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 [ ]:
```

```