Проект по курсу от Megafon

Выполнил: Кузнецов Виктор Владимирович

Данный проект выполнен на основании датасета представленного Мегафон.

Для приведении датасета к нормальному виду был применена библиотека Dask с сохранением в формате parquet с приведением типов данных.

[Apache Parquet](https://parquet.apache.org/documentation/latest/) - это бесплатный и ориентированный на столбцы формат хранения данных с открытым исходным кодом экосистемы Apache Hadoop. Он похож на другие форматы файлов с столбчатым хранилищем, доступные в Hadoop, а именно RCFile и ORC. Он совместим с большинством фреймворков обработки данных в среде Hadoop. Он обеспечивает эффективные схемы сжатия и кодирования данных с повышенной производительностью для обработки сложных объемных данных.

Apache Parquet реализован с использованием алгоритма измельчения и сборки записей, который вмещает сложные структуры данных, которые могут быть использованы для хранения данных. Значения в каждом столбце физически хранятся в смежных ячейках памяти, и это столбчатое хранилище обеспечивает следующие преимущества:

Сжатие по столбцам эффективно и экономит место для хранения

Методы сжатия, специфичные для конкретного типа, могут быть применены, поскольку значения столбцов, как правило, имеют один и тот же тип

Запросы, извлекающие определенные значения столбцов, не должны считывать все данные строк, что повышает производительность

Различные методы кодирования могут быть применены к различным столбцам

Apache Parquet реализован с использованием фреймворка Apache Thrift, что повышает его гибкость; он может работать с рядом языков программирования, таких как C++, Java, Python, PHP и т.д.

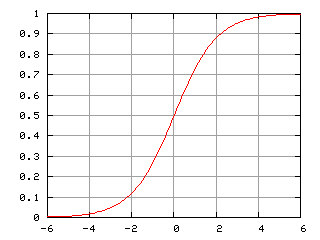
По состоянию на август 2015 года Parquet поддерживает фреймворки обработки больших данных, включая Apache Hive, Apache Drill, Apache Impala, Apache Crunch, Apache Pig, Cascading, Presto и Apache Spark.

[Wiki](https://en.wikipedia.org/wiki/Apache_Parquet)

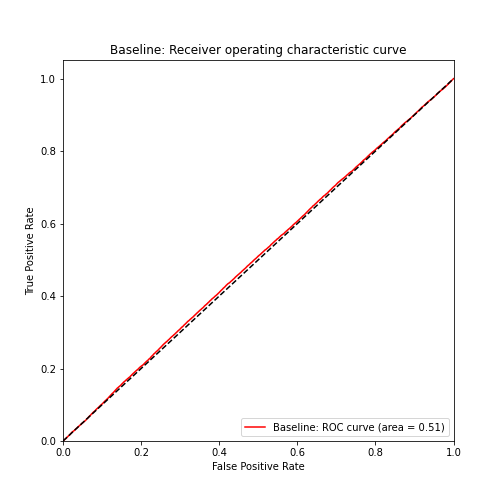
В данных состоят из:

1. Констант - 7
2. Бинарных значений - 23
3. Категорий - 63
4. Численных значений - 171

Для базовой модели взята Логическая регрессия.



Логистическая регрессия или логит-модель (англ. logit model) — это статистическая модель, используемая для прогнозирования вероятности возникновения некоторого события путём его сравнения с логистической кривой. Эта регреcсия выдаёт ответ в виде вероятности бинарного события (1 или 0).



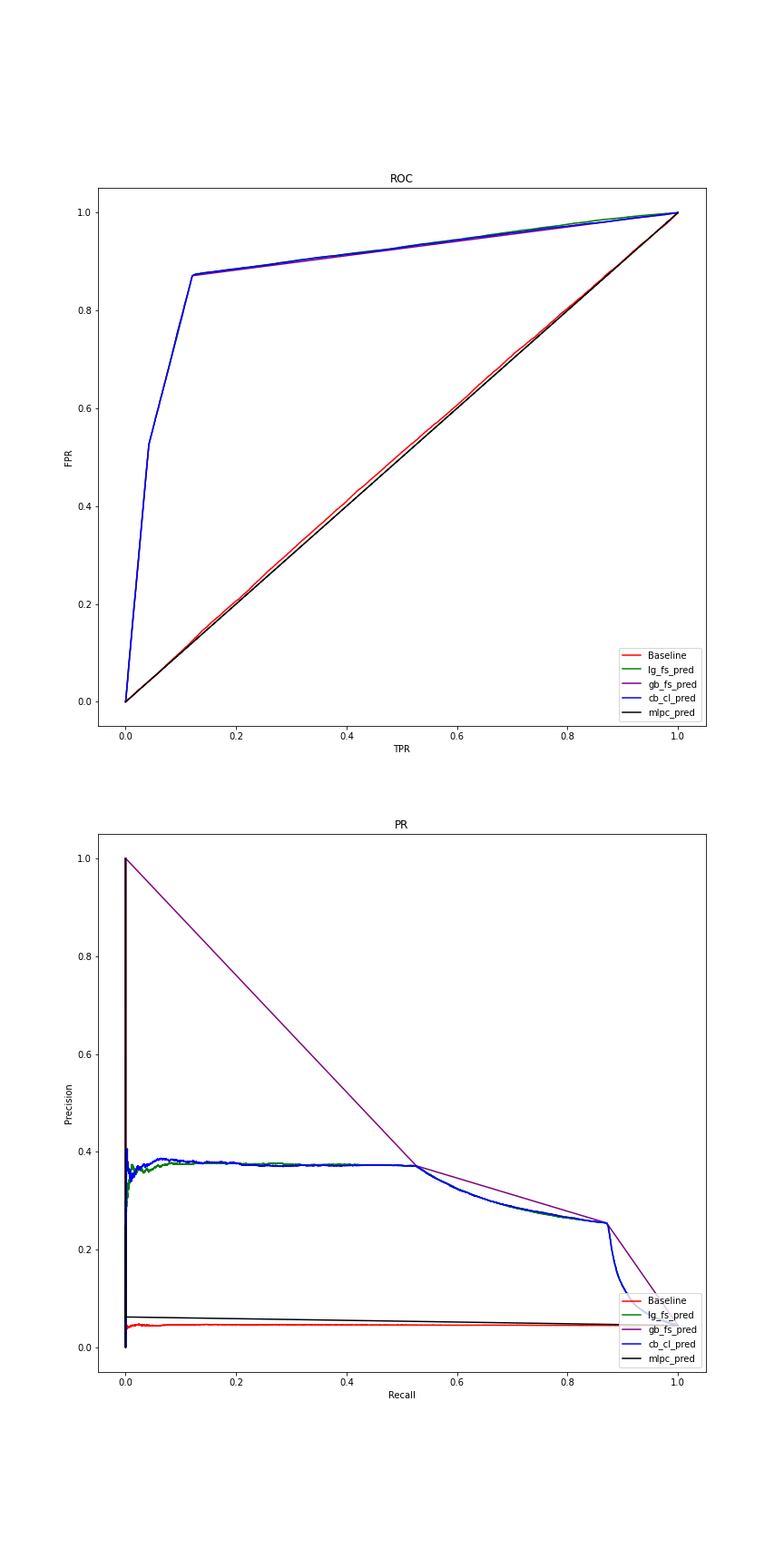
Для нахождения лучшей модели были проведены сравнения со следующими моделями:

LogisticRegression

GradientBoostingClassifier

MLPClassifier

CatBoostClassifier



На матрице решений были получены следующие значения для F1:

Базовая модель

macro avg precision = 0.49, AUC\_PR = 0.046, AUC\_ROC = 0.506

Confusion matrix

[[443978 60673]

[ 20911 2960]]

LogisticRegression

macro avg precision = 0.70, AUC\_PR = 0.310, AUC\_ROC = 0.892

Confusion matrix

[[483865 20786]

[ 11590 12281]]

GradientBoostingClassifier

macro avg precision = 0.49, AUC\_PR = 0.488, AUC\_ROC = 0.889

Confusion matrix

[[504651 0]

[ 23871 0]]

CatBoostClassifier

macro avg precision = 0.70, AUC\_PR = 0.311, AUC\_ROC = 0.890

Confusion matrix

[[484238 20413]

[ 11796 12075]]

MLPClassifier

macro avg precision = 0.49, AUC\_PR = 0.054, AUC\_ROC = 0.500

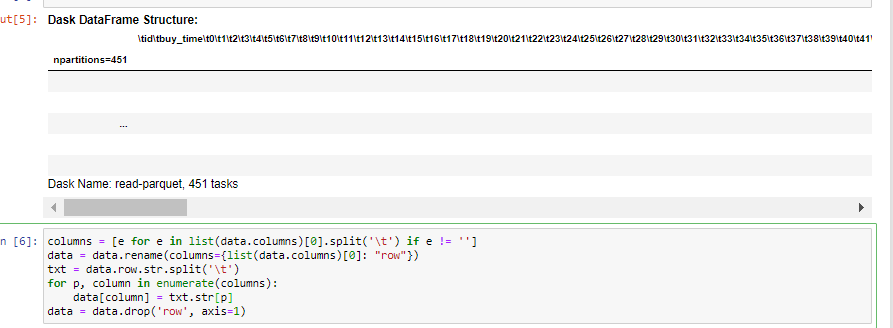
Confusion matrix

[[504643 8]

[ 23870 1]]

Была выбран модель для предсказания CatBoostClassifier в связи с тем что он давал точность на целевой метрике F1 70% и AUC ROC 89%

Принцип подхода заключается в следующем:

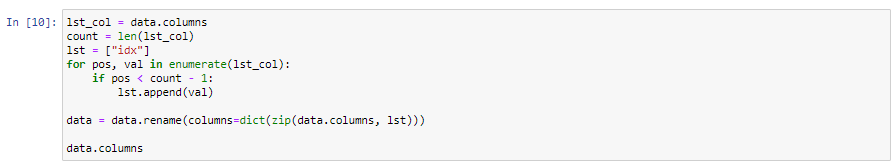
1. Преобразование в формат parquet. Парсил строку из csv файла. 

Объем данных 2.95 Гб

1. Преобразовал строки в численный тип данных. 

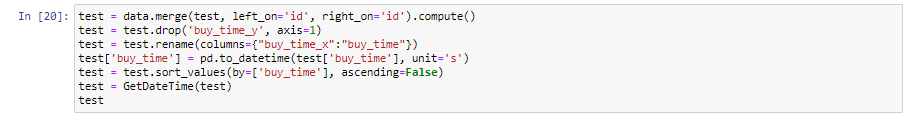
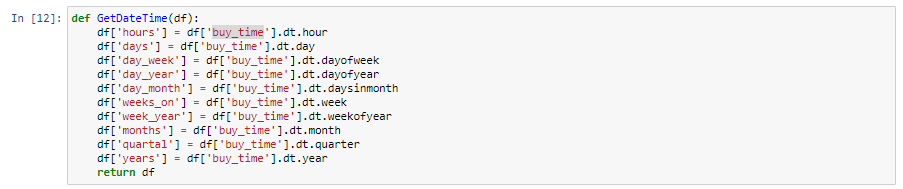
Объем данных 1.58 Гб

1. После обнаружения смещения в данных. Сместил колонки на одну вправо и удалил лишную.

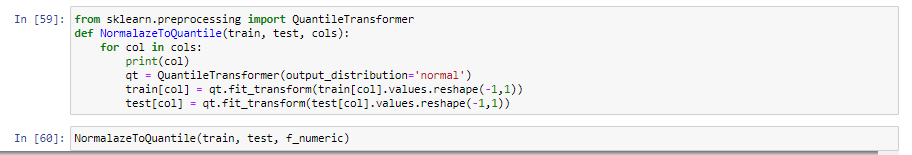




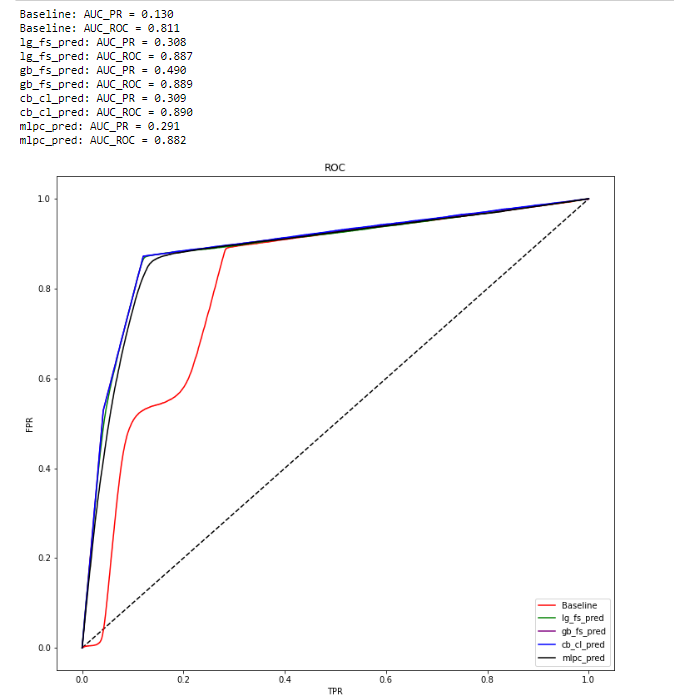
1. Объединил датасет и данные с фичами по полю id
2. Добавил данные по полю buy\_time все возможные типы данных

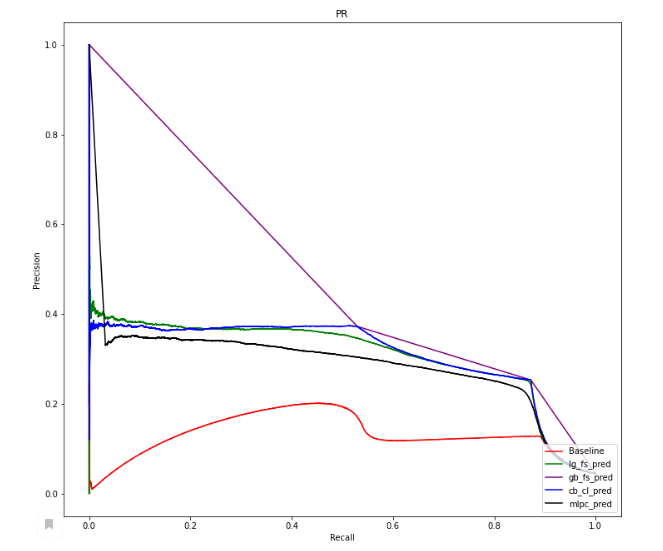


1. Разделил тенировочный датасет на тестовый и тренировочный
2. Разделил колонки на константы, бинарные, категориальные, численные и временные типы данных.
3. Выполнил нормализацию и приведение к стандартному распределению численных данных.



1. После анализа классификаторов был выбран CatBoostClassifier





1. После снижения парогра чувствительности до 30% выбранная модель показала 70% точность

