Построение модели кредитного скоринга для предсказания рейтинга клиента банка с применением машинного обучения

Владимир Полухин

Введение

Начнем с определения: Система кредитного скоринга - это система оценки заемщика, с помощью которой банки и крупные микрофинансовые организации (МФО) могут предсказать, насколько аккуратно человек будет выплачивать кредит. Соответсвенно нашей задачей является реализация данной системы с применением машинного обучения, современных фреймворков по работе с данными, и, возможно, нейтронных сетей, которые могут быть не использованы для решения данной задачи, так как я считаю, что мы способны построить достойную модель без применения нейросетевых технологий. В любом случае данные технологии будут рассмотрены в процессе этой работы, так как это отличный способ научиться их использовать, даже если это не даст значительного прироста.

Одной из основных задач этого проекта является не столько изучение методов применение конкретных моделей машинного обучения, сколько рассмотрение данной проблемы с разных сторон и использование различных подходов, для того, чтобы постараться получить максимальное качество модели. Дополнительно я хочу протестировать новые для меня фреймворки и написать полноценный продакшен код

Я считаю что опыт работы с данным проектом поможет мне лучше понять индустрию, а так же научиться применять современные технологии обработки, хранения и использования данных на реальном бизнес-кейсе, ввиду чего я смогу быть более подготовленным к реальной работе в сфере науки о данных в будущем.

А нужно ли нам машинное обучение?

Финансовые учреждения в настоящее время используют машинное обучение для более точной оценки кредитоспособности, а также для сокращения ручных процессов и ручных ошибок. Машинное обучение помогает кредиторам анализировать большие объемы данных быстрее и точнее, чем традиционные методы. Это позволяет кредиторам принимать более обоснованные решения о потенциальных заемщиках, помогая снизить риск и повысить процент одобренных кредитов.

Использование модели ИИ просто потому что это сейчас популярно и так делают все - фатальная ошибка. Перед применением модели нужно понять, зачем мы это делаем. Может ли мы позволить себе написать несколько конструкций else if? Возможно мы можем применить какую-нибудь другую эвристика при решение этой задачи?

Увы, но нет - мы не можем. Набор данных, с которым нам предстоит работать в данной задачи состоит, в лучшем случае, из нескольких десятков признаков для каждого пользователя, ввиду чего применение стандартных методов написания алгоритмов, основанных на логике, является неэффективным. Множество конструкций else if с таким же множеством условий будут в лучшем случае работать медленно, а в худшем вообще выдавать неверный ответ. Современные, уже реализованные за нас, алгоритмы машинного обучения позволяют куда легче отследить зависимости, так как фактически представляют нам интерфейс для доступа к множеству математических и логических операций, скрытых от глаз обычного пользователя. Не обязательно, хотя и во многих ситуациях нужно, использовать простые алгоритмы машинного обучения, ведь для решения

некоторых задач хватит той же самой логистической регрессии, что, как мы позже узнаем, все же не является нашим случаем.

Все этим факторы в разы упрощают разработку и использование моделей машинного обучения.

Ограничение

Ни для кого не является секретом, что на каждый каждый проект накладываются те или иные ограничения, связанные как с бизнесом, так и с технической составляющей.

Одним из главных ограничений в нашем случае является интерпретируемость модели машинного обучения, так как в данной сфере нежелательно пользоваться «черным ящиком». Нейтронные сети, которые неизвестном как работают, но дают нам значительный прирост в качестве на тесте - это отлично, однако «Центробанк» требует, чтобы логика, по которой модель делает свои предсказания была понятной.

Следовательно, набор моделей, которые мы можем использовать в нашей работает значительно сокращается и состоит из 5-6 вариантов.

Стек технологий

Python - основной язык программирование в сфере ИИ, используемый для обработки данных и использования моделей с помощью библиотек. Примечательно, что большинство библиотек используемых в данной области написаны на C++, так как этот язык позволяет ускорить выполнение операций в несколько раз.

Pandas и **PySpark**- библиотеки для языка программирования Python, позволяющие работать с табличными данными

Numpy - библиотека для языка программирования Python, позволяющая создавать, производить действия, а так же манипулировать векторами, матрицами и н-мерными массивами.

Scikit-learn - библиотека для языка программирования Python, предоставляющая доступ к большому количеству алгоритмов обработки данных и машинного обучения.

SeaBorn - библиотека визуализации на Python, основанная на matplotlib. Она предоставляет из себя высокоуровневый интерфейс для рисования привлекательных графиков.

PyTorch - библиотека для языка Python, позволяющая писать и использовать нейронные сети.

HuggingFace - библиотека, а так же и огромное комьюнити, содержащее заранее обученные модели нейросетей

PostgreSQL - реляционная база данных, позволяющая хранить наборы данных, и извлекать их с помощью языка SQL.

Git - система, используемая для версионирования кода. Все файлы будут храниться на GitHub соответсвенно

FastAPI - Python фреймворк для написания бекенд-составляющей нашего веб-сервиса. Будет использоваться для выкатки в минимальный продакшен.

Vk Cloud - облачный сервис, в котором я буду хранить данные в связке с **DVC** - **Data Version Control** - Система для версионирования данных. Гит для датасетов. Представляет из себя систему контроля версий данных, предоставляющую ту же функциональность, что и гит.

MlFlow Tracking - один из инструментов MlFlow, позволяющий легировать компоненты, версии кода, метрики и выходные файлы при запуске кода машинного обучения

*Дополнительный список библиотек, которые нам понадобятся во время разработки можно будет посмотреть в requirements.txt

Описание набора данных

ID - Идентификатор запроса по кредиту

Customer_ID - Идентификатор клиента

Month - Месяц запроса по кредиту

Name - Имя клиента

Age - Возраст клиента

SSN - Номер страхового полиса

Occupation - Проффесия

Annual_Income - Годовой заработок

Monthly_Inhand_Salary - Месячный заработок

Num_Bank_Accounts - Количество банковских аккаунт клиента

Num_Credit_Card - Количество кредитных кард, которыми клиент уже пользуется

Interest_Rate - Уровень заинтересованности в приобретение кредитной карты

Num_of_Loan - Количество предыдущих заемов

Type_of_Loan - Цель займа

Delay_from_due_date - Среднее количество дней, по которым были просрочены выплаты

Num_of_Delayed_Payment - Количество просроченных выплат

Changed_Credit_Limit - Процентное изменение лимитов по кредитной карте

Num_Credit_Inquiries - Количество запросов по кредитным картам

Credit_Mix - Средний кредитный рейтинг по предыдущим запросам

Outstanding_Debt - Представляет собой оставшуюся задолженность, подлежащую выплате

Credit_Utilization_Ratio - Коэффициент использования кредитной карты

Credit_History_Age - Количество лет с момента первого кредита

Payment_of_Min_Amount - Показывает, была ли выплачена лицом только минимальная сумма

Total_EMI_pre_month - Ежемесечные платежи EMI

Amount_invested_monthly - Представляет собой ежемесячную сумму, инвестированную клиентом

Payment_Behaviour - Платежное поведение клиента

Monthly_Balance - Средняя месячная сумма на балансе клиента

Credit_Score - Целевая переменная - кредитный рейтинг клиента

Этот набор данных взят с сайта **kaggle.com**:
https://www.kaggle.com/datasets/parisrohan/credit-score-classification?select=train.csv

Всего в датасете 27 признаков + целевая переменная. Нужно отметить, что в целевой переменной присутствует 3 класса, следовательно нашей задачей будет являться многоклассовая классификация клиентов банка