

КОДИИМ

Предсказание возраста клиента по его расходам

Барселона:

Трашинская София

Сербина Злата

Перекрёстова Александра

Калугина Ульяна

Содержание

- Постановка задачи
- Работа с данными
- Построение модели и метрика
- Можно радоваться успеху

КОДИИМ



- **Потребность**

Сделать персонализированные продукты, которые удовлетворяют реальным потребностям клиентов

- **Цель**

Создать алгоритм, определяющий возраст клиентов банка

- **Гипотеза**

Возрастная группа зависит от даты транзакции, категории товаров, суммы транзакции



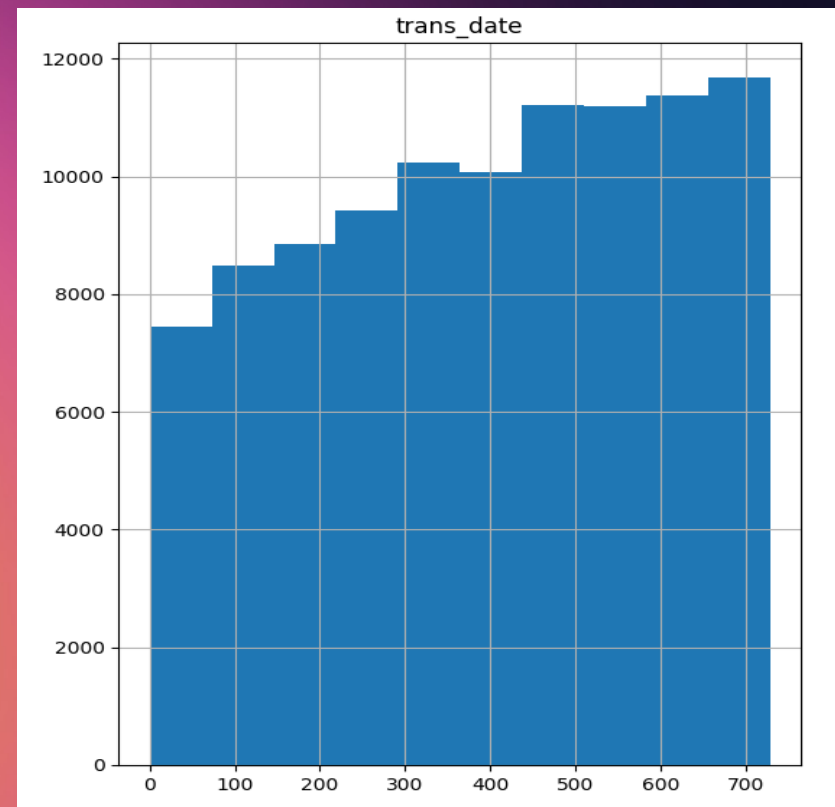
Целевая аудитория

- Магазины розничной торговли, для оперирования ценами на различные категории продуктов
- Интернет магазины, для рекомендации товаров



Описание датасета

- 26.000.000 строк данных
- Целевой признак — возрастная группа, **bins**



	client_id	trans_date	small_group_code	amount_rur	bins
0	46263	539.0	4.0	22.146	1.0
1	19603	715.0	43.0	254.323	2.0
2	2868	722.0	82.0	1.897	0.0
3	759	142.0	20.0	160.594	0.0
4	47654	729.0	18.0	43.048	2.0

Работа с датасетом

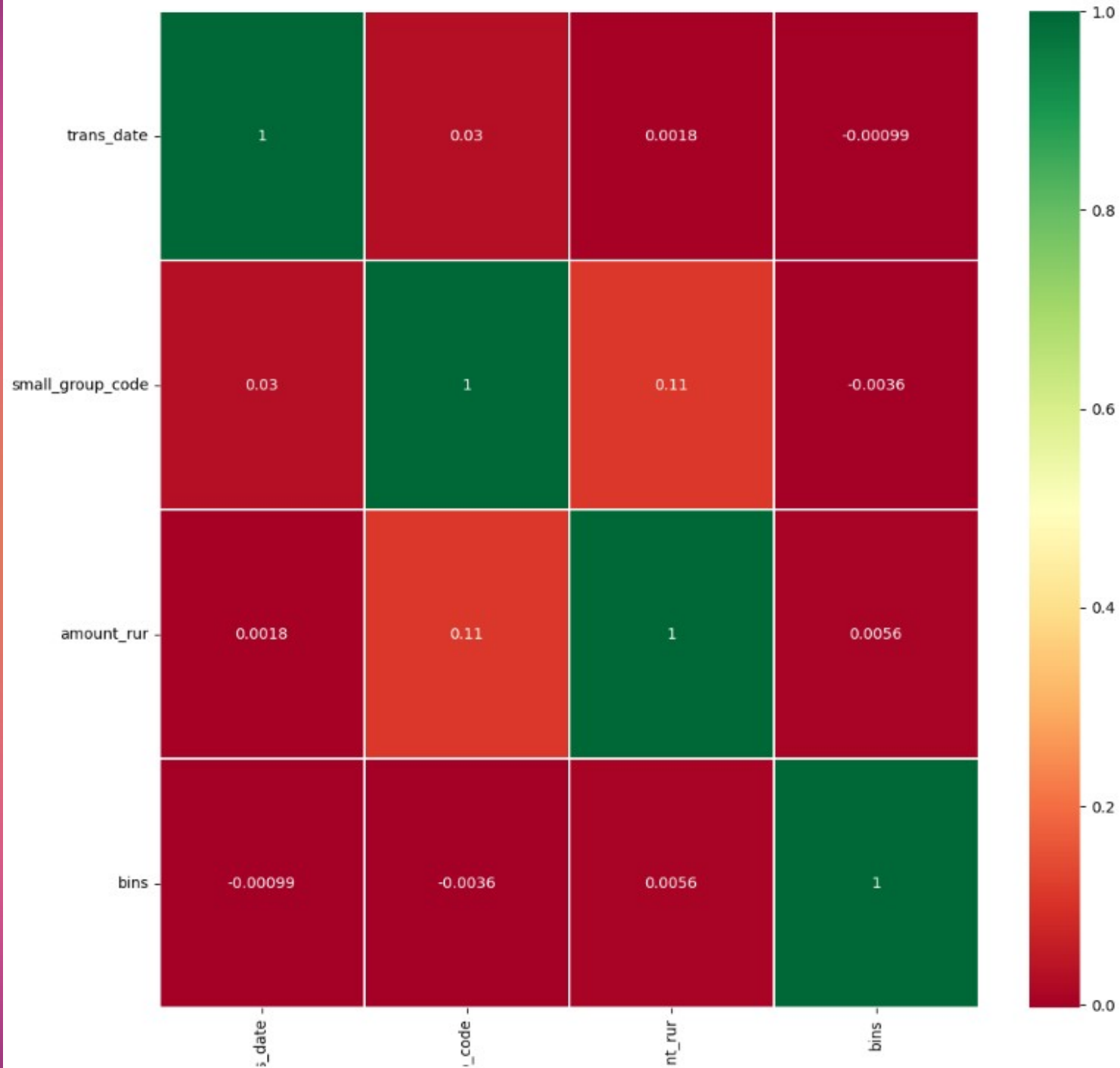
КОДИИМ

- I. Удаление ненужных данных
- II. Подготовка датасета
- III. Анализ датасета

	client_id	trans_date	small_group	amount_rur
count	55312.000000	55312.000000	55312.000000	55312.000000
mean	26027.916402	388.950770	13.443954	66.752129
std	14188.270232	207.794479	19.340799	241.946920
min	582.000000	0.000000	0.000000	0.007000
25%	16476.000000	216.000000	1.000000	11.601000
50%	26959.000000	401.000000	4.000000	25.014000
75%	38624.000000	569.000000	16.000000	57.086250
max	49895.000000	729.000000	152.000000	15094.659000



Корреляция параметров



Метрики

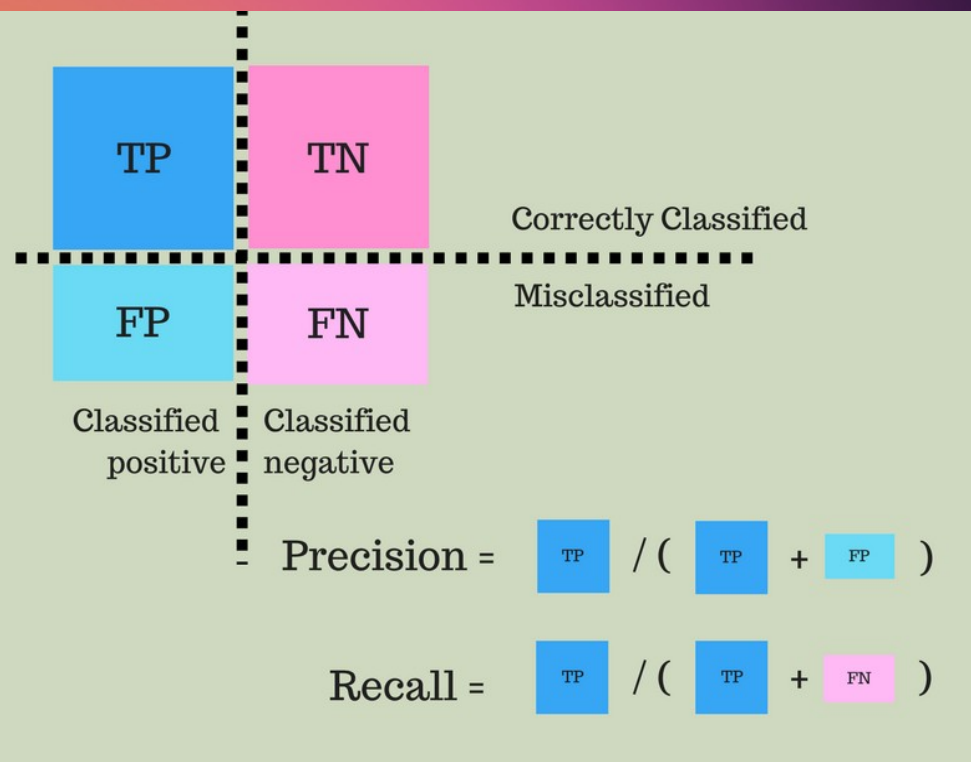
КОДИИМ

Не подходят:

- MAE
- MSE
- WARE

Подходят:

- Accuracy
- Precision and Recall

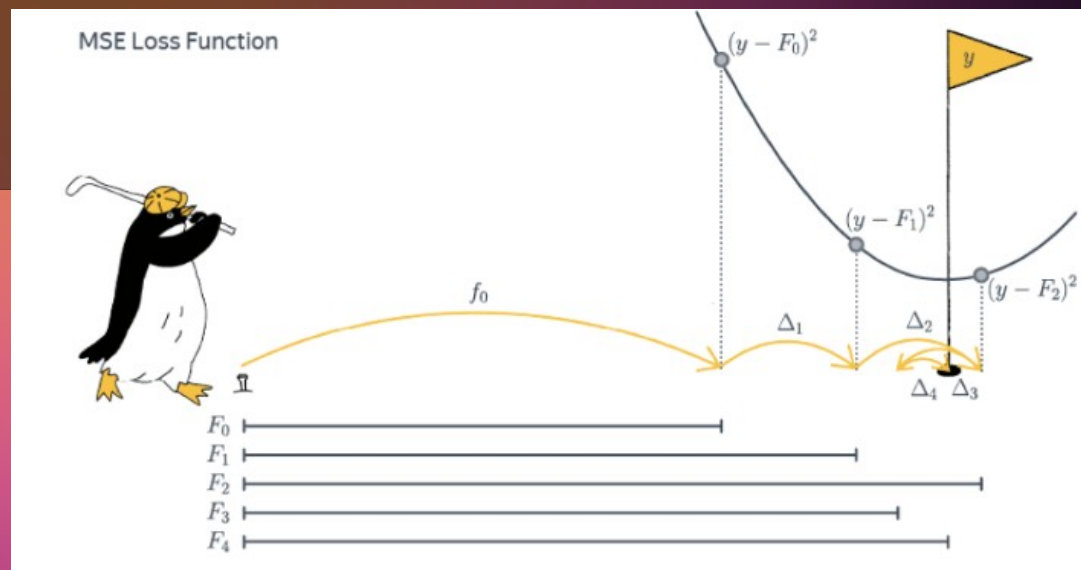


Построение моделей машинного обучения

КОДИИМ

- Библиотеки pandas, scikit-learn, matplotlib и seaborn
- Построение моделей, эксперименты
- Градиентный бустинг

Точность: 57%



GitHub

QR-код:



КОДИИМ

```
### Прописываем параметры для нашей модели
params = {
    'min_samples_split': 10,
    'n_estimators': 100,
    'max_depth': 100,
    'learning_rate': 0.05,
    'min_samples_leaf': 16,
    'min_samples_split': 16
}
```

```
] : ### Тренируем
gbc = GradientBoostingClassifier(**params)
gbc.fit(X_train_wo, y_train_wo)
```

```
] : GradientBoostingClassifier(learning_rate=0.05, max_depth=100,
                               min_samples_leaf=16, min_samples_split=16)
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
] : ### Вычисляем точность
train_accuracy_score=gbc.score(X_train_wo, y_train_wo)
print(train_accuracy_score)

test_accuracy_score=gbc.score(X_test_wo, y_test_wo)
print(test_accuracy_score)

### Предсказание
y_pred_wo = gbc.predict(X_test_wo)
```

КОДИИМ

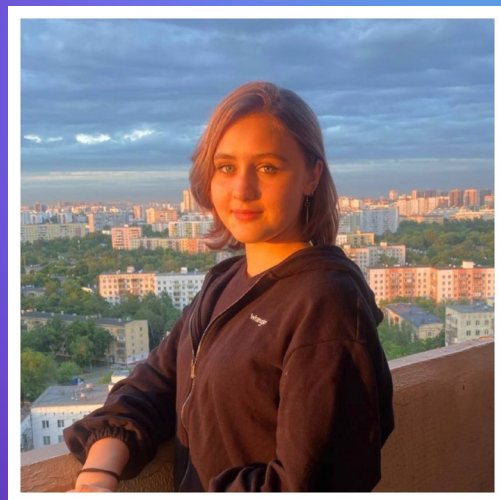
Наша команда

Перекрестова Александра

Калугина Ульяна



Сербина Злата



Трашинская София



Источники:

- <https://ods.ai/competitions/sberbank-sirius-lesson>
- <https://proglib.io/p/reshaem-zadachi-mashinnogo-obucheniya-s-pomoshchyu-algoritma-gradientnogo-bustinga-2021-11-25>

КОДИИМ

Спасибо за внимание!

