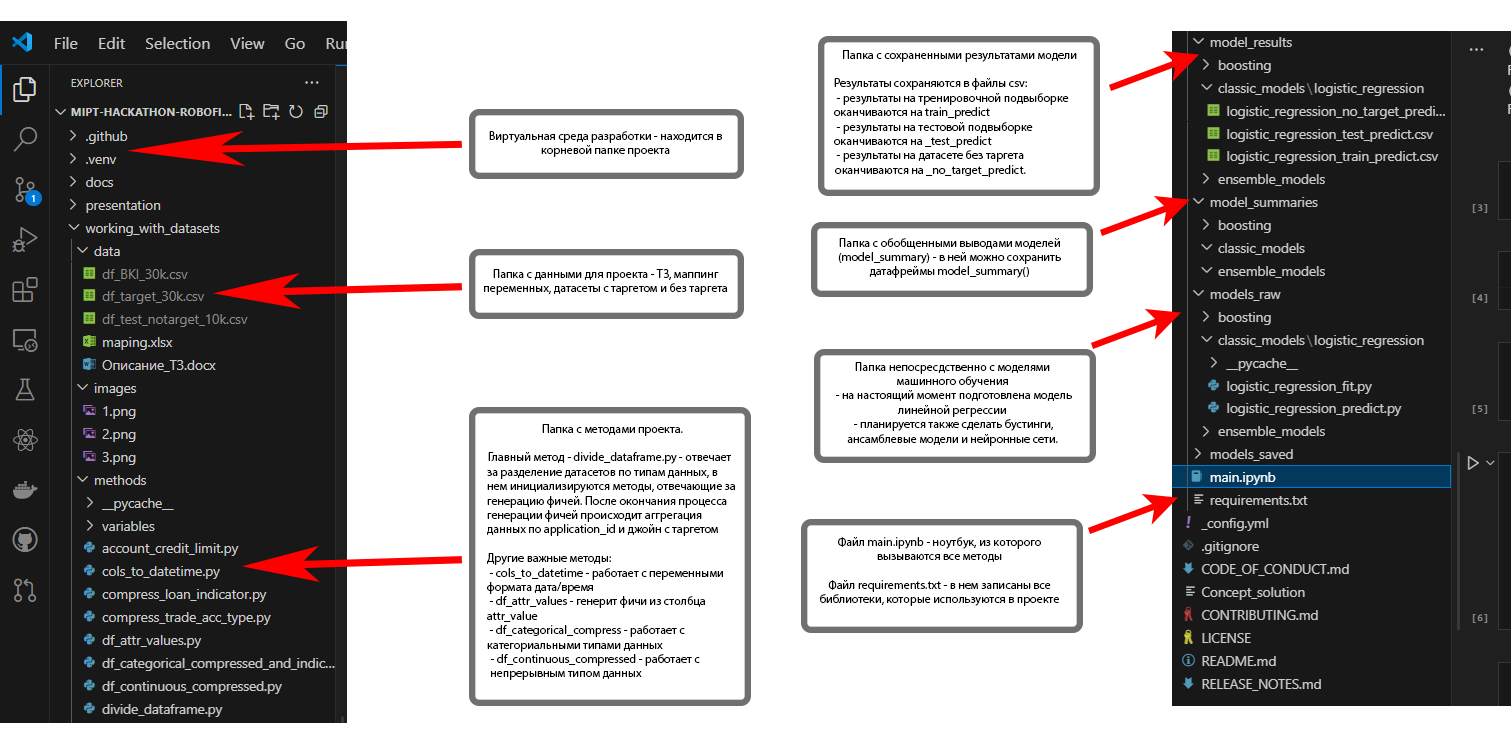
Разработка интерпретируемой скоринговой модели для оценки кредитоспособности клиентов сегмента PDL

Отчет о проделанной работе

1. Структура проекта



Основные библиотеки, используемые в проекте:

* matplotlib
* mlxtend
* numpy
* pandas
* scikit-learn

Точка входа – файл main.ipynb, в нем происходит вызов методов, отвечающих за извлечение, обработку данных, инициализацию моделей машинного обучения (МО), сохранение обученных моделей в бинарный формат, а также извлечение обученных моделей из бинарного формата с целью их повторного использования.

В папке methods находятся различные методы, выполняющие различные манипуляции с данными. В процессе работы над проектом мы старались придерживаться принципов чистоты кода, и каждый метод помещали в отдельный файл (за исключением некоторых небольших вспомогательных функций, которые мы писали в дополнение к основному методу).

Главным методом является метод divide\_dataframe (расположен в одноименном файле в папке methods). В нем происходит разделение столбцов датафрейма по типам (дата/время, категориальные, непрерывные), с последующим применением к различным переменным методов по их аггрегации, а именно:

* Из переменных типа дата/время мы извлекли в основном непрерывные данные (например, дюрация кредита, количество открытых кредитов, количество закрытых кредитов). Также из переменных дата/время мы вывели фичу negative\_scoring: если в ячейках legal\_items\_court\_act\_dt, due\_arrear\_start\_dt, past\_due\_dt, past\_due\_principal\_missed\_date, past\_due\_int\_missed\_date – имелась дата, значит у клиента по данному кредиту имелась просрочка различной степени тяжести. Количество дат в этих ячейках суммировалось (1 ячейка с датой – 1 балл); чем больше дат имелось в этих ячейках, тем больше составлял negative\_scoring. Затем данная переменная делилась на 4 категории (no\_overdue, mild\_overdue, moderate\_overdue, severe\_overdue), и с помощью метода pandas.get\_dummies() превращалась в индикаторный тип. Далее все переменные аггрегировались: индикаторные по сумме в каждой ячейке, для непрерывных считались среднее значение, сумма, минимальное и максимальное значения
* Непрерывные переменные (метод df\_continuous\_compressed) в основном аггрегировались посредством методов min, max, sum, mean
* Категориальные переменные (метод df\_categorical\_compressed) переводились в индикаторные, затем значения в соответствующих столбцах аггрегировались по сумме

Скрипты для тренировки и обучения моделей находятся в папке models\_raw. В данной папке имеются директории для классических моделей МО, ансамблевых моделей и бустингов. В данное время обучена только модель линейной регрессии - директория models\_raw/classic\_models/logistic\_regression.

В этой папке имеется файл logistic\_regression\_fit.py, ответственный за обучение модели и ее сохранение в файл формата pickle (в папку models\_saved) и вывод на печать графиков (optb.binning\_table.plot()) и параметров переменных/фичей (optb.binning\_table.build()); а также файл logistic\_regression\_predict.py, в котором находятся 2 метода: один работает с датасетом, имеющим таргет, другой – с датасетом без таргета.

В этих методах происходит загрузка обученной модели из файла pickle и ее применение на выборке, которую укажет разработчик. После применения модели на данных происходит сохранение предсказанных значений в файлы csv, в папке model\_results. Папки models\_saved, model\_results, model\_summaries (последняя – если возникнет необходимость сохранить model.summary()) разделены по тому же принципу, что и папка models\_raw – в них имеются директории для классических моделей МО, ансамблевых моделей и бустингов.

В прцессе работы над проектом нашей команде удалось сделать следующие пункты ТЗ:

1. Анализ входных данных
2. Создание датасетов, генерация сырых фичей
3. Объединение датасетов с фичами и таргетами
4. Разбивка датасета на трейн/тест
5. Биннинг и группировка фичей
6. Предварительный отбор фичей
7. Многофакторный анализ и отбор фичей в модель – частично (фичи отобрали по наибольшим значениям IV, Gini
8. Построение итоговой модели
9. Оценка всех трех выборок итоговой моделью
10. Оформление word-отчета
11. Презентация

Не сделано:

1. Gini фич во времени на объединенной выборке
2. Многофакторный анализ и отбор фичей с использованием библиотеки Mlxten, отбор наилучшей комбинации фичей (Corr < 0.7 VIF < 10),
3. построение графиков Gini модели на трейн и тесте в зависимости от кол-ва фич и их состава.

Графики, полученные в результате работы программы:

