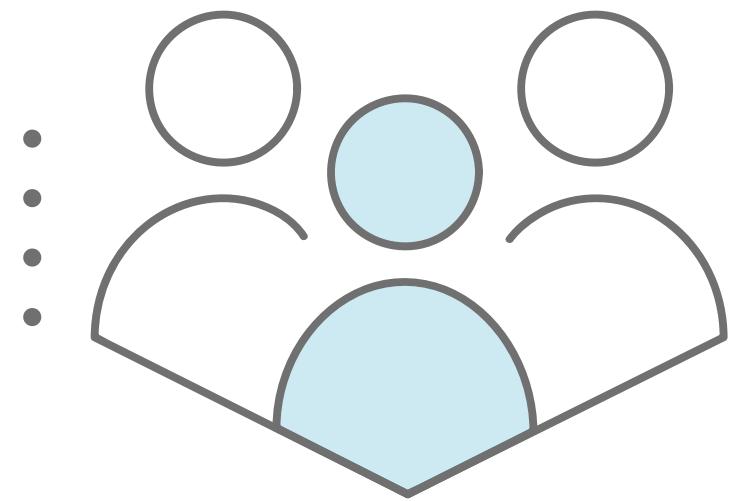
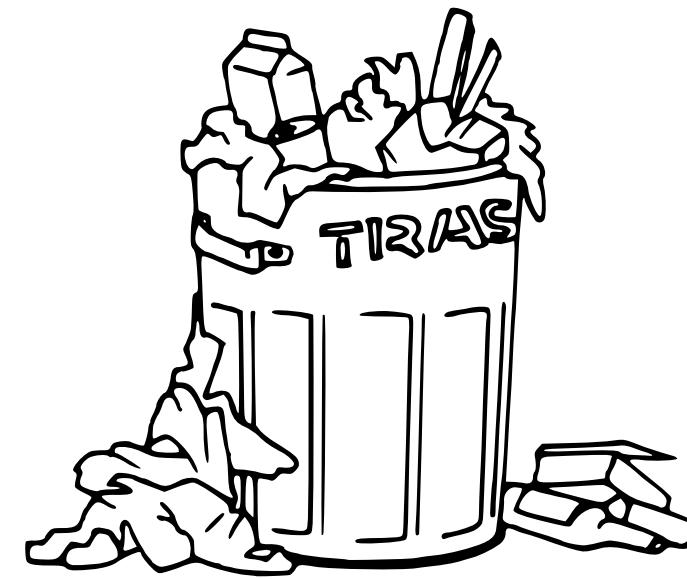


PROJECT SUKUNA





แหล่งที่มาของข้อมูล



1



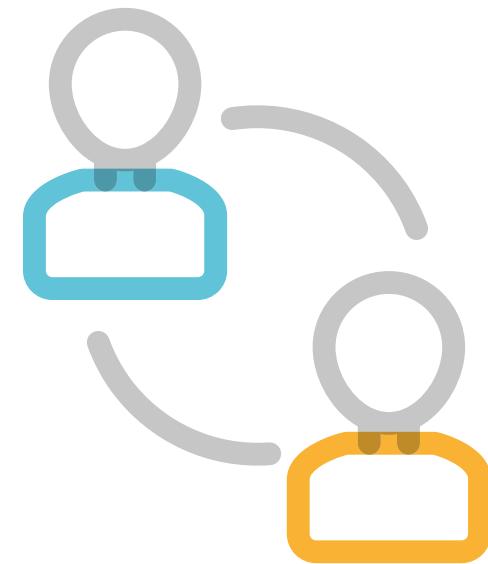
2

<http://data.bangkok.go.th/dataset/10>

<http://data.bangkok.go.th/dataset/garbage>

ข้อมูลจำนวนประชากรในเขต
กรุงเทพมหานคร

ข้อมูลปริมาณขยะมูลฝอย จำแนก
การปักครองกรุงเทพมหานคร



แหล่งที่มาของข้อมูล

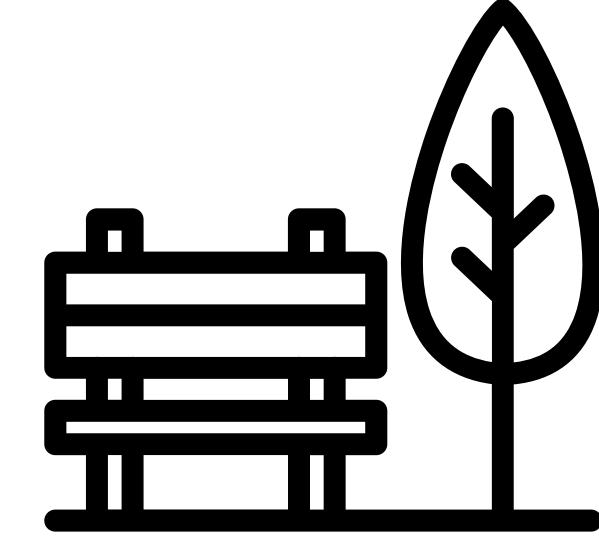
3



<http://data.bangkok.go.th/dataset/popmovement>

ข้อมูลจำนวนประชากรย้ายเข้า-ย้ายออก
ในเขตกรุงเทพมหานคร

4



<https://opendata.data.go.th/uk/dataset/district2>

ข้อมูลเบตการปกครองกรุงเทพมหานคร



วัตถุประสงค์

- กำหนดว่าปริมาณขยะที่เพิ่มขึ้น ซึ่งมีผลมาจากการจำนวนสวนสาธารณะ, จำนวนประชากร, จำนวนการย้ายเข้า และ จำนวนการย้ายออก ในเขตกรุงเทพมหานคร

ประโยชน์

- ปัจจัยที่ส่งผลการเพิ่มขึ้น-ลดลงของขยะ ได้แก่ จำนวนประชากร, การย้ายเข้า-ออก และ จำนวนสวนสาธารณะในเขตกรุงเทพมหานคร

ข้อมูลชุดที่ 1 จำนวนประชากรในกรุงเทพมหานคร



มูลชุดที่ 1 จำนวนประชากรในกรุงเทพมหานคร

	dcode	population48	population49	population50	population51	population52	population53	population54	population55	population56	population57	population58	population59	population60	population61
0	1001	67357	65835	64356	62966	61374	60313	58771	57831	56684	55373	54195	52522	0	50382
1	1002	121336	119927	117867	116742	114488	111496	108815	107969	106811	104394	103230	98450	0	94854
2	1003	126126	133415	138667	143675	147668	151292	154371	157224	159962	162598	164494	167844	0	172990
3	1004	50023	49730	49124	48506	47503	47053	46087	46112	46114	46472	46606	47308	0	48207
4	1005	178986	181390	182335	183836	185901	188164	188252	189737	190544	190659	190528	190828	0	191323
5	1006	149093	149860	150139	150286	150166	149606	148645	148491	52613	51557	149070	148392	0	146841
6	1007	63192	62102	61040	60275	58858	57368	54996	53912	50092	49280	50852	49594	0	48382
7	1008	60001	58768	57461	56464	54601	53526	52093	50930	149056	148964	48615	47450	0	45701
8	1009	98564	98096	98496	97794	96880	95661	94482	93461	92774	92448	92206	91305	0	89237
9	1010	118019	122825	127727	131035	133149	135032	136236	137295	138661	139771	140335	141214	0	142311
10	1011	138327	142460	144800	147482	152528	157477	160850	163317	165724	168309	169517	171933	0	175662
11	1012	88556	88383	88061	87386	85789	84286	82481	81529	81162	80843	80381	79574	0	78031
12	1013	31674	31142	30646	30088	29283	28617	28001	27426	26932	26359	25863	24785	0	23655
13	1014	77232	77343	77202	76477	75493	74693	73533	73084	72495	72203	72283	72102	0	70341
14	1015	136971	134589	132034	129662	126883	124499	121539	119643	117536	115330	113622	111027	0	107754
15	1016	81727	80863	79637	78307	76608	75621	73864	72241	71087	70003	69543	67887	0	67211
16	1017	76213	76402	77033	76948	76987	77292	77720	78207	78943	80002	80549	81190	0	81689
17	1018	87853	86163	84821	82824	80894	79546	77471	76353	75765	75224	74836	73871	0	72171
18	1019	105730	106811	107812	107513	106963	106753	106786	106532	106192	105857	105585	105289	0	105047
19	1020	133669	132394	130540	129401	126823	124352	120032	117950	117503	116653	115479	112581	0	110417
20	1021	132313	137934	141698	145294	150492	155821	161642	165693	169614	173144	175829	179768	0	183878
21	1022	136240	135149	134407	133622	132670	131363	130493	129800	129559	129238	128581	127582	0	125981
22	1023	128493	131344	135554	139585	142772	145361	148298	150285	151877	153175	154018	155229	0	156267
23	1024	94097	93548	92929	92094	90559	89297	87841	86643	85825	84881	84256	83248	0	81806
24	1025	108597	107139	105347	103852	102320	101276	100319	99153	98113	96787	95829	93771	0	91278

ข้อมูลชุดที่ 2 จำนวนขยะในกรุงเทพมหานคร



ข้อมูลชุดที่ 2 จำนวนขยะในเขตกรุงเทพมหานคร

[7] 1 garbage	dcode	garbage53	garbageperday53	garbage54	garbageperday54	garbage55	garbageperday55	garbage56	garbageperday56	garbage57	garbageperday57	garbage58	garbageperday58	garbage59	garbageperday59	garbage60	garbageperday60
0	1000	189819.45	520.05	122869.76	336.63	133599.00	365.02	206932.0	566.94	210940.80	577.92	217247.09	2.74	199522.73	545.14	203794.98	558.34
1	1001	69166.29	189.50	106149.08	290.82	119485.81	326.46	71844.4	196.83	69299.50	189.86	72019.02	197.31	70243.12	200.00	72998.93	0.00
2	1002	62244.26	170.53	105379.61	288.71	116.37	317.96	64255.4	176.04	62915.62	172.37	63580.71	174.19	63750.08	173.51	63331.79	0.00
3	1003	32339.16	88.60	95059.04	260.44	97084.32	265.26	40386.7	110.65	58888.59	161.34	43111.42	118.11	43545.74	131.30	47922.93	0.00
4	1004	54224.38	148.56	94655.76	259.33	100306.22	274.06	60767.8	166.49	94551.77	259.05	61746.75	169.17	58286.80	167.05	60973.16	0.00
5	1005	85742.96	234.91	87370.03	239.37	98955.18	270.37	93232.5	255.43	119561.54	327.57	99176.79	271.72	102066.16	302.24	110318.20	0.00
6	1006	101745.01	278.75	87276.00	239.11	92009.14	251.39	120589.0	330.38	102164.88	279.90	113420.03	310.74	106544.60	338.62	123596.09	0.00
7	1007	75719.25	207.45	86006.01	235.63	94435.68	258.02	104379.0	285.97	30773.10	84.31	104085.17	285.16	30383.52	309.70	113038.70	0.00
8	1008	32203.77	88.23	82532.46	226.12	99912.63	272.99	32474.1	88.97	56855.12	155.77	31207.79	85.50	117706.24	83.97	30650.17	0.00
9	1009	48978.52	134.19	78200.31	214.25	87574.21	239.27	56903.1	155.90	43346.67	118.76	54404.30	149.05	53592.83	147.54	53853.74	0.00
10	1010	60341.96	165.32	75249.90	206.16	84154.63	229.93	69510.6	190.44	70706.48	193.72	69591.76	190.66	69050.94	190.23	69434.96	0.00
11	1011	71764.46	196.61	73467.95	201.28	80198.83	219.12	88997.5	243.83	90386.71	247.63	93967.27	257.44	95759.88	285.68	104273.38	0.00
12	1012	58392.59	159.98	73220.38	200.60	61517.68	168.08	64938.8	177.91	64219.38	175.94	67058.97	183.72	67770.90	185.41	67673.10	0.00
13	1013	21238.74	58.19	70821.47	194.03	79376.40	216.88	21322.2	58.42	20630.96	56.52	21118.32	57.86	20401.97	56.48	20614.70	0.00
14	1014	54181.74	148.44	70297.58	192.60	71079.63	194.21	56694.5	155.33	56102.81	153.71	59333.23	162.56	60367.09	170.74	62318.37	0.00
15	1015	60352.95	165.35	70160.21	192.22	75129.41	205.27	59261.5	162.36	58793.52	161.08	59135.63	162.02	57750.21	163.06	59518.05	0.00
16	1016	31994.30	87.66	69971.85	191.70	70252.06	191.95	31028.6	85.01	29936.32	82.02	31169.03	85.39	29004.66	77.25	28195.81	0.00
17	1017	53821.52	147.46	69778.86	191.17	71742.58	196.02	72966.6	199.91	73513.35	201.41	70072.99	191.98	71851.82	212.64	77612.15	0.00
18	1018	43931.65	120.36	68184.05	186.81	71984.22	196.68	45999.2	126.03	45362.27	124.28	49075.86	134.45	46209.77	125.72	45886.90	0.00
19	1019	40731.33	111.59	65332.59	178.99	68452.05	187.03	44920.4	123.07	46197.79	126.57	48622.29	133.21	48360.44	143.20	52266.78	0.00
20	1020	66327.09	181.72	64951.63	177.95	73143.96	199.85	66923.3	183.35	68311.84	187.16	72829.51	199.53	70786.25	189.32	69100.22	0.00
21	1021	90380.85	247.62	64875.50	177.74	66280.12	181.09	101407.0	277.83	100637.34	275.72	107681.16	295.02	111452.97	333.12	121589.48	0.00
22	1022	54928.37	150.49	62503.79	171.24	66509.47	181.72	56241.6	154.09	56299.54	154.25	60826.77	166.65	58670.93	159.90	58363.16	0.00
23	1023	58451.80	160.14	60880.44	166.80	62531.84	170.85	62327.9	170.76	63247.08	173.28	63411.89	173.73	61380.55	170.56	62256.21	0.00
24	1024	42815.74	117.30	59626.21	163.36	62931.45	171.94	42008.5	115.09	42497.82	116.43	41717.52	114.29	41835.00	117.97	43057.74	0.00

ข้อมูลชุดที่ 3 จำนวนคนย้าย-เข้าออกในกรุงเทพมหานคร



ข้อมูลชุดที่ 3 จำนวนคนย้ายเข้า-ย้ายออกในกรุงเทพมหานคร

	dcode	movein52	movein53	movein54	movein55	movein56	movein57	movein58	movein59	movein60	movein61	moveout52	moveout53	moveout54	moveout55	moveout56	moveout57	moveout58	moveout59	moveout60	moveout61
0	1045	8406	7353	7078	6911	6862	6571	6284	6355	6413	6050	8218	7572	7683	7512	7294	7472	7306	7174	9393	9935
1	1035	9116	8257	7480	7685	7643	7783	7552	7238	6795	6850	10106	9990	10062	9750	9046	8600	8362	7891	9072	9141
2	1049	8941	8203	7716	7085	8209	8761	7381	7126	6860	6706	7153	6679	6426	5819	6418	6476	5567	6083	4329	3948
3	1002	7534	7352	7890	7352	9233	6847	6430	9438	6859	9040	10797	10796	10189	10379	12305	11474	11133	13091	4878	4750
4	1038	9535	8939	7979	7687	7635	7303	6712	6429	6105	6036	8164	8693	8131	7691	7441	7713	7354	7467	5486	5415
5	1050	8341	8811	8034	7220	6840	6857	6022	5615	5815	5651	7301	7483	7409	6492	6655	6886	6188	5789	8086	7984
6	1022	9164	7597	8356	7142	6795	7173	6209	6107	5697	5450	10295	9449	9650	8875	8741	9246	8749	8452	5800	5686
7	1027	11546	9854	9546	9354	9382	8863	8844	8579	8122	7892	10781	9884	9850	9524	9120	8985	9379	8462	6069	7000
8	1006	11369	9980	9547	9719	10213	9317	9292	9016	3919	3396	11425	10915	10488	10134	9851	9852	9549	9826	3518	3503
9	1010	11085	10224	9786	9588	9840	9419	8969	8763	8485	8070	10491	10444	10677	10677	12090	11916	12268	12231	8724	8300
10	1030	11871	9604	10438	10470	9911	9367	8783	8817	8646	8867	12819	10987	11668	11446	10450	11026	10564	10502	11802	12206
11	1040	12286	11375	11061	12532	10631	10613	10064	10285	3184	3160	13304	12797	12475	13649	11020	11661	10185	10126	14151	13804
12	1003	11544	11400	11730	11498	11094	10765	10681	10747	10815	10546	7294	7683	8410	8223	7920	7722	7611	7536	3330	3373
13	1032	12882	11454	11773	11911	11844	11953	11576	12813	11860	11633	11810	11609	10907	10520	11698	11222	10721	10949	3893	3747
14	1023	13146	12321	12791	11300	10798	10729	9852	9588	8759	8835	9564	9528	9685	9322	9157	9296	8407	8410	7403	6931
15	1036	14839	14355	13173	14912	14027	13582	13317	14484	13697	14315	13124	12659	12534	13503	12119	12548	12382	12982	8939	9191
16	1011	16392	15118	14147	12587	12377	12478	11494	11843	11330	11594	10844	9875	10210	9835	9526	9398	9167	9343	7116	7026
17	1005	16617	15512	14312	14858	13709	12711	12317	14337	8324	7999	14262	13287	13711	13304	12965	12767	12299	13537	4914	4890
18	1046	15189	15951	15357	15446	14986	15400	4893	15742	14849	14765	9531	9664	9889	10309	9770	9888	9921	9732	7465	7542
19	1042	15748	17826	16047	15046	15316	14971	14928	14576	14231	15583	14965	15063	14400	14641	16082	15837	15594	15326	9625	9867
20	1021	14546	16245	16665	13912	13611	13274	12968	12693	5605	5293	9939	11869	11591	10680	10530	10476	10046	10256	6198	6553
21	1029	9455	7398	6816	5932	5764	5893	5686	5510	6214	6492	13010	11429	9488	8562	7148	7000	6502	6296	10115	10118
22	1019	6378	5788	6497	5522	5224	5200	5045	4855	5064	4644	6178	5401	5712	5191	4957	4903	4674	4413	11599	10576
23	1026	7985	7269	6430	6480	6812	6034	6027	5378	5454	5943	9142	8418	7806	7407	7146	6824	6587	6152	3390	3341

ข้อมูลชุดที่ 1

dcode = PK ของตาราง คือเขตภายในกรุงเทพมหานคร

population48 = จำนวนประชากรในปี 2548

population48 = จำนวนประชากรในปี 2549 เป็นต้น

ข้อมูลชุดที่ 2

dcode = PK ของตาราง คือเขตภายในกรุงเทพมหานคร

garbage53 = ปริมาณขยะของกรุงเทพในปี 2553

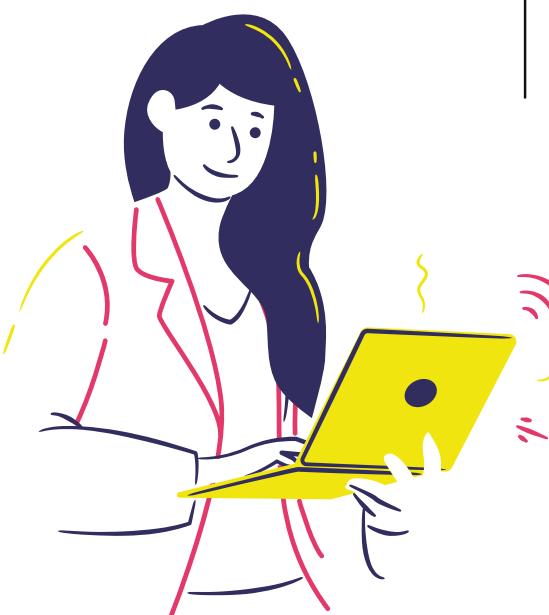
garbageperday53 = ปริมาณขยะของกรุงเทพต่อวันในปี 2553 เป็นต้น

ข้อมูลชุดที่ 3

dcode = PK ของตาราง คือเขตภายในกรุงเทพมหานคร

movein52 = จำนวนคนที่ย้ายเข้ากรุงเทพในปี 2552

moveout52 = จำนวนคนที่ย้ายออกจากกรุงเทพในปี 2552 เป็นต้น



Data Preprocessing

เรียง dcode ใหม่ ให้เป็นไปตามลำดับ

+ Code + Text

[9] 1 moveinmoveout_new = moveinmoveout.sort_values(by=['dcode'], ascending=True)
2 moveinmoveout_new

	dcode	movein52	movein53	movein54	movein55	movein56	movein57	movein58	movein59	movein60	movein61	moveout52	moveout53	moveout54	moveout55	moveout56	moveout57	moveout58	moveout59	moveout60	moveout61
45	1001	3461	2965	2892	3189	2909	2647	2515	2815	2660	2961	4341	3573	3842	3790	3543	3590	3589	3644	10292	11239
3	1002	7534	7352	7890	7352	9233	6847	6430	9438	6859	9040	10797	10796	10189	10379	12305	11474	11133	13091	4878	4750
12	1003	11544	11400	11730	11498	11094	10765	10681	10747	10815	10546	7294	7683	8410	8223	7920	7722	7611	7536	3330	3373
47	1004	1878	1797	1626	1614	1649	1533	1499	1656	1731	2288	5586	5191	5153	5582	6496	6043	5948	5967	5290	5174
17	1005	16617	15512	14312	14858	13709	12711	12317	14337	8324	7999	14262	13287	13711	13304	12965	12767	12299	13537	4914	4890
8	1006	11369	9980	9547	9719	10213	9317	9292	9016	3919	3396	11425	10915	10488	10134	9851	9852	9549	9826	3518	3503
46	1007	2753	2880	2245	2170	2490	2173	2183	2016	1890	2074	6138	6134	5804	5393	10630	9875	9502	8863	8230	8308
48	1008	1797	1587	1441	1413	1380	1364	1249	1200	1200	1222	4773	4340	4392	3992	3859	3852	3615	3340	6400	6252
35	1009	5941	5243	5034	4873	4989	5266	5147	5586	4375	4058	6129	5925	5532	5362	5123	5000	4846	5796	7096	6820
9	1010	11085	10224	9786	9588	9840	9419	8969	8763	8485	8070	10491	10444	10677	10677	12090	11916	12268	12231	8724	8300
16	1011	16392	15118	14147	12587	12377	12478	11494	11843	11330	11594	10844	9875	10210	9835	9526	9398	9167	9343	7116	7026
38	1012	4799	4621	4300	4331	4636	4307	3981	4083	3965	3537	5479	5476	5352	4871	4523	4257	4161	4188	7757	7581
49	1013	1036	815	753	783	816	740	621	673	625	681	1330	1200	1097	1165	1143	1092	1083	1268	4425	4281
33	1014	5667	5338	5467	5890	5773	5822	5438	6717	5843	6447	7636	7098	7286	7772	7976	8278	7643	8460	10887	10506
34	1015	6280	5297	5315	5180	4896	4685	4348	4521	4497	4374	9975	8990	8852	8071	8105	8140	7642	7901	1080	997
44	1016	3852	3577	3046	3141	2974	2956	2994	2486	9749	10087	4611	4093	4083	4242	3583	3514	3212	3366	7618	7760
32	1017	5248	5178	5480	5408	5400	5799	5699	6049	5036	5326	6546	6312	6099	6544	6103	6174	6396	6943	5564	5548
40	1018	4212	3802	4154	3562	3540	3410	15956	2959	2645	2593	7528	6237	6673	5865	6202	6134	5508	5304	11036	11486
22	1019	6378	5788	6497	5522	5224	5200	5045	4855	5064	4644	6178	5401	5712	5191	4957	4903	4674	4413	11599	10576
28	1020	7350	6144	5877	5294	5906	5717	5377	4505	12507	13396	13673	12302	11912	11083	15897	15747	15834	14251	9099	7914
20	1021	14546	16245	16665	13912	13611	13274	12968	12693	5605	5293	9939	11869	11591	10680	10530	10476	10046	10256	6198	6553
6	1022	9164	7597	8356	7142	6795	7173	6209	6107	5697	5450	10295	9449	9650	8875	8741	9246	8749	8452	5800	5686
14	1023	13146	12321	12791	11300	10798	10729	9852	9588	8759	8835	9564	9528	9685	9322	9157	9296	8407	8410	7403	6931
37	1024	5794	4679	4493	4228	4253	4345	4077	4031	4024	4038	7779	6662	6451	6310	5908	6043	5601	5561	15319	14121



Data Preprocessing



หา Missing Value

```
[1] 1 population.isnull().any()
```

```
dcode      False  
population48  False  
population49  False  
population50  False  
population51  False  
population52  False  
population53  False  
population54  False  
population55  False  
population56  False  
population57  False  
population58  False  
population59  False  
population60  False  
population61  False  
dtype: bool
```

จากการตรวจสอบค่า Missing พบว่า ไม่มีค่าไหนในตารางที่มีช่องว่างเลย

```
[11] 1 garbage.isnull().any()
```

```
dcode      False  
garbage53  False  
garbageperday53  False  
garbage54  False  
garbageperday54  False  
garbage55  False  
garbageperday55  False  
garbage56  False  
garbageperday56  False  
garbage57  False  
garbageperday57  False  
garbage58  False  
garbageperday58  False  
garbage59  False  
garbageperday59  False  
garbage60  False  
garbageperday60  False  
garbage61  False  
garbageperday61  False  
dtype: bool
```

จากการตรวจสอบค่า Missing พบว่า ไม่มีค่าไหนในตารางที่มีช่องว่างเลย

```
[12] 1 moveinmoveout.isnull().any()
```

```
dcode      False  
movein52  False  
movein53  False  
movein54  False  
movein55  False  
movein56  False  
movein57  False  
movein58  False  
movein59  False  
movein60  False  
movein61  False  
moveout52  False  
moveout53  False  
moveout54  False  
moveout55  False  
moveout56  False  
moveout57  False  
moveout58  False  
moveout59  False  
moveout60  False  
moveout61  False  
dtype: bool
```

จากการตรวจสอบค่า Missing พบว่า ไม่มีค่าไหนในตารางที่มีช่องว่างเลย

Data Preprocessing



ตัด column ตั้งแต่ปี 48-52 ออก เพราะต้องการตั้งแต่ปี 53

```
[15] 1 population.drop(['population48', 'population49', 'population50', 'population51', 'population52'], axis=1, inplace=True)
```

1 population

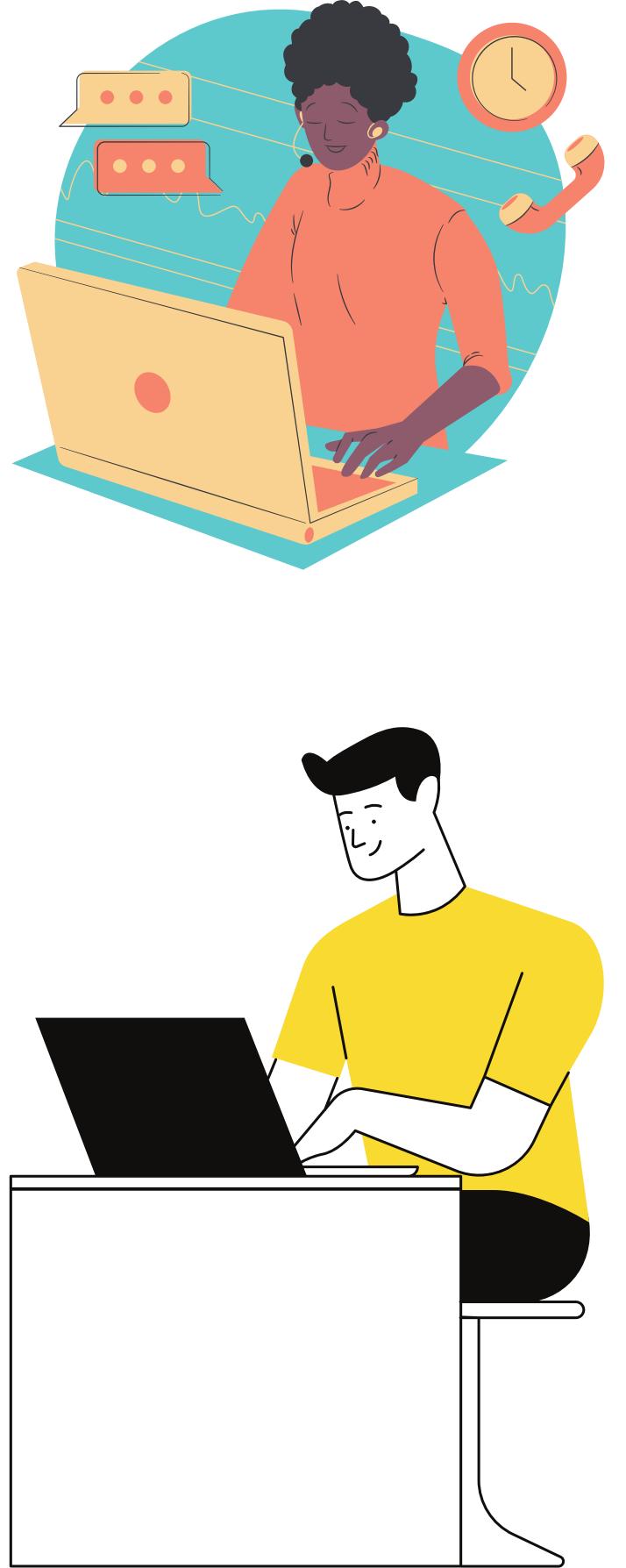
18	1019	106753	106786	106532	106192	105857	105585	105289	105047
19	1020	124352	120032	117950	117503	116653	115479	112581	110417
20	1021	155821	161642	165693	169614	173144	175829	179768	183878
21	1022	131363	130493	129800	129559	129238	128581	127582	125981
22	1023	145361	148298	150285	151877	153175	154018	155229	156267
23	1024	89297	87841	86643	85825	84881	84256	83248	81806
24	1025	101276	100319	99153	98113	96787	95829	93771	91278
25	1026	134480	131847	130202	128838	127260	126191	123966	120761
26	1027	147030	146197	145795	145822	145514	144951	144449	142990
27	1028	88179	86214	85048	83898	82432	81877	80497	78860
28	1029	138653	135001	132169	130511	128995	127961	126136	125299
29	1030	162838	161409	160853	160948	160366	159642	158130	155923
30	1031	98870	96422	94877	93508	92273	91574	90377	88288
31	1032	156567	158457	160816	163485	166364	168369	172761	178290
32	1033	112906	110481	109001	108066	107221	106467	104211	101543
33	1034	115966	115419	115731	116688	118371	119794	121740	123026
34	1035	160451	158646	157970	157156	156030	155106	153668	151174
35	1036	166354	166210	166635	167827	168197	169255	168896	169259
36	1037	76233	72900	73280	73550	73790	73860	72436	71952
37	1038	122520	122180	122152	82637	83520	121943	121000	119709
38	1039	80929	80847	81755	122441	122196	83982	84528	85642
39	1040	193190	192276	191781	192119	191966	192151	192413	193315
40	1041	112908	111120	109858	109049	107797	106935	105588	104577
41	1042	183333	185987	188163	191536	194511	197063	200374	204532

Data Preprocessing

```
[13] 1 population.drop(['population60'], axis=1, inplace=True)

1 population

dcode  population48  population49  population50  population51  population52  population53  population54  population55  population56  population57  population58  population59  population61
0    1001      67357      65835      64356      62966      61374      60313      58771      57831      56684      55373      54195      52522      50382
1    1002     121336     119927     117867     116742     114488     111496     108815     107969     106811     104394     103230     98450      94854
2    1003     126126     133415     138667     143675     147668     151292     154371     157224     159962     162598     164494     167844     172990
3    1004      50023      49730      49124      48506      47503      47053      46087      46112      46114      46472      46606      47308      48207
4    1005     178986     181390     182335     183836     185901     188164     188252     189737     190544     190659     190528     190828     191323
5    1006     149093     149860     150139     150286     150166     149606     148645     148491     52613      51557      149070     148392     146841
6    1007      63192      62102      61040      60275      58858      57368      54996      53912      50092      49280      50852      49594      48382
7    1008      60001      58768      57461      56464      54601      53526      52093      50930      149056     148964     48615      47450      45701
8    1009      98564      98096      98496      97794      96880      95661      94482      93461      92774      92448      92206      91305      89237
9    1010     118019     122825     127727     131035     133149     135032     136236     137295     138661     139771     140335     141214     142311
10   1011     138327     142460     144800     147482     152528     157477     160850     163317     165724     168309     169517     171933     175662
11   1012      88556      88383      88061      87386      85789      84286      82481      81529      81162      80843      80381      79574      78031
12   1013      31674      31142      30646      30088      29283      28617      28001      27426      26932      26359      25863      24785      23655
13   1014      77232      77343      77202      76477      75493      74693      73533      73084      72495      72203      72283      72102      70341
14   1015     136971     134589     132034     129662     126883     124499     121539     119643     117536     115330     113622     111027     107754
15   1016      81727      80863      79637      78307      76608      75621      73864      72241      71087      70003      69543      67887      67211
16   1017      76213      76402      77033      76948      76987      77292      77720      78207      78943      80002      80549      81190      81689
17   1018      87853      86163      84821      82824      80894      79546      77471      76353      75765      75224      74836      73871      72171
18   1019     105730     106811     107812     107513     106963     106753     106786     106532     106192     105857     105585     105289     105047
19   1020     133669     132394     130540     129401     126823     124352     120032     117950     117503     116653     115479     112581     110417
20   1021     132313     137934     141698     145294     150492     155821     161642     165693     169614     173144     175829     179768     183878
21   1022     136240     135149     134407     133622     132670     131363     130493     129800     129559     129238     128581     127582     125981
```

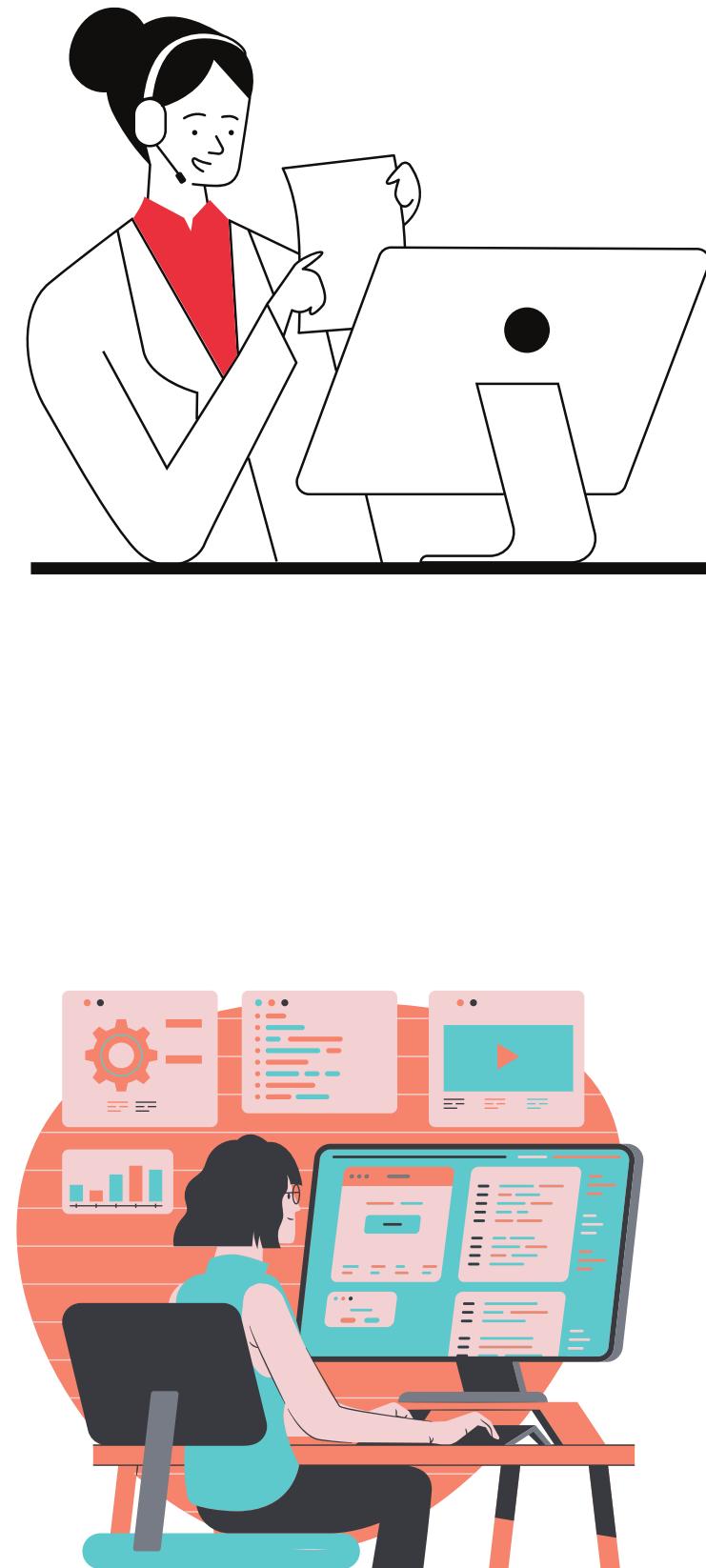


Data Preprocessing

```
[20] 1 moveinmoveout_new.drop(['movein60','moveout60'], axis=1, inplace=True)
```

```
1 moveinmoveout_new
```

	dcode	movein52	movein53	movein54	movein55	movein56	movein57	movein58	movein59	movein61	moveout52	moveout53	moveout54	moveout55	moveout56	moveout57	moveout58	moveout59	moveout61
45	1001	3461	2965	2892	3189	2909	2647	2515	2815	2961	4341	3573	3842	3790	3543	3590	3589	3644	11239
3	1002	7534	7352	7890	7352	9233	6847	6430	9438	9040	10797	10796	10189	10379	12305	11474	11133	13091	4750
12	1003	11544	11400	11730	11498	11094	10765	10681	10747	10546	7294	7683	8410	8223	7920	7722	7611	7536	3373
47	1004	1878	1797	1626	1614	1649	1533	1499	1656	2288	5586	5191	5153	5582	6496	6043	5948	5967	5174
17	1005	16617	15512	14312	14858	13709	12711	12317	14337	7999	14262	13287	13711	13304	12965	12767	12299	13537	4890
8	1006	11369	9980	9547	9719	10213	9317	9292	9016	3396	11425	10915	10488	10134	9851	9852	9549	9826	3503
46	1007	2753	2880	2245	2170	2490	2173	2183	2016	2074	6138	6134	5804	5393	10630	9875	9502	8863	8308
48	1008	1797	1587	1441	1413	1380	1364	1249	1200	1222	4773	4340	4392	3992	3859	3852	3615	3340	6252
35	1009	5941	5243	5034	4873	4989	5266	5147	5586	4058	6129	5925	5532	5362	5123	5000	4846	5796	6820
9	1010	11085	10224	9786	9588	9840	9419	8969	8763	8070	10491	10444	10677	10677	12090	11916	12268	12231	8300
16	1011	16392	15118	14147	12587	12377	12478	11494	11843	11594	10844	9875	10210	9835	9526	9398	9167	9343	7026
38	1012	4799	4621	4300	4331	4636	4307	3981	4083	3537	5479	5476	5352	4871	4523	4257	4161	4188	7581
49	1013	1036	815	753	783	816	740	621	673	681	1330	1200	1097	1165	1143	1092	1083	1268	4281
33	1014	5667	5338	5467	5890	5773	5822	5438	6717	6447	7636	7098	7286	7772	7976	8278	7643	8460	10506
34	1015	6280	5297	5315	5180	4896	4685	4348	4521	4374	9975	8990	8852	8071	8105	8140	7642	7901	997
44	1016	3852	3577	3046	3141	2974	2956	2994	2486	10087	4611	4093	4083	4242	3583	3514	3212	3366	7760
32	1017	5248	5178	5480	5408	5400	5799	5699	6049	5326	6546	6312	6099	6544	6103	6174	6396	6943	5548
40	1018	4212	3802	4154	3562	3540	3410	15956	2959	2593	7528	6237	6673	5865	6202	6134	5508	5304	11486
22	1019	6378	5788	6497	5522	5224	5200	5045	4855	4644	6178	5401	5712	5191	4957	4903	4674	4413	10576
28	1020	7350	6144	5877	5294	5906	5717	5377	4505	13396	13673	12302	11912	11083	15897	15747	15834	14251	7914
20	1021	14546	16245	16665	13912	13611	13274	12968	12693	5293	9939	11869	11591	10680	10530	10476	10046	10256	6553
6	1022	9164	7597	8356	7142	6795	7173	6209	6107	5450	10295	9449	9650	8875	8741	9246	8749	8452	5686
14	1023	13146	12321	12791	11300	10798	10729	9852	9588	8835	9564	9528	9685	9322	9157	9296	8407	8410	6931
37	1024	5794	4679	4493	4228	4253	4345	4077	4031	4038	7779	6662	6451	6310	5908	6043	5601	5561	14121



Data Preprocessing

ตัด column ของปี 60 ออกทั้งตาราง garbage และ moveinmoveout

```
[17] 1 garbage.drop(['garbage60','garbageperday60'], axis=1, inplace=True)
```

```
[18] 1 garbage
```

	dcode	garbage53	garbageperday53	garbage54	garbageperday54	garbage55	garbageperday55	garbage56	garbageperday56	garbage57	garbageperday57	garbage58	garbageperday58	garbage59	garbageperday59	garbage61	garbageperday61
0	1000	189819.45	520.05	122869.76	336.63	133599.00	365.02	206932.0	566.94	210940.80	577.92	217247.09	2.74	199522.73	545.14	219418.80	601.15
1	1001	69166.29	189.50	106149.08	290.82	119485.81	326.46	71844.4	196.83	69299.50	189.86	72019.02	197.31	70243.12	200.00	71372.82	195.54
2	1002	62244.26	170.53	105379.61	288.71	116.37	317.96	64255.4	176.04	62915.62	172.37	63580.71	174.19	63750.08	173.51	62186.37	170.37
3	1003	32339.16	88.60	95059.04	260.44	97084.32	265.26	40386.7	110.65	58888.59	161.34	43111.42	118.11	43545.74	131.30	50319.74	137.86
4	1004	54224.38	148.56	94655.76	259.33	100306.22	274.06	60767.8	166.49	94551.77	259.05	61746.75	169.17	58286.80	167.05	56632.20	155.16
5	1005	85742.96	234.91	87370.03	239.37	98955.18	270.37	93232.5	255.43	119561.54	327.57	99176.79	271.72	102066.16	302.24	112955.13	309.47
6	1006	101745.01	278.75	87276.00	239.11	92009.14	251.39	120589.0	330.38	102164.88	279.90	113420.03	310.74	106544.60	338.62	124970.88	342.39
7	1007	75719.25	207.45	86006.01	235.63	94435.68	258.02	104379.0	285.97	30773.10	84.31	104085.17	285.16	30383.52	309.70	111800.66	306.30
8	1008	32203.77	88.23	82532.46	226.12	99912.63	272.99	32474.1	88.97	56855.12	155.77	31207.79	85.50	117706.24	83.97	29790.81	81.62
9	1009	48978.52	134.19	78200.31	214.25	87574.21	239.27	56903.1	155.90	43346.67	118.76	54404.30	149.05	53592.83	147.54	53164.78	145.66
10	1010	60341.96	165.32	75249.90	206.16	84154.63	229.93	69510.6	190.44	70706.48	193.72	69591.76	190.66	69050.94	190.23	69576.22	190.62
11	1011	71764.46	196.61	73467.95	201.28	80198.83	219.12	88997.5	243.83	90386.71	247.63	93967.27	257.44	95759.88	285.68	109139.10	299.01
12	1012	58392.59	159.98	73220.38	200.60	61517.68	168.08	64938.8	177.91	64219.38	175.94	67058.97	183.72	67770.90	185.41	68049.42	186.44
13	1013	21238.74	58.19	70821.47	194.03	79376.40	216.88	21322.2	58.42	20630.96	56.52	21118.32	57.86	20401.97	56.48	19879.55	54.46
14	1014	54181.74	148.44	70297.58	192.60	71079.63	194.21	56694.5	155.33	56102.81	153.71	59333.23	162.56	60367.09	170.74	64478.78	176.65
15	1015	60352.95	165.35	70160.21	192.22	75129.41	205.27	59261.5	162.36	58793.52	161.08	59135.63	162.02	57750.21	163.06	59648.60	163.42
16	1016	31994.30	87.66	69971.85	191.70	70252.06	191.95	31028.6	85.01	29936.32	82.02	31169.03	85.39	29004.66	77.25	28806.92	78.92
17	1017	53821.52	147.46	69778.86	191.17	71742.58	196.02	72966.6	199.91	73513.35	201.41	70072.99	191.98	71851.82	212.64	83483.37	228.72
18	1018	43931.65	120.36	68184.05	186.81	71984.22	196.68	45999.2	126.03	45362.27	124.28	49075.86	134.45	46209.77	125.72	46112.37	126.34
19	1019	40731.33	111.59	65332.59	178.99	68452.05	187.03	44920.4	123.07	46197.79	126.57	48622.29	133.21	48360.44	143.20	54321.82	148.83
20	1020	66327.09	181.72	64951.63	177.95	73143.96	199.85	66923.3	183.35	68311.84	187.16	72829.51	199.53	70786.25	189.32	61571.42	168.69

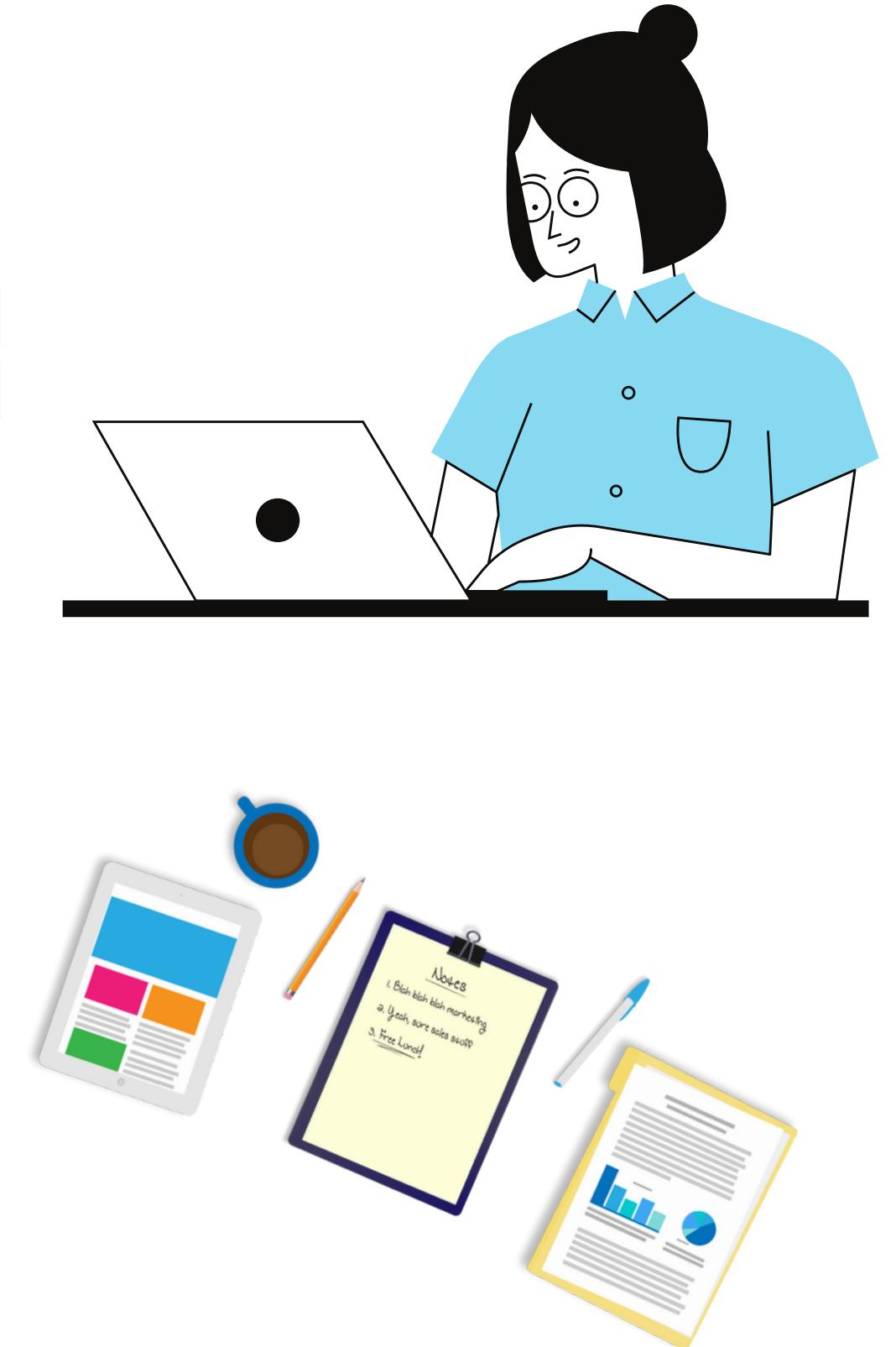
Data Preprocessing

ตัด column ปี 52 ออก เพราะต้องการตั้งแต่ปี 53

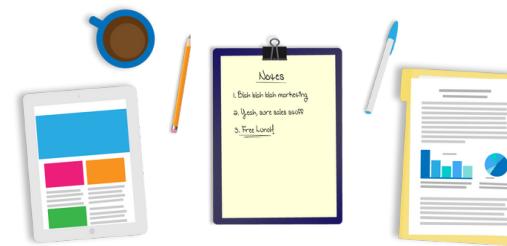
```
[22] 1 moveinmoveout_new.drop(['movein52','moveout52'], axis=1, inplace=True)
```

```
[23] 1 moveinmoveout_new
```

	dcode	movein53	movein54	movein55	movein56	movein57	movein58	movein59	movein61	moveout53	moveout54	moveout55	moveout56	moveout57	moveout58	moveout59	moveout61
45	1001	2965	2892	3189	2909	2647	2515	2815	2961	3573	3842	3790	3543	3590	3589	3644	11239
3	1002	7352	7890	7352	9233	6847	6430	9438	9040	10796	10189	10379	12305	11474	11133	13091	4750
12	1003	11400	11730	11498	11094	10765	10681	10747	10546	7683	8410	8223	7920	7722	7611	7536	3373
47	1004	1797	1626	1614	1649	1533	1499	1656	2288	5191	5153	5582	6496	6043	5948	5967	5174
17	1005	15512	14312	14858	13709	12711	12317	14337	7999	13287	13711	13304	12965	12767	12299	13537	4890
8	1006	9980	9547	9719	10213	9317	9292	9016	3396	10915	10488	10134	9851	9852	9549	9826	3503
46	1007	2880	2245	2170	2490	2173	2183	2016	2074	6134	5804	5393	10630	9875	9502	8863	8308
48	1008	1587	1441	1413	1380	1364	1249	1200	1222	4340	4392	3992	3859	3852	3615	3340	6252
35	1009	5243	5034	4873	4989	5266	5147	5586	4058	5925	5532	5362	5123	5000	4846	5796	6820
9	1010	10224	9786	9588	9840	9419	8969	8763	8070	10444	10677	10677	12090	11916	12268	12231	8300
16	1011	15118	14147	12587	12377	12478	11494	11843	11594	9875	10210	9835	9526	9398	9167	9343	7026
38	1012	4621	4300	4331	4636	4307	3981	4083	3537	5476	5352	4871	4523	4257	4161	4188	7581
49	1013	815	753	783	816	740	621	673	681	1200	1097	1165	1143	1092	1083	1268	4281
33	1014	5338	5467	5890	5773	5822	5438	6717	6447	7098	7286	7772	7976	8278	7643	8460	10506
34	1015	5297	5315	5180	4896	4685	4348	4521	4374	8990	8852	8071	8105	8140	7642	7901	997
44	1016	3577	3046	3141	2974	2956	2994	2486	10087	4093	4083	4242	3583	3514	3212	3366	7760
32	1017	5178	5480	5408	5400	5799	5699	6049	5326	6312	6099	6544	6103	6174	6396	6943	5548
40	1018	3802	4154	3562	3540	3410	15956	2959	2593	6237	6673	5865	6202	6134	5508	5304	11486
22	1019	5788	6497	5522	5224	5200	5045	4855	4644	5401	5712	5191	4957	4903	4674	4413	10576
28	1020	6144	5877	5294	5906	5717	5377	4505	13396	12302	11912	11083	15897	15747	15834	14251	7914
20	1021	16245	16665	13912	13611	13274	12968	12693	5293	11869	11591	10680	10530	10476	10046	10256	6553
6	1022	7597	8356	7142	6795	7173	6209	6107	5450	9449	9650	8875	8741	9246	8749	8452	5686
14	1023	12321	12791	11300	10798	10729	9852	9588	8835	9528	9685	9322	9157	9296	8407	8410	6931



Data Preprocessing



```
[28] 1 tablepop_garb_moveinout = pd.merge(tablepop_garbage, moveinmoveout_new, how='left', on='dcode')
```

```
[29] 1 tablepop_garb_moveinout
```

	dcode	population53	population54	population55	population56	population57	population58	population59	population61	garbage53	garbage54	garbage55	garbage56	garbage57	garbage58	garbage59	garbage61	movein53	movein54	mov
0	1001	60313	58771	57831	56684	55373	54195	52522	50382	69166.29	106149.08	119485.81	71844.4	69299.50	72019.02	70243.12	71372.82	2965	2892	
1	1002	111496	108815	107969	106811	104394	103230	98450	94854	62244.26	105379.61	116.37	64255.4	62915.62	63580.71	63750.08	62186.37	7352	7890	
2	1003	151292	154371	157224	159962	162598	164494	167844	172990	32339.16	95059.04	97084.32	40386.7	58888.59	43111.42	43545.74	50319.74	11400	11730	
3	1004	47053	46087	46112	46114	46472	46606	47308	48207	54224.38	94655.76	100306.22	60767.8	94551.77	61746.75	58286.80	56632.20	1797	1626	
4	1005	188164	188252	189737	190544	190659	190528	190828	191323	85742.96	87370.03	98955.18	93232.5	119561.54	99176.79	102066.16	112955.13	15512	14312	
5	1006	149606	148645	148491	52613	51557	149070	148392	146841	101745.01	87276.00	92009.14	120589.0	102164.88	113420.03	106544.60	124970.88	9980	9547	
6	1007	57368	54996	53912	50092	49280	50852	49594	48382	75719.25	86006.01	94435.68	104379.0	30773.10	104085.17	30383.52	111800.66	2880	2245	
7	1008	53526	52093	50930	149056	148964	48615	47450	45701	32203.77	82532.46	99912.63	32474.1	56855.12	31207.79	117706.24	29790.81	1587	1441	
8	1009	95661	94482	93461	92774	92448	92206	91305	89237	48978.52	78200.31	87574.21	56903.1	43346.67	54404.30	53592.83	53164.78	5243	5034	
9	1010	135032	136236	137295	138661	139771	140335	141214	142311	60341.96	75249.90	84154.63	69510.6	70706.48	69591.76	69050.94	69576.22	10224	9786	
10	1011	157477	160850	163317	165724	168309	169517	171933	175662	71764.46	73467.95	80198.83	88997.5	90386.71	93967.27	95759.88	109139.10	15118	14147	
11	1012	84286	82481	81529	81162	80843	80381	79574	78031	58392.59	73220.38	61517.68	64938.8	64219.38	67058.97	67770.90	68049.42	4621	4300	
12	1013	28617	28001	27426	26932	26359	25863	24785	23655	21238.74	70821.47	79376.40	21322.2	20630.96	21118.32	20401.97	19879.55	815	753	
13	1014	74693	73533	73084	72495	72203	72283	72102	70341	54181.74	70297.58	71079.63	56694.5	56102.81	59333.23	60367.09	64478.78	5338	5467	
14	1015	124499	121539	119643	117536	115330	113622	111027	107754	60352.95	70160.21	75129.41	59261.5	58793.52	59135.63	57750.21	59648.60	5297	5315	



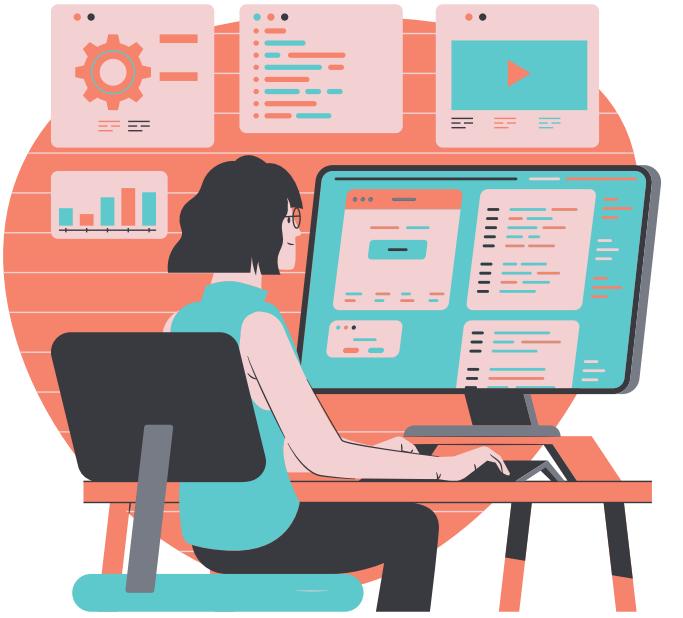
Data Preprocessing

ในตาราง garbage ตัด column garbageperday ของทุกปีออก เนื่องจากต้องการทราบเฉพาะปริมาณรวมขยะของแต่ละปี

```
[24] 1 garbage.drop(['garbageperday53', 'garbageperday54', 'garbageperday55', 'garbageperday56', 'garbageperday57', 'garbageperday58', 'garbageperday59', 'garbageperday61'], axis=1, inplace=True)
```

```
1 garbage
```

	dcode	garbage53	garbage54	garbage55	garbage56	garbage57	garbage58	garbage59	garbage61
0	1000	189819.45	122869.76	133599.00	206932.0	210940.80	217247.09	199522.73	219418.80
1	1001	69166.29	106149.08	119485.81	71844.4	69299.50	72019.02	70243.12	71372.82
2	1002	62244.26	105379.61	116.37	64255.4	62915.62	63580.71	63750.08	62186.37
3	1003	32339.16	95059.04	97084.32	40386.7	58888.59	43111.42	43545.74	50319.74
4	1004	54224.38	94655.76	100306.22	60767.8	94551.77	61746.75	58286.80	56632.20
5	1005	85742.96	87370.03	98955.18	93232.5	119561.54	99176.79	102066.16	112955.13
6	1006	101745.01	87276.00	92009.14	120589.0	102164.88	113420.03	106544.60	124970.88
7	1007	75719.25	86006.01	94435.68	104379.0	30773.10	104085.17	30383.52	111800.66
8	1008	32203.77	82532.46	99912.63	32474.1	56855.12	31207.79	117706.24	29790.81
9	1009	48978.52	78200.31	87574.21	56903.1	43346.67	54404.30	53592.83	53164.78
10	1010	60341.96	75249.90	84154.63	69510.6	70706.48	69591.76	69050.94	69576.22
11	1011	71764.46	73467.95	80198.83	88997.5	90386.71	93967.27	95759.88	109139.10
12	1012	58392.59	73220.38	61517.68	64938.8	64219.38	67058.97	67770.90	68049.42
13	1013	21238.74	70821.47	79376.40	21322.2	20630.96	21118.32	20401.97	19879.55
14	1014	54181.74	70297.58	71079.63	56694.5	56102.81	59333.23	60367.09	64478.78
15	1015	60352.95	70160.21	75129.41	59261.5	58793.52	59135.63	57750.21	59648.60
16	1016	31994.30	69971.85	70252.06	31028.6	29936.32	31169.03	29004.66	28806.92
17	1017	53821.52	69778.86	71742.58	72966.6	73513.35	70072.99	71851.82	83483.37
18	1018	43931.65	68184.05	71984.22	45999.2	45362.27	49075.86	46209.77	46112.37
19	1019	40731.33	65332.59	68452.05	44920.4	46197.79	48622.29	48360.44	54321.82
20	1020	66327.09	64951.63	73143.96	66923.3	68311.84	72829.51	70786.25	61571.42
21	1021	90380.85	64875.50	66280.12	101407.0	100637.34	107681.16	111452.97	122081.56
22	1022	54928.37	62503.79	66509.47	56241.6	56299.54	60826.77	58670.93	59137.82

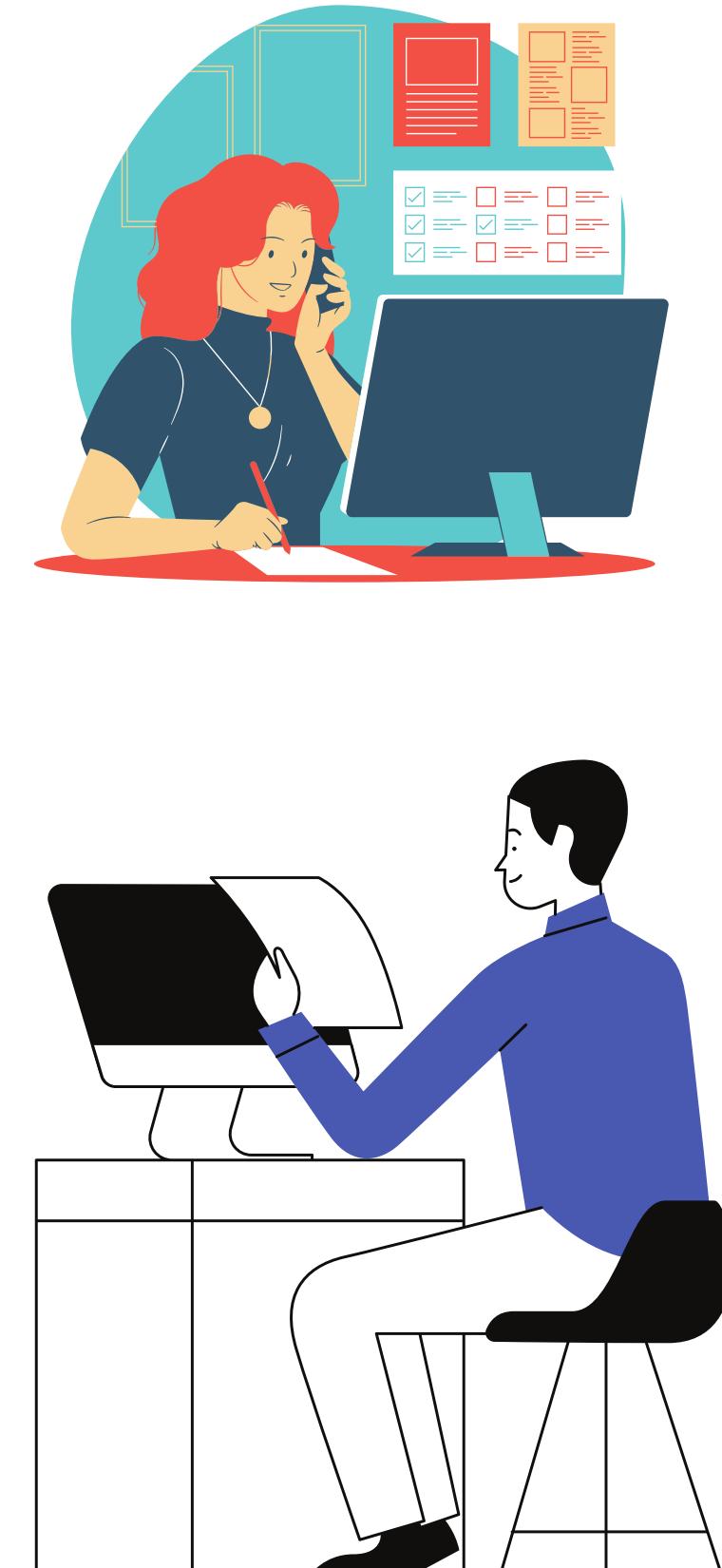


Data Preprocessing

รวมตาราง 2 ตารางเข้าด้วยกัน โดยให้ตาราง population เป็นตารางหลัก

```
1 merged_table = population.merge(garbage,how='left',left_on='dcode',right_on='dcode') #แบบอжаารย์  
2 merged_table
```

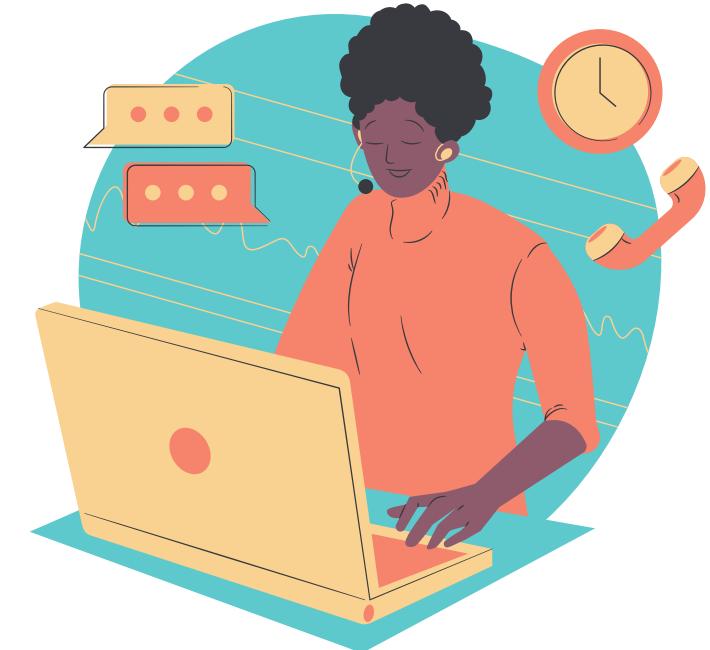
	dcode	population53	population54	population55	population56	population57	population58	population59	population61	garbage53	garbage54	garbage55	garbage56	garbage57	garbage58	garbage59	garbage61
0	1001	60313	58771	57831	56684	55373	54195	52522	50382	69166.29	106149.08	119485.81	71844.4	69299.50	72019.02	70243.12	71372.82
1	1002	111496	108815	107969	106811	104394	103230	98450	94854	62244.26	105379.61	116.37	64255.4	62915.62	63580.71	63750.08	62186.37
2	1003	151292	154371	157224	159962	162598	164494	167844	172990	32339.16	95059.04	97084.32	40386.7	58888.59	43111.42	43545.74	50319.74
3	1004	47053	46087	46112	46114	46472	46606	47308	48207	54224.38	94655.76	100306.22	60767.8	94551.77	61746.75	58286.80	56632.20
4	1005	188164	188252	189737	190544	190659	190528	190828	191323	85742.96	87370.03	98955.18	93232.5	119561.54	99176.79	102066.16	112955.13
5	1006	149606	148645	148491	52613	51557	149070	148392	146841	101745.01	87276.00	92009.14	120589.0	102164.88	113420.03	106544.60	124970.88
6	1007	57368	54996	53912	50092	49280	50852	49594	48382	75719.25	86006.01	94435.68	104379.0	30773.10	104085.17	30383.52	111800.66
7	1008	53526	52093	50930	149056	148964	48615	47450	45701	32203.77	82532.46	99912.63	32474.1	56855.12	31207.79	117706.24	29790.81
8	1009	95661	94482	93461	92774	92448	92206	91305	89237	48978.52	78200.31	87574.21	56903.1	43346.67	54404.30	53592.83	53164.78
9	1010	135032	136236	137295	138661	139771	140335	141214	142311	60341.96	75249.90	84154.63	69510.6	70706.48	69591.76	69050.94	69576.22
10	1011	157477	160850	163317	165724	168309	169517	171933	175662	71764.46	73467.95	80198.83	88997.5	90386.71	93967.27	95759.88	109139.10
11	1012	84286	82481	81529	81162	80843	80381	79574	78031	58392.59	73220.38	61517.68	64938.8	64219.38	67058.97	67770.90	68049.42
12	1013	28617	28001	27426	26932	26359	25863	24785	23655	21238.74	70821.47	79376.40	21322.2	20630.96	21118.32	20401.97	19879.55
13	1014	74693	73533	73084	72495	72203	72283	72102	70341	54181.74	70297.58	71079.63	56694.5	56102.81	59333.23	60367.09	64478.78
14	1015	124499	121539	119643	117536	115330	113622	111027	107754	60352.95	70160.21	75129.41	59261.5	58793.52	59135.63	57750.21	59648.60
15	1016	75621	73864	72241	71087	70003	69543	67887	67211	31994.30	69971.85	70252.06	31028.6	29936.32	31169.03	29004.66	28806.92
16	1017	77292	77720	78207	78943	80002	80549	81190	81689	53821.52	69778.86	71742.58	72966.6	73513.35	70072.99	71851.82	83483.37
17	1018	79546	77471	76353	75765	75224	74836	73871	72171	43931.65	68184.05	71984.22	45999.2	45362.27	49075.86	46209.77	46112.37
18	1019	106753	106786	106532	106192	105857	105585	105289	105047	40731.33	65332.59	68452.05	44920.4	46197.79	48622.29	48360.44	54321.82
19	1020	124352	120032	117950	117503	116653	115479	112581	110417	66327.09	64951.63	73143.96	66923.3	68311.84	72829.51	70786.25	61571.42
20	1021	155821	161642	165693	169614	173144	175829	179768	183878	90380.85	64875.50	66280.12	101407.0	100637.34	107681.16	111452.97	122081.56
21	1022	131363	130493	129800	129559	129238	128581	127582	125981	54928.37	62503.79	66509.47	56241.6	56299.54	60826.77	58670.93	59137.82
22	1023	145361	148298	150285	151877	153175	154018	155229	156267	58451.80	60880.44	62531.84	62327.9	63247.08	63411.89	61380.55	64632.91
23	1024	89297	87841	86643	85825	84881	84256	83248	81806	42815.74	59626.21	62931.45	42008.5	42497.82	41717.52	41835.00	43423.98



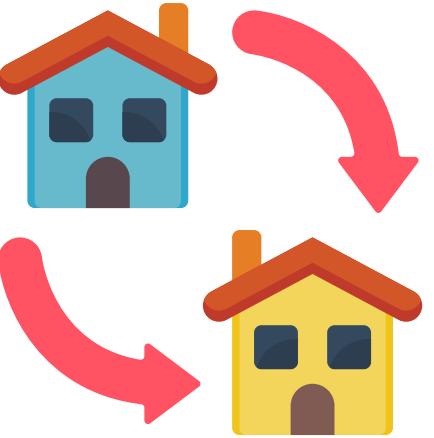
Data Preprocessing

```
i.merge(tablepop_garbage, moveinmoveout_new, how='left', on='dcode')
```

station54	population55	population56	population57	population58	population59	population61	garbage53	garbage54	garbage55	garbage56	garbage57	garbage58	garbage59
58771	57831	56684	55373	54195	52522	50382	69166.29	106149.08	119485.81	71844.4	69299.50	72019.02	70243.1
108815	107969	106811	104394	103230	98450	94854	62244.26	105379.61	116.37	64255.4	62915.62	63580.71	63750.0
154371	157224	159962	162598	164494	167844	172990	32339.16	95059.04	97084.32	40386.7	58888.59	43111.42	43545.7
46087	46112	46114	46472	46606	47308	48207	54224.38	94655.76	100306.22	60767.8	94551.77	61746.75	58286.8
188252	189737	190544	190659	190528	190828	191323	85742.96	87370.03	98955.18	93232.5	119561.54	99176.79	102066.1
148645	148491	52613	51557	149070	148392	146841	101745.01	87276.00	92009.14	120589.0	102164.88	113420.03	106544.6
54996	53912	50092	49280	50852	49594	48382	75719.25	86006.01	94435.68	104379.0	30773.10	104085.17	30383.5
52093	50930	149056	148964	48615	47450	45701	32203.77	82532.46	99912.63	32474.1	56855.12	31207.79	117706.2
94482	93461	92774	92448	92206	91305	89237	48978.52	78200.31	87574.21	56903.1	43346.67	54404.30	53592.8
136236	137295	138661	139771	140335	141214	142311	60341.96	75249.90	84154.63	69510.6	70706.48	69591.76	69050.9
160850	163317	165724	168309	169517	171933	175662	71764.46	73467.95	80198.83	88997.5	90386.71	93967.27	95759.8
82481	81529	81162	80843	80381	79574	78031	58392.59	73220.38	61517.68	64938.8	64219.38	67058.97	67770.9
28001	27426	26932	26359	25863	24785	23655	21238.74	70821.47	79376.40	21322.2	20630.96	21118.32	20401.9
73533	73084	72495	72203	72283	72102	70341	54181.74	70297.58	71079.63	56694.5	56102.81	59333.23	60367.0
121539	119643	117536	115330	113622	111027	107754	60352.95	70160.21	75129.41	59261.5	58793.52	59135.63	57750.2



ทำข้อมูลมาเพิ่ม



- เพิ่มเดิม - เพิ่มชื่อเขตแต่ละเขตว่าชื่ออะไรบ้าง และหาค่าที่หายไป

```
[32] 1 district = pd.read_csv(os.path.join(path, 'district.csv'))
```

- ตารางชื่อเขตภายในกรุงเทพมหานคร

```
[33] 1 district
```

	dcode	dname
0	1001	พระนคร
1	1002	คลอง
2	1003	หนองจอก
3	1004	บางซัก
4	1005	บางเขน
5	1006	บางกะปิ
6	1007	ปทุมวัน
7	1008	ป้อมปราบศัตรูพ่าย
8	1009	พระโขนง
9	1010	มีนบุรี
10	1011	ลาดกระบัง
11	1012	ยานนาวา
12	1013	สัมพันธวงศ์
13	1014	พญาไท
14	1015	ธนบุรี
15	1016	บางกอกใหญ่
16	1017	ทวายขาว
17	1018	คลองสาน
18	1019	คลองเตย
19	1020	บางกอกน้อย
20	1021	บางขุนเทียน

dcode = PK ของตาราง คือรหัสเขตภายในกรุงเทพมหานคร

dname = FK ของตาราง คือชื่อเขตภายในกรุงเทพมหานคร

รวมตาราง population, garbage และ moveinmoveout

```
[41]: 1 pop_gar_moveinout = pd.merge(pop_gar, moveinmoveout_new, how='left', on='dcode')
2 pop_gar_moveinout
```

	dcode	dname	population53	population54	population55	population56	population57	population58	population59	population61	garbage53	garbage54	garbage55	garbage56	garbage57	garbage58	garbage59	garbage61	movein53
0	1001	พระนคร	60313	58771	57831	56684	55373	54195	52522	50382	69166.29	106149.08	119485.81	71844.4	69299.50	72019.02	70243.12	71372.82	2965
1	1002	คุสิต	111496	108815	107969	106811	104394	103230	98450	94854	62244.26	105379.61	116.37	64255.4	62915.62	63580.71	63750.08	62186.37	7351
2	1003	หนองจอก	151292	154371	157224	159962	162598	164494	167844	172990	32339.16	95059.04	97084.32	40386.7	58888.59	43111.42	43545.74	50319.74	1140
3	1004	บางรัก	47053	46087	46112	46114	46472	46606	47308	48207	54224.38	94655.76	100306.22	60767.8	94551.77	61746.75	58286.80	56632.20	1791
4	1005	บางเขน	188164	188252	189737	190544	190659	190528	190828	191323	85742.96	87370.03	98955.18	93232.5	119561.54	99176.79	102066.16	112955.13	1551
5	1006	บางกะปิ	149606	148645	148491	52613	51557	149070	148392	146841	101745.01	87276.00	92009.14	120589.0	102164.88	113420.03	106544.60	124970.88	9980
6	1007	ปทุมวัน	57368	54996	53912	50092	49280	50852	49594	48382	75719.25	86006.01	94435.68	104379.0	30773.10	104085.17	30383.52	111800.66	2880
7	1008	ป้อมปราบศรีพายัพ	53526	52093	50930	149056	148964	48615	47450	45701	32203.77	82532.46	99912.63	32474.1	56855.12	31207.79	117706.24	29790.81	1581
8	1009	พระโขนง	95661	94482	93461	92774	92448	92206	91305	89237	48978.52	78200.31	87574.21	56903.1	43346.67	54404.30	53592.83	53164.78	5240
9	1010	มีนบุรี	135032	136236	137295	138661	139771	140335	141214	142311	60341.96	75249.90	84154.63	69510.6	70706.48	69591.76	69050.94	69576.22	1022
10	1011	ลาดกระบัง	157477	160850	163317	165724	168309	169517	171933	175662	71764.46	73467.95	80198.83	88997.5	90386.71	93967.27	95759.88	109139.10	1511
11	1012	ยานนาวา	84286	82481	81529	81162	80843	80381	79574	78031	58392.59	73220.38	61517.68	64938.8	64219.38	67058.97	67770.90	68049.42	462
12	1013	สัมพันธวงศ์	28617	28001	27426	26932	26359	25863	24785	23655	21238.74	70821.47	79376.40	21322.2	20630.96	21118.32	20401.97	19879.55	81
13	1014	พญาไท	74693	73533	73084	72495	72203	72283	72102	70341	54181.74	70297.58	71079.63	56694.5	56102.81	59333.23	60367.09	64478.78	5338
14	1015	ธนบุรี	124499	121539	119643	117536	115330	113622	111027	107754	60352.95	70160.21	75129.41	59261.5	58793.52	59135.63	57750.21	59648.60	5291
15	1016	บางกอกใหญ่	75621	73864	72241	71087	70003	69543	67887	67211	31994.30	69971.85	70252.06	31028.6	29936.32	31169.03	29004.66	28806.92	3571
16	1017	ทัพยานหง	77292	77720	78207	78943	80002	80549	81190	81689	53821.52	69778.86	71742.58	72966.6	73513.35	70072.99	71851.82	83483.37	5178
17	1018	คลองสาน	79546	77471	76353	75765	75224	74836	73871	72171	43931.65	68184.05	71984.22	45999.2	45362.27	49075.86	46209.77	46112.37	3802

dcode = PK ของตาราง คือเขตภายในกรุงเทพมหานคร

dname = FK ของตาราง คือชื่อเขตภายในกรุงเทพมหานคร

population53 = จำนวนประชากรคนกรุงเทพในปี 2553

garbage53 = ปริมาณขยะของกรุงเทพในปี 2553

movein = จำนวนคนที่ย้ายเข้ากรุงเทพในปี 2553

moveout = จำนวนคนที่ย้ายออกจากรุงเทพในปี 2553 เป็นต้น





Final Project

ข้อมูลเพิ่มเติม เนื่องจากข้อมูลเดิมไม่เพียงพอสำหรับการทำนาย

```
✓ [45] 1 import pandas as pd #ทำงานกับข้อมูลลักษณะตาราง
✓ [45] 1 from google.colab import drive #เชื่อมเข้ากับ Drive
2 drive.mount('/content/drive') # คลิกที่ลิงก์ -> เลือก E-mail -> ลงชื่อเข้าใช้ -> ก็อปปี้ลิงค์ -> นำกลับมาวางในช่องว่าง -> กด enter
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

✓ [46] 1 import os # os เอาไว้จัดการไฟล์และ path
✓ [47] 1 path = '/content/drive/My_Drive/DWDM2021_DATA'
✓ [48] 1 park = pd.read_csv(os.path.join(path, 'park.csv')) #เพิ่มข้อมูลจำนวนสวนสาธารณะเข้ามา
✓ [49] 1 park

dcode num_park54 num_park55 num_park56 num_park57 num_park58 num_park59 num_park60 num_park61 num_park62 area_park54 area_park55 area_park56 area_park57 area_park58 area_park59 area_park60 area_park61 area_park62
0 1001 89 95 97 98 108 111 0 0 113 401221.81 563001.81 567070.0 568117.81 585157.81 606289.56 0.0 0.0 627595.56
1 1002 100 103 106 108 115 115 0 0 123 361593.19 363253.19 364333.0 365977.19 367809.19 361269.19 0.0 0.0 1185376.00
2 1003 176 186 192 199 171 178 0 0 195 1162283.75 1268683.75 1329480.0 1394382.00 1329438.00 1376232.25 0.0 0.0 1424230.25
3 1004 144 168 191 211 233 243 0 0 268 87102.04 103108.04 112934.0 120068.04 126519.04 133979.05 0.0 0.0 154968.77
4 1005 43 44 54 62 68 72 0 0 76 474789.53 482989.53 536238.0 572784.00 665448.00 815448.00 0.0 0.0 1032750.75
5 1006 119 121 127 136 143 157 0 0 172 644095.50 646015.50 742432.0 765112.38 770133.81 793635.31 0.0 0.0 810526.56
6 1007 96 99 104 106 115 122 0 0 133 699892.13 700621.13 715021.0 713280.00 757036.00 776160.00 0.0 0.0 896952.00
7 1008 131 133 136 142 146 112 0 0 116 265522.59 263601.00 264005.0 271279.00 274299.00 256314.64 0.0 0.0 256912.64
8 1009 299 309 332 336 337 332 0 0 345 243367.77 247015.77 257960.0 261424.00 274561.75 279470.22 0.0 0.0 288507.78
9 1010 123 142 165 196 215 226 0 0 167 755547.25 876187.25 985987.0 1076918.25 1219158.25 1248934.75 0.0 0.0 764297.25
10 1011 94 99 99 107 118 114 0 0 164 494749.00 499949.00 502749.0 515804.00 434832.38 410592.38 0.0 0.0 486265.09
11 1012 154 185 199 203 212 215 0 0 238 452777.16 462764.75 477860.0 465028.00 475464.00 489892.00 0.0 0.0 533186.69
```

เลือกมาเฉพาะจำนวนสวนสาธารณะของปี 59

```
[50] 1 park59 = park.iloc[:,[0,6]]
```

เลือกแคปปิทสันใจคือปี 2559

```
✓ [50] 1 park59
```

	dcode	num_park59
0	1001	111
1	1002	115
2	1003	178
3	1004	243
4	1005	72
5	1006	157
6	1007	122
7	1008	112
8	1009	332
9	1010	226
10	1011	114
11	1012	215
12	1013	97

```
[52] 1 park59.isnull().any() #.isnull เพื่อคุณ missing และพบว่าไม่มีค่า missing
```

```
dcode      False  
num_park59  False  
dtype: bool
```

```
1 data2 = pd.merge(pop_gar_moveinout, park59, how='left', on='dcode')  
2 data2  
3 #เมื่อมีการข้อมูล pop_gar_moveinout กับตาราง park59
```

	dcode	dname	population53	population54	population55	population56	population57	population58	population59	population61	garbage53	garbage54	garbage55	garbage56	garbage57	garbage58	garbage59	garbage61	movein5:
0	1001	พระนคร	60313	58771	57831	56684	55373	54195	52522	50382	69166.29	106149.08	119485.81	71844.4	69299.50	72019.02	70243.12	71372.82	2968
1	1002	คุ้งตะเภา	111496	108815	107969	106811	104394	103230	98450	94854	62244.26	105379.61	116.37	64255.4	62915.62	63580.71	63750.08	62186.37	7351
2	1003	หนองจอก	151292	154371	157224	159962	162598	164494	167844	172990	32339.16	95059.04	97084.32	40386.7	58888.59	43111.42	43545.74	50319.74	11400
3	1004	บางรัก	47053	46087	46112	46114	46472	46606	47308	48207	54224.38	94655.76	100306.22	60767.8	94551.77	61746.75	58286.80	56632.20	1791
4	1005	นางยวน	188164	188252	189737	190544	190659	190528	190828	191323	85742.96	87370.03	98955.18	93232.5	119561.54	99176.79	102066.16	112955.13	15512
5	1006	บางกะปิ	149606	148645	148491	52613	51557	149070	148392	146841	101745.01	87276.00	92009.14	120589.0	102164.88	113420.03	106544.60	124970.88	9980
6	1007	ปทุมธานี	57368	54996	53912	50092	49280	50852	49594	48382	75719.25	86006.01	94435.68	104379.0	30773.10	104085.17	30383.52	111800.66	2880
7	1008	ป้อมปราบศัตรูพ่าย	53526	52093	50930	149056	148964	48615	47450	45701	32203.77	82532.46	99912.63	32474.1	56855.12	31207.79	117706.24	29790.81	1581
8	1009	



```
1 data59_2 = data2[['dcode','dname','population59','garbage59','movein59','moveout59','num_park59']]  
2 data59_2  
3  
4 #เลือกเฉพาะcolumn ที่เป็นปี 59
```

	dcode	dname	population59	garbage59	movein59	moveout59	num_park59
0	1001	พระนคร	52522	70243.12	2815	3644	111
1	1002	ดุสิต	98450	63750.08	9438	13091	115
2	1003	หนองจอก	167844	43545.74	10747	7536	178
3	1004	บางรัก	47308	58286.80	1656	5967	243
4	1005	บางเขน	190828	102066.16	14337	13537	72
5	1006	บางกะปิ	148392	106544.60	9016	9826	157
6	1007	ปทุมธานี	49594	30383.52	2016	8863	122
7	1008	ป้อมปราบศัตรูพ่าย	47450	117706.24	1200	3340	112
8	1009	พระโขนง	91305	53592.83	5586	5796	332
9	1010	มีนบุรี	141214	69050.94	8763	12231	226
10	1011	ลาดกระบัง	171933	95759.88	11843	9343	114
11	1012	ยานนาวา	79574	67770.90	4083	4188	215
12	1013	สัมพันธวงศ์	24785	20401.97	673	1268	97
13	1014	พญาไท	72102	60367.09	6717	8460	71
14	1015	ธนบุรี	111027	57750.21	4521	7901	126
15	1016	บางกอกใหญ่	67887	29004.66	2486	3366	100
16	1017	ห้วยขวาง	81190	71851.82	6049	6943	107
17	1018	คลองสาน	73871	46209.77	2959	5304	131

```
✓ 1 data59_2.isnull().any() #หาค่าmissing ปรากฏว่าไม่มีค่า missing เลย  
0s  
dcode      False  
dname      False  
population59  False  
garbage59  False  
movein59  False  
moveout59  False  
num_park59  False  
dtype: bool
```



เพิ่ม column AmountofGarbage

แบ่งปริมาณขยะ ในคอลัมน์ garbage59 เป็น

ปริมาณขยะน้อยมีค่า 0 - 50,000

ปริมาณขยะปานกลางมีค่า ระหว่าง 50,000 - 100,000

ปริมาณขยะมากมีค่า 100,000 - 150,000

```
[56] 1 from pandas.api.types import CategoricalDtype #import มาเพื่อแบ่ง Category ของปริมาณขยะ
```

```
[59] 1 Amount2 = ['low','median','high']
2 data59_2['Amount0fGarbage'] = pd.cut(data59_2.garbage59,
3 bins=[0,50000,100000,150000],
4 labels=Amount2,
5 right=False).astype(str).astype(CategoricalDtype(Amount2,ordered=True))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
"""
```

1 data59_2

	dcode	dname	population59	garbage59	movein59	moveout59	num_park59	Amount0fGarbage
0	1001	พะนัง	52522	70243.12	2815	3644	111	median
1	1002	ศรีสะเกษ	98450	63750.08	9438	13091	115	median
2	1003	หนองจอก	167844	43545.74	10747	7536	178	low
3	1004	นางรักษ์	47308	58286.80	1656	5967	243	median
4	1005	นางเขน	190828	102066.16	14337	13537	72	high
5	1006	นางกะปี	148392	106544.60	9016	9826	157	high
6	1007	ปทุมธานี	49594	30383.52	2016	8863	122	low
7	1008	ป้อมปราบศัตรูพ่าย	47450	117706.24	1200	3340	112	high
8	1009	พระโขนง	91305	53592.83	5586	5796	332	median
9	1010	เมืองนนทบุรี	141214	69050.94	8763	12231	226	median
10	1011	ลาดกระบัง	171933	95759.88	11843	9343	114	median
11	1012	ยานนาวา	79574	67770.90	4083	4188	215	median
12	1013	สัมพันธวงศ์	24785	20401.97	673	1268	97	low
13	1014	พญาไท	72102	60367.09	6717	8460	71	median
14	1015	ธนบุรี	111027	57750.21	4521	7901	126	median
15	1016	บางกอกใหญ่	67887	29004.66	2486	3366	100	low
16	1017	ห้วยขวาง	81190	71851.82	6049	6943	107	median
17	1018	คลองสาน	73871	46209.77	2959	5304	131	low
18	1019	คลองชัก	105289	48360.44	4855	4413	199	low

```
[61] 1 data59_2['AmountOfGarbage'] = data59_2['AmountOfGarbage'].map({'low':1,'median':2,'high':3})
2 data59_2
3 #.map ทำการเปลี่ยน low median high เป็นตัวเลข 1,2 และ 3 ตามลำดับ
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
    """Entry point for launching an IPython kernel.
      dcode      dname population59  garbage59  movein59  moveout59  num_park59  AmountOfGarbage
0  1001   พะนังคร  52522  70243.12   2815   3644    111      2
1  1002     คุสิต  98450  63750.08   9438  13091    115      2
2  1003  หนองจอก  167844  43545.74  10747   7536    178      1
3  1004     บางรัก  47308  58286.80   1656   5967    243      2
4  1005     บางเขน  190828 102066.16  14337  13537     72      3
5  1006     บางกะปิ  148392 106544.60   9016   9826    157      3
6  1007     ปทุมธานี  49594  30383.52   2016   8863    122      1
7  1008  มีอมปารานีคุรุพยา  47450  117706.24   1200   3340    112      3
8  1009     พระโขนง  91305  53592.83   5586   5796    332      2
9  1010     มีนบุรี  141214  69050.94   8763  12231    226      2
10 1011  ลาดกระบัง  171933  95759.88  11843   9343    114      2
11 1012     ยานนาวา  79574  67770.90   4083   4188    215      2
12 1013  สัมพันธวงศ์  24785  20401.97   673   1268     97      1
13 1014     พญาไท  72102  60367.09   6717   8460     71      2
14 1015     ขอนบุรี  111027  57750.21   4521   7901    126      2
15 1016     บางกอกใหญ่  67887  29004.66   2486   3366    100      1
16 1017     ท่องเที่ยว  81190  71851.82   6049   6943    107      2
17 1018  คลองสาน  73871  46209.77   2959   5304    131      1
```

```
[62] 1 X1 = data59_2[['num_park59','population59', 'movein59', 'moveout59']]
2 X1
3 #กำหนดค่าX คือ จำนวนสวนสาธารณะปี59 จำนวนประชากรปี59 และการย้ายเข้า-ออกปี59
```

	num_park59	population59	movein59	moveout59
0	111	52522	2815	3644
1	115	98450	9438	13091
2	178	167844	10747	7536
3	243	47308	1656	5967
4	72	190828	14337	13537
5	157	148392	9016	9826
6	122	49594	2016	8863
7	112	47450	1200	3340
8	332	91305	5586	5796
9	226	141214	8763	12231
10	114	171933	11843	9343
11	215	79574	4083	4188
12	97	24785	673	1268
13	71	72102	6717	8460
14	126	111027	4521	7901
15	100	67887	2486	3366



```
[63] 1 X1.dtypes
num_park59    int64
population59   int64
movein59      int64
moveout59     int64
dtype: object

[64] 1 X = X1.replace('^\d.', '', regex=True).astype(float)
2 #แทนค่าในตารางจาก int ให้เป็น float ทั้งหมด
```

```
1 y1 = data59_2[['AmountOfGarbage']]
2 y1 #ค่า y เป็นส่วนของตัวตั้งนี้
```

	AmountOfGarbage
0	2
1	2
2	1
3	2
4	3
5	3
6	1
7	3
8	2
9	2
10	2
11	2
12	1
13	2
14	2
15	1
16	2
17	1
18	1
19	2
20	3
21	2
22	2
23	1
24	2
25	2
26	2
27	2
28	2
29	3
30	2

```
[67] 1 y1.shape
(50, 1)

1 y2 = data59_2['AmountOfGarbage']
2 y2
```

	y2
0	2
1	2
2	1
3	2
4	3
5	3
6	1
7	3
8	2
9	2
10	2
11	2
12	1
13	2
14	2
15	1
16	2
17	1
18	1
19	2
20	3
21	2
22	2
23	1
24	2
25	2
26	2
27	2
28	2
29	3
30	2

```
[69] 1 y = y2.values
2 y
[2, 2, 1, 2, 3, ..., 2, 2, 1, 1, 2]
Length: 50
Categories (3, int64): [1 < 2 < 3]
```

```
[70] 1 y.shape #ต้องทำให้ shape เป็น (n_sample, ) ถึงจะนำข้อมูลไปวิเคราะห์ได้
(50,)
```



Classification



Classification

Decision Tree

```
[71] 1 from sklearn.model_selection import train_test_split  
2 #import ชึ้นชื่อมุม
```

Train - Test

```
[72] 1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=6) #การ Train-Test เป็นการแบ่งข้อมูลที่ได้มา มา train model เพื่อที่จะได้รับผลลัพธ์ที่จะนำไปใช้เครื่องห่อไป
```

```
[73] 1 X_train.shape #ขนาดข้อมูล  
0s  
(25, 4)
```

Train - Validation

```
[74] 1 X_train2, X_val, y_train2, y_val = train_test_split(X_train, y_train, test_size=0.3, random_state=6) #การ Train-Validation เป็นการแบ่งข้อมูลไว้เพื่อทดสอบเพื่อหาค่าที่ดีที่สุด validation set จะหาหน้าที่ เสมือนเป็นข้อมูลที่ไม่เคยเห็นมาก่อน แล้วใช้ทดสอบ
```

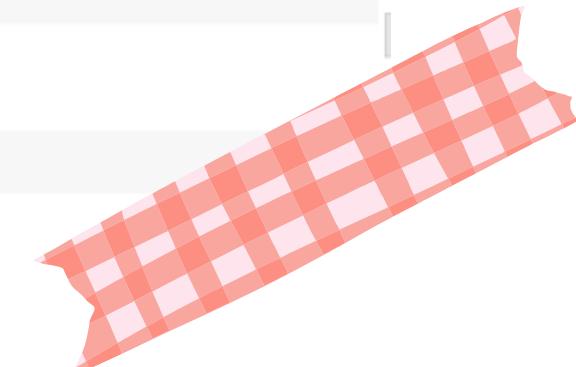
```
[75] 1 X_train2.shape  
0s  
(17, 4)
```

import

```
[76] 1 from sklearn.tree import DecisionTreeClassifier
```

define

```
[77] 1 mytree = DecisionTreeClassifier(random_state=6) #มุ่ง กำหนดrandom เผาที่ 6
```





Classification

train

```
[78] 1 mytree.fit(X_train2,y_train2) #สร้างโมเดลของข้อมูล  
  
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',  
max_depth=None, max_features=None, max_leaf_nodes=None,  
min_impurity_decrease=0.0, min_impurity_split=None,  
min_samples_leaf=1, min_samples_split=2,  
min_weight_fraction_leaf=0.0, presort='deprecated',  
random_state=6, splitter='best')
```

Evaluate

```
[79] 1 from sklearn.metrics import accuracy_score
```

Test

```
[80] 1 mytree_result = mytree.predict(X_val)
```

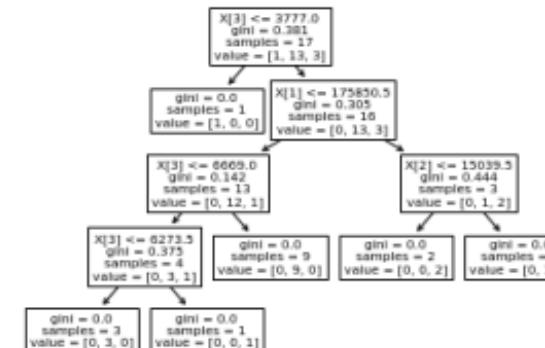
```
[81] 1 DecisionTree = accuracy_score(y_val,mytree_result)  
2 DecisionTree  
3  
4 #ค่าความแม่นยำ เท่ากับ 75%
```

0.75

plot tree

```
[82] 1 from sklearn.tree import plot_tree #import ชื่อมุก  
  
1 plot_tree(mytree) # plot แผนภูมิพื้นที่
```

[Text(148.8, 195.696, 'X[3] <= 3777.0\ngini = 0.381\nsamples = 17\nvalue = [1, 13, 3]'),
Text(111.6000000000001, 152.208, 'gini = 0.0\nsamples = 1\nvalue = [1, 0, 0]'),
Text(186.0, 152.208, 'X[1] <= 175850.5\ngini = 0.305\nsamples = 16\nvalue = [0, 13, 3]'),
Text(111.6000000000001, 108.72, 'X[3] <= 6669.0\ngini = 0.142\nsamples = 13\nvalue = [0, 12, 1]'),
Text(74.4, 65.232, 'X[3] <= 6273.5\ngini = 0.375\nsamples = 4\nvalue = [0, 3, 1]'),
Text(37.2, 21.744, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0]'),
Text(111.6000000000001, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(148.8, 65.232, 'gini = 0.0\nsamples = 9\nvalue = [0, 9, 0]'),
Text(260.4000000000003, 108.72, 'X[2] <= 15039.5\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
Text(223.2000000000002, 65.232, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
Text(297.6, 65.232, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'))



root node คือ $X[3] \leq 3777.0$

information gain = 0.381 ค่ามากสุดเป็น root node

มี data ในล็อกเกอร์ 17 ตัว

แบ่งเป็น 3 กลุ่ม คือ 1, 13, 3



Classification

KNN

```
[84] 1 from sklearn.neighbors import KNeighborsClassifier #import ฟังก์ชัน
```

KNN1

```
[85] 1 # Define  
2 neigh1 = KNeighborsClassifier(n_neighbors=3, weights='uniform') #กำหนดเพื่อแบ่งทีโกลสุด 3 คน และใช้อุปกรณ์ที่มี  
3 # Train  
4 neigh1.fit(X_train2,y_train2)  
5 # Test  
6 knn1_result = neigh1.predict(X_val)  
7 accuracy_score(y_val, knn1_result)  
8
```

0.625

KNN2

```
[86] 1 # Define  
2 neigh2 = KNeighborsClassifier(n_neighbors=7, weights='distance') #กำหนดเพื่อแบ่งทีโกลสุด 7 คน และใช้อุปกรณ์ที่อยู่ใกล้มากกว่าคนที่อยู่ไกล  
3 # Train  
4 neigh2.fit(X_train2,y_train2)  
5 # Test  
6 knn2_result = neigh2.predict(X_val)  
7 accuracy_score(y_val, knn2_result)  
8
```

0.625

KNN3

```
[87] 1 # Define  
2 neigh3 = KNeighborsClassifier(n_neighbors=1) #กำหนดเพื่อแบ่งทีโกลที่สุดคนเดียว  
3 # Train  
4 neigh3.fit(X_train2,y_train2)  
5 # Test  
6 knn3_result = neigh3.predict(X_val)  
7 accuracy_score(y_val, knn3_result)
```

0.75

Classification

Neural Network

```
[88] 1 from sklearn.neural_network import MLPClassifier  
  
[89] 1 perceptron1 = MLPClassifier(random_state=6, max_iter=2000, hidden_layer_sizes=1, learning_rate_init=0.005)  
  
Train - Test  
  
ANN1
```

```
[90] 1 # Train  
2 perceptron1.fit(X_train2,y_train2)  
3 # Test  
4 ann1_result = perceptron1.predict(X_val)  
5 accuracy_score(y_val, ann1_result)
```

0.625

ANN 2

```
[91] 1 perceptron2 = MLPClassifier(random_state=6, max_iter=2000, hidden_layer_sizes=10, learning_rate_init=0.005)  
2 # Train  
3 perceptron2.fit(X_train2,y_train2)  
4 # Test  
5 ann2_result = perceptron2.predict(X_val)  
6 accuracy_score(y_val, ann2_result)
```

0.75

ANN 3

```
[92] 1 perceptron3 = MLPClassifier(random_state=6, max_iter=20000, hidden_layer_sizes=10, learning_rate_init=0.001)  
2 # Train  
3 perceptron3.fit(X_train2,y_train2)  
4 # Test  
5 ann3_result = perceptron3.predict(X_val)  
6 accuracy_score(y_val, ann3_result)  
7
```

0.625

Retrain & Evaluate

KNN

```
[93] 1 # Define  
2 neigh3_final = KNeighborsClassifier(n_neighbors=1)  
3 # Train  
4 neigh3_final.fit(X_train,y_train)  
5 # Test  
6 knn3final_result = neigh3_final.predict(X_test)
```

```
[95] 1 KNN = accuracy_score(y_test, knn3final_result)  
2 KNN
```

0.36

Neural Network

```
[96] 1 perceptron2_final = MLPClassifier(random_state=6, max_iter=2000, hidden_layer_sizes=10, learning_rate_init=0.005)  
2 #Train  
3 perceptron2_final.fit(X_train,y_train)  
4 #Test  
5 perceptron2final_result = perceptron2_final.predict(X_test)
```

```
[97] 1 Neural = accuracy_score(y_test,perceptron2final_result)  
2 Neural
```

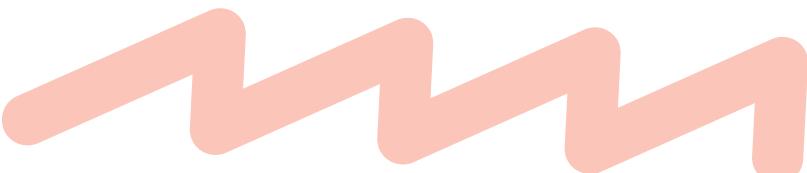
0.6

DecisionTree

```
[98] 1 DecisionTree = accuracy_score(y_val,mytree_result)  
2 DecisionTree
```

0.75

จาก Model ทั้ง 3 ชนิด พบว่า Model ที่มีความแม่นยำมากที่สุด คือ DesisionTree มีความแม่นยำ 75%



เลือกโมเดลทำนาย



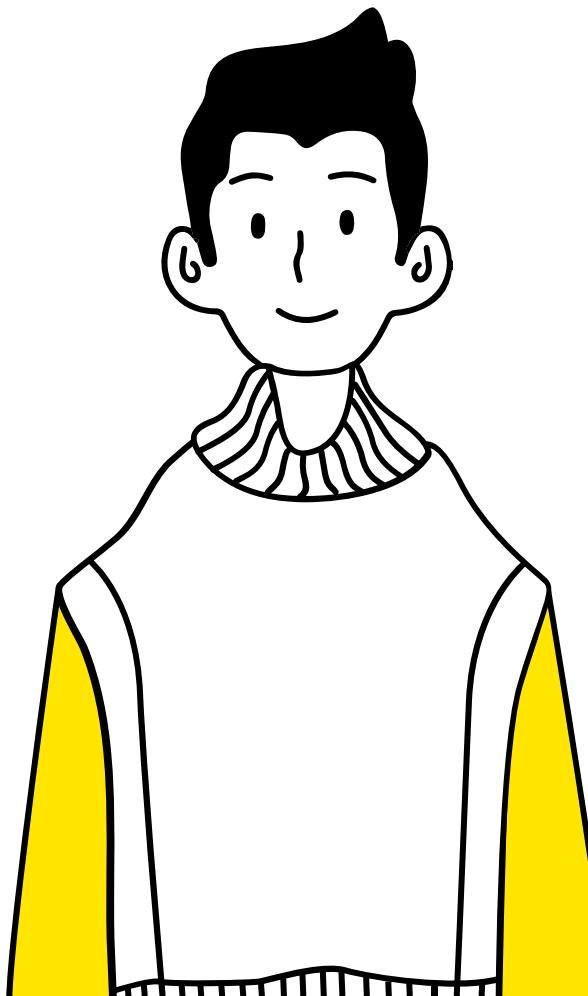
```
[105]: 1 ret = classification_report(y_val, mytree_result)
  2 print(ret)
  3
  4 #1 precision 100% recall 50% f1-score 67% support(จำนวนข้อมูลที่เป็น1) 2 ตัว
  5 #2 precision 71% recall 100% f1-score 83% support(จำนวนข้อมูลที่เป็น2) 5 ตัว
  6 #3 precision 0% recall 0% f1-score 0% support(จำนวนข้อมูลที่เป็น3) 1 ตัว
  7 #ความแม่นยำ 75%
```

	precision	recall	f1-score	support
1	1.00	0.50	0.67	2
2	0.71	1.00	0.83	5
3	0.00	0.00	0.00	1
accuracy			0.75	8
macro avg	0.57	0.50	0.50	8
weighted avg	0.70	0.75	0.69	8

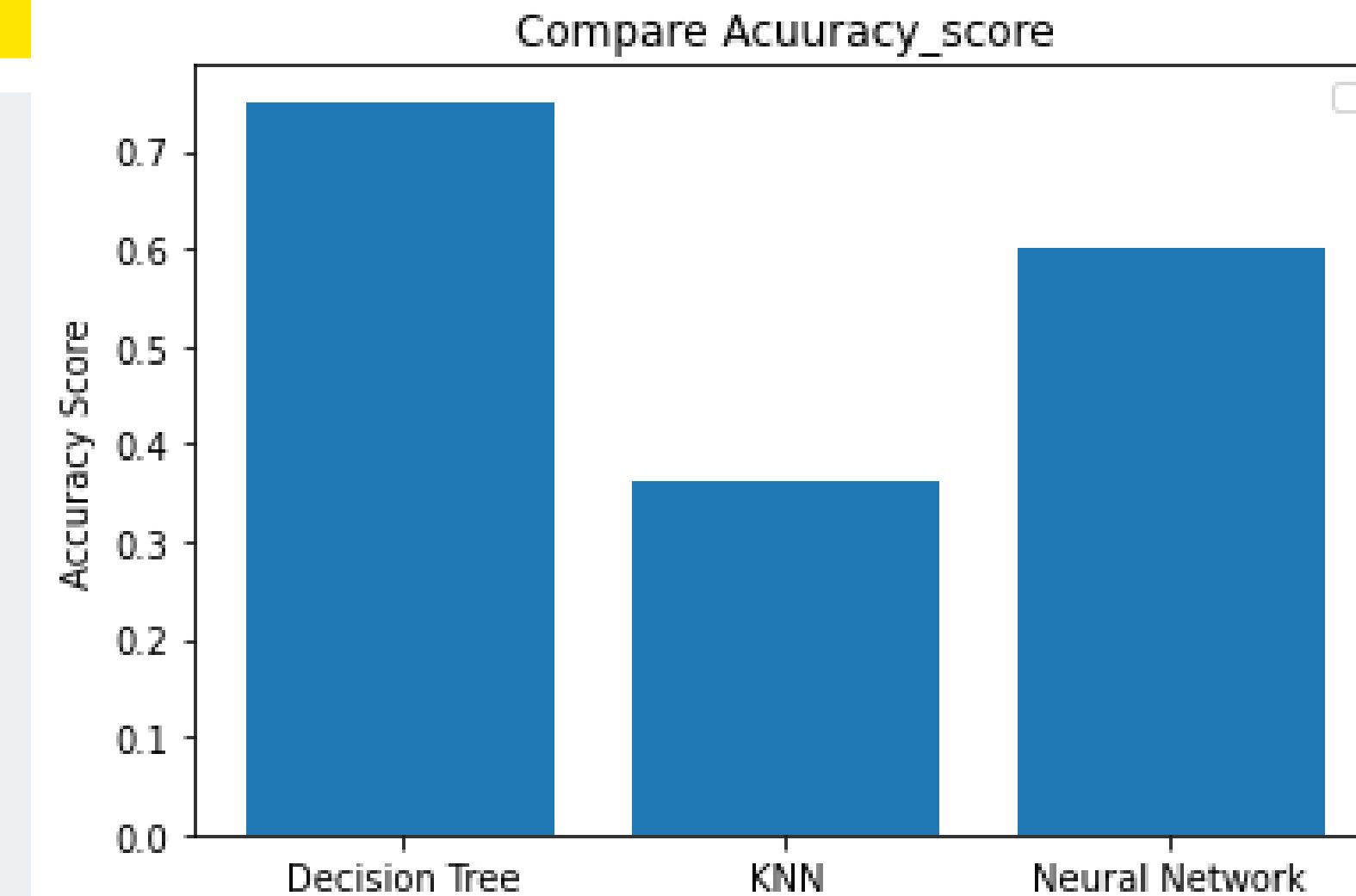
```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272:
    _warn_prf(average, modifier, msg_start, len(result))
```

Visualization

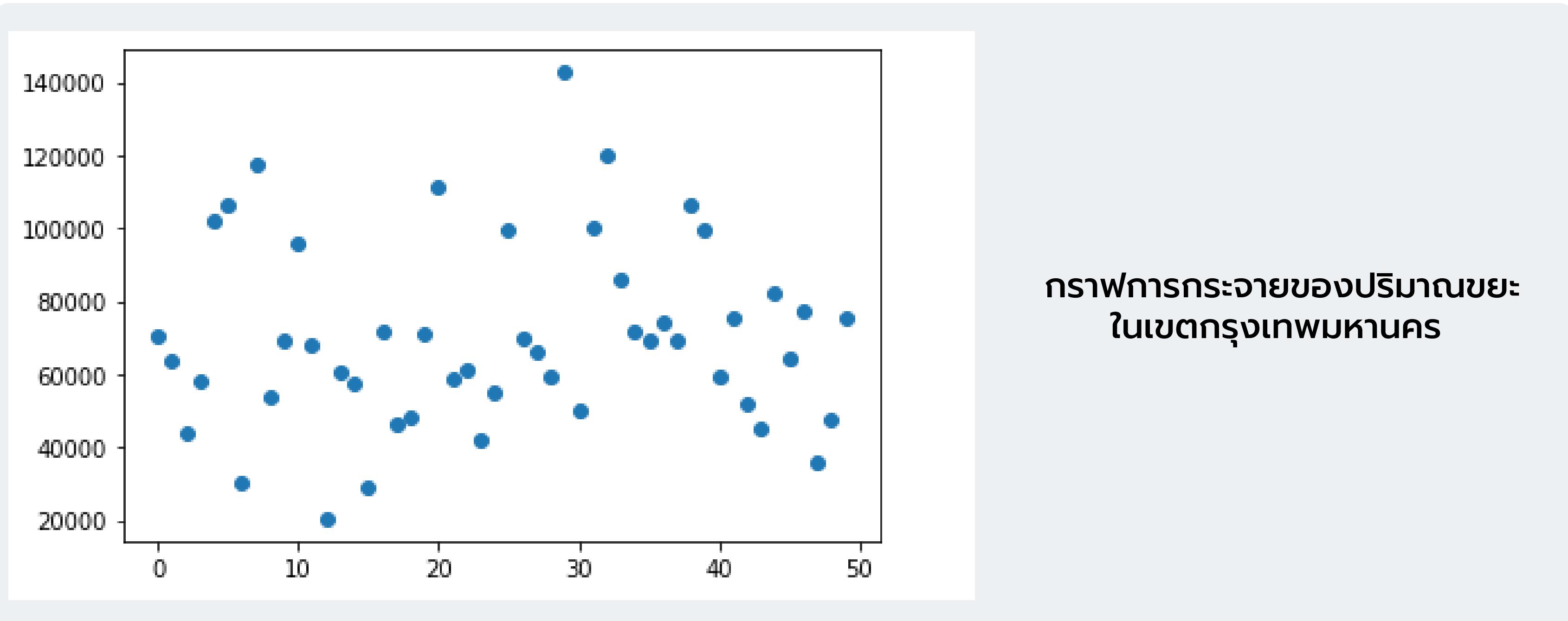
จะเห็นได้จากการเปรียบเทียบความแม่นยำพว่า accuracy_score ของ Dicision Tree มีความแม่นยามากที่สุด



เปรียบเทียบความแม่นยำ ของการทำ Classification ด้วย accuracy_score



Visualization



10

10

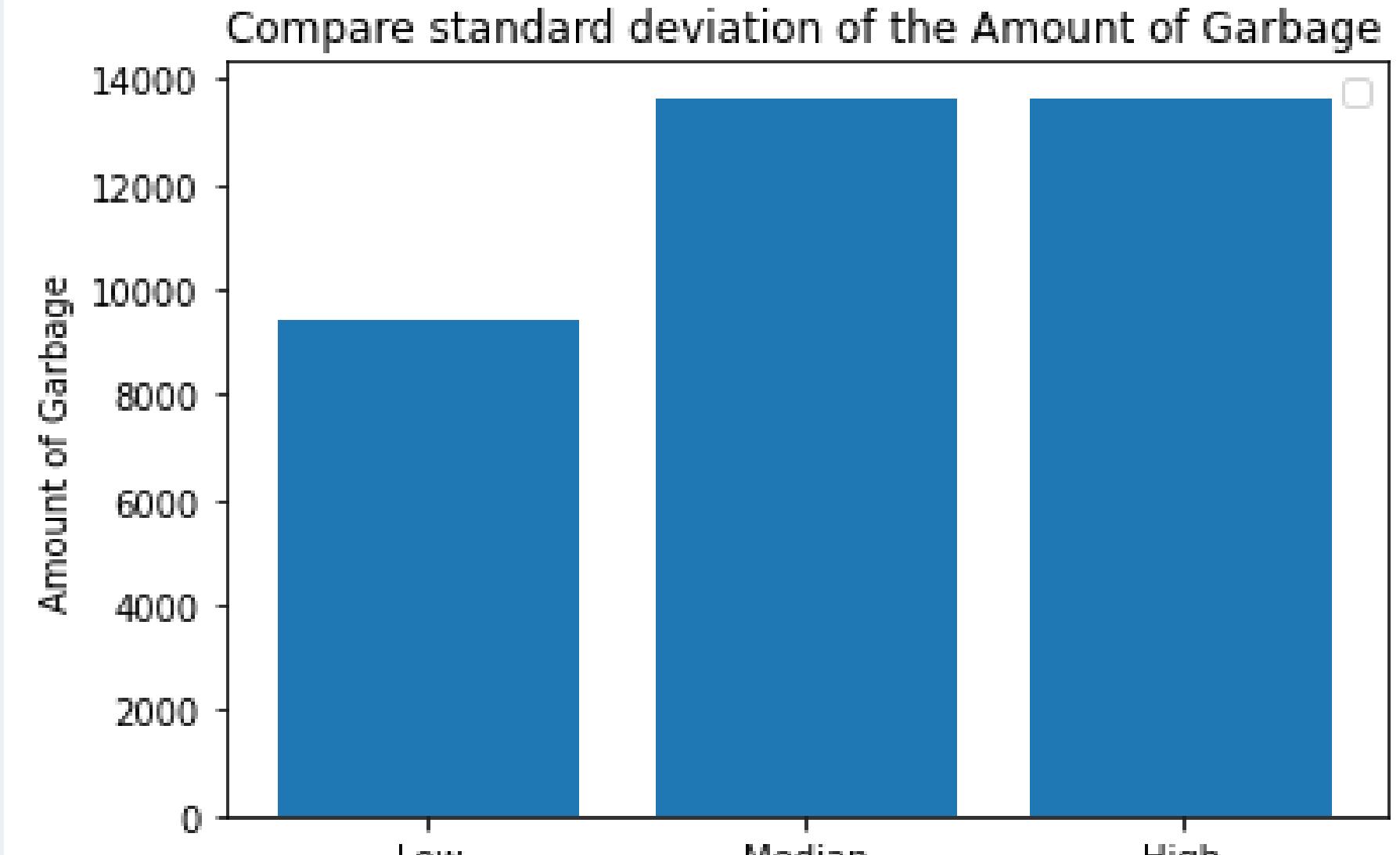
10

Visualization

จากแท่งกราฟ จะเห็นได้ว่าค่าการกระจายของปริมาณขยะปานกลางและปริมาณขยะมาก มีค่าการกระจายที่ใกล้เคียงกัน และ มีค่าการกระจายมากกว่าปริมาณขยะน้อย



เปรียบเทียบค่าการกระจายของปริมาณขยะ
ในเขตพื้นที่กรุงเทพมหานคร



จบการนำเสนอ

