ct-timeseries-xgboost-transactions

September 27, 2024

0.1 Import libraries and load the datasets

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb
from sklearn.model_selection import train_test_split, TimeSeriesSplit
from sklearn.metrics import accuracy_score, roc_auc_score, roc_curve,␣
↪mean_squared_error, mean_absolute_error, r2_score
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler
from datetime import datetime
import calendar
import warnings
from tqdm import tqdm
import plotly.express as px
from sklearn.linear_model import LinearRegression
from datetime import datetime
```

[2]:
```
train_dataset = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/
↪train.csv', parse_dates=['date'])
test_dataset  = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/test.
↪csv', parse_dates=['date'])
store_dataset = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/
↪stores.csv')
oil_dataset   = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/oil.
↪csv',parse_dates=['date'])
holiday_dataset  = pd.read_csv('C:/Users/User/OneDrive - Universidad␣
↪Internacional del Ecuador/Escritorio/Master Primer Semestre/Software for IA/
↪Project 1/holidays_events.csv', parse_dates=['date'])
transactions_dataset = pd.read_csv('C:/Users/User/OneDrive - Universidad␣
↪Internacional del Ecuador/Escritorio/Master Primer Semestre/Software for IA/
↪Project 1/transactions.csv', parse_dates=['date'])
```

0.2 Now it's time to check the train dataset.

[3]:
```
train_dataset.head()
```

[3]:
```
   id        date  store_nbr      family  sales  onpromotion
0   0 2013-01-01          1  AUTOMOTIVE    0.0            0
1   1 2013-01-01          1   BABY CARE    0.0            0
2   2 2013-01-01          1      BEAUTY    0.0            0
3   3 2013-01-01          1   BEVERAGES    0.0            0
4   4 2013-01-01          1       BOOKS    0.0            0
```

[4]:
```
train_dataset.isna().sum()
```

[4]:
```
id             0
date           0
store_nbr      0
family         0
sales          0
onpromotion    0
dtype: int64
```

[5]:
```
train_dataset.shape
```

[5]:
```
(3000888, 6)
```

[6]:
```
train_dataset.describe()
```

[6]:
```
                 id                           date     store_nbr  \
count  3.000888e+06                        3000888  3.000888e+06
mean   1.500444e+06  2015-04-24 08:27:04.703088384  2.750000e+01
min    0.000000e+00            2013-01-01 00:00:00  1.000000e+00
25%    7.502218e+05            2014-02-26 18:00:00  1.400000e+01
50%    1.500444e+06            2015-04-24 12:00:00  2.750000e+01
75%    2.250665e+06            2016-06-19 06:00:00  4.100000e+01
max    3.000887e+06            2017-08-15 00:00:00  5.400000e+01
std    8.662819e+05                            NaN  1.558579e+01

              sales   onpromotion
count  3.000888e+06  3.000888e+06
mean   3.577757e+02  2.602770e+00
min    0.000000e+00  0.000000e+00
25%    0.000000e+00  0.000000e+00
50%    1.100000e+01  0.000000e+00
75%    1.958473e+02  0.000000e+00
max    1.247170e+05  7.410000e+02
std    1.101998e+03  1.221888e+01
```

[7]:
```
day1 = train_dataset['date'].min().strftime('%Y-%m-%d')
last_day = train_dataset['date'].max().strftime('%Y-%m-%d')

day1, last_day
```

[7]:
```
('2013-01-01', '2017-08-15')
```

0.3 Now it's time to check the test dataset.

[8]:
```
test_dataset.head()
```

[8]:

	id	date	store_nbr	family	onpromotion
0	3000888	2017-08-16	1	AUTOMOTIVE	0
1	3000889	2017-08-16	1	BABY CARE	0
2	3000890	2017-08-16	1	BEAUTY	2
3	3000891	2017-08-16	1	BEVERAGES	20
4	3000892	2017-08-16	1	BOOKS	0

[9]:
```
test_dataset.isna().sum()
```

[9]:
```
id             0
date           0
store_nbr      0
family         0
onpromotion    0
dtype: int64
```

[10]:
```
test_dataset.shape
```

[10]:
```
(28512, 5)
```

[11]:
```
test_dataset.describe()
```

[11]:

	id	date	store_nbr	onpromotion
count	2.851200e+04	28512	28512.000000	28512.000000
mean	3.015144e+06	2017-08-23 12:00:00	27.500000	6.965383
min	3.000888e+06	2017-08-16 00:00:00	1.000000	0.000000
25%	3.008016e+06	2017-08-19 18:00:00	14.000000	0.000000
50%	3.015144e+06	2017-08-23 12:00:00	27.500000	0.000000
75%	3.022271e+06	2017-08-27 06:00:00	41.000000	6.000000
max	3.029399e+06	2017-08-31 00:00:00	54.000000	646.000000
std	8.230850e+03	NaN	15.586057	20.683952

[12]:
```
test_day1 = test_dataset['date'].min().strftime('%Y-%m-%d')
test_last_day = test_dataset['date'].max().strftime('%Y-%m-%d')

test_day1, test_last_day
```

[12]: ('2017-08-16', '2017-08-31')

0.4 Now it's time to check the store dataset.

[13]:
```
store_dataset.isna().sum()
```

[13]:
```
store_nbr    0
city         0
state        0
type         0
cluster      0
dtype: int64
```

[14]:
```
store_dataset.shape
```

[14]: (54, 5)

[15]:
```
store_dataset.describe()
```

[15]:

	store_nbr	cluster
count	54.000000	54.000000
mean	27.500000	8.481481
std	15.732133	4.693395
min	1.000000	1.000000
25%	14.250000	4.000000
50%	27.500000	8.500000
75%	40.750000	13.000000
max	54.000000	17.000000

0.5 Now it's time to check the oil dataset.

[16]:
```
oil_dataset.head()
```

[16]:

	date	dcoilwtico
0	2013-01-01	NaN
1	2013-01-02	93.14
2	2013-01-03	92.97
3	2013-01-04	93.12
4	2013-01-07	93.20

[17]:
```
oil_dataset.shape
```

[17]: (1218, 2)

[18]:
```
oil_dataset.isna().sum()
```

[18]:
```
date           0
dcoilwtico    43
dtype: int64
```

[19]:
```
oil_dataset['dcoilwtico'] = oil_dataset['dcoilwtico'].fillna(method='ffill')
oil_dataset['dcoilwtico'] = oil_dataset['dcoilwtico'].fillna(method='bfill')
```

[20]:
```
oil_dataset.isna().sum()
```

[20]:
```
date          0
dcoilwtico    0
dtype: int64
```

[21]:
```
oil_dataset.describe()
```

[21]:

	date	dcoilwtico
count	1218	1218.000000
mean	2015-05-02 12:00:00	67.692159
min	2013-01-01 00:00:00	26.190000
25%	2014-03-03 06:00:00	46.422500
50%	2015-05-02 12:00:00	53.200000
75%	2016-06-30 18:00:00	95.685000
max	2017-08-31 00:00:00	110.620000
std	NaN	25.629744

0.6 Now it's time to check the Holiday dataset.

[22]:
```
holiday_dataset.head()
```

[22]:

	date	type	locale	locale_name	description	transferred
0	2012-03-02	Holiday	Local	Manta	Fundacion de Manta	False
1	2012-04-01	Holiday	Regional	Cotopaxi	Provincializacion de Cotopaxi	False
2	2012-04-12	Holiday	Local	Cuenca	Fundacion de Cuenca	False
3	2012-04-14	Holiday	Local	Libertad	Cantonizacion de Libertad	False
4	2012-04-21	Holiday	Local	Riobamba	Cantonizacion de Riobamba	False

[23]:
```
holiday_dataset.isna().sum()
```

[23]:
```
date           0
type           0
```

```
locale          0
locale_name     0
description     0
transferred     0
dtype: int64
```

[24]:
```
holiday_dataset.shape
```

[24]:
```
(350, 6)
```

[25]:
```
holiday_dataset.describe()
```

[25]:
```
                                date
count                            350
mean   2015-04-24 00:45:15.428571392
min              2012-03-02 00:00:00
25%              2013-12-23 06:00:00
50%              2015-06-08 00:00:00
75%              2016-07-03 00:00:00
max              2017-12-26 00:00:00
```

0.7 Now it's time to check the Transactions dataset.

[26]:
```
transactions_dataset.isna().sum()
```

[26]:
```
date            0
store_nbr       0
transactions    0
dtype: int64
```

[27]:
```
transactions_dataset.shape
```

[27]:
```
(83488, 3)
```

[28]:
```
transactions_dataset.describe()
```

[28]:
```
                                date     store_nbr  transactions
count                          83488  83488.000000  83488.000000
mean   2015-05-20 16:07:40.866232064     26.939237   1694.602158
min              2013-01-01 00:00:00      1.000000      5.000000
25%              2014-03-27 00:00:00     13.000000   1046.000000
50%              2015-06-08 00:00:00     27.000000   1393.000000
75%              2016-07-14 06:00:00     40.000000   2079.000000
max              2017-08-15 00:00:00     54.000000   8359.000000
std                              NaN     15.608204    963.286644
```

```
[29]: train = train_dataset.copy()
      stores = train.groupby(['date', 'store_nbr'], as_index=False)['sales'].sum()
```

```
[30]: train.shape
```

```
[30]: (3000888, 6)
```

```
[31]: px.line(stores, x = "date", y= "sales", color = "store_nbr", title = "Daily␣
       ↪total sales of the stores")
```

It can be observed that from April 16-17, 2016, sales grew significantly, due to the earthquake that struck Ecuador on that date. That is why later this data will be removed. Also, there are stores that were opened after 2013, others since 2015 and so on, so everything before those dates should be removed.

```
[32]: train = train[~((train.store_nbr == 52) & (train.date < "2017-04-20"))]
      train = train[~((train.store_nbr == 22) & (train.date < "2015-10-09"))]
      train = train[~((train.store_nbr == 42) & (train.date < "2015-08-21"))]
      train = train[~((train.store_nbr == 21) & (train.date < "2015-07-24"))]
      train = train[~((train.store_nbr == 29) & (train.date < "2015-03-20"))]
      train = train[~((train.store_nbr == 20) & (train.date < "2015-02-13"))]
      train = train[~((train.store_nbr == 53) & (train.date < "2014-05-29"))]
      train = train[~((train.store_nbr == 36) & (train.date < "2013-05-09"))]
```

```
[33]: start_date = '2016-04-16'
      end_date = '2016-05-02'

      filtered_data = train[(train['date'] >= start_date) & (train['date'] <=␣
       ↪end_date)] # Filter the data in the date range.
      mean_sales_by_class = filtered_data.groupby('store_nbr')['sales'].mean().
       ↪reset_index() # Group by class and calculate the average
      mean_sales_by_class.rename(columns={'sales': 'mean_sales'}, inplace=True) #␣
       ↪Rename sales column for union
      train = train.merge(mean_sales_by_class, on='store_nbr', how='left')# Join the␣
       ↪DataFrames and replace the values
      train.loc[(train['date'] >= start_date) & (train['date'] <= end_date), 'sales']␣
       ↪= train['mean_sales']
      train.drop('mean_sales', axis=1, inplace=True)
```

```
[34]: stores2 = train.groupby(['date', 'store_nbr'], as_index=False)['sales'].sum()
      px.line(stores2, x = "date", y= "sales", color = "store_nbr", title = "Daily␣
       ↪total sales of the stores")
```

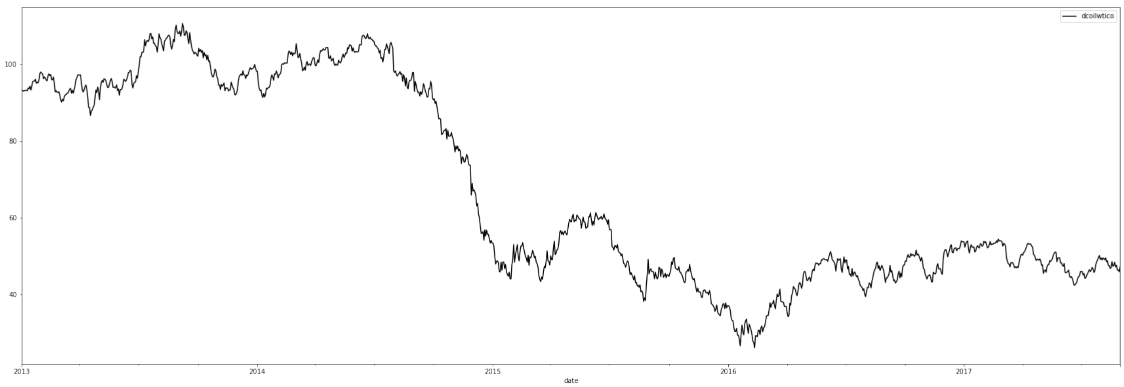
```
[35]: train.shape
```

```
[35]: (2780316, 6)
```

Let's check what happens with the oil

[36]:
```python
oil_dataset.set_index('date').plot(figsize = (30,10),color='black')
```

[36]: <AxesSubplot:xlabel='date'>



[37]:
```python
oil_dataset['date'] = pd.to_datetime(oil_dataset['date'])
oil_dataset.set_index('date', inplace=True)
```

[38]:
```python
train_store = pd.merge(train, store_dataset, on='store_nbr', how='left')
test_store = pd.merge(test_dataset, store_dataset, on='store_nbr', how='left')
```

[39]:
```python
train_store
```

[39]:
```
              id        date  store_nbr                      family     sales  \
0              0  2013-01-01          1                  AUTOMOTIVE     0.000
1              1  2013-01-01          1                   BABY CARE     0.000
2              2  2013-01-01          1                      BEAUTY     0.000
3              3  2013-01-01          1                   BEVERAGES     0.000
4              4  2013-01-01          1                       BOOKS     0.000
…            …         …          …                             …         …
2780311  3000883  2017-08-15          9                     POULTRY   438.133
2780312  3000884  2017-08-15          9              PREPARED FOODS   154.553
2780313  3000885  2017-08-15          9                     PRODUCE  2419.729
2780314  3000886  2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
2780315  3000887  2017-08-15          9                     SEAFOOD    16.000

         onpromotion   city      state type  cluster
0                  0  Quito  Pichincha    D       13
1                  0  Quito  Pichincha    D       13
2                  0  Quito  Pichincha    D       13
3                  0  Quito  Pichincha    D       13
4                  0  Quito  Pichincha    D       13
…                  …      …          …    …        …
```

```
2780311            0  Quito  Pichincha    B        6
2780312            1  Quito  Pichincha    B        6
2780313          148  Quito  Pichincha    B        6
2780314            8  Quito  Pichincha    B        6
2780315            0  Quito  Pichincha    B        6

[2780316 rows x 10 columns]
```

[40]:
```
test_store
```

[40]:
```
            id        date  store_nbr                      family  onpromotion  \
0      3000888  2017-08-16          1                  AUTOMOTIVE            0
1      3000889  2017-08-16          1                   BABY CARE            0
2      3000890  2017-08-16          1                      BEAUTY            2
3      3000891  2017-08-16          1                   BEVERAGES           20
4      3000892  2017-08-16          1                       BOOKS            0
…          …           …         …                          …            …
28507  3029395  2017-08-31          9                     POULTRY            1
28508  3029396  2017-08-31          9              PREPARED FOODS            0
28509  3029397  2017-08-31          9                     PRODUCE            1
28510  3029398  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9
28511  3029399  2017-08-31          9                     SEAFOOD            0

        city      state type  cluster
0      Quito  Pichincha    D       13
1      Quito  Pichincha    D       13
2      Quito  Pichincha    D       13
3      Quito  Pichincha    D       13
4      Quito  Pichincha    D       13
…          …          …    …        …
28507  Quito  Pichincha    B        6
28508  Quito  Pichincha    B        6
28509  Quito  Pichincha    B        6
28510  Quito  Pichincha    B        6
28511  Quito  Pichincha    B        6

[28512 rows x 9 columns]
```

[41]:
```
train_oil = pd.merge(train_store, oil_dataset, on='date', how='left')
test_oil = pd.merge(test_store, oil_dataset, on='date', how='left')
```

[42]:
```
train_oil
```

[42]:
```
            id        date  store_nbr                      family     sales  \
0            0  2013-01-01          1                  AUTOMOTIVE     0.000
1            1  2013-01-01          1                   BABY CARE     0.000
2            2  2013-01-01          1                      BEAUTY     0.000
```

```
3                3 2013-01-01          1                  BEVERAGES     0.000
4                4 2013-01-01          1                      BOOKS     0.000
…              …        …        …                         …        …
2780311  3000883 2017-08-15          9                    POULTRY   438.133
2780312  3000884 2017-08-15          9             PREPARED FOODS   154.553
2780313  3000885 2017-08-15          9                    PRODUCE  2419.729
2780314  3000886 2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
2780315  3000887 2017-08-15          9                    SEAFOOD    16.000

         onpromotion   city      state type  cluster  dcoilwtico
0                  0  Quito  Pichincha    D       13       93.14
1                  0  Quito  Pichincha    D       13       93.14
2                  0  Quito  Pichincha    D       13       93.14
3                  0  Quito  Pichincha    D       13       93.14
4                  0  Quito  Pichincha    D       13       93.14
…              …    …        …  …      …        …
2780311            0  Quito  Pichincha    B        6       47.57
2780312            1  Quito  Pichincha    B        6       47.57
2780313          148  Quito  Pichincha    B        6       47.57
2780314            8  Quito  Pichincha    B        6       47.57
2780315            0  Quito  Pichincha    B        6       47.57

[2780316 rows x 11 columns]
```

[43]:
```python
train_transactions = pd.merge(train_oil, transactions_dataset,␣
 ↪on=['date','store_nbr'],
                                         how='left')
test_transactions = pd.merge(test_oil, transactions_dataset,␣
 ↪on=['date','store_nbr'],
                                         how='left')
```

[44]:
```python
train_transactions.isna().sum()
```

[44]:
```
id                   0
date                 0
store_nbr            0
family               0
sales                0
onpromotion          0
city                 0
state                0
type                 0
cluster              0
dcoilwtico      794211
transactions     25212
dtype: int64
```

[45]:
```
train_transactions['dcoilwtico'].fillna(0, inplace=True)
train_transactions['transactions'].fillna(0, inplace=True)
```

[46]:
```
train_transactions.isna().sum()
```

[46]:
```
id              0
date            0
store_nbr       0
family          0
sales           0
onpromotion     0
city            0
state           0
type            0
cluster         0
dcoilwtico      0
transactions    0
dtype: int64
```

[47]:
```
train_transactions
```

[47]:
```
              id        date  store_nbr                      family     sales  \
0              0  2013-01-01          1                  AUTOMOTIVE     0.000
1              1  2013-01-01          1                   BABY CARE     0.000
2              2  2013-01-01          1                      BEAUTY     0.000
3              3  2013-01-01          1                   BEVERAGES     0.000
4              4  2013-01-01          1                       BOOKS     0.000
…              …           …          …                           …         …
2780311  3000883  2017-08-15          9                     POULTRY   438.133
2780312  3000884  2017-08-15          9              PREPARED FOODS   154.553
2780313  3000885  2017-08-15          9                     PRODUCE  2419.729
2780314  3000886  2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
2780315  3000887  2017-08-15          9                     SEAFOOD    16.000

         onpromotion   city      state type  cluster  dcoilwtico  transactions
0                  0  Quito  Pichincha    D       13       93.14           0.0
1                  0  Quito  Pichincha    D       13       93.14           0.0
2                  0  Quito  Pichincha    D       13       93.14           0.0
3                  0  Quito  Pichincha    D       13       93.14           0.0
4                  0  Quito  Pichincha    D       13       93.14           0.0
…                  …      …          …    …        …           …             …
2780311            0  Quito  Pichincha    B        6       47.57        2155.0
2780312            1  Quito  Pichincha    B        6       47.57        2155.0
2780313          148  Quito  Pichincha    B        6       47.57        2155.0
2780314            8  Quito  Pichincha    B        6       47.57        2155.0
2780315            0  Quito  Pichincha    B        6       47.57        2155.0
```

```
[2780316 rows x 12 columns]
```

[48]:
```python
test_transactions['dcoilwtico'].fillna(0, inplace=True)
test_transactions['transactions'].fillna(0, inplace=True)
```

[49]:
```python
test_transactions.isna().sum()
```

[49]:
```
id              0
date            0
store_nbr       0
family          0
onpromotion     0
city            0
state           0
type            0
cluster         0
dcoilwtico      0
transactions    0
dtype: int64
```

[50]:
```python
test_transactions
```

[50]:
```
            id        date  store_nbr                      family  onpromotion  \
0      3000888  2017-08-16          1                  AUTOMOTIVE            0
1      3000889  2017-08-16          1                   BABY CARE            0
2      3000890  2017-08-16          1                      BEAUTY            2
3      3000891  2017-08-16          1                   BEVERAGES           20
4      3000892  2017-08-16          1                       BOOKS            0
…          …         …        …                           …          …
28507  3029395  2017-08-31          9                     POULTRY            1
28508  3029396  2017-08-31          9              PREPARED FOODS            0
28509  3029397  2017-08-31          9                     PRODUCE            1
28510  3029398  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9
28511  3029399  2017-08-31          9                     SEAFOOD            0

        city      state type  cluster  dcoilwtico  transactions
0      Quito  Pichincha    D       13       46.80           0.0
1      Quito  Pichincha    D       13       46.80           0.0
2      Quito  Pichincha    D       13       46.80           0.0
3      Quito  Pichincha    D       13       46.80           0.0
4      Quito  Pichincha    D       13       46.80           0.0
…        …        …  …      …           …             …
28507  Quito  Pichincha    B        6       47.26           0.0
28508  Quito  Pichincha    B        6       47.26           0.0
28509  Quito  Pichincha    B        6       47.26           0.0
28510  Quito  Pichincha    B        6       47.26           0.0
28511  Quito  Pichincha    B        6       47.26           0.0
```

```
[28512 rows x 11 columns]
```

[51]:
```
datatrain2 = train_transactions.copy()
datatest2 = test_transactions.copy()
datatrain2.head()
```

[51]:
```
   id        date  store_nbr      family  sales  onpromotion   city      state  \
0   0 2013-01-01          1  AUTOMOTIVE    0.0            0  Quito  Pichincha
1   1 2013-01-01          1   BABY CARE    0.0            0  Quito  Pichincha
2   2 2013-01-01          1      BEAUTY    0.0            0  Quito  Pichincha
3   3 2013-01-01          1   BEVERAGES    0.0            0  Quito  Pichincha
4   4 2013-01-01          1       BOOKS    0.0            0  Quito  Pichincha

  type  cluster  dcoilwtico  transactions
0    D       13       93.14           0.0
1    D       13       93.14           0.0
2    D       13       93.14           0.0
3    D       13       93.14           0.0
4    D       13       93.14           0.0
```

[52]:
```
datatrain2 = datatrain2.drop('id', axis=1)
datatrain2.head()
```

[52]:
```
        date  store_nbr      family  sales  onpromotion   city      state  \
0 2013-01-01          1  AUTOMOTIVE    0.0            0  Quito  Pichincha
1 2013-01-01          1   BABY CARE    0.0            0  Quito  Pichincha
2 2013-01-01          1      BEAUTY    0.0            0  Quito  Pichincha
3 2013-01-01          1   BEVERAGES    0.0            0  Quito  Pichincha
4 2013-01-01          1       BOOKS    0.0            0  Quito  Pichincha

  type  cluster  dcoilwtico  transactions
0    D       13       93.14           0.0
1    D       13       93.14           0.0
2    D       13       93.14           0.0
3    D       13       93.14           0.0
4    D       13       93.14           0.0
```

[53]:
```
# split data into X parameter and y as target
X = datatrain2.drop('transactions', axis=1)
Y = datatrain2.iloc[:, 10] # Transaction values are saved
Y
```

[53]:
```
0              0.0
1              0.0
2              0.0
3              0.0
```

```
4             0.0
             …
2780311    2155.0
2780312    2155.0
2780313    2155.0
2780314    2155.0
2780315    2155.0
Name: transactions, Length: 2780316, dtype: float64
```

[54]:
```
Y.shape
```

[54]:
```
(2780316,)
```

[55]:
```
X['date'] = pd.to_datetime(X['date'])
onehot_label = ['family', 'store_nbr','city','state','type']
```

[56]:
```
onehot_encoder = OneHotEncoder(sparse=False)
onehot_encoder
```

[56]:
```
OneHotEncoder(sparse=False)
```

[57]:
```
X_1 = onehot_encoder.fit_transform(X[onehot_label])
X_1
```

```
C:\Users\User\anaconda3\lib\site-
packages\sklearn\preprocessing\_encoders.py:975: FutureWarning:

`sparse` was renamed to `sparse_output` in version 1.2 and will be removed in
1.4. `sparse_output` is ignored unless you leave `sparse` to its default value.
```

[57]:
```
array([[1., 0., 0., …, 0., 1., 0.],
       [0., 1., 0., …, 0., 1., 0.],
       [0., 0., 1., …, 0., 1., 0.],
       …,
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.]])
```

[58]:
```
feature_names = onehot_encoder.get_feature_names_out(onehot_label)
feature_names
```

[58]:
```
array(['family_AUTOMOTIVE', 'family_BABY CARE', 'family_BEAUTY',
       'family_BEVERAGES', 'family_BOOKS', 'family_BREAD/BAKERY',
       'family_CELEBRATION', 'family_CLEANING', 'family_DAIRY',
       'family_DELI', 'family_EGGS', 'family_FROZEN FOODS',
       'family_GROCERY I', 'family_GROCERY II', 'family_HARDWARE',
```

```
       'family_HOME AND KITCHEN I', 'family_HOME AND KITCHEN II',
       'family_HOME APPLIANCES', 'family_HOME CARE', 'family_LADIESWEAR',
       'family_LAWN AND GARDEN', 'family_LINGERIE',
       'family_LIQUOR,WINE,BEER', 'family_MAGAZINES', 'family_MEATS',
       'family_PERSONAL CARE', 'family_PET SUPPLIES',
       'family_PLAYERS AND ELECTRONICS', 'family_POULTRY',
       'family_PREPARED FOODS', 'family_PRODUCE',
       'family_SCHOOL AND OFFICE SUPPLIES', 'family_SEAFOOD',
       'store_nbr_1', 'store_nbr_2', 'store_nbr_3', 'store_nbr_4',
       'store_nbr_5', 'store_nbr_6', 'store_nbr_7', 'store_nbr_8',
       'store_nbr_9', 'store_nbr_10', 'store_nbr_11', 'store_nbr_12',
       'store_nbr_13', 'store_nbr_14', 'store_nbr_15', 'store_nbr_16',
       'store_nbr_17', 'store_nbr_18', 'store_nbr_19', 'store_nbr_20',
       'store_nbr_21', 'store_nbr_22', 'store_nbr_23', 'store_nbr_24',
       'store_nbr_25', 'store_nbr_26', 'store_nbr_27', 'store_nbr_28',
       'store_nbr_29', 'store_nbr_30', 'store_nbr_31', 'store_nbr_32',
       'store_nbr_33', 'store_nbr_34', 'store_nbr_35', 'store_nbr_36',
       'store_nbr_37', 'store_nbr_38', 'store_nbr_39', 'store_nbr_40',
       'store_nbr_41', 'store_nbr_42', 'store_nbr_43', 'store_nbr_44',
       'store_nbr_45', 'store_nbr_46', 'store_nbr_47', 'store_nbr_48',
       'store_nbr_49', 'store_nbr_50', 'store_nbr_51', 'store_nbr_52',
       'store_nbr_53', 'store_nbr_54', 'city_Ambato', 'city_Babahoyo',
       'city_Cayambe', 'city_Cuenca', 'city_Daule', 'city_El Carmen',
       'city_Esmeraldas', 'city_Guaranda', 'city_Guayaquil',
       'city_Ibarra', 'city_Latacunga', 'city_Libertad', 'city_Loja',
       'city_Machala', 'city_Manta', 'city_Playas', 'city_Puyo',
       'city_Quevedo', 'city_Quito', 'city_Riobamba', 'city_Salinas',
       'city_Santo Domingo', 'state_Azuay', 'state_Bolivar',
       'state_Chimborazo', 'state_Cotopaxi', 'state_El Oro',
       'state_Esmeraldas', 'state_Guayas', 'state_Imbabura', 'state_Loja',
       'state_Los Rios', 'state_Manabi', 'state_Pastaza',
       'state_Pichincha', 'state_Santa Elena',
       'state_Santo Domingo de los Tsachilas', 'state_Tungurahua',
       'type_A', 'type_B', 'type_C', 'type_D', 'type_E'], dtype=object)
```

[59]:
```
X_1 = pd.DataFrame(X_1, columns=feature_names)
X_1
```

[59]:

	family_AUTOMOTIVE	family_BABY CARE	family_BEAUTY	family_BEVERAGES	\
0	1.0	0.0	0.0	0.0	
1	0.0	1.0	0.0	0.0	
2	0.0	0.0	1.0	0.0	
3	0.0	0.0	0.0	1.0	
4	0.0	0.0	0.0	0.0	
…	…	…	…	…	
2780311	0.0	0.0	0.0	0.0	
2780312	0.0	0.0	0.0	0.0	

```
2780313              0.0               0.0            0.0              0.0
2780314              0.0               0.0            0.0              0.0
2780315              0.0               0.0            0.0              0.0

         family_BOOKS  family_BREAD/BAKERY  family_CELEBRATION  \
0                 0.0                  0.0                 0.0
1                 0.0                  0.0                 0.0
2                 0.0                  0.0                 0.0
3                 0.0                  0.0                 0.0
4                 1.0                  0.0                 0.0
…                 …                   …                 …
2780311           0.0                  0.0                 0.0
2780312           0.0                  0.0                 0.0
2780313           0.0                  0.0                 0.0
2780314           0.0                  0.0                 0.0
2780315           0.0                  0.0                 0.0

         family_CLEANING  family_DAIRY  family_DELI  …  state_Pastaza  \
0                    0.0           0.0          0.0  …            0.0
1                    0.0           0.0          0.0  …            0.0
2                    0.0           0.0          0.0  …            0.0
3                    0.0           0.0          0.0  …            0.0
4                    0.0           0.0          0.0  …            0.0
…                    …            …          …  …            …
2780311              0.0           0.0          0.0  …            0.0
2780312              0.0           0.0          0.0  …            0.0
2780313              0.0           0.0          0.0  …            0.0
2780314              0.0           0.0          0.0  …            0.0
2780315              0.0           0.0          0.0  …            0.0

         state_Pichincha  state_Santa Elena  \
0                    1.0                0.0
1                    1.0                0.0
2                    1.0                0.0
3                    1.0                0.0
4                    1.0                0.0
…                    …                …
2780311              1.0                0.0
2780312              1.0                0.0
2780313              1.0                0.0
2780314              1.0                0.0
2780315              1.0                0.0

         state_Santo Domingo de los Tsachilas  state_Tungurahua  type_A  \
0                                         0.0               0.0     0.0
1                                         0.0               0.0     0.0
2                                         0.0               0.0     0.0
```

```
3                                         0.0                0.0      0.0
4                                         0.0                0.0      0.0
…                                       …                …      …
2780311                                   0.0                0.0      0.0
2780312                                   0.0                0.0      0.0
2780313                                   0.0                0.0      0.0
2780314                                   0.0                0.0      0.0
2780315                                   0.0                0.0      0.0

         type_B  type_C  type_D  type_E
0           0.0     0.0     1.0     0.0
1           0.0     0.0     1.0     0.0
2           0.0     0.0     1.0     0.0
3           0.0     0.0     1.0     0.0
4           0.0     0.0     1.0     0.0
…        …     …     …     …
2780311     1.0     0.0     0.0     0.0
2780312     1.0     0.0     0.0     0.0
2780313     1.0     0.0     0.0     0.0
2780314     1.0     0.0     0.0     0.0
2780315     1.0     0.0     0.0     0.0

[2780316 rows x 130 columns]
```

[60]:
```
X = pd.concat([X.drop(onehot_label, axis=1), X_1], axis=1)
```

[61]:
```
X.head()
```

[61]:
```
        date  sales  onpromotion  cluster  dcoilwtico  family_AUTOMOTIVE  \
0 2013-01-01    0.0            0       13       93.14                1.0
1 2013-01-01    0.0            0       13       93.14                0.0
2 2013-01-01    0.0            0       13       93.14                0.0
3 2013-01-01    0.0            0       13       93.14                0.0
4 2013-01-01    0.0            0       13       93.14                0.0

   family_BABY CARE  family_BEAUTY  family_BEVERAGES  family_BOOKS  …  \
0               0.0            0.0               0.0           0.0  …
1               1.0            0.0               0.0           0.0  …
2               0.0            1.0               0.0           0.0  …
3               0.0            0.0               1.0           0.0  …
4               0.0            0.0               0.0           1.0  …

   state_Pastaza  state_Pichincha  state_Santa Elena  \
0            0.0              1.0                0.0
1            0.0              1.0                0.0
2            0.0              1.0                0.0
3            0.0              1.0                0.0
```

```
4           0.0              1.0               0.0

   state_Santo Domingo de los Tsachilas  state_Tungurahua  type_A  type_B  \
0                                   0.0               0.0     0.0     0.0
1                                   0.0               0.0     0.0     0.0
2                                   0.0               0.0     0.0     0.0
3                                   0.0               0.0     0.0     0.0
4                                   0.0               0.0     0.0     0.0

   type_C  type_D  type_E
0     0.0     1.0     0.0
1     0.0     1.0     0.0
2     0.0     1.0     0.0
3     0.0     1.0     0.0
4     0.0     1.0     0.0

[5 rows x 135 columns]
```

[62]:
```
X['date'] = X['date'].astype('int64')
```

[63]:
```
# split data into train and test sets
seed = 42
test_size = 0.20
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=test_size,␣
 ↪random_state=seed)
```

[64]:
```
params = {
    'max_depth': 8,                # Tree depth
    'learning_rate': 0.1,          # Learning rate
    'n_estimators': 100,           # Number of trees
    'subsample': 0.8,              # Proportion of samples for each tree
    'colsample_bytree': 0.8,       # Proportion of features for each tree
    'gamma': 0.1,                  # Minimum stop loss to make a split
    'reg_alpha': 0.1,              # Regularización L1
    'reg_lambda': 1.0,             # Regularización L2
    'objective': 'reg:squarederror',
    'eval_metric': 'rmse'
}

# Create and train the model xgboost
model = xgb.XGBRegressor(**params)

model.fit(X_train, y_train, eval_set=[(X_test, y_test)], verbose = 10)
```

```
[0]     validation_0-rmse:891.82463
[10]    validation_0-rmse:428.20338
[20]    validation_0-rmse:308.44610
```

```
[30]	validation_0-rmse:268.44896
[40]	validation_0-rmse:247.01185
[50]	validation_0-rmse:231.94224
[60]	validation_0-rmse:224.12918
[70]	validation_0-rmse:217.12345
[80]	validation_0-rmse:212.57799
[90]	validation_0-rmse:209.48143
[99]	validation_0-rmse:206.54189
```

[64]:
```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=0.8, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric='rmse', feature_types=None,
             gamma=0.1, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.1, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=8, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=100, n_jobs=None,
             num_parallel_tree=None, random_state=None, …)
```

[66]:
```python
# Make the predictions from the model
y_pred = model.predict(X_test)

# Calculate RMSE y RMSLE
def rmsle(y_true, y_pred):
    return np.sqrt(np.mean(np.square(np.log1p(y_pred) - np.log1p(y_true))))

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
rmsle_score = rmsle(y_test, y_pred)

print("RMSE:", rmse)
print("RMSLE:", rmsle_score)
```

```
RMSE: 206.54188696468438
RMSLE: 0.5695888418033543
```

```
<ipython-input-66-44825693e85c>:6: RuntimeWarning:

invalid value encountered in log1p
```

[67]:
```python
X_test
```

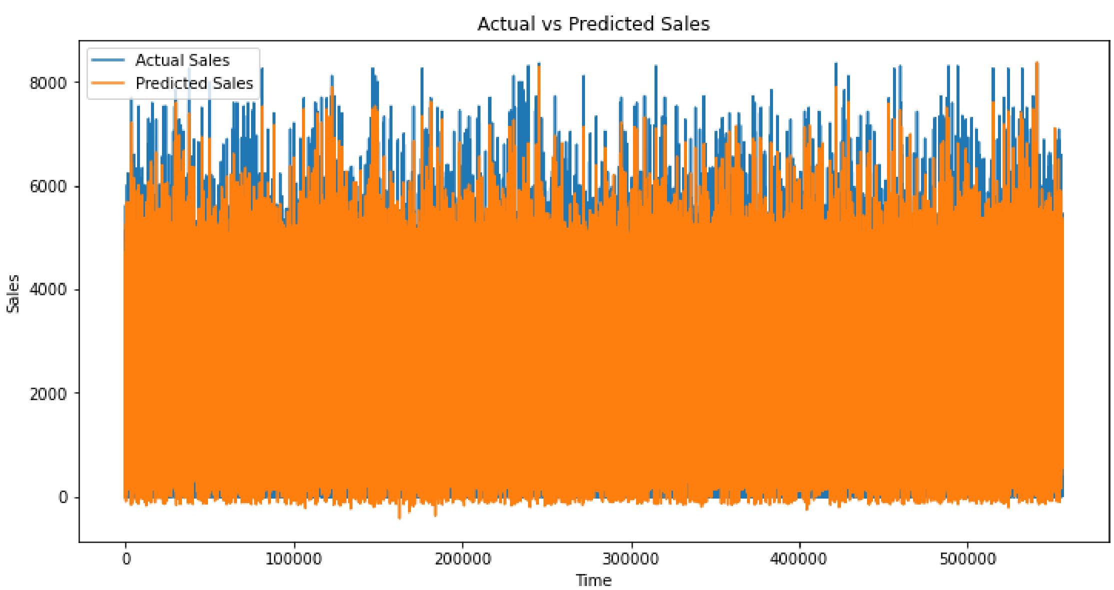
[67]:
```
                       date       sales  onpromotion  cluster  dcoilwtico  \
2751342  1501372800000000000   26.000000            0       14        0.00
1936325  1461196800000000000  335.585504            0        4       43.18
1506823  1439856000000000000   87.000000            0       15       42.58
```

```
1441151  1436400000000000000  1860.000000           1        14        52.76
1602247  1444608000000000000     1.000000           0         3        47.09
…                          …           …          …       …          …
2404350  1484438400000000000  1723.000000          35         7         0.00
1874825  1458172800000000000    33.000000           0         3        40.17
1698747  1449360000000000000    32.000000           0         4         0.00
1213408  1424476800000000000     0.000000           0        16         0.00
227617   1369872000000000000     0.000000           0         3        93.57

         family_AUTOMOTIVE  family_BABY CARE  family_BEAUTY  family_BEVERAGES  \
2751342                1.0               0.0            0.0               0.0
1936325                0.0               0.0            0.0               0.0
1506823                0.0               0.0            0.0               0.0
1441151                0.0               0.0            0.0               0.0
1602247                0.0               0.0            0.0               0.0
…                        …                 …              …                 …
2404350                0.0               0.0            0.0               1.0
1874825                0.0               0.0            0.0               0.0
1698747                0.0               0.0            0.0               0.0
1213408                0.0               0.0            0.0               0.0
227617                 0.0               0.0            0.0               0.0

         family_BOOKS  …  state_Pastaza  state_Pichincha  state_Santa Elena  \
2751342           0.0  …            0.0              1.0                0.0
1936325           0.0  …            0.0              0.0                0.0
1506823           0.0  …            0.0              0.0                0.0
1441151           0.0  …            0.0              1.0                0.0
1602247           0.0  …            0.0              0.0                0.0
…                   …  …              …                …                  …
2404350           0.0  …            1.0              0.0                0.0
1874825           0.0  …            0.0              0.0                0.0
1698747           0.0  …            0.0              0.0                0.0
1213408           0.0  …            0.0              1.0                0.0
227617            0.0  …            0.0              0.0                0.0

         state_Santo Domingo de los Tsachilas  state_Tungurahua  type_A  \
2751342                                   0.0               0.0     1.0
1936325                                   0.0               0.0     0.0
1506823                                   0.0               0.0     0.0
1441151                                   0.0               0.0     1.0
1602247                                   0.0               0.0     0.0
…                                           …                 …       …
2404350                                   0.0               0.0     0.0
1874825                                   0.0               0.0     0.0
1698747                                   1.0               0.0     0.0
1213408                                   0.0               0.0     0.0
227617                                    0.0               0.0     0.0
```

```
         type_B  type_C  type_D  type_E
2751342     0.0     0.0     0.0     0.0
1936325     0.0     0.0     1.0     0.0
1506823     0.0     1.0     0.0     0.0
1441151     0.0     0.0     0.0     0.0
1602247     0.0     1.0     0.0     0.0
…         …     …     …     …
2404350     0.0     1.0     0.0     0.0
1874825     0.0     1.0     0.0     0.0
1698747     0.0     0.0     1.0     0.0
1213408     1.0     0.0     0.0     0.0
227617      0.0     1.0     0.0     0.0

[556064 rows x 135 columns]
```

[68]:

```python
plt.figure(figsize=(12, 6))
plt.plot(y_test.values, label='Actual Sales')
plt.plot(y_pred, label='Predicted Sales')
plt.xlabel('Time')
plt.ylabel('Sales')
plt.title('Actual vs Predicted Sales')
plt.legend()
plt.show()
```



[69]:

```python
y_pred
```

[69]: array([4442.9727, 1060.4113, 1150.1444, …, 1365.796 , 1290.2698,
        1029.6172], dtype=float32)

[70]:
```
y_test
```

[70]:
```
2751342    4305.0
1936325     930.0
1506823    1142.0
1441151    3405.0
1602247    1406.0
             …
2404350     794.0
1874825     933.0
1698747    1312.0
1213408    1178.0
227617     1180.0
Name: transactions, Length: 556064, dtype: float64
```

[71]:
```
datatest2 = test_transactions.copy()
datatest2 = datatest2.drop('id', axis=1)
datatest2 = datatest2.drop('transactions', axis=1)
datatest2.head()
```

[71]:
```
         date  store_nbr      family  onpromotion   city      state type  \
0 2017-08-16          1  AUTOMOTIVE            0  Quito  Pichincha    D
1 2017-08-16          1   BABY CARE            0  Quito  Pichincha    D
2 2017-08-16          1      BEAUTY            2  Quito  Pichincha    D
3 2017-08-16          1   BEVERAGES           20  Quito  Pichincha    D
4 2017-08-16          1       BOOKS            0  Quito  Pichincha    D

   cluster  dcoilwtico
0       13        46.8
1       13        46.8
2       13        46.8
3       13        46.8
4       13        46.8
```

[72]:
```
test_1 = onehot_encoder.transform(datatest2[onehot_label])
test_1 = pd.DataFrame(test_1, columns=feature_names)
datatest2['date'] = pd.to_datetime(datatest2['date'])
datatest2 = pd.concat([datatest2.drop(onehot_label, axis=1), test_1], axis=1)
```

[73]:
```
datatest2['date'] = datatest2['date'].astype('int64')
```

[76]:
```
datatest2
```

[76]:

```
                     date  onpromotion  cluster  dcoilwtico  \
0      1502841600000000000            0       13       46.80
1      1502841600000000000            0       13       46.80
2      1502841600000000000            2       13       46.80
3      1502841600000000000           20       13       46.80
4      1502841600000000000            0       13       46.80
…                      …          …       …          …
28507  1504137600000000000            1        6       47.26
28508  1504137600000000000            0        6       47.26
28509  1504137600000000000            1        6       47.26
28510  1504137600000000000            9        6       47.26
28511  1504137600000000000            0        6       47.26

       family_AUTOMOTIVE  family_BABY CARE  family_BEAUTY  family_BEVERAGES  \
0                    1.0               0.0            0.0               0.0
1                    0.0               1.0            0.0               0.0
2                    0.0               0.0            1.0               0.0
3                    0.0               0.0            0.0               1.0
4                    0.0               0.0            0.0               0.0
…                    …                 …              …                 …
28507                0.0               0.0            0.0               0.0
28508                0.0               0.0            0.0               0.0
28509                0.0               0.0            0.0               0.0
28510                0.0               0.0            0.0               0.0
28511                0.0               0.0            0.0               0.0

       family_BOOKS  family_BREAD/BAKERY  …  state_Pastaza  state_Pichincha  \
0               0.0                  0.0  …            0.0              1.0
1               0.0                  0.0  …            0.0              1.0
2               0.0                  0.0  …            0.0              1.0
3               0.0                  0.0  …            0.0              1.0
4               1.0                  0.0  …            0.0              1.0
…               …                  …  …            …              …
28507           0.0                  0.0  …            0.0              1.0
28508           0.0                  0.0  …            0.0              1.0
28509           0.0                  0.0  …            0.0              1.0
28510           0.0                  0.0  …            0.0              1.0
28511           0.0                  0.0  …            0.0              1.0

       state_Santa Elena  state_Santo Domingo de los Tsachilas  \
0                    0.0                                   0.0
1                    0.0                                   0.0
2                    0.0                                   0.0
3                    0.0                                   0.0
4                    0.0                                   0.0
…                    …                                   …
28507                0.0                                   0.0
```

```
28508                    0.0                                    0.0
28509                    0.0                                    0.0
28510                    0.0                                    0.0
28511                    0.0                                    0.0

       state_Tungurahua  type_A  type_B  type_C  type_D  type_E
0                   0.0     0.0     0.0     0.0     1.0     0.0
1                   0.0     0.0     0.0     0.0     1.0     0.0
2                   0.0     0.0     0.0     0.0     1.0     0.0
3                   0.0     0.0     0.0     0.0     1.0     0.0
4                   0.0     0.0     0.0     0.0     1.0     0.0
…                     …       …       …       …       …       …
28507               0.0     0.0     1.0     0.0     0.0     0.0
28508               0.0     0.0     1.0     0.0     0.0     0.0
28509               0.0     0.0     1.0     0.0     0.0     0.0
28510               0.0     0.0     1.0     0.0     0.0     0.0
28511               0.0     0.0     1.0     0.0     0.0     0.0

[28512 rows x 134 columns]
```

[77]:

```python
datatest2.to_csv('Test_Transactions.csv', index=False)
```

[75]:

```python

```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-75-46349b375080> in <module>
----> 1 result_test_predictions

NameError: name 'result_test_predictions' is not defined
```

[80]:

```python
tabla_nueva = test_transactions.copy()
tabla_nueva
```

[80]:

```
            id        date  store_nbr                      family  onpromotion  \
0      3000888  2017-08-16          1                  AUTOMOTIVE            0
1      3000889  2017-08-16          1                   BABY CARE            0
2      3000890  2017-08-16          1                      BEAUTY            2
3      3000891  2017-08-16          1                   BEVERAGES           20
4      3000892  2017-08-16          1                       BOOKS            0
…          …           …          …                           …            …
28507  3029395  2017-08-31          9                     POULTRY            1
28508  3029396  2017-08-31          9              PREPARED FOODS            0
28509  3029397  2017-08-31          9                     PRODUCE            1
28510  3029398  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9
28511  3029399  2017-08-31          9                     SEAFOOD            0
```

```
        city      state type  cluster  dcoilwtico  transactions
0      Quito  Pichincha    D       13       46.80           0.0
1      Quito  Pichincha    D       13       46.80           0.0
2      Quito  Pichincha    D       13       46.80           0.0
3      Quito  Pichincha    D       13       46.80           0.0
4      Quito  Pichincha    D       13       46.80           0.0
…      …          …   …    …        …           …
28507  Quito  Pichincha    B        6       47.26           0.0
28508  Quito  Pichincha    B        6       47.26           0.0
28509  Quito  Pichincha    B        6       47.26           0.0
28510  Quito  Pichincha    B        6       47.26           0.0
28511  Quito  Pichincha    B        6       47.26           0.0

[28512 rows x 11 columns]
```

[79]:
```
tabla_nueva['predictions'] = result_test_predictions
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-79-95567f006562> in <module>
----> 1 tabla_nueva['predictions'] = result_test_predictions

NameError: name 'result_test_predictions' is not defined
```

[73]:
```
display = tabla_nueva.copy()
stores = display.groupby(['date', 'store_nbr'], as_index=False)['predictions'].
 ↪sum()
```

[74]:
```
px.line(stores, x = "date", y= "predictions", color = "store_nbr", title =␣
 ↪"Daily total sales of the stores")
```

[72]:
```
model.save_model("xgboost_model_transactions.json")
```

[82]:
```
datatrain2.to_csv('VER.csv', index=False)
```

[77]:
```
result_test_predictions
```

[77]:
```
array([ 0.02794334, -0.03940487,  1.9352344 , …,  0.42644447,
        8.658599  ,  0.02794334], dtype=float32)
```

[82]:
```
total_transactions = tabla_nueva['transactions'].sum()

# SHow the total.
print("Total de transacciones:", total_transactions)
```

```
Total de transacciones: 0.0
```

[ ]:

```

```

[ ]:

```

```

[ ]:

```

```

model-xgboost-sales

September 28, 2024

0.1 Import libraries and load the datasets

[1]:
```
import numpy as np  # Linear algebra
import pandas as pd  # Data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb
from sklearn.model_selection import train_test_split, TimeSeriesSplit
from sklearn.metrics import accuracy_score, roc_auc_score, roc_curve,␣
↪mean_squared_error, mean_absolute_error, r2_score
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler
from datetime import datetime
import calendar
import warnings
from tqdm import tqdm
import plotly.express as px
from sklearn.linear_model import LinearRegression
from datetime import datetime
```

[2]:
```
train_dataset = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/
↪train.csv', parse_dates=['date'])
test_dataset  = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/test.
↪csv', parse_dates=['date'])
store_dataset = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/
↪stores.csv')
oil_dataset   = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/oil.
↪csv',parse_dates=['date'])
holiday_dataset  = pd.read_csv('C:/Users/User/OneDrive - Universidad␣
↪Internacional del Ecuador/Escritorio/Master Primer Semestre/Software for IA/
↪Project 1/holidays_events.csv', parse_dates=['date'])
transactions_dataset = pd.read_csv('C:/Users/User/OneDrive - Universidad␣
↪Internacional del Ecuador/Escritorio/Master Primer Semestre/Software for IA/
↪Project 1/transactions.csv', parse_dates=['date'])
```

0.2 Now it's time to check the train dataset.

[3]:
```
train_dataset.head()
```

[3]:
```
   id        date  store_nbr      family  sales  onpromotion
0   0  2013-01-01          1  AUTOMOTIVE    0.0            0
1   1  2013-01-01          1   BABY CARE    0.0            0
2   2  2013-01-01          1      BEAUTY    0.0            0
3   3  2013-01-01          1   BEVERAGES    0.0            0
4   4  2013-01-01          1       BOOKS    0.0            0
```

[4]:
```
train_dataset.isna().sum()
```

[4]:
```
id             0
date           0
store_nbr      0
family         0
sales          0
onpromotion    0
dtype: int64
```

[5]:
```
train_dataset.shape
```

[5]:
```
(3000888, 6)
```

[6]:
```
train_dataset.describe()
```

[6]:
```
                 id                           date     store_nbr  \
count  3.000888e+06                        3000888  3.000888e+06
mean   1.500444e+06  2015-04-24 08:27:04.703088384  2.750000e+01
min    0.000000e+00            2013-01-01 00:00:00  1.000000e+00
25%    7.502218e+05            2014-02-26 18:00:00  1.400000e+01
50%    1.500444e+06            2015-04-24 12:00:00  2.750000e+01
75%    2.250665e+06            2016-06-19 06:00:00  4.100000e+01
max    3.000887e+06            2017-08-15 00:00:00  5.400000e+01
std    8.662819e+05                            NaN  1.558579e+01

              sales   onpromotion
count  3.000888e+06  3.000888e+06
mean   3.577757e+02  2.602770e+00
min    0.000000e+00  0.000000e+00
25%    0.000000e+00  0.000000e+00
50%    1.100000e+01  0.000000e+00
75%    1.958473e+02  0.000000e+00
max    1.247170e+05  7.410000e+02
std    1.101998e+03  1.221888e+01
```

[7]:
```
day1 = train_dataset['date'].min().strftime('%Y-%m-%d')
last_day = train_dataset['date'].max().strftime('%Y-%m-%d')

day1, last_day
```

[7]:
```
('2013-01-01', '2017-08-15')
```

0.3 Now it's time to check the test dataset.

[8]:
```
test_dataset.head()
```

[8]:

	id	date	store_nbr	family	onpromotion
0	3000888	2017-08-16	1	AUTOMOTIVE	0
1	3000889	2017-08-16	1	BABY CARE	0
2	3000890	2017-08-16	1	BEAUTY	2
3	3000891	2017-08-16	1	BEVERAGES	20
4	3000892	2017-08-16	1	BOOKS	0

[9]:
```
test_dataset.isna().sum()
```

[9]:
```
id             0
date           0
store_nbr      0
family         0
onpromotion    0
dtype: int64
```

[10]:
```
test_dataset.shape
```

[10]:
```
(28512, 5)
```

[11]:
```
test_dataset.describe()
```

[11]:

	id	date	store_nbr	onpromotion
count	2.851200e+04	28512	28512.000000	28512.000000
mean	3.015144e+06	2017-08-23 12:00:00	27.500000	6.965383
min	3.000888e+06	2017-08-16 00:00:00	1.000000	0.000000
25%	3.008016e+06	2017-08-19 18:00:00	14.000000	0.000000
50%	3.015144e+06	2017-08-23 12:00:00	27.500000	0.000000
75%	3.022271e+06	2017-08-27 06:00:00	41.000000	6.000000
max	3.029399e+06	2017-08-31 00:00:00	54.000000	646.000000
std	8.230850e+03	NaN	15.586057	20.683952

[12]:
```
test_day1 = test_dataset['date'].min().strftime('%Y-%m-%d')
test_last_day = test_dataset['date'].max().strftime('%Y-%m-%d')

test_day1, test_last_day
```

[12]: ('2017-08-16', '2017-08-31')

0.4 Now it's time to check the store dataset.

[13]:
```
store_dataset.isna().sum()
```

[13]:
```
store_nbr    0
city         0
state        0
type         0
cluster      0
dtype: int64
```

[14]:
```
store_dataset.shape
```

[14]: (54, 5)

[15]:
```
store_dataset.describe()
```

[15]:

	store_nbr	cluster
count	54.000000	54.000000
mean	27.500000	8.481481
std	15.732133	4.693395
min	1.000000	1.000000
25%	14.250000	4.000000
50%	27.500000	8.500000
75%	40.750000	13.000000
max	54.000000	17.000000

0.5 Now it's time to check the oil dataset.

[16]:
```
oil_dataset.head()
```

[16]:

	date	dcoilwtico
0	2013-01-01	NaN
1	2013-01-02	93.14
2	2013-01-03	92.97
3	2013-01-04	93.12
4	2013-01-07	93.20

[17]:
```
oil_dataset.shape
```

[17]: (1218, 2)

[18]:
```
oil_dataset.isna().sum()
```

[18]:
```
date           0
dcoilwtico    43
dtype: int64
```

[19]:
```
oil_dataset['dcoilwtico'] = oil_dataset['dcoilwtico'].fillna(method='ffill')
oil_dataset['dcoilwtico'] = oil_dataset['dcoilwtico'].fillna(method='bfill')
```

[20]:
```
oil_dataset.isna().sum()
```

[20]:
```
date          0
dcoilwtico    0
dtype: int64
```

[21]:
```
oil_dataset.describe()
```

[21]:

	date	dcoilwtico
count	1218	1218.000000
mean	2015-05-02 12:00:00	67.692159
min	2013-01-01 00:00:00	26.190000
25%	2014-03-03 06:00:00	46.422500
50%	2015-05-02 12:00:00	53.200000
75%	2016-06-30 18:00:00	95.685000
max	2017-08-31 00:00:00	110.620000
std	NaN	25.629744

0.6 Now it's time to check the Holiday dataset.

[22]:
```
holiday_dataset.head()
```

[22]:

	date	type	locale	locale_name	description	transferred
0	2012-03-02	Holiday	Local	Manta	Fundacion de Manta	False
1	2012-04-01	Holiday	Regional	Cotopaxi	Provincializacion de Cotopaxi	False
2	2012-04-12	Holiday	Local	Cuenca	Fundacion de Cuenca	False
3	2012-04-14	Holiday	Local	Libertad	Cantonizacion de Libertad	False
4	2012-04-21	Holiday	Local	Riobamba	Cantonizacion de Riobamba	False

[23]:
```
holiday_dataset.isna().sum()
```

[23]:
```
date           0
type           0
```

```
locale          0
locale_name     0
description     0
transferred     0
dtype: int64
```

[24]:
```
holiday_dataset.shape
```

[24]:
```
(350, 6)
```

[25]:
```
holiday_dataset.describe()
```

[25]:

	date
count	350
mean	2015-04-24 00:45:15.428571392
min	2012-03-02 00:00:00
25%	2013-12-23 06:00:00
50%	2015-06-08 00:00:00
75%	2016-07-03 00:00:00
max	2017-12-26 00:00:00

0.7 Now it's time to check the Transactions dataset.

[26]:
```
transactions_dataset.isna().sum()
```

[26]:
```
date            0
store_nbr       0
transactions    0
dtype: int64
```

[27]:
```
transactions_dataset.shape
```

[27]:
```
(83488, 3)
```

[28]:
```
transactions_dataset.describe()
```

[28]:

	date	store_nbr	transactions
count	83488	83488.000000	83488.000000
mean	2015-05-20 16:07:40.866232064	26.939237	1694.602158
min	2013-01-01 00:00:00	1.000000	5.000000
25%	2014-03-27 00:00:00	13.000000	1046.000000
50%	2015-06-08 00:00:00	27.000000	1393.000000
75%	2016-07-14 06:00:00	40.000000	2079.000000
max	2017-08-15 00:00:00	54.000000	8359.000000
std	NaN	15.608204	963.286644

[29]:
```python
train = train_dataset.copy()
stores = train.groupby(['date', 'store_nbr'], as_index=False)['sales'].sum()
```

[30]:
```python
train.shape
```

[30]: (3000888, 6)

[31]:
```python
px.line(stores, x = "date", y= "sales", color = "store_nbr", title = "Daily␣
↪total sales of the stores")
```

It can be observed that from April 16-17, 2016, sales grew significantly, due to the earthquake that struck Ecuador on that date. That is why later this data will be removed. Also, there are stores that were opened after 2013, others since 2015 and so on, so everything before those dates should be removed.

[32]:
```python
train = train[~((train.store_nbr == 52) & (train.date < "2017-04-20"))]
train = train[~((train.store_nbr == 22) & (train.date < "2015-10-09"))]
train = train[~((train.store_nbr == 42) & (train.date < "2015-08-21"))]
train = train[~((train.store_nbr == 21) & (train.date < "2015-07-24"))]
train = train[~((train.store_nbr == 29) & (train.date < "2015-03-20"))]
train = train[~((train.store_nbr == 20) & (train.date < "2015-02-13"))]
train = train[~((train.store_nbr == 53) & (train.date < "2014-05-29"))]
train = train[~((train.store_nbr == 36) & (train.date < "2013-05-09"))]
```

[33]:
```python
start_date = '2016-04-16'
end_date = '2016-05-02'

filtered_data = train[(train['date'] >= start_date) & (train['date'] <=␣
↪end_date)] # Filter the data in the date range
mean_sales_by_class = filtered_data.groupby('store_nbr')['sales'].mean().
↪reset_index() # Grouping by class and calculating the average
mean_sales_by_class.rename(columns={'sales': 'mean_sales'}, inplace=True) #␣
↪Rename the sales column for the union

train = train.merge(mean_sales_by_class, on='store_nbr', how='left')# Join␣
↪DataFrames and replace values
train.loc[(train['date'] >= start_date) & (train['date'] <= end_date), 'sales']␣
↪= train['mean_sales']
train.drop('mean_sales', axis=1, inplace=True)
```

[34]:
```python
stores2 = train.groupby(['date', 'store_nbr'], as_index=False)['sales'].sum()
px.line(stores2, x = "date", y= "sales", color = "store_nbr", title = "Daily␣
↪total sales of the stores")
```

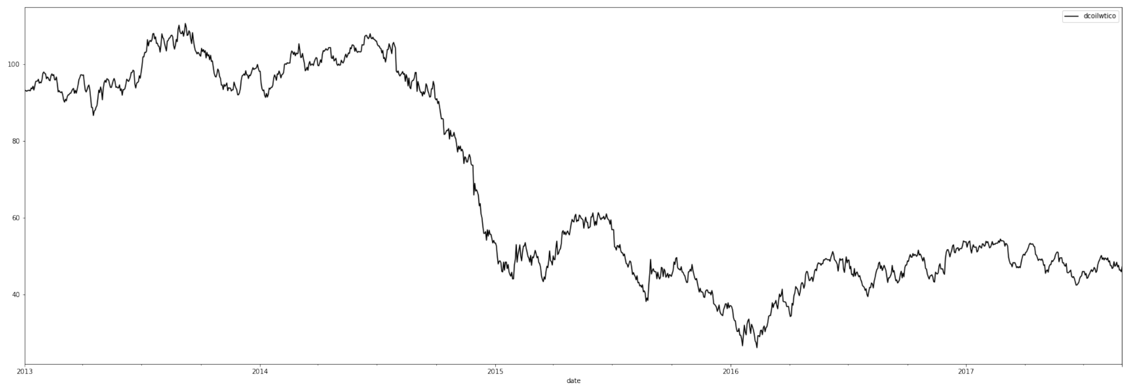
[35]:
```python
train.shape
```

[35]: (2780316, 6)

Let's check what happens with the oil

[36]:
```
oil_dataset.set_index('date').plot(figsize = (30,10),color='black')
```

[36]:
```
<AxesSubplot:xlabel='date'>
```



[37]:
```
oil_dataset['date'] = pd.to_datetime(oil_dataset['date'])
oil_dataset.set_index('date', inplace=True)
```

[38]:
```
train_store = pd.merge(train, store_dataset, on='store_nbr', how='left')
test_store = pd.merge(test_dataset, store_dataset, on='store_nbr', how='left')
```

[39]:
```
train_store
```

[39]:
```
              id        date  store_nbr                      family     sales  \
0              0  2013-01-01          1                  AUTOMOTIVE     0.000
1              1  2013-01-01          1                   BABY CARE     0.000
2              2  2013-01-01          1                      BEAUTY     0.000
3              3  2013-01-01          1                   BEVERAGES     0.000
4              4  2013-01-01          1                       BOOKS     0.000
…            …           …         …                          …         …
2780311  3000883  2017-08-15          9                     POULTRY   438.133
2780312  3000884  2017-08-15          9              PREPARED FOODS   154.553
2780313  3000885  2017-08-15          9                     PRODUCE  2419.729
2780314  3000886  2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
2780315  3000887  2017-08-15          9                     SEAFOOD    16.000

         onpromotion   city      state type  cluster
0                  0  Quito  Pichincha    D       13
1                  0  Quito  Pichincha    D       13
2                  0  Quito  Pichincha    D       13
3                  0  Quito  Pichincha    D       13
4                  0  Quito  Pichincha    D       13
…                  …      …          …    …        …
```

```
2780311           0  Quito  Pichincha    B        6
2780312           1  Quito  Pichincha    B        6
2780313         148  Quito  Pichincha    B        6
2780314           8  Quito  Pichincha    B        6
2780315           0  Quito  Pichincha    B        6

[2780316 rows x 10 columns]
```

[40]:
```
test_store
```

[40]:
```
            id        date  store_nbr                     family  onpromotion  \
0      3000888  2017-08-16          1                 AUTOMOTIVE            0
1      3000889  2017-08-16          1                  BABY CARE            0
2      3000890  2017-08-16          1                     BEAUTY            2
3      3000891  2017-08-16          1                  BEVERAGES           20
4      3000892  2017-08-16          1                      BOOKS            0
…          …         …        …                          …          …
28507  3029395  2017-08-31          9                    POULTRY            1
28508  3029396  2017-08-31          9             PREPARED FOODS            0
28509  3029397  2017-08-31          9                    PRODUCE            1
28510  3029398  2017-08-31          9 SCHOOL AND OFFICE SUPPLIES            9
28511  3029399  2017-08-31          9                    SEAFOOD            0

        city      state type  cluster
0      Quito  Pichincha    D       13
1      Quito  Pichincha    D       13
2      Quito  Pichincha    D       13
3      Quito  Pichincha    D       13
4      Quito  Pichincha    D       13
…      …          …  …        …
28507  Quito  Pichincha    B        6
28508  Quito  Pichincha    B        6
28509  Quito  Pichincha    B        6
28510  Quito  Pichincha    B        6
28511  Quito  Pichincha    B        6

[28512 rows x 9 columns]
```

[41]:
```
train_oil = pd.merge(train_store, oil_dataset, on='date', how='left')
test_oil = pd.merge(test_store, oil_dataset, on='date', how='left')
```

[42]:
```
train_oil
```

[42]:
```
            id        date  store_nbr                     family     sales  \
0            0  2013-01-01          1                 AUTOMOTIVE     0.000
1            1  2013-01-01          1                  BABY CARE     0.000
2            2  2013-01-01          1                     BEAUTY     0.000
```

```
3              3  2013-01-01          1                       BEVERAGES      0.000
4              4  2013-01-01          1                           BOOKS      0.000
…            …         …          …                            …         …
2780311  3000883  2017-08-15          9                         POULTRY    438.133
2780312  3000884  2017-08-15          9                  PREPARED FOODS    154.553
2780313  3000885  2017-08-15          9                         PRODUCE   2419.729
2780314  3000886  2017-08-15          9   SCHOOL AND OFFICE SUPPLIES    121.000
2780315  3000887  2017-08-15          9                         SEAFOOD     16.000

         onpromotion   city      state type  cluster  dcoilwtico
0                  0  Quito  Pichincha    D       13       93.14
1                  0  Quito  Pichincha    D       13       93.14
2                  0  Quito  Pichincha    D       13       93.14
3                  0  Quito  Pichincha    D       13       93.14
4                  0  Quito  Pichincha    D       13       93.14
…                …    …          …    …        …          …
2780311            0  Quito  Pichincha    B        6       47.57
2780312            1  Quito  Pichincha    B        6       47.57
2780313          148  Quito  Pichincha    B        6       47.57
2780314            8  Quito  Pichincha    B        6       47.57
2780315            0  Quito  Pichincha    B        6       47.57

[2780316 rows x 11 columns]
```

[43]:
```python
train_transactions = pd.merge(train_oil, transactions_dataset,␣
 ↪on=['date','store_nbr'],
                                         how='left')
test_transactions = pd.merge(test_oil, transactions_dataset,␣
 ↪on=['date','store_nbr'],
                                         how='left')
```

[44]:
```python
train_transactions.isna().sum()
```

[44]:
```
id                   0
date                 0
store_nbr            0
family               0
sales                0
onpromotion          0
city                 0
state                0
type                 0
cluster              0
dcoilwtico      794211
transactions     25212
dtype: int64
```

[45]:
```
train_transactions['dcoilwtico'].fillna(0, inplace=True)
train_transactions['transactions'].fillna(0, inplace=True)
```

[46]:
```
train_transactions.isna().sum()
```

[46]:
```
id              0
date            0
store_nbr       0
family          0
sales           0
onpromotion     0
city            0
state           0
type            0
cluster         0
dcoilwtico      0
transactions    0
dtype: int64
```

[47]:
```
train_transactions
```

[47]:
```
              id        date  store_nbr                      family     sales  \
0              0  2013-01-01          1                  AUTOMOTIVE     0.000
1              1  2013-01-01          1                   BABY CARE     0.000
2              2  2013-01-01          1                      BEAUTY     0.000
3              3  2013-01-01          1                   BEVERAGES     0.000
4              4  2013-01-01          1                       BOOKS     0.000
…            …           …          …                          …          …
2780311  3000883  2017-08-15          9                     POULTRY   438.133
2780312  3000884  2017-08-15          9              PREPARED FOODS   154.553
2780313  3000885  2017-08-15          9                     PRODUCE  2419.729
2780314  3000886  2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
2780315  3000887  2017-08-15          9                     SEAFOOD    16.000

         onpromotion   city      state type  cluster  dcoilwtico  transactions
0                  0  Quito  Pichincha    D       13       93.14           0.0
1                  0  Quito  Pichincha    D       13       93.14           0.0
2                  0  Quito  Pichincha    D       13       93.14           0.0
3                  0  Quito  Pichincha    D       13       93.14           0.0
4                  0  Quito  Pichincha    D       13       93.14           0.0
…                  …      …          …    …        …           …             …
2780311            0  Quito  Pichincha    B        6       47.57        2155.0
2780312            1  Quito  Pichincha    B        6       47.57        2155.0
2780313          148  Quito  Pichincha    B        6       47.57        2155.0
2780314            8  Quito  Pichincha    B        6       47.57        2155.0
2780315            0  Quito  Pichincha    B        6       47.57        2155.0
```

```
[2780316 rows x 12 columns]
```

[48]:
```
test_transactions['dcoilwtico'].fillna(0, inplace=True)
test_transactions['transactions'].fillna(0, inplace=True)
```

[49]:
```
test_transactions.isna().sum()
```

[49]:
```
id              0
date            0
store_nbr       0
family          0
onpromotion     0
city            0
state           0
type            0
cluster         0
dcoilwtico      0
transactions    0
dtype: int64
```

[50]:
```
test_transactions
```

[50]:
```
             id        date  store_nbr                      family  onpromotion  \
0       3000888  2017-08-16          1                  AUTOMOTIVE            0
1       3000889  2017-08-16          1                   BABY CARE            0
2       3000890  2017-08-16          1                      BEAUTY            2
3       3000891  2017-08-16          1                   BEVERAGES           20
4       3000892  2017-08-16          1                       BOOKS            0
…           …         …          …                           …          …
28507   3029395  2017-08-31          9                     POULTRY            1
28508   3029396  2017-08-31          9              PREPARED FOODS            0
28509   3029397  2017-08-31          9                     PRODUCE            1
28510   3029398  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9
28511   3029399  2017-08-31          9                     SEAFOOD            0

         city      state type  cluster  dcoilwtico  transactions
0       Quito  Pichincha    D       13       46.80           0.0
1       Quito  Pichincha    D       13       46.80           0.0
2       Quito  Pichincha    D       13       46.80           0.0
3       Quito  Pichincha    D       13       46.80           0.0
4       Quito  Pichincha    D       13       46.80           0.0
…         …          …  …        …           …             …
28507   Quito  Pichincha    B        6       47.26           0.0
28508   Quito  Pichincha    B        6       47.26           0.0
28509   Quito  Pichincha    B        6       47.26           0.0
28510   Quito  Pichincha    B        6       47.26           0.0
28511   Quito  Pichincha    B        6       47.26           0.0
```

```
[28512 rows x 11 columns]
```

[51]:
```python
datatrain2 = train_transactions.copy()
datatest2 = test_transactions.copy()
datatrain2.head()
```

[51]:
```
   id        date  store_nbr      family  sales  onpromotion   city      state  \
0   0 2013-01-01          1  AUTOMOTIVE    0.0            0  Quito  Pichincha
1   1 2013-01-01          1   BABY CARE    0.0            0  Quito  Pichincha
2   2 2013-01-01          1      BEAUTY    0.0            0  Quito  Pichincha
3   3 2013-01-01          1   BEVERAGES    0.0            0  Quito  Pichincha
4   4 2013-01-01          1       BOOKS    0.0            0  Quito  Pichincha

  type  cluster  dcoilwtico  transactions
0    D       13       93.14           0.0
1    D       13       93.14           0.0
2    D       13       93.14           0.0
3    D       13       93.14           0.0
4    D       13       93.14           0.0
```

[52]:
```python
datatrain2 = datatrain2.drop('id', axis=1)
datatrain2.head()
```

[52]:
```
        date  store_nbr      family  sales  onpromotion   city      state  \
0 2013-01-01          1  AUTOMOTIVE    0.0            0  Quito  Pichincha
1 2013-01-01          1   BABY CARE    0.0            0  Quito  Pichincha
2 2013-01-01          1      BEAUTY    0.0            0  Quito  Pichincha
3 2013-01-01          1   BEVERAGES    0.0            0  Quito  Pichincha
4 2013-01-01          1       BOOKS    0.0            0  Quito  Pichincha

  type  cluster  dcoilwtico  transactions
0    D       13       93.14           0.0
1    D       13       93.14           0.0
2    D       13       93.14           0.0
3    D       13       93.14           0.0
4    D       13       93.14           0.0
```

[53]:
```python
datatrain2 = datatrain2.drop('transactions', axis=1) #It makes noise, it has␣
↪nothing to do with predicting sales.
```

[54]:
```python
def add_features(df):
    df['date'] = pd.to_datetime(df['date'])
    df['weekday'] = df['date'].dt.weekday
    df['year'] = df['date'].dt.year
    df['month'] = df['date'].dt.month
    df['day'] = df['date'].dt.day
```

```
    df['eomd'] = df['date'].apply(lambda x: calendar.monthrange(x.year, x.
↪month)[1])
    df['payday'] = ((df['day'] == 15) | (df['day'] == df['eomd'])).astype(int)
    df['is_weekend'] = df['weekday'].isin([5, 6]).astype(int)
    df.drop(['eomd'], axis=1, inplace=True)
    return df

datatrain2 = add_features(datatrain2)
datatest2 = add_features(datatest2)
```

[55]:
```
datatrain2
```

[55]:
```
               date  store_nbr                      family     sales  \
0        2013-01-01          1                  AUTOMOTIVE     0.000
1        2013-01-01          1                   BABY CARE     0.000
2        2013-01-01          1                      BEAUTY     0.000
3        2013-01-01          1                   BEVERAGES     0.000
4        2013-01-01          1                       BOOKS     0.000
…               …         …                          …       …
2780311  2017-08-15          9                     POULTRY   438.133
2780312  2017-08-15          9              PREPARED FOODS   154.553
2780313  2017-08-15          9                     PRODUCE  2419.729
2780314  2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
2780315  2017-08-15          9                     SEAFOOD    16.000

         onpromotion   city      state type  cluster  dcoilwtico  weekday  \
0                  0  Quito  Pichincha    D       13       93.14        1
1                  0  Quito  Pichincha    D       13       93.14        1
2                  0  Quito  Pichincha    D       13       93.14        1
3                  0  Quito  Pichincha    D       13       93.14        1
4                  0  Quito  Pichincha    D       13       93.14        1
…                  …    …          …    …        …          …
2780311            0  Quito  Pichincha    B        6       47.57        1
2780312            1  Quito  Pichincha    B        6       47.57        1
2780313          148  Quito  Pichincha    B        6       47.57        1
2780314            8  Quito  Pichincha    B        6       47.57        1
2780315            0  Quito  Pichincha    B        6       47.57        1

         year  month  day  payday  is_weekend
0        2013      1    1       0           0
1        2013      1    1       0           0
2        2013      1    1       0           0
3        2013      1    1       0           0
4        2013      1    1       0           0
…           …     …    …       …           …
2780311  2017      8   15       1           0
2780312  2017      8   15       1           0
```

```
2780313  2017       8    15         1             0
2780314  2017       8    15         1             0
2780315  2017       8    15         1             0

[2780316 rows x 16 columns]
```

[56]:
```python
datatest2 = datatest2.drop(['id','transactions'], axis=1)
```

[57]:
```python
datatest2
```

[57]:
```
             date  store_nbr                      family  onpromotion   city  \
0      2017-08-16          1                  AUTOMOTIVE            0  Quito
1      2017-08-16          1                   BABY CARE            0  Quito
2      2017-08-16          1                      BEAUTY            2  Quito
3      2017-08-16          1                   BEVERAGES           20  Quito
4      2017-08-16          1                       BOOKS            0  Quito
…             …         …                           …           …     …
28507  2017-08-31          9                     POULTRY            1  Quito
28508  2017-08-31          9              PREPARED FOODS            0  Quito
28509  2017-08-31          9                     PRODUCE            1  Quito
28510  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9  Quito
28511  2017-08-31          9                     SEAFOOD            0  Quito

           state type  cluster  dcoilwtico  weekday  year  month  day  payday  \
0      Pichincha    D       13       46.80        2  2017      8   16       0
1      Pichincha    D       13       46.80        2  2017      8   16       0
2      Pichincha    D       13       46.80        2  2017      8   16       0
3      Pichincha    D       13       46.80        2  2017      8   16       0
4      Pichincha    D       13       46.80        2  2017      8   16       0
…            …    …        …           …        …     …      …    …       …
28507  Pichincha    B        6       47.26        3  2017      8   31       1
28508  Pichincha    B        6       47.26        3  2017      8   31       1
28509  Pichincha    B        6       47.26        3  2017      8   31       1
28510  Pichincha    B        6       47.26        3  2017      8   31       1
28511  Pichincha    B        6       47.26        3  2017      8   31       1

       is_weekend
0               0
1               0
2               0
3               0
4               0
…             …
28507           0
28508           0
28509           0
28510           0
```

```
28511           0

[28512 rows x 15 columns]
```

[58]:
```
def add_lag(df, lags):
    for lag in lags:
        df[f'sales_lag_{lag}'] = df.groupby(['store_nbr', 'family'])['sales'].
↪transform(lambda x: x.shift(lag))
    return df

def add_rolling_mean(df, windows):
    for window in windows:
        df[f'sales_roll_mean_{window}'] = df.groupby(['store_nbr',␣
↪'family'])['sales'].transform(
            lambda x: x.shift(1).rolling(window=window, min_periods=1).mean())␣
↪+ add_noise(df)
    return df

def add_ewm(df, alphas, lags):
    for alpha in alphas:
        for lag in lags:
            df[f'sales_ewm_alpha_{str(alpha).replace(".", "")}_lag_{lag}'] = df.
↪groupby(['store_nbr', 'family'])['sales'].transform(
                lambda x: x.shift(lag).ewm(alpha=alpha).mean())
    return df

def add_noise(df):
    return np.random.normal(scale=2.0, size=(len(df),))
```

[59]:
```
dataset_completo = pd.concat([datatrain2, datatest2], axis=0, ignore_index=True)
```

[60]:
```
dataset_completo
```

[60]:
```
               date  store_nbr                      family  sales  onpromotion  \
0        2013-01-01          1                  AUTOMOTIVE    0.0            0
1        2013-01-01          1                   BABY CARE    0.0            0
2        2013-01-01          1                      BEAUTY    0.0            0
3        2013-01-01          1                   BEVERAGES    0.0            0
4        2013-01-01          1                       BOOKS    0.0            0
…               …          …                           …      …            …
2808823  2017-08-31          9                     POULTRY    NaN            1
2808824  2017-08-31          9              PREPARED FOODS    NaN            0
2808825  2017-08-31          9                     PRODUCE    NaN            1
2808826  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES    NaN            9
2808827  2017-08-31          9                     SEAFOOD    NaN            0

          city      state type  cluster  dcoilwtico  weekday  year  month  \
```

```
0        Quito  Pichincha    D       13       93.14        1  2013       1
1        Quito  Pichincha    D       13       93.14        1  2013       1
2        Quito  Pichincha    D       13       93.14        1  2013       1
3        Quito  Pichincha    D       13       93.14        1  2013       1
4        Quito  Pichincha    D       13       93.14        1  2013       1
…        …          …   …       …       …       …    …     …
2808823  Quito  Pichincha    B        6       47.26        3  2017       8
2808824  Quito  Pichincha    B        6       47.26        3  2017       8
2808825  Quito  Pichincha    B        6       47.26        3  2017       8
2808826  Quito  Pichincha    B        6       47.26        3  2017       8
2808827  Quito  Pichincha    B        6       47.26        3  2017       8

         day  payday  is_weekend
0          1       0           0
1          1       0           0
2          1       0           0
3          1       0           0
4          1       0           0
…      …     …         …
2808823   31       1           0
2808824   31       1           0
2808825   31       1           0
2808826   31       1           0
2808827   31       1           0

[2808828 rows x 16 columns]
```

[61]:
```
lags = [7, 14, 30]
windows = [7, 30]
ewm_alphas = [0.95, 0.9, 0.8]
ewm_lags = [7, 30]

dataset_completo = add_lag(dataset_completo, lags)
dataset_completo = add_rolling_mean(dataset_completo, windows)
dataset_completo = add_ewm(dataset_completo, ewm_alphas, ewm_lags)
```

[62]:
```
dataset_completo.fillna(0, inplace=True)
```

[63]:
```
train_data = dataset_completo[dataset_completo['date'] <= '2017-08-15'].copy()
test_data = dataset_completo[dataset_completo['date'] > '2017-08-15'].copy()
```

[64]:
```
c_feat = ['family', 'city', 'state', 'type', 'cluster', 'store_nbr']

train_data_enc = train_data
test_data_enc = test_data

train_data_enc[c_feat] = train_data_enc[c_feat].astype(str)
```

```python
test_data_enc[c_feat] = test_data_enc[c_feat].astype(str)

label_encoders = {}
for col in c_feat:
    le = LabelEncoder()
    train_data_enc[col] = le.fit_transform(train_data[col])
    test_data_enc[col] = le.transform(test_data[col])
    label_encoders[col] = le
```

[65]:
```python
train_data_enc.head()
```

[65]:
```
         date  store_nbr  family  sales  onpromotion  city  state  type  \
0  2013-01-01          0       0    0.0            0    18     12     3
1  2013-01-01          0       1    0.0            0    18     12     3
2  2013-01-01          0       2    0.0            0    18     12     3
3  2013-01-01          0       3    0.0            0    18     12     3
4  2013-01-01          0       4    0.0            0    18     12     3

   cluster  dcoilwtico  …  sales_lag_14  sales_lag_30  sales_roll_mean_7  \
0        4       93.14  …           0.0           0.0                0.0
1        4       93.14  …           0.0           0.0                0.0
2        4       93.14  …           0.0           0.0                0.0
3        4       93.14  …           0.0           0.0                0.0
4        4       93.14  …           0.0           0.0                0.0

   sales_roll_mean_30  sales_ewm_alpha_095_lag_7  sales_ewm_alpha_095_lag_30  \
0                 0.0                        0.0                         0.0
1                 0.0                        0.0                         0.0
2                 0.0                        0.0                         0.0
3                 0.0                        0.0                         0.0
4                 0.0                        0.0                         0.0

   sales_ewm_alpha_09_lag_7  sales_ewm_alpha_09_lag_30  \
0                       0.0                        0.0
1                       0.0                        0.0
2                       0.0                        0.0
3                       0.0                        0.0
4                       0.0                        0.0

   sales_ewm_alpha_08_lag_7  sales_ewm_alpha_08_lag_30
0                       0.0                        0.0
1                       0.0                        0.0
2                       0.0                        0.0
3                       0.0                        0.0
4                       0.0                        0.0

[5 rows x 27 columns]
```

[66]:
```
test_data_enc.nunique()
```

[66]:
```
date                         16
store_nbr                    54
family                       33
sales                         1
onpromotion                 212
city                         22
state                        16
type                          5
cluster                      17
dcoilwtico                   12
weekday                       7
year                          1
month                         1
day                          16
payday                        2
is_weekend                    2
sales_lag_7                3838
sales_lag_14               6981
sales_lag_30               7877
sales_roll_mean_7         12475
sales_roll_mean_30        28512
sales_ewm_alpha_095_lag_7      11907
sales_ewm_alpha_095_lag_30     26909
sales_ewm_alpha_09_lag_7       11961
sales_ewm_alpha_09_lag_30      27177
sales_ewm_alpha_08_lag_7       12093
sales_ewm_alpha_08_lag_30      27957
dtype: int64
```

[67]:
```
train_data_enc.drop(['date'], axis=1, inplace=True)
test_data_enc.drop(['date'], axis=1, inplace=True)
```

[68]:
```
# split data into X parameter and y as target
X = train_data_enc.drop('sales', axis=1)
Y = train_data_enc['sales'] #Sales values are stored
```

[69]:
```
sub_frac = 0.20
sub_size = int(len(train_data_enc) * sub_frac)
sub_train_data = train_data_enc.iloc[-sub_size:]
X_sub = sub_train_data.drop(columns=["sales"])
y_sub = sub_train_data["sales"]
split_index = int(0.8 * len(X_sub))
X_sub_train, X_sub_val = X_sub.iloc[:split_index], X_sub.iloc[split_index:]
y_sub_train, y_sub_val = y_sub.iloc[:split_index], y_sub.iloc[split_index:]
```

[70]:
```
X_sub_train
```

[70]:
```
         store_nbr  family  onpromotion  city  state  type  cluster  \
2224253         15      20            0     0     15     3       16
2224254         15      21            0     0     15     3       16
2224255         15      22            0     0     15     3       16
2224256         15      23            0     0     15     3       16
2224257         15      24            0     0     15     3       16
…              …       …          …     …     …     …       …
2669098         31      25           15    12      8     3       11
2669099         31      26            0    12      8     3       11
2669100         31      27            0    12      8     3       11
2669101         31      28            0    12      8     3       11
2669102         31      29            0    12      8     3       11

         dcoilwtico  weekday  year  …  sales_lag_14  sales_lag_30  \
2224253       48.80        0  2016  …         0.000         3.000
2224254       48.80        0  2016  …         5.000         3.000
2224255       48.80        0  2016  …        10.000       102.000
2224256       48.80        0  2016  …         3.000         7.000
2224257       48.80        0  2016  …       464.267       937.561
…               …       …     …  …           …             …
2669098       44.79        2  2017  …       327.000       276.000
2669099       44.79        2  2017  …        13.000         6.000
2669100       44.79        2  2017  …         6.000         9.000
2669101       44.79        2  2017  …       129.766       102.120
2669102       44.79        2  2017  …        76.103        66.841

         sales_roll_mean_7  sales_roll_mean_30  sales_ewm_alpha_095_lag_7  \
2224253           2.943738            2.366726                   1.140243
2224254           6.171246            3.291172                   3.959749
2224255          35.824166           48.152743                  18.694221
2224256           5.934672            7.056525                   8.854749
2224257         532.643387          528.147369                 333.839627
…                      …                  …                         …
2669098         408.566867          395.299217                 382.692886
2669099           7.764104            6.533647                  12.508165
2669100           8.373200           13.827274                   8.246353
2669101         163.671476          154.443927                 144.814504
2669102          78.810006           75.556595                  99.037499

         sales_ewm_alpha_095_lag_30  sales_ewm_alpha_09_lag_7  \
2224253                    2.904625                  1.261890
2224254                    3.014740                  3.937968
2224255                  100.267419                 19.852470
2224256                    6.942269                  8.717963
2224257                  919.926326                350.590773
```

```
…                           …                          …
2669098            282.564800                 382.817104
2669099              6.149280                  12.035590
2669100              9.336862                   8.490592
2669101            105.359477                 142.682329
2669102             68.852780                  95.886530

         sales_ewm_alpha_09_lag_30  sales_ewm_alpha_08_lag_7  \
2224253                   2.817010                  1.454090
2224254                   3.057877                  3.942965
2224255                  98.999623                 23.397018
2224256                   6.868308                  8.462892
2224257                 902.810984                380.122449
…                                …                         …
2669098                 289.237787                385.708620
2669099                   6.294468                 11.167688
2669100                   9.644782                  8.999754
2669101                 108.890740                139.005542
2669102                  70.764559                 89.630539

         sales_ewm_alpha_08_lag_30
2224253                   2.656349
2224254                   3.222909
2224255                  97.439864
2224256                   6.669013
2224257                 869.362086
…                                …
2669098                 302.173320
2669099                   6.558949
2669100                  10.156342
2669101                 116.591112
2669102                  74.187087

[444850 rows x 25 columns]
```

[71]:

```
X_sub_val
```

[71]:

```
         store_nbr  family  onpromotion  city  state  type  cluster  \
2669103         31      30          182    12      8     3       11
2669104         31      31            0    12      8     3       11
2669105         31      32            0    12      8     3       11
2669106         32       0            0     3      0     1       13
2669107         32       1            0     3      0     1       13
…              …       …           …     …      …     …        …
2780311         53      28            0    18     12     1       13
2780312         53      29            1    18     12     1       13
2780313         53      30          148    18     12     1       13
```

```
2780314          53       31              8     18      12      1        13
2780315          53       32              0     18      12      1        13

         dcoilwtico  weekday  year  …  sales_lag_14  sales_lag_30  \
2669103       44.79        2  2017  …   3100.723000      2087.228
2669104       44.79        2  2017  …      0.000000         0.000
2669105       44.79        2  2017  …     16.041000        11.622
2669106       44.79        2  2017  …      8.000000         5.000
2669107       44.79        2  2017  …      0.000000         0.000
…               …        …    …   …          …            …
2780311       47.57        1  2017  …    570.196000       571.333
2780312       47.57        1  2017  …     50.462997       125.960
2780313       47.57        1  2017  …   2470.461000      2041.967
2780314       47.57        1  2017  …    203.000000         0.000
2780315       47.57        1  2017  …     19.316000        18.334

         sales_roll_mean_7  sales_roll_mean_30  sales_ewm_alpha_095_lag_7  \
2669103        2148.117542         2194.171214               3.076788e+03
2669104          -1.730613           -0.304716               2.906419e-20
2669105          15.047145           17.812054               9.807021e+00
2669106           9.574054            9.708823               6.959840e+00
2669107          -0.325352            0.874885               7.125015e-03
…                      …                  …                          …
2780311         370.865470          427.691049               3.635538e+02
2780312         117.218226          105.237922               1.133920e+02
2780313        1511.806331         1593.259695               2.268981e+03
2780314         149.798463           77.645551               1.689173e+02
2780315          17.204703           17.064402               1.607812e+01

         sales_ewm_alpha_095_lag_30  sales_ewm_alpha_09_lag_7  \
2669103                 2112.222254              3.006306e+03
2669104                    0.000125              9.090099e-16
2669105                   12.386494              9.800724e+00
2669106                    5.010213              6.939375e+00
2669107                    0.002376              2.700090e-02
…                              …                         …
2780311                  568.342793              3.696978e+02
2780312                  126.753250              1.140108e+02
2780313                 2026.020436              2.239195e+03
2780314                    0.095601              1.680472e+02
2780315                   19.158657              1.647858e+01

         sales_ewm_alpha_09_lag_30  sales_ewm_alpha_08_lag_7  \
2669103                2137.997922              2.872034e+03
2669104                   0.000990              2.727304e-11
2669105                  13.157456              9.795243e+00
2669106                   5.041422              6.963869e+00
```

```
2669107                  0.009019          9.605202e-02
…                               …                     …
2780311                564.910993          3.839708e+02
2780312                127.378549          1.157539e+02
2780313               2007.645869          2.181927e+03
2780314                  0.184619          1.668051e+02
2780315                 19.882348          1.725391e+01

         sales_ewm_alpha_08_lag_30
2669103               2189.211287
2669104                  0.007680
2669105                 14.703066
2669106                  5.167166
2669107                  0.032575
…                               …
2780311                556.377248
2780312                128.059505
2780313               1963.919169
2780314                  0.354174
2780315                 21.028540

[111213 rows x 25 columns]
```

[72]:
```
y_sub_train
```

[72]:
```
2224253      0.00000
2224254      5.00000
2224255     30.00000
2224256      2.00000
2224257    512.20700
             …
2669098    392.00000
2669099      3.00000
2669100     10.00000
2669101    130.44499
2669102     90.39100
Name: sales, Length: 444850, dtype: float64
```

[73]:
```
y_sub_val
```

[73]:
```
2669103    3292.113
2669104       0.000
2669105      16.652
2669106       9.000
2669107       0.000
             …
2780311     438.133
```

```
2780312     154.553
2780313    2419.729
2780314     121.000
2780315      16.000
Name: sales, Length: 111213, dtype: float64
```

[74]:

```
# Define the Optuna objective function
def objective(trial):
    params = {
        # 'tree_method': 'gpu_hist',
        'tree_method': 'hist',
        'n_jobs': -1,
        'objective': 'reg:squarederror',
        'n_estimators': trial.suggest_int('n_estimators', 100, 300),
        'verbosity': 2,
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.1,␣
↪log=True),
        'max_depth': trial.suggest_int('max_depth', 6, 14),
        'subsample': trial.suggest_float('subsample', 0.6, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.3, 1.0),
        'min_child_weight': trial.suggest_int('min_child_weight', 10, 24),
        'reg_lambda': trial.suggest_float('reg_lambda', 0.001, 1, log=True),
        'colsample_bynode': trial.suggest_float('colsample_bynode', 0.3, 0.9)
    }

    model = xgb.XGBRegressor(**params)

    model.fit(X_sub_train, y_sub_train, eval_set=[(X_sub_val, y_sub_val)],␣
↪verbose=10)
    y_pred = model.predict(X_sub_val)
    rmse = mean_squared_error(y_sub_val, y_pred, squared=False)
    return rmse
```

[221]:

```
import optuna
# Create and optimize the study
study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=10)
```

```
[I 2024-09-24 11:48:23,836] A new study created in memory with name: no-
name-5cbca059-aed8-473c-96f1-a26f42608d70
```

```
Collecting optuna
  Downloading optuna-4.0.0-py3-none-any.whl (362 kB)
Requirement already satisfied: PyYAML in c:\users\user\anaconda3\lib\site-
packages (from optuna) (5.4.1)
Requirement already satisfied: tqdm in c:\users\user\anaconda3\lib\site-packages
(from optuna) (4.65.0)
Requirement already satisfied: packaging>=20.0 in
```

```
c:\users\user\anaconda3\lib\site-packages (from optuna) (20.9)
Requirement already satisfied: sqlalchemy>=1.3.0 in
c:\users\user\anaconda3\lib\site-packages (from optuna) (1.4.7)
Collecting alembic>=1.5.0
  Downloading alembic-1.13.3-py3-none-any.whl (233 kB)
Requirement already satisfied: numpy in c:\users\user\anaconda3\lib\site-
packages (from optuna) (1.24.4)
Collecting colorlog
  Downloading colorlog-6.8.2-py3-none-any.whl (11 kB)
Requirement already satisfied: importlib-metadata in
c:\users\user\anaconda3\lib\site-packages (from alembic>=1.5.0->optuna) (4.12.0)
Collecting typing-extensions>=4
  Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Collecting Mako
  Downloading Mako-1.3.5-py3-none-any.whl (78 kB)
Collecting importlib-resources
  Downloading importlib_resources-6.4.5-py3-none-any.whl (36 kB)
Requirement already satisfied: pyparsing>=2.0.2 in
c:\users\user\anaconda3\lib\site-packages (from packaging>=20.0->optuna) (2.4.7)
Requirement already satisfied: greenlet!=0.4.17 in
c:\users\user\anaconda3\lib\site-packages (from sqlalchemy>=1.3.0->optuna)
(1.0.0)
Requirement already satisfied: colorama in c:\users\user\anaconda3\lib\site-
packages (from colorlog->optuna) (0.4.4)
Requirement already satisfied: zipp>=0.5 in c:\users\user\anaconda3\lib\site-
packages (from importlib-metadata->alembic>=1.5.0->optuna) (3.4.1)
Requirement already satisfied: MarkupSafe>=0.9.2 in
c:\users\user\anaconda3\lib\site-packages (from Mako->alembic>=1.5.0->optuna)
(1.1.1)
Installing collected packages: typing-extensions, Mako, importlib-resources,
colorlog, alembic, optuna
  Attempting uninstall: typing-extensions
    Found existing installation: typing-extensions 3.7.4.3
    Uninstalling typing-extensions-3.7.4.3:
      Successfully uninstalled typing-extensions-3.7.4.3
Successfully installed Mako-1.3.5 alembic-1.13.3 colorlog-6.8.2 importlib-
resources-6.4.5 optuna-4.0.0 typing-extensions-4.12.2
[0]     validation_0-rmse:1263.17726
[10]    validation_0-rmse:927.81752
[20]    validation_0-rmse:685.97674
[30]    validation_0-rmse:517.12289
[40]    validation_0-rmse:401.52054
[50]    validation_0-rmse:326.29905
[60]    validation_0-rmse:279.63914
[70]    validation_0-rmse:254.73066
[80]    validation_0-rmse:242.00556
[90]    validation_0-rmse:235.68210
[100]   validation_0-rmse:232.64770
```

```
[110]   validation_0-rmse:232.87556
[120]   validation_0-rmse:233.80662
[130]   validation_0-rmse:234.59401
[140]   validation_0-rmse:235.50646
[150]   validation_0-rmse:236.87766
[160]   validation_0-rmse:237.16284
[170]   validation_0-rmse:238.01588
[180]   validation_0-rmse:238.89105
[190]   validation_0-rmse:239.94140
[199]   validation_0-rmse:240.45423
```

```
[I 2024-09-24 11:48:35,561] Trial 0 finished with value: 240.4542306523302 and
parameters: {'n_estimators': 200, 'learning_rate': 0.031065338310324722,
'max_depth': 10, 'subsample': 0.7911859587955324, 'colsample_bytree':
0.35153102079041143, 'min_child_weight': 20, 'reg_lambda': 0.001565686132561521,
'colsample_bynode': 0.7381078096942799}. Best is trial 0 with value:
240.4542306523302.
```

```
[0]     validation_0-rmse:1219.91416
[10]    validation_0-rmse:644.55055
[20]    validation_0-rmse:367.80380
[30]    validation_0-rmse:260.70752
[40]    validation_0-rmse:233.53132
[50]    validation_0-rmse:230.75151
[60]    validation_0-rmse:234.67735
[70]    validation_0-rmse:237.72813
[80]    validation_0-rmse:241.07592
[90]    validation_0-rmse:242.61972
[100]   validation_0-rmse:243.97176
[110]   validation_0-rmse:245.72842
[116]   validation_0-rmse:246.68532
```

```
[I 2024-09-24 11:48:41,713] Trial 1 finished with value: 246.6853195141644 and
parameters: {'n_estimators': 117, 'learning_rate': 0.06336642478270658,
'max_depth': 9, 'subsample': 0.7504524840733682, 'colsample_bytree':
0.8134763132986811, 'min_child_weight': 24, 'reg_lambda': 0.001775010118255181,
'colsample_bynode': 0.6863293797054588}. Best is trial 0 with value:
240.4542306523302.
```

```
[0]     validation_0-rmse:1253.56326
[10]    validation_0-rmse:859.14330
[20]    validation_0-rmse:598.74879
[30]    validation_0-rmse:433.19970
[40]    validation_0-rmse:331.73284
[50]    validation_0-rmse:277.25611
[60]    validation_0-rmse:250.75663
[70]    validation_0-rmse:238.71044
[80]    validation_0-rmse:233.93337
[90]    validation_0-rmse:232.92849
[100]   validation_0-rmse:232.86181
```

```
[107]	validation_0-rmse:233.46253
```

```
[I 2024-09-24 11:48:46,185] Trial 2 finished with value: 233.462525330446 and
parameters: {'n_estimators': 108, 'learning_rate': 0.03845215473666802,
'max_depth': 6, 'subsample': 0.8999715997177624, 'colsample_bytree':
0.7783806308728796, 'min_child_weight': 18, 'reg_lambda': 0.001613420081838701,
'colsample_bynode': 0.4384887115389177}. Best is trial 2 with value:
233.462525330446.
```

```
[0]	validation_0-rmse:1214.18893
[10]	validation_0-rmse:608.52941
[20]	validation_0-rmse:340.83563
[30]	validation_0-rmse:247.39462
[40]	validation_0-rmse:230.81555
[50]	validation_0-rmse:232.67419
[60]	validation_0-rmse:237.61236
[70]	validation_0-rmse:241.09514
[80]	validation_0-rmse:243.97546
[90]	validation_0-rmse:245.64461
[100]	validation_0-rmse:246.20660
[110]	validation_0-rmse:247.69346
[120]	validation_0-rmse:249.66427
[130]	validation_0-rmse:250.56743
[140]	validation_0-rmse:251.78234
[150]	validation_0-rmse:252.34130
[154]	validation_0-rmse:252.59289
```

```
[I 2024-09-24 11:48:56,429] Trial 3 finished with value: 252.59288638374716 and
parameters: {'n_estimators': 155, 'learning_rate': 0.06802861771440158,
'max_depth': 13, 'subsample': 0.7516161029943194, 'colsample_bytree':
0.7814968722273645, 'min_child_weight': 23, 'reg_lambda': 0.0032059966122648595,
'colsample_bynode': 0.32865053937889244}. Best is trial 2 with value:
233.462525330446.
```

```
[0]	validation_0-rmse:1286.90869
[10]	validation_0-rmse:1136.53274
[20]	validation_0-rmse:1004.91342
[30]	validation_0-rmse:889.01716
[40]	validation_0-rmse:786.65850
[50]	validation_0-rmse:696.85750
[60]	validation_0-rmse:618.87901
[70]	validation_0-rmse:551.09057
[80]	validation_0-rmse:492.85614
[90]	validation_0-rmse:442.86184
[100]	validation_0-rmse:399.77178
[110]	validation_0-rmse:363.69019
[120]	validation_0-rmse:333.34374
[130]	validation_0-rmse:308.54605
[140]	validation_0-rmse:288.33606
[150]	validation_0-rmse:272.27979
```

```
[160]	validation_0-rmse:259.71937
[164]	validation_0-rmse:255.63662
```

[I 2024-09-24 11:49:07,124] Trial 4 finished with value: 255.6366163872131 and parameters: {'n_estimators': 165, 'learning_rate': 0.012356757631156543, 'max_depth': 10, 'subsample': 0.6827681148108359, 'colsample_bytree': 0.7852015432048611, 'min_child_weight': 11, 'reg_lambda': 0.12425090400936481, 'colsample_bynode': 0.7218020205445259}. Best is trial 2 with value: 233.462525330446.

```
[0]	validation_0-rmse:1252.64047
[10]	validation_0-rmse:847.90351
[20]	validation_0-rmse:581.47595
[30]	validation_0-rmse:413.49877
[40]	validation_0-rmse:314.28253
[50]	validation_0-rmse:262.65954
[60]	validation_0-rmse:239.05415
[70]	validation_0-rmse:230.94231
[80]	validation_0-rmse:229.32346
[90]	validation_0-rmse:230.41550
[100]	validation_0-rmse:232.08427
[101]	validation_0-rmse:232.22311
```

[I 2024-09-24 11:49:13,052] Trial 5 finished with value: 232.2231105679051 and parameters: {'n_estimators': 102, 'learning_rate': 0.03898328580347557, 'max_depth': 10, 'subsample': 0.9198347778200742, 'colsample_bytree': 0.6778891268431566, 'min_child_weight': 19, 'reg_lambda': 0.0063891513721647825, 'colsample_bynode': 0.3727824150583263}. Best is trial 5 with value: 232.2231105679051.

```
[0]	validation_0-rmse:1239.17141
[10]	validation_0-rmse:753.46616
[20]	validation_0-rmse:472.49893
[30]	validation_0-rmse:320.92911
[40]	validation_0-rmse:253.96994
[50]	validation_0-rmse:232.40284
[60]	validation_0-rmse:230.04406
[70]	validation_0-rmse:232.47649
[80]	validation_0-rmse:236.12491
[90]	validation_0-rmse:238.93982
[100]	validation_0-rmse:241.27489
[110]	validation_0-rmse:243.61790
[120]	validation_0-rmse:245.10310
[130]	validation_0-rmse:245.96747
[140]	validation_0-rmse:246.79154
[150]	validation_0-rmse:247.75851
[160]	validation_0-rmse:248.38997
[170]	validation_0-rmse:249.15604
[180]	validation_0-rmse:249.77817
[190]	validation_0-rmse:250.44421
```

```
[200]	validation_0-rmse:250.87539
[210]	validation_0-rmse:250.95929
[220]	validation_0-rmse:251.20990
[230]	validation_0-rmse:251.56885
[232]	validation_0-rmse:251.62709
```

```
[I 2024-09-24 11:49:27,696] Trial 6 finished with value: 251.6270851868489 and
parameters: {'n_estimators': 233, 'learning_rate': 0.048894822838632265,
'max_depth': 13, 'subsample': 0.8208158344870562, 'colsample_bytree':
0.929577476968376, 'min_child_weight': 14, 'reg_lambda': 0.004790998457743749,
'colsample_bynode': 0.4324033920690218}. Best is trial 5 with value:
232.2231105679051.
```

```
[0]	validation_0-rmse:1277.87982
[10]	validation_0-rmse:1052.31689
[20]	validation_0-rmse:867.57537
[30]	validation_0-rmse:718.70736
[40]	validation_0-rmse:598.00493
[50]	validation_0-rmse:502.10793
[60]	validation_0-rmse:425.95197
[70]	validation_0-rmse:367.04671
[80]	validation_0-rmse:323.18008
[90]	validation_0-rmse:291.25707
[100]	validation_0-rmse:268.61306
[110]	validation_0-rmse:252.67430
[120]	validation_0-rmse:242.73070
[130]	validation_0-rmse:236.89599
[140]	validation_0-rmse:233.06380
[150]	validation_0-rmse:231.08461
[160]	validation_0-rmse:230.51785
[170]	validation_0-rmse:230.53107
[180]	validation_0-rmse:231.13444
[190]	validation_0-rmse:231.70764
[200]	validation_0-rmse:232.42149
[210]	validation_0-rmse:233.39674
[220]	validation_0-rmse:234.42151
[230]	validation_0-rmse:235.45016
[240]	validation_0-rmse:236.30730
[250]	validation_0-rmse:236.99700
[260]	validation_0-rmse:237.68112
[270]	validation_0-rmse:238.30967
[280]	validation_0-rmse:238.87732
```

```
[I 2024-09-24 11:49:40,321] Trial 7 finished with value: 238.8773185687529 and
parameters: {'n_estimators': 281, 'learning_rate': 0.0194421159441089,
'max_depth': 8, 'subsample': 0.6566370142696688, 'colsample_bytree':
0.6308228955349335, 'min_child_weight': 19, 'reg_lambda': 0.006385061674934632 5,
'colsample_bynode': 0.7033495756734451}. Best is trial 5 with value:
232.2231105679051.
```

```
[0]	validation_0-rmse:1277.89760
[10]	validation_0-rmse:1053.30713
[20]	validation_0-rmse:869.62925
[30]	validation_0-rmse:720.39220
[40]	validation_0-rmse:600.25213
[50]	validation_0-rmse:505.78286
[60]	validation_0-rmse:430.77242
[70]	validation_0-rmse:373.52746
[80]	validation_0-rmse:330.66272
[90]	validation_0-rmse:298.85373
[100]	validation_0-rmse:276.20379
[110]	validation_0-rmse:260.36905
[120]	validation_0-rmse:250.29184
[130]	validation_0-rmse:243.09567
[140]	validation_0-rmse:238.82733
[150]	validation_0-rmse:235.85489
[160]	validation_0-rmse:234.66632
[170]	validation_0-rmse:234.00208
[180]	validation_0-rmse:233.59691
[190]	validation_0-rmse:233.59890
[200]	validation_0-rmse:233.81074
[210]	validation_0-rmse:234.02590
[220]	validation_0-rmse:234.43607
[230]	validation_0-rmse:234.74917
[240]	validation_0-rmse:234.86610
[250]	validation_0-rmse:235.06810
[260]	validation_0-rmse:235.42442
[270]	validation_0-rmse:236.37430
[280]	validation_0-rmse:236.96884
```

```
[I 2024-09-24 11:49:51,034] Trial 8 finished with value: 236.96883465833656 and
parameters: {'n_estimators': 281, 'learning_rate': 0.019578082903368696,
'max_depth': 7, 'subsample': 0.770612513098197, 'colsample_bytree':
0.5130463434099811, 'min_child_weight': 19, 'reg_lambda': 0.3163695294183683,
'colsample_bynode': 0.47316206000267463}. Best is trial 5 with value:
232.2231105679051.
```

```
[0]	validation_0-rmse:1281.27511
[10]	validation_0-rmse:1083.43112
[20]	validation_0-rmse:918.75003
[30]	validation_0-rmse:781.74990
[40]	validation_0-rmse:666.89364
[50]	validation_0-rmse:571.88218
[60]	validation_0-rmse:495.79232
[70]	validation_0-rmse:434.23070
[80]	validation_0-rmse:384.38070
[90]	validation_0-rmse:346.17352
[100]	validation_0-rmse:316.29330
[110]	validation_0-rmse:293.47128
```

```
[120]	validation_0-rmse:276.46978
[130]	validation_0-rmse:264.77171
[140]	validation_0-rmse:256.39790
[150]	validation_0-rmse:250.86792
[160]	validation_0-rmse:246.91602
[170]	validation_0-rmse:243.75566
[176]	validation_0-rmse:242.33551
```

```
[I 2024-09-24 11:49:58,075] Trial 9 finished with value: 242.3355072009143 and
parameters: {'n_estimators': 177, 'learning_rate': 0.017290014583136143,
'max_depth': 6, 'subsample': 0.6225581875175491, 'colsample_bytree':
0.3064463907706278, 'min_child_weight': 23, 'reg_lambda': 0.08435051957284508,
'colsample_bynode': 0.5699923760121177}. Best is trial 5 with value:
232.2231105679051.
```

[222]:
```
best_params_optuna = study.best_params
print(f"Best parameters found with Optuna: {best_params_optuna}")
```

```
Best parameters found with Optuna: {'n_estimators': 102, 'learning_rate':
0.03898328580347557, 'max_depth': 10, 'subsample': 0.9198347778200742,
'colsample_bytree': 0.6778891268431566, 'min_child_weight': 19, 'reg_lambda':
0.0063891513721647825, 'colsample_bynode': 0.3727824150583263}
```

[224]:
```
final_model = xgb.XGBRegressor(**best_params_optuna)
final_model.fit(X_subsample, y_subsample, verbose=True)
```

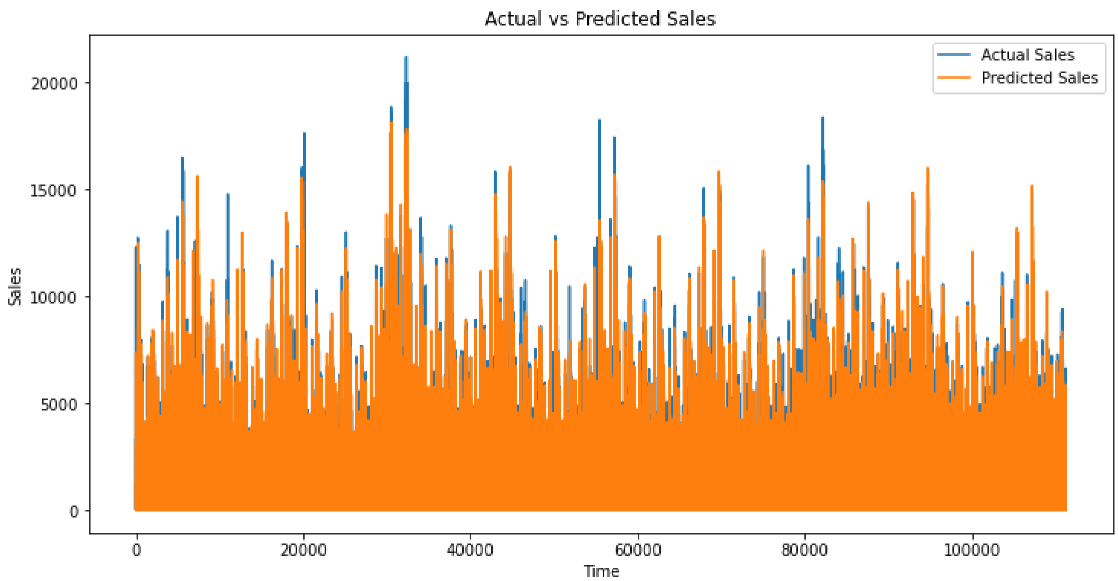
[224]:
```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=0.3727824150583263,
             colsample_bytree=0.6778891268431566, device=None,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, feature_types=None, gamma=None, grow_policy=None,
             importance_type=None, interaction_constraints=None,
             learning_rate=0.03898328580347557, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=10, max_leaves=None,
             min_child_weight=19, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=102, n_jobs=None,
             num_parallel_tree=None, random_state=None, …)
```

[231]:
```
y_pred = final_model.predict(X_sub_val)
rmse = mean_squared_error(y_sub_val, y_pred, squared=False)

# Calcular RMSE y RMSLE
def rmsle(y_true, y_pred):
    return np.sqrt(np.mean(np.square(np.log1p(y_pred) - np.log1p(y_sub_val))))

rmse = np.sqrt(mean_squared_error(y_sub_val, y_pred))
rmsle_score = rmsle(y_sub_val, y_pred)
```

```
print("RMSE:", rmse)
print("RMSLE:", rmsle_score)
```

```
RMSE: 192.53072440343328
RMSLE: 1.0988500778621835
```

[232]:
```
plt.figure(figsize=(12, 6))
plt.plot(y_sub_val.values, label='Actual Sales')
plt.plot(y_pred, label='Predicted Sales')
plt.xlabel('Time')
plt.ylabel('Sales')
plt.title('Actual vs Predicted Sales')
plt.legend()
plt.show()
```



[233]:
```
tabla_nueva = pd.read_csv('C:/Users/User/OneDrive - Universidad Internacional␣
↪del Ecuador/Escritorio/Master Primer Semestre/Software for IA/Project 1/test.
↪csv', parse_dates=['date'])
tabla_nueva
```

[233]:

	id	date	store_nbr	family	onpromotion
0	3000888	2017-08-16	1	AUTOMOTIVE	0
1	3000889	2017-08-16	1	BABY CARE	0
2	3000890	2017-08-16	1	BEAUTY	2
3	3000891	2017-08-16	1	BEVERAGES	20
4	3000892	2017-08-16	1	BOOKS	0
…	…	…	…	…	…

```
28507  3029395 2017-08-31          9                       POULTRY            1
28508  3029396 2017-08-31          9                PREPARED FOODS            0
28509  3029397 2017-08-31          9                       PRODUCE            1
28510  3029398 2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9
28511  3029399 2017-08-31          9                       SEAFOOD            0

[28512 rows x 5 columns]
```

[234]:
```
y_test_pred = final_model.predict(test_data_encoded)
```

[243]:
```
test_data_encoded
```

[243]:
```
         store_nbr  family  onpromotion  city  state  type  cluster  \
2780316          0       0            0    18     12     3        4
2780317          0       1            0    18     12     3        4
2780318          0       2            2    18     12     3        4
2780319          0       3           20    18     12     3        4
2780320          0       4            0    18     12     3        4
…              …     …          …    …     …    …      …
2808823         53      28            1    18     12     1       13
2808824         53      29            0    18     12     1       13
2808825         53      30            1    18     12     1       13
2808826         53      31            9    18     12     1       13
2808827         53      32            0    18     12     1       13

         dcoilwtico  weekday  year  …  sales_lag_14  sales_lag_30  \
2780316       46.80        2  2017  …           4.0      2.000000
2780317       46.80        2  2017  …           0.0      0.000000
2780318       46.80        2  2017  …           2.0      5.000000
2780319       46.80        2  2017  …        2645.0   2381.000000
2780320       46.80        2  2017  …           0.0      1.000000
…               …      …   …  …           …           …
2808823       47.26        3  2017  …           0.0    570.196000
2808824       47.26        3  2017  …           0.0     50.462997
2808825       47.26        3  2017  …           0.0   2470.461000
2808826       47.26        3  2017  …           0.0    203.000000
2808827       47.26        3  2017  …           0.0     19.316000

         sales_roll_mean_7  sales_roll_mean_30  sales_ewm_alpha_095_lag_7  \
2780316           7.028078            6.576761               6.857370e+00
2780317          -0.780095            1.424639               0.000000e+00
2780318           2.553573           -0.660333               3.907132e+00
2780319        1755.320339         2069.534184               2.315387e+03
2780320           1.040751            3.134128               1.855469e-12
…                      …                  …                        …
2808823           0.000000          445.950872               4.307176e+02
2808824           0.000000          114.808586               1.525120e+02
```

```
2808825          0.000000        1662.489077             2.366993e+03
2808826          0.000000         155.241974             1.240873e+02
2808827          0.000000          23.798672             1.605715e+01

         sales_ewm_alpha_095_lag_30  sales_ewm_alpha_09_lag_7  \
2780316                    2.009781              6.728925e+00
2780317                    0.000000              0.000000e+00
2780318                    4.857144              3.827138e+00
2780319                 2318.700769              2.317378e+03
2780320                    0.950238              9.000000e-10
…                               …                       …
2808823                  565.318991              4.239357e+02
2808824                   51.246066              1.504432e+02
2808825                 2423.706728              2.315776e+03
2808826                  195.609900              1.272182e+02
2808827                   18.970154              1.612746e+01

         sales_ewm_alpha_09_lag_30  sales_ewm_alpha_08_lag_7  \
2780316                   2.038480              6.511056e+00
2780317                   0.000000             8.942588e-322
2780318                   4.727317              3.698718e+00
2780319                2262.147948              2.312079e+03
2780320                   0.901801              4.096000e-07
…                              …                       …
2808823                 560.611289              4.122763e+02
2808824                  52.444705              1.462783e+02
2808825                2378.620204              2.217030e+03
2808826                 188.128407              1.334267e+02
2808827                  18.664387              1.630320e+01

         sales_ewm_alpha_08_lag_30
2780316               2.151308e+00
2780317             1.064056e-305
2780318               4.501235e+00
2780319               2.166968e+03
2780320               8.128513e-01
…                              …
2808823               5.514217e+02
2808824               5.602257e+01
2808825               2.292419e+03
2808826               1.728145e+02
2808827               1.816738e+01

[28512 rows x 25 columns]
```

[235]:
```
tabla_nueva['sales'] = y_test_pred
```

[238]:
```
display = tabla_nueva.copy()
stores = display.groupby(['date', 'store_nbr'], as_index=False)['sales'].sum()
```

[240]:
```
px.line(stores, x = "date", y= "sales", color = "store_nbr", title = "Daily␣
↪total sales of the stores")
```

[244]:
```
final_model.save_model("xgboost_model_sales.json")
```

[242]:
```
test_data_encoded.to_csv('Test_sales.csv', index=False)
```

```
import pandas as pd
from fastapi import FastAPI, Form
from starlette.responses import HTMLResponse
from fastapi.staticfiles import StaticFiles
import plotly.express as px
import plotly.io as pio
import xgboost as xgb
import random


app = FastAPI()
app.mount("/static", StaticFiles(directory="static"), name="static")


loaded_model_sales = xgb.XGBRegressor()
loaded_model_sales.load_model("xgboost_model_sales.json")

loaded_model_transactions = xgb.XGBRegressor()
loaded_model_transactions.load_model("xgboost_model_transactions.json")


test_dataset_sales = pd.read_csv('Test_sales.csv')
test_dataset_transactions = pd.read_csv('Test_Transactions.csv')
tabla_nueva = pd.read_csv('test.csv')
```

```
}}

predictions_sales = loaded_model_sales.predict(test_dataset_sales)
tabla_nueva['sales'] = predictions_sales

stores = tabla_nueva.groupby(['date', 'store_nbr'], as_index=False)['sales'].sum()
test_dataset_transactions.insert(1, 'sales', predictions_sales)

predictions_transactions = loaded_model_transactions.predict(test_dataset_transactions)
tabla_nueva['transactions'] = predictions_transactions

tabla_plot = tabla_nueva.groupby(['date', 'store_nbr'])['transactions'].mean().reset_index()
fechas_unicas = stores['date'].unique()
tiendas_unicas = stores['store_nbr'].unique()



@app.get("/", response_class=HTMLResponse)
def render_menu():
  html_content = '''
  <html>
    <head>
      <title>Sales and Transactions Dashboard</title>
      <style>
        body {{
          text-align: center;
          font-family: Arial, sans-serif;
          background-color: #f0f0f0;
        }}
```

```
img {{
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 200px;
}}
.menu {{
  margin-top: 20px;
  margin-bottom: 30px;
}}
.content {{
  margin-top: 30px;
}}
.center {{
  margin-left: auto;
  margin-right: auto;
  width: 80%;
}}
h1 {{
  text-align: center;
}}
.prediction {{
  margin-top: 20px;
  font-size: 18px;
  color: green;
}}
```

```
.container {{
    background-color: white;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 400px;
    margin: 20px auto;
}}
table {{
    width: 100%;
    margin: 10px 0;
}}
td {{
    padding: 8px;
    text-align: left;
}}
input, select {{
    width: calc(100% - 16px);
    padding: 8px;
    margin: 4px 0;
    border: 1px solid #ddd;
    border-radius: 4px;
}}
.button {{
    width: 100%;
    padding: 10px;
```

```
        background-color: #4CAF50;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }}
    .button:hover {{
        background-color: #45a049;
    }}
    .prediction-result {{
        margin-top: 20px;
        padding: 10px;
        background-color: #e7f3e7;
        border: 1px solid #d4edda;
        border-radius: 4px;
        color: #155724;
        display: none;
        text-align: center;
    }}
</style>
<script>
    function predictSales() {{
        var date = document.getElementById('sales-date').value;
        var store = document.getElementById('sales-store').value;
        var salesValue = document.getElementById('sales-' + date + '-' + store).innerText;
```

```
                var prediction = "On " + date + " the store " + store + " sold " + salesValue + "
products.";
                document.getElementById('sales-prediction').innerText = prediction;
            }}


            function predictTransactions() {{
                var date = document.getElementById('transactions-date').value;
                var store = document.getElementById('transactions-store').value;
                var transactionsValue = document.getElementById('transactions-' + date + '-' +
store).innerText;
                var prediction = "On " + date + " the store " + store + " got " + transactionsValue + "
transactions.";
                document.getElementById('transactions-prediction').innerText = prediction;
            }}
        </script>
    </head>
    <body>
        <img src="/static/Corporación_Favorita_Logo.png" alt="Dashboard Logo" />


        <h1>Sales and Transactions Dashboard</h1>


        <div id="xgboost" class="content" style="display:block;">
            <h2>Sales per Store</h2>
            <div class="center">
                {graph_sales}
            </div>
```

```
<label for="sales-date">Date:</label>
<select id="sales-date">
  {fechas_options}
</select>

<label for="sales-store">Store:</label>
<select id="sales-store">
  {tiendas_options}
</select>

<button onclick="predictSales()">Predict</button>
<div id="sales-prediction" class="prediction"></div>

<h2>Transactions per Store</h2>
<div class="center">
  {graph_transactions}
</div>

<label for="transactions-date">Date:</label>
<select id="transactions-date">
  {fechas_options}
</select>

<label for="transactions-store">Store:</label>
<select id="transactions-store">
  {tiendas_options}
```

```
            </select>


            <button onclick="predictTransactions()">Predict</button>
            <div id="transactions-prediction" class="prediction"></div>
        </div>


        <div class="container">
            <h1>Sales prediction by Product</h1>
            <form action="/predict" method="post" id="prediction-form">
                <table>
                    <tr>
                        <td><label for="store_nbr">Store Number:</label></td>
                        <td><input type="number" id="store_nbr" name="store_nbr" min="0"
max="53" value="0"></td>
                    </tr>
                    <tr>
                        <tr>
                            <td><label for="onpromotion">On Promotion:</label></td>
                            <td><input type="number" id="onpromotion" name="onpromotion"
min="0" max="646" value="0"></td>
                        </tr>
                    </tr>
                    <tr>
                        <td><label for="weekday">Week Day:</label></td>
                        <td><input type="number" id="weekday" name="weekday" min="0" max="6"
value="0"></td>
                    </tr>
```

```
      <tr>
        <td><label for="date">Date:</label></td>
        <td><input type="date" id="date" name="date"></td>
      </tr>
      <tr>
        <td><label for="payday">Pay day?:</label></td>
        <td>
          <select id="payday" name="payday">
            <option value="1">Yes</option>
            <option value="0">No</option>
          </select>
        </td>
      </tr>
      <tr>
        <td><label for="is_weekend">Is weekend?:</label></td>
        <td>
          <select id="is_weekend" name="is_weekend">
            <option value="1">Yes</option>
            <option value="0">No</option>
          </select>
        </td>
      </tr>
    </table>
    <button type="submit" class="button">Predict</button>
  </form>
  <div id="prediction-result" class="prediction-result"></div>
```

```
        </div>

        <script>
          document.getElementById('prediction-form').addEventListener('submit', async
function(event) {{
            event.preventDefault();
            const formData = new FormData(this);
            const response = await fetch('/predict', {{
              method: 'POST',
              body: formData
            }});
            const result = await response.text();
            document.getElementById('prediction-result').innerHTML = "The sales prediction
is: " + result;
            document.getElementById('prediction-result').style.display = 'block';
          }});
        </script>
      </body>
    </html>
    '''

    fig_sales = px.line(stores, x="date", y="sales", color="store_nbr")
    fig_sales.update_layout(width=1200, height=500)
    graph_sales = pio.to_html(fig_sales, full_html=False)

    fig_transactions = px.line(tabla_plot, x="date", y="transactions", color="store_nbr")
    fig_transactions.update_layout(width=1200, height=500)
```

```
    graph_transactions = pio.to_html(fig_transactions, full_html=False)


    fechas_options = ''.join([f'<option value="{fecha}">{fecha}</option>' for fecha in
fechas_unicas])
    tiendas_options = ''.join([f'<option value="{tienda}">{tienda}</option>' for tienda in
tiendas_unicas])


    sales_data = ''.join([f'<span id="sales-{row["date"]}-{row["store_nbr"]}"
style="display:none;">{row["sales"]}</span>'
                          for idx, row in stores.iterrows()])
    transactions_data = ''.join([f'<span id="transactions-{row["date"]}-{row["store_nbr"]}"
style="display:none;">{row["transactions"]}</span>'
                                 for idx, row in tabla_plot.iterrows()])


    return HTMLResponse(content=html_content.format(
        graph_sales=graph_sales,
        graph_transactions=graph_transactions,
        fechas_options=fechas_options,
        tiendas_options=tiendas_options
    ) + sales_data + transactions_data)


@app.post("/predict")
async def predict_sales(
    store_nbr: int = Form(...),
    onpromotion: int = Form(...),
    weekday: int = Form(...),
    date: str = Form(...),
```

```
    payday: int = Form(...),
    is_weekend: int = Form(...)
):

    date_obj = pd.to_datetime(date)
    year = date_obj.year
    month = date_obj.month
    day = date_obj.day



    fila_test1 = test_dataset_sales.iloc[random.randint(0, 28500)].copy()
    fila_test1['store_nbr'] = store_nbr
    fila_test1['family'] = random.randint(0, 32)
    fila_test1['onpromotion'] = onpromotion
    fila_test1['city'] = random.randint(0, 21)
    fila_test1['state'] = random.randint(0, 15)
    fila_test1['type'] = random.randint(0, 4)
    fila_test1['cluster'] = random.randint(0, 16)
    fila_test1['dcoilwtico'] = random.uniform(40, 50)
    fila_test1['weekday'] = weekday
    fila_test1['year'] = year
    fila_test1['month'] = month
    fila_test1['day'] = day
    fila_test1['payday'] = payday
    fila_test1['is_weekend'] = is_weekend
```

```
    fila_test = fila_test1.values.reshape(1, -1)
    prediccion_fila = loaded_model_sales.predict(fila_test)



    return (str(prediccion_fila[0]) + " And the number of On promotion products were: " + str(fila_test1['onpromotion']))
```

```
import pandas as pd

# Load datasets
train_df = pd.read_csv('/content/train.csv')
stores_df = pd.read_csv('/content/stores.csv')
oil_df = pd.read_csv('/content/oil.csv')
holidays_events_df = pd.read_csv('/content/holidays_events.csv')
transactions_df = pd.read_csv('/content/transactions.csv')

train_df['date'] = pd.to_datetime(train_df['date'])
oil_df['date'] = pd.to_datetime(oil_df['date'])
holidays_events_df['date'] = pd.to_datetime(holidays_events_df['date'])
transactions_df['date'] = pd.to_datetime(transactions_df['date'])

# Merge datasets
train_df = train_df.merge(stores_df, on='store_nbr', how='left')

train_df = train_df.merge(oil_df, on='date', how='left')

train_df = train_df.merge(holidays_events_df, on='date', how='left')

train_df = train_df.merge(transactions_df, on=['date', 'store_nbr'], how='left')

train_df['dcoilwtico'] = train_df['dcoilwtico'].fillna(method='ffill')
train_df['type_y'] = train_df['type_y'].fillna('not-holiday')


# Feature engineering
train_df['day_of_week'] = train_df['date'].dt.dayofweek


train_df['lagged_sales'] = train_df.groupby(['store_nbr', 'family'])['sales'].shift(1)


# Finalizing the training DataFrame
train_df = train_df.drop(columns=['transactions'])
train_df = train_df.drop(columns=['transferred', 'description', 'locale', 'locale_name','city','state','type_x'], errors='ignore')
```

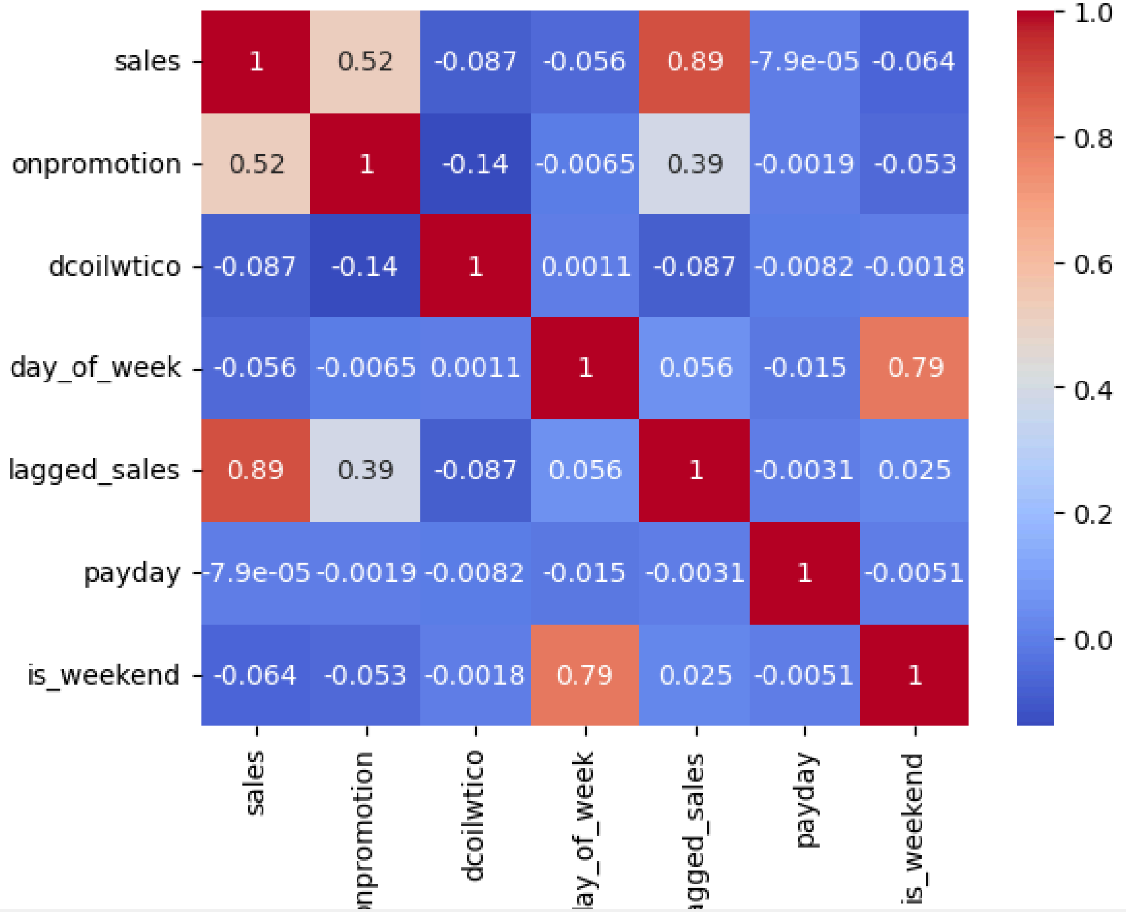
```
train_df['dcoilwtico'] = train_df['dcoilwtico'].ffill()
train_df['lagged_sales'] = train_df['lagged_sales'].ffill()
nan_counts = train_df.isna().sum()

#removing noise
start_date = '2016-04-01'
end_date = '2016-05-31'
train_df = train_df[(train_df['date'] < start_date) | (train_df['date'] > end_date)]

# adding a function to identify paydays
def is_payday(date):
    if date.day == 15 or (date.day == 1 and date != date + pd.offsets.MonthEnd(0)):
        return 1
    else:
        return 0

train_df['payday'] = train_df['date'].apply(is_payday)
train_df = train_df[train_df['store_nbr'] == 1]


train_df.fillna(0, inplace=True)
train_df['month'] = train_df['date'].dt.month
train_df['day'] = train_df['date'].dt.day
train_df['is_weekend'] = train_df['day_of_week'].apply(lambda x: 1 if x >= 5 else 0)  # Saturday and Sunday

print(train_df)
```

```
...          ...        ...        ...                  ...
3052594  2999134 2017-08-15          1              POULTRY
3052595  2999135 2017-08-15          1       PREPARED FOODS
3052596  2999136 2017-08-15          1              PRODUCE
```

```
...              ...           ...     ...         ...      ...          ...
3052594   234.892000             0      13       47.57  Holiday            1
3052595    42.822998             0      13       47.57  Holiday            1
3052596  2240.230000             7      13       47.57  Holiday            1
3052597     0.000000             0      13       47.57  Holiday            1
3052598    22.487000             0      13       47.57  Holiday            1

         lagged_sales  payday  month  day  is_weekend
0               0.000       1      1    1           0
1               0.000       1      1    1           0
2               0.000       1      1    1           0
3               0.000       1      1    1           0
4               0.000       1      1    1           0
...               ...     ...    ...  ...         ...
3052594       270.047       1      8   15           0
3052595        72.004       1      8   15           0
3052596      2611.755       1      8   15           0
3052597         0.000       1      8   15           0
3052598        14.129       1      8   15           0

[54384 rows x 15 columns]
<ipython-input-17-60443828ebe3>:59: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
  train_df.fillna(0, inplace=True)
<ipython-input-17-60443828ebe3>:60: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
  train_df['month'] = train_df['date'].dt.month
<ipython-input-17-60443828ebe3>:61: SettingWithCopyWarning:
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers
  train_df['is_weekend'] = train_df['day_of_week'].apply(lambda x: 1 if x >= 5 else 0)  # Saturday and Sunday
```

```
import seaborn as sns
import matplotlib.pyplot as plt

#correlation matrix
corr_matrix = train_df[['sales', 'onpromotion', 'dcoilwtico', 'day_of_week', 'lagged_sales', 'payday', 'is_weekend']].corr()

sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
plt.show()
```




```
from statsmodels.tsa.stattools import adfuller

#  ADF test to get p value
subset = train_df['sales'].iloc[:10000]
adf_result = adfuller(subset)

print(f'ADF Statistic: {adf_result[0]}')
print(f'p-value: {adf_result[1]}')
```

```
ADF Statistic: -13.194351039463504
p-value: 1.1272777578336565e-24
```

```
train_df['sales_seasonal_diff'] = train_df['sales'].diff(12).dropna()

adf_result_seasonal = adfuller(train_df['sales_seasonal_diff'].dropna().iloc[:10000])  # Adjust the number of rows as needed
print(f'ADF Statistic (after seasonal differencing): {adf_result_seasonal[0]}')
print(f'p-value (after seasonal differencing): {adf_result_seasonal[1]}')
```

```
ADF Statistic (after seasonal differencing): -23.435167848963008
p-value (after seasonal differencing): 0.0
```

```
train_df['date'] = pd.to_datetime(train_df['date'])  # Convert to datetime if not already done
train_df.index = pd.date_range(start='2013-01-01', periods=len(train_df), freq='D')

print(train_df)
```

```
                 id       date  store_nbr                      family  \
2013-01-01        0 2013-01-01          1                  AUTOMOTIVE
2013-01-02        1 2013-01-01          1                   BABY CARE
2013-01-03        2 2013-01-01          1                      BEAUTY
2013-01-04        3 2013-01-01          1                   BEVERAGES
2013-01-05        4 2013-01-01          1                       BOOKS
...             ...        ...        ...                         ...
2161-11-20  2999134 2017-08-15          1                     POULTRY
2161-11-21  2999135 2017-08-15          1              PREPARED FOODS
2161-11-22  2999136 2017-08-15          1                     PRODUCE
2161-11-23  2999137 2017-08-15          1  SCHOOL AND OFFICE SUPPLIES
2161-11-24  2999138 2017-08-15          1                     SEAFOOD

                 sales  onpromotion  cluster  dcoilwtico   type_y  \
2013-01-01    0.000000            0       13        0.00  Holiday
2013-01-02    0.000000            0       13        0.00  Holiday
2013-01-03    0.000000            0       13        0.00  Holiday
2013-01-04    0.000000            0       13        0.00  Holiday
2013-01-05    0.000000            0       13        0.00  Holiday
...                ...          ...      ...         ...      ...
2161-11-20  234.892000            0       13       47.57  Holiday
```

```
2161-11-21    42.822998           0       13       47.57  Holiday
2161-11-22  2240.230000           7       13       47.57  Holiday
2161-11-23     0.000000           0       13       47.57  Holiday
2161-11-24    22.487000           0       13       47.57  Holiday

            day_of_week  lagged_sales  payday  month  day  is_weekend  \
2013-01-01            1         0.000       1      1    1           0
2013-01-02            1         0.000       1      1    1           0
2013-01-03            1         0.000       1      1    1           0
2013-01-04            1         0.000       1      1    1           0
2013-01-05            1         0.000       1      1    1           0
...                 ...           ...     ...    ...  ...         ...
2161-11-20            1       270.047       1      8   15           0
2161-11-21            1        72.004       1      8   15           0
2161-11-22            1      2611.755       1      8   15           0
2161-11-23            1         0.000       1      8   15           0
2161-11-24            1        14.129       1      8   15           0

            sales_seasonal_diff
2013-01-01                  NaN
2013-01-02                  NaN
2013-01-03                  NaN
2013-01-04                  NaN
2013-01-05                  NaN
...                         ...
2161-11-20           201.892000
2161-11-21            42.822998
2161-11-22          2084.230000
2161-11-23            -9.000000
2161-11-24             5.487000

[54384 rows x 16 columns]
```

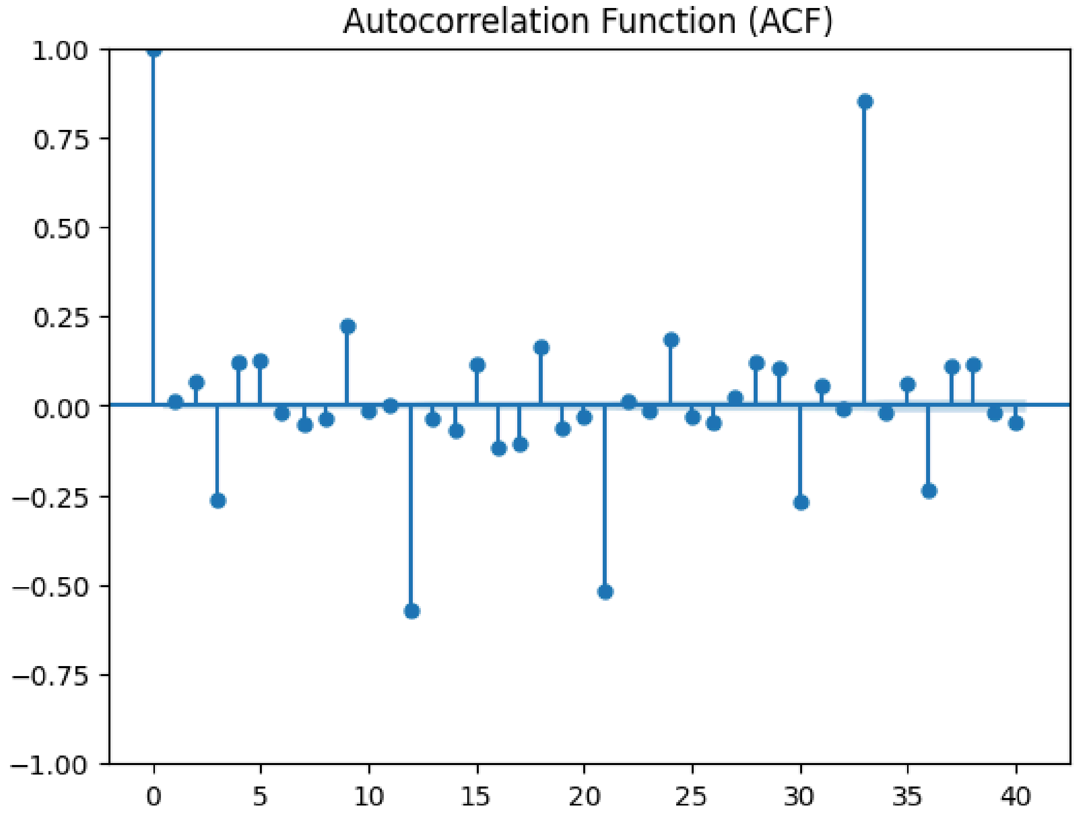
```
target = 'sales'
exog_vars = ['onpromotion', 'dcoilwtico', 'day_of_week', 'payday', 'lagged_sales', 'is_weekend']

y = train_df[target]
X = train_df[exog_vars]

# Split data into training and testing sets
train_size = int(len(y) * 0.8)
```

```
y_train, y_test = y[:train_size], y[train_size:]
X_train, X_test = X[:train_size], X[train_size:]

print("Target and exogenous features are ready for model training.")
```

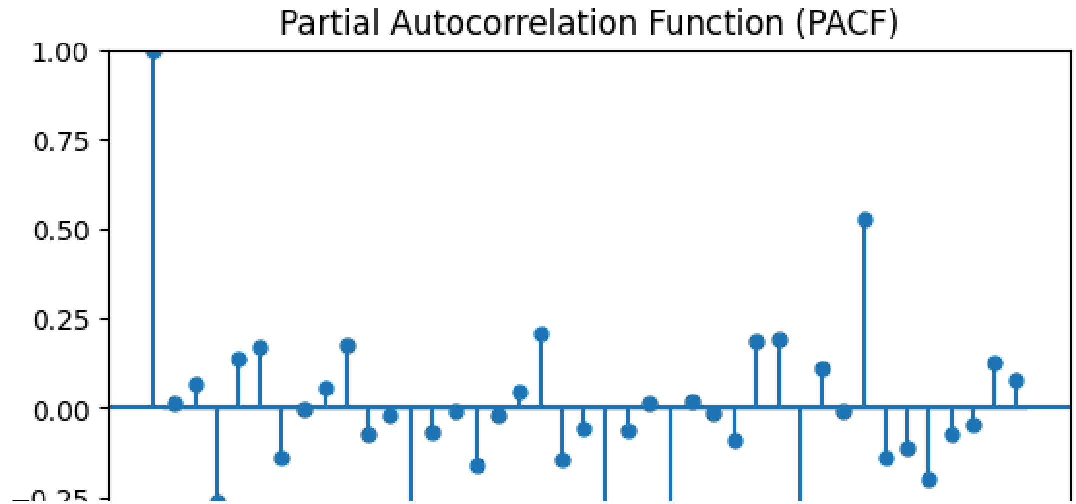
```
Target and exogenous features are ready for model training.
```

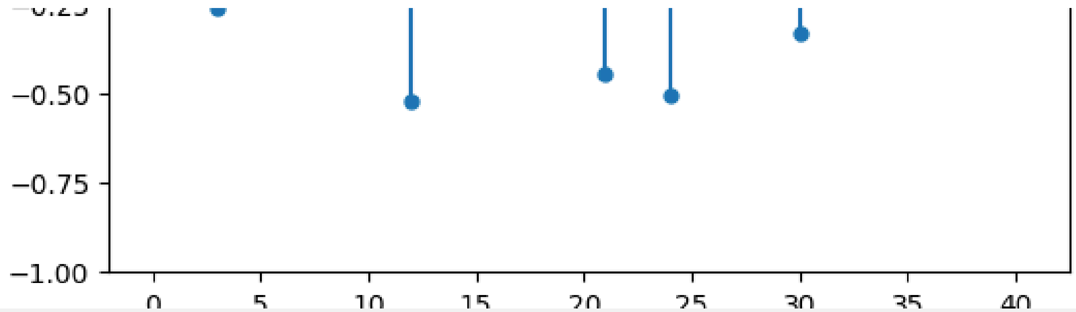
```
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

sales_diff_subset = train_df['sales_seasonal_diff'].dropna()

# Plot ACF and PACF for the subset of data
plt.figure(figsize=(12, 6))
plot_acf(sales_diff_subset, lags=40)
plt.title('Autocorrelation Function (ACF)')
plt.show()

plt.figure(figsize=(12, 6))
plot_pacf(sales_diff_subset, lags=40)
plt.title('Partial Autocorrelation Function (PACF)')
plt.show()
```

<Figure size 1200x600 with 0 Axes>



<Figure size 1200x600 with 0 Axes>






```
# Configure SARIMAX parameters

from statsmodels.tsa.statespace.sarimax import SARIMAX

p, d, q = 1, 0, 1
P, D, Q, S = 1, 1, 1, 12

sarimax_model = SARIMAX(y_train, exog=X_train, order=(p, d, q),
                        seasonal_order=(P, D, Q, S))

sarimax_results = sarimax_model.fit()

print(sarimax_results.summary())
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/base/model.py:607: ConvergenceWarning: Maximum Likelihood optimization failed to
  warnings.warn("Maximum Likelihood optimization failed to "
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                 sales   No. Observations:            43507
Model:             SARIMAX(1, 0, 1)x(1, 1, 1, 12)   Log Likelihood         -296369.528
Date:                            Thu, 26 Sep 2024   AIC                     592761.056
Time:                                    23:58:14   BIC                     592856.540
Sample:                                01-01-2013   HQIC                    592791.161
                                     - 02-13-2132
Covariance Type:                              opg
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
```

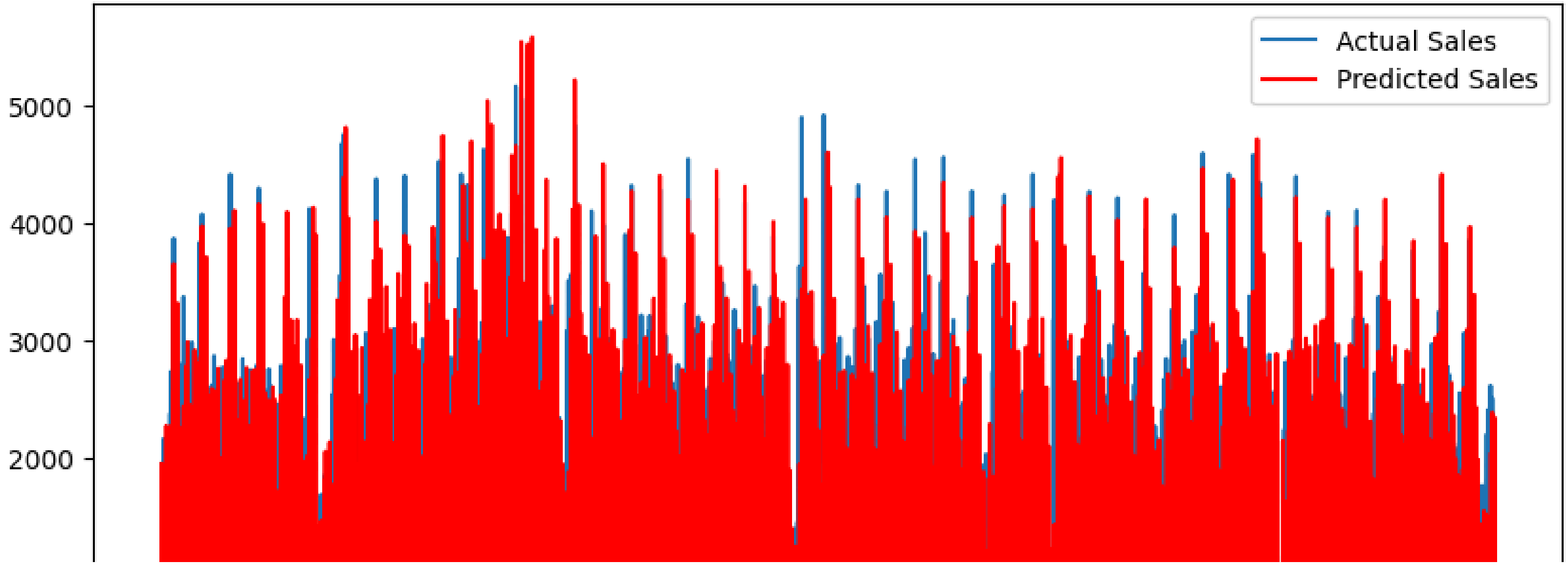
```
onpromotion      9.9681      0.062    161.799      0.000       9.847      10.089
dcoilwtico       0.1594      0.056      2.861      0.004       0.050       0.269
day_of_week    -23.3985      0.955    -24.503      0.000     -25.270     -21.527
payday          -5.4362      3.673     -1.480      0.139     -12.634       1.762
lagged_sales     0.8435      0.001    895.846      0.000       0.842       0.845
is_weekend      21.8836      4.571      4.788      0.000      12.925      30.842
ar.L1           -0.9889      0.005   -186.627      0.000      -0.999      -0.979
ma.L1            0.9916      0.005    215.432      0.000       0.983       1.001
ar.S.L12         0.0388      0.007      5.788      0.000       0.026       0.052
ma.S.L12        -1.0000      0.034    -29.349      0.000      -1.067      -0.933
sigma2        5.05e+04   1706.996     29.585      0.000    4.72e+04    5.38e+04
===================================================================================
Ljung-Box (L1) (Q):                   7.68   Jarque-Bera (JB):           2685034.59
Prob(Q):                              0.01   Prob(JB):                         0.00
Heteroskedasticity (H):               2.16   Skew:                             1.99
Prob(H) (two-sided):                  0.00   Kurtosis:                        41.28
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

```python
# Forecast sales on the test set
predictions = sarimax_results.predict(start=len(y_train), end=len(y_train) + len(y_test) - 1,
                                      exog=X_test)

print(predictions)
```

```
2132-02-14     -33.715938
2132-02-15     -41.796383
2132-02-16     -60.465189
2132-02-17     -32.562656
2132-02-18     -51.662643
                  ...
2161-11-20     271.063371
2161-11-21     105.233304
2161-11-22    2344.903722
2161-11-23      41.767920
2161-11-24      55.138267
```

```
Freq: D, Name: predicted_mean, Length: 10877, dtype: float64
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 5))
plt.plot(y_test.index, y_test, label='Actual Sales')
plt.plot(y_test.index, predictions, label='Predicted Sales', color='red')
plt.legend()
plt.show()
```



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

```
from fastapi import FastAPI
from fastapi.responses import HTMLResponse
from fastapi.staticfiles import StaticFiles
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pydantic import BaseModel


app = FastAPI()

app.mount("/static", StaticFiles(directory="static"), name="static")

#File path
train_df =
pd.read_csv(r'C:\Users\Surface\Downloads\store-sales-time-series-forecasti
ng\train.csv')
stores_df =
pd.read_csv(r'c:\Users\Surface\Downloads\store-sales-time-series-forecasti
ng\stores.csv')
oil_df =
pd.read_csv(r'C:\Users\Surface\Downloads\store-sales-time-series-forecasti
ng\oil.csv')
holidays_events_df =
pd.read_csv(r'c:\Users\Surface\Downloads\store-sales-time-series-forecasti
ng\holidays_events.csv')
transactions_df =
pd.read_csv(r'C:\Users\Surface\Downloads\store-sales-time-series-forecasti
ng\transactions.csv')

train_df['date'] = pd.to_datetime(train_df['date'])
oil_df['date'] = pd.to_datetime(oil_df['date'])
holidays_events_df['date'] = pd.to_datetime(holidays_events_df['date'])
transactions_df['date'] = pd.to_datetime(transactions_df['date'])

train_df = train_df.merge(stores_df, on='store_nbr', how='left')
train_df = train_df.merge(oil_df, on='date', how='left')
train_df = train_df.merge(holidays_events_df, on='date', how='left')
train_df = train_df.merge(transactions_df, on=['date', 'store_nbr'],
how='left')

# Fill missing values and  feature engineering
train_df['dcoilwtico'] = train_df['dcoilwtico'].fillna(method='ffill')
train_df['day_of_week'] = train_df['date'].dt.dayofweek
train_df['lagged_sales'] = train_df.groupby(['store_nbr',
'family'])['sales'].shift(1)
```

```
train_df['payday'] = train_df['date'].apply(lambda x: 1 if x.day == 1 or
x.day == 15 else 0)
train_df['is_weekend'] = train_df['day_of_week'].apply(lambda x: 1 if x >=
5 else 0)
train_df.fillna(0, inplace=True)

#removing noise
start_date = '2016-04-01'
end_date = '2016-05-31'
train_df = train_df[(train_df['date'] < start_date) | (train_df['date'] >
end_date)]
train_df = train_df[train_df['store_nbr'] == 1]

train_df['date'] = pd.to_datetime(train_df['date'])
train_df.index = pd.date_range(start='2013-01-01', periods=len(train_df),
freq='D')

#train_df = train_df.head(1000)

target = 'sales'
exog_vars = ['onpromotion', 'dcoilwtico', 'day_of_week', 'payday',
'lagged_sales', 'is_weekend']

y = train_df[target]
X = train_df[exog_vars]

train_size = int(len(y) * 0.8)
y_train, y_test = y[:train_size], y[train_size:]
X_train, X_test = X[:train_size], X[train_size:]

#  SARIMAX model
p, d, q = 1, 0, 1
P, D, Q, S = 1, 1, 1, 12

sarimax_model = SARIMAX(y_train, exog=X_train, order=(p, d, q),
                        seasonal_order=(P, D, Q, S))
sarimax_results = sarimax_model.fit()

@app.get("/", response_class=HTMLResponse)
def read_root():
    with open("index.html") as f:
        return f.read()

class SalesData(BaseModel):
    onpromotion: int
    dcoilwtico: float
    day_of_week: int
```

```
    payday: int
    lagged_sales: float
    is_weekend: int

@app.post("/predict/")
def predict_sales(data: SalesData):
    exog = pd.DataFrame({
        'onpromotion': [data.onpromotion],
        'dcoilwtico': [data.dcoilwtico],
        'day_of_week': [data.day_of_week],
        'payday': [data.payday],
        'lagged_sales': [data.lagged_sales],
        'is_weekend': [data.is_weekend]
    })

    # Predict sales using the SARIMAX model
    prediction = sarimax_results.predict(start=len(y_train),
end=len(y_train), exog=exog)[0]

    return {"predicted_sales": prediction}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sales Prediction</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
        }

        h1 {
            text-align: center;
            color: #333;
        }

        form {
            background-color: #fff;
            padding: 20px;
            border-radius: 5px;
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
```

```
            max-width: 400px;
            margin: 20px auto;
        }

        label {
            display: block;
            margin-bottom: 8px;
            color: #555;
        }

        input[type="number"] {
            width: 100%;
            padding: 8px;
            margin-bottom: 15px;
            border: 1px solid #ddd;
            border-radius: 4px;
        }

        button {
            background-color: #28a745;
            color: white;
            padding: 10px 15px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            width: 100%;
            font-size: 16px;
        }

        button:hover {
            background-color: #218838;
        }

        #result {
            text-align: center;
            font-size: 24px;
            margin-top: 20px;
            color: #333;
        }
    </style>
</head>
<body>
    <h1>Sales Prediction</h1>
    <form id="prediction-form">
        <label for="onpromotion">On Promotion:</label>
        <input type="number" id="onpromotion" name="onpromotion"
required><br>
```

```
        <label for="dcoilwtico">DCOILWTICO (Oil Price):</label>
        <input type="number" id="dcoilwtico" name="dcoilwtico" step="0.01"
required><br>

        <label for="day_of_week">Day of Week (Integer: 0 for Monday, 6 for
Sunday):</label>
        <input type="number" id="day_of_week" name="day_of_week" min="0"
max="6" required><br>

        <label for="payday">Payday (Integer: 1 for payday, 0 for
non-payday):</label>
        <input type="number" id="payday" name="payday" required><br>

        <label for="lagged_sales">Lagged Sales:</label>
        <input type="number" id="lagged_sales" name="lagged_sales"
step="0.01"><br>

        <label for="is_weekend">Is Weekend (Integer: 1 for weekend, 0
otherwise):</label>
        <input type="number" id="is_weekend" name="is_weekend"
required><br>

        <button type="submit">Predict</button>
    </form>


    <h2 id="result"></h2>

    <script>

document.getElementById("prediction-form").addEventListener("submit",
function(event) {
            event.preventDefault();

            const formData = new FormData(this);
            const data = {};
            formData.forEach((value, key) => {
                data[key] = value;
            });

            fetch("/predict/", {
                method: "POST",
                headers: {
                    "Content-Type": "application/json"
                },
                body: JSON.stringify(data)
```

```
            })
            .then(response => response.json())
            .then(data => {
                document.getElementById("result").innerText = `Predicted
Sales: ${data.predicted_sales}`;
            })
            .catch(error => {
                console.error("Error:", error);
            });
        });
    </script>
</body>
</html>
```