUC = 0.72, PR-AUC: 0.53).
- Разработана система скоринга, позволяющая сегментировать клиентов по уровню риска.
- Важные признаки: возраст, баланс, активность клиента, число продуктов.
- Практическая значимость: модель позволяет идентифицировать клиентов с высоким риском оттока и принимать превентивные меры.

Рекомендации для банка:

- Сфокусироваться на клиентах из групп высокого риска (Группы 4–5).
- Разработать персонализированные предложения для удержания клиентов.
- Мониторить динамику оттока по странам и демографическим группам.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Теоретическая часть

1. Хргиан А.Х. "Физика атмосферы" - М.: Изд-во МГУ, 2020. - 450 с.
2. Логистическая регрессия в машинном обучении / Машинное обучение [Электронный ресурс]. URL: https://machinelearning.ru/wiki/index.php/Логистическая_регрессия
3. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. — Springer, 2021. — 426 p.
4. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. — Springer, 2021. — 426 p.

Практическая часть

1. Документация библиотеки scikit-learn [Электронный ресурс]. URL: https://scikit-learn.org/stable/
2. Pandas User Guide: Time Series / Date functionality [Электронный ресурс]. URL: https://pandas.pydata.org/docs/user_guide/timeseries.html
3. Оценка качества моделей классификации / Habr [Электронный ресурс]. URL: https://habr.com/ru/companies/ods/articles/328372/
4. Kaggle: Churn for Bank Customers Dataset [Электронный ресурс]. — URL: https://www.kaggle.com/datasets/mathchi/churn-for-bank-customers/data

ПРИЛОЖЕНИЯ

Приложение А — код на языке программирование Python, в котором происходит обработка данных.

Приложение Б — код на языке программирование Python, в котором происходит анализ данных

Приложение В — код на языке программирование Python, в котором происходит визуализация данных.

Приложение Г — код на языке программирование Python, в котором происходит система скоринга

Приложение А

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    roc_auc_score, log_loss, confusion_matrix, roc_curve,
    precision_recall_curve, average_precision_score,
    classification_report
)
from sklearn.calibration import calibration_curve
import warnings
warnings.filterwarnings('ignore')

# Загрузка данных
churn_data = pd.read_csv('churn.csv')
churn_data.head()

churn_data.info()

churn_data.isnull().sum()

churn_data.duplicated().sum()

churn_data.describe()

grouped_by_exit = churn_data['Exited'].value_counts()

plt.figure(figsize=(7, 7))
plt.pie(grouped_by_exit, labels=['Лояльные', 'Ушедшие'],
colors=['lime', 'red'], autopct='%1.1f%%',
        startangle=90, explode=[0.05, 0], textprops={'fontsize':
14})
plt.title('Распределение клиентов по оттоку', fontsize=14)
plt.show()

# Анализ оттока по странам
plt.figure(figsize=(10, 8))
country_churn =
churn_data.groupby('Geography')['Exited'].mean().sort_values(ascen
ding=False)
sns.barplot(x=country_churn.index, y=country_churn.values,
palette='viridis')
plt.title('Доля оттока клиентов по странам', fontsize=14)
plt.ylabel('Доля оттока')
plt.xlabel('Страна')
```

```
for i, v in enumerate(country_churn.values):
    plt.text(i, v + 0.005, f'{v:.3f}', ha='center', va='bottom')
plt.tight_layout()
plt.show()

# Предобработка данных для Германии
german_data = churn_data[churn_data['Geography'] ==
'Germany'].copy()

data = german_data.drop(['RowNumber', 'CustomerId', 'Surname',
'Geography'], axis=1)

le = LabelEncoder()
data['Gender'] = le.fit_transform(data['Gender'])  # Male=1,
Female=0

data.head()
```

Приложение Б

```
X = data.drop('Exited', axis=1)
y = data['Exited']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = LogisticRegression(
    C=0.01,
    penalty='l2',
    solver='saga',
    class_weight='balanced',
    max_iter=1000,
    random_state=42
)

model.fit(X_train_scaled, y_train)

y_pred = model.predict(X_test_scaled)
y_proba = model.predict_proba(X_test_scaled)[:, 1]

acc = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_proba)
pr_auc = average_precision_score(y_test, y_proba)
log_loss = log_loss(y_test, y_proba)

print(f"Accuracy: {acc:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1: {f1:.4f}")
print(f"ROC-AUC: {roc_auc:.4f}")
print(f"PR-AUC: {pr_auc:.4f}")
print(f"Логарифмическая функция потерь: {log_loss:.4f}")
```

Приложение В

```
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Лояльные', 'Ушедшие'],
            yticklabels=['Лояльные', 'Ушедшие'])
plt.title("Матрица ошибок логистической регрессии", fontsize=14)
plt.xlabel('Предсказание', fontsize=12)
plt.ylabel('Факт', fontsize=12)
plt.tight_layout()
plt.show()

fpr, tpr, _ = roc_curve(y_test, y_proba)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'(AUC = {roc_auc:.3f})', linewidth=2)
plt.plot([0,1], [0,1], 'k--')
plt.title("ROC-кривая логистической регрессии")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend(loc='lower right', fontsize=11, frameon=True,
shadow=True)
plt.grid(True)
plt.tight_layout()
plt.show()

precision_vals, recall_vals, _ = precision_recall_curve(y_test,
y_proba)
pr_auc = average_precision_score(y_test, y_proba)

plt.figure(figsize=(8, 6))
plt.plot(recall_vals, precision_vals, label=f'PR-AUC =
{pr_auc:.3f}', linewidth=2, color='red')
plt.title("Precision-Recall кривая (PR-AUC)")
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.legend(loc='upper right', fontsize=11, frameon=True,
shadow=True)
plt.grid(True)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 6))
prob_true, prob_pred = calibration_curve(y_test, y_proba,
n_bins=10)
plt.plot(prob_pred, prob_true, 's-', label='Модель',
         linewidth=2, markersize=8)
plt.plot([0, 1], [0, 1], 'k--', label='Идеальная калибровка',
         linewidth=2)
plt.title('Калибровка вероятностей', fontsize=16)
```

```
plt.xlabel('Средняя предсказанная вероятность', fontsize=14)
plt.ylabel('Доля фактического оттока', fontsize=14)
plt.legend(loc='best', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()

feature_names = X.columns.tolist()
coefficients = pd.DataFrame({
    'Признак': feature_names,
    'Коэффициент': model.coef_[0],
    'Абсолютное_значение': np.abs(model.coef_[0])
}).sort_values('Абсолютное_значение', ascending=False)

plt.figure(figsize=(10, 6))
colors = ['red' if coef > 0 else 'green' for coef in
coefficients['Коэффициент']]
bars = plt.barh(coefficients['Признак'],
coefficients['Абсолютное_значение'],
                color=colors, edgecolor='black', alpha=0.7)
plt.title('Важность признаков в модели', fontsize=16)
plt.xlabel('Абсолютное значение коэффициента', fontsize=14)
plt.grid(True, axis='x', alpha=0.3)

for bar, coef in zip(bars, coefficients['Коэффициент']):
    width = bar.get_width()
    plt.text(width + 0.005, bar.get_y() + bar.get_height()/2,
             f'{coef}', va='center', fontsize=10)

plt.tight_layout()
plt.show()
```

Приложение Г

```
# Система скоринга клиентов

# Создает систему скоринга с заданным количеством групп
def create_scoring_system(probabilities, n_groups=5):
    percentiles = np.linspace(0, 100, n_groups + 1)[1:-1]
    thresholds = np.percentile(probabilities, percentiles)
    return np.concatenate([[0], thresholds, [1]])

# Присваивает группу риска на основе вероятности
def assign_scoring_group(probability, thresholds, n_groups=5):
    for i in range(len(thresholds) - 1):
        if thresholds[i] <= probability < thresholds[i+1]:
            return i + 1, f"Группа {i+1}"
    return n_groups, f"Группа {n_groups}"

# Создание системы скоринга с 5 группами
scoring_thresholds = create_scoring_system(y_proba, n_groups=5)

# Применение скоринга
risk_groups = []
scores = []
score_points = []

for prob in y_proba:
    score, group = assign_scoring_group(prob, scoring_thresholds,
n_groups=5)
    risk_groups.append(group)
    scores.append(score)
    score_points.append(int(prob * 100))  # Преобразование в шкалу
0-100

# Создание DataFrame с результатами скоринга
scoring_results = pd.DataFrame({
    'Фактический_отток': y_test.values,
    'Вероятность_оттока': y_proba,
    'Группа_риска': risk_groups,
    'Скор_балл': scores,
    'Скор_процент': score_points,
    'Предсказание': y_pred
})

# Анализ эффективности скоринговой системы

# Распределение по группам риска
risk_distribution =
scoring_results['Группа_риска'].value_counts().sort_index()
print("1. Распределение клиентов по группам риска:")
for group in sorted(risk_distribution.index):
    count = risk_distribution[group]
```

```
    percentage = count / len(scoring_results) * 100
    print(f"{group}: {count:3d} клиентов {percentage:5.1f}%")

# Доля оттока по группам риска
print("2. Доля оттока по группам риска:")
churn_by_risk =
scoring_results.groupby('Группа_риска')['Фактический_отток'].mean(
).sort_index()

for group in sorted(churn_by_risk.index):
    churn_rate = churn_by_risk[group]
    count = risk_distribution[group]
    print(f"{group}:{churn_rate*100:5.1f}% оттока ({int(count *
churn_rate)} из {count} клиентов)")

print("3. Пороги вероятностей для групп риска:")
for i in range(len(scoring_thresholds) - 1):
    print(f"Группа {i+1}: {scoring_thresholds[i]:.3f} -
{scoring_thresholds[i+1]:.3f}")

# Доля оттока по группам риска
plt.figure(figsize=(10, 8))
groups = sorted(churn_by_risk.index)
churn_rates = [churn_by_risk[group] for group in groups]
colors = ['green', 'orange', 'red']

bars = plt.bar(groups, churn_rates, color=colors, alpha=0.8,
linewidth=2)
plt.title('Доля фактического оттока по группам риска',
fontsize=16)
plt.xlabel('Группа риска', fontsize=14)
plt.ylabel('Доля оттока', fontsize=14)
plt.xticks(rotation=0)

for bar, rate in zip(bars, churn_rates):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 0.0001,
             f'{rate*100:.1f}%', ha='center', va='bottom')

# Линия среднего оттока
mean_churn = y_test.mean()
plt.axhline(y=mean_churn, color='blue', linestyle='--',
linewidth=2, alpha=0.7,
            label=f'Средний отток: {mean_churn*100:.1f}%')
plt.legend(loc='upper left', fontsize=12)

plt.grid(True)
plt.tight_layout()
plt.show()

# Функция для анализа конкретного клиента
def analyze_client(client_idx, group_name):
```

```
    # Получение данных клиента
    client_data = X_test.iloc[client_idx]
    client_scaled = scaler.transform(client_data.values.reshape(1,
-1))

    # Прогнозирование
    client_prob = model.predict_proba(client_scaled)[0, 1]
    client_pred = model.predict(client_scaled)[0]
    client_actual = y_test.iloc[client_idx]

    # Анализ вклада признаков
    feature_contributions = model.coef_[0] * client_scaled[0]

    return {
        'idx': client_idx,
        'data': client_data,
        'probability': client_prob,
        'prediction': client_pred,
        'actual': client_actual,
        'contributions': feature_contributions
    }

# Анализ клиентов из разных групп риска

# Словарь для хранения проанализированных клиентов
analyzed_clients = {}

# Анализ клиентов из каждой группы риска
group_name = f"Группа 2"
group_num = 2
# Находим всех клиентов в этой группе
clients_in_group = scoring_results[scoring_results['Группа_риска']
== group_name]

# Берем случайного клиента из группы
random_idx = np.random.choice(clients_in_group.index)

# Анализируем клиента
client_info = analyze_client(random_idx, group_name)
analyzed_clients[group_name] = client_info

print(f"{group_name.upper()} (риск {'низкий' if group_num <= 2
else 'средний' if group_num == 3 else 'высокий'})")

# Вывод характеристик клиента
print("ХАРАКТЕРИСТИКИ КЛИЕНТА:")

# Расшифровка категориальных признаков
gender_decoded = "Мужчина" if client_info['data']['Gender'] == 1
else "Женщина"
geography_decoded = "Испания" if
client_info['data'].get('Geography_Spain', 0) == 1 else "Германия"
```

```
if client_info['data'].get('Geography_Germany', 0) == 1 else
"Франция"

print(f"  Пол: {gender_decoded}")
print(f"  Возраст: {int(client_info['data']['Age'])} лет")
print(f"  Кредитный рейтинг:
{int(client_info['data']['CreditScore'])}")
print(f"  Баланс: {client_info['data']['Balance']:,.2f}")
print(f"  Страна: {geography_decoded}")
print(f"  Количество продуктов:
{int(client_info['data']['NumOfProducts'])}")
print(f"  Стаж в банке: {int(client_info['data']['Tenure'])} лет")
print(f"  Активный клиент: {'Да' if
client_info['data']['IsActiveMember'] == 1 else 'Нет'}")
print(f"  Есть кредитная карта: {'Да' if
client_info['data']['HasCrCard'] == 1 else 'Нет'}")
print(f"  Оценочная зарплата:
{client_info['data']['EstimatedSalary']:,.2f}")

# Вывод результатов прогнозирования
print()
print(f"РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ:")
print(f"  Вероятность оттока: {client_info['probability']:.3f}
({client_info['probability']*100:.1f}%)")
print(f"  Прогноз модели: {'Уйдет' if client_info['prediction'] ==
1 else 'Останется'}")
print(f"  Фактический результат: {'Ушел' if client_info['actual']
== 1 else 'Остался'}")

# Проверка правильности прогноза
if client_info['prediction'] == client_info['actual']:
    print(f"  Результат: Модель правильно спрогнозировала")
else:
    print(f"  Результат: Модель ошиблась")
```