IMPORT DATASET AND LIBRARIES

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import zipfile
```

In [2]:
```
feature = pd.read_csv('Features_data_set.csv')
sales = pd.read_csv('sales_data_set.csv')
stores = pd.read_csv('stores_data_set.csv')
```

In [3]:
```
stores
```

Out[3]:

	Store	Type	Size
0	1	A	151315
1	2	A	202307
2	3	B	37392
3	4	A	205863
4	5	B	34875
5	6	A	202505
6	7	B	70713
7	8	A	155078
8	9	B	125833
9	10	B	126512
10	11	A	207499
11	12	B	112238
12	13	A	219622
13	14	A	200898
14	15	B	123737
15	16	B	57197
16	17	B	93188
17	18	B	120653
18	19	A	203819
19	20	A	203742
20	21	B	140167
21	22	B	119557
22	23	B	114533
23	24	A	203819
24	25	B	128107
25	26	A	152513
26	27	A	204184
27	28	A	206302
28	29	B	93638
29	30	C	42988
30	31	A	203750
31	32	A	203007
32	33	A	39690
33	34	A	158114
34	35	B	103681
35	36	A	39910
36	37	C	39910
37	38	C	39690
38	39	A	184109
39	40	A	155083
40	41	A	196321
41	42	C	39690
42	43	C	41062
43	44	C	39910
44	45	B	118221

In [4]:
```
feature
```

Out[4]:

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	05/02/2010	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
1	1	12/02/2010	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106	True
2	1	19/02/2010	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106	False

3	1	26/02/2010	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106	False
4	1	05/03/2010	46.50	2.625	NaN	NaN	NaN	NaN	NaN	211.350143	8.106	False
5	1	12/03/2010	57.79	2.667	NaN	NaN	NaN	NaN	NaN	211.380643	8.106	False
6	1	19/03/2010	54.58	2.720	NaN	NaN	NaN	NaN	NaN	211.215635	8.106	False
7	1	26/03/2010	51.45	2.732	NaN	NaN	NaN	NaN	NaN	211.018042	8.106	False
8	1	02/04/2010	62.27	2.719	NaN	NaN	NaN	NaN	NaN	210.820450	7.808	False
9	1	09/04/2010	65.86	2.770	NaN	NaN	NaN	NaN	NaN	210.622857	7.808	False
10	1	16/04/2010	66.32	2.808	NaN	NaN	NaN	NaN	NaN	210.488700	7.808	False
11	1	23/04/2010	64.84	2.795	NaN	NaN	NaN	NaN	NaN	210.439123	7.808	False
12	1	30/04/2010	67.41	2.780	NaN	NaN	NaN	NaN	NaN	210.389546	7.808	False
13	1	07/05/2010	72.55	2.835	NaN	NaN	NaN	NaN	NaN	210.339968	7.808	False
14	1	14/05/2010	74.78	2.854	NaN	NaN	NaN	NaN	NaN	210.337426	7.808	False
15	1	21/05/2010	76.44	2.826	NaN	NaN	NaN	NaN	NaN	210.617093	7.808	False
16	1	28/05/2010	80.44	2.759	NaN	NaN	NaN	NaN	NaN	210.896761	7.808	False
17	1	04/06/2010	80.69	2.705	NaN	NaN	NaN	NaN	NaN	211.176428	7.808	False
18	1	11/06/2010	80.43	2.668	NaN	NaN	NaN	NaN	NaN	211.456095	7.808	False
19	1	18/06/2010	84.11	2.637	NaN	NaN	NaN	NaN	NaN	211.453772	7.808	False
20	1	25/06/2010	84.34	2.653	NaN	NaN	NaN	NaN	NaN	211.338653	7.808	False
21	1	02/07/2010	80.91	2.669	NaN	NaN	NaN	NaN	NaN	211.223533	7.787	False
22	1	09/07/2010	80.48	2.642	NaN	NaN	NaN	NaN	NaN	211.108414	7.787	False
23	1	16/07/2010	83.15	2.623	NaN	NaN	NaN	NaN	NaN	211.100385	7.787	False
24	1	23/07/2010	83.36	2.608	NaN	NaN	NaN	NaN	NaN	211.235144	7.787	False
25	1	30/07/2010	81.84	2.640	NaN	NaN	NaN	NaN	NaN	211.369903	7.787	False
26	1	06/08/2010	87.16	2.627	NaN	NaN	NaN	NaN	NaN	211.504662	7.787	False
27	1	13/08/2010	87.00	2.692	NaN	NaN	NaN	NaN	NaN	211.639421	7.787	False
28	1	20/08/2010	86.65	2.664	NaN	NaN	NaN	NaN	NaN	211.603363	7.787	False
29	1	27/08/2010	85.22	2.619	NaN	NaN	NaN	NaN	NaN	211.567306	7.787	False
...	...	...	...	...	...	...	...	...	...	...	...	...
8160	45	04/01/2013	32.87	3.592	1341.33	30325.14	8.93	35.85	3682.17	192.659622	8.625	False
8161	45	11/01/2013	38.78	3.611	3877.36	15559.85	3.81	152.18	2403.14	192.759980	8.625	False
8162	45	18/01/2013	41.45	3.605	14746.10	4071.06	4.02	483.58	1467.78	192.809507	8.625	False
8163	45	25/01/2013	26.49	3.583	3130.28	1362.10	0.20	332.68	1361.74	192.838701	8.625	False
8164	45	01/02/2013	34.92	3.615	14508.96	1092.53	827.90	26424.02	700.93	192.867895	8.625	False
8165	45	08/02/2013	28.99	3.753	53311.88	531.33	78.26	24823.94	3233.44	192.897089	8.625	True
8166	45	15/02/2013	35.87	3.814	9362.02	2017.68	0.36	5012.39	6411.71	192.943471	8.625	False
8167	45	22/02/2013	31.48	3.859	10781.51	2735.67	23.38	1516.76	3061.21	193.032822	8.625	False
8168	45	01/03/2013	39.72	3.890	6614.32	147.82	5.60	27.55	1668.95	193.122173	8.625	False
8169	45	08/03/2013	36.13	3.860	16382.54	88.67	34.62	3096.92	3486.91	193.211524	8.625	False
8170	45	15/03/2013	42.81	3.834	9867.03	NaN	11.08	912.87	1360.36	193.296277	8.625	False
8171	45	22/03/2013	36.55	3.800	11923.74	NaN	308.00	1764.47	1647.31	193.369533	8.625	False
8172	45	29/03/2013	40.68	3.784	5444.00	NaN	350.84	53.90	1722.11	193.442790	8.625	False
8173	45	05/04/2013	43.94	3.763	16427.83	5341.41	182.59	1523.83	1743.09	193.516047	8.335	False
8174	45	12/04/2013	57.39	3.724	8760.15	1713.11	21.08	1302.31	1380.74	193.589304	8.335	False
8175	45	19/04/2013	56.27	3.676	1399.81	39.89	44.38	60.83	1445.05	193.589304	8.335	False
8176	45	26/04/2013	50.64	3.615	1260.65	NaN	57.52	40.51	2476.18	193.589304	8.335	False
8177	45	03/05/2013	56.07	3.592	8345.40	6.00	92.96	3580.32	2242.24	NaN	NaN	False
8178	45	10/05/2013	58.86	3.583	4689.18	440.82	53.09	375.22	5738.20	NaN	NaN	False
8179	45	17/05/2013	60.59	3.614	4515.35	667.88	6.12	522.70	2541.62	NaN	NaN	False
8180	45	24/05/2013	67.11	3.627	3249.34	481.82	58.48	1183.23	1309.30	NaN	NaN	False
8181	45	31/05/2013	65.88	3.646	6474.49	411.38	77.06	9.38	4227.27	NaN	NaN	False
8182	45	07/06/2013	70.71	3.633	9977.82	744.29	80.00	4825.71	3597.34	NaN	NaN	False
8183	45	14/06/2013	70.01	3.632	2471.44	517.87	348.54	2612.33	3459.39	NaN	NaN	False
8184	45	21/06/2013	70.13	3.626	4989.34	385.31	178.56	2463.42	3117.94	NaN	NaN	False
8185	45	28/06/2013	76.05	3.639	4842.29	975.03	3.00	2449.97	3169.69	NaN	NaN	False
8186	45	05/07/2013	77.50	3.614	9090.48	2268.58	582.74	5797.47	1514.93	NaN	NaN	False
8187	45	12/07/2013	79.37	3.614	3789.94	1827.31	85.72	744.84	2150.36	NaN	NaN	False
8188	45	19/07/2013	82.84	3.737	2961.49	1047.07	204.19	363.00	1059.46	NaN	NaN	False
8189	45	26/07/2013	76.06	3.804	212.02	851.73	2.06	10.88	1864.57	NaN	NaN	False

8190 rows × 12 columns

In [5]:
```
sales
```

Out[5]:

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	05/02/2010	24924.50	False
1	1	1	12/02/2010	46039.49	True
2	1	1	19/02/2010	41595.55	False
3	1	1	26/02/2010	19403.54	False
4	1	1	05/03/2010	21827.90	False
5	1	1	12/03/2010	21043.39	False
6	1	1	19/03/2010	22136.64	False
7	1	1	26/03/2010	26229.21	False

8	1	1	02/04/2010	57258.43	False
9	1	1	09/04/2010	42960.91	False
10	1	1	16/04/2010	17596.96	False
11	1	1	23/04/2010	16145.35	False
12	1	1	30/04/2010	16555.11	False
13	1	1	07/05/2010	17413.94	False
14	1	1	14/05/2010	18926.74	False
15	1	1	21/05/2010	14773.04	False
16	1	1	28/05/2010	15580.43	False
17	1	1	04/06/2010	17558.09	False
18	1	1	11/06/2010	16637.62	False
19	1	1	18/06/2010	16216.27	False
20	1	1	25/06/2010	16328.72	False
21	1	1	02/07/2010	16333.14	False
22	1	1	09/07/2010	17688.76	False
23	1	1	16/07/2010	17150.84	False
24	1	1	23/07/2010	15360.45	False
25	1	1	30/07/2010	15381.82	False
26	1	1	06/08/2010	17508.41	False
27	1	1	13/08/2010	15536.40	False
28	1	1	20/08/2010	15740.13	False
29	1	1	27/08/2010	15793.87	False
...	...	...	...	...	...
421540	45	98	06/04/2012	778.70	False
421541	45	98	13/04/2012	559.14	False
421542	45	98	20/04/2012	605.80	False
421543	45	98	27/04/2012	619.41	False
421544	45	98	04/05/2012	694.25	False
421545	45	98	11/05/2012	893.60	False
421546	45	98	18/05/2012	745.44	False
421547	45	98	25/05/2012	795.94	False
421548	45	98	01/06/2012	874.64	False
421549	45	98	08/06/2012	713.50	False
421550	45	98	15/06/2012	856.35	False
421551	45	98	22/06/2012	622.62	False
421552	45	98	29/06/2012	690.52	False
421553	45	98	06/07/2012	659.65	False
421554	45	98	13/07/2012	695.21	False
421555	45	98	20/07/2012	845.30	False
421556	45	98	27/07/2012	657.63	False
421557	45	98	03/08/2012	516.46	False
421558	45	98	10/08/2012	727.49	False
421559	45	98	17/08/2012	500.16	False
421560	45	98	24/08/2012	415.40	False
421561	45	98	31/08/2012	346.04	False
421562	45	98	07/09/2012	352.44	True
421563	45	98	14/09/2012	605.96	False
421564	45	98	21/09/2012	467.30	False
421565	45	98	28/09/2012	508.37	False
421566	45	98	05/10/2012	628.10	False
421567	45	98	12/10/2012	1061.02	False
421568	45	98	19/10/2012	760.01	False
421569	45	98	26/10/2012	1076.80	False

421570 rows × 5 columns

In [6]:
```
feature['Date'] = pd.to_datetime(feature['Date'])
sales['Date'] = pd.to_datetime(sales['Date'])
```

In [7]:
```
feature
```

Out[7]:

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	2010-05-02	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
1	1	2010-12-02	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106	True
2	1	2010-02-19	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106	False
3	1	2010-02-26	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106	False
4	1	2010-05-03	46.50	2.625	NaN	NaN	NaN	NaN	NaN	211.350143	8.106	False
5	1	2010-12-03	57.79	2.667	NaN	NaN	NaN	NaN	NaN	211.380643	8.106	False
6	1	2010-03-19	54.58	2.720	NaN	NaN	NaN	NaN	NaN	211.215635	8.106	False
7	1	2010-03-26	51.45	2.732	NaN	NaN	NaN	NaN	NaN	211.018042	8.106	False
8	1	2010-02-04	62.27	2.719	NaN	NaN	NaN	NaN	NaN	210.820450	7.808	False
9	1	2010-09-04	65.86	2.770	NaN	NaN	NaN	NaN	NaN	210.622857	7.808	False

10	1	2010-04-16	66.32	2.808	NaN	NaN	NaN	NaN	NaN	210.488700	7.808	False
11	1	2010-04-23	64.84	2.795	NaN	NaN	NaN	NaN	NaN	210.439123	7.808	False
12	1	2010-04-30	67.41	2.780	NaN	NaN	NaN	NaN	NaN	210.389546	7.808	False
13	1	2010-07-05	72.55	2.835	NaN	NaN	NaN	NaN	NaN	210.339968	7.808	False
14	1	2010-05-14	74.78	2.854	NaN	NaN	NaN	NaN	NaN	210.337426	7.808	False
15	1	2010-05-21	76.44	2.826	NaN	NaN	NaN	NaN	NaN	210.617093	7.808	False
16	1	2010-05-28	80.44	2.759	NaN	NaN	NaN	NaN	NaN	210.896761	7.808	False
17	1	2010-04-06	80.69	2.705	NaN	NaN	NaN	NaN	NaN	211.176428	7.808	False
18	1	2010-11-06	80.43	2.668	NaN	NaN	NaN	NaN	NaN	211.456095	7.808	False
19	1	2010-06-18	84.11	2.637	NaN	NaN	NaN	NaN	NaN	211.453772	7.808	False
20	1	2010-06-25	84.34	2.653	NaN	NaN	NaN	NaN	NaN	211.338653	7.808	False
21	1	2010-02-07	80.91	2.669	NaN	NaN	NaN	NaN	NaN	211.223533	7.787	False
22	1	2010-09-07	80.48	2.642	NaN	NaN	NaN	NaN	NaN	211.108414	7.787	False
23	1	2010-07-16	83.15	2.623	NaN	NaN	NaN	NaN	NaN	211.100385	7.787	False
24	1	2010-07-23	83.36	2.608	NaN	NaN	NaN	NaN	NaN	211.235144	7.787	False
25	1	2010-07-30	81.84	2.640	NaN	NaN	NaN	NaN	NaN	211.369903	7.787	False
26	1	2010-06-08	87.16	2.627	NaN	NaN	NaN	NaN	NaN	211.504662	7.787	False
27	1	2010-08-13	87.00	2.692	NaN	NaN	NaN	NaN	NaN	211.639421	7.787	False
28	1	2010-08-20	86.65	2.664	NaN	NaN	NaN	NaN	NaN	211.603363	7.787	False
29	1	2010-08-27	85.22	2.619	NaN	NaN	NaN	NaN	NaN	211.567306	7.787	False
...	...	...	...	...	...	...	...	...	...	...	...	...
8160	45	2013-04-01	32.87	3.592	1341.33	30325.14	8.93	35.85	3682.17	192.659622	8.625	False
8161	45	2013-11-01	38.78	3.611	3877.36	15559.85	3.81	152.18	2403.14	192.759980	8.625	False
8162	45	2013-01-18	41.45	3.605	14746.10	4071.06	4.02	483.58	1467.78	192.809507	8.625	False
8163	45	2013-01-25	26.49	3.583	3130.28	1362.10	0.20	332.68	1361.74	192.838701	8.625	False
8164	45	2013-01-02	34.92	3.615	14508.96	1092.53	827.90	26424.02	700.93	192.867895	8.625	False
8165	45	2013-08-02	28.99	3.753	53311.88	531.33	78.26	24823.94	3233.44	192.897089	8.625	True
8166	45	2013-02-15	35.87	3.814	9362.02	2017.68	0.36	5012.39	6411.71	192.943471	8.625	False
8167	45	2013-02-22	31.48	3.859	10781.51	2735.67	23.38	1516.76	3061.21	193.032822	8.625	False
8168	45	2013-01-03	39.72	3.890	6614.32	147.82	5.60	27.55	1668.95	193.122173	8.625	False
8169	45	2013-08-03	36.13	3.860	16382.54	88.67	34.62	3096.92	3486.91	193.211524	8.625	False
8170	45	2013-03-15	42.81	3.834	9867.03	NaN	11.08	912.87	1360.36	193.296277	8.625	False
8171	45	2013-03-22	36.55	3.800	11923.74	NaN	308.00	1764.47	1647.31	193.369533	8.625	False
8172	45	2013-03-29	40.68	3.784	5444.00	NaN	350.84	53.90	1722.11	193.442790	8.625	False
8173	45	2013-05-04	43.94	3.763	16427.83	5341.41	182.59	1523.83	1743.09	193.516047	8.335	False
8174	45	2013-12-04	57.39	3.724	8760.15	1713.11	21.08	1302.31	1380.74	193.589304	8.335	False
8175	45	2013-04-19	56.27	3.676	1399.81	39.89	44.38	60.83	1445.05	193.589304	8.335	False
8176	45	2013-04-26	50.64	3.615	1260.65	NaN	57.52	40.51	2476.18	193.589304	8.335	False
8177	45	2013-03-05	56.07	3.592	8345.40	6.00	92.96	3580.32	2242.24	NaN	NaN	False
8178	45	2013-10-05	58.86	3.583	4689.18	440.82	53.09	375.22	5738.20	NaN	NaN	False
8179	45	2013-05-17	60.59	3.614	4515.35	667.88	6.12	522.70	2541.62	NaN	NaN	False
8180	45	2013-05-24	67.11	3.627	3249.34	481.82	58.48	1183.23	1309.30	NaN	NaN	False
8181	45	2013-05-31	65.88	3.646	6474.49	411.38	77.06	9.38	4227.27	NaN	NaN	False
8182	45	2013-07-06	70.71	3.633	9977.82	744.29	80.00	4825.71	3597.34	NaN	NaN	False
8183	45	2013-06-14	70.01	3.632	2471.44	517.87	348.54	2612.33	3459.39	NaN	NaN	False
8184	45	2013-06-21	70.13	3.626	4989.34	385.31	178.56	2463.42	3117.94	NaN	NaN	False
8185	45	2013-06-28	76.05	3.639	4842.29	975.03	3.00	2449.97	3169.69	NaN	NaN	False
8186	45	2013-05-07	77.50	3.614	9090.48	2268.58	582.74	5797.47	1514.93	NaN	NaN	False
8187	45	2013-12-07	79.37	3.614	3789.94	1827.31	85.72	744.84	2150.36	NaN	NaN	False
8188	45	2013-07-19	82.84	3.737	2961.49	1047.07	204.19	363.00	1059.46	NaN	NaN	False
8189	45	2013-07-26	76.06	3.804	212.02	851.73	2.06	10.88	1864.57	NaN	NaN	False

8190 rows × 12 columns

In [8]:
```
sales
```

Out[8]:

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	2010-05-02	24924.50	False
1	1	1	2010-12-02	46039.49	True
2	1	1	2010-02-19	41595.55	False
3	1	1	2010-02-26	19403.54	False
4	1	1	2010-05-03	21827.90	False
5	1	1	2010-12-03	21043.39	False
6	1	1	2010-03-19	22136.64	False
7	1	1	2010-03-26	26229.21	False
8	1	1	2010-02-04	57258.43	False
9	1	1	2010-09-04	42960.91	False
10	1	1	2010-04-16	17596.96	False
11	1	1	2010-04-23	16145.35	False
12	1	1	2010-04-30	16555.11	False
13	1	1	2010-07-05	17413.94	False

14	1	1	2010-05-14	18926.74	False
15	1	1	2010-05-21	14773.04	False
16	1	1	2010-05-28	15580.43	False
17	1	1	2010-04-06	17558.09	False
18	1	1	2010-11-06	16637.62	False
19	1	1	2010-06-18	16216.27	False
20	1	1	2010-06-25	16328.72	False
21	1	1	2010-02-07	16333.14	False
22	1	1	2010-09-07	17688.76	False
23	1	1	2010-07-16	17150.84	False
24	1	1	2010-07-23	15360.45	False
25	1	1	2010-07-30	15381.82	False
26	1	1	2010-06-08	17508.41	False
27	1	1	2010-08-13	15536.40	False
28	1	1	2010-08-20	15740.13	False
29	1	1	2010-08-27	15793.87	False
...	...	...	...	...	...
421540	45	98	2012-06-04	778.70	False
421541	45	98	2012-04-13	559.14	False
421542	45	98	2012-04-20	605.80	False
421543	45	98	2012-04-27	619.41	False
421544	45	98	2012-04-05	694.25	False
421545	45	98	2012-11-05	893.60	False
421546	45	98	2012-05-18	745.44	False
421547	45	98	2012-05-25	795.94	False
421548	45	98	2012-01-06	874.64	False
421549	45	98	2012-08-06	713.50	False
421550	45	98	2012-06-15	856.35	False
421551	45	98	2012-06-22	622.62	False
421552	45	98	2012-06-29	690.52	False
421553	45	98	2012-06-07	659.65	False
421554	45	98	2012-07-13	695.21	False
421555	45	98	2012-07-20	845.30	False
421556	45	98	2012-07-27	657.63	False
421557	45	98	2012-03-08	516.46	False
421558	45	98	2012-10-08	727.49	False
421559	45	98	2012-08-17	500.16	False
421560	45	98	2012-08-24	415.40	False
421561	45	98	2012-08-31	346.04	False
421562	45	98	2012-07-09	352.44	True
421563	45	98	2012-09-14	605.96	False
421564	45	98	2012-09-21	467.30	False
421565	45	98	2012-09-28	508.37	False
421566	45	98	2012-05-10	628.10	False
421567	45	98	2012-12-10	1061.02	False
421568	45	98	2012-10-19	760.01	False
421569	45	98	2012-10-26	1076.80	False

421570 rows × 5 columns

MERGE DATASET INTO ONE DATAFRAME

In [9]:
```
df = pd.merge(sales, feature, on = ['Store','Date','IsHoliday'])
```

In [10]:
```
df
```

Out[10]:

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI
0	1	1	2010-05-02	24924.50	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
1	1	2	2010-05-02	50605.27	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
2	1	3	2010-05-02	13740.12	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
3	1	4	2010-05-02	39954.04	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
4	1	5	2010-05-02	32229.38	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
5	1	6	2010-05-02	5749.03	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
6	1	7	2010-05-02	21084.08	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358

In [11]:
```
df.head()
```

Out[11]:

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemp

0	1	1	2010-05-02	24924.50	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
1	1	2	2010-05-02	50605.27	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
2	1	3	2010-05-02	13740.12	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
3	1	4	2010-05-02	39954.04	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358
4	1	5	2010-05-02	32229.38	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358

In [12]:
```
df = pd.merge(df, stores, on = ['Store'], how = 'left')
```

In [13]:
```
df.head()
```

Out[13]:

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemp
0	1	1	2010-05-02	24924.50	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
1	1	2	2010-05-02	50605.27	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
2	1	3	2010-05-02	13740.12	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
3	1	4	2010-05-02	39954.04	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
4	1	5	2010-05-02	32229.38	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	

EXPLORE MERGED DATASET

In [14]:
```
# check the number of non-null values in the dataframe
df.isnull().sum()
```

Out[14]:
```
Store              0
Dept               0
Date               0
Weekly_Sales       0
IsHoliday          0
Temperature        0
Fuel_Price         0
MarkDown1     270889
MarkDown2     310322
MarkDown3     284479
MarkDown4     286603
MarkDown5     270138
CPI                0
Unemployment       0
Type               0
Size               0
dtype: int64
```

In [15]:
```
# Fill up NaN elements with zeros
df = df.fillna(0)
```

In [16]:
```
df
```

Out[16]:

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI
0	1	1	2010-05-02	24924.50	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358
1	1	2	2010-05-02	50605.27	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358
2	1	3	2010-05-02	13740.12	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358
3	1	4	2010-05-02	39954.04	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358
4	1	5	2010-05-02	32229.38	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358
5	1	6	2010-05-02	5749.03	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358
6	1	7	2010-05-02	21084.08	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358

In [17]:
```
# Statistical summary of the combined dataframe
df.describe()
```

Out[17]:

	Store	Dept	Weekly_Sales	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown
count	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.0000
mean	22.200546	44.260317	15981.258123	60.090059	3.361027	2590.074819	879.974298	468.087665	1083.132268	1662.7723
std	12.785297	30.492054	22711.183519	18.447931	0.458515	6052.385934	5084.538801	5528.873453	3894.529945	4207.6293
min	1.000000	1.000000	-4988.940000	-2.060000	2.472000	0.000000	-265.760000	-29.100000	0.000000	0.0000
25%	11.000000	18.000000	2079.650000	46.680000	2.933000	0.000000	0.000000	0.000000	0.000000	0.0000
50%	22.000000	37.000000	7612.030000	62.090000	3.452000	0.000000	0.000000	0.000000	0.000000	0.0000
75%	33.000000	74.000000	20205.852500	74.280000	3.738000	2809.050000	2.200000	4.540000	425.290000	2168.0400
max	45.000000	99.000000	693099.360000	100.140000	4.468000	88646.760000	104519.540000	141630.610000	67474.850000	108519.2800

In [18]:
```
df['Type'].value_counts()
```

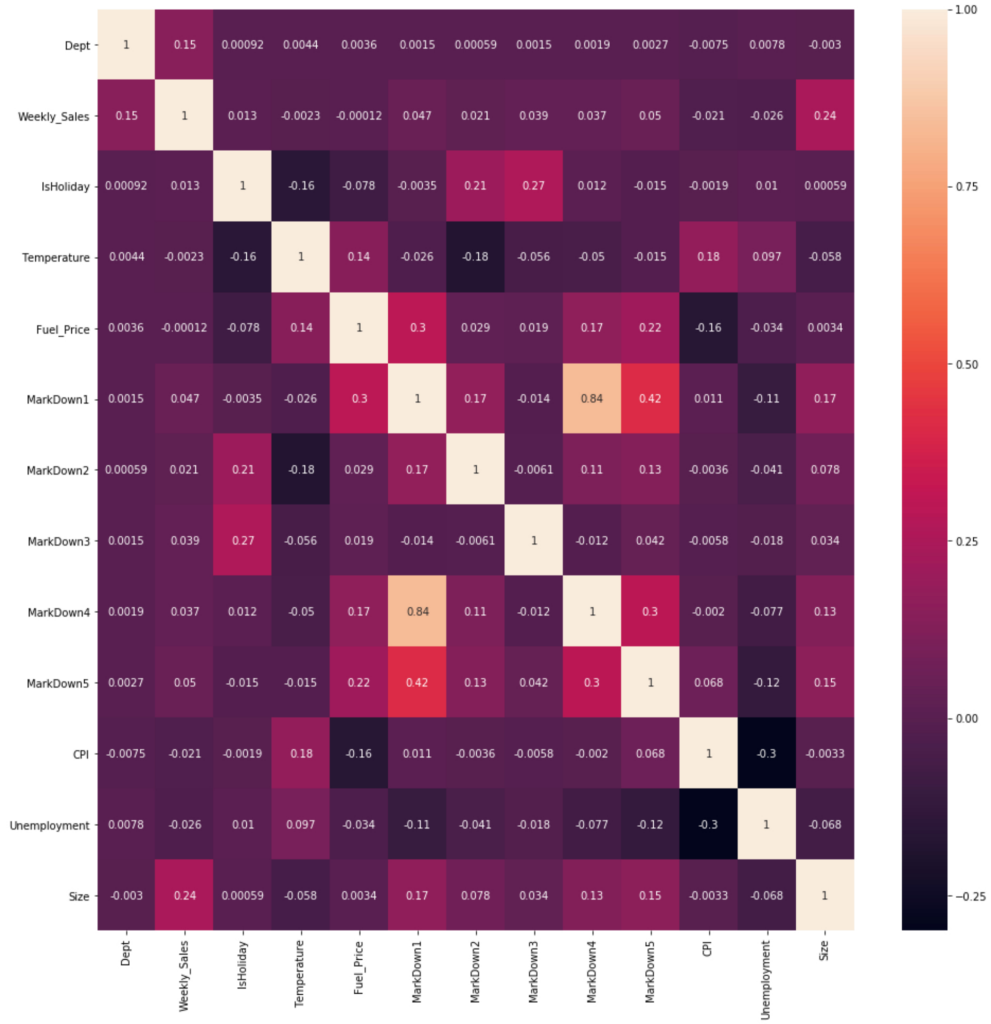
Out[18]:
```
A    215478
```

```
B    163495
C     42597
Name: Type, dtype: int64
```

PERFORM EXPLORATORY DATA ANALYSIS

In [19]:
```
corr_matrix = df.drop(columns = ['Store']).corr()
```

In [20]:
```
plt.figure(figsize = (16,16))
sns.heatmap(corr_matrix, annot = True)
plt.show()
```



PERFORM DATA VISUALIZATION

In [21]:
```
df_type = df.groupby('Type').mean()
```

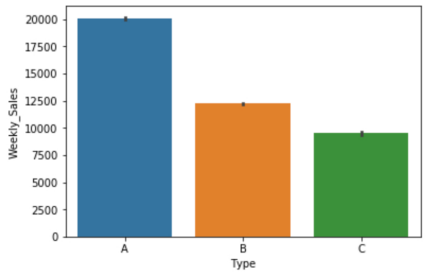
In [22]:
```
df_type
```

Out[22]:

Type	Store	Dept	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	U
A	21.736419	44.622156	20099.568043	0.070471	60.531945	3.343999	3102.403194	1083.216159	549.644930	1325.891281	2147.830168	174.408895	
B	18.450417	43.112273	12237.075977	0.070412	57.562951	3.382523	2553.465968	827.500452	481.215226	1043.927675	1324.921913	167.176656	
C	38.942015	46.836350	9519.532538	0.069582	67.554266	3.364654	138.960203	53.274338	5.142226	5.603993	505.826631	170.429314	

In [23]:
```
sns.barplot(x = df['Type'], y = df['Weekly_Sales'], data = df)
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2ecdc31208>
```



PREPARE THE DATA BEFORE TRAINING

In [24]:
```
df_target = df['Weekly_Sales']
df_final = df.drop(columns = ['Weekly_Sales', 'Date'])
```

In [25]:
```
df_final = pd.get_dummies(df_final, columns = ['Type', 'Store', 'Dept'], drop_first = True)
```

In [26]:
```
df_final.shape
```

Out[26]: (421570, 137)

In [27]:
```
df_target.shape
```

Out[27]: (421570,)

In [28]:
```
df_final
```

Out[28]:

	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	...	Dept_90	Dept_9
0	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
1	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
2	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
3	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
4	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
5	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
6	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
7	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
8	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
9	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
10	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
11	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
12	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
13	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
14	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
15	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
16	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
17	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
18	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
19	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
20	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
21	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
22	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
23	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
24	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
25	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	
26	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	

27	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	(
28	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	(
29	False	42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	...	0	(
...	...	...	...	...	...	...	...	...	...	...	...	...	..
421540	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421541	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421542	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421543	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421544	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421545	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421546	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421547	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421548	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421549	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421550	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421551	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421552	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421553	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421554	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421555	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421556	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421557	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421558	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421559	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421560	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421561	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421562	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	1	(
421563	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	'
421564	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421565	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421566	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421567	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421568	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(
421569	False	58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	...	0	(

421570 rows × 137 columns

In [29]:
```
X = np.array(df_final).astype('float32')
y = np.array(df_target).astype('float32')
```

In [30]:
```
y = y.reshape(-1,1)
y.shape
```

Out[30]:
```
(421570, 1)
```

In [31]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.15)
X_test, X_val, y_test, y_val = train_test_split(X_test, y_test, test_size = 0.5)
```

In [32]:
```
X_train
```

Out[32]:
```
array([[ 0.   , 74.69 ,  3.711, ...,  0.   ,  0.   ,  0.   ],
       [ 0.   , 68.12 ,  3.014, ...,  0.   ,  0.   ,  0.   ],
       [ 0.   , 43.15 ,  3.312, ...,  0.   ,  0.   ,  0.   ],
       ...,
       [ 0.   , 74.36 ,  3.827, ...,  0.   ,  0.   ,  0.   ],
       [ 0.   , 78.04 ,  3.659, ...,  0.   ,  0.   ,  0.   ],
       [ 0.   , 68.5  ,  2.725, ...,  0.   ,  0.   ,  0.   ]],
      dtype=float32)
```

TRAIN XGBOOST USING SAGEMAKER

In [33]:
```
train_data = pd.DataFrame({'Target': y_train[:,0]})
for i in range(X_train.shape[1]):
    train_data[i] = X_train[:,i]
```

In [34]:
```
train_data.head()
```

Out[34]:

	Target	0	1	2	3	4	5	6	7	8	...	127	128	129	130	131	132	133	134	135	136
0	49668.179688	0.0	74.690002	3.711	0.000000	0.0	0.00	0.000000	0.0	208.438690	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	3773.419922	0.0	68.120003	3.014	0.000000	0.0	0.00	0.000000	0.0	126.381546	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	10116.959961	0.0	43.150002	3.312	0.000000	0.0	0.00	0.000000	0.0	127.300934	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	12896.700195	0.0	59.669998	3.808	3847.570068	0.0	92.93	1216.189941	2403.0	131.098328	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	26548.849609	0.0	69.709999	2.792	0.000000	0.0	0.00	0.000000	0.0	132.598389	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 138 columns

In [35]:
```
val_data = pd.DataFrame({'Target':y_val[:,0]})
for i in range(X_val.shape[1]):
    val_data[i] = X_val[:,i]
```

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

Out[36]:

	Target	0	1	2	3	4	5	6	7	8	...	127	128	129	130	131	132	133	134	135	13
0	3315.000000	0.0	64.500000	3.985	6855.080078	0.0	3.66	2468.469971	2737.330078	142.851685	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	67459.343750	0.0	36.000000	2.910	0.000000	0.0	0.00	0.000000	0.000000	135.519516	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	26638.460938	0.0	85.800003	3.699	0.000000	0.0	0.00	0.000000	0.000000	218.878479	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
3	2465.290039	0.0	47.889999	4.000	1134.329956	0.0	2.15	625.260010	1894.979980	138.833618	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	5361.799805	0.0	70.410004	3.735	1291.329956	201.0	0.27	696.469971	1682.130005	197.738937	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

5 rows × 138 columns

In [37]:
```
val_data.shape
```

Out[37]:
```
(31618, 138)
```

In [38]:
```
train_data.to_csv('train.csv', header = False, index = False)
val_data.to_csv('validation.csv', header = False, index = False)
```

In [39]:
```
import sagemaker
import boto3

sagemaker_session = sagemaker.Session()

bucket = 'retail-sales-predictions'
prefix = 'XGBoostregressor'
key = 'XGBoostregressor'


role = sagemaker.get_execution_role()
```

In [40]:
```
print(role)
```

```
arn:aws:iam::743472474152:role/service-role/AmazonSageMaker-ExecutionRole-20200531T075337
```

In [41]:
```
import os
with open('train.csv','rb') as f:

    boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train', key)).upload_fileobj(f)


s3_train_data = 's3://{}/{}/train/{}'.format(bucket, prefix, key)
print('uploaded training data location: {}'.format(s3_train_data))
```

```
uploaded training data location: s3://retail-sales-predictions/XGBoostregressor/train/XGBoostregressor
```

In [42]:
```
with open('validation.csv','rb') as f:
    boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'validation', key)).upload_fileobj(f)

s3_validation_data = 's3://{}/{}/validation/{}'.format(bucket, prefix, key)
print('uploaded validation data location: {}'.format(s3_validation_data))
```

```
uploaded validation data location: s3://retail-sales-predictions/XGBoostregressor/validation/XGBoostregressor
```

In [43]:
```
output_location = 's3://{}/{}/output'.format(bucket, prefix)
print('training artifacts will be uploaded to: {}'.format(output_location))
```

```
training artifacts will be uploaded to: s3://retail-sales-predictions/XGBoostregressor/output
```

In [44]:
```
from sagemaker.amazon.amazon_estimator import get_image_uri

container = get_image_uri(boto3.Session().region_name, 'xgboost','0.90-2')
```

In [45]:
```
Xgboost_regressor1 = sagemaker.estimator.Estimator(container,
                                    role,
                                    train_instance_count = 1,
                                    train_instance_type = 'ml.m5.2xlarge',
                                    output_path = output_location,
                                    sagemaker_session = sagemaker_session)

Xgboost_regressor1.set_hyperparameters(max_depth = 10,
                           objective = 'reg:linear',
                           colsample_bytree = 0.3,
                           alpha = 10,
                           eta = 0.1,
                           num_round = 100
                           )
```

In [46]:
```
train_input = sagemaker.session.s3_input(s3_data = s3_train_data, content_type='csv',s3_data_type = 'S3Prefix')
valid_input = sagemaker.session.s3_input(s3_data = s3_validation_data, content_type='csv',s3_data_type = 'S3Prefix')


data_channels = {'train': train_input,'validation': valid_input}


Xgboost_regressor1.fit(data_channels)
```

```
INFO:sagemaker-containers:Failed to parse hyperparameter objective value reg:linear to Json.
Returning the value itself
INFO:sagemaker-containers:No GPUs detected (normal if no gpus installed)
INFO:sagemaker_xgboost_container.training:Running XGBoost Sagemaker in algorithm mode
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
[17:59:54] 358334x137 matrix with 49091758 entries loaded from /opt/ml/input/data/train?format=csv&label_column=0&delimiter=,
INFO:root:Determined delimiter of CSV input is ','
[17:59:54] 31618x137 matrix with 4331666 entries loaded from /opt/ml/input/data/validation?format=csv&label_column=0&delimiter=,
INFO:root:Single node training.
```

```
INFO:root:Train matrix has 358334 rows
INFO:root:Validation matrix has 31618 rows
[17:59:54] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:sq
uarederror.
[0]#011train-rmse:26723.9#011validation-rmse:27269.2
[1]#011train-rmse:25589.3#011validation-rmse:26118
```

DEPLOY THE MODEL TO MAKE PREDICTIONS

In [47]:
```
Xgboost_regressor = Xgboost_regressor1.deploy(initial_instance_count = 1, instance_type = 'ml.t2.medium')
```

```
----------------!
```

In [48]:
```
from sagemaker.predictor import csv_serializer, json_deserializer

Xgboost_regressor.content_type = 'text/csv'
Xgboost_regressor.serializer = csv_serializer
Xgboost_regressor.deserializer = None
```

In [49]:
```
X_test.shape
```

Out[49]:
```
(31618, 137)
```

In [50]:
```
predictions1 = Xgboost_regressor.predict(X_test[0:10000])
```

In [51]:
```
predictions2 = Xgboost_regressor.predict(X_test[10000:20000])
```

In [52]:
```
predictions3 = Xgboost_regressor.predict(X_test[20000:30000])
```

In [53]:
```
predictions4 = Xgboost_regressor.predict(X_test[30000:31618])
```

In [54]:
```
predictions4
```

Out[54]:
```
b'48527.07421875,7029.36328125,2466.084716796875,16457.404296875,25649.703125,44692.03125,4833.6787109375,27276.082
03125,14986.9921875,1913.6829833984375,-190.27484130859375,4932.23046875,-1159.119140625,35972.171875,2363.79150390
625,17072.529296875,2185.8603515625,1070.990478515625,6698.3037109375,4066.58056640625,20706.38671875,16184.4042968
75,14929.1005859375,44982.16796875,11947.6953125,6095.2138671875,13043.431640625,91403.8515625,13560.9580078125,-33
1.18597412109375,14530.71875,10597.267578125,6297.89208984375,5665.7236328125,1159.1861572265625,4754.62646484375,1
8988.947265625,95525.6328125,35534.23046875,1538.0316162109375,13964.4501953125,11310.3720703125,6405.509765625,341
00.1484375,4367.82958984375,17228.876953125,17213.6015625,932.7844848632812,2689.062744140625,4341.47802734375,1075
3.0322265625,9570.87109375,5649.17724609375,11516.2294921875,-2778.6865234375,27442.359375,41873.52734375,8686.5546
875,9195.697265625,19779.541015625,2091.699951171875,5595.443359375,30179.404296875,11557.673828125,21304.24609375,
16223.4921875,2975.59326171875,11046.1767578125,8065.96484375,6403.42724609375,11606.66015625,7947.033203125,11389.
814453125,9267.62890625,2191.492431640625,10754.5185546875,46607.14453125,1632.7021484375,10669.6435546875,9271.912
109375,7683.94384765625,15280.3720703125,5475.21337890625,5680.95849609375,64348.92578125,63313.3984375,4109.868652
34375,9271.912109375,46334.02734375,59772.671875,11199.6435546875,1727.33154296875,9234.203125,26138.154296875,1633
8.083984375,11045.201171875,1010.9727783203125,11086.345703125,1049.1083984375,11912.1484375,18541.447265625,1233.5
079345703125,7587.33935546875,5158.35693359375,3123.771484375,23286.20703125,8320.857421875,16835.546875,8281.28613
28125,1804.28564453125,14381.9716796875,16940.15234375,25820.43359375,22290.234375,2971.97900390625,15867.625,1138
3.8046875,1622.7041015625,11516.2294921875,2223.298095703125,13410.083984375,8610.140625,5825.044921875,11943.10253
90625,16238.5458984375,9829.7109375,46452.01171875,10986.2138671875,12122.2138671875,5554.91064453125,4542.1914062
5,22794.890625,6369.35888671875,10065.3310546875,18538.916015625,8887.3994140625,2207.68115234375,30101.6875,13243.
```

In [55]:
```
def bytes_2_array(x):
    l = str(x).split(',')
    l[0] = l[0][2:]
    l[-1] = l[-1][:-1]
    for i in range(len(l)):
        l[i] = float(l[i])
    l = np.array(l).astype('float32')
    return l.reshape(-1,1)
```

In [56]:
```
predicted_values_1 = bytes_2_array(predictions1)
```

In [57]:
```
predicted_values_1.shape
```

Out[57]:
```
(10000, 1)
```

In [58]:
```
predicted_values_2 = bytes_2_array(predictions2)
predicted_values_2.shape
```

Out[58]:
```
(10000, 1)
```

In [59]:
```
predicted_values_3 = bytes_2_array(predictions3)
predicted_values_3.shape
```

Out[59]:
```
(10000, 1)
```

In [60]:
```
predicted_values_4 = bytes_2_array(predictions4)
predicted_values_4.shape
```

Out[60]:
```
(1618, 1)
```

In [61]:
```
predicted_values = np.concatenate((predicted_values_1, predicted_values_2, predicted_values_3, predicted_values_4))
```

In [62]:
```
predicted_values.shape
```

Out[62]:
```
(31618, 1)
```

In [63]:
```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from math import sqrt
k = X_test.shape[1]
n = len(X_test)
RMSE = float(format(np.sqrt(mean_squared_error(y_test, predicted_values)),'.3f'))
MSE = mean_squared_error(y_test, predicted_values)
MAE = mean_absolute_error(y_test, predicted_values)
r2 = r2_score(y_test, predicted_values)
adj_r2 = 1-(1-r2)*(n-1)/(n-k-1)

print('RMSE =',RMSE, '\nMSE =',MSE, '\nMAE =',MAE, '\nR2 =', r2, '\nAdjusted R2 =', adj_r2)
```

```
RMSE = 7403.337
MSE = 54809396.0
```

```
MSE = 34809390.0
MAE = 4348.4414
R2 = 0.8934328644388627
Adjusted R2 = 0.8929690875147244
```

In [64]:
```
Xgboost_regressor.delete_endpoint()
```

TRAIN THE MODEL WITH BEST PARAMETERS

(After running the tuning job in AWS Sagemake)

In [65]:
```
Xgboost_regressor = sagemaker.estimator.Estimator(container,
                                    role,
                                    train_instance_count=1,
                                    train_instance_type='ml.m4.xlarge',
                                    output_path=output_location,
                                    sagemaker_session=sagemaker_session)


Xgboost_regressor.set_hyperparameters(max_depth=25,
                            objective='reg:linear',
                            colsample_bytree = 0.3913546819101119,
                            alpha = 1.0994354985124635,
                            eta = 0.23848185159806115,
                            num_round = 237
                            )
```

In [66]:
```
train_input = sagemaker.session.s3_input(s3_data = s3_train_data, content_type='csv',s3_data_type = 'S3Prefix')
valid_input = sagemaker.session.s3_input(s3_data = s3_validation_data, content_type='csv',s3_data_type = 'S3Prefix')
data_channels = {'train': train_input,'validation': valid_input}
Xgboost_regressor.fit(data_channels)
```

```
[120]#011train-rmse:1779.3#011validation-rmse:5333.46
[121]#011train-rmse:1755.02#011validation-rmse:5331.17
[122]#011train-rmse:1748.79#011validation-rmse:5330.86
[123]#011train-rmse:1748.04#011validation-rmse:5330.4
[124]#011train-rmse:1735.48#011validation-rmse:5326.18
[125]#011train-rmse:1730.56#011validation-rmse:5324.96
[126]#011train-rmse:1729.54#011validation-rmse:5324.59
[127]#011train-rmse:1718.21#011validation-rmse:5322.11
[128]#011train-rmse:1712.99#011validation-rmse:5320.62
[129]#011train-rmse:1711.25#011validation-rmse:5319.62
[130]#011train-rmse:1708.98#011validation-rmse:5318.87
[131]#011train-rmse:1705.61#011validation-rmse:5318.4
[132]#011train-rmse:1700.91#011validation-rmse:5316.36
[133]#011train-rmse:1693.19#011validation-rmse:5315.2
[134]#011train-rmse:1692.02#011validation-rmse:5314.92
[135]#011train-rmse:1674.72#011validation-rmse:5305.6
[136]#011train-rmse:1671.23#011validation-rmse:5304.21
[137]#011train-rmse:1668.71#011validation-rmse:5303.39
[138]#011train-rmse:1653.38#011validation-rmse:5301.18
[139]#011train-rmse:1642.56#011validation-rmse:5296.42
```

In [67]:
```
Xgboost_regressor = Xgboost_regressor.deploy(initial_instance_count = 1,
                                             instance_type = 'ml.m4.xlarge')
```

```
---------------!
```

In [68]:
```
from sagemaker.predictor import csv_serializer, json_deserializer

Xgboost_regressor.content_type = 'text/csv'
Xgboost_regressor.serializer = csv_serializer
Xgboost_regressor.deserializer = None
```

In [71]:
```
predictions1 = Xgboost_regressor.predict(X_test[0:10000])
```

In [72]:
```
predicted_values_1 = bytes_2_array(predictions1)
predicted_values_1.shape
```

Out[72]:
```
(10000, 1)
```

In [73]:
```
predictions2 = Xgboost_regressor.predict(X_test[10000:20000])
predicted_values_2 = bytes_2_array(predictions2)
predicted_values_2.shape
```

Out[73]:
```
(10000, 1)
```

In [74]:
```
predictions3 = Xgboost_regressor.predict(X_test[20000:30000])
predicted_values_3 = bytes_2_array(predictions3)
predicted_values_3.shape
```

Out[74]:
```
(10000, 1)
```

In [75]:
```
predictions4 = Xgboost_regressor.predict(X_test[30000:31618])
predicted_values_4 = bytes_2_array(predictions4)
predicted_values_4.shape
```

Out[75]:
```
(1618, 1)
```

In [76]:
```
predicted_values = np.concatenate((predicted_values_1, predicted_values_2, predicted_values_3, predicted_values_4))
```

In [77]:
```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from math import sqrt
k = X_test.shape[1]
n = len(X_test)
RMSE = float(format(np.sqrt(mean_squared_error(y_test, predicted_values)),'.3f'))
MSE = mean_squared_error(y_test, predicted_values)
MAE = mean_absolute_error(y_test, predicted_values)
r2 = r2_score(y_test, predicted_values)
adj_r2 = 1-(1-r2)*(n-1)/(n-k-1)

print('RMSE =',RMSE, '\nMSE =',MSE, '\nMAE =',MAE, '\nR2 =', r2, '\nAdjusted R2 =', adj_r2)
```

```
RMSE = 4332.043
MSE = 18766602.0
MAE = 1892.6307
```

```
R2 = 0.9635116807218502
Adjusted R2 = 0.9633528846690832
```

In [78]:
```
Xgboost_regressor.delete_endpoint()
```

In [ ]: