Аналитическая записка

# Исследование отделений Банка

Программист: Бакшеев Сергей

Дата: 30.01.2020

Оглавление

[Исследование отделений Банка 1](#_Toc65417933)

[Аннотация 2](#_Toc65417934)

[Описание данных 2](#_Toc65417935)

[Используемые программные библиотеки 3](#_Toc65417936)

[Этап 1. Обработка данных 3](#_Toc65417937)

[Этап 2. Кластеризация. 5](#_Toc65417938)

[Этап 3. Интерпретация полученных классов. 7](#_Toc65417939)

[Этап 4. Поиск аномалий 7](#_Toc65417940)

[Этап 5. Формулировка рекомендаций 9](#_Toc65417941)

## Аннотация

Цель данной работы – проанализировать данные по отделениям банка, сформировать рекомендации для повышения эффективности управления. В рамках задачи была произведена предобработка данных, кластеризация с помощью методов Kmeans, DBSCAN, вычисление аномалий методом OneClassSVM. В конце работы составлены рекомендации по дальнейшему применению реализованных программных методов.

## Описание данных

- post\_id -- Почтовый индекс отделения от заказчика

- open\_date -- Дата открытия отделения

- street -- Улица, выделенная из response

- address -- Адрес целиком, выделенный из response

- place\_type -- Тип точки

- shop\_centre -- Флаг нахождения в торговом центре

- dns -- Флаг нахождения в магазине DNS

- lat -- широта

- lon -- долгота

- response -- необработанный ответ геокодера, содержащий адрес

- district -- Округ, выделенный из response

- post2 -- Почтовый индекс, выделенный из response

- is\_center -- Находится ли в центре Москвы

- price -- цена квадратного метра покупки жилья в районе Москвы/по городу вне Москвы

- parking\_count -- количество парковок в ближайшей окрестности

- parking\_min\_distance -- минимальное расстояние до парковки (если есть рядом)

- finance\_count -- количество финансовых учреждений в ближайшей окрестности

- shop\_count -- количество магазинов в ближайшей окрестности

- distance\_100 -- интенсивность человекопотока в окрестности 100 метров по gps трекам

- distance\_500 -- интенсивность человекопотока в окрестности 500 метров по gps трекам

- cafe\_count -- количество кафе в ближайшей окрестности

- culture\_count -- количество культурных объектов в ближайшей окрестности

- job\_gov\_count -- количество мест работы и государственных учреждений в ближайшей окрестности

- metro\_count -- количество станций метро в ближайшей окрестности

- metro\_min\_distance -- расстояние до ближайшей станции метро (если есть рядом)

- transport\_count -- количество остановок в ближайшей окрестности

- transport\_min\_distance -- расстояние до ближайшей остановки (если есть рядом)

- working\_days -- количество рабочих дней в неделе

- work\_hours -- максимальная продолжительность рабочего дня в течение недели

- weekday\_availability -- количество нерабочих дней на неделе

- office\_hours -- строка с сырым временем работы отделения

- is\_moscow -- флаг того, что город Москва

## Используемые программные библиотеки

numpy 1.19.5

pandas 0.23.4

matplotlib 3.0.2

seaborn 0.9.0

scipy 1.5.2

sklearn 0.20.2

## Этап 1. Обработка данных

Были получены данные о 717 отделениях. Больше половины численные (int64, float64), а так же есть бинарные (bool) и строковые (object). Так же в данных есть пропуски:

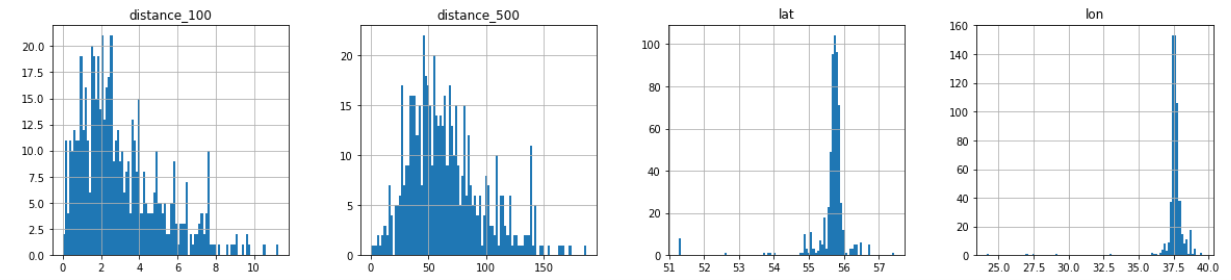
* post\_id - почтовый индекс
* open\_date - дата открытия отделения
* price - цена покупки жилья (квадратный метр)
* parking\_min\_distance - минимальное расстояние до парковки
* metro\_min\_distance - минимальное расстояние до станции метро
* transport\_min\_distance - минимальное расстояние до ближайшей остановки
* office\_hours - "сырое" время работы отделения

В рамках обработки пустых значений было решено отказаться от «post\_id», так как он необходим только для идентификации отделения. Строки с пропусками в полях «open\_date», «price» было решено удалить из таблицы, так как их мало и значения полей важны для последующего анализа. Пропуски «parking\_min\_distance», «metro\_min\_distance», «transport\_min\_distance» было решено заменить большим значением дистанции (добавили 5 км). Пропуски «office\_hours» были заменены на среднее значение по столбцу.

В рамках приведения важных категориальных данных к числовому виду были выполнены следующие действия:

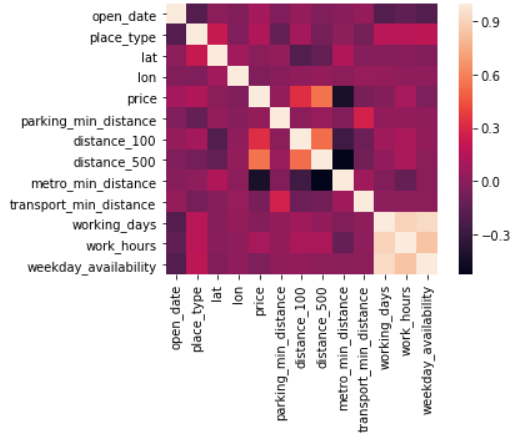
* Значения True и False в столбцах «shop\_centre», «dns», «is\_center», «is\_moscow» были заменены на численные аналоги: 1 и 0 соответственно.
* Для столбца «place\_type» все уникальные категории ('other' , 'ТП', 'ОПС Б1', 'ОПС Б1\_2' , 'КЦ') были заменены на разные числа
* Для столбца «office\_hours» был реализован алгоритм подсчёта общего количества рабочих часов (в минутах). Так как данные в столбце «сырые», подсчёт ограничивался стандартным шаблоном написания времени. В случае ошибки в шаблоне время выставлялось средним значением.

По полученным данным были построены гистаграммы для поиска характера распределений и возможных вбросов:



Видно, что на графике lat есть значения 51,2, которые могут плохо повлиять на алгоритм.

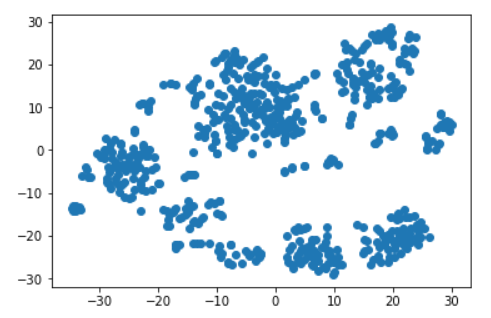
Так же была выявлена сильная корреляция между столбцами "количество рабочих дней в неделе", "максимальная продолжительность рабочего дня", "количество нерабочих дней на неделе":



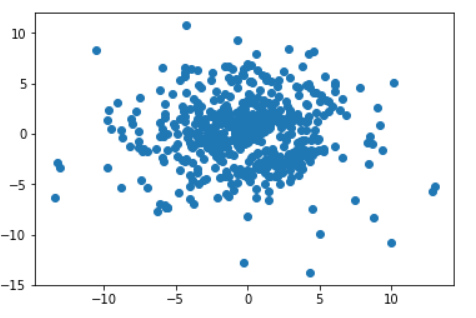
## Этап 2. Кластеризация.

Были построены графики данных с помощью методов понижения размерности для отмасштабированных целочисленных и вещественных признаков.

TSNE:



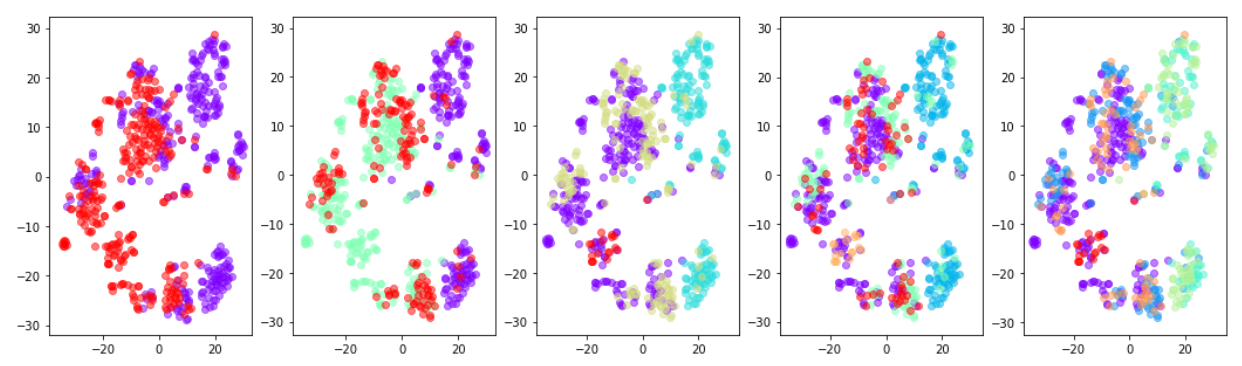
MDS:



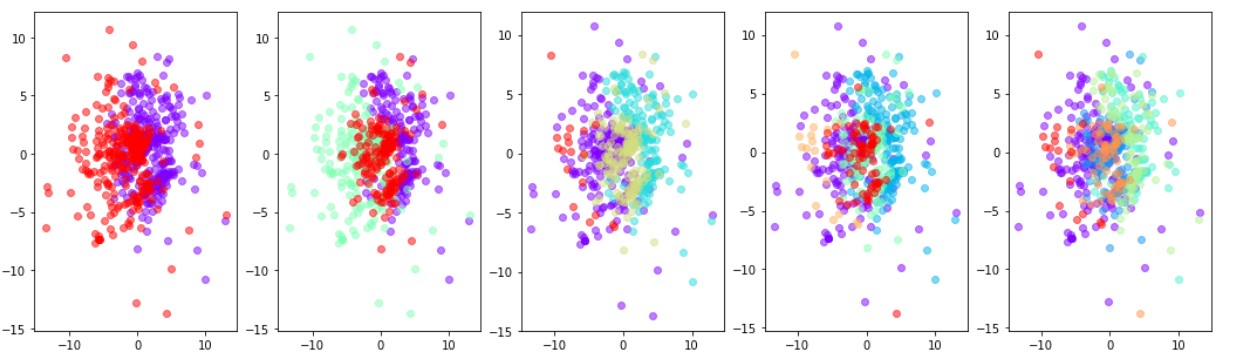
Наиболее наглядным оказался метод TSNE. Только для вещественных признаков он показал три кластера, а вместе с целочисленными можно увидеть 6 классов.

Далее была построена кластеризация методом KMeans для количества кластеров от двух до шести:

На данных TSNE:

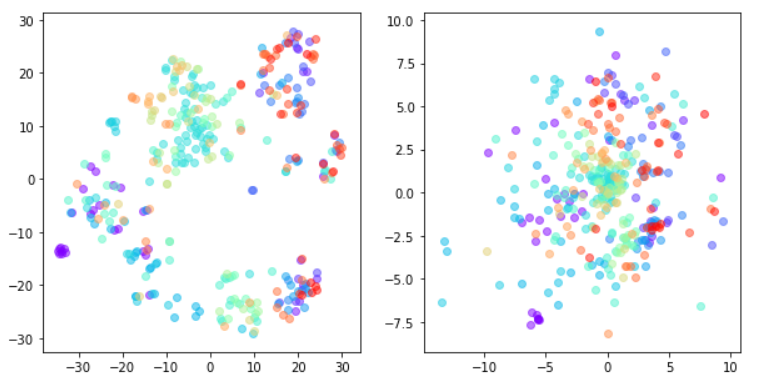


На данных MDS:



Сделан вывод о хорошей интерпретируемости при разделении на 2 или 3 кластера данным методом, так как облака точек классов не сильно перекрывают друг друга.

Так как кластеров нужно получить не очень много, было решено попробовать также метод кластеризации DBSCAN:



Но данный метод оказалось сложно интерпретировать в рамках реализованных методов понижения размерности.

## Этап 3. Интерпретация полученных классов.

По данным разбиения K-means на 3 кластера были выведены и проанализированы первые 10 элементов каждого класса.

**Первый класс.** Отделения, которые находятся далеко от метро и имеют вокруг достаточно среднее число культурных и финансовых организаций. Цена аренды таких помещений на квадратный метр составляет примерно 80 000.

**Второй класс.** Отделения рядом с одной или двумя станциями метро. Есть много финансовых и рабочих организаций, что похоже на бизнес-центры. Аренда таких помещений примерно 220 000.

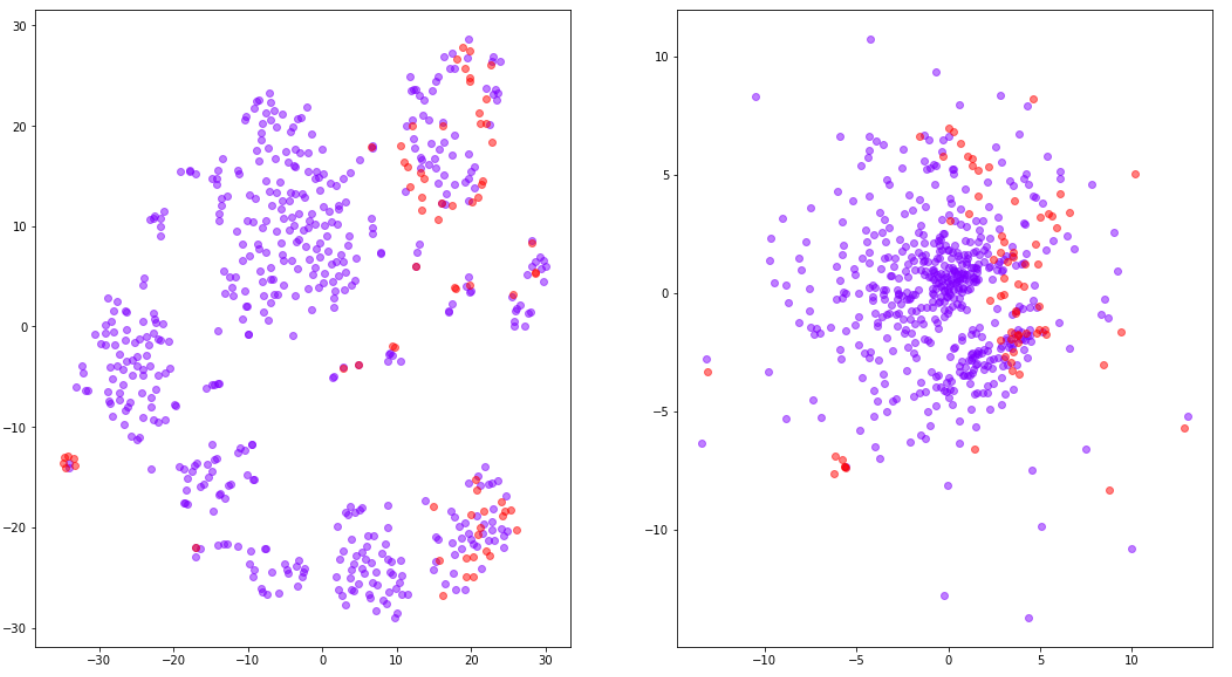
**Третий класс.** Отделения, где слабо развита инфраструктура, но все равно есть плотность населения. Рядом может быть станция метро и всегда есть парковка. Похоже на спальные районы. Аренда помещений примерно 140 000.

## Этап 4. Поиск аномалий

В данной работе использовался метод OneClassSVM с параметрами (kernel=’rbf’, gamma=100, nu=0.0008). Данные параметры были выбраны для лучшей эффективности и интерпретируемости полученных аномалий.

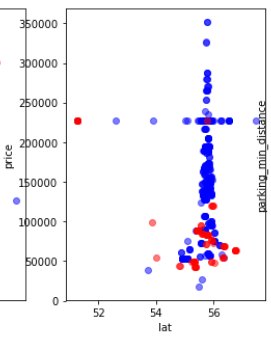
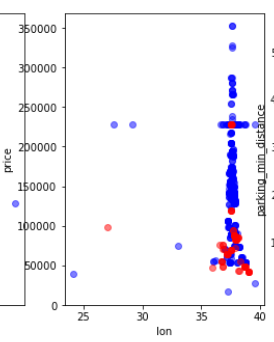
Получившийся метод в среднем считает аномалиями 11.8% объектов. Это хороший результат, так как данных относительно мало и есть время для последующей более детальной проверки полученных объектов.

При отображении на графиках пониженной размерности (TSNE, MDS) получаем:

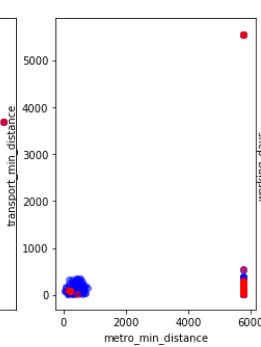


Видно, что большинство красных точек (аномалий) находится на краях или вдалеке от кластеров.

Также были проанализированы пересечения признаков отделений на предмет аномалий. Выявлено, что большинство аномалий связано с низкой арендой помещений на определенной долготе и широте (вероятней всего в Москве).

Также алгоритм считает аномалией близость транспорта при отсутствии метро:



## Этап 5. Формулировка рекомендаций

Основная польза данного программного решения - своевременный поиск аномалий в работе отделений. Это позволит в реальном времени определять:

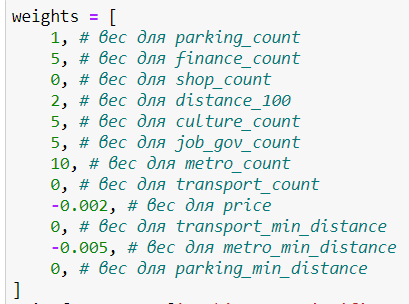
* где людей стало меньше, а цена аренды жилья не изменилась
* где, наоборот, появилось метро, и увеличилась плотность населения
* отделения, где среднее рабочее время сильно упало/возросло
* и т. д.

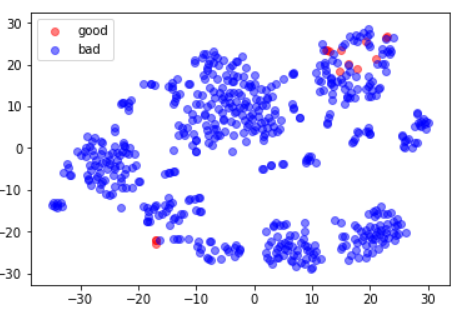
Разбиение на кластеры позволяет определять для каждой выборки отделений свои правила работы. Ориентируясь на полученные данные можно, например, составлять более эффективную рекламу для клиентов (для бизнес центров предлагать кредиты и инвестиции, для спальных районов - вклады).

В данных нет однозначной оценки качества отделения (прибыли). Желательно добавить её среднее ежегодное значение для более качественного оценивания. Тогда можно было определять, относится ли новое место для открытия к максимально прибыльному кластеру или нет.

Можно также попробовать выставить оценку по требуемым гиперпараметрам (весам признаков, которые укажет руководство). Например, result = w1 \* parking\_count + w2 \* distance\_100 - w3 \* price - w4 \* metro\_min\_distance и т. д.

Например:





Так мы можем определять, в каком направлении кластеров стоит развиваться.

Так же можно искать самые "плохие" отделения:

