Store Sales - Time Series Forecasting

Use machine learning to predict grocery sales

Kaggle link: Store Sales - Time Series Forecasting

Context

Forecasts aren't just for meteorologists. Governments forecast economic growth. Scientists attempt to predict the future population. And businesses forecast product demand—a common task of professional data scientists. Forecasts are especially relevant to brick-and-mortar grocery stores, which must dance delicately with how much inventory to buy. Predict a little over, and grocers are stuck with overstocked, perishable goods. Guess a little under, and popular items quickly sell out, leading to lost revenue and upset customers. More accurate forecasting, thanks to machine learning, could help ensure retailers please customers by having just enough of the right products at the right time.

Current subjective forecasting methods for retail have little data to back them up and are unlikely to be automated. The problem becomes even more complex as retailers add new locations with unique needs, new products, ever-transitioning seasonal tastes, and unpredictable product marketing.

Potential Impact

If successful, you'll have flexed some new skills in a real world example. For grocery stores, more accurate forecasting can decrease food waste related to overstocking and improve customer satisfaction. The results of this ongoing competition, over time, might even ensure your local store has exactly what you need the next time you shop.

Dataset Description

In this competition, you will predict sales for the thousands of product families sold at Favorita stores located in Ecuador. The training data includes dates, store and product information, whether that item was being promoted, as well as the sales numbers. Additional files include supplementary information that may be useful in building your models.

File Descriptions and Data Field Information

train.csv The training data, comprising time series of features store_nbr, family, and onpromotion as well as the target sales. **store_nbr** identifies the store at which the products are sold. **family** identifies the type of product sold. **sales** gives the total sales for a product family at a particular store at a given date. Fractional values are possible since products can be sold in fractional units (1.5 kg of cheese, for instance, as opposed

to 1 bag of chips). **onpromotion** gives the total number of items in a product family that were being promoted at a store at a given date.

test.csv The test data, having the same features as the training data. You will predict the target sales for the dates in this file. The dates in the test data are for the 15 days after the last date in the training data.

sample_submission.csv A sample submission file in the correct format.

stores.csv Store metadata, including city, state, type, and cluster. cluster is a grouping of similar stores.

oil.csv Daily oil price. Includes values during both the train and test data timeframes. (Ecuador is an oil-dependent country and it's economical health is highly vulnerable to shocks in oil prices.)

holidays_events.csv

- Holidays and Events, with metadata
- NOTE: Pay special attention to the transferred column. A holiday that is transferred officially falls on that calendar day, but was moved to another date by the government. A transferred day is more like a normal day than a holiday. To find the day that it was actually celebrated, look for the corresponding row where type is Transfer. For example, the holiday Independencia de Guayaquil was transferred from 2012-10-09 to 2012-10-12, which means it was celebrated on 2012-10-12. Days that are type Bridge are extra days that are added to a holiday (e.g., to extend the break across a long weekend). These are frequently made up by the type Work Day which is a day not normally scheduled for work (e.g., Saturday) that is meant to payback the Bridge.
- Additional holidays are days added a regular calendar holiday, for example, as typically happens around Christmas (making Christmas Eve a holiday).

Additional Notes

- Wages in the public sector are paid every two weeks on the 15 th and on the last day of the month. Supermarket sales could be affected by this.
- A magnitude 7.8 earthquake struck Ecuador on April 16, 2016. People rallied in relief efforts donating water and other first need products which greatly affected supermarket sales for several weeks after the earthquake.

Importing all libraries

In [404…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import *
from sklearn.linear_model import *
from math import *
from sklearn.ensemble import *
from sklearn.feature_selection import *
```

```
from sklearn.feature_extraction import *
from sklearn.naive_bayes import *
from sklearn.discriminant_analysis import *
from sklearn.preprocessing import *
from sklearn.metrics import *
from sklearn.neighbors import *
from sklearn.cluster import *
from sklearn.svm import *
import warnings
warnings.filterwarnings("ignore")
```

Importing the training datasets

In [405…
```
df_train_1 = pd.read_csv("train1.csv")
df_train_1.head()
```

Out[405]:

	id	date	store_nbr	family	sales	onpromotion
0	0	1/1/13	1	AUTOMOTIVE	0.0	0
1	1	1/1/13	1	BABY CARE	0.0	0
2	2	1/1/13	1	BEAUTY	0.0	0
3	3	1/1/13	1	BEVERAGES	0.0	0
4	4	1/1/13	1	BOOKS	0.0	0

In [406…
```
df_train_2 = pd.read_csv("train2.csv")
df_train_2.head()
```

Out[406]:

	id	date	store_nbr	family	sales	onpromotion
0	262143	5/28/13	14	MEATS	310.173000	0
1	262144	5/28/13	14	PERSONAL CARE	227.000000	0
2	262145	5/28/13	14	PET SUPPLIES	0.000000	0
3	262146	5/28/13	14	PLAYERS AND ELECTRONICS	0.000000	0
4	262147	5/28/13	14	POULTRY	47.809002	0

In [407…
```
df_train_3 = pd.read_csv("train3.csv")
df_train_3.head()
```

Out[407]:

	id	date	store_nbr	family	sales	onpromotion
0	524286	10/22/13	2	HOME AND KITCHEN I	0.0	0
1	524287	10/22/13	2	HOME AND KITCHEN II	0.0	0
2	524288	10/22/13	2	HOME APPLIANCES	0.0	0
3	524289	10/22/13	2	HOME CARE	0.0	0
4	524290	10/22/13	2	LADIESWEAR	0.0	0

In [408…
```
df_train_4 = pd.read_csv("train4.csv")
df_train_4.head()
```

Out[408]:

	id	date	store_nbr	family	sales	onpromotion
0	786429	3/19/14	25	CELEBRATION	8.000	0
1	786430	3/19/14	25	CLEANING	485.000	0
2	786431	3/19/14	25	DAIRY	588.000	0
3	786432	3/19/14	25	DELI	116.357	0
4	786433	3/19/14	25	EGGS	119.000	0

In [409…
```
print("Total length of training dataframes : ",(len(df_train_1)+len(df_train
```
```
Total length of training dataframes :  1048575
```

Merging all training dataframes into one single dataframe

In [410…
```
df_train = pd.concat([df_train_1,df_train_2,df_train_3, df_train_4])
df_train.head()
```

Out[410]:

	id	date	store_nbr	family	sales	onpromotion
0	0	1/1/13	1	AUTOMOTIVE	0.0	0
1	1	1/1/13	1	BABY CARE	0.0	0
2	2	1/1/13	1	BEAUTY	0.0	0
3	3	1/1/13	1	BEVERAGES	0.0	0
4	4	1/1/13	1	BOOKS	0.0	0

In [411…
```
print("Total length of merged dataframe : ",(len(df_train)))
```
```
Total length of merged dataframe :  1048575
```

Importing testing dataset

In [412…
```
df_test = pd.read_csv("test.csv")
df_test.head()
```

Out[412]:

	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

In [413…
```
print("Total length of testing dataframe : ",(len(df_test)))
```
```
Total length of testing dataframe :  28512
```

Checking for null values in both training and testing datasets

In [414…
```
df_train.isna().sum()
```

Out[414]:
```
id             0
date           0
store_nbr      0
family         0
sales          0
onpromotion    0
dtype: int64
```

In [415…
```
df_test.isna().sum()
```

Out[415]:
```
id             0
date           0
store_nbr      0
family         0
onpromotion    0
dtype: int64
```

Exploring other datasets

In [416…
```
df_stores = pd.read_csv("stores.csv")
df_stores.head()
```

Out[416]:

	store_nbr	city	state	type	cluster
0	1	Quito	Pichincha	D	13
1	2	Quito	Pichincha	D	13
2	3	Quito	Pichincha	D	8
3	4	Quito	Pichincha	D	9
4	5	Santo Domingo	Santo Domingo de los Tsachilas	D	4

In [417…
```
df_oil = pd.read_csv("oil.csv")
df_oil.head()
```

Out[417]:

	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

In [418…
```
df_holiday_events = pd.read_csv("holidays_events.csv")
df_holiday_events.head()
```

Out[418]:

	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

In [419…
```
df_transactions = pd.read_csv("transactions.csv")
df_transactions.head()
```

Out[419]:

	date	store_nbr	transactions
0	2013-01-01	25	770
1	2013-01-02	1	2111
2	2013-01-02	2	2358
3	2013-01-02	3	3487
4	2013-01-02	4	1922

In [420…
```
print("Total length of stores dataframe         :  ",len(df_stores))
print("Total length of oil dataframe            :  ",len(df_oil))
print("Total length of holiday events dataframe :  ",len(df_holiday_events))
print("Total length of transactions dataframe   :  ",len(df_transactions))
```

```
Total length of stores dataframe         :   54
Total length of oil dataframe            :   1218
Total length of holiday events dataframe :   350
Total length of transactions dataframe   :   83488
```

Checking for null values in stores, oil, holiday events, transactions dataframes

In [421…
```
df_stores.isna().sum()
```

Out[421]:
```
store_nbr    0
city         0
state        0
type         0
cluster      0
dtype: int64
```

In [422…
```
df_oil.isna().sum()
```

Out[422]:
```
date           0
dcoilwtico    43
dtype: int64
```

In [423…
```
df_holiday_events.isna().sum()
```

Out[423]:
```
date           0
type           0
locale         0
locale_name    0
description    0
transferred    0
dtype: int64
```

In [424…
```
df_transactions.isna().sum()
```

Out[424]:
```
date            0
store_nbr       0
transactions    0
dtype: int64
```

Filling in null values in oil dataset

In [425…
```
df_oil_nona = df_oil.copy()
modal = df_oil_nona["dcoilwtico"].mode()
na = df_oil_nona[df_oil_nona["dcoilwtico"].isna()==True].index.values
df_oil_nona.loc[0,"dcoilwtico"] = df_oil_nona.loc[0:10,"dcoilwtico"].mean()
for i in na[1:]:
    df_oil_nona.loc[i,"dcoilwtico"] = df_oil_nona.loc[(i-10):(i+10),"dcoilwt
df_oil_nona.head()
```

Out[425]:

	date	dcoilwtico
0	2013-01-01	93.366
1	2013-01-02	93.140
2	2013-01-03	92.970
3	2013-01-04	93.120
4	2013-01-07	93.200

In [426…
```
df_oil_nona.isna().sum()
```

Out[426]:
```
date          0
dcoilwtico    0
dtype: int64
```

In [427…
```
df_holiday_events.head()
```

Out[427]:

	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

Preprocessing the holiday events dataset

In [428…
```
before_transfer = df_holiday_events[df_holiday_events["transferred"]==True]
before_transfer.head()
```

Out[428]:

	date	type	locale	locale_name	description	transferred
19	2012-10-09	Holiday	National	Ecuador	Independencia de Guayaquil	True
72	2013-10-09	Holiday	National	Ecuador	Independencia de Guayaquil	True
135	2014-10-09	Holiday	National	Ecuador	Independencia de Guayaquil	True
255	2016-05-24	Holiday	National	Ecuador	Batalla de Pichincha	True
266	2016-07-25	Holiday	Local	Guayaquil	Fundacion de Guayaquil	True

In [429…
```
after_transfer = df_holiday_events[df_holiday_events["type"]=="Transfer"]
after_transfer.head()
```

Out[429]:

	date	type	locale	locale_name	description	transferred
20	2012-10-12	Transfer	National	Ecuador	Traslado Independencia de Guayaquil	False
73	2013-10-11	Transfer	National	Ecuador	Traslado Independencia de Guayaquil	False
136	2014-10-10	Transfer	National	Ecuador	Traslado Independencia de Guayaquil	False
256	2016-05-27	Transfer	National	Ecuador	Traslado Batalla de Pichincha	False
265	2016-07-24	Transfer	Local	Guayaquil	Traslado Fundacion de Guayaquil	False

In [430…
```
df_holiday_trans = df_holiday_events.drop(before_transfer.index.values,axis=
ts = after_transfer.index.values
for i in ts:
    st = df_holiday_trans.loc[i,"description"]
    ls = st.split(" ")
    if "Traslado" in ls:
        ls.remove("Traslado")
    df_holiday_trans.loc[i,"description"] = " ".join(ls)
```

In [431…
```
df_holiday_trans[df_holiday_trans["type"]=="Bridge"]
```

Out[431]:

	date	type	locale	locale_name	description	transferred
35	2012-12-24	Bridge	National	Ecuador	Puente Navidad	False
39	2012-12-31	Bridge	National	Ecuador	Puente Primer dia del ano	False
156	2014-12-26	Bridge	National	Ecuador	Puente Navidad	False
160	2015-01-02	Bridge	National	Ecuador	Puente Primer dia del ano	False
277	2016-11-04	Bridge	National	Ecuador	Puente Dia de Difuntos	False

In [432…
```
df_holiday_trans["type"].unique()
```

Out[432]:
```
array(['Holiday', 'Transfer', 'Additional', 'Bridge', 'Work Day', 'Event'],
      dtype=object)
```

In [433…
```
df_holiday_trans[df_holiday_trans["type"]=="Additional"].head()
```

Out[433]:

	date	type	locale	locale_name	description	transferred
28	2012-12-05	Additional	Local	Quito	Fundacion de Quito-1	False
31	2012-12-21	Additional	National	Ecuador	Navidad-4	False
33	2012-12-22	Additional	National	Ecuador	Navidad-3	False
34	2012-12-23	Additional	National	Ecuador	Navidad-2	False
36	2012-12-24	Additional	National	Ecuador	Navidad-1	False

In [434…
```
work = df_holiday_trans[df_holiday_trans["type"]=="Work Day"].index.values
work
```

Out[434]:
```
array([ 42,  43, 149, 161, 283])
```

In [435…
```
df_holiday = df_holiday_trans.drop(work,axis=0,inplace=False)
df_holiday.head()
```

Out[435]:

	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

Displaying stores, oil, holiday events, transactions dataframes once again

In [436…
```
df_stores.head()
```

Out[436]:

	store_nbr	city	state	type	cluster
0	1	Quito	Pichincha	D	13
1	2	Quito	Pichincha	D	13
2	3	Quito	Pichincha	D	8
3	4	Quito	Pichincha	D	9
4	5	Santo Domingo	Santo Domingo de los Tsachilas	D	4

In [437…
```
df_oil_nona.head()
```

Out[437]:

	date	dcoilwtico
0	2013-01-01	93.366
1	2013-01-02	93.140
2	2013-01-03	92.970
3	2013-01-04	93.120
4	2013-01-07	93.200

In [438…
```
df_holiday.head()
```

Out[438]:

	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

In [439…
```
df_transactions.head()
```

Out[439]:

	date	store_nbr	transactions
0	2013-01-01	25	770
1	2013-01-02	1	2111
2	2013-01-02	2	2358
3	2013-01-02	3	3487
4	2013-01-02	4	1922

Displaying training and testing datasets once again

In [440…
```
df_train.head()
```

Out[440]:

	id	date	store_nbr	family	sales	onpromotion
0	0	1/1/13	1	AUTOMOTIVE	0.0	0
1	1	1/1/13	1	BABY CARE	0.0	0
2	2	1/1/13	1	BEAUTY	0.0	0
3	3	1/1/13	1	BEVERAGES	0.0	0
4	4	1/1/13	1	BOOKS	0.0	0

In [441…
```
df_test.head()
```

Out[441]:

	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

Modifying "date" column in training dataset

In [442…
```
df_train.head()
```

Out[442]:

	id	date	store_nbr	family	sales	onpromotion
0	0	1/1/13	1	AUTOMOTIVE	0.0	0
1	1	1/1/13	1	BABY CARE	0.0	0
2	2	1/1/13	1	BEAUTY	0.0	0
3	3	1/1/13	1	BEVERAGES	0.0	0
4	4	1/1/13	1	BOOKS	0.0	0

In [443…
```
df_train_mod = df_train.copy()
df_train_mod["date"] = pd.to_datetime(df_train_mod["date"])
df_train_mod.head()
```

Out[443]:

	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

Combining training and testing datasets with stores, oil, holiday events, transactions

Combining with stores dataset

In [444…
```
print("Length of training dataframe : ",(len(df_train_mod)))
print("Length of testing dataframe  : ",(len(df_test)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

In [445…
```
df_stores.head()
```

Out[445]:

	store_nbr	city	state	type	cluster
0	1	Quito	Pichincha	D	13
1	2	Quito	Pichincha	D	13
2	3	Quito	Pichincha	D	8
3	4	Quito	Pichincha	D	9
4	5	Santo Domingo	Santo Domingo de los Tsachilas	D	4

In [446…
```
df_train_mod["date"] = pd.to_datetime(df_train_mod["date"])

df_train_mod_1 = pd.merge(left=df_train_mod,right=df_stores,how="left",on="s
df_test_mod_1 = pd.merge(left=df_test,right=df_stores,how="left",on="store_n
```

In [447…
```
df_train_mod_1.head()
```

Out[447]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [448…
```
df_test_mod_1.head()
```

Out[448]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [449…

```
print("Length of training dataframe : ",(len(df_train_mod_1)))
print("Length of testing dataframe  : ",(len(df_test_mod_1)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

In [450…

```
df_train_mod_1.isna().sum()
```

Out[450]:

```
id             0
date           0
store_nbr      0
family         0
sales          0
onpromotion    0
city           0
state          0
type           0
cluster        0
dtype: int64
```

In [451…

```
df_test_mod_1.isna().sum()
```

Out[451]:

```
id             0
date           0
store_nbr      0
family         0
onpromotion    0
city           0
state          0
type           0
cluster        0
dtype: int64
```

Combining with oil dataset

In [452…

```
df_oil_nona.head()
```

Out[452]:

	date	dcoilwtico
0	2013-01-01	93.366
1	2013-01-02	93.140
2	2013-01-03	92.970
3	2013-01-04	93.120
4	2013-01-07	93.200

In [453…
```
df_train_mod_1["date"] = pd.to_datetime(df_train_mod_1["date"])
df_test_mod_1["date"] = pd.to_datetime(df_test_mod_1["date"])

df_oil_nona["date"] = pd.to_datetime(df_oil_nona["date"])

df_train_mod_2 = pd.merge(left=df_train_mod_1,right=df_oil_nona,how="left",c
df_test_mod_2 = pd.merge(left=df_test_mod_1,right=df_oil_nona,how="left",on=
```

In [454…
```
df_train_mod_2.head()
```

Out[454]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [455…
```
df_test_mod_2.head()
```

Out[455]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [456…
```
print("Length of training dataframe : ",(len(df_train_mod_2)))
print("Length of testing dataframe  : ",(len(df_test_mod_2)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

In [457…
```
df_train_mod_2.isna().sum()
```

Out[457]:
```
id                  0
date                0
store_nbr           0
family              0
sales               0
onpromotion         0
city                0
state               0
type                0
cluster             0
dcoilwtico     299376
dtype: int64
```

In [458…
```
df_test_mod_2.isna().sum()
```

Out[458]:
```
id                 0
date               0
store_nbr          0
family             0
onpromotion        0
city               0
state              0
type               0
cluster            0
dcoilwtico      7128
dtype: int64
```

In [459…
```
hh = df_train_mod_2[~df_train_mod_2["dcoilwtico"].isna()]
df_train_mod_3 = df_train_mod_2.fillna(value=hh["dcoilwtico"].mean(),inplace
df_train_mod_3.head()
```

Out[459]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [460…
```
hh2 = df_test_mod_2[~df_test_mod_2["dcoilwtico"].isna()]
df_test_mod_3 = df_test_mod_2.fillna(value=hh2["dcoilwtico"].mean(),inplace=
df_test_mod_3.head()
```

Out[460]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [461…

```
print("Length of training dataframe : ",(len(df_train_mod_3)))
print("Length of testing dataframe  : ",(len(df_test_mod_3)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

Combining with holiday events dataframe

In [462…

```
df_holiday_mod = df_holiday.drop(["type","transferred"],axis=1,inplace=False
df_holiday_mod.head()
```

Out[462]:

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

In [463…

```
df_train_mod_3["date"] = pd.to_datetime(df_train_mod_3["date"])
df_test_mod_3["date"] = pd.to_datetime(df_test_mod_3["date"])

df_holiday_mod["date"] = pd.to_datetime(df_holiday_mod["date"])

df_train_mod_4 = pd.merge(left=df_train_mod_3,right=df_holiday_mod,how="left
df_test_mod_4 = pd.merge(left=df_test_mod_3,right=df_holiday_mod,how="left",
```

In [464…

```
df_train_mod_4.head()
```

Out[464]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [465…

```
df_test_mod_4.head()
```

Out[465]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [466…

```
print("Length of training dataframe : ",(len(df_train_mod_4)))
print("Length of testing dataframe  : ",(len(df_test_mod_4)))
```

```
Length of training dataframe :  1064613
Length of testing dataframe  :  28512
```

In [467…

```
train_na_values = df_train_mod_4.isna().sum()
train_na_values
```

Out[467]:

```
id                 0
date               0
store_nbr          0
family             0
sales              0
onpromotion        0
city               0
state              0
type               0
cluster            0
dcoilwtico         0
locale        911361
locale_name   911361
description   911361
dtype: int64
```

In [468…
```
test_na_values = df_test_mod_4.isna().sum()
test_na_values
```

Out[468]:
```
id                0
date              0
store_nbr         0
family            0
onpromotion       0
city              0
state             0
type              0
cluster           0
dcoilwtico        0
locale        26730
locale_name   26730
description   26730
dtype: int64
```

In [469…
```
print("Number of null values in training dataset : ",(train_na_values["local
print("Number of null values in testing dataset  : ",(test_na_values["locale
```

```
Number of null values in training dataset :  85.6
Number of null values in testing dataset  :  93.75
```

Since the resultant training and testing datasets have a very high percentage of null values, this cannot be accepted. This means that the parameter needed to be considered can be only if a day is a holiday or not.

In [470…
```
df_holiday.head()
```

Out[470]:

	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

In [471…
```
holiday_date1 = df_holiday["date"].drop_duplicates(ignore_index=True,inplace
holiday_date = pd.DataFrame(columns=["date","holiday?"])
holiday_date["date"] = holiday_date1
holiday_date["holiday?"] = 1
```

In [472…
```
holiday_date.head()
```

Out[472]:

	date	holiday?
0	2012-03-02	1
1	2012-04-01	1
2	2012-04-12	1
3	2012-04-14	1
4	2012-04-21	1

In [473…
```
print("Number of holidays : ",len(holiday_date))
```

```
Number of holidays :  296
```

In [474…
```
holiday_date["date"] = pd.to_datetime(holiday_date["date"])

df_train_mod_5 = pd.merge(left=df_train_mod_3,right=holiday_date,how="left",
df_test_mod_5 = pd.merge(left=df_test_mod_3,right=holiday_date,how="left",on
```

In [475…
```
df_train_mod_5.head()
```

Out[475]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [476…
```
df_test_mod_5.head()
```

Out[476]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [477…
```
print("Length of training dataframe : ",(len(df_train_mod_5)))
print("Length of testing dataframe  : ",(len(df_test_mod_5)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

In [478…
```
print("Number of null values in 'holiday' column in training dataset : ",df_
print("Number of null values in 'holiday' column in testing dataset  : ",df_
```

```
Number of null values in 'holiday' column in training dataset :  911361
Number of null values in 'holiday' column in testing dataset  :  26730
```

In [479…
```
df_train_mod_6 = df_train_mod_5.fillna(value=0,inplace=False)
df_test_mod_6 = df_test_mod_5.fillna(value=0,inplace=False)
```

In [480…
```
df_train_mod_6.head()
```

Out[480]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [481…
```
df_test_mod_6.head()
```

Out[481]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [482…
```
print("Length of training dataframe : ",(len(df_train_mod_6)))
print("Length of testing dataframe  : ",(len(df_test_mod_6)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

In [483…
```
print("Number of null values in 'holiday' column in training dataset : ",df_
print("Number of null values in 'holiday' column in testing dataset  : ",df_
```

```
Number of null values in 'holiday' column in training dataset :  0
Number of null values in 'holiday' column in testing dataset  :  0
```

Combining with transactions dataset

In [484…
```
df_transactions.head()
```

Out[484]:

	date	store_nbr	transactions
0	2013-01-01	25	770
1	2013-01-02	1	2111
2	2013-01-02	2	2358
3	2013-01-02	3	3487
4	2013-01-02	4	1922

In [485…
```
print("Length of transactions dataset : ",len(df_transactions))
```

```
Length of transactions dataset :  83488
```

In [486…
```
df_transactions["date"] = pd.to_datetime(df_transactions["date"])

df_train_mod_7 = pd.merge(left=df_train_mod_6,right=df_transactions,how="lef
df_test_mod_7 = pd.merge(left=df_test_mod_6,right=df_transactions,how="left"
```

In [487…
```
df_train_mod_7.head()
```

Out[487]:

	id	date	store_nbr	family	sales	onpromotion	city	state	type	cluster
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13
1	1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13
2	2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13
3	3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13
4	4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13

In [488…
```
df_test_mod_7.head()
```

Out[488]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [489…
```
print("Length of training dataframe : ",(len(df_train_mod_7)))
print("Length of testing dataframe  : ",(len(df_test_mod_7)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

In [490…
```
df_train_mod_7.isna().sum()
```

Out[490]:
```
id                   0
date                 0
store_nbr            0
family               0
sales                0
onpromotion          0
city                 0
state                0
type                 0
cluster              0
dcoilwtico           0
holiday?             0
transactions    145266
dtype: int64
```

In [491…
```
df_test_mod_7.isna().sum()
```

Out[491]:
```
id                   0
date                 0
store_nbr            0
family               0
onpromotion          0
city                 0
state                0
type                 0
cluster              0
dcoilwtico           0
holiday?             0
transactions     28512
dtype: int64
```

In [492…
```
train_null = df_train_mod_7["transactions"].isna().sum()
test_null = df_test_mod_7["transactions"].isna().sum()
```

In [493…
```
print("Number of null values in 'transactions' column in training dataset :
print("Number of null values in 'transactions' column in testing dataset  :
```

```
Number of null values in 'transactions' column in training dataset :  145266
Number of null values in 'transactions' column in testing dataset  :  28512
```

In [494…
```
print("Percentage of null values in 'transactions' column in training datase
print("Percentage of null values in 'transactions' column in testing dataset
```

```
Percentage of null values in 'transactions' column in training dataset :  1
3.85
Percentage of null values in 'transactions' column in testing dataset  :  10
0.0
```

In the testing dataset, there are no transactional values at all. Hence, let us calculate the perentage of all null values in both the datasets.

In [495…
```
total_per = (train_null+test_null)*100/(len(df_train_mod_7)+len(df_test_mod_
print("Total percentage : ",total_per.round(2))
```

```
Total percentage :  16.13
```

Although the null values for transactional data account for 16.13%, the testing dataset do not have any transactional values, hence we shall ignore the column.

Final datasets

In [496…
```
train_col = df_train_mod_6.columns.values
loc = np.where(train_col=="sales")[0][0]
train_col = np.delete(train_col,loc)
train_col = np.append(train_col,"sales")
train_col
```

Out[496]:
```
array(['id', 'date', 'store_nbr', 'family', 'onpromotion', 'city',
       'state', 'type', 'cluster', 'dcoilwtico', 'holiday?', 'sales'],
      dtype=object)
```

In [497…
```
final_train = df_train_mod_6[train_col]
final_test = df_test_mod_6.copy()
```

In [498…
```
final_train.head()
```

Out[498]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster	dcoilwt
0	0	2013-01-01	1	AUTOMOTIVE	0	Quito	Pichincha	D	13	93.3
1	1	2013-01-01	1	BABY CARE	0	Quito	Pichincha	D	13	93.3
2	2	2013-01-01	1	BEAUTY	0	Quito	Pichincha	D	13	93.3
3	3	2013-01-01	1	BEVERAGES	0	Quito	Pichincha	D	13	93.3
4	4	2013-01-01	1	BOOKS	0	Quito	Pichincha	D	13	93.3

In [499…
```
final_test.head()
```

Out[499]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13

In [500…
```
print("Length of training dataframe : ",(len(final_train)))
print("Length of testing dataframe  : ",(len(final_test)))
```

```
Length of training dataframe :  1048575
Length of testing dataframe  :  28512
```

Checking null values in final datasets

In [501…
```
final_train.isna().sum()
```

Out[501]:
```
id             0
date           0
store_nbr      0
family         0
onpromotion    0
city           0
state          0
type           0
cluster        0
dcoilwtico     0
holiday?       0
sales          0
dtype: int64
```

In [502…
```
final_test.isna().sum()
```

Out[502]:
```
id             0
date           0
store_nbr      0
family         0
onpromotion    0
city           0
state          0
type           0
cluster        0
dcoilwtico     0
holiday?       0
dtype: int64
```

Exporting the datasets to respective CSV files

In [503…
```
final_train.to_csv("train_eda.csv",index=False)
final_test.to_csv("test_eda.csv",index=False)
```

In [503…

In [ ]:

Data Preprocessing

Now, we are done with exploratory data analysis of both training and testing datasets. Now, we should get into preprocessing for both the datasets as some of the features are not numerical.

Importing all packages

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import *
from sklearn.linear_model import *
from math import *
from sklearn.ensemble import *
from sklearn.feature_selection import *
from sklearn.feature_extraction import *
from sklearn.naive_bayes import *
from sklearn.discriminant_analysis import *
from sklearn.preprocessing import *
from sklearn.metrics import *
from sklearn.neighbors import *
from sklearn.cluster import *
```

In [2]:
```
df_train = pd.read_csv("train_eda.csv")
df_test = pd.read_csv("test_eda.csv")
```

In [3]:
```
df_train.head()
```

Out[3]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster	dcoilwti
0	0	2013-01-01	1	AUTOMOTIVE	0	Quito	Pichincha	D	13	93.3(
1	1	2013-01-01	1	BABY CARE	0	Quito	Pichincha	D	13	93.3(
2	2	2013-01-01	1	BEAUTY	0	Quito	Pichincha	D	13	93.3(
3	3	2013-01-01	1	BEVERAGES	0	Quito	Pichincha	D	13	93.3(
4	4	2013-01-01	1	BOOKS	0	Quito	Pichincha	D	13	93.3(

In [4]:
```
df_test.head()
```

Out[4]:

	id	date	store_nbr	family	onpromotion	city	state	type	cluster	d
0	3000888	2017-08-16	1	AUTOMOTIVE	0	Quito	Pichincha	D	13	
1	3000889	2017-08-16	1	BABY CARE	0	Quito	Pichincha	D	13	
2	3000890	2017-08-16	1	BEAUTY	2	Quito	Pichincha	D	13	
3	3000891	2017-08-16	1	BEVERAGES	20	Quito	Pichincha	D	13	
4	3000892	2017-08-16	1	BOOKS	0	Quito	Pichincha	D	13	

In [5]:
```
print("Length of training dataset : ",len(df_train))
print("Length of testing dataset  : ",len(df_test))
```

```
Length of training dataset :  1048575
Length of testing dataset  :  28512
```

Description on both training and testing datasets

In [6]:
```
df_train.describe().round(2)
```

Out[6]:

	id	store_nbr	onpromotion	cluster	dcoilwtico	holiday?	sale
count	1048575.00	1048575.00	1048575.00	1048575.00	1048575.00	1048575.00	1048575.0
mean	524287.00	27.49	0.11	8.48	99.19	0.13	244.5
std	302697.67	15.58	2.38	4.65	4.31	0.34	806.5
min	0.00	1.00	0.00	1.00	86.65	0.00	0.0
25%	262143.50	14.00	0.00	4.00	96.29	0.00	0.0
50%	524287.00	27.00	0.00	9.00	99.19	0.00	1.0
75%	786430.50	41.00	0.00	13.00	101.92	0.00	120.0
max	1048574.00	54.00	196.00	17.00	110.62	1.00	46271.0

In [7]:
```
df_test.describe().round(2)
```

Out[7]:

	id	store_nbr	onpromotion	cluster	dcoilwtico	holiday?
count	28512.00	28512.00	28512.00	28512.00	28512.00	28512.00
mean	3015143.50	27.50	6.97	8.48	47.24	0.06
std	8230.85	15.59	20.68	4.65	0.65	0.24
min	3000888.00	1.00	0.00	1.00	45.96	0.00
25%	3008015.75	14.00	0.00	4.00	47.00	0.00
50%	3015143.50	27.50	0.00	8.50	47.24	0.00
75%	3022271.25	41.00	6.00	13.00	47.46	0.00
max	3029399.00	54.00	646.00	17.00	48.59	1.00

Information about both training and testing datasets

In [8]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 12 columns):
 #   Column       Non-Null Count    Dtype  
---  ------       --------------    -----  
 0   id           1048575 non-null  int64  
 1   date         1048575 non-null  object 
 2   store_nbr    1048575 non-null  int64  
 3   family       1048575 non-null  object 
 4   onpromotion  1048575 non-null  int64  
 5   city         1048575 non-null  object 
 6   state        1048575 non-null  object 
 7   type         1048575 non-null  object 
 8   cluster      1048575 non-null  int64  
 9   dcoilwtico   1048575 non-null  float64
 10  holiday?     1048575 non-null  float64
 11  sales        1048575 non-null  float64
dtypes: float64(3), int64(4), object(5)
memory usage: 96.0+ MB
```

In [9]:
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28512 entries, 0 to 28511
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   id           28512 non-null  int64  
 1   date         28512 non-null  object 
 2   store_nbr    28512 non-null  int64  
 3   family       28512 non-null  object 
 4   onpromotion  28512 non-null  int64  
 5   city         28512 non-null  object 
 6   state        28512 non-null  object 
 7   type         28512 non-null  object 
 8   cluster      28512 non-null  int64  
 9   dcoilwtico   28512 non-null  float64
 10  holiday?     28512 non-null  float64
dtypes: float64(2), int64(4), object(5)
memory usage: 2.4+ MB
```

Conversion of date from object type to datetime[64ns] type

In [10]:
```
df_train["date"] = pd.to_datetime(df_train["date"])
df_test["date"] = pd.to_datetime(df_test["date"])
```

Checking for "store" column

In [11]:
```
store_num_train = df_train["store_nbr"].unique()
store_num_train.sort()
store_num_train
```

Out[11]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
       52, 53, 54])
```

In [12]:
```
store_num_test = df_test["store_nbr"].unique()
store_num_test.sort()
store_num_test
```

Out[12]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
       52, 53, 54])
```

Subtracting each element of store numbers for convenience

In [13]:
```
df_train_1 = df_train.copy()
df_test_1 = df_test.copy()
df_train_1["store_nbr"] = df_train_1["store_nbr"] - 1
df_test_1["store_nbr"] = df_test_1["store_nbr"] - 1
```

Binary encoding function

In [14]:
```
def dec_to_bin(num):
    n = num
    st = []
    while n > 0:
        r = int(n % 2)
        n = int(n / 2)
        st.append(r)
    fin = st[::-1]
    return fin
```

In [15]:
```
def bin_to_dec(num):
    n = num[::-1]
    st = 0
    ctr = 0
    for i in n:
        st += i * pow(2,ctr)
        ctr += 1
    return int(st)
```

In [16]:
```
def max_len_bin(x, lt):
    ctr = len(x)
    if ctr < lt:
        u = np.zeros(lt - ctr)
        v = u.astype("int")
```

```
        t = list(v)
        f = t + x
    else:
        f = x
    return f
```

In [17]:
```
def column_names(nam, col_num):
    lis = []
    for i in range(0, col_num):
        t = nam + "_" + str(i)
        lis.append(t)
    return lis
```

In [18]:
```
def binary_encoder(df,col_name):
    bin_convert = lambda x: dec_to_bin(x)
    bin_num = list(map(bin_convert,df[col_name]))
    bin_len = lambda x: len(x)
    max_bin = max(list(map(bin_len,bin_num)))
    equal_len = lambda x: max_len_bin(x,max_bin)
    equal_ele = list(map(equal_len,bin_num))
    cols = column_names(col_name,max_bin)
    df_bin = pd.DataFrame(equal_ele,columns=cols)
    return df_bin
```

In [19]:
```
binary_encoder(df_train_1,"store_nbr").head()
```

Out[19]:

	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
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

Performing binary encoding for "store_nbr" column

In [20]:
```
store_num_train = binary_encoder(df_train_1,"store_nbr")
store_num_test = binary_encoder(df_test_1,"store_nbr")
```

In [21]:
```
store_num_train.head()
```

Out[21]:

	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
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

In [22]:
```
store_num_test.head()
```

Out[22]:

	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
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

In [23]:
```
print("Number of rows in training set : ",len(store_num_train))
print("Number of rows in testing set  : ",len(store_num_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

In [24]:
```
df_train_2 = df_train_1.copy()
df_test_2 = df_test_1.copy()
df_train_2[store_num_train.columns.values] = store_num_train
df_test_2[store_num_test.columns.values] = store_num_test
df_train_2.drop("store_nbr",axis=1,inplace=True)
df_test_2.drop("store_nbr",axis=1,inplace=True)
df_train_2 = df_train_2[["id","date","store_nbr_0","store_nbr_1","store_nbr_
df_test_2 = df_test_2[["id","date","store_nbr_0","store_nbr_1","store_nbr_2"
```

In [25]:
```
df_train_2.head()
```

Out[25]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
0	0	2013-01-01	0	0	0	0	0	0
1	1	2013-01-01	0	0	0	0	0	0
2	2	2013-01-01	0	0	0	0	0	0
3	3	2013-01-01	0	0	0	0	0	0
4	4	2013-01-01	0	0	0	0	0	0

In [26]:
```
df_test_2.head()
```

Out[26]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_
0	3000888	2017-08-16	0	0	0	0	0	
1	3000889	2017-08-16	0	0	0	0	0	
2	3000890	2017-08-16	0	0	0	0	0	
3	3000891	2017-08-16	0	0	0	0	0	
4	3000892	2017-08-16	0	0	0	0	0	

In [27]:
```
print("Number of rows in training set : ",len(df_train_2))
print("Number of rows in testing set  : ",len(df_test_2))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Checking for "family" column

In [28]:
```
df_train_2["family"].unique()
```

Out[28]:
```
array(['AUTOMOTIVE', 'BABY CARE', 'BEAUTY', 'BEVERAGES', 'BOOKS',
       'BREAD/BAKERY', 'CELEBRATION', 'CLEANING', 'DAIRY', 'DELI', 'EGGS',
       'FROZEN FOODS', 'GROCERY I', 'GROCERY II', 'HARDWARE',
       'HOME AND KITCHEN I', 'HOME AND KITCHEN II', 'HOME APPLIANCES',
       'HOME CARE', 'LADIESWEAR', 'LAWN AND GARDEN', 'LINGERIE',
       'LIQUOR,WINE,BEER', 'MAGAZINES', 'MEATS', 'PERSONAL CARE',
       'PET SUPPLIES', 'PLAYERS AND ELECTRONICS', 'POULTRY',
       'PREPARED FOODS', 'PRODUCE', 'SCHOOL AND OFFICE SUPPLIES',
       'SEAFOOD'], dtype=object)
```

In [29]:
```
df_test_2["family"].unique()
```

Out[29]:
```
array(['AUTOMOTIVE', 'BABY CARE', 'BEAUTY', 'BEVERAGES', 'BOOKS',
       'BREAD/BAKERY', 'CELEBRATION', 'CLEANING', 'DAIRY', 'DELI', 'EGGS',
       'FROZEN FOODS', 'GROCERY I', 'GROCERY II', 'HARDWARE',
       'HOME AND KITCHEN I', 'HOME AND KITCHEN II', 'HOME APPLIANCES',
       'HOME CARE', 'LADIESWEAR', 'LAWN AND GARDEN', 'LINGERIE',
       'LIQUOR,WINE,BEER', 'MAGAZINES', 'MEATS', 'PERSONAL CARE',
       'PET SUPPLIES', 'PLAYERS AND ELECTRONICS', 'POULTRY',
       'PREPARED FOODS', 'PRODUCE', 'SCHOOL AND OFFICE SUPPLIES',
       'SEAFOOD'], dtype=object)
```

Mapping family elements to numbers

In [30]:
```
ctr = 0
diction = dict()
for i in df_train_2["family"].unique():
    diction[i] = ctr
    ctr += 1
diction
```

Out[30]:
```
{'AUTOMOTIVE': 0,
 'BABY CARE': 1,
 'BEAUTY': 2,
 'BEVERAGES': 3,
 'BOOKS': 4,
 'BREAD/BAKERY': 5,
 'CELEBRATION': 6,
 'CLEANING': 7,
 'DAIRY': 8,
 'DELI': 9,
 'EGGS': 10,
 'FROZEN FOODS': 11,
 'GROCERY I': 12,
 'GROCERY II': 13,
 'HARDWARE': 14,
 'HOME AND KITCHEN I': 15,
 'HOME AND KITCHEN II': 16,
 'HOME APPLIANCES': 17,
 'HOME CARE': 18,
 'LADIESWEAR': 19,
 'LAWN AND GARDEN': 20,
 'LINGERIE': 21,
 'LIQUOR,WINE,BEER': 22,
 'MAGAZINES': 23,
 'MEATS': 24,
 'PERSONAL CARE': 25,
 'PET SUPPLIES': 26,
 'PLAYERS AND ELECTRONICS': 27,
 'POULTRY': 28,
 'PREPARED FOODS': 29,
 'PRODUCE': 30,
 'SCHOOL AND OFFICE SUPPLIES': 31,
 'SEAFOOD': 32}
```

In [31]:
```
fam_train = pd.DataFrame(df_train_2["family"].map(diction))
fam_test = pd.DataFrame(df_test_2["family"].map(diction))
```

In [32]:
```
fam_train.head()
```

Out[32]:

	family
0	0
1	1
2	2
3	3
4	4

In [33]:
```
fam_test.head()
```

Out[33]:

	family
0	0
1	1
2	2
3	3
4	4

In [34]:
```
print("Number of rows in training set : ",len(fam_train))
print("Number of rows in testing set  : ",len(fam_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Performing binary encoding on "family" column

In [35]:
```
family_train = binary_encoder(fam_train,"family")
family_test = binary_encoder(fam_test,"family")
```

In [36]:
```
family_train.head()
```

Out[36]:

	family_0	family_1	family_2	family_3	family_4	family_5
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0

In [37]:
```
family_test.head()
```

Out[37]:

	family_0	family_1	family_2	family_3	family_4	family_5
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0

In [38]:
```
df_train_3 = df_train_2.copy()
df_test_3 = df_test_2.copy()
df_train_3[family_train.columns.values] = family_train
df_test_3[family_test.columns.values] = family_test
df_train_3.drop("family",axis=1,inplace=True)
df_test_3.drop("family",axis=1,inplace=True)
df_train_3 = df_train_3[["id","date","store_nbr_0","store_nbr_1","store_nbr_
df_test_3 = df_test_3[["id","date","store_nbr_0","store_nbr_1","store_nbr_2"
```

In [39]:
```
df_train_3.head()
```

Out[39]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
0	0	2013-01-01	0	0	0	0	0	0
1	1	2013-01-01	0	0	0	0	0	0
2	2	2013-01-01	0	0	0	0	0	0
3	3	2013-01-01	0	0	0	0	0	0
4	4	2013-01-01	0	0	0	0	0	0

5 rows × 22 columns

In [40]:
```
df_test_3.head()
```

Out[40]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_
0	3000888	2017-08-16	0	0	0	0	0	
1	3000889	2017-08-16	0	0	0	0	0	
2	3000890	2017-08-16	0	0	0	0	0	
3	3000891	2017-08-16	0	0	0	0	0	
4	3000892	2017-08-16	0	0	0	0	0	

5 rows × 21 columns

In [41]:
```
print("Number of rows in training set : ",len(df_train_3))
print("Number of rows in testing set  : ",len(df_test_3))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Checking for "city" column

In [42]:
```
df_train_3["city"].unique()
```

Out[42]:
```
array(['Quito', 'Cayambe', 'Latacunga', 'Riobamba', 'Ibarra',
       'Santo Domingo', 'Guaranda', 'Puyo', 'Ambato', 'Guayaquil',
       'Salinas', 'Daule', 'Babahoyo', 'Quevedo', 'Playas', 'Libertad',
       'Cuenca', 'Loja', 'Machala', 'Esmeraldas', 'Manta', 'El Carmen'],
      dtype=object)
```

In [43]:
```
df_test_3["city"].unique()
```

Out[43]:
```
array(['Quito', 'Cayambe', 'Latacunga', 'Riobamba', 'Ibarra',
       'Santo Domingo', 'Guaranda', 'Puyo', 'Ambato', 'Guayaquil',
       'Salinas', 'Daule', 'Babahoyo', 'Quevedo', 'Playas', 'Libertad',
       'Cuenca', 'Loja', 'Machala', 'Esmeraldas', 'Manta', 'El Carmen'],
      dtype=object)
```

In [44]:
```
print("Number of rows in training set : ",len(df_train_3["city"]))
print("Number of rows in testing set  : ",len(df_test_3["city"]))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Mapping city elements to numbers

In [45]:
```
city = dict()
ctr = 0
for i in df_train_3["city"].unique():
    city[i] = ctr
    ctr +=1
city
```

Out[45]:
```
{'Quito': 0,
 'Cayambe': 1,
 'Latacunga': 2,
 'Riobamba': 3,
 'Ibarra': 4,
 'Santo Domingo': 5,
 'Guaranda': 6,
 'Puyo': 7,
 'Ambato': 8,
 'Guayaquil': 9,
 'Salinas': 10,
 'Daule': 11,
 'Babahoyo': 12,
 'Quevedo': 13,
 'Playas': 14,
 'Libertad': 15,
 'Cuenca': 16,
 'Loja': 17,
 'Machala': 18,
 'Esmeraldas': 19,
 'Manta': 20,
 'El Carmen': 21}
```

In [46]:
```
df_train_4 = df_train_3.copy()
df_test_4 = df_test_3.copy()
city_train = pd.DataFrame(df_train_4["city"].map(city))
city_test = pd.DataFrame(df_test_4["city"].map(city))
```

In [47]:
```
print("Number of rows in training set : ",len(city_train))
print("Number of rows in testing set  : ",len(city_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Performing binary encoding on "city" column

In [48]:
```
city_bin_train = binary_encoder(city_train,"city")
city_bin_test = binary_encoder(city_test,"city")
```

In [49]:
```
city_bin_train.head()
```

Out[49]:

	city_0	city_1	city_2	city_3	city_4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

In [50]:
```
city_bin_test.head()
```

Out[50]:

	city_0	city_1	city_2	city_3	city_4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

In [51]:
```
print("Number of rows in training set : ",len(city_bin_train))
print("Number of rows in testing set  : ",len(city_bin_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

In [52]:
```
df_train_5 = df_train_4.copy()
df_test_5 = df_test_4.copy()
df_train_5[city_bin_train.columns.values] = city_bin_train
df_test_5[city_bin_test.columns.values] = city_bin_test
df_train_5.drop("city",axis=1,inplace=True)
df_test_5.drop("city",axis=1,inplace=True)
df_train_5 = df_train_5[["id","date","store_nbr_0","store_nbr_1","store_nbr_
df_test_5 = df_test_5[["id","date","store_nbr_0","store_nbr_1","store_nbr_2"
```

In [53]:
```
df_train_5.head()
```

Out[53]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
0	0	2013-01-01	0	0	0	0	0	0
1	1	2013-01-01	0	0	0	0	0	0
2	2	2013-01-01	0	0	0	0	0	0
3	3	2013-01-01	0	0	0	0	0	0
4	4	2013-01-01	0	0	0	0	0	0

5 rows × 26 columns

In [54]:
```
df_test_5.head()
```

Out[54]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_
0	3000888	2017-08-16	0	0	0	0	0	
1	3000889	2017-08-16	0	0	0	0	0	
2	3000890	2017-08-16	0	0	0	0	0	
3	3000891	2017-08-16	0	0	0	0	0	
4	3000892	2017-08-16	0	0	0	0	0	

5 rows × 25 columns

In [55]:
```
print("Number of rows in training set : ",len(df_train_5))
print("Number of rows in testing set  : ",len(df_test_5))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Checking for "state" feature

In [56]:
```
df_train_5["state"].unique()
```

Out[56]:
```
array(['Pichincha', 'Cotopaxi', 'Chimborazo', 'Imbabura',
       'Santo Domingo de los Tsachilas', 'Bolivar', 'Pastaza',
       'Tungurahua', 'Guayas', 'Santa Elena', 'Los Rios', 'Azuay', 'Loja',
       'El Oro', 'Esmeraldas', 'Manabi'], dtype=object)
```

In [57]:
```
df_test_5["state"].unique()
```

Out[57]:
```
array(['Pichincha', 'Cotopaxi', 'Chimborazo', 'Imbabura',
       'Santo Domingo de los Tsachilas', 'Bolivar', 'Pastaza',
       'Tungurahua', 'Guayas', 'Santa Elena', 'Los Rios', 'Azuay', 'Loja',
       'El Oro', 'Esmeraldas', 'Manabi'], dtype=object)
```

Mapping "state" feature with numbers

In [58]:
```
state = dict()
ctr = 0
for i in df_train_5["state"].unique():
    state[i] = ctr
    ctr += 1
state
```

Out[58]:
```
{'Pichincha': 0,
 'Cotopaxi': 1,
 'Chimborazo': 2,
 'Imbabura': 3,
 'Santo Domingo de los Tsachilas': 4,
 'Bolivar': 5,
 'Pastaza': 6,
 'Tungurahua': 7,
 'Guayas': 8,
 'Santa Elena': 9,
 'Los Rios': 10,
 'Azuay': 11,
 'Loja': 12,
 'El Oro': 13,
 'Esmeraldas': 14,
 'Manabi': 15}
```

In [59]:
```
df_train_6 = df_train_5.copy()
df_test_6 = df_test_5.copy()
df_train_6["state"] = df_train_6["state"].map(state)
df_test_6["state"] = df_test_6["state"].map(state)
```

In [60]:
```
state_train = binary_encoder(df_train_6,"state")
state_test = binary_encoder(df_test_6,"state")
```

In [61]:
```
state_train.head()
```

Out[61]:

	state_0	state_1	state_2	state_3
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

In [62]:
```
state_test.head()
```

Out[62]:

	state_0	state_1	state_2	state_3
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

In [64]:
```
print("Number of rows in training set : ",len(state_train))
print("Number of rows in testing set  : ",len(state_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

In [65]:
```
df_train_7 = df_train_6.copy()
df_test_7 = df_test_6.copy()
df_train_7[state_train.columns.values] = state_train
df_test_7[state_test.columns.values] = state_test
df_train_7.drop("state",axis=1,inplace=True)
df_test_7.drop("state",axis=1,inplace=True)
```

```
df_train_7 = df_train_7[["id","date","store_nbr_0","store_nbr_1","store_nbr_
df_test_7 = df_test_7[["id","date","store_nbr_0","store_nbr_1","store_nbr_2"
```

In [66]:
```
df_train_7.head()
```

Out[66]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
0	0	2013-01-01	0	0	0	0	0	0
1	1	2013-01-01	0	0	0	0	0	0
2	2	2013-01-01	0	0	0	0	0	0
3	3	2013-01-01	0	0	0	0	0	0
4	4	2013-01-01	0	0	0	0	0	0

5 rows × 29 columns

In [67]:
```
df_test_7.head()
```

Out[67]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_
0	3000888	2017-08-16	0	0	0	0	0	
1	3000889	2017-08-16	0	0	0	0	0	
2	3000890	2017-08-16	0	0	0	0	0	
3	3000891	2017-08-16	0	0	0	0	0	
4	3000892	2017-08-16	0	0	0	0	0	

5 rows × 28 columns

In [68]:
```
print("Number of rows in training set : ",len(df_train_7))
print("Number of rows in testing set  : ",len(df_test_7))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Checking for "cluster" feature

In [74]:
```
clu_train = df_train_7["cluster"].unique()
clu_test = df_test_7["cluster"].unique()

clu_train.sort()
clu_test.sort()
```

```
clu_train = clu_train - 1
clu_test = clu_test - 1
```

In [75]:
```
clu_train
```

Out[75]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16])
```

In [76]:
```
clu_test
```

Out[76]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16])
```

In [100…
```
df_train_8 = df_train_7.copy()
df_test_8 = df_test_7.copy()
df_train_8["cluster"] = df_train_8["cluster"] - 1
df_test_8["cluster"] = df_test_8["cluster"] - 1
```

In [101…
```
cluster_train = binary_encoder(df_train_8,"cluster")
cluster_test = binary_encoder(df_test_8,"cluster")
```

In [102…
```
cluster_train.head()
```

Out[102]:

	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
0	0	1	1	0	0
1	0	1	1	0	0
2	0	1	1	0	0
3	0	1	1	0	0
4	0	1	1	0	0

In [103…
```
cluster_test.head()
```

Out[103]:

	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
0	0	1	1	0	0
1	0	1	1	0	0
2	0	1	1	0	0
3	0	1	1	0	0
4	0	1	1	0	0

In [104…
```
print("Number of rows in training set : ",len(cluster_train))
print("Number of rows in testing set  : ",len(cluster_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

In [106…
```
df_train_9 = df_train_8.copy()
df_test_9 = df_test_8.copy()
df_train_9[cluster_train.columns.values] = cluster_train
df_test_9[cluster_test.columns.values] = cluster_test
df_train_9.drop("cluster",axis=1,inplace=True)
df_test_9.drop("cluster",axis=1,inplace=True)
df_train_9 = df_train_9[["id","date","store_nbr_0","store_nbr_1","store_nbr_
df_test_9 = df_test_9[["id","date","store_nbr_0","store_nbr_1","store_nbr_2"
```

In [107…
```
df_train_9.head()
```

Out[107]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
0	0	2013-01-01	0	0	0	0	0	0
1	1	2013-01-01	0	0	0	0	0	0
2	2	2013-01-01	0	0	0	0	0	0
3	3	2013-01-01	0	0	0	0	0	0
4	4	2013-01-01	0	0	0	0	0	0

5 rows × 33 columns

In [108…
```
df_test_9.head()
```

Out[108]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store
0	3000888	2017-08-16	0	0	0	0	0	
1	3000889	2017-08-16	0	0	0	0	0	
2	3000890	2017-08-16	0	0	0	0	0	
3	3000891	2017-08-16	0	0	0	0	0	
4	3000892	2017-08-16	0	0	0	0	0	

5 rows × 32 columns

In [109…
```
print("Number of rows in training set : ",len(df_train_9))
print("Number of rows in testing set  : ",len(df_test_9))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Checking for "type" column

In [112…
```
type_train = df_train_9["type"].unique()
type_test = df_test_9["type"].unique()
type_train.sort()
type_test.sort()
```

In [113…
```
type_train
```

Out[113]:
```
array(['A', 'B', 'C', 'D', 'E'], dtype=object)
```

In [114…
```
type_test
```

Out[114]:
```
array(['A', 'B', 'C', 'D', 'E'], dtype=object)
```

In [115…]:
```
type = dict()
ctr = 0
for i in type_train:
    type[i] = ctr
    ctr += 1
type
```

Out[115]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4}
```

In [116…]:
```
df_train_10 = df_train_9.copy()
df_test_10 = df_test_9.copy()
```

In [117…]:
```
df_train_10["type"] = df_train_10["type"].map(type)
df_test_10["type"] = df_test_10["type"].map(type)
```

In [118…]:
```
type_tr = binary_encoder(df_train_10,"type")
type_te = binary_encoder(df_test_10,"type")
```

In [119…]:
```
type_tr.head()
```

Out[119]:

	type_0	type_1	type_2
0	0	1	1
1	0	1	1
2	0	1	1
3	0	1	1
4	0	1	1

In [120…]:
```
type_te.head()
```

Out[120]:

	type_0	type_1	type_2
0	0	1	1
1	0	1	1
2	0	1	1
3	0	1	1
4	0	1	1

In [121…]:
```
df_train_11 = df_train_10.copy()
df_test_11 = df_test_10.copy()
df_train_11[type_tr.columns.values] = type_tr
df_test_11[type_te.columns.values] = type_te
df_train_11.drop("type",axis=1,inplace=True)
df_test_11.drop("type",axis=1,inplace=True)
df_train_11 = df_train_11[["id","date","store_nbr_0","store_nbr_1","store_nb
df_test_11 = df_test_11[["id","date","store_nbr_0","store_nbr_1","store_nbr_
```

In [122…]:
```
df_train_11.head()
```

Out[122]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store_nbr_5
0	0	2013-01-01	0	0	0	0	0	0
1	1	2013-01-01	0	0	0	0	0	0
2	2	2013-01-01	0	0	0	0	0	0
3	3	2013-01-01	0	0	0	0	0	0
4	4	2013-01-01	0	0	0	0	0	0

5 rows × 35 columns

In [123…
```
df_test_11.head()
```

Out[123]:

	id	date	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	store_nbr_4	store
0	3000888	2017-08-16	0	0	0	0	0	
1	3000889	2017-08-16	0	0	0	0	0	
2	3000890	2017-08-16	0	0	0	0	0	
3	3000891	2017-08-16	0	0	0	0	0	
4	3000892	2017-08-16	0	0	0	0	0	

5 rows × 34 columns

In [124…
```
print("Number of rows in training set : ",len(df_train_11))
print("Number of rows in testing set  : ",len(df_test_11))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

Checking for date feature

In [146…
```
df_train_11["date"].dtype
```

Out[146]:
```
dtype('<M8[ns]')
```

In [147…
```
df_test_11["date"].dtype
```

Out[147]:
```
dtype('<M8[ns]')
```

In [176…
```
y_year = lambda x: x.year
map_year = map(y_year,df_train_11["date"])
lis_year = list(map_year)
```

```
y_month = lambda x: x.month
map_month = map(y_month,df_train_11["date"])
lis_month = list(map_month)

y_day = lambda x: x.day
map_day = map(y_day,df_train_11["date"])
lis_day = list(map_day)

df_train_12 = df_train_11.copy()
df_train_12.drop("date",axis=1,inplace=True)

df_train_12["date_year"] = lis_year
df_train_12["date_month"] = lis_month
df_train_12["date_day"] = lis_day

df_train_12 = df_train_12[["id","date_year","date_month","date_day","store_n
```

In [177…

```
y_year = lambda x: x.year
map_year = map(y_year,df_test_11["date"])
lis_year = list(map_year)

y_month = lambda x: x.month
map_month = map(y_month,df_test_11["date"])
lis_month = list(map_month)

y_day = lambda x: x.day
map_day = map(y_day,df_test_11["date"])
lis_day = list(map_day)

df_test_12 = df_test_11.copy()
df_test_12.drop("date",axis=1,inplace=True)

df_test_12["date_year"] = lis_year
df_test_12["date_month"] = lis_month
df_test_12["date_day"] = lis_day

df_test_12 = df_test_12[["id","date_year","date_month","date_day","store_nbr
```

In [180…

```
df_train_12.head()
```

Out[180]:

	id	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_
0	0	2013	1	1	0	0	0	
1	1	2013	1	1	0	0	0	
2	2	2013	1	1	0	0	0	
3	3	2013	1	1	0	0	0	
4	4	2013	1	1	0	0	0	

5 rows × 37 columns

In [181…

```
df_test_12.head()
```

Out[181]:

	id	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	sto
0	3000888	2017	8	16	0	0	0	
1	3000889	2017	8	16	0	0	0	
2	3000890	2017	8	16	0	0	0	
3	3000891	2017	8	16	0	0	0	
4	3000892	2017	8	16	0	0	0	

5 rows × 36 columns

In [182…
```
print("Number of rows in training set : ",len(df_train_12))
print("Number of rows in testing set  : ",len(df_test_12))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

In [183…
```
retr_df_train = df_train_12.drop("id",axis=1,inplace=False)
retr_df_test = df_test_12.drop("id",axis=1,inplace=False)
```

In [187…
```
retr_df_train.head()
```

Out[187]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3
0	2013	1	1	0	0	0	0
1	2013	1	1	0	0	0	0
2	2013	1	1	0	0	0	0
3	2013	1	1	0	0	0	0
4	2013	1	1	0	0	0	0

5 rows × 36 columns

In [188…
```
retr_df_test.head()
```

Out[188]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3
0	2017	8	16	0	0	0	0
1	2017	8	16	0	0	0	0
2	2017	8	16	0	0	0	0
3	2017	8	16	0	0	0	0
4	2017	8	16	0	0	0	0

5 rows × 35 columns

In [189…
```
print("Number of rows in training set : ",len(retr_df_train))
print("Number of rows in testing set  : ",len(retr_df_test))
```

```
Number of rows in training set :  1048575
Number of rows in testing set  :  28512
```

In [191…
```
X = retr_df_train.drop("sales",axis=1,inplace=False)
y = retr_df_train["sales"]
```

In [192…
```
X.head()
```

Out[192]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3
0	2013	1	1	0	0	0	0
1	2013	1	1	0	0	0	0
2	2013	1	1	0	0	0	0
3	2013	1	1	0	0	0	0
4	2013	1	1	0	0	0	0

5 rows × 35 columns

In [193…
```
y.head()
```

Out[193]:
```
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
Name: sales, dtype: float64
```

In [195…
```
print("Number of rows in X set : ",len(X))
print("Number of rows in y set : ",len(y))
```

```
Number of rows in X set :  1048575
Number of rows in y set :  1048575
```

In [230…
```
f_reg = f_regression(X,y)
p_values1 = f_reg[1]
p_values2 = p_values1.round(2)
p_values = pd.DataFrame(columns=["features","p-values"])
p_values["features"] = X.columns.values
p_values["p-values"] = p_values2
```

In [231…
```
p_values.head()
```

Out[231]:

	features	p-values
0	date_year	0.00
1	date_month	0.05
2	date_day	0.00
3	store_nbr_0	0.00
4	store_nbr_1	0.00

In [232…
```
print("Length of p-values dataframe : ",len(p_values))
```

```
Length of p-values dataframe :  35
```

In [233…
```
p_values.head(5)
```

Out[233]:

	features	p-values
0	date_year	0.00
1	date_month	0.05
2	date_day	0.00
3	store_nbr_0	0.00
4	store_nbr_1	0.00

In [234…
```
p_values[5:11]
```

Out[234]:

	features	p-values
5	store_nbr_2	0.00
6	store_nbr_3	0.46
7	store_nbr_4	0.00
8	store_nbr_5	0.00
9	family_0	0.00
10	family_1	0.00

In [235…
```
p_values[10:15]
```

Out[235]:

	features	p-values
10	family_1	0.0
11	family_2	0.0
12	family_3	0.0
13	family_4	0.0
14	family_5	0.0

In [236…
```
p_values[15:20]
```

Out[236]:

	features	p-values
15	onpromotion	0.0
16	city_0	0.0
17	city_1	0.0
18	city_2	0.0
19	city_3	0.0

In [237…
```
p_values[20:25]
```

Out[237]:

	features	p-values
20	city_4	0.0
21	state_0	0.0
22	state_1	0.0
23	state_2	0.0
24	state_3	0.0

In [238…
```
p_values[25:30]
```

Out[238]:

	features	p-values
25	type_0	0.0
26	type_1	0.0
27	type_2	0.0
28	cluster_0	0.0
29	cluster_1	0.0

In [239…
```
p_values[30:35]
```

Out[239]:

	features	p-values
30	cluster_2	0.0
31	cluster_3	0.0
32	cluster_4	0.0
33	dcoilwtico	0.0
34	holiday?	0.0

In [240…
```
X.to_csv("train_X_preprocessed.csv",index=False)
y.to_csv("train_y_preprocessed.csv",index=False)
retr_df_test.to_csv("test_preprocessed.csv",index=False)
```

In [ ]:
```
```

Machine Learning

Now, we are done with both exploratory data analysis and preprocessing. Now we will go ahead with performing machine learning.

Importing all packages

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import *
from sklearn.linear_model import *
from math import *
from sklearn.ensemble import *
from sklearn.feature_selection import *
from sklearn.feature_extraction import *
from sklearn.naive_bayes import *
from sklearn.discriminant_analysis import *
from sklearn.preprocessing import *
from sklearn.metrics import *
from sklearn.neighbors import *
from sklearn.cluster import *
from sklearn.kernel_approximation import *
from sklearn.svm import *
```

In [2]:
```
X_train = pd.read_csv("train_X_preprocessed.csv")
y_train = pd.read_csv("train_y_preprocessed.csv")
X_test = pd.read_csv("test_preprocessed.csv")
```

In [3]:
```
X_train.head()
```

Out[3]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	s
0	2013	1	1	0	0	0	0	
1	2013	1	1	0	0	0	0	
2	2013	1	1	0	0	0	0	
3	2013	1	1	0	0	0	0	
4	2013	1	1	0	0	0	0	

5 rows × 35 columns

In [4]:
```
y_train.head()
```

Out[4]:

	sales
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0

In [5]:
```
X_test.head()
```

Out[5]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	s
0	2017	8	16	0	0	0	0	
1	2017	8	16	0	0	0	0	
2	2017	8	16	0	0	0	0	
3	2017	8	16	0	0	0	0	
4	2017	8	16	0	0	0	0	

5 rows × 35 columns

In [6]:
```
print("Number of rows in X_train : ",len(X_train))
print("Number of rows in y_train : ",len(y_train))
print("Number of rows in X_test  : ",len(X_test))
```

```
Number of rows in X_train :  1048575
Number of rows in y_train :  1048575
Number of rows in X_test  :  28512
```

In [7]:
```
X_train.dtypes
```

Out[7]:
```
date_year         int64
date_month        int64
date_day          int64
store_nbr_0       int64
store_nbr_1       int64
store_nbr_2       int64
store_nbr_3       int64
store_nbr_4       int64
store_nbr_5       int64
family_0          int64
family_1          int64
family_2          int64
family_3          int64
family_4          int64
family_5          int64
onpromotion       int64
city_0            int64
city_1            int64
city_2            int64
city_3            int64
city_4            int64
state_0           int64
state_1           int64
state_2           int64
state_3           int64
type_0            int64
type_1            int64
type_2            int64
cluster_0         int64
cluster_1         int64
cluster_2         int64
cluster_3         int64
cluster_4         int64
dcoilwtico      float64
holiday?        float64
dtype: object
```

In [8]:
```
X_test.dtypes
```

Out[8]:
```
date_year         int64
date_month        int64
date_day          int64
store_nbr_0       int64
store_nbr_1       int64
store_nbr_2       int64
store_nbr_3       int64
store_nbr_4       int64
store_nbr_5       int64
family_0          int64
family_1          int64
family_2          int64
family_3          int64
family_4          int64
family_5          int64
onpromotion       int64
city_0            int64
city_1            int64
city_2            int64
city_3            int64
city_4            int64
state_0           int64
state_1           int64
state_2           int64
state_3           int64
type_0            int64
type_1            int64
type_2            int64
cluster_0         int64
cluster_1         int64
cluster_2         int64
cluster_3         int64
cluster_4         int64
dcoilwtico      float64
holiday?        float64
dtype: object
```

In [9]:
```
y_train.dtypes
```

Out[9]:
```
sales    float64
dtype: object
```

In [10]:
```
X_total = pd.concat([X_train,X_test])
X_total.head()
```

Out[10]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	s
0	2013	1	1	0	0	0	0	
1	2013	1	1	0	0	0	0	
2	2013	1	1	0	0	0	0	
3	2013	1	1	0	0	0	0	
4	2013	1	1	0	0	0	0	

5 rows × 35 columns

In [11]:
```
col1 = ["store_nbr_0","store_nbr_1","store_nbr_2","store_nbr_3","store_nbr_4
```

In [12]:
```
col2 = ["date_year","date_month","date_day","dcoilwtico"]
```

In [13]:
```
len(col1)+len(col2)
```

Out[13]:
```
35
```

In [14]:
```
ss = StandardScaler()

X_total_1 = X_total.copy()
h = X_total[col2].to_numpy()
g = ss.fit_transform(h)
o = pd.DataFrame(g,columns=col2)
X_total_1[col2] = o
X_total_1 = X_total_1[X_total.columns.values]
X_total_1 = X_total_1.to_numpy()
```

In [15]:
```
X_train_1 = pd.DataFrame(X_total_1[0:len(X_train),:],columns=X_train.columns
X_test_1 = pd.DataFrame(X_total_1[len(X_train):,:],columns=X_test.columns.va
y_train_1 = y_train["sales"]
```

In [16]:
```
X_train_1.head()
```

Out[16]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	s
0	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
1	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
2	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
3	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
4	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	

5 rows × 35 columns

In [17]:
```
X_test_1.head()
```

Out[17]:

	date_year	date_month	date_day	store_nbr_0	store_nbr_1	store_nbr_2	store_nbr_3	s
0	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
1	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
2	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
3	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	
4	-0.633647	-1.475632	-1.668899	0.0	0.0	0.0	0.0	

5 rows × 35 columns

In [18]:
```
y_train_1.head()
```

Out[18]:
```
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
Name: sales, dtype: float64
```

In [19]:
```
print("Number of rows in X_train : ",len(X_train_1))
print("Number of rows in y_train : ",len(y_train_1))
print("Number of rows in X_test  : ",len(X_test_1))
```

```
Number of rows in X_train :  1048575
Number of rows in y_train :  1048575
Number of rows in X_test  :  28512
```

In [20]:
```
sgd = SGDRegressor()
model = sgd.fit(X_train_1,y_train_1)
```

In [21]:
```
y_test_1 = model.predict(X_test_1)
```

In [22]:
```
y_test_1
```

Out[22]:
```
array([356.97883586, 269.66602761, 288.34827773, ..., 218.09859663,
       347.65563895,   9.59378225])
```

In [23]:
```
org = pd.read_csv("test.csv")
id = org["id"]
id.head()
```

Out[23]:
```
0    3000888
1    3000889
2    3000890
3    3000891
4    3000892
Name: id, dtype: int64
```

In [24]:
```
final_df = pd.DataFrame(columns=["id","sales"])
final_df["id"] = id
final_df["sales"] = y_test_1.round(2)
```

In [25]:
```
final_df.head()
```

Out[25]:

	id	sales
0	3000888	356.98
1	3000889	269.67
2	3000890	288.35
3	3000891	688.99
4	3000892	497.23

In [26]:
```
final_df.to_csv("amith_submission.csv",index=False)
```

In [27]:
```
final_df2 = pd.DataFrame(columns=["id","sales"])
final_df2["id"] = id
final_df2["sales"] = y_test_1
```

In [28]:
```
final_df2.head()
```

Out[28]:

	id	sales
0	3000888	356.978836
1	3000889	269.666028
2	3000890	288.348278
3	3000891	688.992633
4	3000892	497.230492

In [29]:
```
final_df2.to_csv("amith_submission2.csv",index=False)
```

In [ ]:
```

```