Прогноз продаж

Кузнецов Г.И.

Введение

Используемый набор данных — данные магазина.

Он управляет более чем 3000 аптек в 7 европейских странах.

Материалы:

- habr.com/ru/companies/mvideo/articles/769190/
- https://habr.com/ru/companies/mvideo/articles/769756/

Цель: предсказать ежедневные продажи на срок до шести недель вперед

Задачи:

- 1. EDA
- 2. Анализ временных рядов
- 3.Прогнозирование моделирование
- 4. Результаты
- 5. Презентация

Описание данных

Имеем достаточно **объемный** набор данных о продажах различных магазинов со всей подробной информацией о них.

Размеры датасетов:

```
print(store.shape)
print(train.shape)
print(test.shape)
```
✓ 0.0s

```
(1115, 15)
(844392, 17)
(35093, 14)
```

```
store.info()
```
✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Store                         1115 non-null   int64
 1   CompetitionDistance           1115 non-null   float64
 2   CompetitionOpenSinceMonth     1115 non-null   float64
 3   CompetitionOpenSinceYear      1115 non-null   float64
 4   Promo2                        1115 non-null   int64
 5   Promo2SinceWeek               1115 non-null   float64
 6   Promo2SinceYear               1115 non-null   float64
 7   StoreType_b                   1115 non-null   bool
 8   StoreType_c                   1115 non-null   bool
 9   StoreType_d                   1115 non-null   bool
 10  Assortment_b                  1115 non-null   bool
 11  Assortment_c                  1115 non-null   bool
 12  PromoInterval_Jan,Apr,Jul,Oct 1115 non-null   bool
 13  PromoInterval_Mar,Jun,Sept,Dec 1115 non-null  bool
 14  PromoInterval_None            1115 non-null   bool
dtypes: bool(8), float64(5), int64(2)
memory usage: 69.8 KB
```

Пример обработанного датасета с магазинами*

Предобработка

Данные оказались очень грязными, при этом обрабатывать отдельно пришлось несколько датасетов. В целом, из основного:

- приведение к числовому типу
- работа с датой (разобрать на отдельный фичи)
- исправление ошибочных значений (не соответствующих типу колонки)
- работа с пропусками

Также сразу убрал закрытые магазины.

Предобработка: результаты

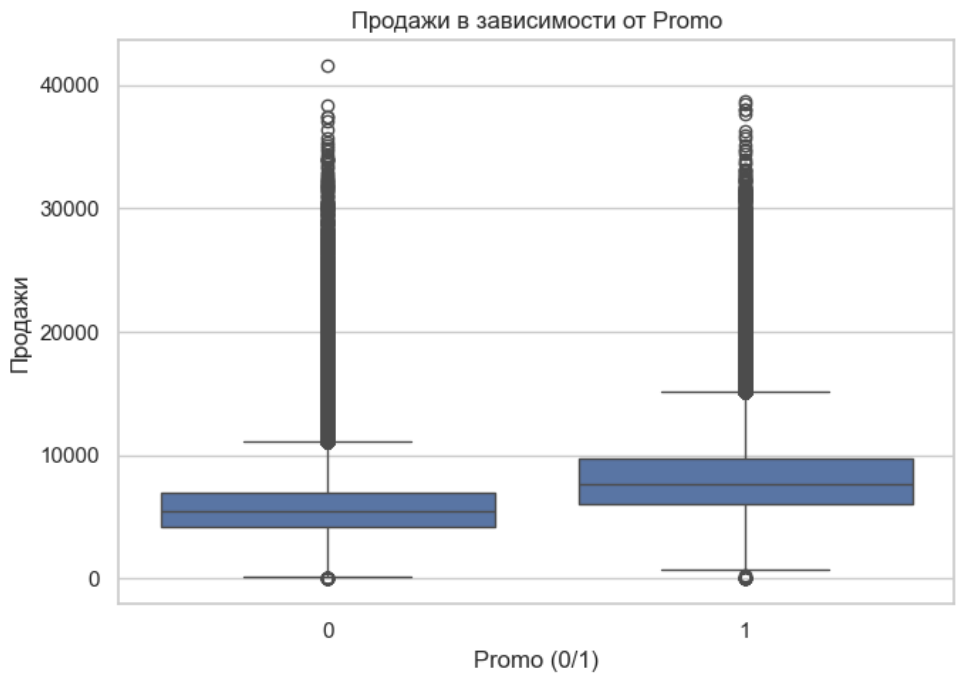
```
train.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
Index: 844392 entries, 0 to 1017190
Data columns (total 17 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Store            844392 non-null  int64
 1   DayOfWeek        844392 non-null  int64
 2   Sales            844392 non-null  int64
 3   Customers        844392 non-null  int64
 4   Open             844392 non-null  int64
 5   Promo            844392 non-null  int64
 6   SchoolHoliday    844392 non-null  int64
 7   StateHoliday__0  844392 non-null  bool
 8   StateHoliday__a  844392 non-null  bool
 9   StateHoliday__b  844392 non-null  bool
 10  StateHoliday__c  844392 non-null  bool
 11  Year             844392 non-null  int32
 12  Month            844392 non-null  int32
 13  Week             844392 non-null  UInt32
 14  Day              844392 non-null  int32
 15  DayOfYear        844392 non-null  int32
 16  IsWeekend        844392 non-null  int64
dtypes: UInt32(1), bool(4), int32(4), int64(8)
memory usage: 78.1 MB
```

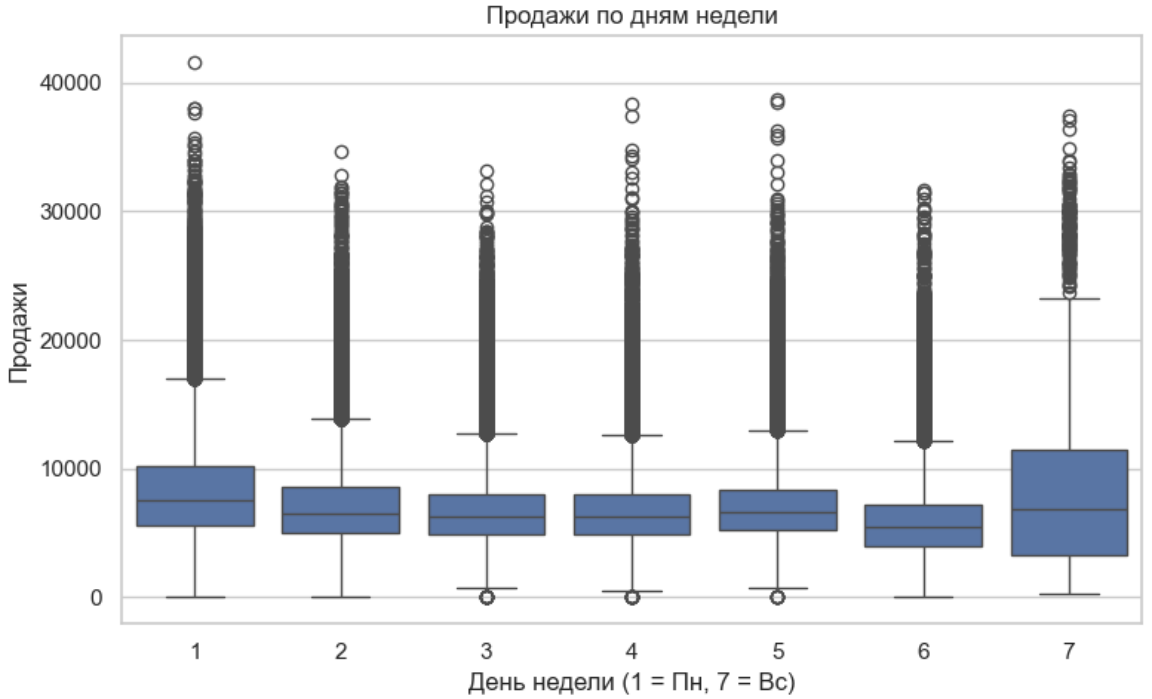
```
test.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
Index: 35093 entries, 0 to 41087
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Id               35093 non-null  int64
 1   Store            35093 non-null  int64
 2   DayOfWeek        35093 non-null  int64
 3   Open             35093 non-null  int64
 4   Promo            35093 non-null  int64
 5   SchoolHoliday    35093 non-null  int64
 6   StateHoliday__0  35093 non-null  bool
 7   StateHoliday__a  35093 non-null  bool
 8   Year             35093 non-null  int32
 9   Month            35093 non-null  int32
 10  Week             35093 non-null  UInt32
 11  Day              35093 non-null  int32
 12  DayOfYear        35093 non-null  int32
 13  IsWeekend        35093 non-null  int64
dtypes: UInt32(1), bool(2), int32(4), int64(7)
memory usage: 2.9 MB
```

```
store.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 15 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Store                           1115 non-null   int64
 1   CompetitionDistance             1115 non-null   float64
 2   CompetitionOpenSinceMonth       1115 non-null   float64
 3   CompetitionOpenSinceYear        1115 non-null   float64
 4   Promo2                          1115 non-null   int64
 5   Promo2SinceWeek                 1115 non-null   float64
 6   Promo2SinceYear                 1115 non-null   float64
 7   StoreType_b                     1115 non-null   bool
 8   StoreType_c                     1115 non-null   bool
 9   StoreType_d                     1115 non-null   bool
 10  Assortment_b                    1115 non-null   bool
 11  Assortment_c                    1115 non-null   bool
 12  PromoInterval_Jan,Apr,Jul,Oct   1115 non-null   bool
 13  PromoInterval_Mar,Jun,Sept,Dec  1115 non-null   bool
 14  PromoInterval_None              1115 non-null   bool
dtypes: bool(8), float64(5), int64(2)
memory usage: 69.8 KB
```

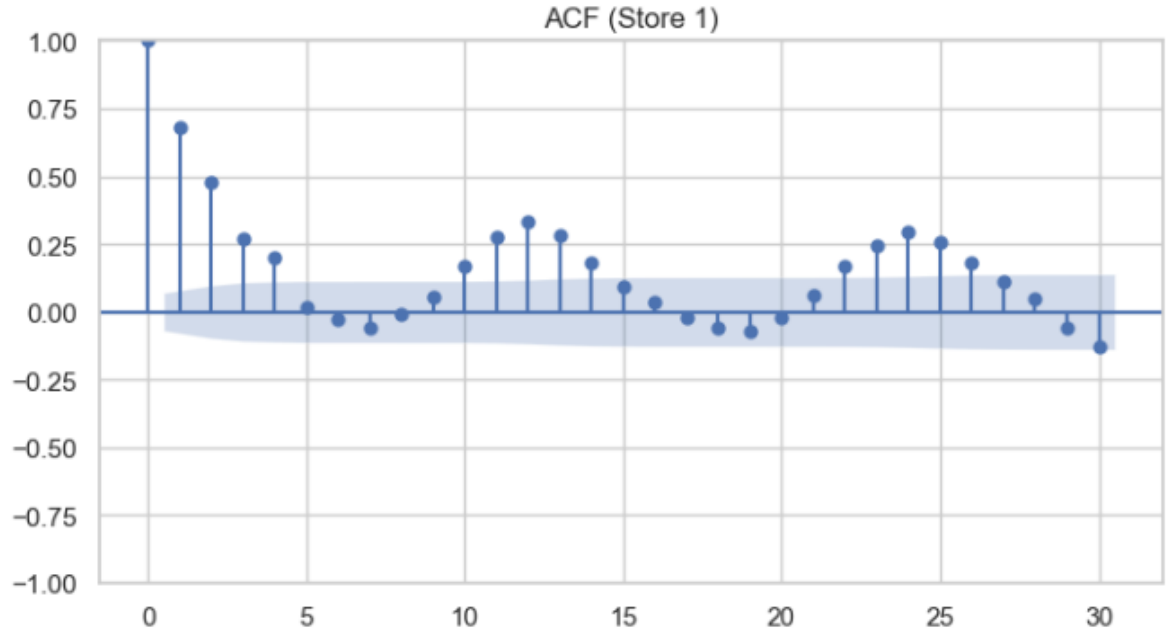
EDA

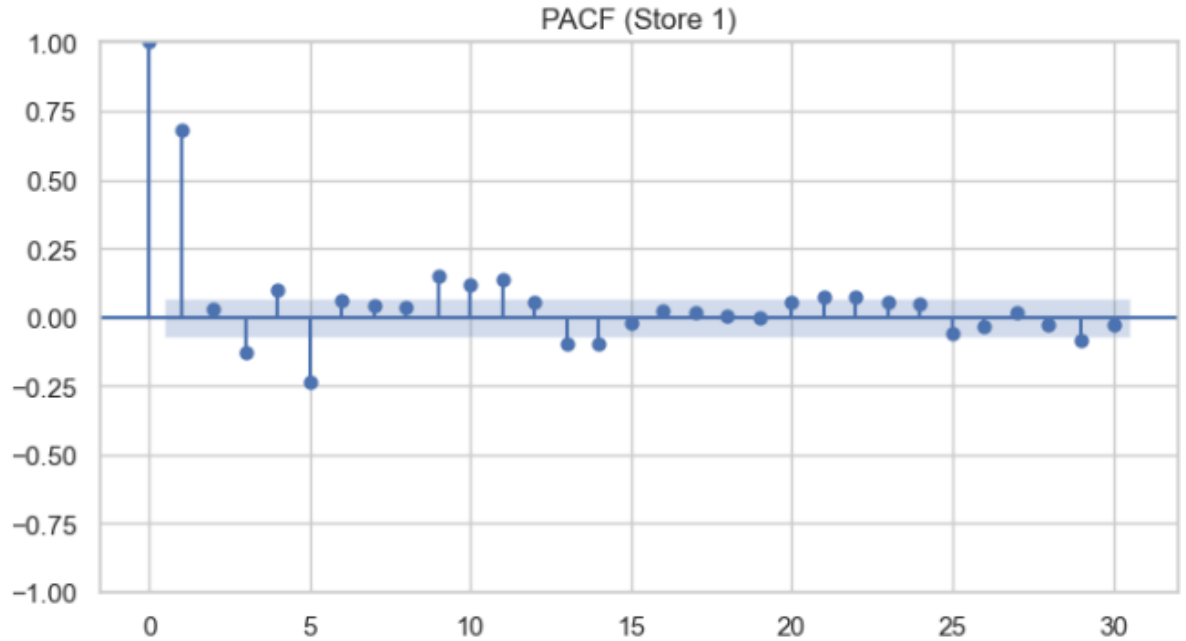

Распределение продаж
Частота
Продажи
0
10000
20000
30000
40000
50000
60000
0
5000
10000
15000
20000
25000


Продажи в зависимости от Promo
Продажи
Promo (0/1)
0
10000
20000
30000
40000
0
1


Продажи по дням недели
Продажи
День недели (1 = Пн, 7 = Вс)
0
10000
20000
30000
40000
1
2
3
4
5
6
7

Временные ряды





Моделирование

```
X_train = train_merged.drop('Sales', axis=1)
y_train = train_merged['Sales']
X_test = test_merged
```

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)

train_preds = model.predict(X_train)

mae = mean_absolute_error(y_train, train_preds)
rmse = root_mean_squared_error(y_train, train_preds)
r2 = r2_score(y_train, train_preds)

print(f"Train MAE: {mae:.2f}")
print(f"Train RMSE: {rmse:.2f}")
print(f"Train R2: {r2:.4f}")
```

✓ 1.2s

```
Train MAE: 929.47
Train RMSE: 1277.32
Train R2: 0.8307
```

```
from sklearn.ensemble import HistGradientBoostingRegressor

model = HistGradientBoostingRegressor(random_state=42)
model.fit(X_train, y_train)

train_preds = model.predict(X_train)

mae = mean_absolute_error(y_train, train_preds)
rmse = root_mean_squared_error(y_train, train_preds)
r2 = r2_score(y_train, train_preds)

print(f"Train MAE: {mae:.2f}")
print(f"Train RMSE: {rmse:.2f}")
print(f"Train R2: {r2:.4f}")
```

✓ 9.0s

```
Train MAE: 566.57
Train RMSE: 773.01
Train R2: 0.9380
```

Прогнозирование

```
test_preds = model.predict(X_test)

predicted = pd.DataFrame()
predicted["Date"] = pd.to_datetime(dict(year=test_merged["Year"],
                                        month=test_merged["Month"],
                                        day=test_merged["Day"]))
predicted["Store"] = test_merged["Store"]
predicted["PredictedSales"] = test_preds

predicted.head()
```

✓ 0.1s

	Date	Store	PredictedSales
0	2015-09-17	1	2022.169540
1	2015-09-17	3	2343.529136
2	2015-09-17	7	2303.708169
3	2015-09-17	8	2160.992208
4	2015-09-17	9	2449.678458

Date	Store	PredictedSales	Date	Store	PredictedSales
2015-08-01	1	1871.120673	2015-08-27	1	1726.489610
2015-08-03	1	2457.648947	2015-08-28	1	1728.613092
2015-08-04	1	2286.201628	2015-08-29	1	1762.063296
2015-08-05	1	2159.811143	2015-08-31	1	2788.270236
2015-08-06	1	2068.399824	2015-09-01	1	2503.506443
2015-08-07	1	2068.399824	2015-09-02	1	2381.215120
2015-08-08	1	1623.032260	2015-09-03	1	2081.310300
2015-08-10	1	1706.004617	2015-09-04	1	2081.310300
2015-08-11	1	1677.690820	2015-09-05	1	1624.149807
2015-08-12	1	1677.690820	2015-09-07	1	1674.560825
2015-08-13	1	1677.690820	2015-09-08	1	1663.080585
2015-08-14	1	1624.810794	2015-09-09	1	1663.080585
2015-08-15	1	1623.032260	2015-09-10	1	1663.080585
2015-08-17	1	2408.956639	2015-09-11	1	1610.200560
2015-08-18	1	2222.156571	2015-09-12	1	1607.247092
2015-08-19	1	2104.047255	2015-09-14	1	2392.885384
2015-08-20	1	2008.249709	2015-09-15	1	2238.271623
2015-08-21	1	2008.249709	2015-09-16	1	2128.298193
2015-08-22	1	1561.902727	2015-09-17	1	2022.169540
2015-08-24	1	1645.653007			
2015-08-25	1	1601.986460			
2015-08-26	1	1625.831391			

Выводы

- проведен исследовательский анализ данных (EDA), выявлены ключевые зависимости и особенности продаж;
- выполнен анализ временных рядов: сезонность, тренды, проверка стационарности;
- построены и обучены модели для прогнозирования продаж, протестированы различные алгоритмы;
- спрогнозированы продажи на срок до шести недель вперёд!

Спасибо за внимание