

Для моделирования выбран CatBoost, т.к. показал лучший результат с минимальной предобработкой

Преимущества:

- CatBoost имеет встроенную поддержку как процессора, так и графического процессора (но я этим не воспользовалась, в следующий раз)
- можем использовать категориальные функции напрямую с CatBoost, алгоритм автоматически кодирует категориальные признаки
- CatBoost обеспечивает отличные результаты со значениями параметров по умолчанию

Также использована библиотека **Auto_ViML**:

- позволяет быстро и автоматически сделать развед. анализ и отбор признаков (Feature Selection By XGBoost)
- получить визуальное представление: Auto_ViML_Linear_Model.ipynb

для развед. анализа использована библиотека **SweetViz**

- Программы генерируют красивые и полезные графики (в том числе)

Принцип составления индивидуальных предложений для выбранных абонентов

- пропустить данные выбранных абонентов (профиль) через модель и на основании предикта предлагать или нет конкретную услугу
- если мы не хотим беспокоить абонентов и быть уверенными в том, что предложенная услуга будет приобретена, нужно выставить ограничение по точности Precision (например $\geq 80\%$) и максимизировать полноту Recall
- если же заказчик требует чтобы мы нашли как можно больше абонентов которые могут купить конкретную услугу, то есть ограничение на полноту Recall (допустим $\geq 80\%$), и максимизировать точность Precision

пример:

```

❏ Normal Balanced Accuracy = 50.00%
Test results since target variable is present in test data:
[[96478    0]
 [ 7474    0]]
      precision    recall  f1-score   support

      0       0.93      1.00      0.96      96478
      1       0.00      0.00      0.00       7474

   accuracy              0.93      103952
  macro avg       0.46      0.50      0.48      103952
 weighted avg       0.86      0.93      0.89      103952

Modified Threshold Balanced Accuracy = 80.94%
[[85610 10868]
 [ 2007  5467]]
      precision    recall  f1-score   support

      0       0.98      0.89      0.93      96478
      1       0.33      0.73      0.46       7474

   accuracy              0.88      103952
  macro avg       0.66      0.81      0.69      103952
 weighted avg       0.93      0.88      0.90      103952

```

Финальная модель (0.5):

```

[ ] print_score(y_test, model_y_pred)
    print(classification_report(y_test, model_y_pred > 0.5))

[0.7314240052788819, 0.7422912579238272, 0.74769494174991]
      precision    recall  f1-score   support

      0.0       0.98      0.90      0.94      185817
      1.0       0.45      0.79      0.57       18545

   accuracy              0.89      204362
  macro avg       0.71      0.85      0.76      204362
 weighted avg       0.93      0.89      0.91      204362

```