1. **Вступление. Цель и значимость проекта для бизнеса.**

Проект решает задачу прогнозирования оттока клиентов. Привлечение клиентов является непростой задачей. Компании, которые хотят удерживать устойчивые позиции на рынке среди конкурентов, готовы выделять значительный размер маркетингового бюджета на привлечение новых клиентов, повышение лояльности существующих клиентов и удержание клиентов, которые готовы прекратить сотрудничество с компанией. Уход клиентов к конкурентам значительно ослабляет позиции компании на рынке. Решение задачи прогнозирования оттока клиентов существенным образом помогает компании в рациональном распределении ресурсов и формировании оптимальной структуры бюджета.

2. **Техническая часть.**

- ***Входные данные.***

Входные данные представлены 4 таблицами со столбцами:

personal.csv - данные о клиентах,

customerID - идентификатор клиента,
gender - пол клиента,
SeniorCitizen - наличие пенсионного статуса по возрасту,
Partner - наличие супруга(и),
Dependents - наличие иждивенцев,

contract.csv - данные о договорах,

BeginDate - дата начала пользования услугами,
EndDate - дата окончания пользования услугами,
Type - тип оплаты: ежемесячный/годовой и тп,
PaperlessBilling - безналичный расчет,
PaymentMethod - способ оплаты,
MonthlyCharges - ежемесячные траты,
TotalCharges - всего потрачено денег на услуги,

internet.csv - данные о пользовании интернетом,

InternetService - тип подключения: DSL или оптоволокно,
OnlineSecurity - наличие блокировки небезопасных сайтов,
DeviceProtection - наличие антивируса,
OnlineBackup - наличие облачного хранилища для резервного копирования данных,
TechSupport - наличие выделенной линии технической поддержки,
StreamingTV - наличие стримингового телевидения,
StreamingMovies - наличие каталога фильмов,

phone.csv - данные о пользовании телефонией,

MultipleLines - возможность подключения телефонного аппарата к нескольким линиям одновременно

- ***Предобработка данных и анализ.***

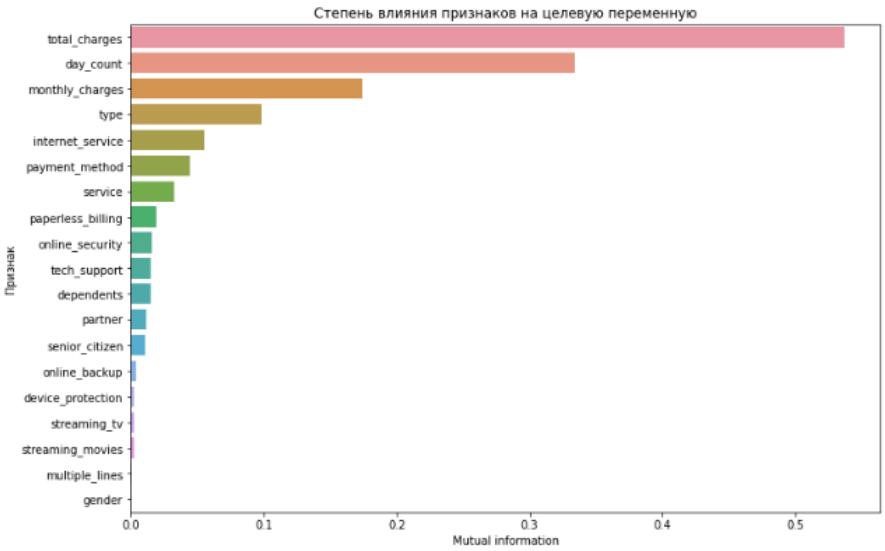
- Преобразование типа данных столбца TotalCharges из строкового в float64, пропуски по только что пришедшим клиентам заполнены значением из MonthlyCharges
- Данные в столбцах BeginDate и EndDate приведены к типу данных datetime64. Значения столбца EndDate по лояльным клиентам вместо ‘no’ заполнены датой выгрузки данных из базы (задается константой в разделе загрузки необходимых библиотек и модулей)
- В столбцах строкового типа данных устранены неявные дубликаты путем приведения данных к нижнему регистру с удалением лишних пробелов
- Данные в столбце SeniorCitizen заполнены ‘yes/no’ для унификация отображения бинарных признаков.
- Выделены новые признаки:
    - целевая переменная churn – признак ухода клиента (0/1) из данных столбца EndDate,
    - признак day_count – признак количества дней сотрудничества клиента (разница EndDate и BeginDate),
    - признак service – тип предоставляемых услуг (internet/phone/both),
    - service_count – количество предоставляемых опций на услугах интернета и телефонии
- Таблицы обьединены по столбцу customerID. Если клиент не пользуется услугой телефонии или интернета по соответствующим признакам из таблиц internet и phone проставляется значение ‘no’.
- Исследовательский анализ включает
    - Проверку данных на наличие аномалий: boxplot (количественных переменных) и value_counts() (категориальных переменных)
    - Проверку границ BeginDate и выгрузки данных по уходам EndDate
    - Анализ распределений количественных переменных TotalCharges, MonthlyCharges и day_count в разрезе данных churn
    - Проведена оценка влияния параметров категориальных признаков (по большей части предоставляемых опций на услугах интернета и телефонии) на увеличение/снижение вероятности ухода клиента
    - Составлен “портрет” склонного к оттоку клиента

- ***Построение модели и анализ.***

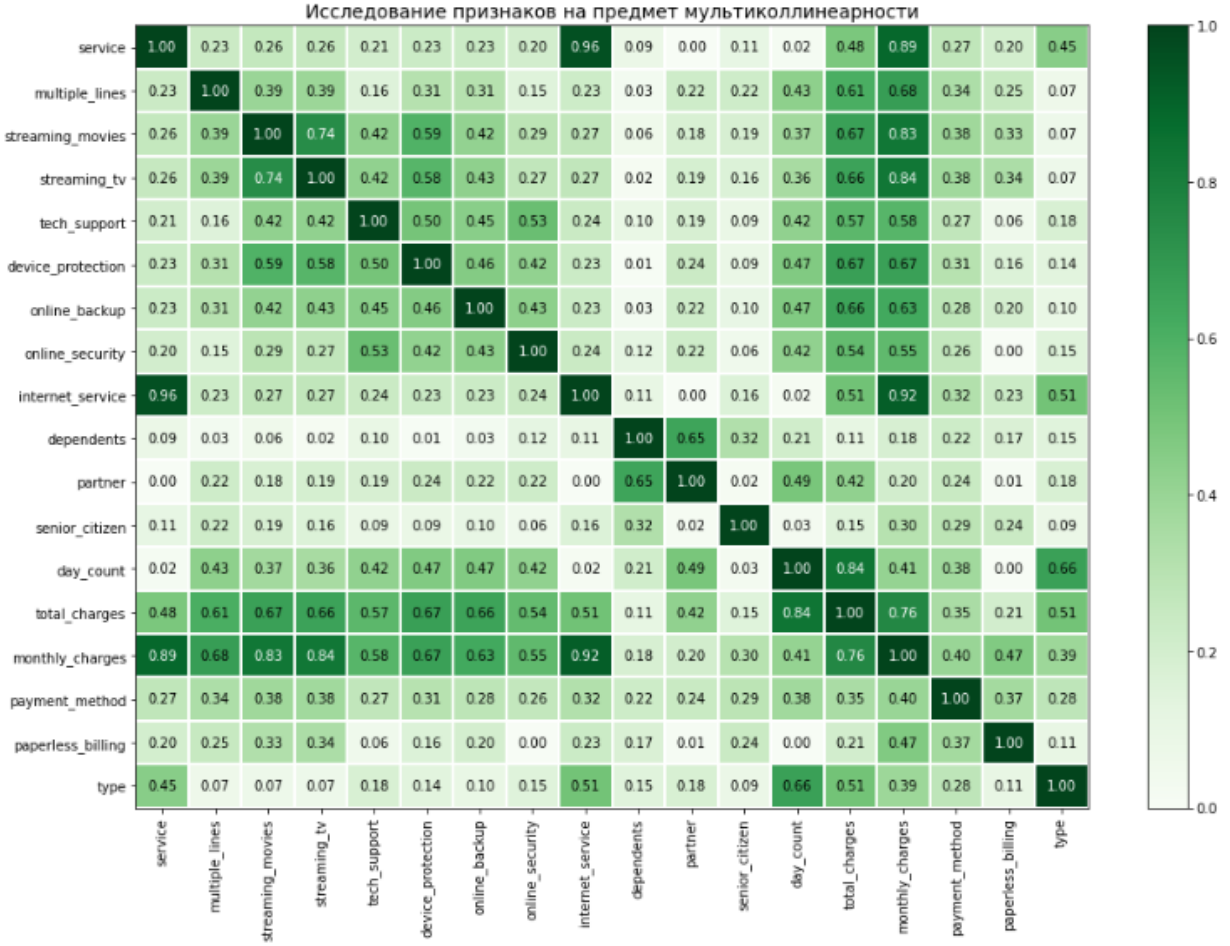
  Чтобы избежать утечки данных удалим признаки customerID, BeginDate, EndDate, сгенерированные для анализа признаки begin_month и end_month.

  Проверка на наличие полных дубликатов выявила наличие таковых. Удалили.

  Проведена оценка степени влияния каждого признака на целевую переменную по Mutual Information



Для обучения оставлены признаки 'type', 'paperless_billing', ‘gender,’ 'payment_method', 'monthly_charges', 'total _charges', 'day_count', 'senior_citizen', 'partner', 'dependents', 'internet_service', 'online_security', 'online_backup', ' device_protection', 'tech_support', 'streaming_tv', 'streaming_movies', 'multiple_lines', 'service'.
Анализ мультиколлинеарности показал значительную корреляцию между признаками 'monthly_charges', 'strea ming_tv', 'internet_service': затемненные области на матрице из библиотеки Phik.



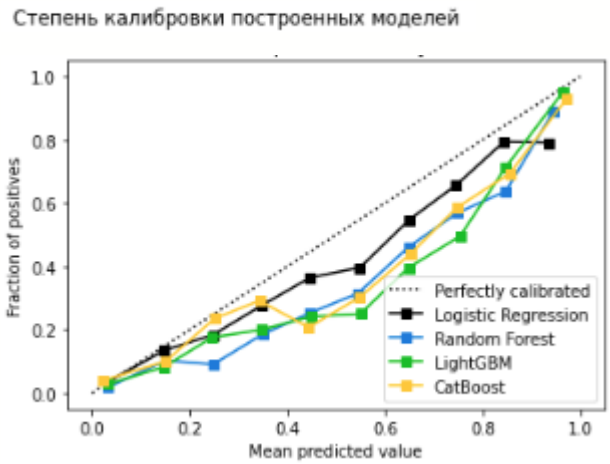
Произведено удаление коррелирующих признаков. Корреляция по оставшимся признакам не достигает 70%.
Произведен анализ на предмет сбалансированности целевого показателя. Наблюдается дисбаланс классов 0 к ласс - 73%, 1 класс – 27%. Произведен расчет классов методом compute_class_weight с целью дальнейшей пере дачи в модели при обучении.
Тк планируется составление пайплана со встроенным кроссвалидатором, то разбиение итоговых данных произ ведено на обучающую и тестовую выборку с долей тестовых данных 25%, со стратификацией по целевому пока зателю, перемешиванием shuffle=True и random_state=270323 для воспроизводимости результата.

В рамках проекта был построен ряд Pipeline с перебором параметров с помощью RandomizedSearchCV для мо делей LogisticRegression, RandomForestClassifier, LGBMClassifier, CatBoostClassifier.
Для линейных моделейбыло проведено масштабирование численных признаков методом StandardScaler, кодирование категориальных признаков методом OneHotEncoder(drop='first'). Борьба с дисбалансом производилась встраиванием метода SMOTE в пайплан, передачей параметра class_weights='balanced' в модель или непосредственно заранее рассчитанных долей классов методом compute_class_weight. Для бустинговых моделей производился подбор оптимального шага обучения. Для деревянной модели случайного леса был произведен подбор количества деревьев в лесу, глубины дерева, критерия gini или entropy. В RandomizedSearchCV передавались пайплайны с перечнем перебираемых параметров с разбиением обучающего датасета на 5 частей для кроссвалидации, ограничением итераций до 100. Максимизируемая метрика ROC-AUC.

В результате обучения получены следующие величины метрик по разным моделям (см. табл)

	Method	Roc_auc	Время обучения, сек
0	Pipeline+RSCV+LogReg	0.84	39
1	Pipeline+RSCV+RandForest	0.85	1.57
2	Pipeline+RSCV+LightGBM	0.89	11
3	Pipeline+RSCV+CatBoost	0.91	1.22

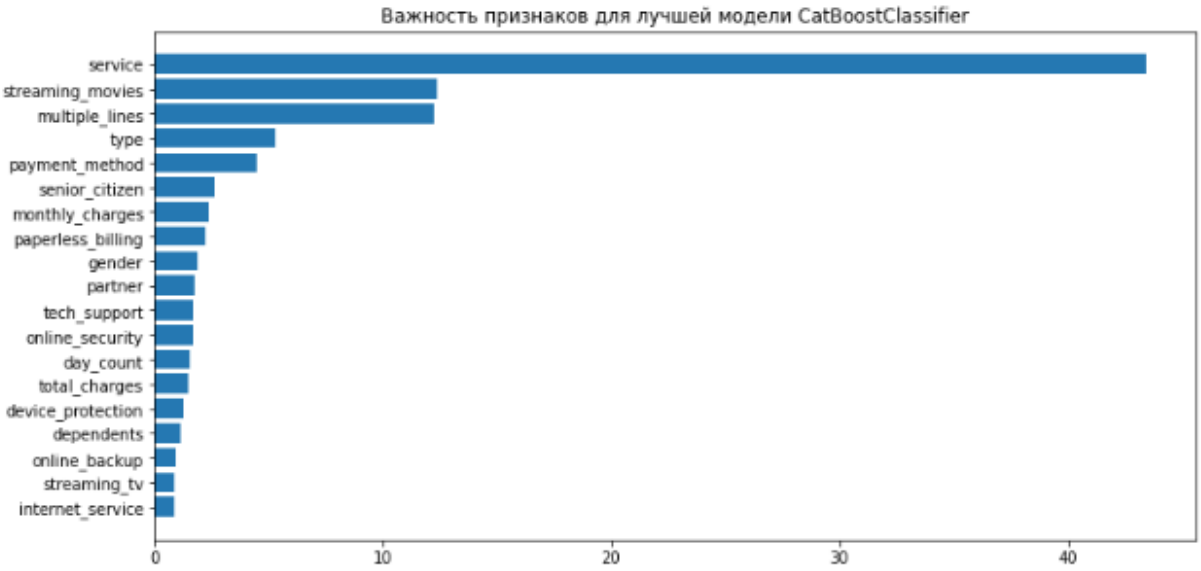
Проведено исследование надежности модели (устойчивости на тестовых данных) с помощью plot_calibration_curve().



По степени калибровки лучшей является модель Logistic Regression, затем идет CatBoost и Random Forest. Учитывая величину метрики, можно утверждать, что CatBoost является лучшей моделью.

Лучшей моделью оказалась бустинговая модель CatBoostClassifier с шагом обучения 0,1 и весами для классов {0: 0.68, 1: 1.88}, встроенная в pipeline с кодированием категориальных переменных методом OrdinalEncoder. Метрика ROC_AUC составила как на обучающей, так и на тестовой выборке 91%.

Рассмотрим, на какие признаки в основном опиралась модель при расчете прогноза вероятности ухода клиента.

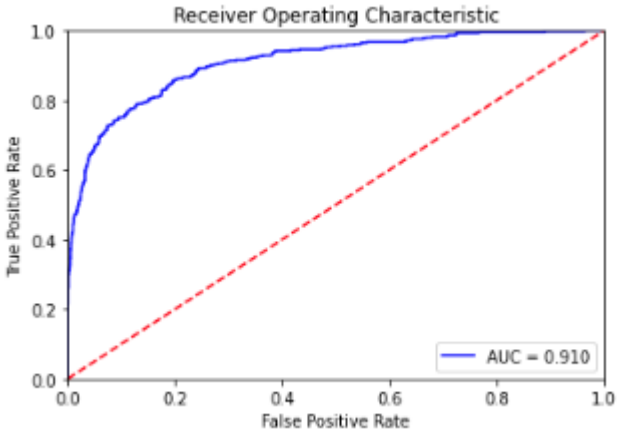


Наиболее важный признак для прогнозирования модели оказался service, затем в равной степени streaming_movies и multiple_lines (который ранее был практически незначимый по матрице Phik), в еще меньшей - type и payment_method и далее все остальные.

- ***Тестирование.***

3. Анализ результатов работы модели. Бизнес-анализ альтернативных издержек. Рекомендации руководству компании.

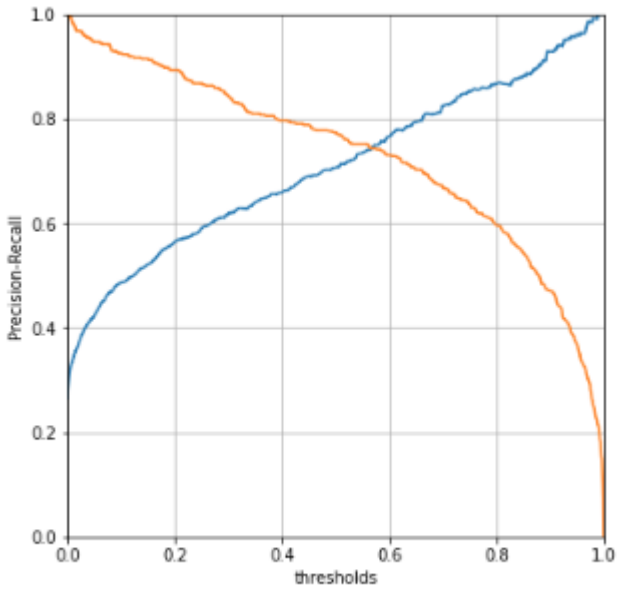
Для анализа соотношения True Positive Rate и False Positive Rate была построена ROC-кривая



Произведен расчет precision, recall, F-меры и ROC-AUC на разных порогах классификации.

```
Порог: 0.28 | Точность: 0.608, Полнота: 0.859, F1: 0.712, ROC-Auc: 0.829
Порог: 0.30 | Точность: 0.618, Полнота: 0.842, F1: 0.713, ROC-Auc: 0.827
Порог: 0.32 | Точность: 0.629, Полнота: 0.829, F1: 0.715, ROC-Auc: 0.826
Порог: 0.34 | Точность: 0.634, Полнота: 0.812, F1: 0.712, ROC-Auc: 0.821
Порог: 0.36 | Точность: 0.647, Полнота: 0.807, F1: 0.718, ROC-Auc: 0.824
Порог: 0.38 | Точность: 0.656, Полнота: 0.805, F1: 0.723, ROC-Auc: 0.826
Порог: 0.40 | Точность: 0.660, Полнота: 0.797, F1: 0.722, ROC-Auc: 0.824
Порог: 0.42 | Точность: 0.669, Полнота: 0.792, F1: 0.725, ROC-Auc: 0.825
Порог: 0.44 | Точность: 0.685, Полнота: 0.788, F1: 0.733, ROC-Auc: 0.829
Порог: 0.46 | Точность: 0.692, Полнота: 0.779, F1: 0.733, ROC-Auc: 0.827
Порог: 0.48 | Точность: 0.702, Полнота: 0.777, F1: 0.738, ROC-Auc: 0.829
Порог: 0.50 | Точность: 0.706, Полнота: 0.773, F1: 0.738, ROC-Auc: 0.829
Порог: 0.52 | Точность: 0.717, Полнота: 0.760, F1: 0.738, ROC-Auc: 0.826
Порог: 0.54 | Точность: 0.733, Полнота: 0.752, F1: 0.742, ROC-Auc: 0.826
Порог: 0.56 | Точность: 0.740, Полнота: 0.749, F1: 0.745, ROC-Auc: 0.827
Порог: 0.58 | Точность: 0.751, Полнота: 0.741, F1: 0.746, ROC-Auc: 0.826
Порог: 0.60 | Точность: 0.766, Полнота: 0.730, F1: 0.748, ROC-Auc: 0.825
Порог: 0.62 | Точность: 0.779, Полнота: 0.724, F1: 0.750, ROC-Auc: 0.825
Порог: 0.64 | Точность: 0.786, Полнота: 0.709, F1: 0.745, ROC-Auc: 0.820
Порог: 0.66 | Точность: 0.796, Полнота: 0.700, F1: 0.745, ROC-Auc: 0.818
```

Построена кривая Precision-Recall с расчетом наилучшего порога классификации.




```
Best threshold = 0.568
```

При пороге 0,568 наблюдаем оптимальное соотношение параметров: высокая метрика precision (0.74) при сохранении на должном уровне recall (0.749) и остальных метрик: на лчень хорошем уровне F1 = 74.5%, ROC-Auc = 82.7%. На этом пороге модель максимизирует долю истинных положительных и долю истинных отрицательных результатов, минимизируя вероятность ложной тревоги - ошибки первого рода. На этом пороге показатели точности и полноты 0.610 и 0.627 соответственно. На дальнейших порогах точность продолжает расти, остальные метрики начинают постепенно падать.

С другой стороны в данной задаче определения вероятности ухода клиента более важно максимизировать метрику precision и ложно отправить скидку или письмо клиенту, который на самом деле не хотел уходить из компании, чтобы оставить его. Чем не предоставить скидку и потерять клиента. Здесь нужно смотреть выручку, которую приносит клиент (определить категорию клиента по типу тарифа), возможный размер предоставляемой скидки и пр. Эти вопросы нужно рассматривать исходя из требований заказчика.

Далее произведен расчет вероятности ухода для каждого клиента из тестового набора данных и выведен список клиентов, отнесенных моделью к 1 классу.

```
probabilities_test
```

```
array([[0.93349766, 0.06650234],
       [0.84105922, 0.15894078],
       [0.99756652, 0.00243348],
       ...,
       [0.06210931, 0.93789069],
       [0.41805488, 0.58194512],
       [0.41627428, 0.58372572]])
```

На основании величины вероятности ухода клиента, можно сгруппировать «отточных» клиентов по величине вероятности и каждой группе предложить различные системы скидок. На основе этих данных можно провести оптимизацию маркетинговой политики, предлагаемых клиенту продуктов и условий сотрудничества.

Имея выгрузку данных по клиентам, которые склонны к оттоку, можно уже рассчитать альтернативные издержки на предоставление скидки и упущенную выгоду от ухода клиента с проведением соответствующего анализа и составлением прогнозов для компании в обоих случаях. Таким образом, можно добиться сокращения неоптимальных расходов компании и целенаправленной точечной работой с отдельными группами клиентов.

4. Дальнейшие пути развития проекта.

- Хотела, но не реализовала пока: выбор лучшей модели осуществлять не только на основании величины метрики и калибровки модели, но также исходя из принципов минимизации разброса возможных вариантов метрики вокруг среднего генеральной совокупности (мы же не знаем, какие тестовые данные будут предоставлены), для этого нужно определить доверительные интервалы для метрик всех построенных моделей.

- Рассчитать альтернативные издержки бизнеса с учетом предлагаемых скидок для удержания клиента. Составить прогноз выручки и чистой прибыли компании в обоих случаях.

- Провести развертывание построенной модели машинного обучения в рабочей среде в качестве API с помощью Flask.