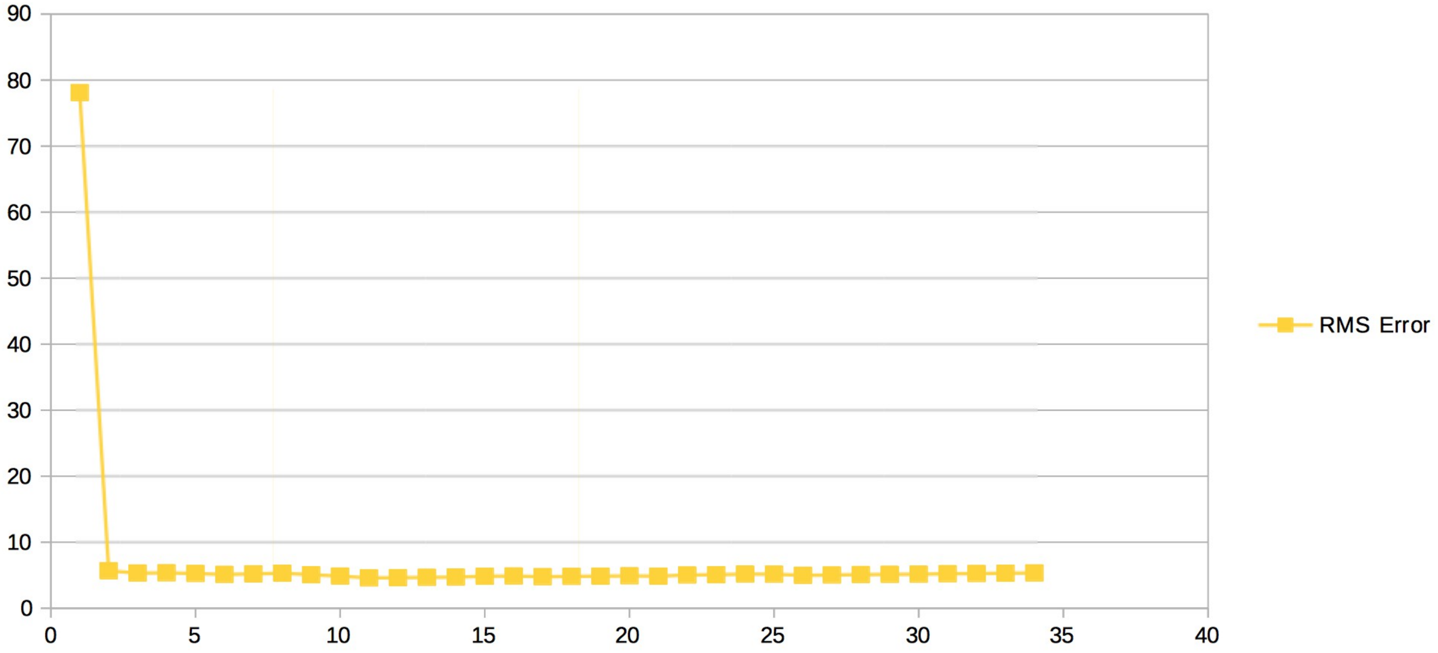
CENG 499 – Introduction to Machine Learning Spring 2017 – Homework 1
Part 4 Report

Onat Büyükakkuş – 2035772

To generate new input files according to 2016 and 2017 datas, I needed to calculate average of values. I used 5 columns: Open, High, Low, Close and Adj Close. I didn't take Volume attribute into my calculations because difference between 2016 and 2017 values of that attribute was so high.

For Analytical Solver part, first, I adapted my Analytical Solver source code to find w values from 2016 datas and calculate RMS Error from that w values and 2017 datas. After that, I created a loop, from 1 to 35, which indicates the window size. Then I created new .csv files for both 2016 and 2017 according to that window size (I uploaded two files named “part4-goog-2016.csv” and “part4-goog-2016.csv” with the minimum RMS Error.) and I ran the adapted Analytical Solver source code. For 34 different window sizes, I got the following datas and graph. (This data and graph are also visible at “part4-analyticalsolver-results.xlsx”)

Window Size	RMS Error
1	78.0968332935
2	5.6656064404
3	5.326190262
4	5.3578811863
5	5.2566112596
6	5.1174622527
7	5.2034961263
8	5.3083853047
9	5.0810821591
10	4.8668569974
11	4.5929804909
12	4.6311952219
13	4.6802517222
14	4.7110820897
15	4.8466010766
16	4.875983723
17	4.776502015
18	4.8217914659
19	4.8543057265
20	4.9096133267
21	4.8601421271
22	5.0305812308
23	5.0811131599
24	5.174624797
25	5.1547637499
26	4.9775909057
27	5.0317196479
28	5.10003532
29	5.1272149135
30	5.1510551059
31	5.221528341
32	5.2495612333
33	5.3101269581
34	5.3312910239



As it can be seen, best result i.e. minimum RMS error was (4.593) at when window size was equal to 11. So, I used 11 as window size for Iterative Solver and Weighted kNN Solver.

I ran my adapted Iterative Solver code with lambda = 0.0000001 and minimum change in RMS Error = 0.00001. I also uploaded results for this part as "part4-output-for-iterativesolver.txt". Given RMS Error was equal to 5.2039 which is very close to Analytical Solver.

Finally, I ran my adapted Weighted kNN Solver with k=5 i.e. 5 nearest neighbors for all rows of 2017 data. I got the results below. As it can be seen difference percentages are all below %10 which means that 2016 data is a good predictor for 2017 data. I also uploaded result for this solver as "part4-output-for-weightedknnsolver.txt".

Row No.	Weighted Class Label	Original Class Label	Difference Percentage (%)
1	787.4421698793	851.271997	7.4981706606
2	788.3029206298	848.714014	7.1179563874
3	788.3894510106	846.278015	6.8403719538
4	788.435690059	844.583997	6.6480429585
5	788.4170953645	845.402991	6.7406782614
6	786.5933371445	842.838	6.6732471549
7	786.4858730677	837.914002	6.1376380881
8	786.3755814349	834.838	5.8050087041
9	786.3050009604	830.229993	5.2907016622
10	783.3681253938	826.75802	5.2481976052
11	783.3487884026	829.166199	5.5257209776
12	783.3432143494	831.852002	5.831420437
13	783.3500359319	832.568994	5.9116972194
14	783.3519892625	824.154004	4.9507755273
15	783.3356300633	827.522009	5.3396016609
16	783.3584607719	827.642004	5.3505673968
17	783.3908164883	829.624011	5.5727888656
18	783.8669169025	830.414002	5.6052866384
19	783.8800898528	830.755981	5.6425583708
20	786.2873009401	825.777014	4.7821279099
21	783.886111408	822.325989	4.6745303087
22	786.2034250029	819.757983	4.093227354
23	783.8424204774	819.780005	4.3838083758
24	783.3687632973	818.185791	4.2553938342
25	783.3266779555	812.814001	3.6278069778
26	783.275180505	808.765991	3.1518153308
27	783.2353131093	807.758008	3.0358962273
28	783.2102879576	806.041992	2.8325700483
29	783.1662687187	799.86001	2.0870828736
30	783.0407370891	802.467993	2.420938415
31	782.9983342236	797.112012	1.7706015671
32	781.6096973128	796.690002	1.8928698301
33	781.6437573922	796.441992	1.8580429908
34	781.6212604421	806.988	3.143384977
35	782.216556049	828.744006	5.614212545
36	791.0506861827	833.424011	5.084245745
37	788.7813231712	832.357996	5.2353281927
38	791.7722540947	822.752197	3.7654038504
39	792.4198665031	814.095996	2.6626011678
40	792.2900522669	805.109998	1.5923222622
41	788.4694556219	804.149988	1.9499512046
42	786.4335233672	805.029004	2.3099143684
43	785.9269677129	804.761999	2.3404474007
44	783.1287755276	808.230798	3.1057988058
45	781.7188531992	805.283997	2.9263146776
46	781.7475372361	806.067993	3.0171717492
47	781.7173564848	806.015991	3.0146591118
48	783.6952248348	806.499218	2.8275282426
49	785.0618000361	801.532812	2.0549392012
50	784.9143262987	790.724011	0.7347297692
51	782.835157295	787.332007	0.5711503743
52	779.5460391767	783.304004	0.4797581532