**Project - Bank credit scoring**

**Автор: Александра Гвоздева**

## Задание

Построение модели для прогнозирования вероятности дефолта заемщика.

### Описание признаков

* client\_id - идентификатор клиента
* education - уровень образования
* sex - пол заемщика
* age - возраст заемщика
* car - флаг наличия автомобиля
* car\_type - флаг автомобиля иномарки
* decline\_app\_cnt - количество отказанных прошлых заявок
* good\_work - флаг наличия “хорошей” работы
* bki\_request\_cnt - количество запросов в БКИ
* home\_address - категоризатор домашнего адреса
* work\_address - категоризатор рабочего адреса
* income - доход заемщика
* foreign\_passport - наличие загранпаспорта
* sna - связь заемщика с клиентами банка
* first\_time - давность наличия информации о заемщике
* score\_bki - скоринговый балл по данным из БКИ
* region\_rating - рейтинг региона
* app\_date - дата подачи заявки
* default - флаг дефолта по кредиту

### В результате работы над данными:

* был произведен разведывательный анализ данных
* была произведена очистка данных и обработка выбросов (в финальной версии удаления и сглаживания выбросов нет, так как это ухудшало качество модели, оставлено только логарифмирование и использование RobustScaler для непрерывных признаков)
* созданы новые признаки
* отобраны признаки для модели по значимости
* подобраны оптимальные гиперпараметры для модели
* проведена проверка модели на переобучение - результат отрицательный

**Были достигнуты следующие показатели:**

* ROC AUC - 0.74
* F1-score - 0.34
* Precision score - 0.67
* Recall score - 0.23

### Выводы

* Разведывательный анализ данных показал, что данные достаточно чистые. Пропусков немного (только в колонке education). Пропуски в колонке education были заполнены случайными значениями из уже существующих.
* Колонки в датасете делятся на бинарные, категориальные и с непрерывным типом данных. Бинарные колонки были обработаны с помощью LabelEncoder, категориальные - с помощью OneHotEncoder.
* Некоторые колонки с непрерывным типом данных ('decline\_app\_cnt','bki\_request\_cnt','income') имеют "тяжелый" правый хвост. Это колонки были прологарифмированы. Операция была проведена над объединенным датасетом
* Были сгенерированы дополнительные непрерывные признаки (4 колонки) и категориальные признаки (3 колонки). Новые категориальные признаки повышают качество модели. Новые непрерывные признаки сильно скоррелированы, поэтому их использование ухудшает качество модели. Три наиболее скоррелированных признака были обработаны методом главных компонент. Это дало улучшение целевой метрики.
* Была предпринята попытка сгенерировать полиномиальные признаки (вручную и с помощью функции Polynomial Features). Полиномиальные признаки ухудшают модель, поэтому было принято решение от них отказаться. Но: позднее выяснилось, что некоторые полиномиальные признаки просто были сильно скоррелированы. Я оставила только некоторые из них, это дало улучшение модели.
* Работа с выбросами в данном датасете не принесла улучшения качества модели. Выбросы были классифицированы методом IQR. Т.к. удаление выбросов приводит к ухудшению качества модели, выбросы было решено оставить.
* Был использован RobustScaler для нормализации непрерывных признаков.
* Наибольшее значение для модели имеют признаки score\_bki и sna.
* Была сделана попытка удалить наименее значимые для модели признаки из числа категориальных и непрерывных. Удаление наименее значимого категориального признака (app\_day\_of\_week) не приводит к улучшению модели. Однако, удаление наименее значимого непрерывного признака (age) улучшает целевую метрику. Если удалить следующий по значимости непрерывный признак (pca\_component), то метрика ухудшится.
* Было испробовано добавление параметра WOE и IV для переменных age, sex и sna. Это не принесло улучшения качества модели, поэтому от параметров WOE и IV было решено отказаться.
* С помощью метода GridSerchCV были подобраны оптимальные параметры для модели. Оказалось, что оптимальная модель - это Logistic Regression Balanced.
* Был подобран гиперпараметр С=0.01, что соответствует максимальному вкладу регуляризации. Это дало улучшение целевой метрики по сравнению с С=1.0 (значение по умолчанию).
* Было испробовано деление выборки на обучающую и тестовую с помощью StratifiedShuffleSplit (деление на части с учетом долей классов). Это не дало увеличения целевой метрики. От данного способа деления выборки было принято решение отказаться в пользу train\_test\_split.
* На обучающей и тестовой выборке были измерены основные метрики (f1\_score, precision\_score, recall\_score, accuracy\_score). Т.к. значения метрик на обучающей и на тестовой выборке практически не отличаются, можно сделать вывод, что переобучения модели нет.