Mestrado em Data Science

Professor Luís Roque

Professor Tiago Otto

Professor João Reis

END-TO-END MACHINE LEARNING

Caso de estudo: DSMarket

Alisson Pereira Anjos

Jefferson Oliveira Melo

João Rego

Índice

Enquadramento

DSMarket – Loja de próxima geração

A DSMarket, anteriormente conhecido como TradiStores, é uma empresa do retalho com presença no Mercado Americano. A DSMarket apresenta serviço ao público, nomeadamente as suas lojas que estão presentes em três estados: New York, Boston e Philadelphia. A DSMarket encontra-se num processo de transformação digital e com isso fez rebranding, e de momento encontra-se em fase de restruturação interna de modo que no espaço de 5 anos seja reconhecida como a loja da próxima geração.

DSMarket – Caso de estudo

O presente trabalho consiste na criação de um cenário realista de trabalho para um cientista de dados. Como tal, foram fornecidos três ficheiros de dados de modo que seja possível executar as seguintes tarefas:

- Análise
- Agrupamento
- Modelo de Previsão de Vendas
- Reabastecimento de loja - com MLops

Data

Tal como referido anteriormente, foram disponibilizados três ficheiros de dados:

- calendar_with_events.csv
- item_prices.csv
- item_sales.csv

Vamos agora ver as variáveis e os seus significados nos respetivos ficheiros.

Ficheiro calendar_with_events.csv

Dados referentes ao calendário

- date - data no formato y-m-d
- weekday – dia da semana
- weekday_int – dia numérico da semana (1-Sábado, 7-Sexta)
- d – Identificador do dia
- event – o nome do evento, caso tenha

Ficheiro item_prices.csv

Dados referentes aos preços dos itens

- item – id do produto
- category – categoria do produto
- store_code – Código da loja alfanumérico
- yearweek – data do período do preço (formato ano-semana)
- sell_price – preço de venda do produto referente ao yearweek.

Nota: Os preços dos produtos são definidos semanalmente, de 7 em 7 dias. Caso não esteja presente o preço do produto, é indicador que não existiu venda do produto no referido yearweek.

Ficheiro item_sales.csv

Dados referentes às vendas

- id – identificador das vendas (combinação do item + store_code)
- item – id do produto
- category – categoria do produto
- department – departamento do produto
- store – loja do produto
- store_code – código da loja
- region – região do produto

Tarefa 1 – Análise

De forma a começar a explorar os ficheiros de dados, fomos averiguar quantas observações tinham em cada ficheiro e as suas respetivas dimensões. Constatou-se que:

- Dataset de vendas: (30490, 1920)
- Dataset de preços: (6965706, 5)
- Dataset dos eventos: (1913, 5)

De seguida, fomos ver as primeiras 5 observações de cada dataset:

Dataset de vendas

id	item	category	department	store	store_code	region	d_1	d_2	d_3	...
ACCESORIES_1_001_NYC_1	ACCESORIES_1_001	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...
ACCESORIES_1_002_NYC_1	ACCESORIES_1_002	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...
ACCESORIES_1_003_NYC_1	ACCESORIES_1_003	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...
ACCESORIES_1_004_NYC_1	ACCESORIES_1_004	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...
ACCESORIES_1_005_NYC_1	ACCESORIES_1_005	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...

Dataset de preços

	item	category	store_code	yearweek	sell_price
0	ACCESORIES_1_001	ACCESORIES	NYC_1	201328.0	12.7414
1	ACCESORIES_1_001	ACCESORIES	NYC_1	201329.0	12.7414
2	ACCESORIES_1_001	ACCESORIES	NYC_1	201330.0	10.9858
3	ACCESORIES_1_001	ACCESORIES	NYC_1	201331.0	10.9858
4	ACCESORIES_1_001	ACCESORIES	NYC_1	201332.0	10.9858

Dataset de eventos:

	date	weekday	weekday_int	d	event
0	2011-01-29	Saturday	1	d_1	NaN
1	2011-01-30	Sunday	2	d_2	NaN
2	2011-01-31	Monday	3	d_3	NaN
3	2011-02-01	Tuesday	4	d_4	NaN
4	2011-02-02	Wednesday	5	d_5	NaN

Observações retiradas dos ficheiros:

1. **Sales:**

   Estão incluídos todos os id's dos items, como também a categoria, departamento, loja, região e uma determinada coluna de vendas para cada dia, desde 2011-01-29 até 2016-04-24. Temos, portanto, informação referente a 1913 dias.

2. **Prices:**

   Têm os preços de venda dos items, como o respetivo código da loja e uma coluna com o ano_semana.

3. **Calendar:**

   Tem features das datas. É apresentado uma coluna event que denomina a presença de um determinado evento no dia ou não.

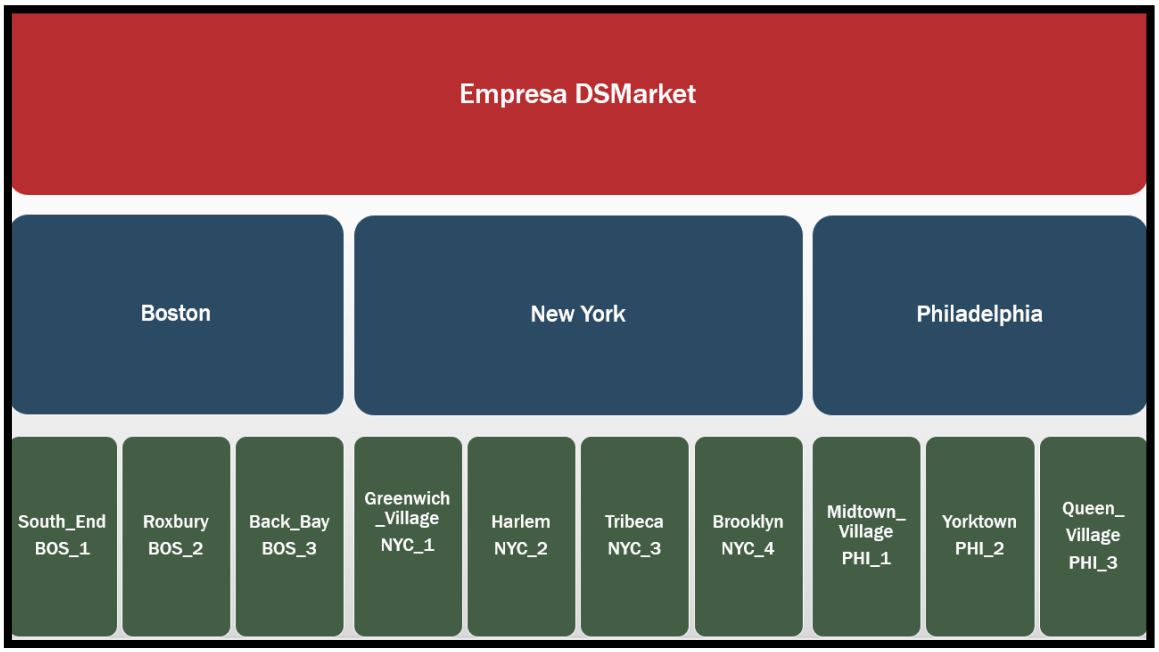
   No total existem a presença de 4 eventos: *SuperBowl, Ramadan starts, Thanksgiving, Newyear, Easter.*

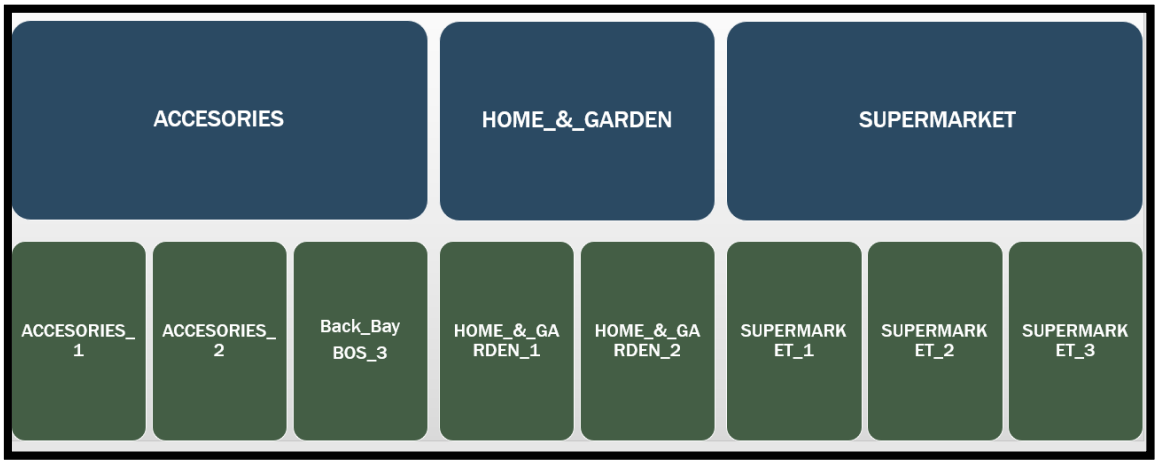
Seguidamente, fomos tomar conhecimento da estrutura interna da empresa DSMarket.

```
#Informações acerca dos dados

print('Nº de items:',len(sales['id'].unique()))
print('Nº de departamentos:',len(sales['department'].unique()))

print('Nº de categorias:',len(sales['category'].unique()))
print('Nº de lojas:',len(sales['store'].unique()))
print('Nº de regiões:',len(sales['region'].unique()))
```

```
Nº de items: 30490
Nº de departamentos: 7
Nº de categorias: 3
Nº de lojas: 10
Nº de regiões: 3
```

Empresa DSMarket – Estrutura de Negócio

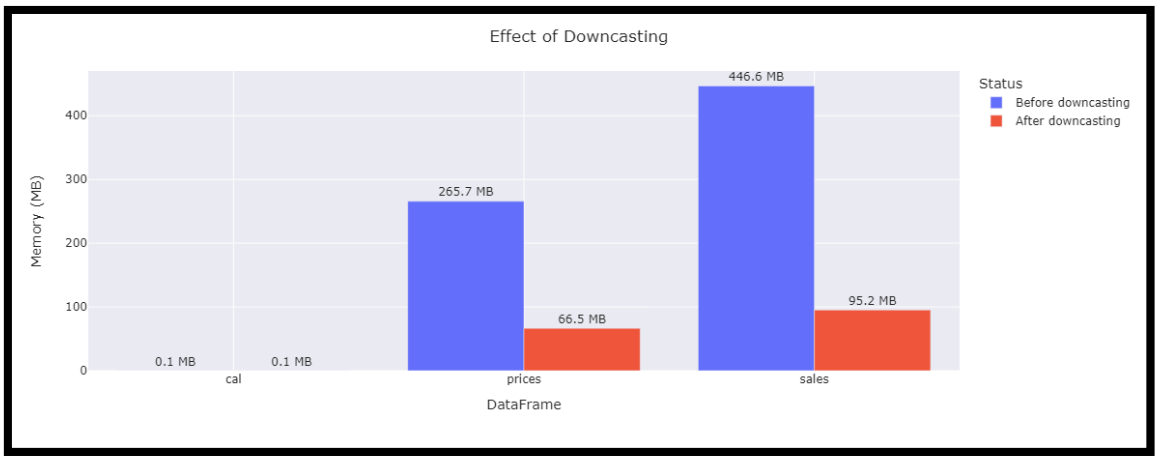




Redução do tamanho de dados

Devido ao facto de termos encontrado limitações a nível de tempo e de forma a conseguirmos quebrar de certo modo problemas futuros de erro de memória que iriam parar a execução do código e forçar o reiniciamento do script desde o início, utilizou-se a técnica de *downcast*. Esta técnica consiste na redução do tamanho dos dataframes, não perdendo a informação armazenada nos dataframes.

Para uma visualização do trade-off da memória dos ficheiros, averiguou-se primeiramente a memória presente nos 3 ficheiros de dados, seguido da aplicação do *downcasting* nos três ficheiros de dados e por fim a memória presente nos ficheiros de dados.



Como se pode verificar, os ficheiros prices e sales tiveram um decréscimo considerável, tal como pretendido. No entanto o ficheiro calendar não sofreu alterações de tamanho, isto, devido aos datatypes das suas variáveis.

Análise Exploratória

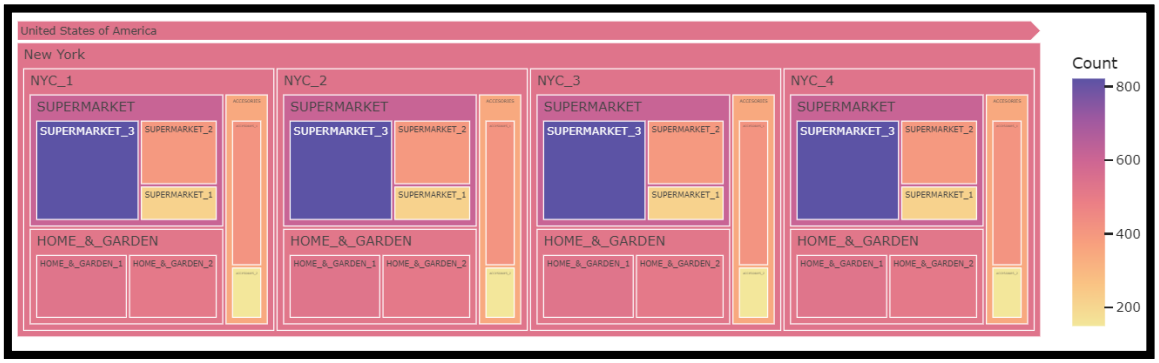
O objetivo da análise exploratória é descrever as propriedades dos nossos dados. Por isso, iremos recorrer a métodos descritivos que nos permitam ter uma melhor perceção do comportamento dos dados, isto é, recolher evidências estatísticas que nos permitam efetuar previsões com o menor erro possível.

Na perspetiva de negócio o nosso objetivo no presente ponto, é identificar o comportamento dos nossos produtos nas diferentes regiões, lojas e respetivas categorias.

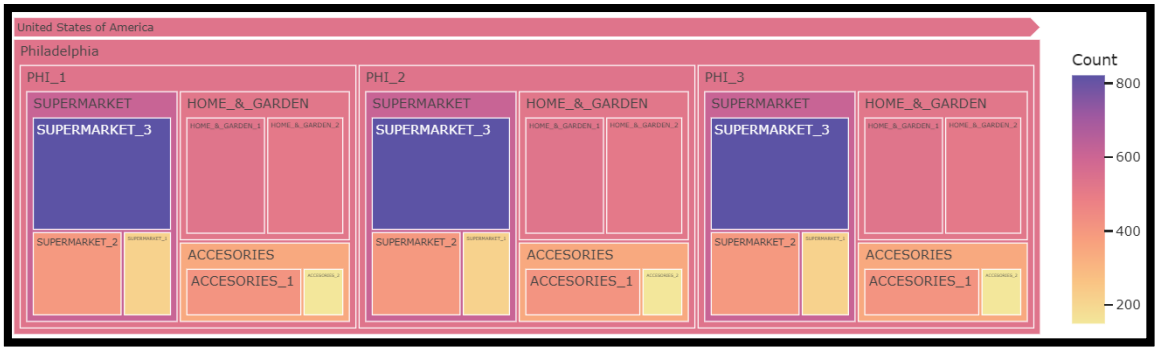
Comecemos então, por visualizar a distribuição dos items nas regiões de New York, Philadelphia e Boston.

Distribuição dos Items

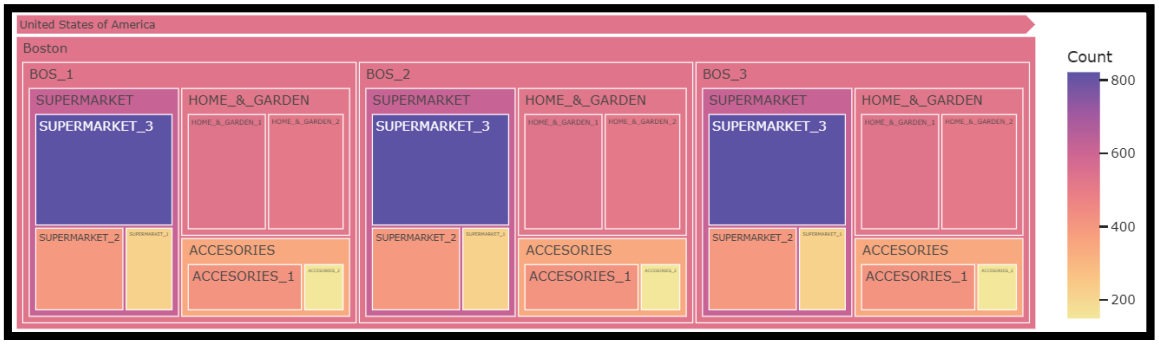
New York:



Philadelphia:



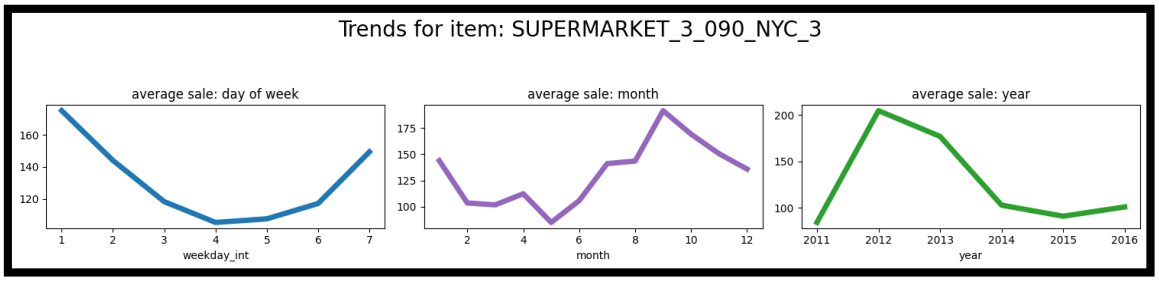
Boston:

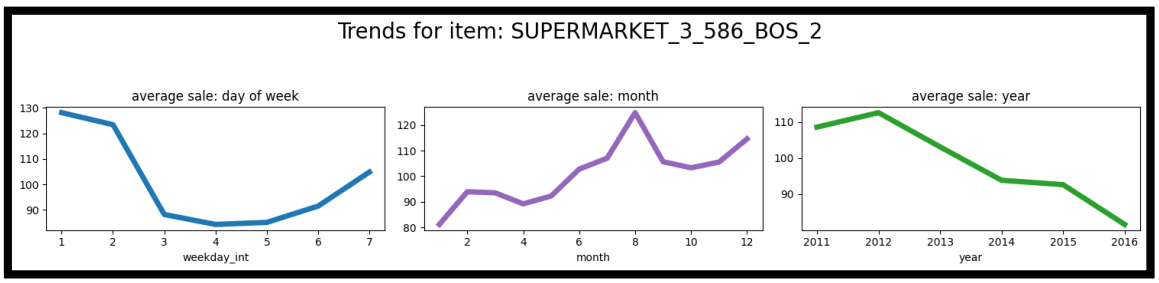


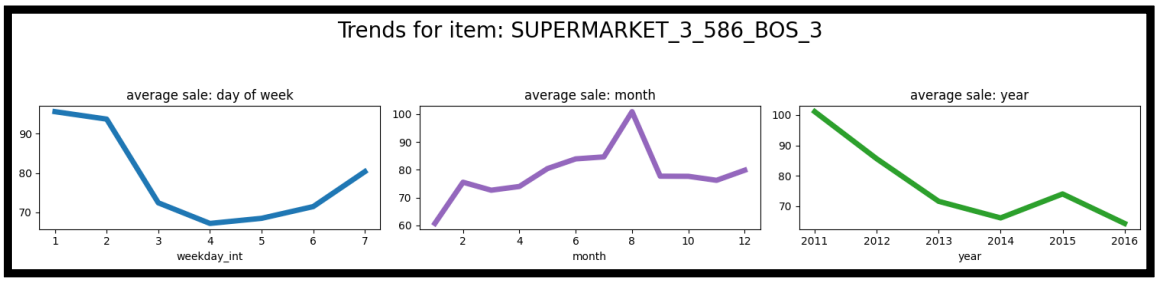
Análise descritiva das série

Item individual

Verificou-se quais os três itens mais vendidos, e representou-se os gráficos de tendência semanal, mensal e anual.

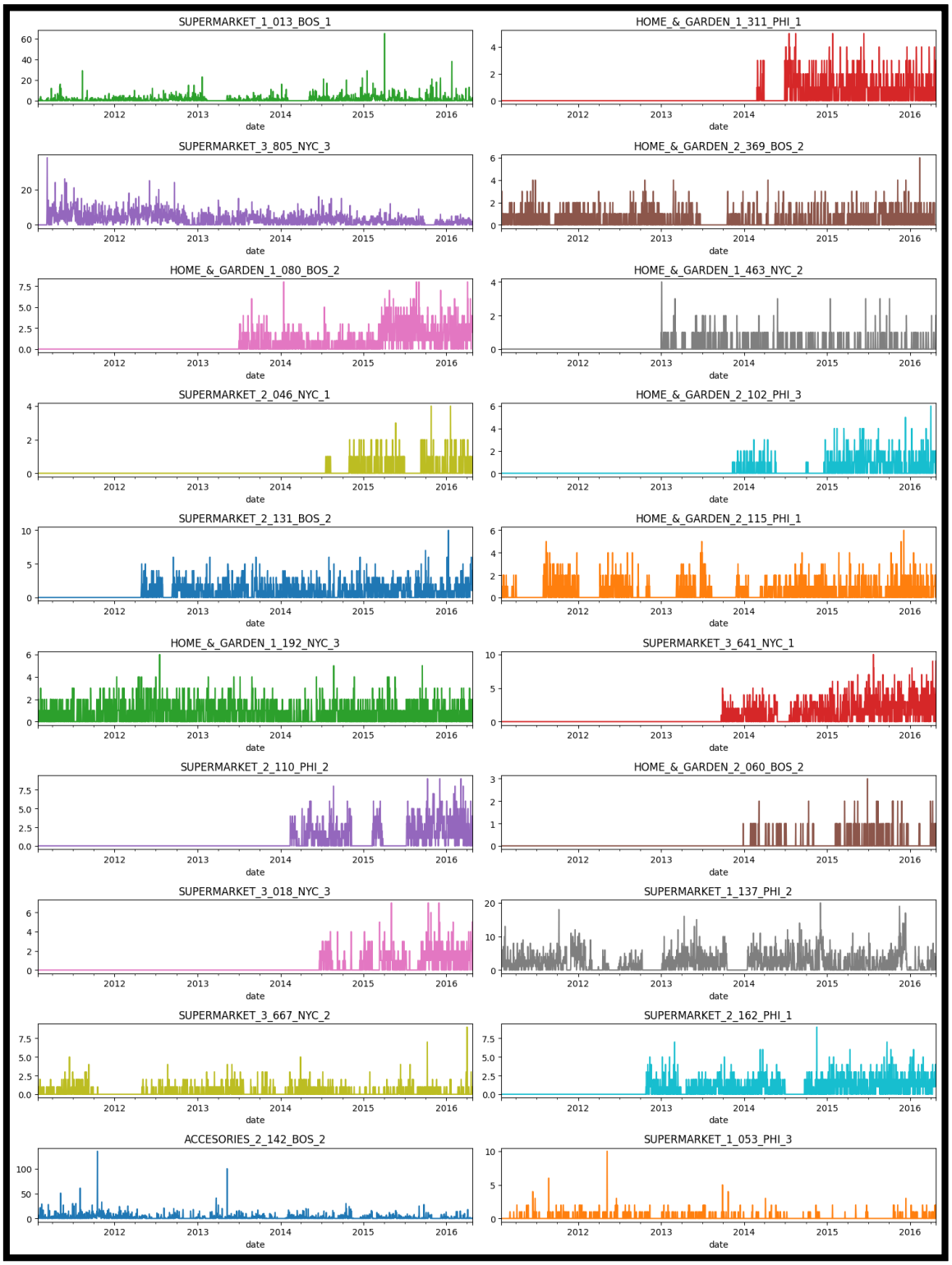






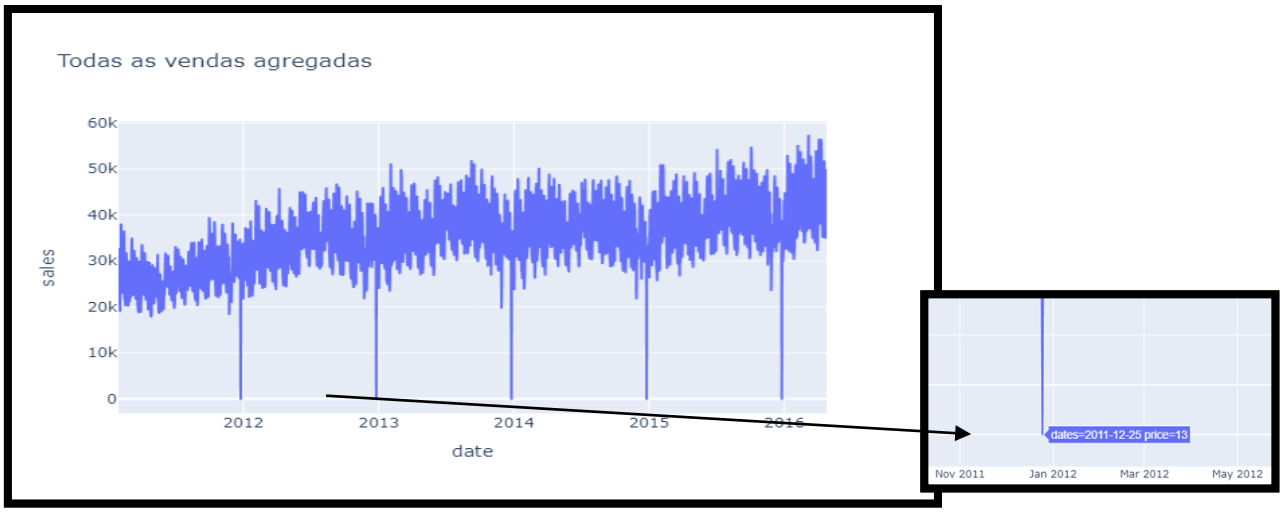
Como podemos observar os comportamentos nos diferentes períodos são muito semelhantes. Relativamente à tendência semanal, existe evidência que no dia 4, terça-feira, as vendas atingem o mínimo semanal. Relativamente à tendência mensal, no mês 8, o produto SUPERMARKET_3_586 atinge o seu ponto máximo mensal de vendas, enquanto o produto SUPERMARKET_3_090 é no mês 9. Anualmente todos os produtos apresentam uma tendência decrescente.

Vendas de 20 itens



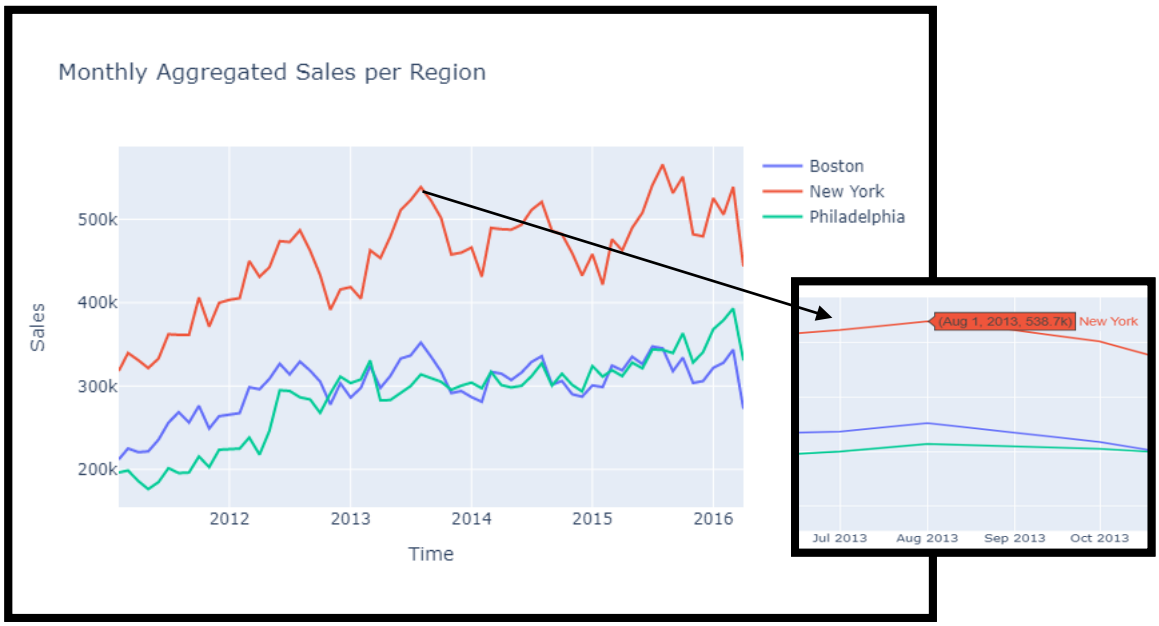
Série das vendas

Começaremos por visualizar os dados da série que representam as vendas totais com a finalidade de verificar o comportamento da série bem como a presença de padrões tais como a tendência e caso haja os diferentes tipos de sazonalidade.



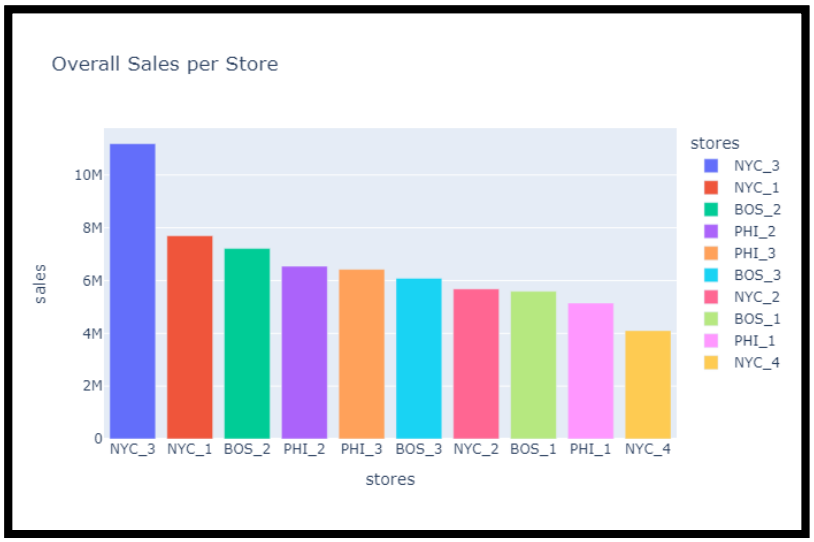
Observando o plot da série verifica-se que os dados apresentam dois comportamentos distintos relativamente à tendência. Verifica-se uma tendência um pouco decrescente e depois uma tendência crescente. É verificado também a presença de padrões cíclicos. Vendas muito próximo de zero podem ser observados todos os anos no Natal.

Vendas mensais por região



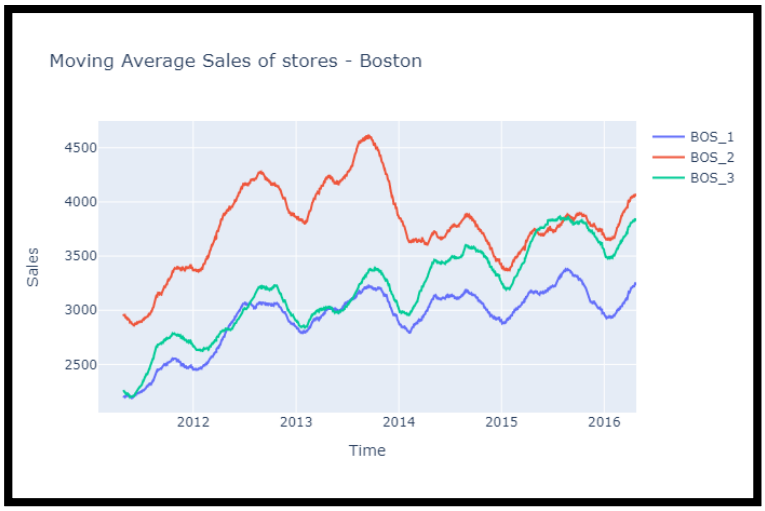
Visualizando as vendas mensais em cada região, destacasse a performance de vendas na região de New York comparativamente a Boston e Philadelphia. Tal facto, pode dever-se pelo número de lojas presentes em New York ser superior às de Boston e Philadelphia. Existe também evidência, de um comportamento semelhante entre a série de Boston e Philadelphia. Repare-se que inicialmente Boston apresenta uma performance superior relativamente a Philadelphia, contudo, em meados de 2015, verifica-se que a performance supramencionada é revertida. É visível também um padrão no que toca a um crescimento acentuado das três séries, em meados de agosto durante os o intervalo de tempo apresentado.

Total de vendas por lojas



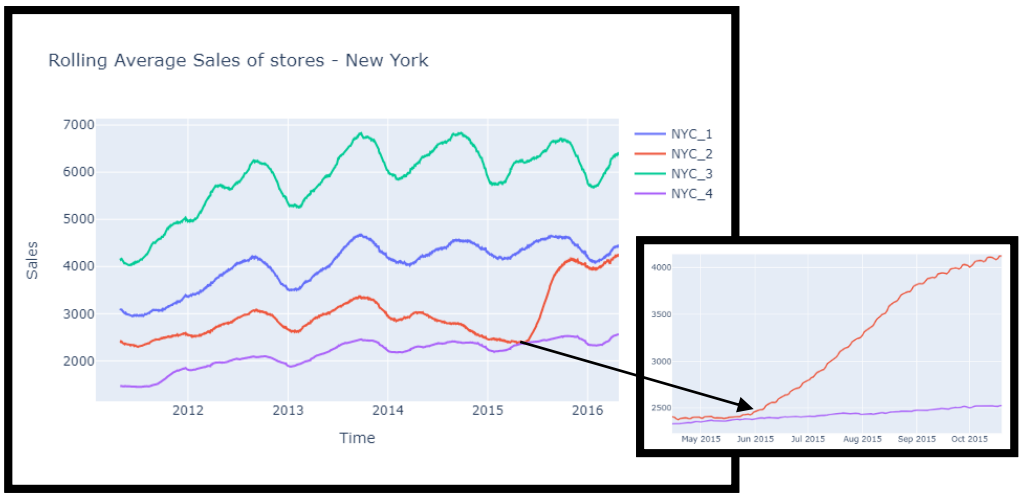
Como podemos examinar, a região de New York predomina tanto no maior número de vendas por loja como no menor. As duas lojas que apresentam o maior número de vendas são NYC_3 e NYC_1 respetivamente, enquanto a que apresenta uma pior performance é NYC_4. Repare-se também que as lojas tanto de Philadelphia como de Boston apresentam número de vendas semelhantes.

Médias moveis de vendas por loja – Boston



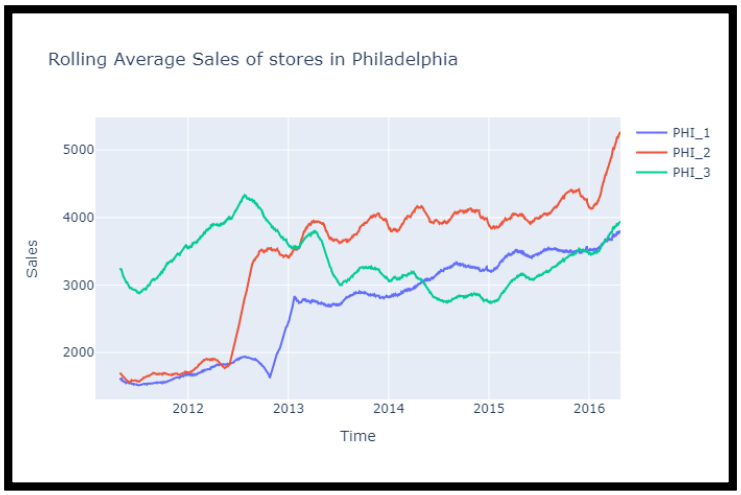
Na região de Boston a loja BOS_2 teve a melhor performance e a BOS_1 a pior. A séries de vendas BOS_2 e BOS_1 apresentam comportamentos parecidos no que toca à tendência , no entanto focando apenas nas series de BOS_1 e BOS_3, estas apresentam comportamentos quase idênticos nos primeiros 3 anos, apresentando algumas vezes o mesmo montante de vendas. Repare-se também que a série BOS_3 atinge a melhor performance de vendas no início de maio.

Médias moveis de vendas por loja – New York



Na região de New York, tal como referido anteriormente a loja NYC_1 e a a loja NYC_4 apresentam a melhor e a pior performance respetivamente. Verifica-se também que os comportamentos/tendências das vendas das 4 lojas são muito semelhantes, com a exceção da loja NYC_2 que em meados de junho de 2015 tem um comportamento inesperado e num espaço de 5 meses quase consegue duplicar as vendas, o que corresponde a aumento na ordem dos 80%-100%

Médias moveis de vendas por loja – Philadelphia



Na região de Philadelphia, não existe evidência de tendência nas séries das lojas. A loja PHI_1 apresenta a melhor performance de vendas nos primeiros dois anos, mas, no entanto, apresenta um decrescimento nos dois anos seguintes. Em 2015, apresenta novamente um crescimento. No momento inicial PHI_2 e PHI_3 apresentam vendas semelhantes. No entanto, PHI_2 apresenta um crescimento exponencial a meio de 2012, seguindo com uma tendência crescente, enquanto PHI_1 apresentou uma tendência alta crescente no fim de 2012 até ao inicio de 2013, e depois seguiu com uma tendência crescente.

Vendas totais por categorias

Categoria ACCESORIES:

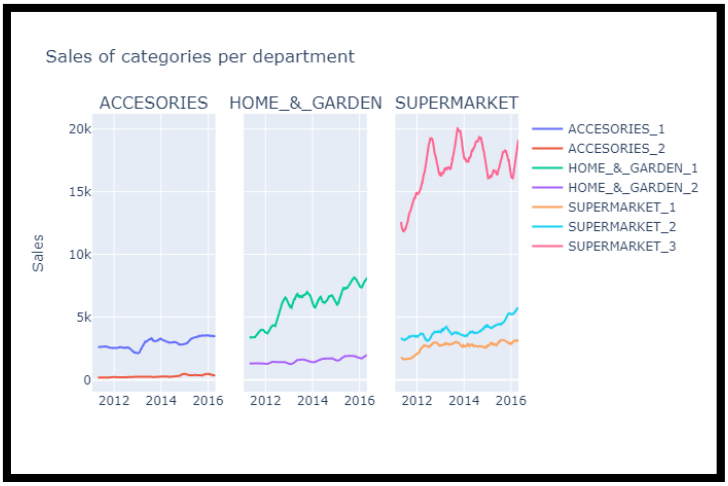
- ACESSORIES_1 apresenta um nível de vendas superior ao ACESSORIES_2
- ACESSORIES_2 apresenta um comportamento constante e a tender para zero

Categoria HOME_&_GARDEN:

- HOME_&_GARDEN_2 apresenta um crescimento exponencial de vendas e superior ao HOME_&_GARDEN_1
- HOME_&_GARDEN_1 apresenta um comportamento constante e com pouca variabilidade no que toca ao número de vendas realizado

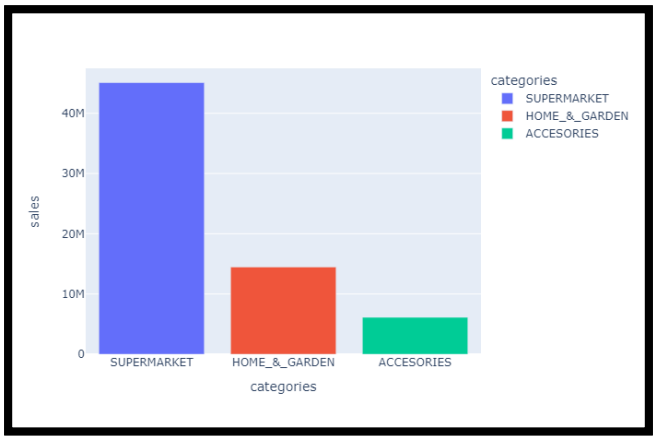
Categoria SUPERMARKET:

- Todos os departamentos do SUPERMARKET apresentam um comportamento crescente ao longo dos anos
- SUPERMARKET_2 e SUPERMARKET_1 apresentam um comportamento semelhante e com pouco variabilidade
- SUPERMARKET_ 3 é o departamento que apresenta uma maior taxa de crescimento, no entanto também apresenta a maior variabilidade nas vendas



Observações:

A categoria **SUPERMARKET** apresenta o maior número de vendas e a categoria ACESSORIES o menor



Categoria ACCESORIES:

- ACESSORIES_1 apresenta um nível de vendas superior ao ACESSORIES_2
- ACESSORIES_2 apresenta um comportamento constante e a tender para zero

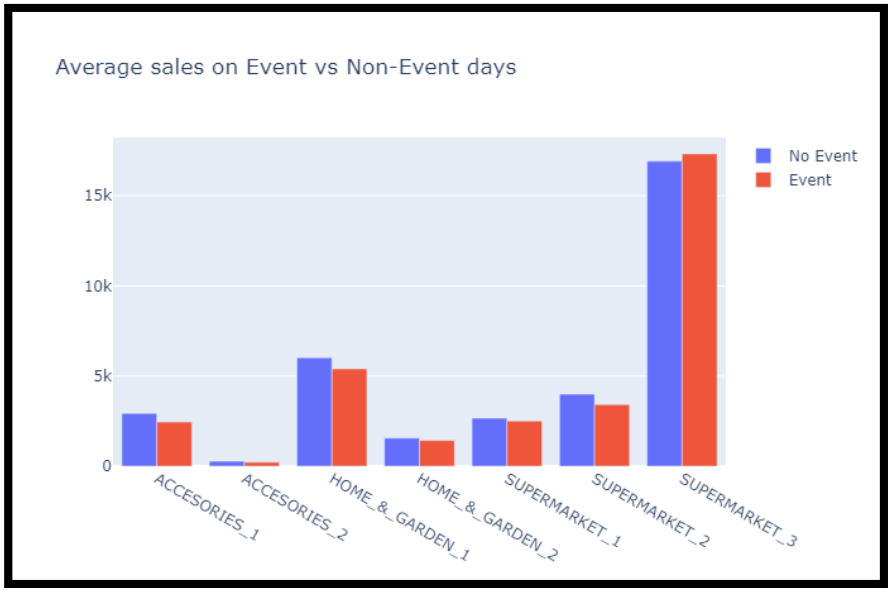
Categoria HOME_&_GARDEN:

- HOME_&_GARDEN_2 apresenta um crescimento exponencial de vendas e superior ao HOME_&_GARDEN_1
- HOME_&_GARDEN_1 apresenta um comportamento constante e com pouca variabilidade no que toca ao número de vendas realizado

Categoria SUPERMARKET:

- Todos os departamentos do SUPERMARKET apresentam um comportamento crescente ao longo dos anos
- SUPERMARKET_2 e SUPERMARKET_1 apresentam um comportamento semelhante e com pouco variabilidade
- SUPERMARKET_ 3 é o departamento que apresenta uma maior taxa de crescimento, no entanto também apresenta a maior variabilidade nas vendas

Média de Vendas em dias de Evento vs Não Evento

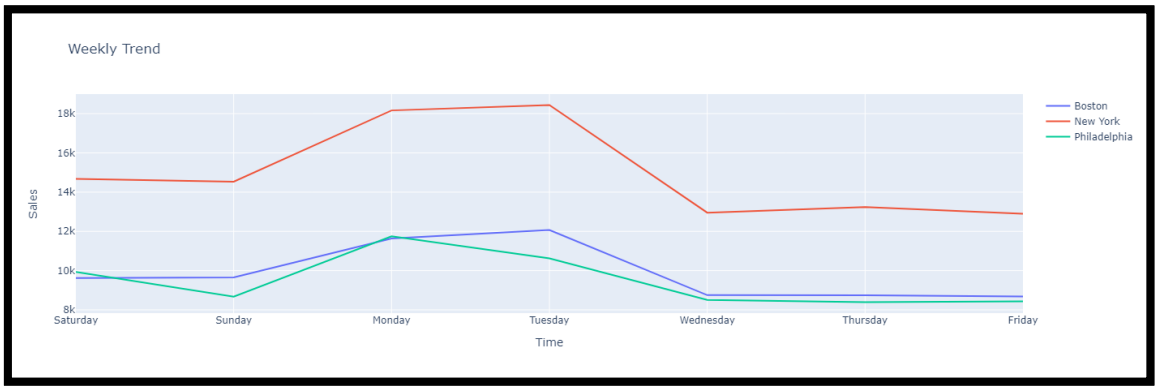


Observando a média das vendas diárias que ocorre evento e que não ocorre conseguimos constatar que a maioria das categorias dos produtos apresenta uma média de vendas superior em dias que não ocorre evento. No entanto, verifica-se também uma exceção. Na categoria SUPERMARKET_3 a média das vendas diárias é superior em dias que ocorrem eventos.

Análise da sazonalidade

Vamos agora fazer uma análise de sazonalidade semanal e mensal, para entendermos o comportamento das vendas nos diferentes períodos temporais.

Sazonalidade semanal por região



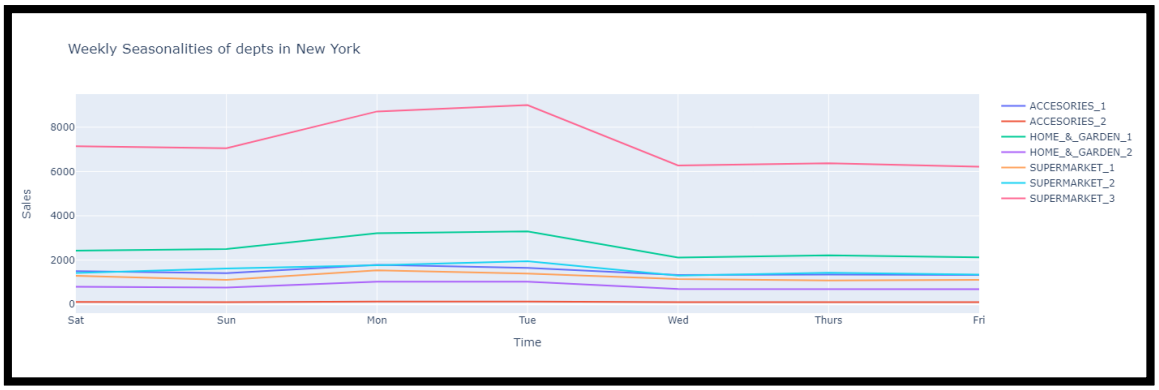
A tendência é semelhante nas três regiões, os momentos em que ocorrem o maior número de vendas são no começo da semana, segunda-feira e terça-feira.

Sazonalidade mensal por região

Tal como se pode constatar, as maiores vendas ocorrem no período de agosto, sendo visível tambem pequenas quedas no mês de novembro e maio.

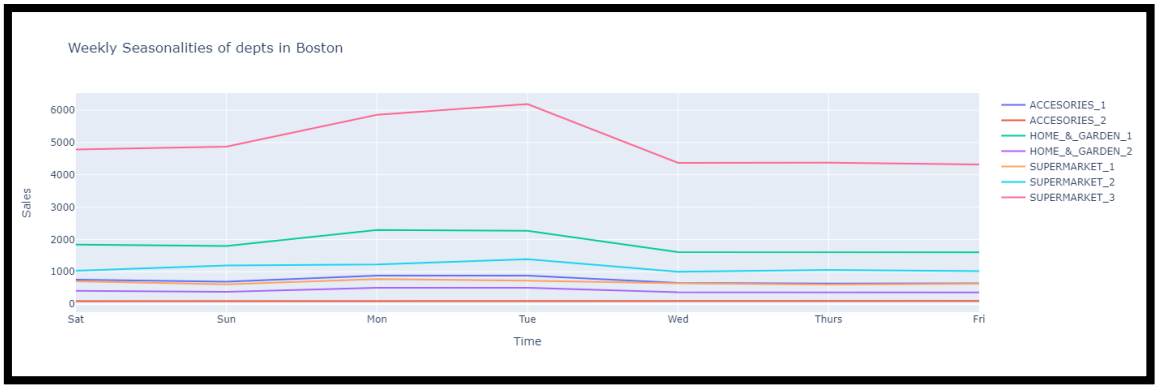
Sazonalidade semanal por categoria e região

Região de New York



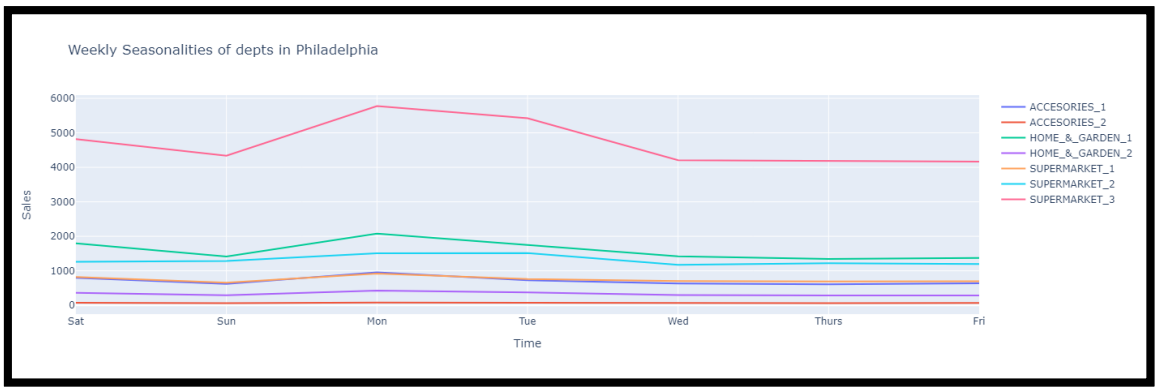
Aparentemente o maior pico de vendas entre todos os departamentos acontece entre segunda-feira e terça-feira, sendo o departamento com mais vendas o SUPERMARKET_3 e com menos vendas o ACCESORIES_2.

Região de Boston



A análise da região de Boston é análoga ao de New York, no entanto embora o comportamento das séries seja semelhante, o número de vendas alcançado por cada categoria é diferente.

Região de Philadelphia



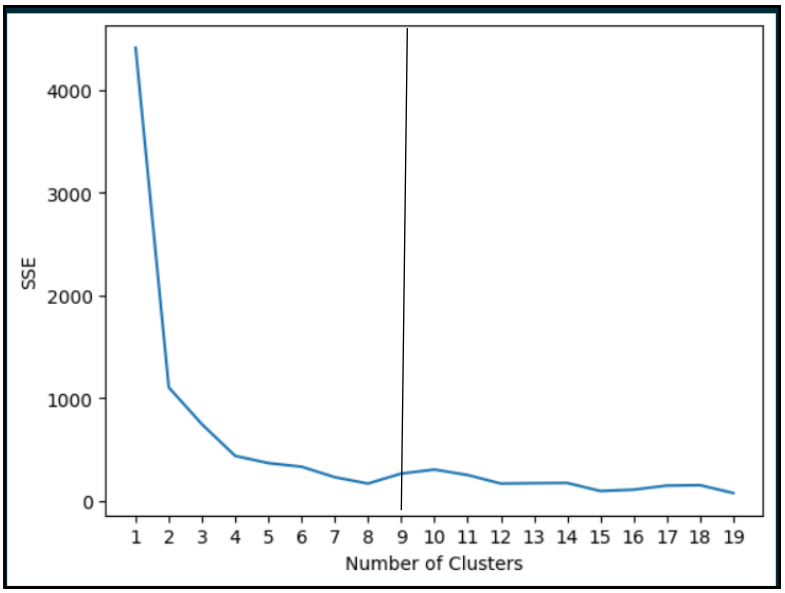
A análise da região de Philadelphia é análoga aos anteriores.

Em suma, verificamos que as tendências semanais para todas as regiões e departamentos foram praticamente idênticas. Verificou-se também quedas nas vendas no mês de maio em todas as regiões, independente do departamento ou região em questão. Por fim, Boston e New York apresentaram os melhores resultados de vendas, verificados em agosto e Philadelphia apresentou a maior quantidade de vendas em fevereiro.

Clustering – Tarefa 2

Clustering dos departamentos por loja mensal

Na análise de clusters, o método Elbow é utilizado para determinar o número de clusters num dataset. Para esboçar o gráfico representado em baixo, recorremos ao k-means, que gerou clusters de 1 a 20.



Assim sendo, iremos visualizar os 4 clusters gerados que demonstra a semelhança entre os departamentos de cada loja.

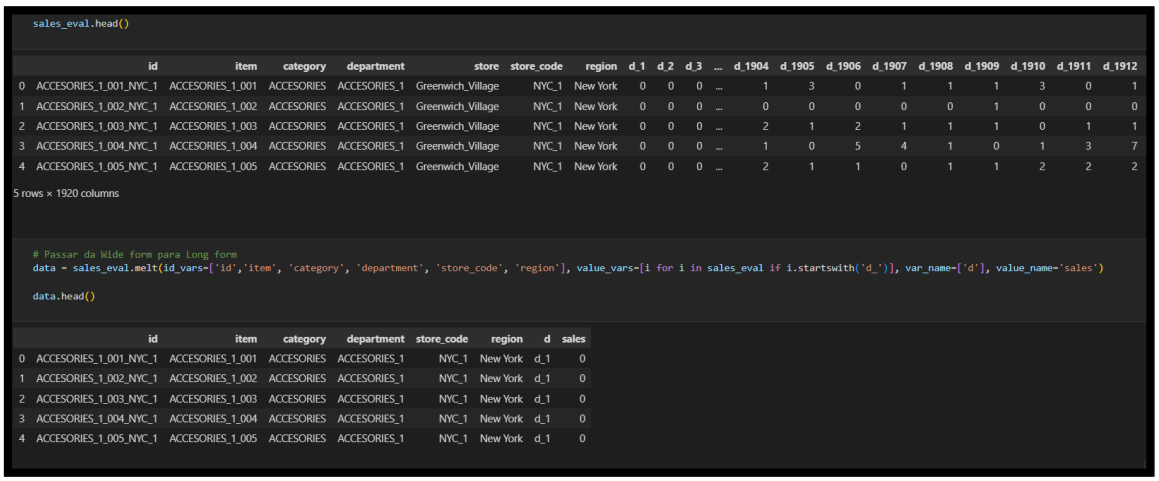
Como podemos ver no cluster 0 as vendas estão compreendidas de 40000 a 100000. Em contraste, temos o cluster 3, onde o intervalo de vendas está entre o período de 0 a 20000.

Feature engineering

Agora que foi realizado o clustering, iremos começar por aplicar a feature engineering. O grande objetivo da feature engineering é a preparação do ficheiro para a fase de modelação. Deste modo começaremos por passar o dataframe da wide form para a long form de modo que consigamos trabalhar melhor com os dados e aplicar os encodings que achemos necessários.

Wide form para Long form

Passar o dataframe da wide form para a long form consiste em converter todas as vendas das colunas das datas para linhas, e para cada linha obter as vendas na mesma data exata, como podemos observar na figura apresenta em baixo.



```
sales_eval.head()
```

	id	item	category	department	store	store_code	region	d_1	d_2	d_3	...	d_1904	d_1905	d_1906	d_1907	d_1908	d_1909	d_1910	d_1911	d_1912
0	ACCESORIES_1_001_NYC_1	ACCESORIES_1_001	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...	1	3	0	1	1	1	3	0	1
1	ACCESORIES_1_002_NYC_1	ACCESORIES_1_002	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...	0	0	0	0	0	1	0	0	0
2	ACCESORIES_1_003_NYC_1	ACCESORIES_1_003	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...	2	1	2	1	1	1	0	1	1
3	ACCESORIES_1_004_NYC_1	ACCESORIES_1_004	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...	1	0	5	4	1	0	1	3	7
4	ACCESORIES_1_005_NYC_1	ACCESORIES_1_005	ACCESORIES	ACCESORIES_1	Greenwich_Village	NYC_1	New York	0	0	0	...	2	1	1	0	1	1	2	2	2

5 rows × 1920 columns

```
# Passar da Wide form para Long form
data = sales_eval.melt(id_vars=['id','item', 'category', 'department', 'store_code', 'region'], value_vars=[i for i in sales_eval if i.startswith('d_')], var_name=['d'], value_name='sales')

data.head()
```

	id	item	category	department	store_code	region	d	sales
0	ACCESORIES_1_001_NYC_1	ACCESORIES_1_001	ACCESORIES	ACCESORIES_1	NYC_1	New York	d_1	0
1	ACCESORIES_1_002_NYC_1	ACCESORIES_1_002	ACCESORIES	ACCESORIES_1	NYC_1	New York	d_1	0
2	ACCESORIES_1_003_NYC_1	ACCESORIES_1_003	ACCESORIES	ACCESORIES_1	NYC_1	New York	d_1	0
3	ACCESORIES_1_004_NYC_1	ACCESORIES_1_004	ACCESORIES	ACCESORIES_1	NYC_1	New York	d_1	0
4	ACCESORIES_1_005_NYC_1	ACCESORIES_1_005	ACCESORIES	ACCESORIES_1	NYC_1	New York	d_1	0

Label Encoding

As variáveis categóricas não são usualmente um grande desafio com que se lidar, uma vez que o Scikit-learn oferece funções simples tais como:

- LabelEncoder
- OneHotEncoder

- OrdinalEncoder

Estas funções podem transformar categorias em variáveis numéricas, e, portanto, em variáveis binárias que os modelos machine learning lidam de forma simples. Contudo, se o número de categorias com que se tem que lidar é grande, a utilização do One Hot Encoding não é aconselhada, uma vez que a transformação irá gerar bastantes zeros.

No nosso caso, poderíamos ter utilizado o One Hot Encoding, uma vez que temos poucas variáveis categóricas. No entanto decidimos que iremos transformar as variáveis categóricas em numéricas, através da aplicação do label encoding.

```
cols = data.dtypes.index.tolist()
d_types = data.dtypes.values.tolist()

for i,type in enumerate(d_types):
    if type.name == 'category':
        data[cols[i]] = data[cols[i]].cat.codes
```

Preparação do ficheiro para modelação

Como último passo da preparação do ficheiro para a fase de modelação iremos introduzir lags e médias móveis ao dataframe.

```python
#Intoducing lags and rolling features
#lag features

lags = [1,2,3,5,7,14,21,28]
for lag in lags:
    data["lag_" + str(lag)] = data.groupby("id")["sales"].shift(lag).astype(np.float16)
```

```python
#rolling mean features

data['rolling_mean_10'] = data.groupby('id')['sales'].transform(lambda x: x.rolling(10).mean())
data['rolling_mean_20'] = data.groupby('id')['sales'].transform(lambda x: x.rolling(20).mean())
data['rolling_mean_30'] = data.groupby('id')['sales'].transform(lambda x: x.rolling(30).mean())
```

```python
data['event'].unique()
```

```
array([-1,  3,  2,  4,  1,  0], dtype=int8)
```

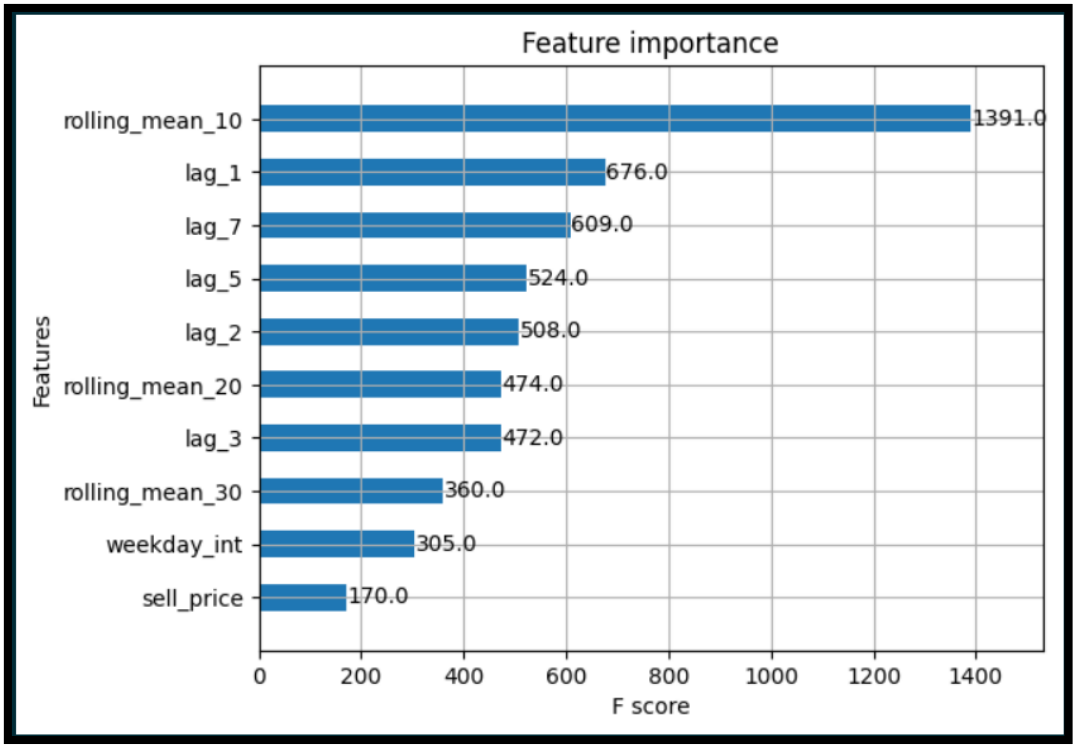
Modelação

A estratégia utilizada para dividir a base de dados foi:

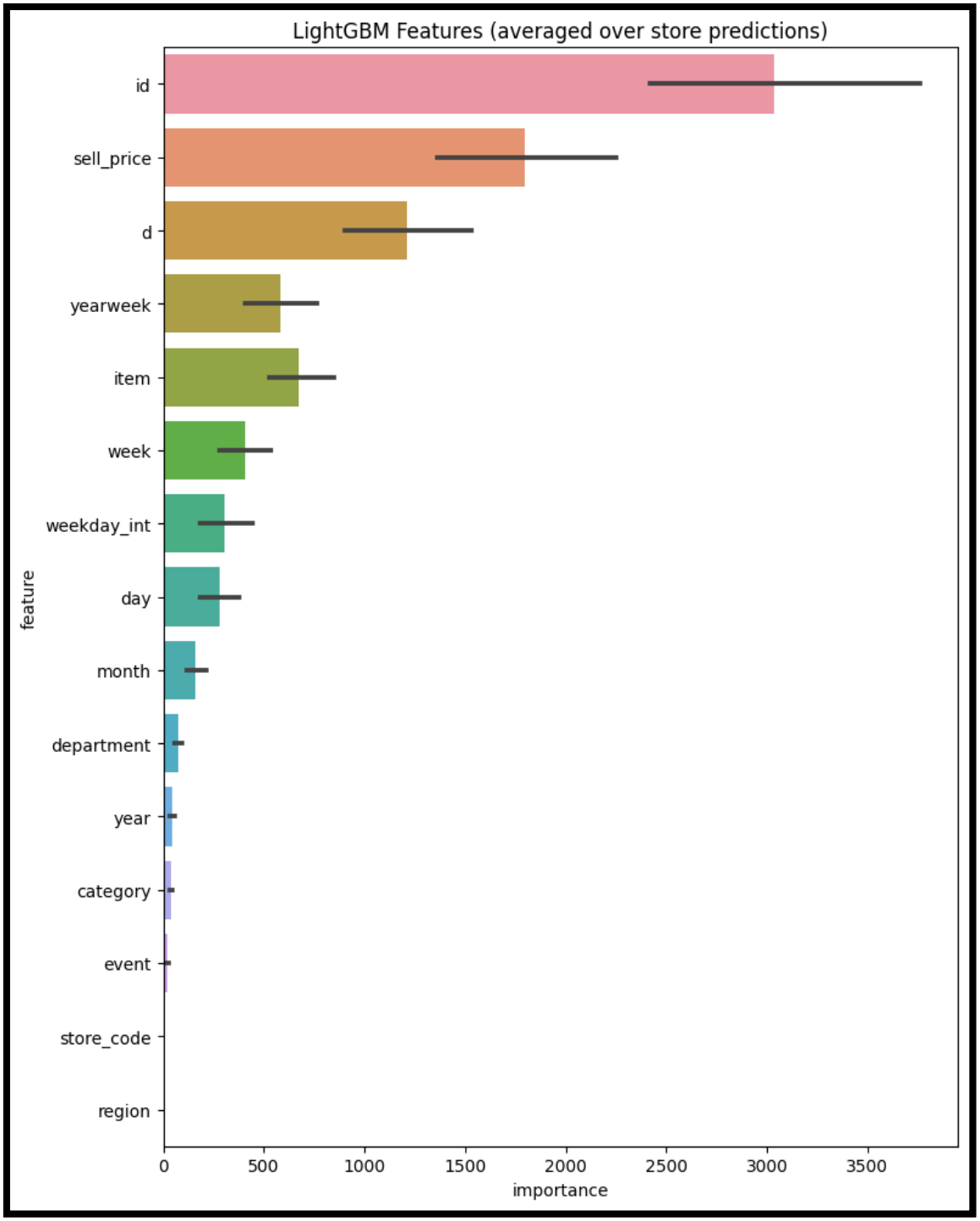
- Train: 60%
- Validation: 20%
- Test: 20%

Aplicou-se os algoritmos XGBoost e LightGBM, onde o XGBoost gerou um único modelo onde unificou todas as lojas e o LighGBM gerou um modelo para cada loja, sendo essencial para a tarefa MLOps. A importância de dividir em lojas deve-se ao facto de ser pretendido fazer previsão de stock por produto, loja e região.

Feature Importance XGBoost



Feature importance LightGBM


LightGBM Features (averaged over store predictions)
id
sell_price
d
yearweek
item
week
weekday_int
day
month
department
year
category
event
store_code
region
feature
0
500
1000
1500
2000
2500
3000
3500
importance