Предсказание ухода клиентов телекоммуникационной компании

Майкл Оши

1. Введение

Поставленной задачей была разработка модели для предсказывания ухода клиентов телекоммуникационной компании «Ниединогоразрыва.ком». Общая цель исследования – помогать компании лучше понимать отток клиентов чтобы в будущем предотвратить уход клиента с помощью специальных предложений и тому подобного.

Несмотря на эту общую цель, задача заключалась в том, чтобы предсказать, *уже ли ушёл клиент* (а не уйдёт ли). Такое решение в постановке задачи возможно связано с количеством доступных данных.

Для выполнения задачи было предусмотрено несколько ключевых шагов.

1. Обработка данных
2. Создание новых признаков
3. Сравнение разных моделей
4. Улучшение итоговой модели

1. Приведение данных в нужную модели форму. Главным образом это заключалось в том, чтобы изменить тип (dtype) данных.
2. Конструирование новых признаков с информацией из исходных признаков. Например, использовать данные о наличии услуг стриминг ТВ и стриминг фильмов чтобы создать признак о пакете услуг стриминг.
3. Тестирование разных моделей чтобы найти самую/ые обещающую/ие. Для этого шага модели применились с параметрами по умолчанию.
4. Улучшение лучшей/их модели/ей.

При исполнении проекта, все шаги были пройдены успешно и по порядку.

Важной частью проекта, не описанной в плане, являются проверка и выбор признаков. В этом шаге включались проверка корреляции признака с целевым признаком, проверка мультиколлинеарности признаков, а потом использование этой информации для принятия решения о том, какие признаки использовать. Это было сделано после создание новых признаков и до сравнения разных моделей. Таким образом, итоговый план работы получился таким:

1. Обработка данных
2. Создание новых признаков
3. Проверка и выбор признаков
4. Сравнение разных моделей
5. Улучшение итоговой модели

2. Данные

Данные были получены в четырёх разных .csv файлах, соответственно было создано четыре таблицы. Таблицы были удивительно цельными. Единственные найденные мной пропуски были в столбце с данными об общей сумме платежей (в таблице с данными о контрактах), а выяснилось, что все эти пропуски были у записей о новых контрактах. Иными словами, даже эти пропуски не были пропусками, а отсутствие платежей.

Нужно было переводить некоторые данные на новые типы. Были внесены следующие изменения:

- ‘BeginDate’ :
  - dtype: str
  - Было переведено на datetime а потом разбито на два отдельных признака с информацией о месяце и о годе заключения контракта.
- ‘EndDate’ :
  - dtype: str
  - Было переведено на булев тип (bool). В этом столбец было либо ‘No’, либо дата расторжения контракта, но для данного проекта важен только факт расторжения, а не дата.

Кроме этого, чтобы подготовить таблицу со всеми данными к использованию в модели, все категориальные признаки (т.е. все остальные признаки кроме ‘MonthlyCharges’ и ‘TotalCharges’), которые были типа str, были закодированными в числа. Например, в столбце с данными о типе контракта ‘Type’ – ’Month-to-month’, ‘One year’, ‘Two year’ стали 0, 1, 2.

3. Создание признаков

Были созданы следующие признаки:

- ′single_parent′ : родитель-одиночка
- ′single_mother′ : мать-одиночка
- ′single_father′ : отец-одиночка
- ′unnecessary_mail′ : этот признак показывает людей, которые платят электронно но всё равно получают квитанции по почте
- ′streaming_package′ : этот признак показывает, когда у человека есть обе услуги стриминг (и фильмы, и телевидение)
- ′begin_year′ : год заключения контракта
- ′begin_month′ : месяц заключения контракта (не включая информацию о годе заключения)

Создание этих признаков было частью исследовательского процесса, в итоге не все были использованы для обучения модели.

Чтобы решить какие признаки (оригинальные и новые) использовать, были проведены проверки корреляции и мультиколлинеарности. На основе этой информации, было решено использовать следующие признаки для обучения модели:

- ‘TotalCharges’
- ‘PaymentMethod’
- ‘InternetService’
- ‘MultipleLines’
- ‘Partner’
- ‘StreamingMovies’
- ‘StreamingTV’
- ‘gender’
- ‘OnlineSecurity’
- ‘OnlineBackup’
- ‘DeviceProtection’
- ‘Type’
- ‘TechSupport’
- ‘SeniorCitizen’
- ‘PaperlessBilling’
- ‘Dependents’
- ‘unnecessary_mail’
- ‘single_mother’
- ‘single_father’
- ‘begin_month’

Из этих признаков, самым спорным выбором является использование ‘TotalCharges’. Это спорный момент логически и по информации от проверки мультиколлинеарности, которая была высокая. Однако, мультиколлинеарность

не обязательно высказывается плохо на способность модели предсказывать, что является целью этого проекта. По поводу логического момента, этот признак действительно может дать слишком много информации модели; то есть, если эти данные сильно влияют на модель, она может оказаться хуже при предсказывании ухода новых клиентов (у которых общая сумма платежей не может быть высокой), а это значительный недостаток. С другой стороны, стоит думать о вопросе лояльности клиента к компании. Этот фактор может оказать огромное влияние на выбор клиента о возможном расторжении контракта, и лояльность тесно связана с длительностью контракта, которая, в свою очередь, связана с общей суммой платежей. Из-за того, что мультиколлинеарность не будет влиять на достижение цели проекта, и из-за вопроса лояльности клиента, было решено оставить 'TotalCharges' в наборе признаков для тестирования модели.

4. Модели

Тестирование моделей производилось со параметрами по умолчанию и проводилось несколько раз. Были протестированы следующие модели:

- LogisticRegression
- RidgeClassifier
- RandomForestClassifier
- ExtraTreesClassifier
- CatBoostClassifier
- LGBMClassifier
- XGBClassifier
- DummyClassifier

Была значительная разница между roc_auc_score результатами моделей, которые используют градиентный бустинг, и результатами моделей без бустнга; у группы с бустингом результаты были лучше.

	roc_auc_score
xgb_class	0.773131
lgb_class	0.770091
cat_boost	0.768463
rand_forest	0.738327
extra_trees	0.703266
log_reg	0.670505
ridge	0.646500
dummy_class	0.500000

Среди группы с бустнгом, у XGBClassifier и LGBMClassifier всегда был лучший результат. Однако, результаты всех трёх моделей – LGBMClassifier, XGBClassifier, и CatBoostClassifier – похожи; они чаще всего были около 0.77.

После тестирования с параметрами по умолчанию, несколько раз были тестированы лучшие модели (LGBM, XGB, CatBoost) с использованием Optuna для определения лучших параметров. Чаще всего получалось так, что у LGBMClassifier лучший результат на валидационной выборке, а у XGBClassifier лучший на тестовой выборке. Поэтому, ещё один раунд тестирования был проведён для этих моделей, опять используя Optuna но с бóльшим количеством итераций.

	valid_roc_auc	test_roc_auc
lgb	0.894629	0.898389
xgb	0.890198	0.914141
cat	0.884630	0.889190

После очередного теста, оказалось, что лучший результат на тестовой выборке получился у модели LGMBClassifier, поэтому эта модель была выбрана как финальной.

	valid_roc_auc	test_roc_auc
lgb	0.895654	0.909151
xgb	0.889087	0.908369

5. Результаты

Финальная модель: LGBMClassifier

Параметры:

n_estimators	65
max_depth	8
num_leaves	65
learning_rate	0.1435469272328213
reg_alpha	0.003208183290030983
reg_lambda	0.7390814771070586
min_child_samples	4

Результаты:

Выборка	roc_auc_score
валидационная	0.89565
тестовая	0.90915

Разные этапы проекта были важны для поиска лучшей модели. Качество финальной модели не зависло только от выбора начальной модели и параметров, но тоже зависло от выбора признаков.

Подробный список этапов следующий:

1. Обработка данных
2. Создание новых признаков
3. Проверка и выбор признаков
4. Сравнение разных моделей, используя все новые признаки
   a. Параметры: по умолчанию
   b. Лучшие: XGB, LGBM, CatBoost
5. Улучшение итоговой модели
   a. LGBM и XGB оба дали хорошие результаты, поэтому было использовано Optuna чтобы улучшить обе модели
   b. В конце концов, LGBM выдал лучший результат

Вышеописанным образом был достигнут лучший результат.

6. Выводы и возможности для улучшений

Удалось найти модель, у которой roc_auc_score выше 0.90.

Стоит отметить, что самые важные признаки для модели были 'TotalCharges' и 'begin_month'.

Как выше отмечалось, у 'TotalCharges' была высокая мультиколлинеарность, а также есть логическая связь между этим признаком и уходом/не уходом клиента. Для улучшения модели, было бы хорошо найти способ учесть лояльность клиента без использования 'TotalCharges'. Например, компания Мегафон предоставляет и мобильную связь, и домашний интернет (среди других услуг). Если клиент долго использовал мобильные услуги компании, есть возможность что эта лояльность выскажется на его лояльность к услугам домашнего интернета. Над вопросом оценки лояльности клиента нужно думать по больше, но это точно одно направление в поиске способов улучшить результат.

Высокая важность 'begin_month' интересна тем, что это намекает на некую сезонность в данных (признак держит данные исключительно о месяце заключения контракта, не о годе). Чтобы больше разбираться в этом, возможно стоило бы рассмотреть данные BeginDate как datetime объекты, чтобы

использовать возможности, например, библиотеки statsmodels.

Последнее, о чём стоит прокомментировать, это возникшие проблемы в проекте. Не было особых проблем как таковых. Данные были получены в хорошем состоянии, а трудные моменты на других этапах не возникали (во многом благодаря хорошему состоянию данных).