

```
In [0]: import pandas as pd
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.model_selection import train_test_split
```

```
In [2]: from google.colab import drive
drive.mount('/content/drive')
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Aauth%3A2.0%3Aoauth2%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:
.....
Mounted at /content/drive

```
In [0]: feature = pd.read_csv("/content/drive/My Drive/walmart/features.csv", parse_dates=True)
stores = pd.read_csv("/content/drive/My Drive/walmart/stores.csv")
train = pd.read_csv("/content/drive/My Drive/walmart/train.csv", parse_dates=True)
test = pd.read_csv("/content/drive/My Drive/walmart/test.csv", parse_dates=True)
sample = pd.read_csv("/content/drive/My Drive/walmart/sampleSubmission.csv")
```

```
In [0]: #Merging information between the data [Train and Test]
dfTrainTmp      = pd.merge(train, stores)
dfTestTmp       = pd.merge(test, stores)

#Merging the feature with the data [Train and Test]
train           = pd.merge(dfTrainTmp, feature)
test            = pd.merge(dfTestTmp, feature)
```

```
In [0]: #Split the field Date
train['Year']    = pd.to_datetime(train['Date']).dt.year
train['Month']   = pd.to_datetime(train['Date']).dt.month
train['Day']     = pd.to_datetime(train['Date']).dt.day
train['Days']    = train['Month']*30+train['Day']

#Converting type of store to numeric
train['Type']    = train['Type'].replace('A',1)
train['Type']    = train['Type'].replace('B',2)
train['Type']    = train['Type'].replace('C',3)

#Counting the passend days util the holiday
train['daysHoliday'] = train['IsHoliday']*train['Days']
#Covertng the sales to log scale
train['logSales']    = np.log(4990+train['Weekly_Sales'])

#Same with test data
test['Year']        = pd.to_datetime(test['Date']).dt.year
test['Month']       = pd.to_datetime(test['Date']).dt.month
test['Day']         = pd.to_datetime(test['Date']).dt.day
test['Days']        = test['Month']*30+test['Day']
test['Type']        = test['Type'].replace('A',1)
test['Type']        = test['Type'].replace('B',2)
test['Type']        = test['Type'].replace('C',3)
test['daysHoliday'] = test['IsHoliday']*test['Days']
```

```
In [6]: feature.groupby(["Store"]).head()
```

Out[6]:

Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4
-------	------	-------------	------------	-----------	-----------	-----------	-----------

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4
0	1	2010-02-05	42.31	2.572	NaN	NaN	NaN	NaN
1	1	2010-02-12	38.51	2.548	NaN	NaN	NaN	NaN
2	1	2010-02-19	39.93	2.514	NaN	NaN	NaN	NaN
3	1	2010-02-26	46.63	2.561	NaN	NaN	NaN	NaN
4	1	2010-03-05	46.50	2.625	NaN	NaN	NaN	NaN
...
8008	45	2010-02-05	27.31	2.784	NaN	NaN	NaN	NaN
8009	45	2010-02-12	27.73	2.773	NaN	NaN	NaN	NaN
8010	45	2010-02-19	31.27	2.745	NaN	NaN	NaN	NaN
8011	45	2010-02-26	34.89	2.754	NaN	NaN	NaN	NaN
8012	45	2010-03-05	37.13	2.777	NaN	NaN	NaN	NaN

225 rows × 12 columns



```
In [7]: #https://www.kaggle.com/yepp2411/walmart-prediction-1-eda-with-time-and-space
grouped=stores.groupby('Type')
print(grouped.describe()['Size'].round(2))
plt.style.use('ggplot')
labels=['A store', 'B store', 'C store']
sizes=grouped.describe()['Size'].round(1)
sizes=[(22/(17+6+22))*100, (17/(17+6+22))*100, (6/(17+6+22))*100] # conve
```

rt to the proportion

```
fig, axes = plt.subplots(1,1, figsize=(10,10))

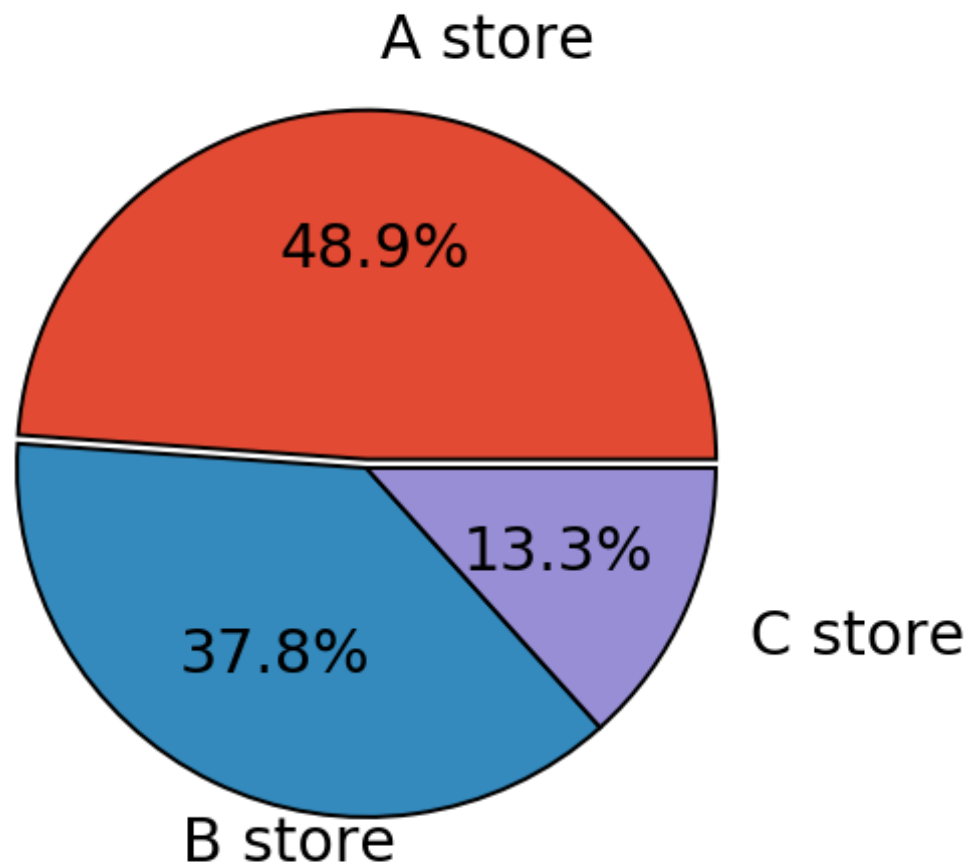
wprops={'edgecolor':'black',
        'linewidth':2}

tprops = {'fontsize':30}

axes.pie(sizes,
        labels=labels,
        explode=(0.02,0,0),
        autopct='%1.1f%%',
        pctdistance=0.6,
        labeldistance=1.2,
        wedgeprops=wprops,
        textprops=tprops,
        radius=0.8,
        center=(0.5,0.5))
plt.show()
```

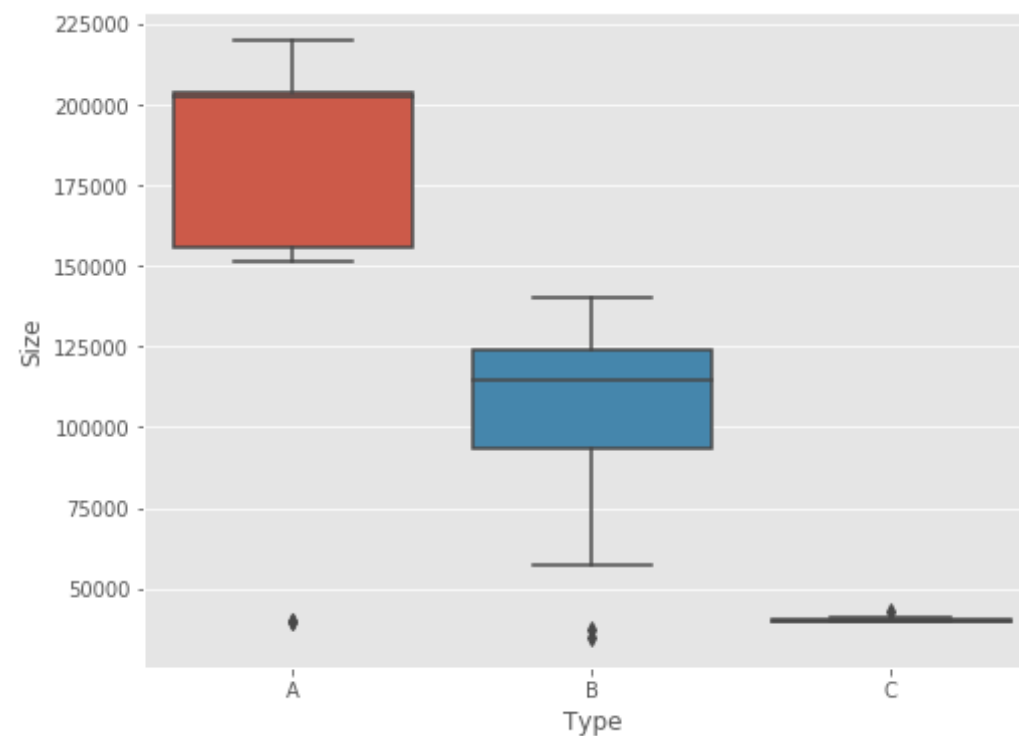
	count	mean	std	...	50%	75%	max
Type				...			
A	22.0	177247.73	49392.62	...	202406.0	203819.0	219622.0
B	17.0	101190.71	32371.14	...	114533.0	123737.0	140167.0
C	6.0	40541.67	1304.15	...	39910.0	40774.0	42988.0

[3 rows x 8 columns]



```
In [8]: data = pd.concat([stores['Type'], stores['Size']], axis=1)
        f, ax = plt.subplots(figsize=(8, 6))
```

```
fig = sns.boxplot(x='Type', y='Size', data=data)
```



```
In [0]: #Dropping markdown data as most of it is null
train = train.drop(['CPI', 'Unemployment', 'Date',
                    'Markdown1', 'Markdown2', 'Markdown3',
                    'Markdown4', 'Markdown5', 'Weekly_Sales'], axis=1)

test = test.drop(['CPI', 'Unemployment', 'Date',
                  'Markdown1', 'Markdown2', 'Markdown3',
                  'Markdown4', 'Markdown5'], axis=1)
```

Removing features :CPI,Unemployment,Date,Markdown1,Markdown2,Markdown3,

MarkDown4,MarkDown5,Weekly_Sales

In [10]: `train.head()`

Out[10]:

	Store	Dept	IsHoliday	Type	Size	Temperature	Fuel_Price	Year	Month	Day	Days	days
0	1	1	False	1	151315	42.31	2.572	2010	2	5	65	
1	1	2	False	1	151315	42.31	2.572	2010	2	5	65	
2	1	3	False	1	151315	42.31	2.572	2010	2	5	65	
3	1	4	False	1	151315	42.31	2.572	2010	2	5	65	
4	1	5	False	1	151315	42.31	2.572	2010	2	5	65	

In [11]: [#https://github.com/leandromferreira/Kaggle-Walmart-Recruiting-Store-Sales-Forecasting](https://github.com/leandromferreira/Kaggle-Walmart-Recruiting-Store-Sales-Forecasting)

```
etr = ExtraTreesRegressor(n_estimators=1000, verbose=0, n_jobs=4)

#The file which contains sales for test data
result = open('/content/drive/My Drive/walmart/Untitled Folder/result_etr.csv', 'w')
result.write('Id,Weekly_Sales\n')
size = sample['Id'].count() #test size
i=0;

#Here for every row in sample file i.e for every dept store data we will fit our model separately and predict sales.
while (i < size):

    Id          = sample['Id'][i]
    tmpStr      = Id.split('_')
    Store       = int(tmpStr[0])
    #Store ID
    Dept        = int(tmpStr[1])
    #Dept ID
    dataF1      = train.loc[train['Dept']==Dept]
```

```

        #Get the data from Dept ID from all data
        tmpDf = dataF1.loc[dataF1['Store']==Store]
        #Get the data form Store ID from the filtering data
        store_cnt = tmpDf['Store'].count()
        dept_cnt = dataF1['Dept'].count()
        tmpF = dataF1.loc[train['IsHoliday']==1]
        dataF1 = pd.concat([dataF1,tmpF*4])
        #Reforcing holiday data
        dataF2 = dataF1.loc[dataF1['Store']==Store]
        #Filtering
        testF1 = test.loc[test['Dept']==Dept]

        testF1 = testF1.loc[testF1['Store']==Store]
        testRows = testF1['Store'].count()
        k = i + testRows

        if (store_cnt < 10) and (dept_cnt!=0): #When the number of dataframes stores is too small RF fails then we will only work with department data
            y=np.asarray(dataF1['logSales'], dtype="|S6")
            X_train, X_test, y_train, y_test = train_test_split(dataF1.drop(['logSales'],axis=1),y)

            trained_model = etr.fit(dataF1.drop(['logSales'],axis=1),np.asarray(dataF1['logSales'],dtype=float))
        else:
            y=np.asarray(dataF2['logSales'], dtype="|S6")
            X_train, X_test, y_train, y_test = train_test_split(dataF2.drop(['logSales'],axis=1),y)
            trained_model = etr.fit(dataF2.drop(['logSales'],axis=1),np.asarray(dataF2['logSales'],dtype=float))

        tmpP_RF_Submiss = ( np.exp(pd.to_numeric(trained_model.predict(testF1))) - 4990 )

        for j in range(i,k):
            result.write('%s,%s\n'%(sample['Id'][j],tmpP_RF_Submiss[j-i]))

```



```
        i+=testRows  
        print (i)  
result.close()
```

39
78
117
156
195
234
273
312
351
390
429
468
507
546
585
624
656
695
734
773
812
851
890
929
968
1007
1046
1085
1124
1163
1202
1241
1280
1319
1358
1397

1436
1475
1514
1553
1592
1599
1638
1642

1680
1719
1758
1782
1821
1860
1899
1938
1977
2016
2055
2094
2133
2172
2211
2250
2289
2328
2367
2406
2445
2484
2523
2562
2601
2640
2679
2718
2757
2783
2822

2861
2900
2939
2978
3017
3056
3095
3134

3173
3212
3251
3290
3329
3368
3407
3442
3481
3520
3559
3598
3637
3676
3715
3754
3793
3832
3871
3910
3949
3988
4027
4066
4105
4144
4183
4222
4261
4300
4339

4378
4382
4421
4424
4463
4502
4541
4572

4611
4650
4689
4728
4767
4806
4845
4884
4923
4924
4963
5002
5041
5080
5119
5158
5197
5236
5275
5314
5353
5392
5431
5470
5509
5548
5580
5619
5658
5697
5736

5775
5814
5853
5892
5931
5970
6009
6048

6087
6126
6165
6204
6234
6273
6312
6351
6390
6429
6468
6507
6546
6585
6624
6663
6702
6741
6780
6819
6858
6897
6935
6974
7013
7052
7091
7130
7169
7174
7201

7240
7259
7298
7337
7360
7399
7438
7477

7516
7555
7594
7629
7668
7707
7746
7785
7824
7863
7902
7923
7962
8001
8040
8053
8092
8131
8170
8209
8248
8287
8326
8365
8404
8443
8482
8521
8560
8599
8638

8677
8715
8754
8793
8832
8871
8910
8949

8988
9027
9066
9105
9144
9183
9222
9261
9300
9339
9378
9417
9456
9495
9496
9535
9574
9613
9652
9658
9697
9701
9740
9779
9818
9853
9892
9931
9970
10009
10048

10087
10126
10165
10204
10243
10282
10321
10360

10399
10438
10477
10516
10555
10594
10633
10672
10711
10750
10789
10828
10856
10895
10934
10973
11012
11051
11090
11129
11168
11207
11246
11285
11324
11363
11402
11441
11480
11508
11528

11567
11606
11645
11684
11723
11762
11801
11840

11879
11918
11957
11996
12035
12074
12113
12152
12191
12230
12269
12308
12347
12386
12390
12429
12430
12463
12502
12521
12560
12599
12631
12644
12683
12722
12761
12800
12839
12878
12891

12930
12969
13008
13047
13086
13125
13164
13178

13217
13256
13295
13300
13303
13342
13381
13420
13459
13498
13537
13576
13615
13654
13693
13732
13771
13810
13849
13888
13927
13957
13996
14035
14074
14113
14152
14191
14230
14269
14308

14347
14386
14425
14464
14503
14542
14581
14620

14659
14698
14737
14776
14815
14854
14893
14896
14935
14936
14975
15014
15053
15087
15126
15165
15204
15243
15282
15321
15360
15399
15438
15477
15516
15555
15594
15633
15672
15711
15750

15789
15828
15867
15906
15945
15984
16023
16062

16091
16130
16169
16208
16247
16286
16325
16364
16403
16442
16481
16520
16559
16598
16637
16676
16715
16743
16758
16797
16836
16875
16914
16953
16992
17031
17070
17109
17148
17187
17226

17265
17304
17343
17382
17421
17460
17499
17538

17577
17616
17631
17670
17675
17712
17742
17781
17804
17843
17882
17919
17941
17980
18019
18058
18097
18136
18175
18214
18253
18292
18331
18370
18409
18448
18487
18526
18565
18604
18643

18682
18721
18760
18799
18838
18877
18916
18955

18994
19033
19072
19111
19150
19189
19228
19267
19306
19345
19384
19418
19457
19496
19535
19574
19613
19652
19691
19730
19769
19808
19847
19886
19925
19964
20003
20042
20081
20120
20159

20198
20237
20276
20315
20354
20357
20396
20397

20436
20475
20502
20541
20580
20615
20654
20693
20732
20771
20810
20849
20888
20927
20966
21005
21044
21083
21122
21161
21200
21239
21278
21317
21356
21357
21396
21435
21459
21498
21537

21576
21615
21654
21693
21732
21771
21810
21849

21888
21927
21966
22005
22044
22083
22113
22152
22191
22230
22269
22308
22347
22386
22425
22464
22503
22542
22581
22620
22659
22698
22737
22776
22815
22854
22893
22932
22971
23010
23021

23060
23063
23097
23124
23163
23183
23222
23261

23284
23323
23362
23401
23440
23479
23518
23522
23561
23600
23635
23674
23713
23752
23791
23807
23846
23885
23890
23894
23933
23972
24011
24050
24089
24128
24167
24206
24245
24284
24323

24362
24401
24440
24479
24518
24547
24586
24625

24664
24703
24742
24781
24820
24859
24898
24937
24976
25015
25054
25093
25132
25171
25210
25249
25288
25327
25366
25405
25444
25483
25484
25523
25532
25571
25610
25649
25688
25725
25764

25803
25842
25881
25920
25959
25998
26037
26076

26115
26154
26193
26232
26271
26310
26349
26388
26427
26466
26505
26520
26559
26598
26637
26670
26676
26715
26754
26793
26832
26871
26910
26949
26988
27027
27066
27105
27144
27183
27222

27261
27300
27333
27372
27411
27450
27489
27528

27567
27606
27645
27684
27723
27762
27801
27840
27879
27918
27957
27996
28035
28074
28113
28152
28191
28230
28269
28272
28311
28315
28354
28393
28428
28467
28506
28544
28583
28622
28661

28700
28739
28778
28817
28856
28895
28934
28973

29012
29051
29090
29129
29168
29207
29246
29285
29324
29363
29402
29432
29471
29510
29549
29588
29627
29666
29705
29744
29783
29822
29861
29900
29939
29978
30017
30056
30091
30130
30169

30208
30247
30286
30325
30364
30403
30442
30481

30520
30559
30598
30637
30676
30715
30754
30793
30832
30871
30910
30949
30988
31027
31029
31056
31095
31128
31167
31204
31243
31282
31321
31360
31399
31438
31477
31516
31555
31594
31633

31672
31711
31750
31789
31828
31867
31906
31924

31963
31971
32010
32049
32055
32094
32133
32172
32211
32250
32289
32328
32367
32406
32445
32484
32523
32562
32601
32640
32679
32716
32755
32794
32833
32872
32911
32950
32989
33028
33067

33106
33145
33184
33223
33262
33301
33340
33379

33418
33457
33496
33535
33574
33613
33652
33660
33699
33700
33738
33777
33816
33855
33886
33925
33964
34003
34042
34081
34120
34159
34198
34237
34276
34315
34354
34393
34432
34471
34510

34549
34588
34627
34666
34705
34744
34783
34822

34861
34891
34930
34969
35008
35047
35086
35125
35164
35203
35242
35281
35320
35359
35398
35437
35476
35515
35545
35584
35623
35662
35701
35740
35779
35818
35857
35896
35935
35974
36013

36052
36091
36130
36169
36208
36247
36286
36325

36364
36403
36404
36443
36482
36485
36524
36563
36566
36605
36632
36671
36710
36749
36788
36827
36866
36905
36944
36983
37022
37061
37100
37139
37178
37217
37256
37295
37334
37373
37412

37451
37490
37492
37531
37570
37597
37636
37675

37714
37753
37792
37831
37870
37909
37948
37987
38026
38065
38104
38143
38182
38221
38251
38290
38329
38368
38407
38446
38485
38524
38563
38602
38641
38680
38719
38758
38797
38836
38875

38914
38953
38954
38993
39032
39071
39110
39149

39188
39191
39230
39269
39308
39335
39374
39413
39451
39490
39529
39568
39607
39646
39685
39724
39763
39802
39841
39880
39919
39958
39997
40036
40075
40114
40136
40175
40214
40253
40292

40295
40334
40373
40412
40451
40490
40529
40568

40607
40646
40685
40724
40763
40802
40841
40880
40919
40945
40984
41023
41062
41101
41140
41179
41218
41257
41296
41335
41374
41413
41452
41491
41530
41569
41607
41646
41685
41724
41763

41802
41841
41856
41895
41897
41934
41943
41982

42002
42041
42080
42108
42123
42162
42201
42240
42279
42318
42357
42396
42435
42474
42513
42552
42591
42630
42669
42708
42747
42756
42795
42834
42873
42901
42909
42948
42987
43026
43065

43104
43143
43182
43221
43260
43299
43338
43377

43416
43455
43494
43533
43566
43605
43644
43683
43722
43761
43800
43839
43878
43917
43956
43995
44034
44073
44112
44151
44190
44229
44268
44307
44346
44371
44410
44449
44460
44499
44502

44541
44580
44619
44645
44684
44723
44760
44799

44838
44877
44916
44955
44994
45033
45072
45111
45150
45189
45228
45267
45306
45345
45384
45423
45432
45471
45510
45549
45586
45591
45630
45669
45708
45747
45786
45825
45864
45903
45942

45981
46020
46059
46098
46137
46176
46215
46250

46289
46328
46367
46406
46445
46484
46523
46562
46601
46640
46679
46718
46757
46796
46835
46874
46913
46952
46991
47030
47069
47108
47109
47148
47155
47194
47196
47198
47237
47276
47315

47345
47384
47423
47461
47500
47539
47578
47617

47656
47695
47734
47773
47812
47851
47890
47929
47968
48007
48046
48085
48124
48163
48202
48206
48245
48284
48289
48328
48367
48406
48445
48484
48523
48562
48601
48640
48679
48718
48757

48796
48835
48874
48913
48947
48966
49005
49044

49083
49122
49161
49200
49239
49278
49317
49356
49395
49434
49473
49512
49551
49590
49629
49668
49707
49746
49785
49824
49863
49902
49941
49946
49985
50020
50059
50084
50123
50162
50201

50240
50279
50318
50357
50396
50435
50474
50513

50552
50591
50630
50669
50708
50747
50786
50825
50864
50903
50942
50981
51020
51059
51088
51127
51166
51205
51244
51283
51322
51361
51400
51439
51478
51517
51556
51595
51634
51673
51712

51744
51762
51801
51840
51879
51918
51957
51996

52035
52074
52113
52152
52191
52230
52269
52308
52347
52386
52425
52464
52503
52542
52581
52620
52659
52661
52700
52702
52741
52780
52819
52858
52893
52932
52971
53010
53049
53088
53127

53166
53205
53244
53245
53284
53323
53362
53401

53440
53479
53518
53557
53596
53635
53674
53713
53752
53755
53794
53833
53862
53901
53940
53979
54018
54057
54096
54135
54174
54213
54252
54291
54330
54369
54408
54447
54486
54519
54558

54597
54636
54675
54714
54753
54792
54831
54870

54909
54948
54987
55026
55065
55104
55143
55182
55221
55260
55299
55338
55377
55416
55455
55456
55495
55534
55565
55604
55643
55676
55715
55754
55793
55832
55871
55910
55949
55983
56022

56061
56100
56139
56178
56217
56256
56295
56334

56363
56402
56441
56479
56481
56520
56559
56598
56637
56676
56715
56754
56793
56832
56871
56910
56949
56988
57027
57066
57105
57143
57182
57221
57260
57299
57338
57377
57416
57455
57494

57533
57572
57611
57650
57689
57728
57767
57806

57845
57884
57923
57962
58001
58040
58079
58081
58120
58159
58189
58228
58267
58306
58345
58384
58423
58462
58501
58540
58579
58618
58657
58696
58735
58774
58813
58852
58891
58930
58969

58983
59022
59026
59065
59099
59138
59177
59216

59255
59294
59333
59372
59411
59450
59489
59528
59567
59606
59645
59684
59723
59757
59796
59835
59874
59913
59952
59991
60030
60069
60108
60147
60186
60225
60264
60303
60342
60381
60420

60459
60498
60537
60576
60615
60654
60656
60695

60701
60740
60779
60818
60846
60885
60924
60963
61002
61041
61080
61119
61158
61197
61236
61275
61314
61353
61392
61431
61470
61509
61548
61587
61626
61645
61684
61723
61762
61798
61802

61841
61880
61919
61958
61997
62036
62075
62114

62153
62192
62231
62270
62309
62348
62387
62426
62462
62500
62539
62578
62617
62656
62695
62734
62773
62812
62851
62890
62929
62968
63007
63046
63085
63124
63163
63202
63241
63280
63319

63358
63359
63398
63437
63440
63479
63515
63554

63587
63626
63665
63704
63743
63782
63821
63860
63899
63938
63939
63978
64017
64056
64095
64134
64173
64212
64251
64290
64329
64368
64407
64446
64485
64524
64563
64592
64631
64670
64709

64748
64787
64826
64865
64904
64943
64982
65021

65060
65099
65138
65177
65216
65248
65262
65301
65340
65379
65418
65457
65496
65535
65574
65613
65652
65691
65730
65769
65808
65847
65886
65925
65964
66003
66042
66081
66120
66159
66198

66203
66205
66244
66283
66322
66341
66380
66419

66455
66494
66533
66572
66611
66650
66689
66728
66767
66806
66845
66884
66923
66962
67001
67040
67079
67118
67157
67196
67235
67274
67276
67315
67354
67393
67432
67471
67510
67549
67588

67627
67666
67705
67744
67783
67822
67861
67900

67927
67966
68005
68044
68083
68122
68161
68200
68239
68278
68317
68356
68395
68434
68473
68512
68551
68590
68629
68668
68707
68746
68785
68824
68828
68867
68875
68910
68949
68988
69014

69053
69092
69131
69170
69209
69248
69287
69326

69365
69404
69443
69482
69521
69560
69599
69638
69677
69716
69755
69794
69833
69872
69911
69950
69977
70016
70055
70094
70133
70172
70211
70250
70289
70328
70367
70406
70445
70484
70523

70562
70601
70637
70676
70715
70754
70793
70832

70871
70910
70949
70988
71027
71066
71105
71144
71183
71222
71261
71300
71339
71378
71417
71456
71495
71534
71573
71612
71616
71655
71694
71733
71763
71802
71841
71880
71919
71958
71997

72036
72075
72114
72116
72155
72194
72233
72272

72311
72350
72389
72428
72467
72506
72545
72584
72623
72662
72701
72740
72768
72807
72846
72885
72924
72963
73002
73041
73080
73119
73158
73197
73236
73275
73314
73353
73392
73428
73467

73506
73545
73584
73623
73662
73701
73740
73779

73818
73857
73896
73935
73974
74013
74052
74091
74130
74169
74208
74247
74286
74325
74364
74371
74410
74414
74453
74492
74523
74562
74601
74631
74670
74709
74748
74787
74826
74865
74904

74943
74982
75021
75060
75099
75138
75177
75216

75255
75294
75333
75372
75411
75450
75489
75513
75552
75591
75630
75669
75708
75747
75786
75825
75864
75903
75942
75981
76020
76059
76098
76137
76176
76192
76231
76270
76309
76348
76387

76426
76465
76504
76543
76582
76621
76660
76699

76738
76777
76816
76855
76894
76933
76972
77011
77050
77089
77090
77091
77123
77162
77176
77215
77254
77269
77308
77347
77386
77425
77464
77503
77542
77581
77620
77659
77698
77737
77776

77815
77854
77893
77932
77945
77984
78023
78062

78067
78106
78145
78184
78223
78262
78294
78333
78372
78411
78450
78489
78528
78567
78606
78645
78665
78692
78715
78754
78766
78778
78780
78819
78820
78824
78863
78874
78911
78950
78958

78997
79036
79075
79091
79130
79135
79174
79208

79247
79286
79325
79364
79399
79438
79477
79516
79555
79594
79630
79669
79708
79747
79786
79825
79864
79903
79942
79981
80020
80021
80060
80099
80138
80177
80216
80255
80294
80333
80372

80411
80450
80489
80528
80567
80606
80645
80680

80719
80758
80797
80836
80875
80914
80953
80992
81031
81070
81109
81148
81187
81226
81265
81304
81343
81382
81421
81460
81499
81538
81577
81616
81624
81663
81666
81705
81706
81745
81775

81814
81853
81891
81930
81969
82008
82047
82086

82125
82127
82166
82205
82244
82283
82322
82361
82400
82439
82478
82517
82556
82595
82634
82673
82712
82751
82775
82814
82853
82892
82931
82970
83009
83048
83087
83126
83165
83204
83243

83282
83321
83360
83399
83431
83470
83509
83548

83587
83626
83665
83704
83743
83782
83821
83860
83899
83938
83977
84016
84055
84094
84133
84172
84211
84250
84289
84328
84367
84378
84417
84424
84463
84502
84531
84570
84609
84648
84687

84726
84765
84804
84843
84882
84921
84960
84999

85038
85077
85116
85155
85194
85233
85272
85311
85350
85389
85428
85467
85506
85532
85571
85610
85649
85688
85727
85766
85805
85844
85883
85922
85961
86000
86039
86078
86117
86144
86175

86214
86247
86252
86267
86299
86330
86354
86393

86432
86436
86475
86479
86518
86541
86547
86584
86589
86592
86631
86670
86671
86675
86714
86753
86792
86831
86870
86909
86933
86972
87011
87050
87089
87128
87167
87206
87245
87284
87285

87324
87363
87402
87441
87480
87519
87558
87597

87636
87675
87714
87753
87792
87831
87870
87909
87942
87969
88008
88047
88086
88125
88164
88203
88242
88281
88320
88359
88398
88437
88476
88515
88554
88593
88632
88671
88710
88712
88751

88790
88829
88868
88876
88915
88917
88956
88995

89034
89054
89093
89132
89171
89209
89248
89287
89326
89365
89404
89443
89482
89521
89560
89599
89638
89677
89716
89755
89794
89833
89872
89911
89950
89989
90028
90052
90091
90130
90169

90208
90247
90286
90325
90364
90403
90442
90481

90520
90559
90598
90637
90676
90714
90717
90756
90795
90834
90873
90912
90951
90990
91029
91068
91107
91146
91185
91224
91263
91302
91341
91380
91419
91458
91497
91536
91575
91614
91620

91648
91687
91715
91754
91792
91831
91868
91907

91946
91985
92024
92063
92102
92141
92180
92219
92258
92297
92336
92375
92414
92453
92492
92509
92548
92550
92589
92627
92666
92705
92744
92783
92806
92816
92855
92894
92932
92971
93010

93049
93088
93127
93166
93205
93241
93280
93319

93352
93360
93363
93378
93407
93409
93443
93464
93468
93507
93546
93585
93590
93629
93639
93646
93654
93659
93663
93702
93741
93748
93787
93826
93865
93904
93943
93982
93992
94031
94070

94109
94148
94187
94226
94265
94304
94343
94345

94384
94423
94462
94501
94540
94544
94583
94622
94661
94700
94739
94778
94817
94856
94895
94934
94958
94979
95018
95056
95085
95095
95134
95141
95159
95198
95210
95248
95279
95288
95327

95366
95405
95413
95452
95476
95515
95516
95554

95593
95632
95671
95672
95691
95730
95769
95808
95847
95886
95925
95963
96002
96041
96080
96119
96158
96197
96236
96275
96314
96353
96358
96397
96436
96475
96514
96552
96587
96626
96665

96704
96743
96782
96821
96860
96899
96938
96977

97002
97020
97059
97064
97084
97086
97125
97132
97147
97186
97201
97220
97258
97263
97264
97303
97342
97381
97392
97431
97458
97496
97498
97533
97572
97611
97650
97689
97728
97767
97806

97845
97884
97923
97954
97993
98032
98071
98110

98149
98188
98227
98266
98305
98344
98345
98384
98423
98462
98501
98540
98579
98618
98657
98696
98735
98774
98813
98852
98891
98930
98969
99008
99030
99069
99108
99147
99186
99225
99264

99303
99342
99381
99420
99459
99498
99537
99576

99615
99654
99693
99732
99771
99810
99849
99888
99890
99929
99968
100007
100040
100079
100118
100157
100196
100235
100274
100313
100352
100391
100395
100434
100473
100512
100551
100590
100629
100668
100707

100746
100785
100824
100863
100902
100941
100980
101019

101049
101088
101127
101166
101205
101244
101283
101322
101361
101400
101439
101478
101517
101556
101595
101634
101673
101710
101712
101751
101790
101829
101868
101907
101946
101985
102024
102063
102102
102141
102180

102219
102258
102297
102336
102375
102414
102453
102492

102531
102570
102609
102612
102651
102653
102691
102724
102763
102786
102825
102864
102903
102942
102981
103020
103059
103098
103137
103176
103215
103254
103293
103332
103371
103410
103449
103488
103527
103566
103605

103644
103683
103722
103761
103787
103826
103865
103904

103943
103982
104021
104060
104099
104138
104177
104216
104255
104294
104333
104372
104411
104446
104485
104524
104563
104602
104641
104680
104719
104758
104797
104836
104875
104914
104953
104992
105031
105070
105109

105148
105187
105226
105265
105304
105343
105351
105390

105391
105429
105468
105507
105538
105577
105616
105655
105694
105733
105772
105811
105850
105889
105928
105967
106006
106045
106084
106123
106162
106201
106240
106279
106318
106357
106396
106435
106474
106513
106541

106580
106619
106658
106697
106736
106739
106778
106817

106856
106895
106934
106973
107012
107051
107090
107129
107155
107173
107212
107219
107249
107255
107294
107308
107338
107377
107379
107409
107446
107447
107486
107525
107564
107565
107604
107641
107680
107682
107715

107754
107793
107832
107841
107880
107919
107958
107997

108036
108075
108113
108152
108191
108230
108269
108308
108347
108386
108425
108464
108503
108542
108581
108620
108659
108698
108713
108752
108791
108830
108869
108908
108947
108986
109025
109064
109103
109134
109168

109207
109226
109227
109266
109271
109299
109332
109349

109351
109390
109429
109468
109507
109516
109555
109557
109594
109612
109651
109690
109693
109732
109771
109810
109849
109888
109927
109966
109976
110015
110054
110093
110132
110171
110210
110249
110288
110327
110366

110405
110444
110483
110522
110561
110596
110635
110674

110713
110752
110791
110830
110869
110908
110947
110986
111009
111045
111084
111087
111119
111149
111187
111226
111257
111296
111319
111356
111381
111420
111459
111497
111498
111537
111564
111603
111604
111642
111659

111698
111737
111776
111815
111854
111893
111932
111971

112010
112048
112087
112126
112165
112204
112243
112282
112321
112360
112399
112438
112477
112516
112555
112594
112633
112672
112711
112750
112789
112828
112867
112906
112945
112984
113023
113062
113095
113134
113173

113212
113251
113290
113329
113368
113407
113446
113485

113524
113563
113602
113641
113680
113719
113758
113797
113836
113837
113876
113915
113954
113993
114003
114042
114043
114072
114111
114131
114170
114209
114245
114284
114323
114362
114401
114440
114479
114518
114557

114596
114635
114674
114713
114752
114791
114830
114869

114908
114947
114986
115025
115064

```
In [0]: sam1=pd.read_csv('/content/drive/My Drive/walmart/Untitled Folder/resul  
t_etr.csv')
```

```
In [0]: sampleSubmission.to_csv('/content/drive/My Drive/walmart/Untitled Folde  
r/result_etr',index=False)
```

```
In [15]: sampleSubmission.head()
```

Out[15]:

	Id	Weekly_Sales
0	1_1_2012-11-02	34912.929233
1	1_1_2012-11-09	21725.406629
2	1_1_2012-11-16	24255.418720
3	1_1_2012-11-23	22587.728104
4	1_1_2012-11-30	26415.242360

Observtion:

- only using ExtraTreesRegressor the score is 2746.44705

```
In [0]: 
```

```
In [0]: 
```

```
In [0]: 
```

```
In [0]: 
```

```
In [0]: 
```

```
In [0]: 
```