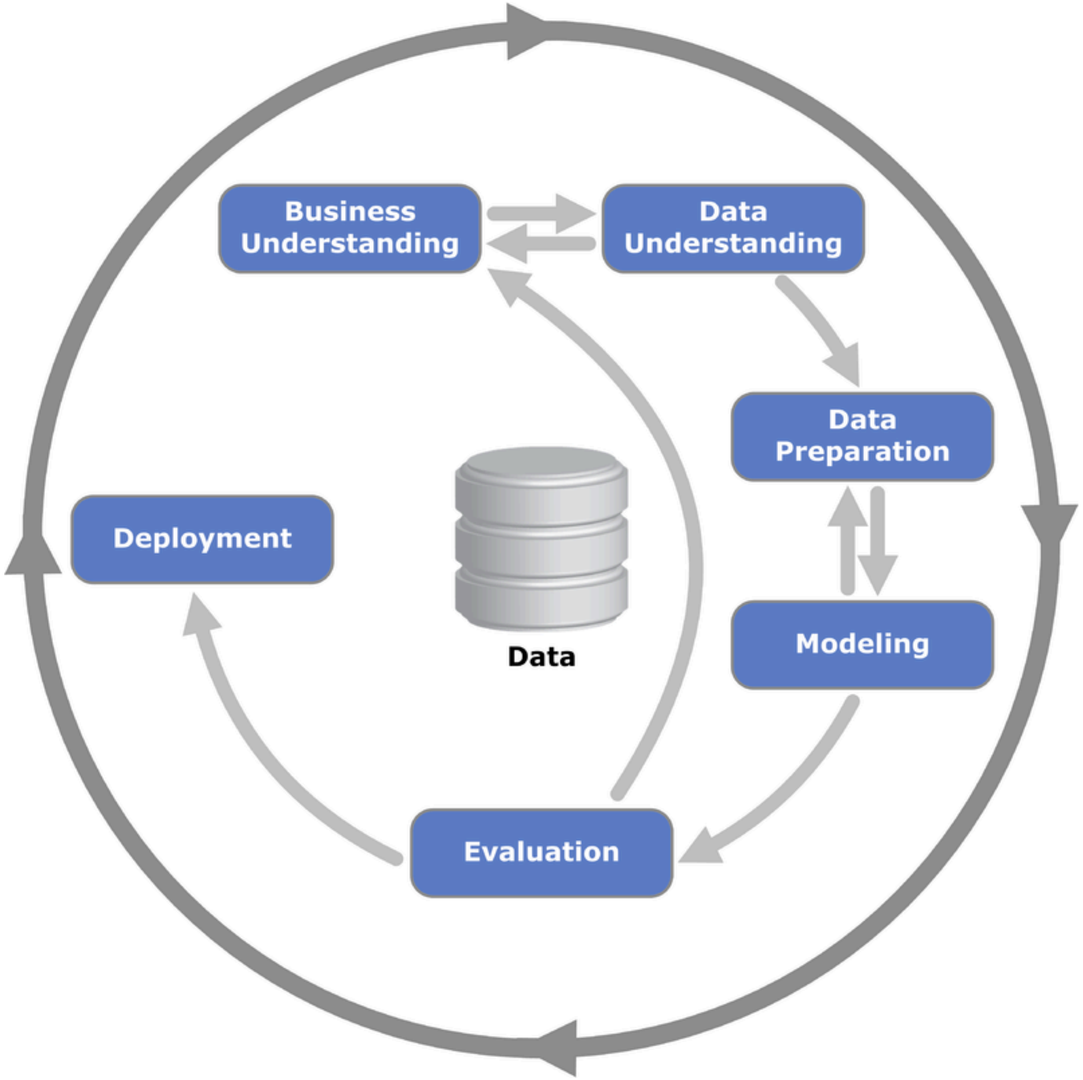
Прогнозирование продаж магазинов Rossmann с использованием методологии CRISP

Бизнес-анализ. Описание задачи

Предсказать продажи магазинов Rossmann в период с 01.08.2015 по 17.09.2015, основываясь на данных, полученных с 01.01.2013 по 31.07.2015, используя подходы, заложенные в CRISP

Описание задачи

Цель: провести как можно больше циклов CRISP, скор вторичен



Анализ данных. Описание данных

Сбор данных не требуется, так как датасет был предоставлен на соревновании. Сами данные представляют собой 3 таблицы:

- **Train - информация по продажам каждого магазина за период с 01.01.2013 по 31.07.2015. Размер - 1,017,209 x 8**
- **Store - информация о каждом из магазинов. Размер - 1115 x10**
- **Test - аналогично Train, с отсутствием колонок Sales и Customers, а также период - 01.08.2015 по 17.09.2015. Размер - 41088 x 8**

Описание данных. Train

- **Sales: продажи в конкретный день (Target)**
- **Customers: число клиентов в этот день**
- **Open: индикатор - открыт магазин или нет**
- **Promo: индикатор - действует ли промо в магазине в этот день или нет**
- **StateHoliday: указывает государственный праздник**
- **SchoolHoliday: указывает на то, что было оказано влияние на конкретный магазин в этот день закрытием школ**

Описание данных. Store

- StoreType: 4 типа магазина (a, b, c, d)
- Assortment: уровень ассортимента (a = basic, b = extra, c = extended)
- CompetitionDistance: расстояние в метрах до ближайшего магазина-конкурента
- CompetitionOpenSince[Month/Year]: год/месяц открытия ближайшего конкурента.
- Promo2: продолжающаяся и последовательная акция для некоторых магазинов: 0 = магазин не участвует, 1 = магазин участвует
- Promo2Since [Year/Week]: год и календарная неделя, когда магазин начал участвовать в Promo2
- PromoInterval: последовательные интервалы запуска Promo2

Эксперимент №1

Линейная регрессия

Идея эксперимента

Идея: построить простейшую линейную модель на количественных признаках.

Произведем merge таблиц Store и Train и выделим необходимые данные:

	Store	DayOfWeek	Promo	SchoolHoliday	Promo2	Open	CompetitionDistance
0	1	5	1	1	0	1	1270.0
1	2	5	1	1	1	1	570.0

Подготовка данных

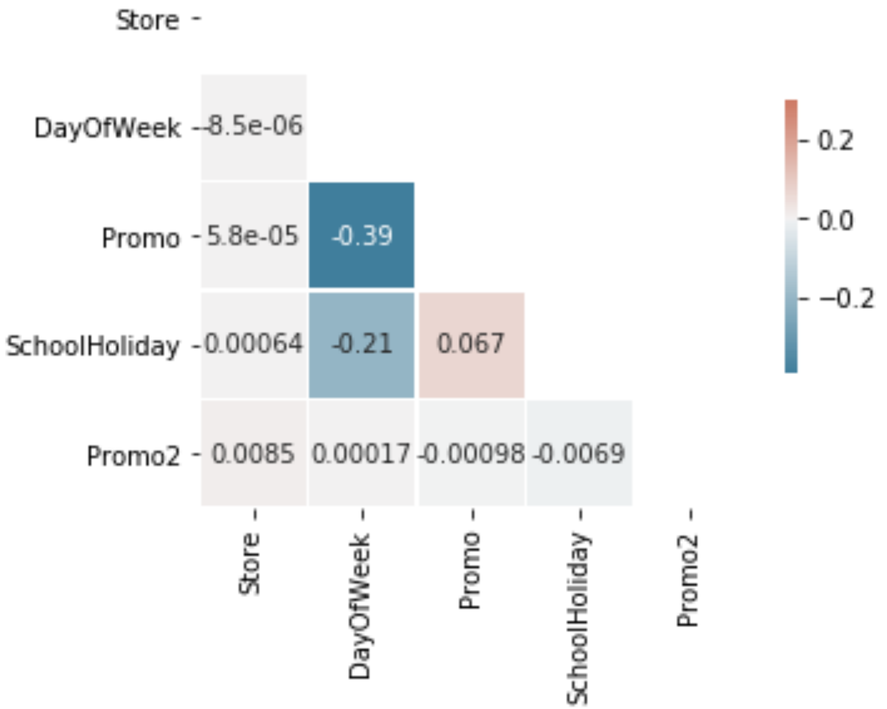
В последних двух колонках есть NaN-ы. В рамках этого эксперимента выбросим эти признаки.

В таблице Test отсутствует признак Customers - также исключим его.

	Store	DayOfWeek	Promo	SchoolHoliday	Promo2
0	1	5	1	1	0
1	2	5	1	1	1

Анализ данных

Построим корреляционную матрицу:



Высоко скорректированных признаков нет.

Моделирование

Применим простейшую линейную регрессию.

```
lr = LinearRegression()
lr.fit(X_train, y_train)
y_test = lr.predict(X_test)
```

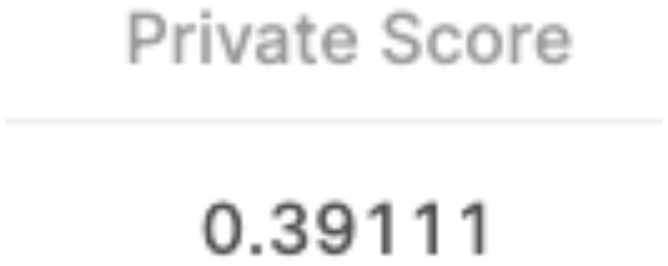
Коэффициенты модели:

	coef	std err	t	P>|t|	[0.025	0.975]
Store	3.3144	0.010	318.026	0.000	3.294	3.335
DayOfWeek	237.7303	1.430	166.248	0.000	234.928	240.533
Promo	4871.8091	7.513	648.415	0.000	4857.083	4886.535
SchoolHoliday	1432.8032	9.869	145.175	0.000	1413.459	1452.147
Promo2	504.6863	7.360	68.576	0.000	490.262	519.111

Для всех признаков p-value < 0.05, соответственно каждый коэффициент можно считать статистически значимым.

Оценка эксперимента

Результат:



Вывод: не самый плохой скор, учитывая, что это самая простая модель на минимальном количестве признаков. Однако глобально все еще довольно слабо.

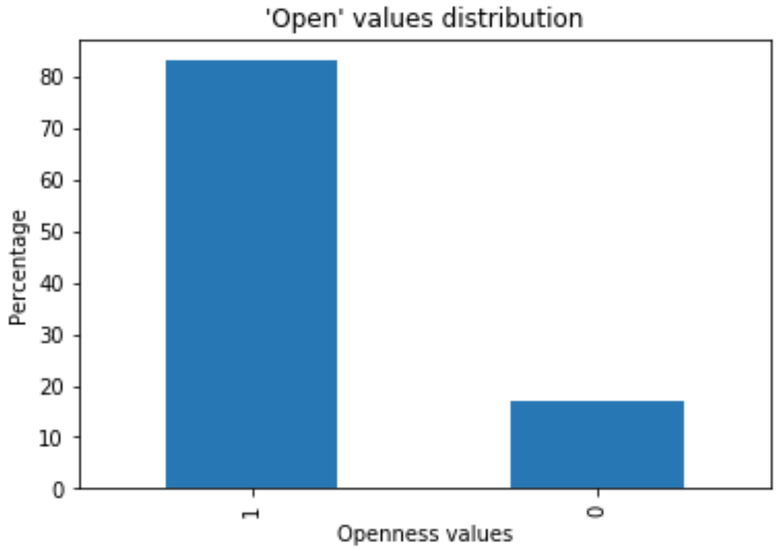
Эксперимент №2

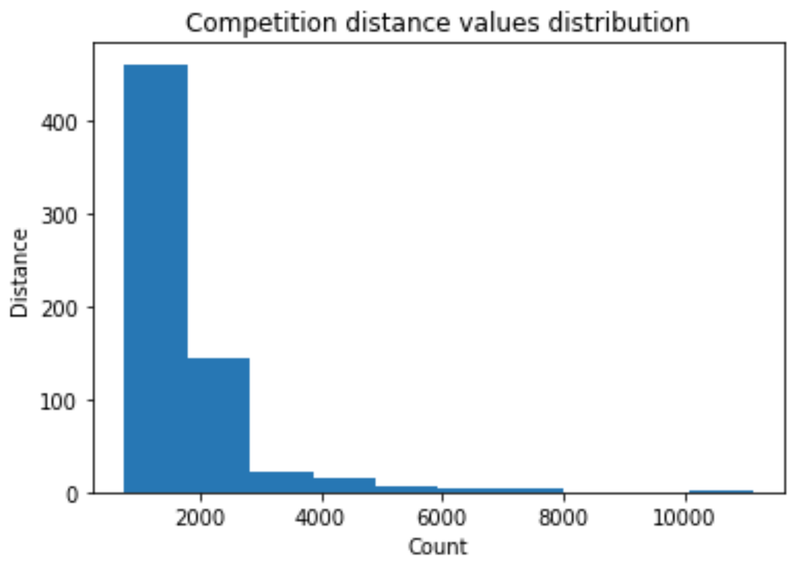
Линейная регрессия с дополнительными признаками

Идея эксперимента

Идея: в прошлом эксперименте мы исключили колонки с NaN-ами. В этот раз добавим эти признаки, заполнив пропущенные значения.

Посмотрим на значения признаков Open и CompetitionDistance:





Подготовка данных

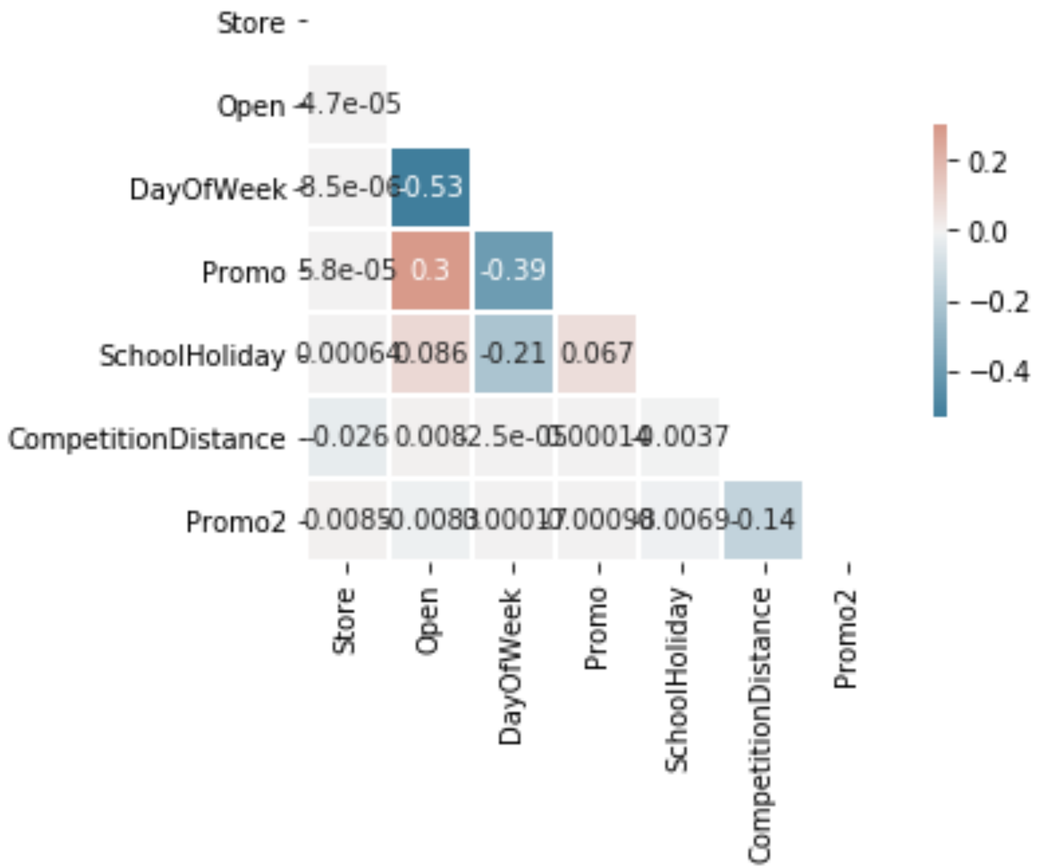
На основании этого можно заполнить NaN-ы для Open нулями, а NaN-ы для CompetitionDistance - медианным значением.

```
X_train.isna().sum()
```

```
Store                 0
Open                  0
DayOfWeek             0
Promo                 0
SchoolHoliday         0
CompetitionDistance   0
Promo2                0
```

Анализ данных

Матрица корреляции:



Моделирование

Также применим LinearRegression:

```
lr = LinearRegression()
lr.fit(X_train, y_train)
y_test = lr.predict(X_test)
```

Коэффициенты модели:

const	967.2707	13.530	71.490	0.000	940.752	993.789
Store	0.0593	0.008	7.372	0.000	0.043	0.075
Open	5849.1015	8.170	715.907	0.000	5833.088	5865.115
DayOfWeek	-108.6862	1.623	-66.954	0.000	-111.868	-105.505
Promo	2071.4714	5.827	355.482	0.000	2060.050	2082.893
SchoolHoliday	60.0670	6.903	8.702	0.000	46.538	73.596
CompetitionDistance	-0.0180	0.000	-53.026	0.000	-0.019	-0.017
Promo2	-701.0235	5.224	-134.187	0.000	-711.263	-690.784

Для всех признаков p-value < 0.05, соответственно каждый коэффициент можно считать статистически значимым.

Оценка эксперимента

Результат:

Private Score
0.44916

Вывод: на удивление скор не улучшился (а даже довольно сильно ухудшился). Природа пропущенных значений неизвестна, а значит, что если мы и хотим учитывать данные признаки в дальнейшем, то это нужно делать осторожно.

Эксперимент №3

Линейная регрессия без признаков с малыми весами

Идея эксперимента

Обратимся к предыдущей модели:

const	967.2707	13.530	71.490	0.000	940.752	993.789
Store	0.0593	0.008	7.372	0.000	0.043	0.075
Open	5849.1015	8.170	715.907	0.000	5833.088	5865.115
DayOfWeek	-108.6862	1.623	-66.954	0.000	-111.868	-105.505
Promo	2071.4714	5.827	355.482	0.000	2060.050	2082.893
SchoolHoliday	60.0670	6.903	8.702	0.000	46.538	73.596
CompetitionDistance	-0.0180	0.000	-53.026	0.000	-0.019	-0.017
Promo2	-701.0235	5.224	-134.187	0.000	-711.263	-690.784

Видим, что у признаков Store и CompetitionDistance малые веса.
Идея: улучшить скор путем выбрасывания этих признаков.

Оценка эксперимента

Результат:

Private Score
0.44567

Вывод: Скор улучшился незначительно относительно эксперимента №2, однако все еще сильно ниже, чем у эксперимента №1. Следовательно

Эксперимент №4

Линейная регрессия с MinMaxScale

Идея и оценка эксперимента

Хорошей практикой является нормализация признаков с помощью MinMaxScaler. Проделаем это:

Результат:

Private Score
0.44567

Вывод: Скор не изменился. Все еще использовался признак Open, так как ожидалось, что с помощью нормализации скор улучшиться (а признак Open использовать хочется - он выглядит довольно важным).

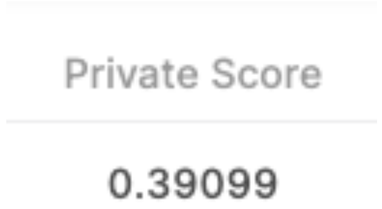
Эксперимент №5

Линейная регрессия с MinMaxScale для лучшего набора параметров

Идея и оценка эксперимента

Предыдущие эксперименты показали, что без параметра «Open» скор получается лучше. Применим MinMaxScale к тому, что было в эксперименте 1 (набор признаков, который пока дает лучший результат):

Результат:



Вывод: Скор незначительно изменился в лучшую сторону (0.390 против 0.391). MinMaxScale однозначно стоит применять в дальнейшем, признак «Open» пока будем опускать. Больше из LR ничего не выжать, нужно начать учитывать категориальные признаки.

Эксперимент №6

Линейная регрессия с добавлением простейших категориальных признаков

Идея эксперимента. Обработка данных

Добавим к предыдущему набору признаков еще и закодированные категориальные. К ним будут относиться StateHoliday, StoreType и Assortment.

Чтобы избежать линейной зависимости полученных признаков, будем удалять по одному после кодирования.

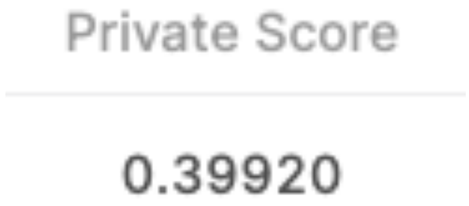
```
= pd.get_dummies(merged_train[cat_cols], drop_first=True)
pd.get_dummies(merged_test[cat_cols], drop_first=True)
```

Датасет будет иметь вид:

DayOfWeek	Promo	SchoolHoliday	Promo2	StateHoliday_a	StateHoliday_b	StateHoliday_c	StoreType_b	StoreType_c	StoreType_d	Assortment_b	Assortment_c
5	1	1	0	0	0	0	0	1	0	0	0
5	1	1	1	0	0	0	0	0	0	0	0

Оценка эксперимента

Результат:



Вывод: Скор ухудшился, но незначительно. Вероятно, происходит переобучение, «запасы» простейшей линейной модели иссякают. Имеет смысл перейти к более продвинутым моделям.

Эксперимент №7

Использование xgboost

Идея эксперимента

Будем обучать XGBRegressor на датасете из предыдущего эксперимента. Тюнинг гиперпараметров будем проводить с помощью CV и прохода по циклу из различных значений.

Окончательно выбраны следующие параметры:

```
xgb = XGBRegressor(n_jobs=-1, random_state=42, n_estimators=1000,
                   learning_rate=0.2, max_depth=10, subsample=0.9,
                   colsample_bytree=0.7, booster='gbtree'
                  )
```

Оценка эксперимента

Результат:

Вывод: ожидалось, что скор значительно улучшится, но в итоге он немного упал. Непонятно, с чем это связано. Попробуем тщательнее поработать с признаками.

Эксперимент №8

Использование xgboost с добавлением признака Open

Идея и оценка эксперимента

Попробуем добавить в датасет значения признака Open.

Результат:

Private Score
0.40566

Вывод: Скор незначительно изменился в лучшую сторону. Будем использовать этот набор параметров, но также необходимы более продвинутые способы обработки имеющихся данных.

Эксперимент №9

Использование xgboost с другим параметром lr

Идея и оценка эксперимента

На CV немногим хуже показала себя модель с параметром learning_rate = 0.9. Попробуем ее.

Результат:

Private Score
0.40565

Вывод: Скор минимально изменился в лучшую сторону. Можно использовать это значение learning_rate.

Эксперимент №10

Проверка гипотезы о выходном дне

Идея и оценка эксперимента

Гипотеза: нам не так важно значение DayOfWeek, сколько тот факт, выходной это или нет. Для реализации заменим все значения DayOfWeek в датасете на 0, если значение <6, и на 1 в противном случае.

Результат:

Private Score
0.42292

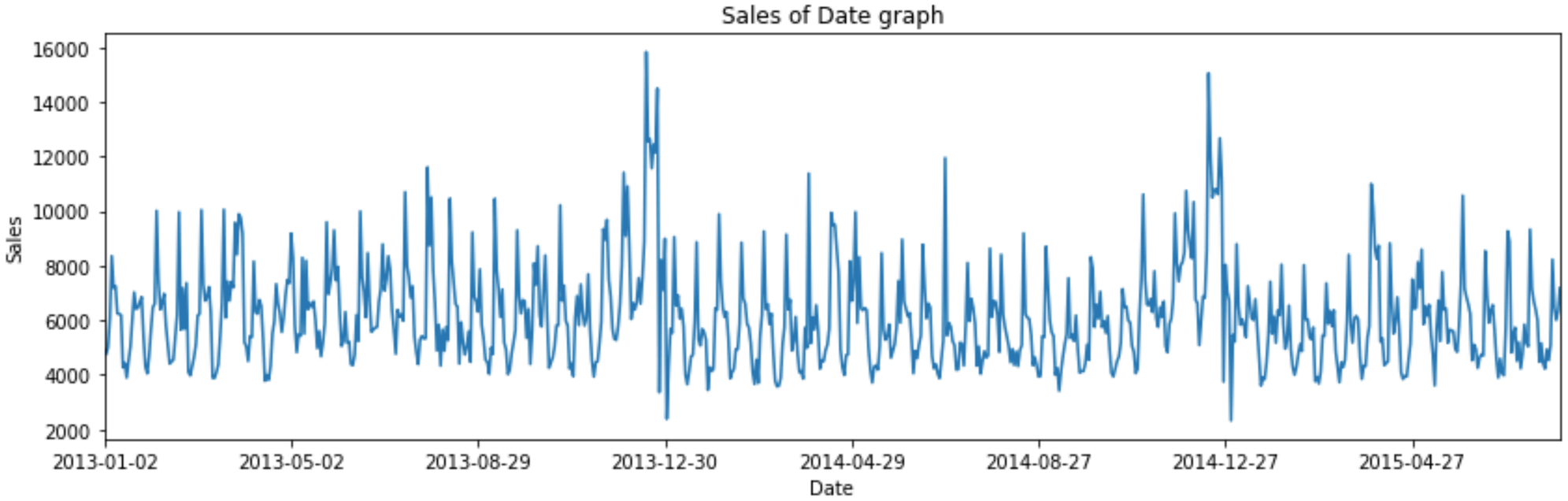
Вывод: Гипотеза несостоятельна

Эксперимент №11

Добавление новых признаков, связанных с датой

Идея эксперимента

Посмотрим на график Sales(Date) для одного из магазинов:



Можно заметить некоторую периодичность и сезонность, соответственно введем признаки, связанные с датами.

Обработка данных

Будем выделять отдельно день недели (Day), неделю в году (WeekOfYear), месяц (Month) и год (Year).

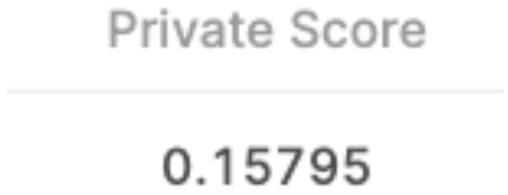
Помимо этого введем 2 параметра - Promo2Open и IsPromo2Month. Они соответствуют числу месяцев с тех пор, когда Promo2 было запущено и индикатору того, что текущий месяц это Promo2 месяц

```
def promo_cols(df):
    df['Promo2Open'] = 12 * (df.Year - df.Promo2SinceYear) +  (df.WeekOfYear - df.Promo2SinceWeek)*7/30.5
    df['Promo2Open'] = df['Promo2Open'].map(lambda x: 0 if x < 0 else x).fillna(0) * df['Promo2']
    df['IsPromo2Month'] = df.apply(check_promo_month, axis=1) * df['Promo2']
```

Оценка эксперимента

Далее запустим XGBoost с теми же параметрами, что и в предыдущем разе.

Результат:



Вывод: значительное увеличение скора в следствие выделения периодичности в датасете, а также более тщательной работой с признаком Promo2.

Эксперимент №12

Удаление магазинов с нулевой выручкой

Идея эксперимента. Анализ данных

В датасете имеются пропуски, NaN-ы и нулевые значения. Интересуют, в частности, нулевые значения Sales. Это может быть связано с закрытыми магазинами:

Store	DayOfWeek	Date	Sales	Customers	Open
292	5	2015-07-31	0	0	0
876	5	2015-07-31	0	0	0
292	4	2015-07-30	0	0	0
876	4	2015-07-30	0	0	0
292	3	2015-07-29	0	0	0

Подобных наблюдений - 16%. Чтобы избежать смещения предсказания, имеет смысл удалить их.

Анализ данных

Однако это необязательное условие. Есть наблюдения с нулевыми Sales и работающими магазинами:

Store	DayOfWeek	Date	Sales	Customers	Open	Promo
971	5	2015-05-15	0	0	1	0
674	4	2015-03-26	0	0	1	0
699	4	2015-02-05	0	0	1	1
708	3	2014-10-01	0	0	1	1
357	1	2014-09-22	0	0	1	0

Таких наблюдений - всего 54. С ними ничего не поделаешь, причины неясны, удалять нельзя.

Оценка эксперимента

Аналогично запустим XGBoost.

Результат:

Private Score
0.14315

Вывод: не такое значительное, но все-таки заметное улучшение скора. Гипотеза удаления магазинов с нулевым доходом, когда они закрыты состоятельна.

Итоги

- **В этот раз был освоен правильный способ использования CRISP. Удалось убедиться в удобстве и эффективности этого подхода**

- **Удалось провести оптимальный тюнинг гиперпараметров в моделях с XGBoost, что позволило получить хороший скор.**

- **Было проведено большое количество последовательных экспериментов с отличием в одну новую идею, что позволило не потеряться и выделить оптимальные датасеты и модели**

- **Благодаря методологии CRISP были зафиксированы все полученные численные результаты**

Дальнейшие планы

- **Использовать CRISP во всех последующих работах**
- **Тестить большее количество гипотез, моделей**
- **Набивать руку на подобных соревнованиях, что позволит заранее видеть несостоятельные гипотезы, как следствие делать бОльшие шаги между экспериментами**