Курсовой проект от компании Megafon

В данном проекте осуществляется предсказание возможности подключения услуги клиентом компании, для формирования ему предложения.

**Входной набор данных:**

* файл data\_train.csv с набором признаков: id, vas\_id, buy\_time, target
* файл features.csv.zip с дополнительным набором признаков клиентов

**Тестовый набор данных:**

* Файл data\_test.csv с набором признаков: id, vas\_id, buy\_time

**Выходной набор данных:**

* Файл с предсказаниями answers\_test.csv с набором признаков: id, vas\_id, buy\_time, target
* Работающая модель в формате pickle - model.pkl
* Код модели в виде jupyter-ноутбука
* Описание работы в виде файла «Описание работы» в формате pdf

**Целевая переменная** target. Для которой, 1 означает подключение услуги абонентом, 0 – не подключение услуги.

**Метрика качества** для оценки результата обучения модели - F1 score.

Загрузка данных, формирование датафреймов

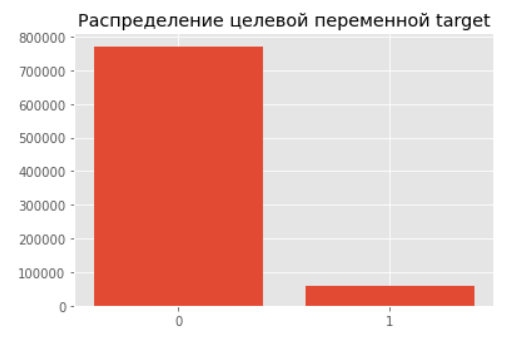
Так как файл features.csv слишком большой, для работы с ним была использована библиотека dask.

Файл data\_train.csv соединён с файлом features.csv посредством inner join по полям id и buy\_time.

Из результирующего набора полей, удалены служебные поля Unnamed: 0\_x', 'Unnamed: 0\_y

Итог – файл X\_train, для последующей загрузки в pipline.

После оценки распределения целевой переменной был получен следующий график



Очевидно, что присутствует дисбаланс классов, что может негативно сказаться на обучении модели для значения целевой переменной 1.

*При подобном раскладе, метрика f1\_score для модели градиентного бустинга, составляла 0.15*

После балансировки классов, количество объектов со значениями целевой переменной 0 и 1 стало равным по 2642

Из датафрейма X\_train, было удалено поле target, создан новый датафрейм Y\_train, со значениями данного поля. Таким образом, был получен итоговый набор данных: X\_train и Y\_train, для загрузки в pipeline

Создание Pipeline

Произведена оценка признаков в датасете X\_train:

Все признаки 256

Константные признаки 16

Вещественные признаки 248

Бинарные признаки 1

Категориальные признаки 1

Остальные признаки 0

В таком виде данные признаки загружены в pipeline.

Обработка пропусков выполнена следующим образом:

* вещественные признаки заполнены средним значением,
* категориальные – наиболее часто встречающимся. Игнорируются неизвестные значения

Для сравнения прогноза результата было использовано две модели – логистическая регрессия и градиентный бустинг. В обоих случаях подбирались оптимальные параметры при помощи GridSearchCV.

Далее подробно рассмотрены результаты тестирования каждой модели.

Логистическая регрессия

Значение f1 score при различных параметрах strategy: ['most\_frequent', 'constant']

Best f1 score: 0.85

Best parameters set found on development set:

{'pipeline\_\_featureunion\_\_categorical\_features\_\_simpleimputer\_\_fill\_value': -1, 'pipeline\_\_featureunion\_\_categorical\_features\_\_simpleimputer\_\_strategy': 'most\_frequent'}

Grid scores on development set:

0.853 (+/-0.005) for {'pipeline\_\_featureunion\_\_categorical\_features\_\_simpleimputer\_\_fill\_value': -1, 'pipeline\_\_featureunion\_\_categorical\_features\_\_simpleimputer\_\_strategy': 'most\_frequent'}

0.853 (+/-0.005) for {'pipeline\_\_featureunion\_\_categorical\_features\_\_simpleimputer\_\_fill\_value': -1, 'pipeline\_\_featureunion\_\_categorical\_features\_\_simpleimputer\_\_strategy': 'constant'}

Сравнение предсказания модели с известными значениями целевой переменной:

Логистическая регрессия

precision recall f1-score support

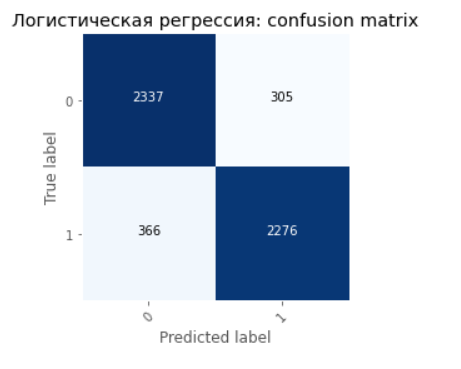
0 0.86 0.88 0.87 2642

1 0.88 0.86 0.87 2642

accuracy 0.87 5284

macro avg 0.87 0.87 0.87 5284

weighted avg 0.87 0.87 0.87 5284



Градиентный бустинг

Значение f1 score при различных параметрах max\_depth [3, 5], n\_estimators": [50, 100]

Best f1 score: 0.88

Best parameters set found on development set:

{'gradientboostingclassifier\_\_max\_depth': 3, 'gradientboostingclassifier\_\_n\_estimators': 100}

Grid scores on development set:

0.872 (+/-0.007) for {'gradientboostingclassifier\_\_max\_depth': 3, 'gradientboostingclassifier\_\_n\_estimators': 50}

0.876 (+/-0.005) for {'gradientboostingclassifier\_\_max\_depth': 3, 'gradientboostingclassifier\_\_n\_estimators': 100}

0.875 (+/-0.003) for {'gradientboostingclassifier\_\_max\_depth': 5, 'gradientboostingclassifier\_\_n\_estimators': 50}

0.874 (+/-0.003) for {'gradientboostingclassifier\_\_max\_depth': 5, 'gradientboostingclassifier\_\_n\_estimators': 100}

Сравнение предсказания модели с известными значениями целевой переменной:

Градиентный бустинг

precision recall f1-score support

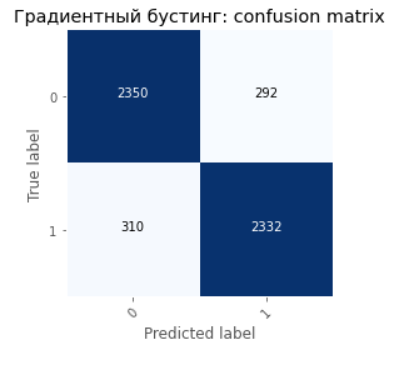
0 0.88 0.89 0.89 2642

1 0.89 0.88 0.89 2642

accuracy 0.89 5284

macro avg 0.89 0.89 0.89 5284

weighted avg 0.89 0.89 0.89 5284



Заключение

Сравнивая результаты двух моделей, видно, что метрика f1 score выше у градиентного бустинга - 0.89, против 0.87 у логистической регрессии.

В результате выбрана модель градиентного бустинга т.к. у ней выше f1 score и true negativ true positiv лучше, чем у логистической регрессии.

Результат предсказания на тестовых данных имеет значения в диапазоне от 0 до 1.

Принято решение для целевой переменной target проставлять:

* 1 для значения больше или равно 0.5
* 0 для значений меньших 0.5