

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА
«РАЗРАБОТКА МОДЕЛИ ДЛЯ
КЛАССИФИКАЦИИ КЛИЕНТОВ БАНКА НА
ОСНОВЕ БОЛЬШИХ ДАННЫХ»
по программе профессиональной
переподготовки «Анализ данных на языке
Python»

Выполнила: Албитова Диана Сергеевна

Научный руководитель: к.т.н. Семендяев Родион Юрьевич

Санкт-Петербург, 2023

Введение

Целью данной работы является разработка наиболее эффективной модели машинного обучения для классификации клиентов банка на основе больших данных.

В данной работе поставлены следующие **задачи**:

1. Визуализировать данные с помощью библиотек Matplotlib и Seaborn;
2. Выявить наиболее важные признаки для классификации клиентов;
3. Провести отбор гиперпараметров и исследовать эффективность различных моделей классификации, таких как метод К-ближайших соседей (KNN), Логистическая регрессия (Logistic Regression), метод опорных векторов (SVC), классификатор дерева решений (Decision Tree Classifier), метод случайного леса (Random Forest Classifier);
4. Выявить наиболее эффективную модель машинного обучения для классификации клиентов банка в области возвращаемости кредитов.



Опыт расчета кредитного рейтинга

FICO SCORE
The score lenders use.


VantageScore

Наиболее распространенные скоринговые системы

Опыт расчета кредитного рейтинга

FICO SCORE
The score lenders use.


VantageScore

Наиболее распространенные скоринговые системы


ОБЪЕДИНЁННОЕ
КРЕДИТНОЕ БЮРО

 **НБКИ**

Скоринг на основе кредитной истории

Опыт расчета кредитного рейтинга

FICO SCORE
The score lenders use.


VantageScore

Наиболее распространенные скоринговые системы


ОБЪЕДИНЁННОЕ
КРЕДИТНОЕ БЮРО

 **НБКИ**

Скоринг на основе кредитной истории


СБЕР

 **Альфа-Банк**

Новые подходы к скорингу в России

 **ВТБ**

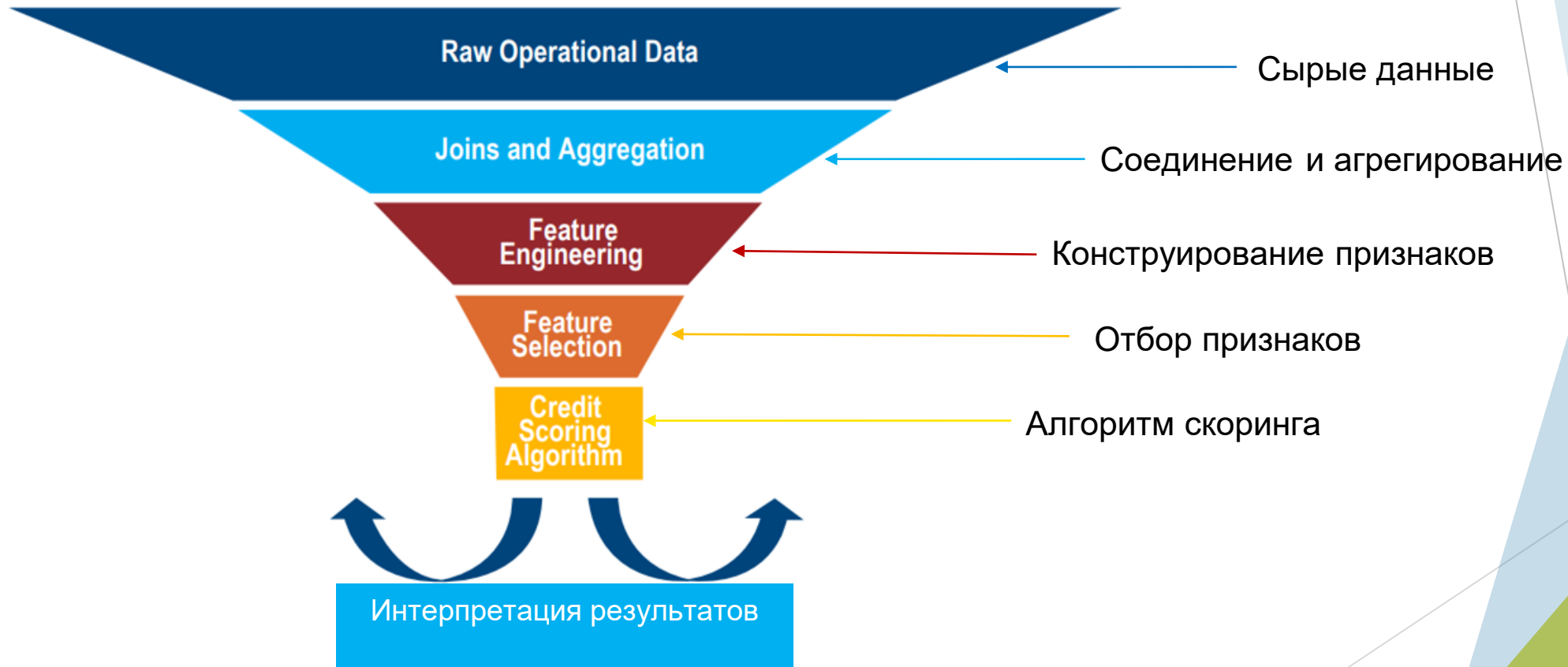


Необходимость поиска
новых решений в скоринге

5

Формирование скоринговой модели

Процесс построения скоринговой модели

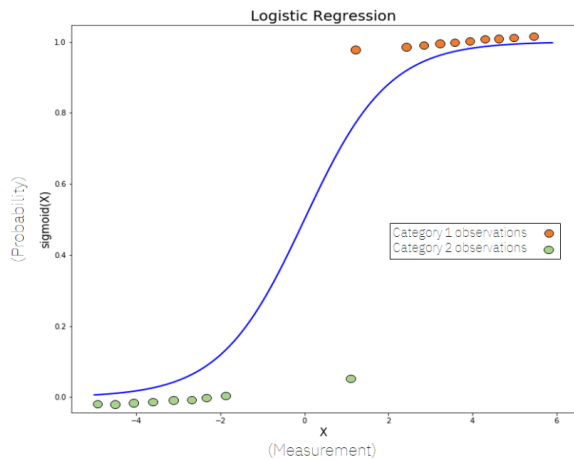
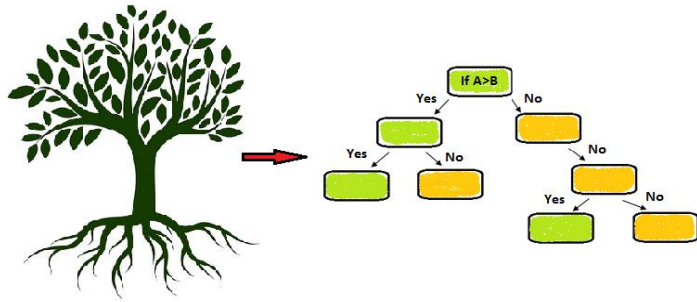


Всемирный Банк, 2019

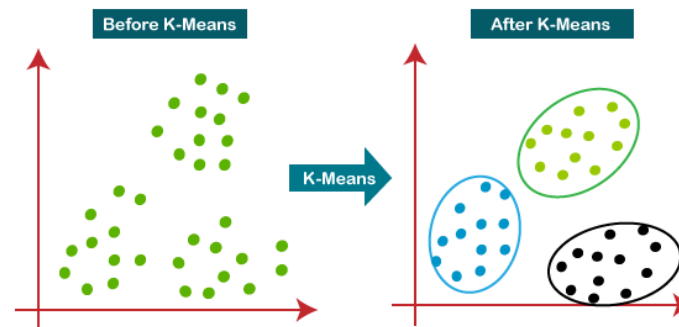
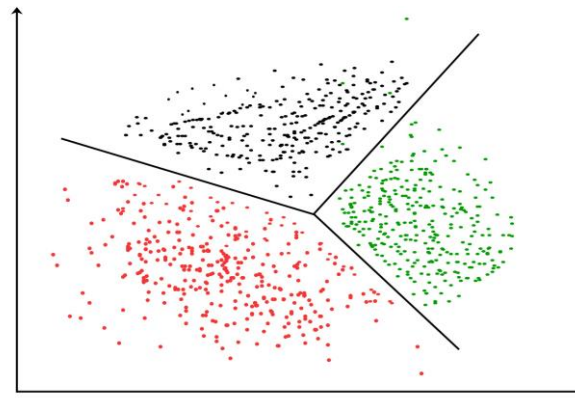
Существующие методы кредитного скоринга

- Руководство Всемирного Банка говорит о широком использовании алгоритмов машинного обучения в скоринге.
- В данной работе будут использованы алгоритмы обучения с учителем для построения модели, классифицирующей клиентов банка на благонадежных и неблагонадежных

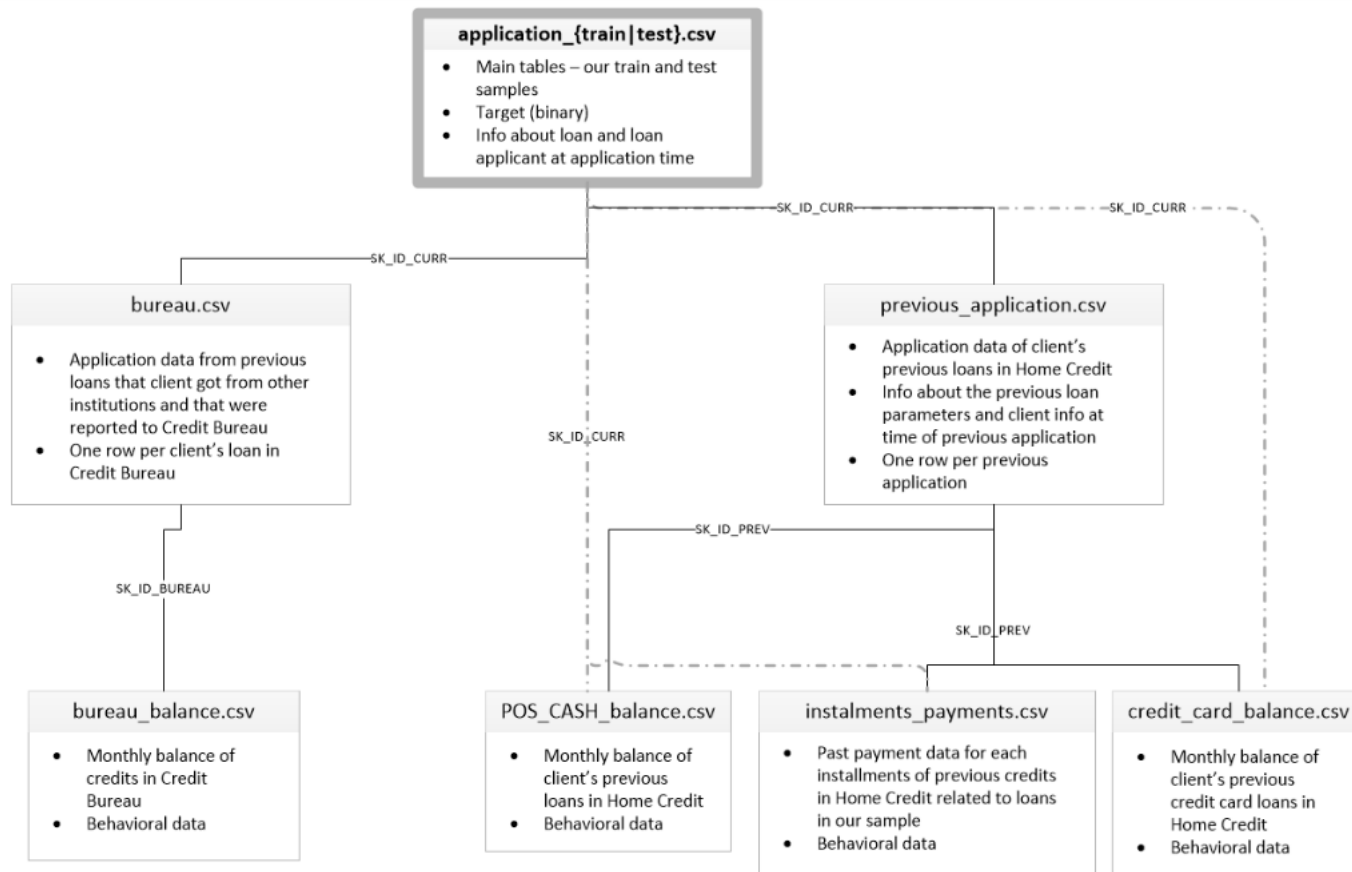
Обучение с учителем



Обучение без учителя



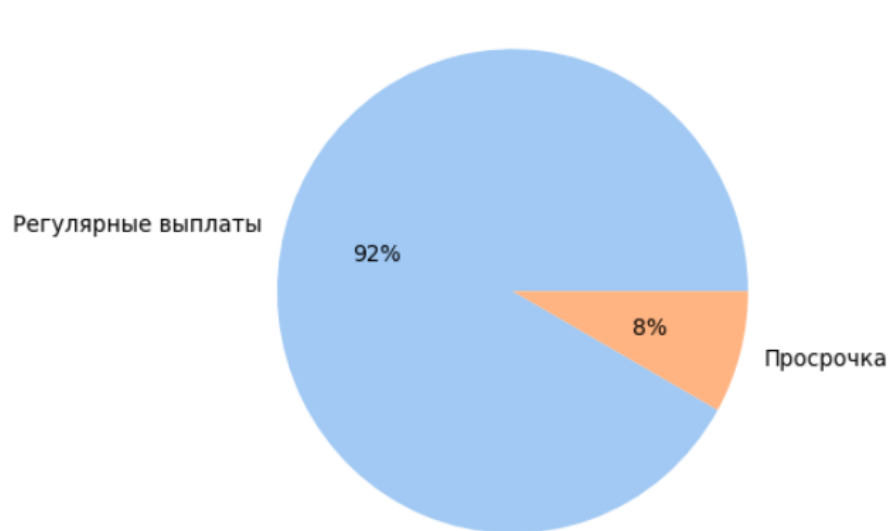
Работа с данными



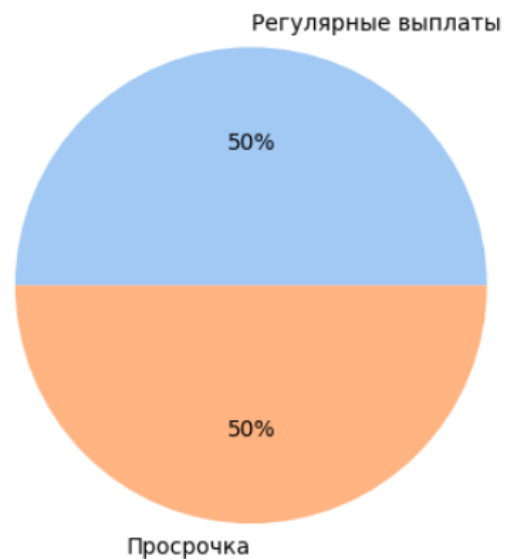
- ▶ Слияние датасетов Home Credit Bank из трех источников:
 1. Старые данные банка
 2. Данные кредитных бюро
 3. Данные о новых запросах клиентов с таргетом
- ▶ Проведение **One hot encoding**

Сбалансированность данных

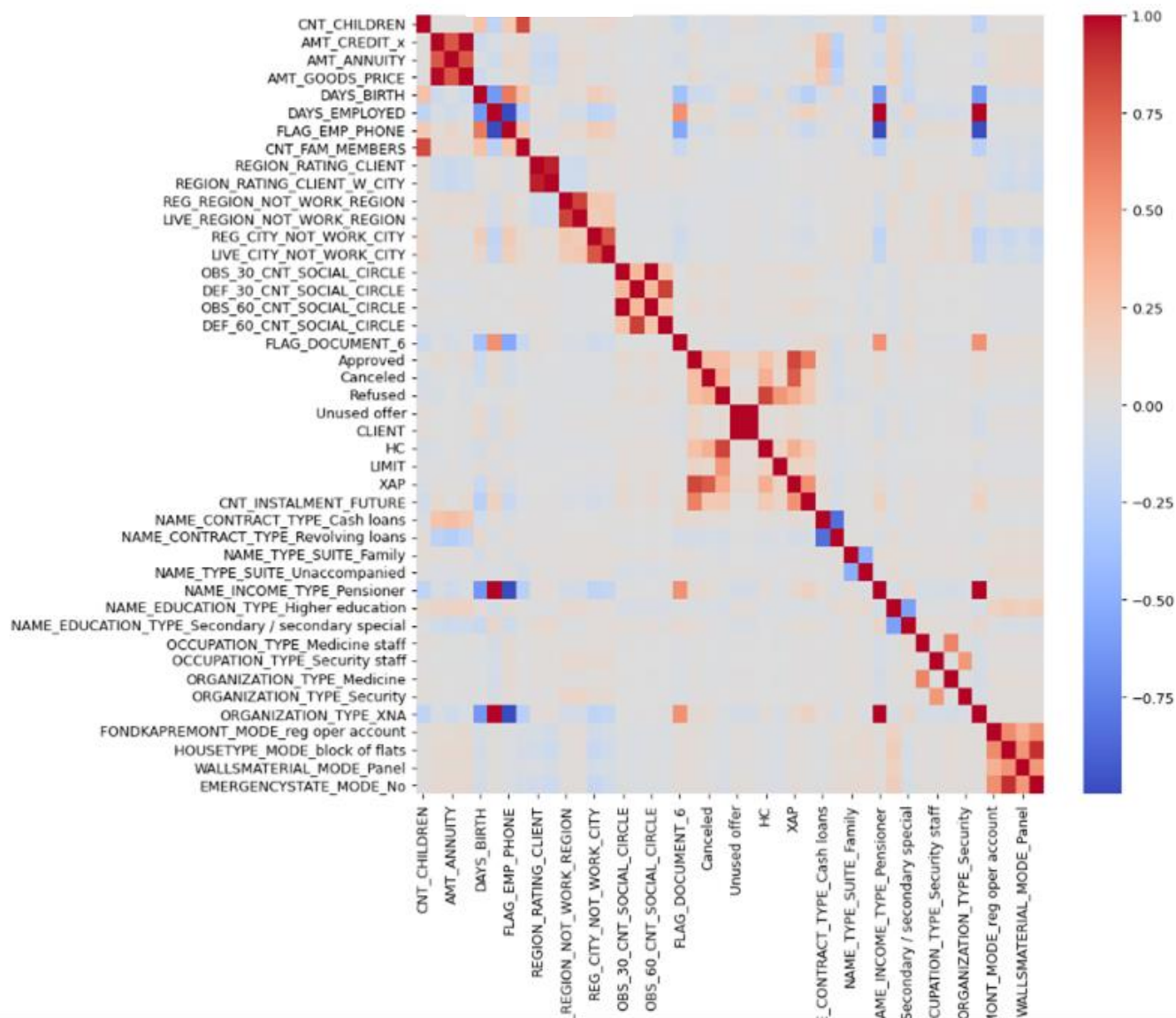
Изначальные данные по таргету несбалансированы



Сбалансированные данные (Метод увеличения числа примеров миноритарного класса - SMOTE)



Отбор признаков



Избавляемся от мультиколлинеарности: отбрасываем 14 сильно коррелирующих друг с другом признаков

Отбор признаков



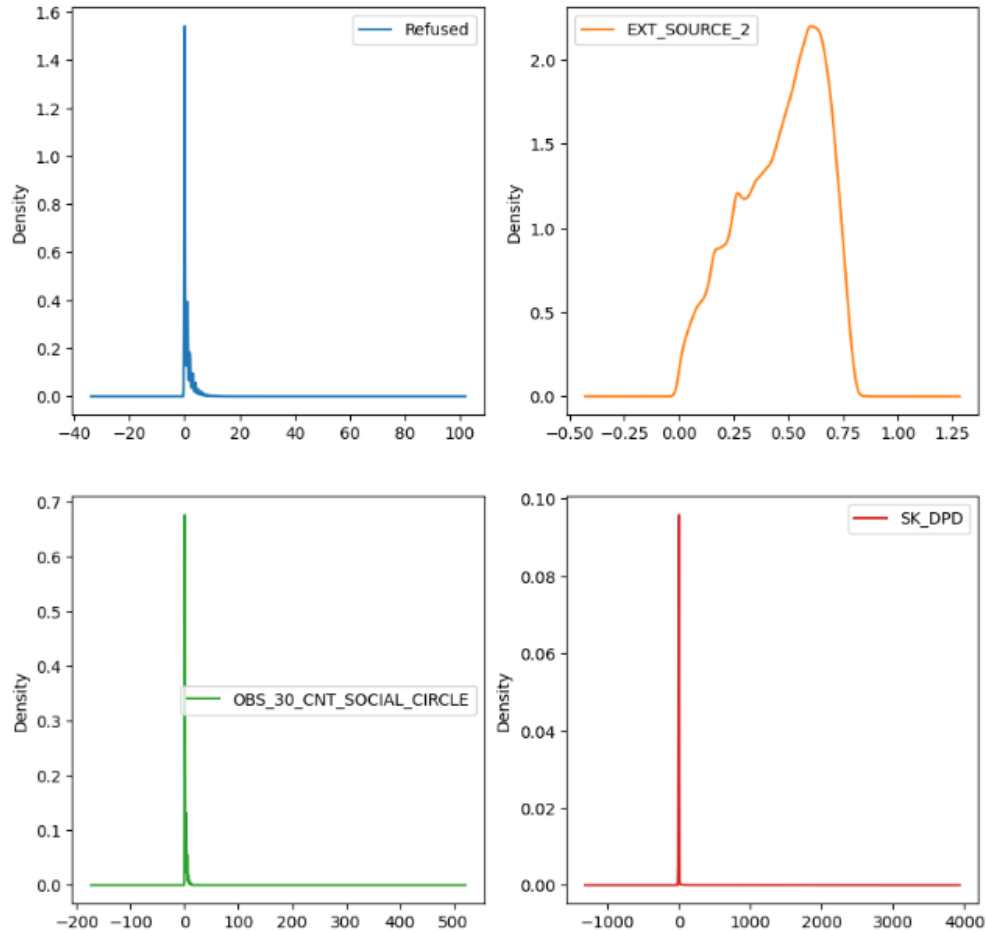
С помощью случайного леса были отобраны 20 лучших признаков.

KBest: 20 признаков, точность – 0.844.

Самым важным признаком оказался признак «Refused» («Отказ») из старых данных банка.

Итоговый размер данных составил **526466** строк и **21** признак (включая таргет).

Визуализация данных

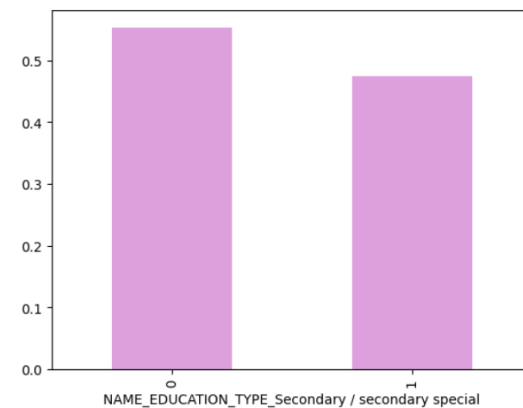
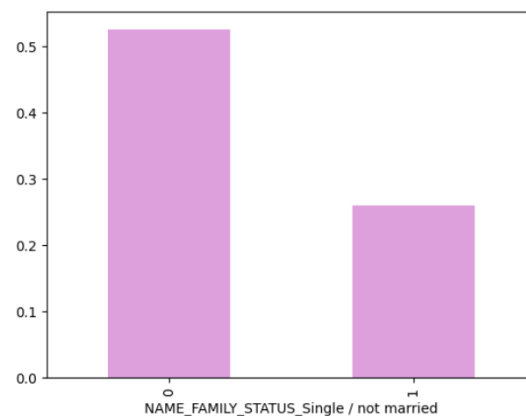
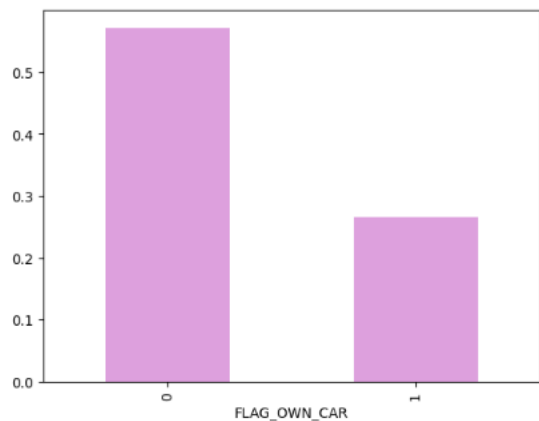
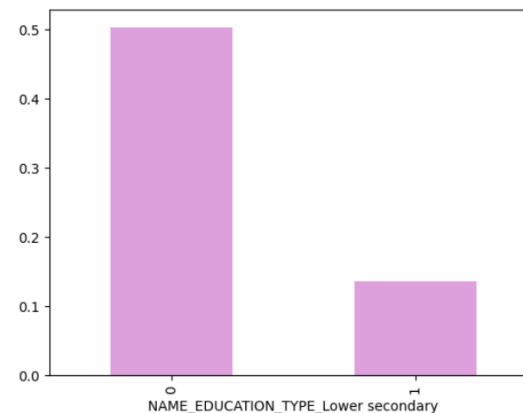
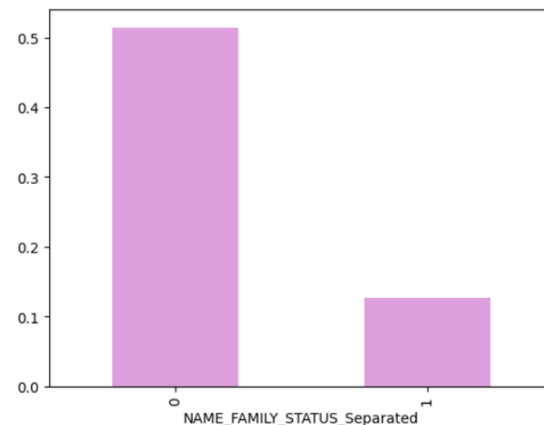
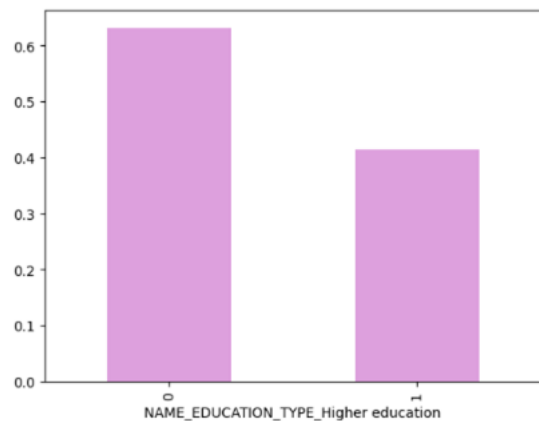


Распределение не бинарных данных далеко от нормального, но используемые алгоритмы ML не предполагают нормальности.

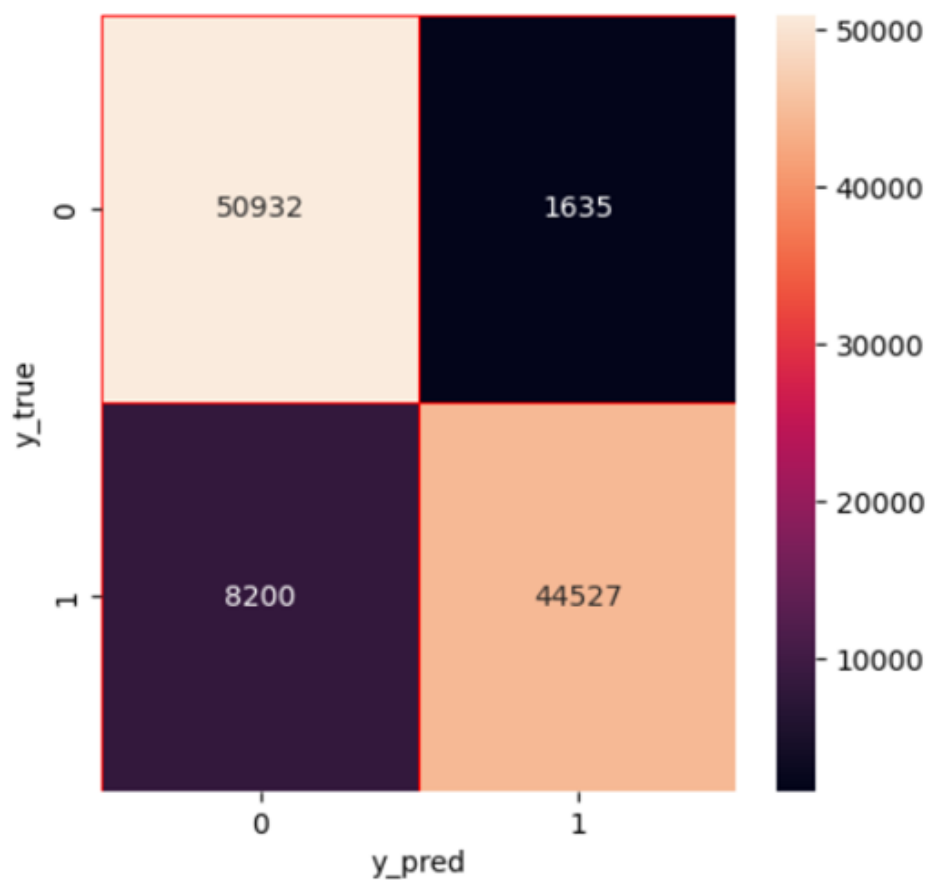
Визуализация бинамиальных данных



Визуализация бинамиальных данных в разрезе таргета



Метод k-ближайших соседей



Матрица ошибок

	precision	recall	f1-score	support
0	0.86	0.97	0.91	52567
1	0.96	0.84	0.90	52727
accuracy			0.91	105294
macro avg	0.91	0.91	0.91	105294
weighted avg	0.91	0.91	0.91	105294

Отчет о классификации

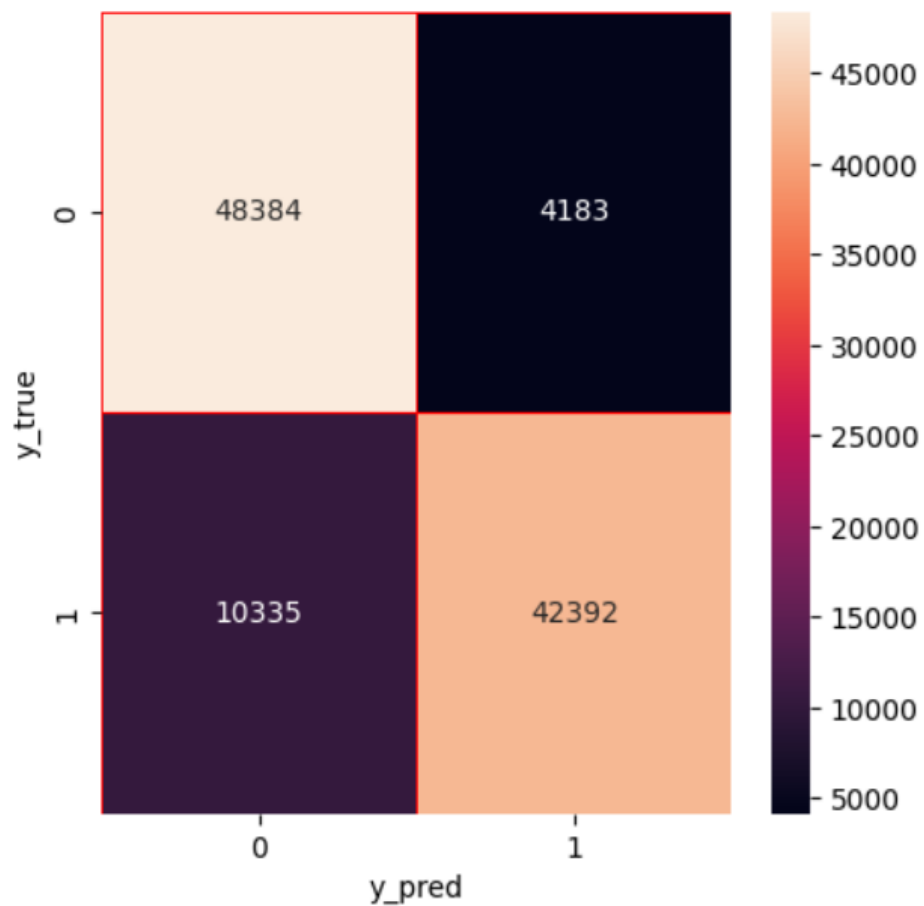
```
from sklearn.model_selection import GridSearchCV #co
knn2 = KNeighborsClassifier() #создаем словарь всех
param_grid = {"n_neighbors": np.arange(1, 10)} # исп
knn_gscv = GridSearchCV(knn2, param_grid, cv=5)
knn_gscv.fit(X_train, y_train)
```

```
knn_gscv.best_params_,knn_gscv.best_score_
```

```
({'n_neighbors': 2}, 0.9028235518825)
```

Подбор гиперпараметров

Логистическая регрессия



Матрица ошибок

	precision	recall	f1-score	support
0	0.82	0.92	0.87	52567
1	0.91	0.80	0.85	52727
accuracy			0.86	105294
macro avg	0.87	0.86	0.86	105294
weighted avg	0.87	0.86	0.86	105294

Отчет о классификации

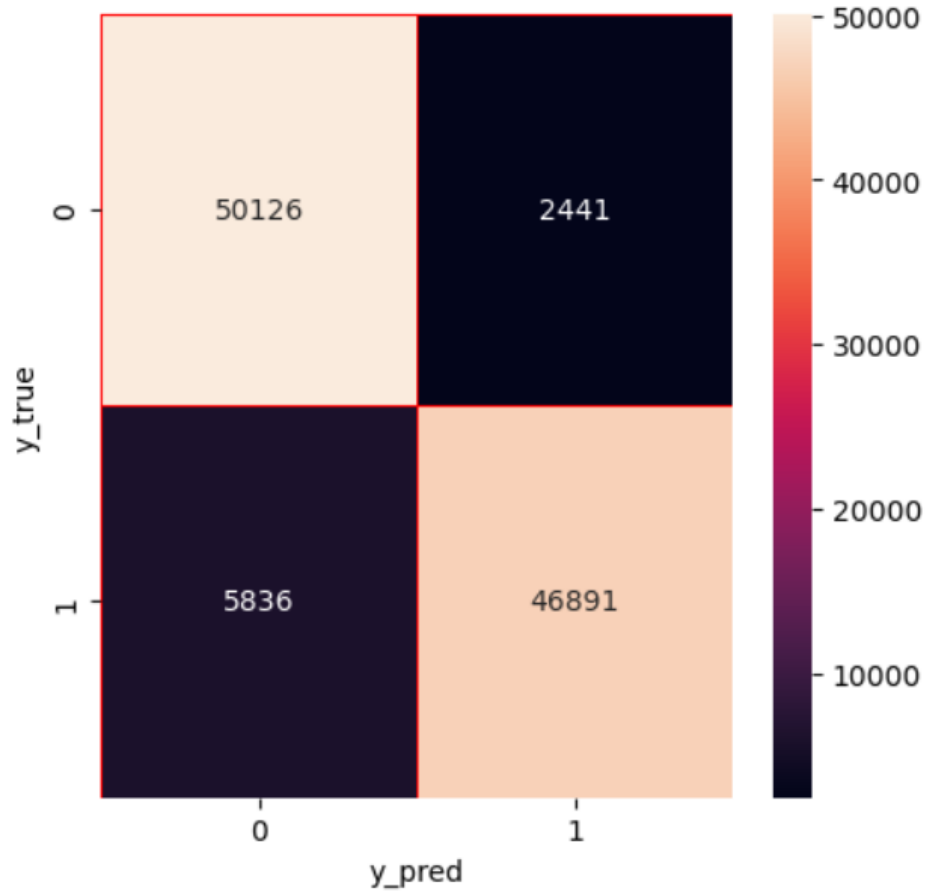
```
# Grid search
from sklearn.model_selection import GridSearchCV

#Определение гиперпараметров для логистической регрессии
hyperparameters = {"C": [1, 2, 3, 4],
                    "penalty": ["l1", "l2"],
                    "solver": ["sag", "saga"]}# L1 и L2

[ ] logreg_best = LogisticRegression(penalty='l2', C= 1, solver = 'saga')
    logreg_best.fit(X_train, y_train)
```

Подбор гиперпараметров

Случайный лес



Матрица ошибок

	precision	recall	f1-score	support
0	0.95	0.90	0.92	55962
1	0.89	0.95	0.92	49332
accuracy			0.92	105294
macro avg	0.92	0.92	0.92	105294
weighted avg	0.92	0.92	0.92	105294

Отчет о классификации

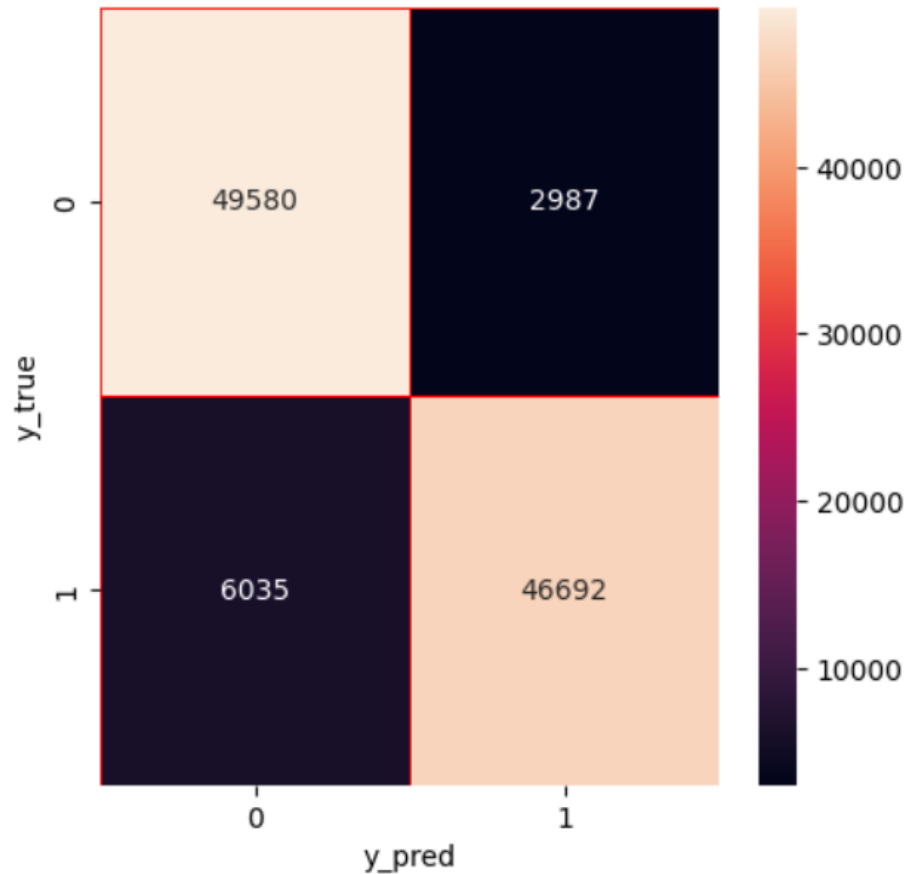
```
param_grid = {  
    'n_estimators': [80, 90, 100, 110, 120],  
    'max_features': ['sqrt', 'log2', None],  
    'max_depth': [None, 3, 6, 9],  
    'max_leaf_nodes': [None, 3, 6, 9],  
}
```

```
grid_search.best_params_
```

```
{'max_depth': None,  
 'max_features': None,  
 'max_leaf_nodes': None,  
 'n_estimators': 80}
```

Подбор гиперпараметров

Дерево решений



Матрица ошибок

	precision	recall	f1-score	support
0	0.89	0.94	0.92	52567
1	0.94	0.89	0.91	52727
accuracy			0.91	105294
macro avg	0.92	0.91	0.91	105294
weighted avg	0.92	0.91	0.91	105294

Отчет о классификации

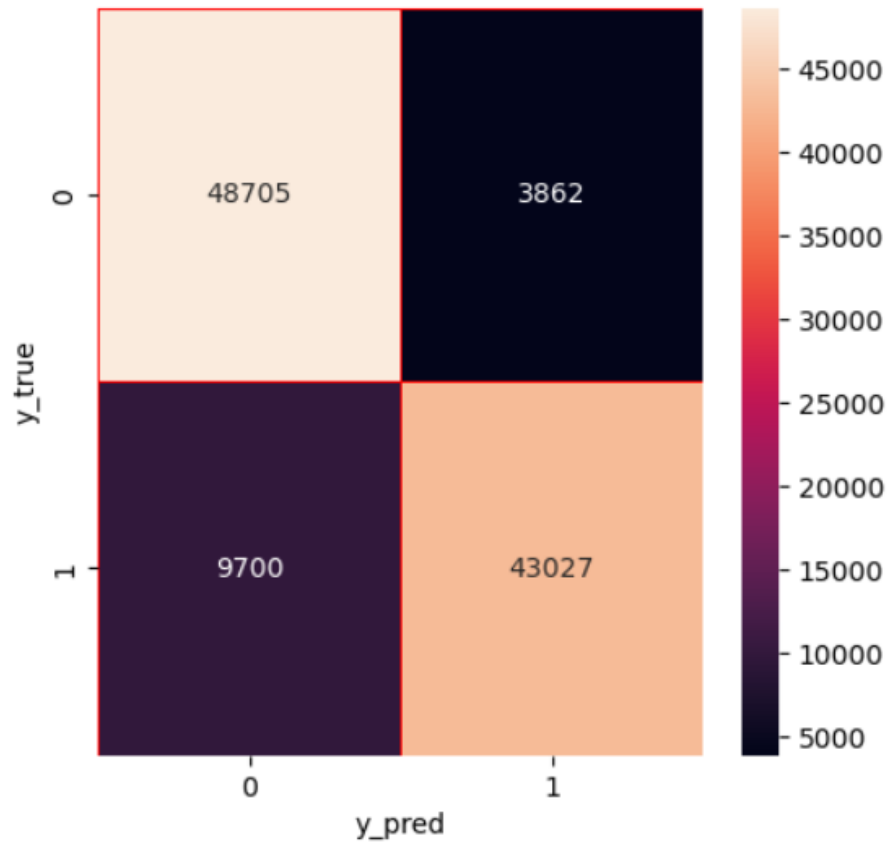
```
from sklearn.model_selection import GridSearchCV
params = {
    'max_depth': [None, 1, 2],
    'min_samples_leaf': [1, 2, 5],
    'criterion': ["gini", "entropy"]
}
```

```
dt_best = grid_search.best_estimator_
print(dt_best)
```

```
DecisionTreeClassifier(criterion='entropy', min_samples_leaf=5, random_state=42)
```

Подбор гиперпараметров

Метод опорных векторов



Матрица ошибок

	precision	recall	f1-score	support
0	0.83	0.93	0.88	52567
1	0.92	0.82	0.86	52727
accuracy			0.87	105294
macro avg	0.88	0.87	0.87	105294
weighted avg	0.88	0.87	0.87	105294

Отчет о классификации

```
# выбираем параметры для подбора гиперпараметров
param_grid = {'C':[0.1,1], 'gamma':[1,0.1], 'kernel':['linear', 'rbf']}

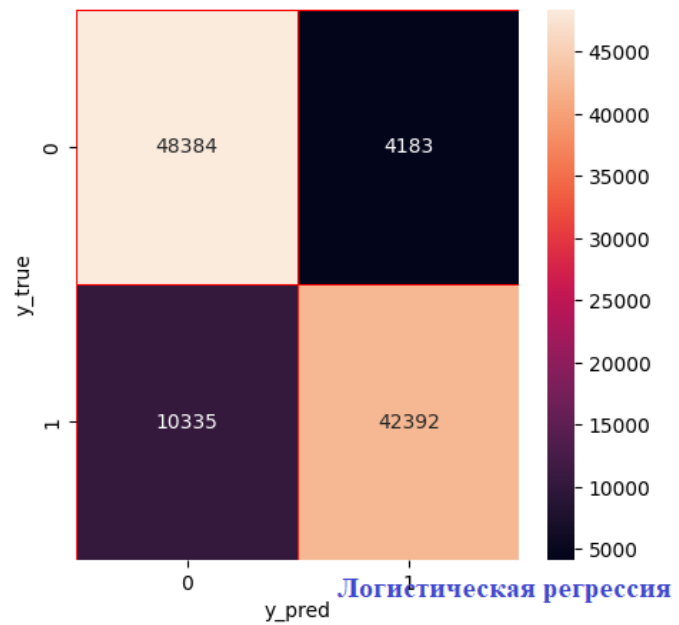
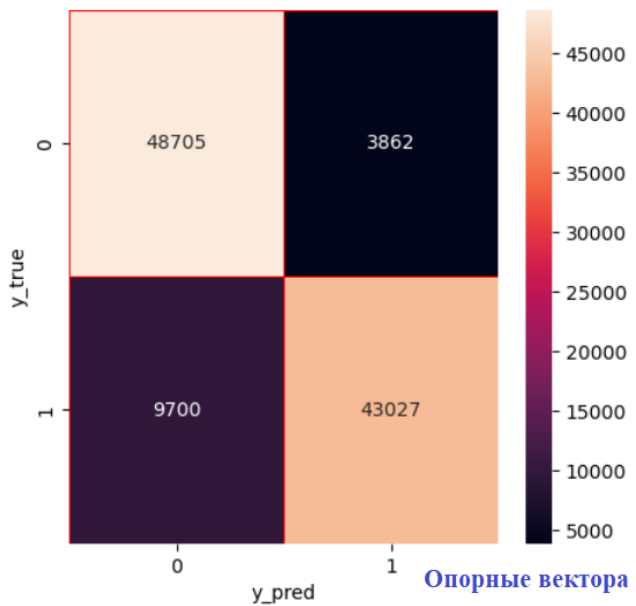
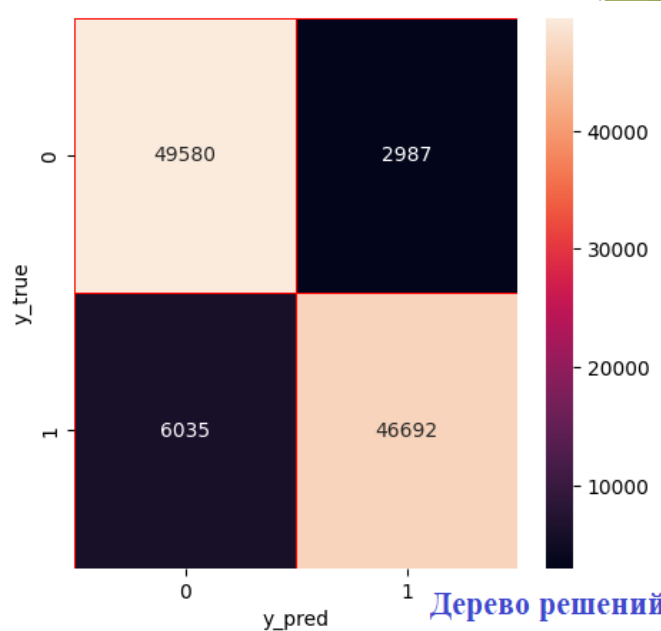
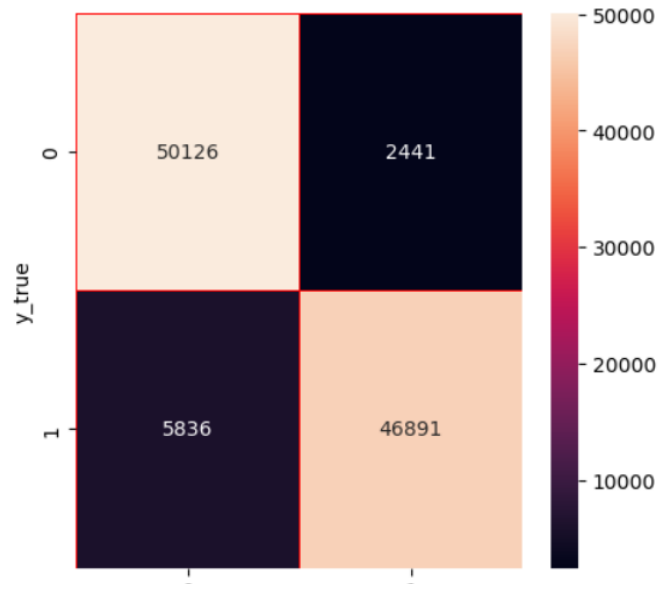
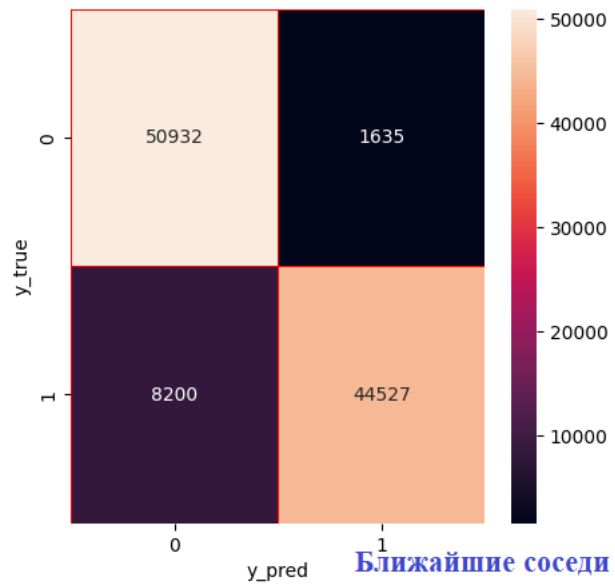
grid = GridSearchCV(SVC(),param_grid, scoring = 'f1',n_jobs=4, return_train_score=True, refit = True, verbose=2, cv=3)

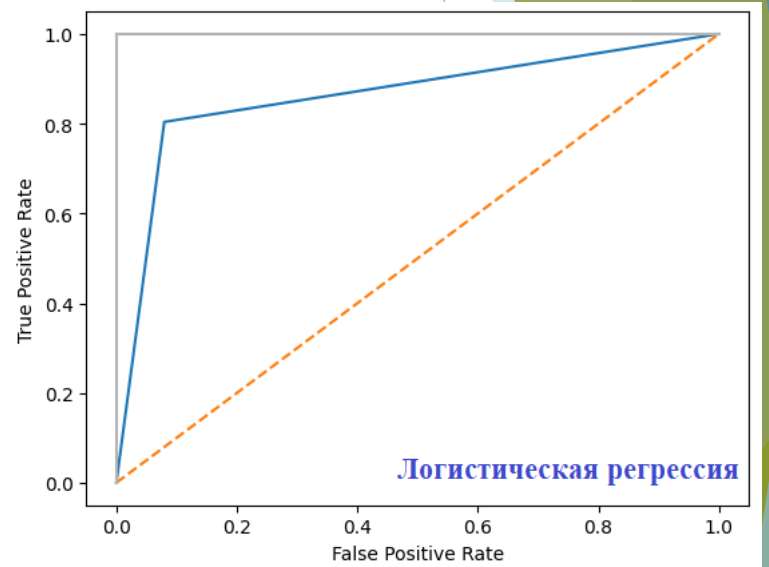
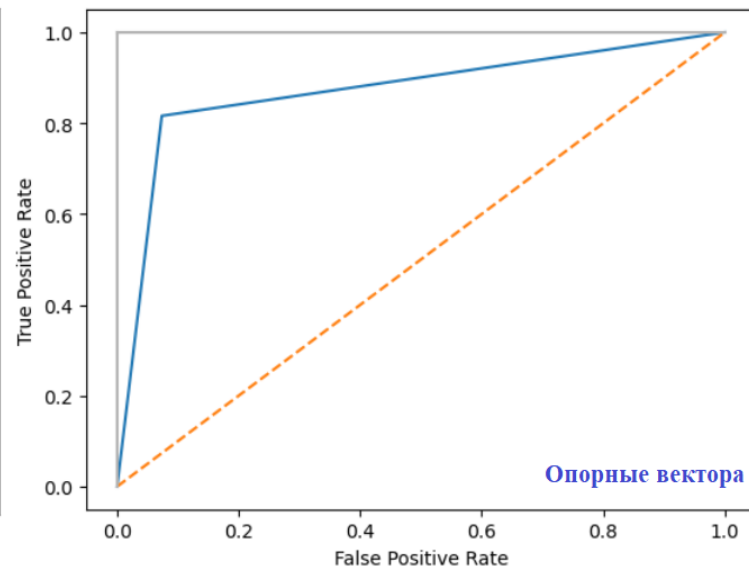
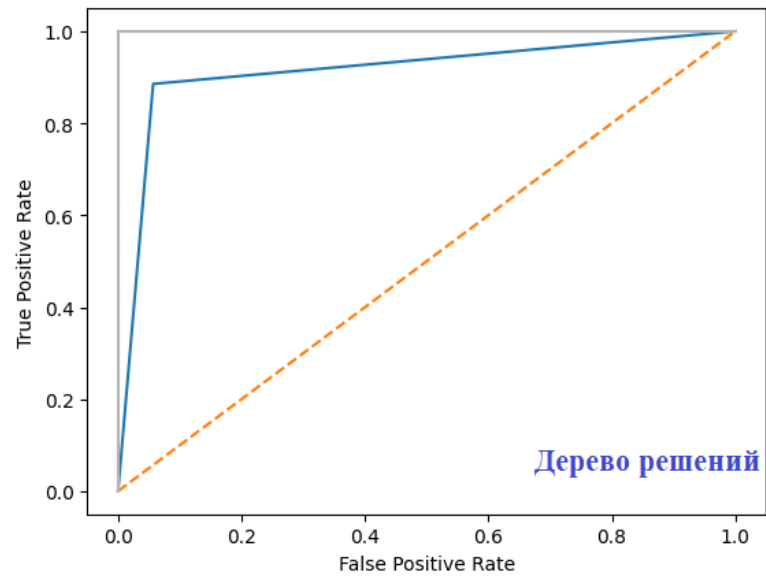
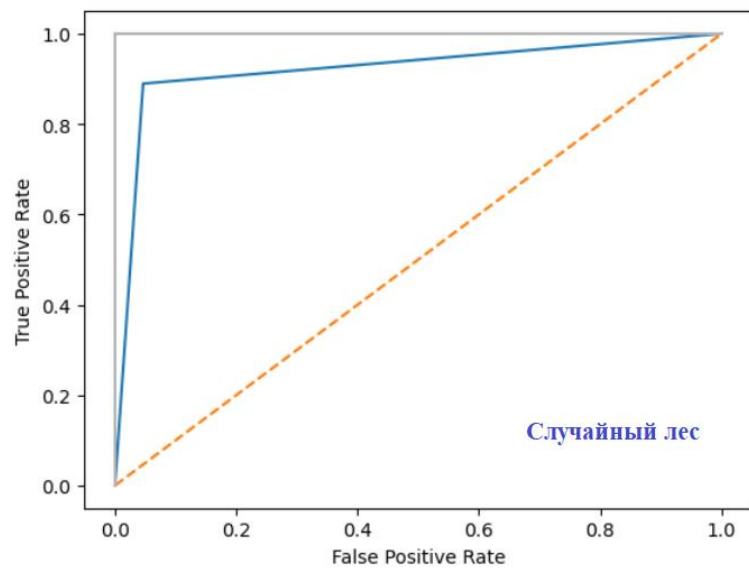
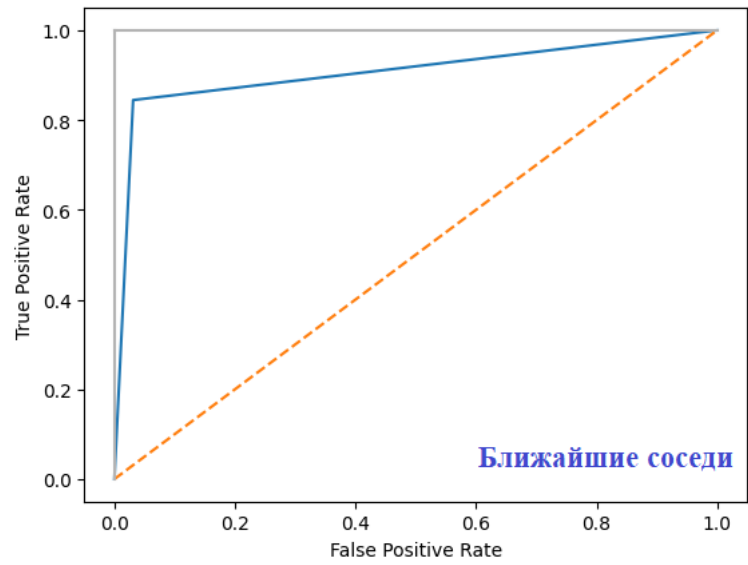
grid.fit(X_train, y_train)

Fitting 3 folds for each of 8 candidates, totalling 24 fits
GridSearchCV(cv=3, estimator=SVC(), n_jobs=4,
             param_grid={'C': [0.1, 1], 'gamma': [1, 0.1],
                        'kernel': ['linear', 'rbf']},
             return_train_score=True, scoring='f1', verbose=2)

grid.best_params_
{'C': 1, 'gamma': 1, 'kernel': 'rbf'}
```

Подбор гиперпараметров





Выводы

1. В рамках данной работы были визуализированы данные с помощью библиотек Matplotlib и Seaborn,
2. Были выявлены наиболее важные признаки для классификации клиентов
3. Был проведен отбор гиперпараметров и исследована эффективность различных моделей классификации, таких как метод К-ближайших соседей (KNN), метод опорных векторов (SVC), классификатор дерева решений (Decision Tree Classifier), метод случайного леса (Random Forest Classifier)
4. Наиболее эффективными моделями машинного обучения для классификации клиентов банка на благонадежных и неблагонадежных оказались метод **к-ближайших соседей** и метод **Случайного леса**. Наихудшие метрики показала Логистическая регрессия.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА
«РАЗРАБОТКА МОДЕЛИ ДЛЯ
КЛАССИФИКАЦИИ КЛИЕНТОВ БАНКА НА
ОСНОВЕ БОЛЬШИХ ДАННЫХ»
по программе профессиональной
переподготовки «Анализ данных на языке
Python»

Выполнила: Албитова Диана Сергеевна

Научный руководитель: к.т.н. Семендяев Родион Юрьевич

Санкт-Петербург, 2023