

Прогнозирование события подписки на краткосрочный депозит после прямой маркетинговой компании

Выполнил: Серов И.Р.

Научный руководитель: Мещеряков А.О.

Цель - Обучить модель машинного обучения, позволяющую прогнозировать событие подписки клиента на краткосрочный депозит после прямой маркетинговой компании.

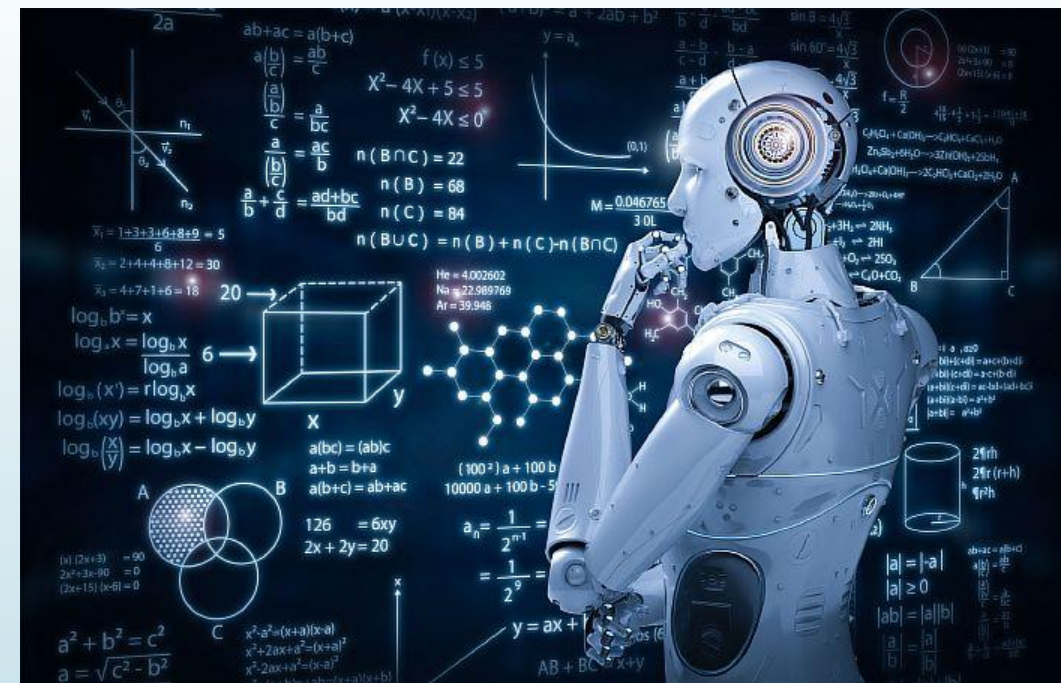
Задачи:

1. Оценить описательные статистики набора данных.

2. Произвести подготовку данных

3. Произвести отбор модели с наилучшим качеством классификации.

4. Оценить качество модель на тестовых данных.



В данной работе представлены данные связанные с кампаниями прямого маркетинга португальского банковского учреждения взятые за период с 2008 по 2010 годы(взяты с сайта <https://archive.ics.uci.edu>).

Данные включают в себя 21 колонку и 41188 записей о клиентах банка.

Факторными признаками выступают 20 колонок в том числе:

Данные о клиентах банка (возраст, семейное положение, работа, образование, наличие кредита).

Данные, связанные с последними контактами с клиентами в рамках текущей кампании (тип связи, день и месяц последнего контакта, продолжительность).

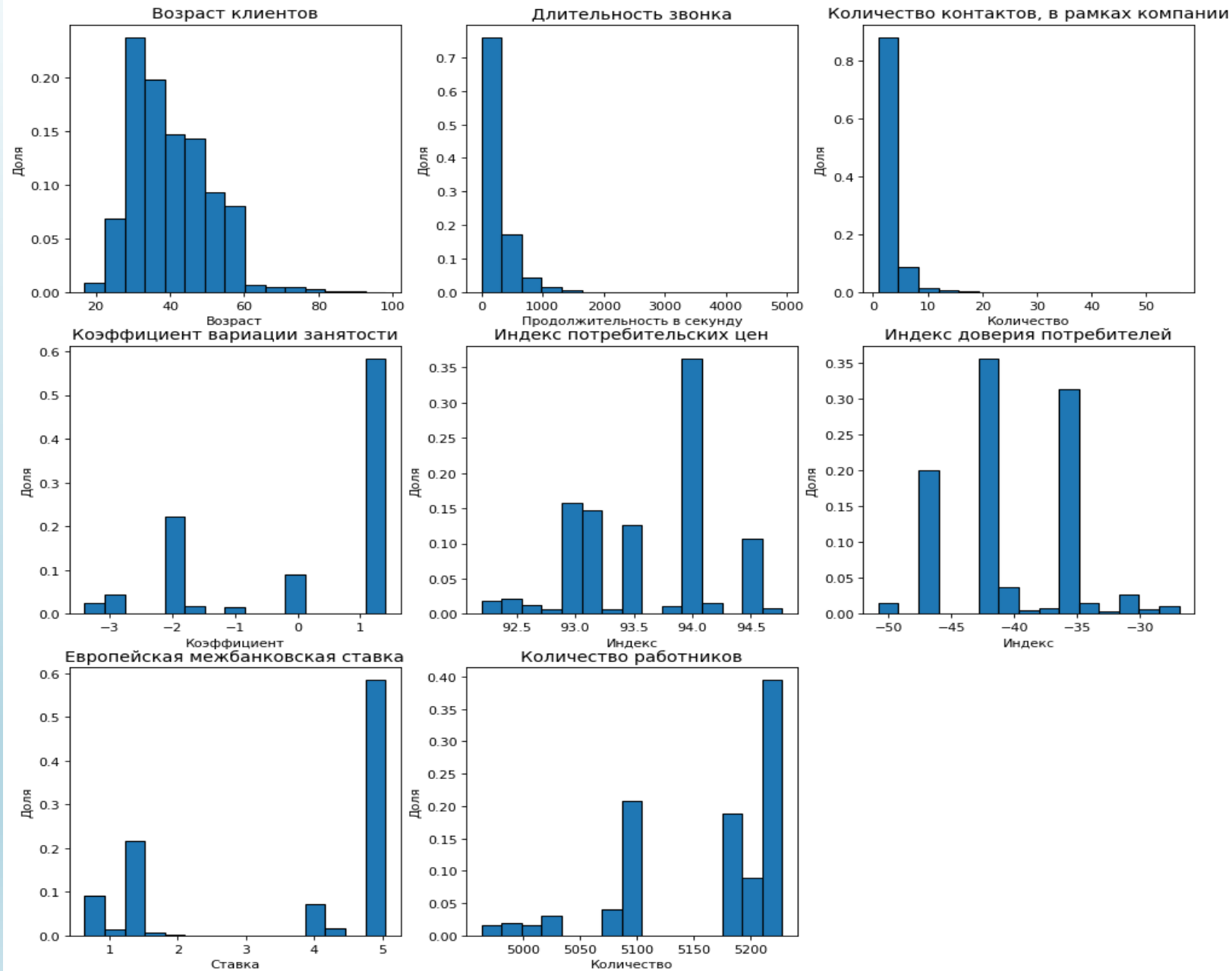
Данные, связанные с проведением предыдущих компаний для этих клиентов.

Данные связанные с атрибутами социального и экономического контекста (индекс потребительских цен, индекс доверия потребителей, коэффициент вариации занятости, европейская межбанковская ставка).

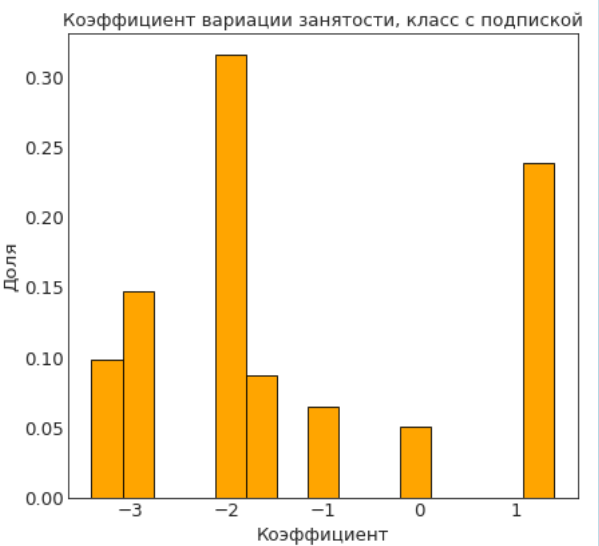
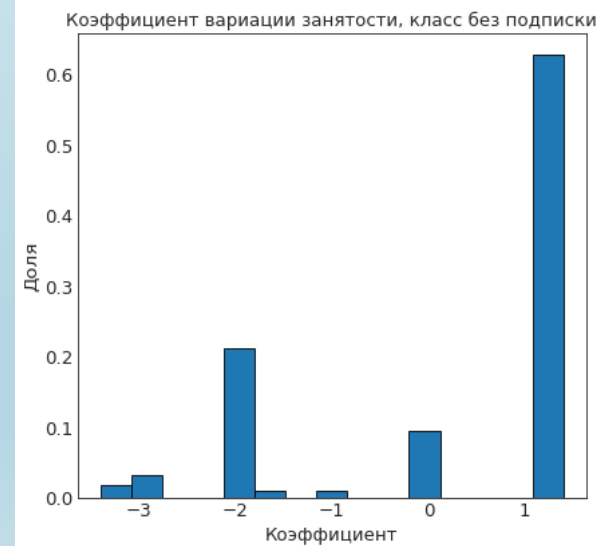
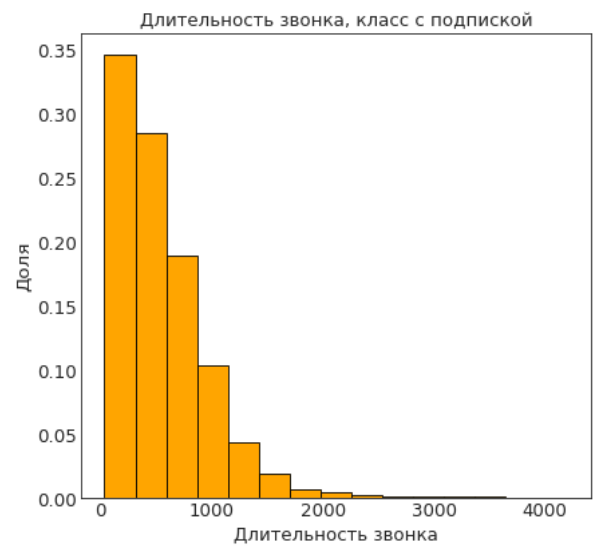
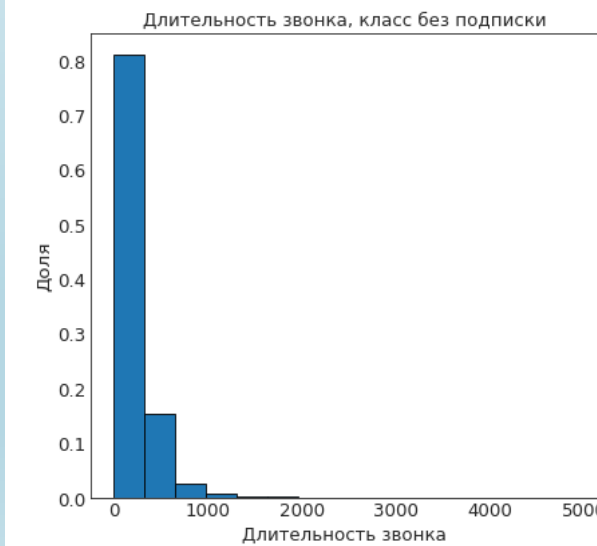
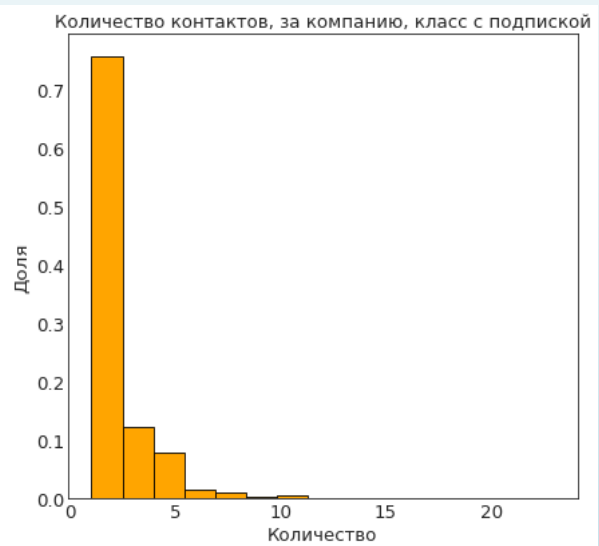
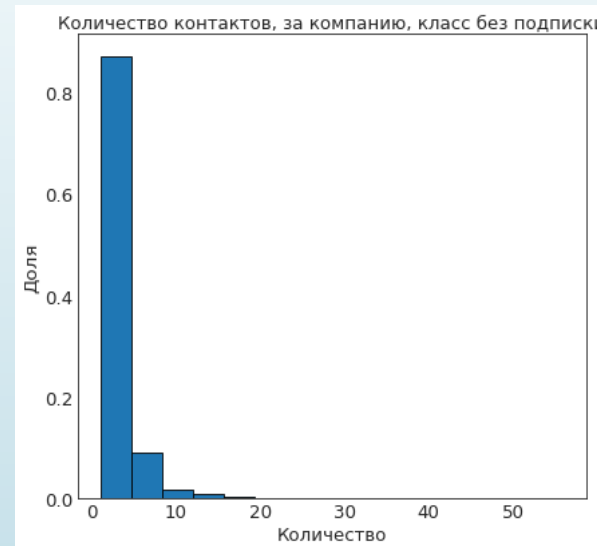
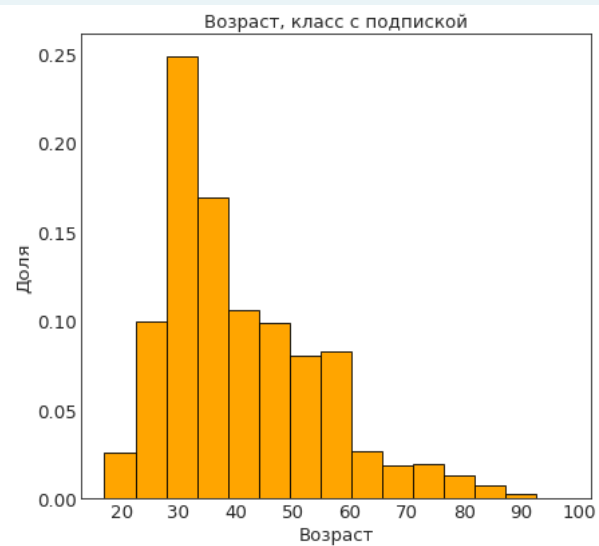
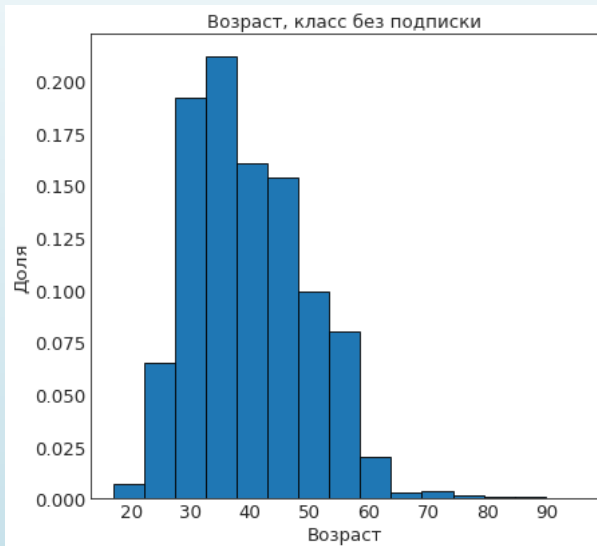
Данные представлены без пропущенных значений, в числовом и категориальном формате.

	Колонки	Тип	Пропущенные значения
0	Возраст	int64	non-null
1	Работа	object	non-null
2	Семейное положение	object	non-null
3	Образование	object	non-null
4	Наличие кредита по умолчанию	object	non-null
5	Наличие кредита на жилье	object	non-null
6	Наличие личного кредита	object	non-null
7	Тип связи	object	non-null
8	Месяц последнего контакта	object	non-null
9	День последнего контакта	object	non-null
10	Длительность звонка	int64	non-null
11	Количество контактов, в этой кампании	int64	non-null
12	Количество дней, с последнего контакта	int64	non-null
13	Количество контактов, до этой кампании	int64	non-null
14	Результат предыдущей маркетинговой кампании	object	non-null
15	Коэффициент вариации занятости	float64	non-null
16	Индекс потребительских цен	float64	non-null
17	Индекс доверия потребителей	float64	non-null
18	Европейская межбанковская ставка	float64	non-null
19	Количество работников	float64	non-null
20	Оформлен ли срочный депозит?	object	non-null

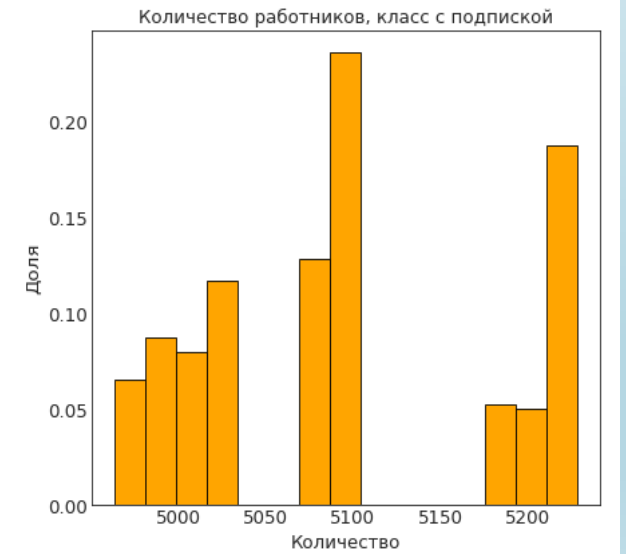
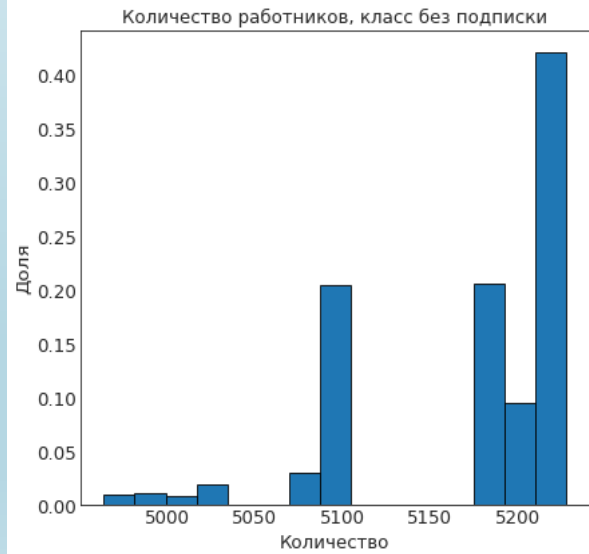
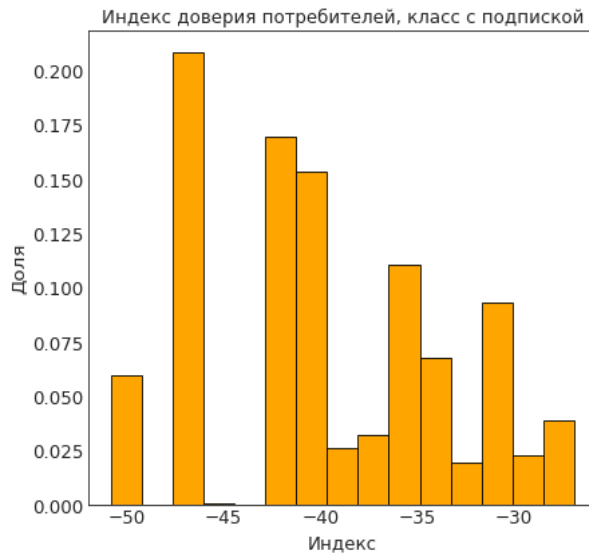
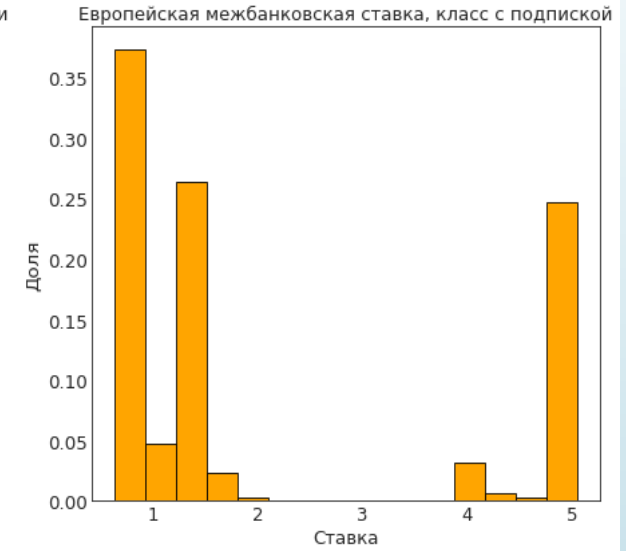
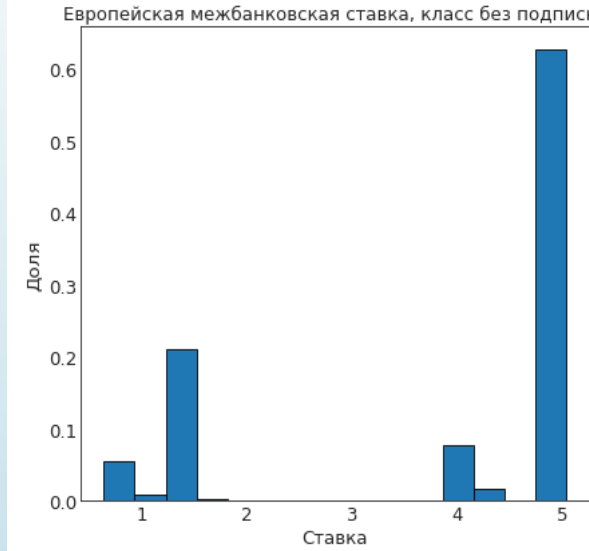
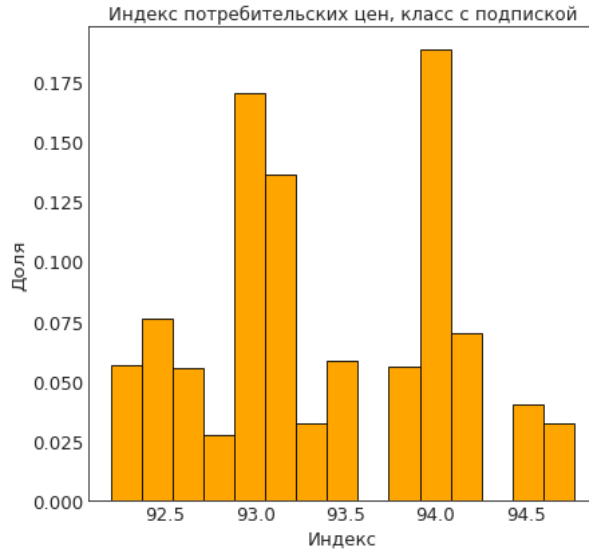
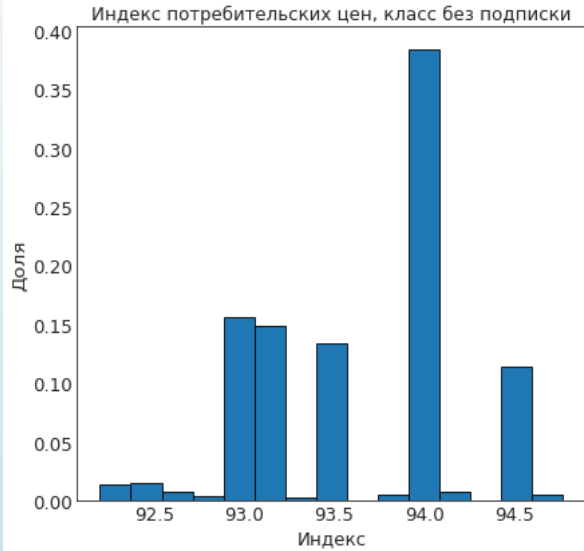
Распределения числовых признаков по всем данным



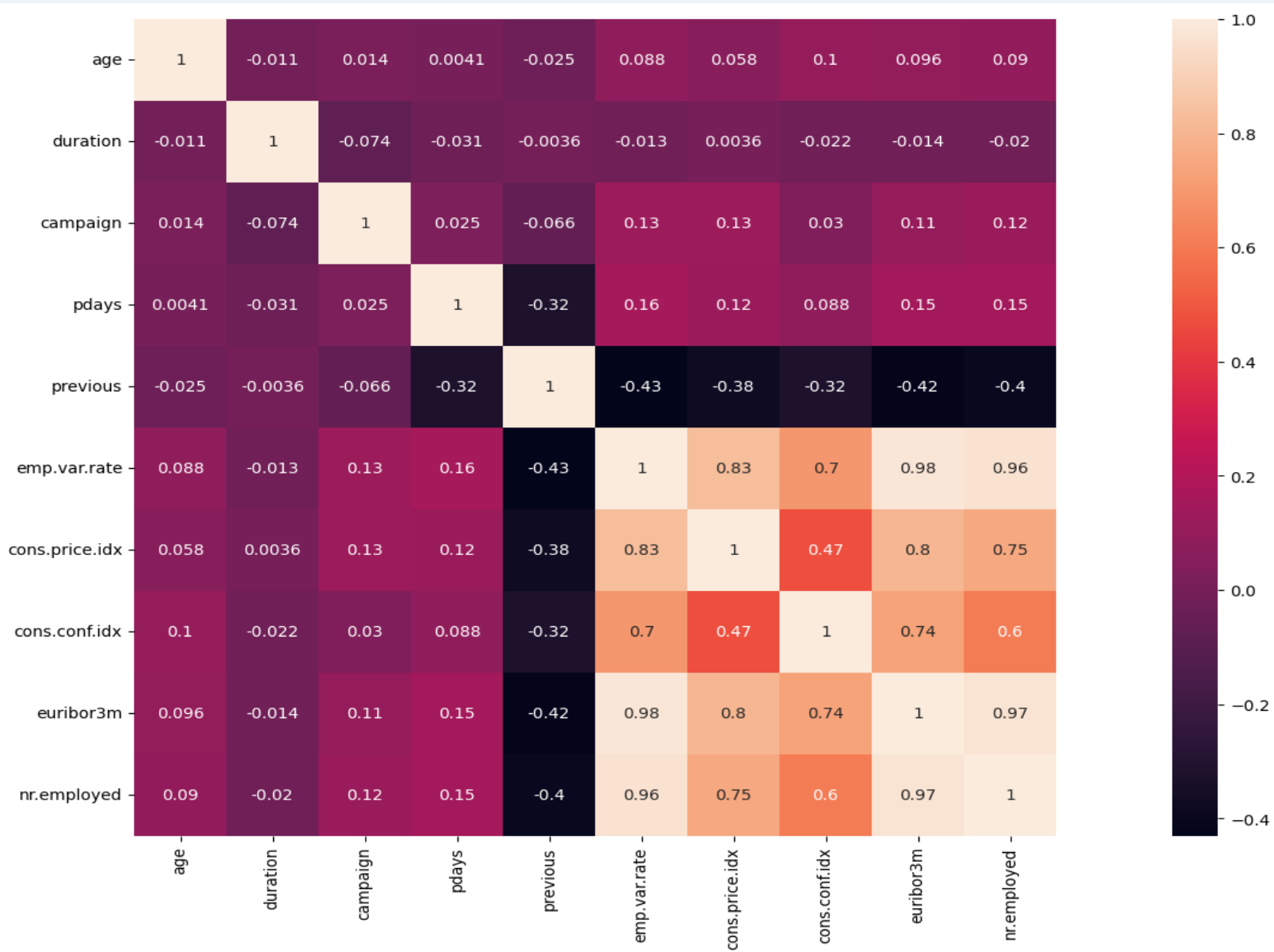
Сравнение распределения данных по классам



Сравнение распределения данных по классам



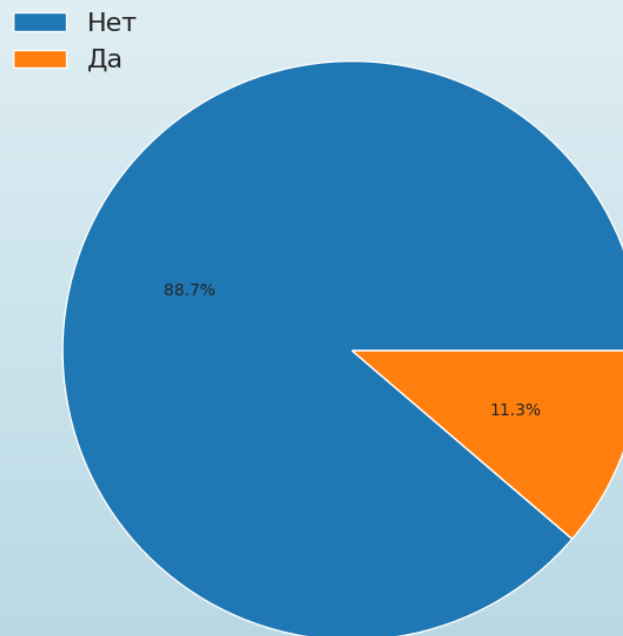
Корреляции числовых факторов



Подготовка данных:

- перекодировка целевой переменной.
- удалим колонку Duration и Default.
- разделим данные на X - факторные признаки и y - целевая переменная.
- произведем деление данных на тестовую, обучающую и валидационную выборку с помощью функции `train_test_split` в пропорциях 70/15/15(с применением `shuffle` и `stratify`).
- перекодируем остальные признаки (проверим способы кодировки `Label Encoder`, `Target Encoder` и функцию `get_dummies`).
- сбалансируем данные (проверим модели `Smote`, `RandomUnderSampler`, `ADASYN`).

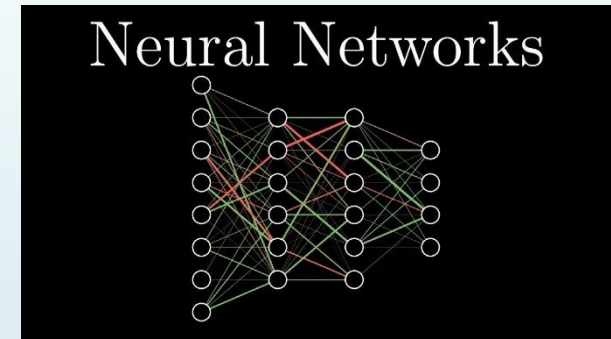
Доля клиентов банка разделенная по подписке



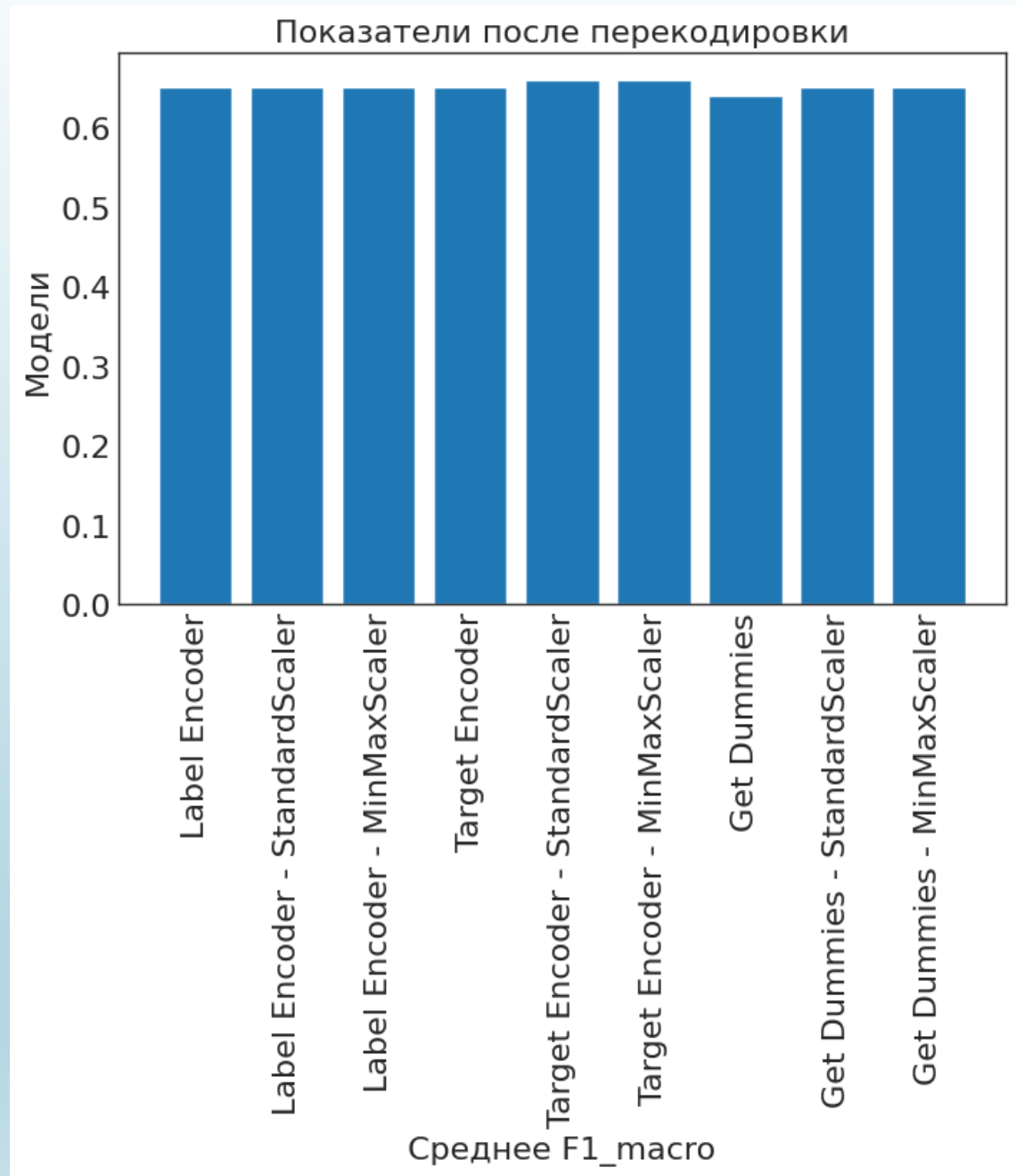
После перекодировки проверим полученные данные на моделях:

LinearDiscriminantAnalysis
Gaussian Naive Bayes
XGBClassifier
RandomForestClassifier
KNeighborsClassifier
MLPClassifier
ExtraTreesClassifier
GradientBoostingClassifier
LogisticRegression
AdaBoostClassifier
QuadraticDiscriminantAnalysis
DecisionTreeClassifier

Используя кросс-валидацию разобьем данные на 10 подвыборок, возьмем метрику F1_macro и сравним среднее по моделям. (так же проверим окажут ли влияние процедуры нормализации и стандартизации).



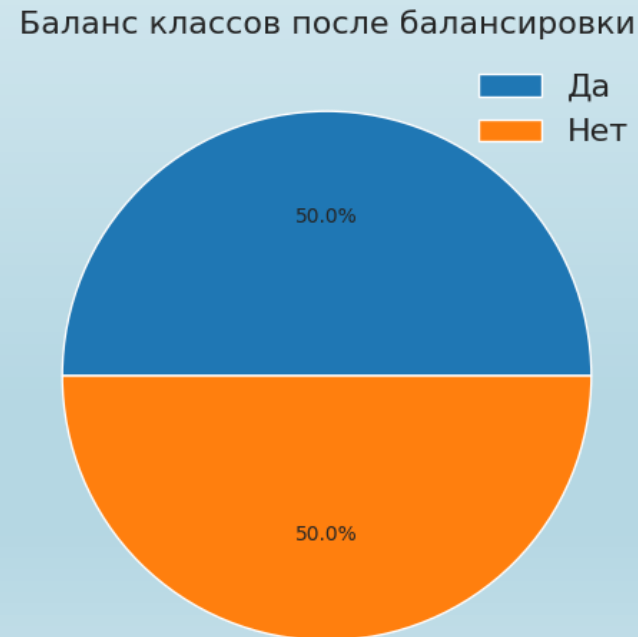
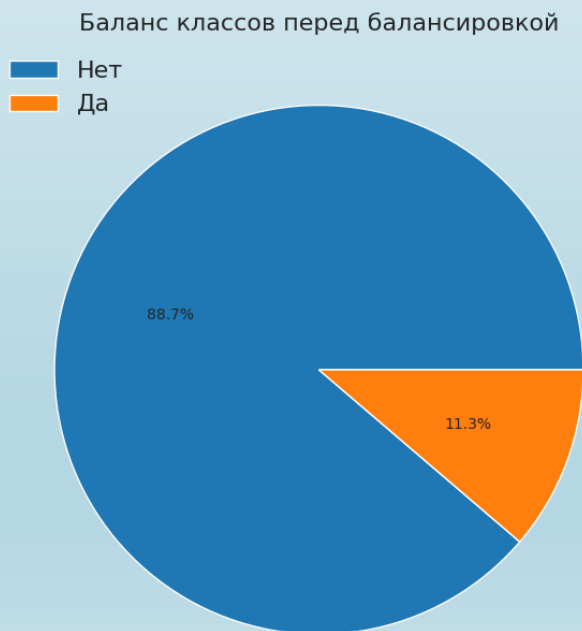
Способ кодировки	Среднее F1_macro	Стандартное отклонение
0 Label Encoder	0.65	0.02
1 Label Encoder - StandardScaler	0.65	0.01
2 Label Encoder - MinMaxScaler	0.65	0.01
3 Target Encoder	0.65	0.02
4 Target Encoder - StandardScaler	0.66	0.01
5 Target Encoder - MinMaxScaler	0.66	0.01
6 Get Dummies	0.64	0.02
7 Get Dummies - StandardScaler	0.65	0.02
8 Get Dummies - MinMaxScaler	0.65	0.02



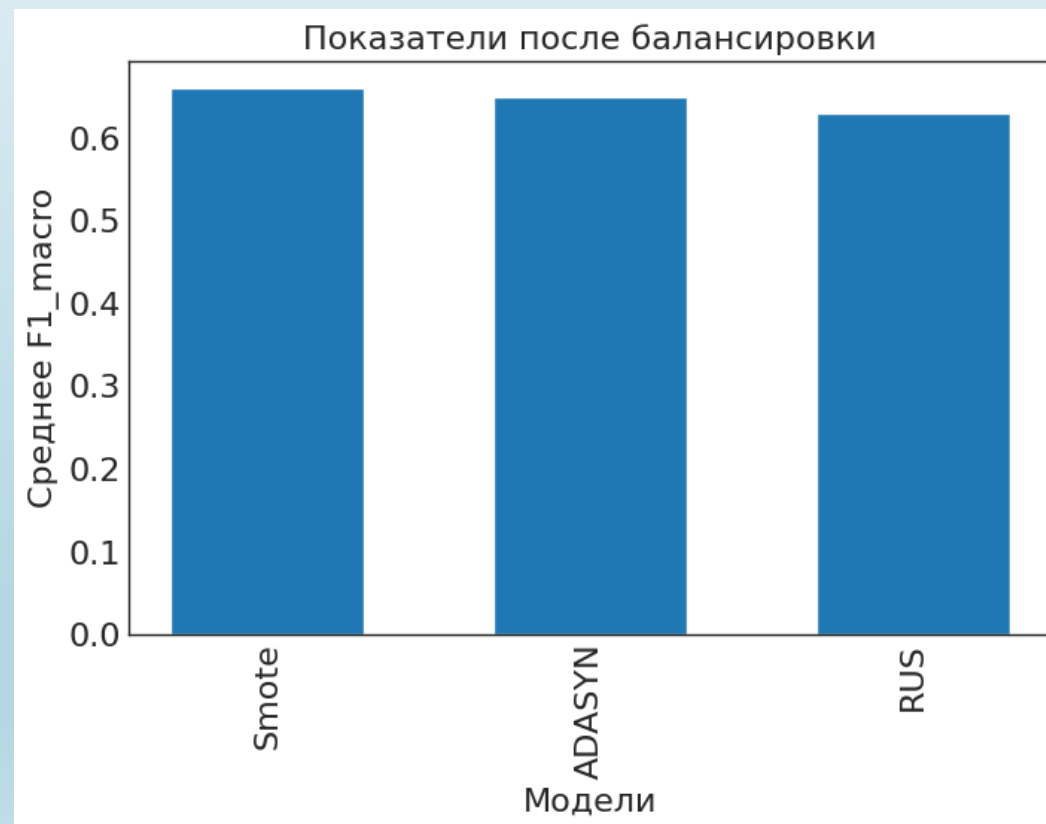
Для перекодировки используем Target Encoder, далее сбалансируем обучающий набор на моделях:

- Smote,
- RandomUnderSampler,
- ADASYN.

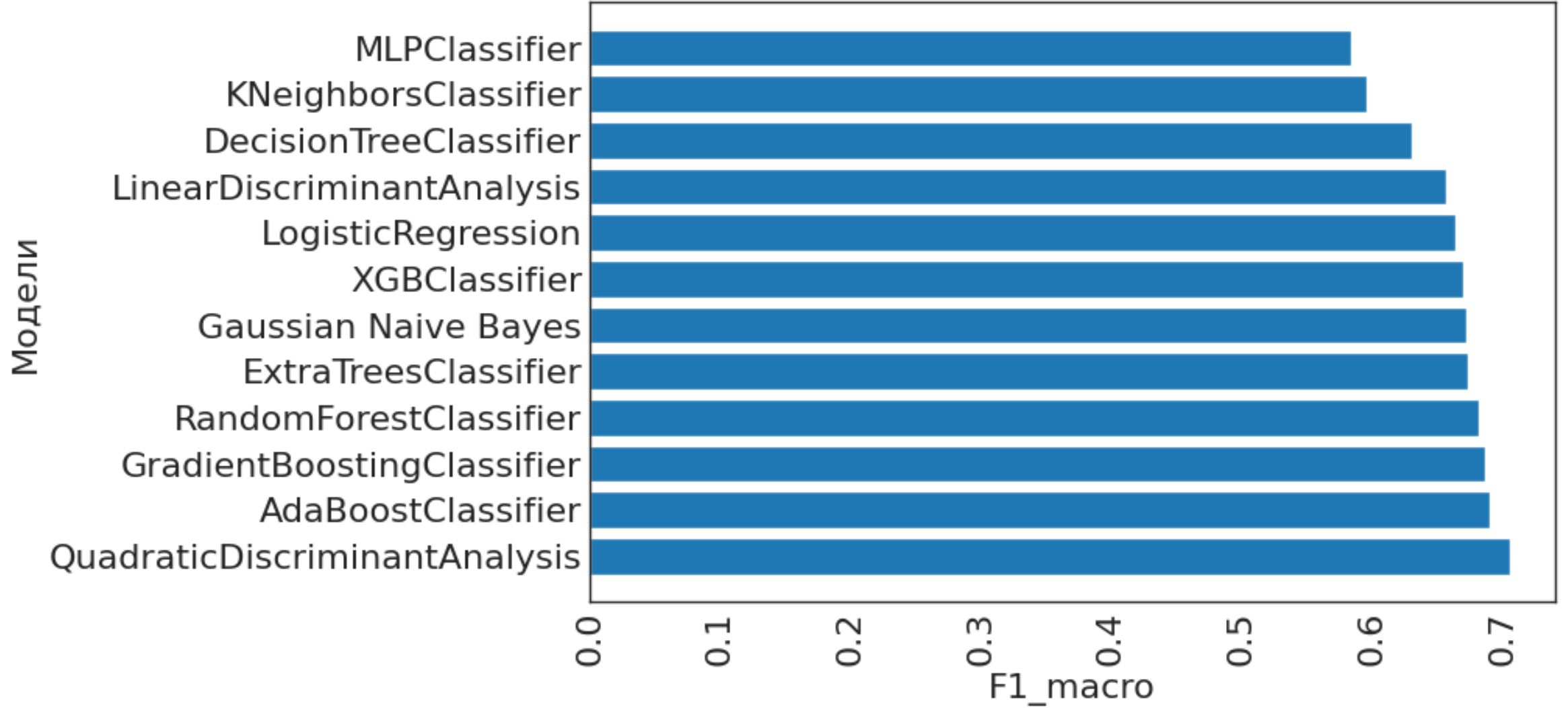
Обучим модели на сбалансированном обучающем наборе и проверим на валидационной несбалансированной выборке.



	Модель	Среднее F1_macro
0	Smote	0.66
1	ADASYN	0.65
2	RandomUnderSampler	0.63



F1_macro на всех моделях



Выберем три модели:

- QuadraticDiscriminantAnalysis
- XGBClassifier
- RandomForestClassifier

Для обучения подготовлен обучающий набор, после обучения сделаем прогноз (predict) на валидационной выборки, и проверим работу модели с помощью classification_report и confusion_matrix.

QuadraticDiscriminantAnalysis

Classification report на валидационном наборе

	precision	recall	f1_score	support
1 класс (0)	0.94	0.91	0.93	4660
2 класс (1)	0.44	0.54	0.48	592
accuracy			0.87	5252
macro avg	0.69	0.72	0.71	5252
weighted avg	0.88	0.87	0.88	5252

Confusion matrix:

	Positive	Negative
Positive	4259	401
Negative	275	317

RandomForestClassifier

Classification report

	precision	recall	f1_score	support
1 класс (0)	0.94	0.96	0.94	4660
2 класс (1)	0.52	0.35	0.42	592
accuracy			0.89	5252
macro avg	0.72	0.65	0.68	5252
weighted avg	0.88	0.89	0.88	5252

XGBClassifier

	precision	recall	f1_score	support
1 класс (0)	0.94	0.97	0.94	4660
2 класс (1)	0.60	0.31	0.41	592
accuracy			0.90	5252
macro avg	0.76	0.64	0.68	5252
weighted avg	0.88	0.90	0.88	5252

Confusion matrix

	Positive	Negative
Positive	4465	195
Negative	384	208

	Positive	Negative
Positive	4537	123
Negative	408	184

RandomForestClassifier

Оптимальные значения гипер-параметров:

```
'criterion': 'log_loss'  
'n_estimators': 800  
'max_depth': 23  
'min_samples_split': 40  
'max_leaf_nodes': 100  
'min_samples_leaf': 30  
'max_features': 'log2'  
'bootstrap': False
```

	precision	recall	f1_score	support
1 класс (0)	0.94	0.92	0.93	4660
2 класс (1)	0.47	0.55	0.51	592
accuracy			0.88	5252
macro avg	0.71	0.74	0.72	5252
weighted avg	0.89	0.88	0.88	5252

	Positive	Negative
Positive	4297	363
Negative	265	327

XGBClassifier

Оптимальные значения гипер-параметров:

```
learning_rate: 0.077  
n_estimators: 228  
max_depth: 17  
gamma: 9  
min_child_weight: 45  
reg_lambda: 1.45  
reg_alpha: 1.93  
scale_pos_weight: 3.84  
subsample: 0.82  
colsample_bytree: 0.40
```

	precision	recall	f1_score	support
1 класс (0)	0.94	0.91	0.93	4660
2 класс (1)	0.45	0.57	0.51	592
accuracy			0.87	5252
macro avg	0.70	0.74	0.72	5252
weighted avg	0.89	0.87	0.88	5252

	Positive	Negative
Positive	4251	409
Negative	252	340

Выберем модель XGBClassifier и проверим на тестовых данных

	precision	recall	f1_score	support
1 класс (0)	0.94	0.91	0.93	5483
2 класс (1)	0.45	0.56	0.50	696
accuracy			0.87	6179
macro avg	0.69	0.74	0.71	6179
weighted avg	0.89	0.87	0.88	6179

	Positive	Negative
Positive	5002	481
Negative	307	389

Выводы:

Сильным дисбаланс классов в данных не позволяет добиться равномерного определения обоих классов (первый класс определяется гораздо полнее и точнее).

Балансировка классов к сожалению не помогла сильно улучшить модели.

Тем не менее, был произведен отбор лучшей модели из представленных:

xgboostclassifier за счет тонкости настроек и гибкости, все же полнее определяет второй класс, что в рамках задачи оптимизации маркетинговой компании, является приоритетной задачей (с точки зрения снижения расходов на проведение такой компании).

Большой объём наблюдений второго класса смог бы увеличить прогностические качества модели.

Спасибо за внимание!