



新北市五股區不動產 買賣分析



Lucturer: Janice

目錄

1. 資料介紹

2. 資料處理

3. 變數挑選

4. 決策樹

A. CART

B. C4.5

5. 關聯法則

6. SVM

7. Random Forest

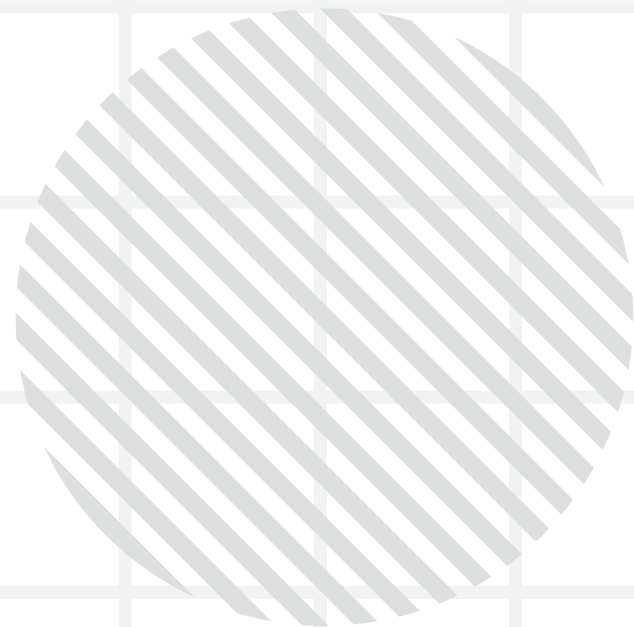
8. KNN

9. SVM/RF/KNN模型比較

10. Hierarchical Cluster

11. K-means

12. 結論



1. 資料介紹



來源

政府公開資料平台：
歷年不動產實價登錄資訊-買賣案件-111年-五股區

district	rps01	rps02	rps03	rps04	rps05	rps06	rps07	rps08	rps09	rps10	rps11	rps12	rps13	rps14	rps15	rps16	rps17	rps18	rps19	rps20	rps21	rps22	rps23	rps24	rps25	rps26	rps27	rps28	rps29	rps30	rps31
五股區	土地	更寮段更	33.06		一般農業	農牧用地	1110210	土地1建物0車位0			其他				0	0	0	0	有	無	1500000	45372		0	0		RPRVML	0	0	0	無
五股區	房地(土增)	新北市五	32.15	住			1110208	土地1建物六層	十五層	住宅大樓	住家用	鋼筋混凝土	1100716	161.79	3	2	2	2	有	有	1.6E+07	107675	坡道平面	28.24	2100000		RPUNML	79.79	0	8.06	有
五股區	房地(土增)	新北市五	20.66	住			1110217	土地1建物十五層	十五層	住宅大樓	住家用	鋼筋混凝土	1100716	113.97	2	2	1	1	有	有	1.2E+07	120145	坡道平面	28.24	2000000		RPPSML	53.14	0	3.32	有
五股區	房地(土增)	新北市五	32.15	住			1110209	土地1建物十二層	十五層	住宅大樓	住家用	鋼筋混凝土	1100716	189.9	3	2	2	2	有	有	1.9E+07	114675	坡道平面	56.48	4000000		RPRSML	79.79	0	8.06	有
五股區	房地(土增)	新北市五	35.05	住			1110216	土地1建物十三層	二十三層	住宅大樓	住家用	鋼筋混凝土	1040925	205.51	3	2	2	2	有	有	2.2E+07	111773	坡道平面	31.05	2000000		RPXNML	104.44	5.2	8.45	有
五股區	房地(土增)	新北市五	20.66	住			1110215	土地1建物十五層	十五層	住宅大樓	住家用	鋼筋混凝土	1100716	113.97	2	2	1	1	有	有	1.2E+07	115829	坡道平面	28.24	2000000		RPSSML	53.14	0	3.32	有
五股區	房地(土增)	新北市五	20.66	住			1110215	土地1建物十二層	十五層	住宅大樓	住家用	鋼筋混凝土	1100716	114.05	2	2	1	1	有	有	1.2E+07	117119	坡道平面	28.24	2000000		RPTSML	53.14	0	3.32	有
五股區	車位	新北市五	0.44	住			1110215	土地1建物地下一層	二十一層	其他	防空避難	鋼筋混凝土	830402	37.86	0	0	0	0	有	有	1400000		坡道平面	37.86	1400000	僅車位交	RPVOML	0	0	0	有
五股區	房地(土增)	新北市五	36.42	住			1110213	土地1建物四層	五層	公寓(5樓)	住家用	鋼筋混凝土	720505	106.65	3	2	2	2	有	無	5400000	50633		0	0	陽台外推	RPPNML	93.9	0	12.75	無

變數



目標變數

rps04	都市使用土地分區	變數重新命名： 農 -> A(農業區) 商 -> C(商業區) 工 -> I(工業區) 住 -> R(住宅區) 都市 -> U(都市用地)
-------	----------	--

預測變數

類別型變數

District	鄉鎮市區
rps01	交易標的
rps02	土地區段位置建物區段門牌
rps05	非都市土地使用分區
rps06	非都市土地使用編定
rps07	交易年月日
rps08	交易筆棟數
rps11	建物型態

rps12	主要用途
rps13	主要建材
rps14	建築完成年月
rps19	建物現況格局-隔間
rps20	有無管理組織
rps23	車位類別
rps26	備註
rps27	編號
rps31	電梯



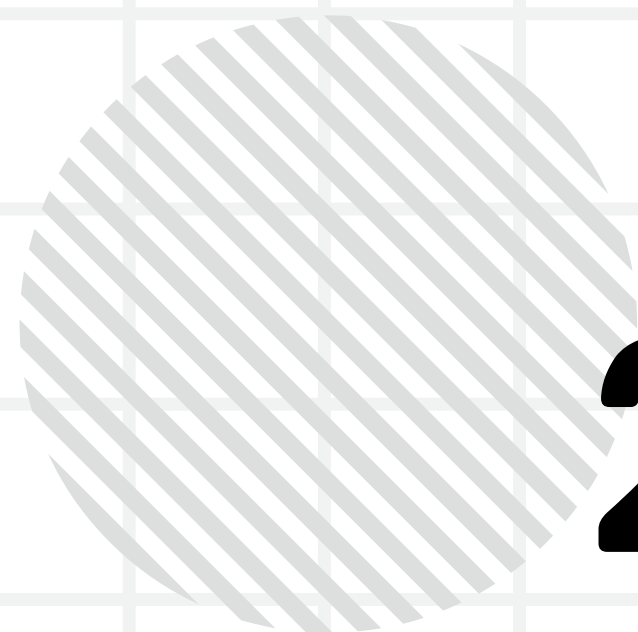
預測變數



連續型變數

rps03	土地移轉總面積平方公尺
rps09	移轉層次
rps10	總樓層數
rps15	建物移轉總面積平方公尺
rps16	建物現況格局-房
rps17	建物現況格局-廳
rps18	建物現況格局-衛

rps21	總價元
rps22	單價元平方公尺
rps24	車位移轉總面積平方公尺
rps25	車位總價元
rps28	主建物面積
rps29	附屬建物面積
rps30	陽台面積



2. 資料預處理



原始資料

0	district	1786	non-null	object
1	rps01	1786	non-null	object
2	rps02	1786	non-null	object
3	rps03	1786	non-null	float64
4	rps04	1731	non-null	object
5	rps05	36	non-null	object
6	rps06	36	non-null	object
7	rps07	1786	non-null	int64
8	rps08	1786	non-null	object
9	rps09	1565	non-null	object
10	rps10	1565	non-null	object
11	rps11	1786	non-null	object
12	rps12	1565	non-null	object
13	rps13	1565	non-null	object
14	rps14	1565	non-null	float64
15	rps15	1786	non-null	float64
16	rps16	1786	non-null	int64
17	rps17	1786	non-null	int64
18	rps18	1786	non-null	int64
19	rps19	1786	non-null	object
20	rps20	1786	non-null	object
21	rps21	1786	non-null	int64
22	rps22	1765	non-null	float64
23	rps23	1191	non-null	object
24	rps24	1786	non-null	float64
25	rps25	1786	non-null	int64
26	rps26	853	non-null	object
27	rps27	1786	non-null	object
28	rps28	1786	non-null	float64
29	rps29	1786	non-null	float64
30	rps30	1786	non-null	float64
31	rps31	1786	non-null	object

- 刪除變數
- 重新命名變數
- 轉換變數型態
- 遺失值處理
- 離散化
- 轉碼

變數處理

- 刪除變數

- 1.district(鄉鎮市區)
- 2.rps05(非都市土地使用分區)
- 3.rps06(非都市土地使用編定)
- 4.rps07(交易年月日)
- 5.rps09(移轉層次)
- 6.rps13(主要建材)
- 7.rps14(建築完成年月)
- 8.rps15(建物移轉總面積平方公尺)
- 9.rps22(單價元平方公尺)
- 10.rps23(車位類別)
- 11.rps24(車位移轉總面積平方公尺)
- 12.rps25(車位總價元)
- 13.rps26(備註)
- 14.rps27(編號)



變數處理

- 重新命名類別型變數

rps01(交易標的)	土地 -> LAND 車位 -> PARKING 房地(土地+建物) -> LAND&BUILDING 房地(土地+建物)+車位 -> LAND&BUILDING&PARKING
rps11(建物型態)	公寓 -> APARTMENT 住宅大樓 -> BUILDING 透天厝 -> HOUSE 華廈 -> MANSION 其他 -> OTHER
rps19(建物現況格局-隔間) rps20(有無管理組織) rps31(電梯)	無 -> NO 有 -> YES
rps04(都市使用土地分區)	農 -> A(農業區) 商 -> C(商業區) 工 -> I(工業區) 住 -> R(住宅區) 都市 -> U(都市用地)

變數處理

- 重新命名類別型變數

rps12(主要用途)	住家用-> 1 辦公用 -> 2 複合使用(工商、住商、住工)-> 3 工業用 -> 4 停車用 -> 5 商業用 -> 6 其他 -> 7
-------------	--

- 連續型變數轉類別型變數

rps29(附屬建物面積) rps30(陽台面積)	如果數值>0，轉為YES(有附屬建物或陽台) 如果數值=0，轉為NO(沒有附屬建物或陽台)
------------------------------	--

- 變數轉換

將rps08(交易筆棟數)轉換為land、building、parking三個連續型變數

變數處理

🔍 缺失值處理

資料中rps04、rps10、rps12，rps04和rps12為類別變數，使用眾數進行處理；rps10為連續型變數使用中位數處理

🔍 離散化

針對資料中rps21進行離散化，使用等次數進行並分為5組

🔍 轉碼

因為類別型變數皆沒有順序關係，因此使用One-Hot-Encoder進行編碼

變數處理

處理前

0	district	1786	non-null	object
1	rps01	1786	non-null	object
2	rps02	1786	non-null	object
3	rps03	1786	non-null	float64
4	rps04	1731	non-null	object
5	rps05	36	non-null	object
6	rps06	36	non-null	object
7	rps07	1786	non-null	int64
8	rps08	1786	non-null	object
9	rps09	1565	non-null	object
10	rps10	1565	non-null	object
11	rps11	1786	non-null	object
12	rps12	1565	non-null	object
13	rps13	1565	non-null	object
14	rps14	1565	non-null	float64
15	rps15	1786	non-null	float64
16	rps16	1786	non-null	int64
17	rps17	1786	non-null	int64
18	rps18	1786	non-null	int64
19	rps19	1786	non-null	object
20	rps20	1786	non-null	object
21	rps21	1786	non-null	int64
22	rps22	1765	non-null	float64
23	rps23	1191	non-null	object
24	rps24	1786	non-null	float64
25	rps25	1786	non-null	int64
26	rps26	853	non-null	object
27	rps27	1786	non-null	object
28	rps28	1786	non-null	float64
29	rps29	1786	non-null	float64
30	rps30	1786	non-null	float64
31	rps31	1786	non-null	object

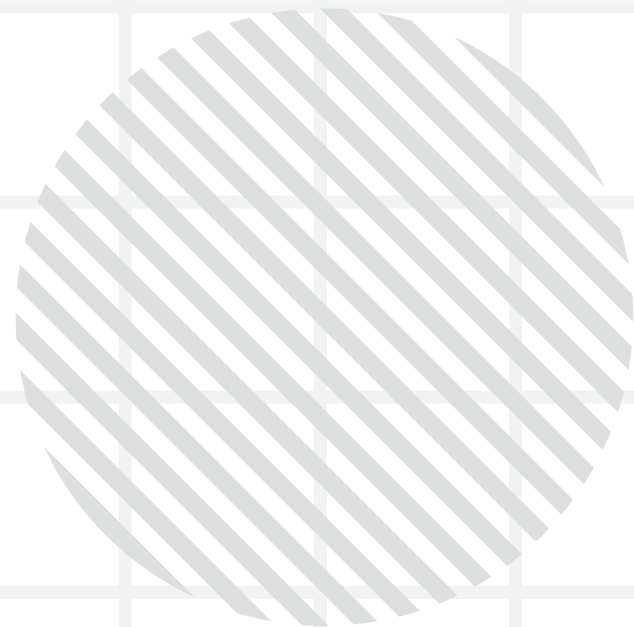
處理後

0	id	1785	non-null	int64
1	district	1785	non-null	object
2	rps01	1785	non-null	object
3	rps02	1785	non-null	object
4	rps03	1785	non-null	float64
5	rps04	1766	non-null	object
6	rps05	35	non-null	object
7	rps06	35	non-null	object
8	rps07	1785	non-null	int64
9	rps09	1565	non-null	object
10	rps10	1565	non-null	float64
11	rps11	1785	non-null	object
12	rps12	1565	non-null	float64
13	rps13	1565	non-null	object
14	rps14	1565	non-null	float64
15	rps15	1785	non-null	float64
16	rps16	1785	non-null	int64
17	rps17	1785	non-null	int64
18	rps18	1785	non-null	int64
19	rps19	1785	non-null	object
20	rps20	1785	non-null	object
21	rps21	1785	non-null	int64
22	rps22	1764	non-null	float64
23	rps23	1191	non-null	object
24	rps24	1785	non-null	float64
25	rps25	1785	non-null	int64
26	rps26	852	non-null	object
27	rps27	1785	non-null	object
28	rps28	1785	non-null	float64
29	rps29	1785	non-null	object
30	rps30	1785	non-null	object
31	rps31	1785	non-null	object
32	land	1785	non-null	int64
33	building	1785	non-null	int64
34	parking	1785	non-null	int64



資料切割

```
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(X_one,y_one,test_size=0.2,ran  
dom_state=409421)
```



3. 變數挑選



變數挑選

Chi-square test

變數選擇	['LAND' 'rps10' 'BUIDING' 'MANSION' 'OTHER' 'rps16' 'rps20_NO' 'rps28' 'rps30_NO' 'rps31_NO']
訓練正確率	77.80%
測試正確率	77.31%

Modeled feature importance

變數選擇	['LAND' 'rps10' 'MANSION' 'rps12' 'rps18' 'land']
訓練正確率	77.24%
測試正確率	76.47%

變數挑選


🔍 Chi-square test

訓練正確率	77.80%
測試正確率	77.31%

🔍 全部變數



訓練正確率	78.22%
測試正確率	76.47%



4. 決策樹 - CART



CART

🔍 參數設定

設定 $M=16\sim 25$ 、 $C=0.05\sim 0.5$ ，找尋正確率前五名的模型。

M \ C	0.05	0.1	0.15	0.2	0.25	...	0.05
	0.05	0.1	0.15	0.2	0.25	...	0.05
16	0.811625	0.782913	0.776611	0.772409	0.716387	...	0.716387
...						
23	0.810924	0.782213	0.77591	0.772409	0.716387	...	0.716387
24	0.807423	0.778711	0.772409	0.772409	0.716387		0.716387
25	0.807423	0.778711	0.772409	0.772409	0.716387		0.716387

CART(PYTHON)

🔍 模型選擇

分別根據正確率最高的前五名做超參數校調。

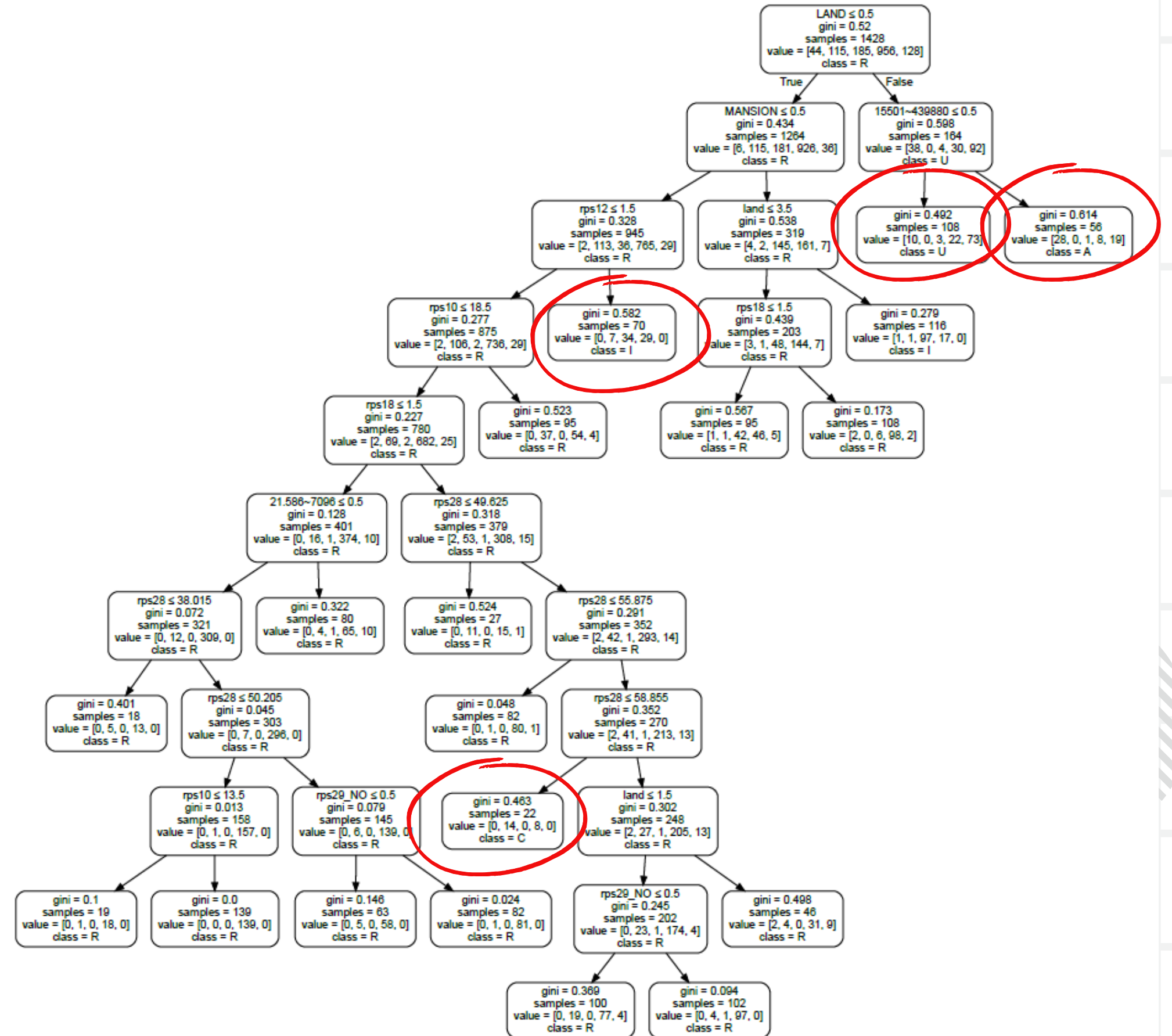
	M=16 C=0.05	M=23 C=0.05	M=24 C=0.05	M=16 C=0.1	M=23 C=0.1
訓練正確率	0.811625	0.810924	0.807423	0.782913	0.782213
測試正確率	0.803922	0.803922	0.792717	0.77591	0.77591
葉子個數	33	31	31	19	18
模型深度	12	12	12	10	10

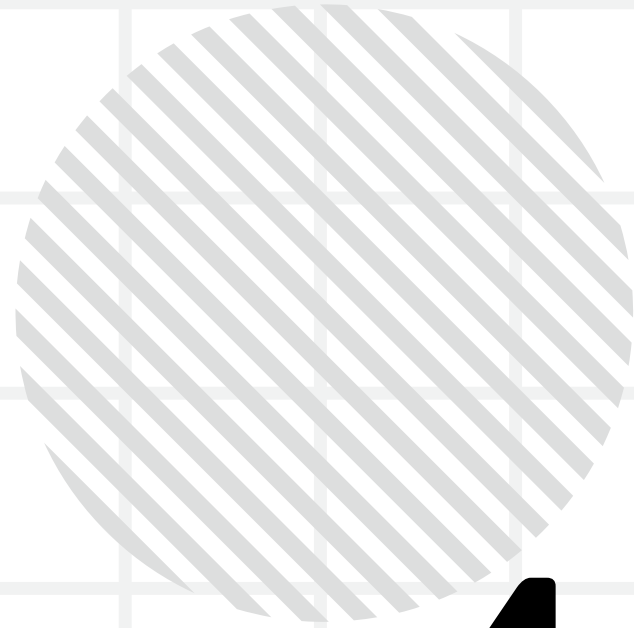
CART



決策樹

用graviz將決策樹畫出來，
可以看到畫出來的結果大部分皆為R，
我們推測是因為R為眾數，
但還是可以看到有幾條為其他項。
(圈出來的幾項分別為A、I、C、U)





4. 決策樹 - C4.5



C4.5


🔍 參數設定

因為模型在 $M=2\sim 15$ 時，複雜度過高，因此設定 $M=16\sim 25$ ， $R=F$ ，找尋正確率最高的模型。在 $M=22$ 時，正確率為0.8039216最高。

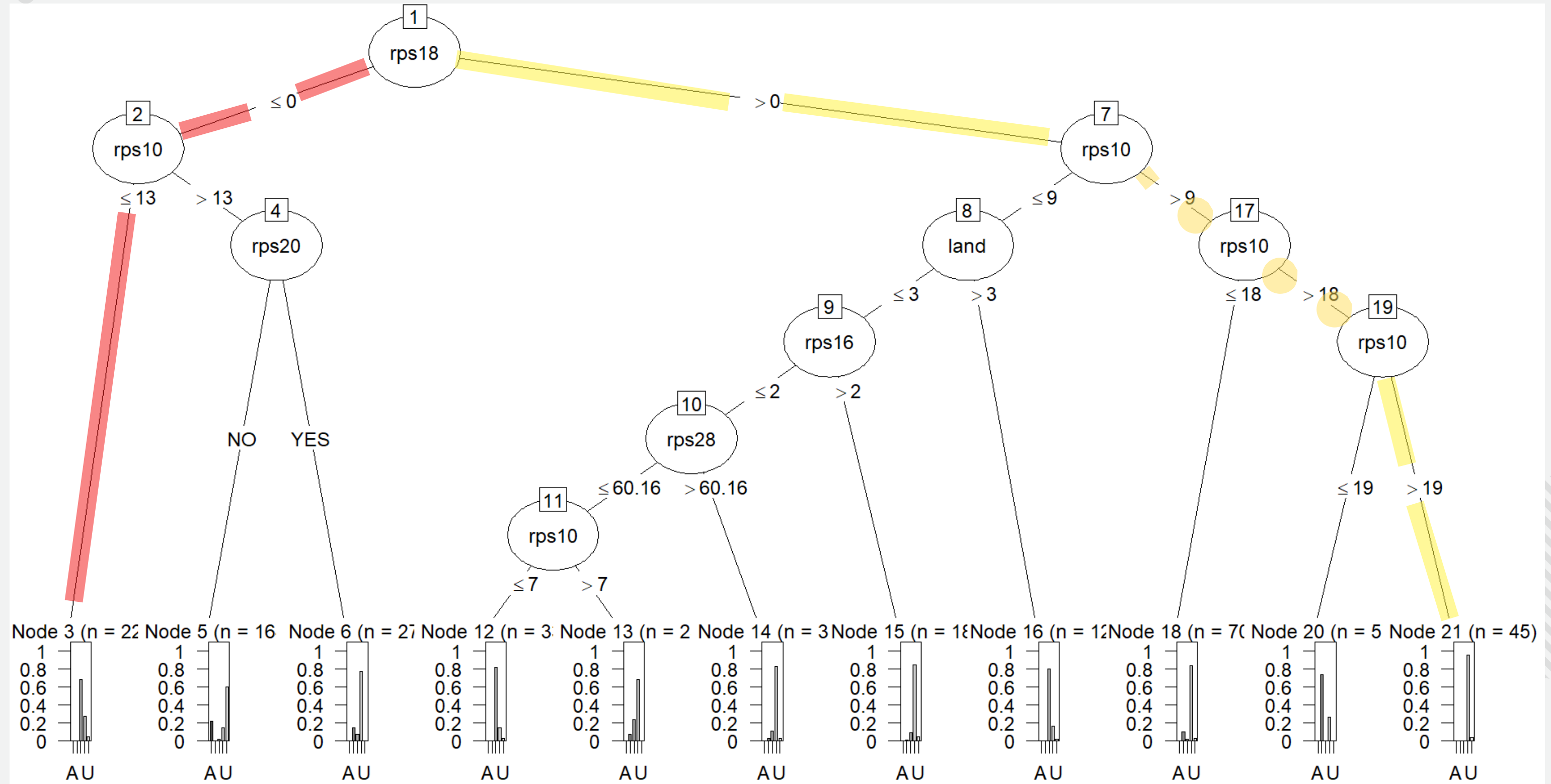
M	16	17	18	19	20	21	22	23	24	25
正確率	0.7955 182	0.80322 13	0.8032 213	0.8032 213	0.8032 213	0.8032 213	0.8039 216	0.7962 185	0.7962 185	0.7962 185

CART與C4.5比較

🔍 針對全變數/部分變數、建模/測試正確率

		C4.5(RWEKA)	CART(Python)
全變數	建模正確率	 80.39%	78.22%
	測試正確率	76.19%	76.47%
部分變數	建模正確率	80.18%	77.80%
	測試正確率	77.31%	77.31%

決策樹



決策樹

🔍 紅線

當 $rps18$ (建物現況格局-衛) ≤ 0 時，又
 $rps10$ (總樓層數) ≤ 13
➡ 土地分區為I(工業區)

🔍 黃線

當 $rps18$ (建物現況格局-衛) > 0 時，又 $rps10$ (總樓層數) > 19
➡ 土地分區為R(住宅區)

```
rps18 <= 0
|   rps10 <= 13: I (22.0/7.0)
|   rps10 > 13
|   |   rps20 = NO: U (164.0/65.0)
|   |   rps20 = YES: R (27.0/6.0)
rps18 > 0
|   rps10 <= 9
|   |   land <= 3
|   |   |   rps16 <= 2
|   |   |   |   rps28 <= 60.16
|   |   |   |   |   rps10 <= 7: I (33.0/6.0)
|   |   |   |   |   rps10 > 7: R (25.0/8.0)
|   |   |   |   |   rps28 > 60.16: R (35.0/6.0)
|   |   |   |   rps16 > 2: R (185.0/29.0)
|   |   |   land > 3: I (128.0/25.0)
|   |   rps10 > 9
|   |   |   rps10 <= 18: R (707.0/111.0)
|   |   |   |   rps10 > 18
|   |   |   |   |   rps10 <= 19: C (57.0/15.0)
|   |   |   |   |   rps10 > 19: R (45.0/2.0)
```

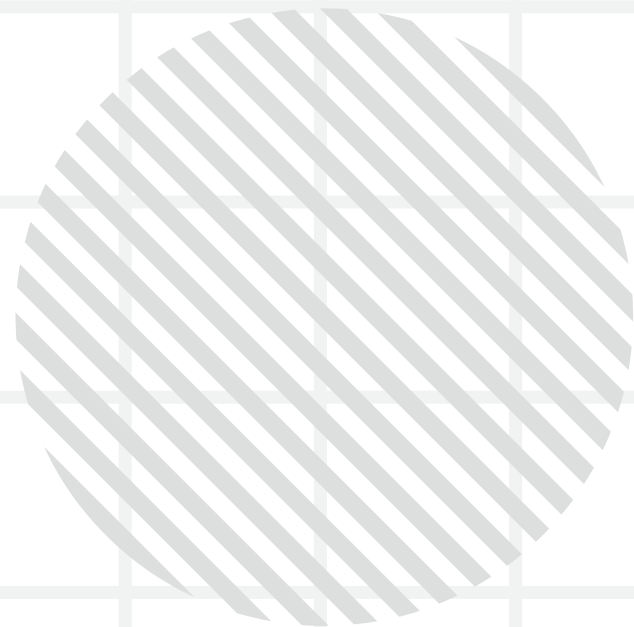
樂觀法則/悲觀法則

🔍 樂觀法則正確率

0.8039

🔍 悲觀法則正確率

$$1 - ((1785 * (1 - 0.8039) + 11 * 0.5) / 1785) = 0.8008$$



5.關聯法則



關聯法則

support : 0.05
confidence : 0.6

R

PYTHON

Excel 尋找及取代 尋找 {rps04=A}

搜尋範圍: 工... 大小寫須相符
儲存格內容須完全相符
全半形須相符

搜尋: 循序 查詢: 公式

全部尋找 上一個

Microsoft Excel 找不到您要搜尋的資料。

若您確定現行工作表中有您所要的資料，請檢查您鍵入的內容並再試一次。

確定

Excel 尋找及取代 尋找 frozenset({'A'})

搜尋範圍: 工... 大小寫須相符
儲存格內容須完全相符
全半形須相符

搜尋: 循序 查詢: 公式

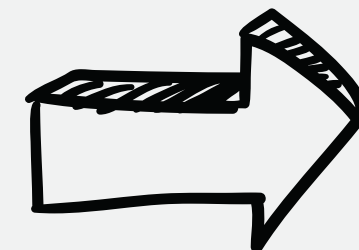
全部尋找 上一個 下一步 關閉

Microsoft Excel 找不到您要搜尋的資料。

若您確定現行工作表中有您所要的資料，請檢查您鍵入的內容並再試一次。

確定

沒有找到rps04=A的結果



重抽樣

關聯法則 R

support : 0.05
confidence : 0.6
共120條法則

重抽樣：使用undersampling，配合最少數rps04=A

rps04=A,rps04=C,rps04=I,rps04=R,rps04=U都各抽52筆

rules	support	confidence	coverage	lift	count
{rps11=MANSION} => {rps04=I}	0.13461538	0.71428571	0.18846154	3.57142857	35
{rps10=2} => {rps04=I}	0.13461538	0.66037736	0.20384615	3.30188679	35
{rps01=LAND,rps21=5} => {rps04=A}	0.11923077	0.75609756	0.15769231	3.7804878	31
{rps11=OTHER,rps21=5} => {rps04=A}	0.11923077	0.75609756	0.15769231	3.7804878	31
{rps21=5,rps30=NO} => {rps04=A}	0.11923077	0.68888889	0.17307692	3.44444444	31
{rps18=[0,1],rps21=5} => {rps04=A}	0.11923077	0.64583333	0.18461538	3.22916667	31
{rps21=5,rps31=NO} => {rps04=A}	0.11923077	0.67391304	0.17692308	3.36956522	31
{rps20=NO,rps21=5} => {rps04=A}	0.11923077	0.65957447	0.18076923	3.29787234	31
{rps16=[0,2],rps21=5} => {rps04=A}	0.11923077	0.63265306	0.18846154	3.16326531	31
{rps17=[0,2],rps21=5} => {rps04=A}	0.11923077	0.64583333	0.18461538	3.22916667	31
{rps01=LAND&BUILDING&PARKING,rps11=BUIDING,rps17=[2,4],rps29=YES} => {rps04=C}	0.11538462	0.6	0.19230769	3	30
{rps10=3,rps21=5,rps29=NO} => {rps04=A}	0.11153846	0.6744186	0.16538462	3.37209302	29
{rps12=1,rps21=5,rps29=NO} => {rps04=A}	0.11153846	0.64444444	0.17307692	3.22222222	29
{rps19=YES,rps21=5,rps29=NO} => {rps04=A}	0.11153846	0.63043478	0.17692308	3.15217391	29
{rps01=LAND,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps11=OTHER,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps10=3,rps30=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps18=[0,1],rps19=YES,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps10=3,rps31=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps10=3,rps20=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps12=1,rps18=[0,1],rps30=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps12=1,rps16=[0,2],rps30=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps16=[0,2],rps19=YES,rps30=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps12=1,rps17=[0,2],rps30=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps17=[0,2],rps19=YES,rps30=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps12=1,rps18=[0,1],rps31=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27
{rps12=1,rps18=[0,1],rps20=NO,land=[1,2]} => {rps04=A}	0.10384615	0.6	0.17307692	3	27

關聯法則 R

support : 0.05
confidence : 0.6

以support由大到小排序，選出五個不同類別的法則

	rules	support	confidence	coverage	lift	count
1	{rps11=MANSION} => {rps04=I}	0.134615385	0.71428571	0.18846154	3.57142857	35
26	{rps01=LAND,rps21=5} => {rps04=A}	0.119230769	0.75609756	0.15769231	3.7804878	31
2523	{rps01=LAND&BUILDING&PARKING,rps11=BUIDING,rps17=[2,4],rps29=YES} => {rps04=C}	0.115384615	0.6	0.19230769	3	30
422	{rps12=1,rps21=1,rps31=NO} => {rps04=U}	0.084615385	0.61111111	0.13846154	3.05555556	22
7	{rps21=3,parking=[1,2]} => {rps04=R}	0.076923077	0.60606061	0.12692308	3.03030303	20

rps01:交易標的

rps11:建物型態

rps12:主要用途

rps21:總價元

rps29_yes:有附屬建物或陽台

rps31:有無電梯

rps04:都市使用土地分區

工 -> I(工業區)

農 -> A(農業區)

商 -> C(商業區)

都市 -> U(都市用地)

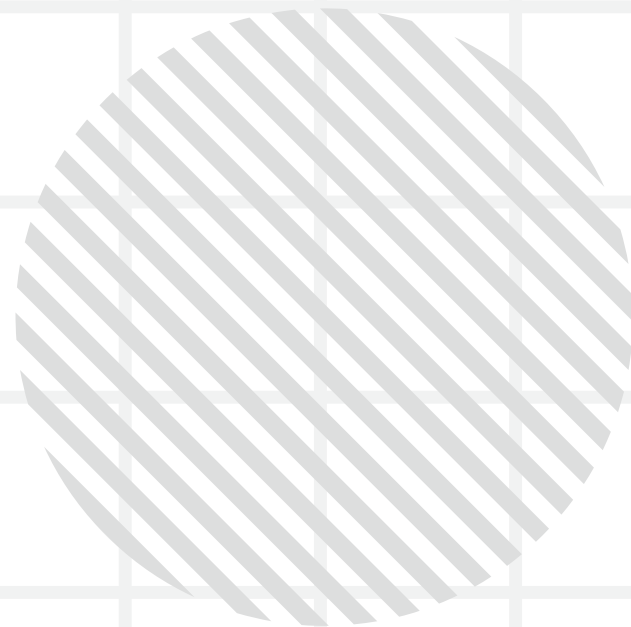
住 -> R(住宅區)

關聯法則 PYTHON

support : 0.05
confidence : 0.6

以support由大到小排序，選出五個不同類別的法則

	antecedents	consequents	support	confidence	lift	leverage
1	frozenset({ 'MANSION' })	frozenset({'I'})	0.13846154	0.69230769	3.46153846	0.098461538
2	frozenset({'rps21:15501~439880', 'LAND' })	frozenset({'A'})	0.11923077	0.86111111	4.30555556	0.091538462
3	frozenset({'rps17', 'rps29_YES', 'BUIDING', 'LAND&BUILDING&PARKING' })	frozenset({'C'})	0.11153846	0.60416667	3.02083333	0.074615385
4	frozenset({'rps21:21.586~7096', 'OTHER', 'rps12', 'land' })	frozenset({'U'})	0.08461538	0.61111111	3.05555556	0.056923077
5	frozenset({'parking', 'rps21:9901~11880' })	frozenset({'R'})	0.07307692	0.65517241	1.73821253	0.031035503



6. SVM



SVM 參數值設定



🔍 核函數 (Kernel) : rbf = 高斯

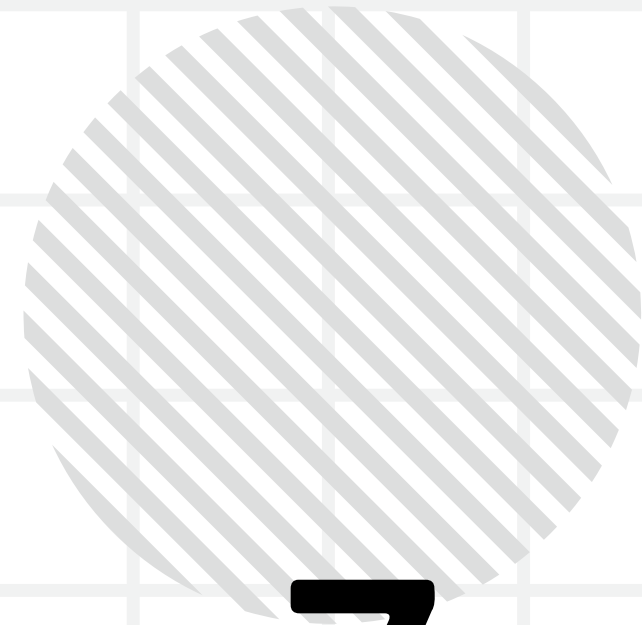
🔍 gamma 值控制每個訓練樣本的影響範圍

- gamma 值小，具較大的影響範圍，決策邊界較平滑，適用於具有較多特徵的數據集或具有較簡單的決策邊界的問題
- gamma 值大，具較小的影響範圍，決策邊界可能會更關注個別的數據點，適用於具有較少特徵的數據集或具有更複雜的決策邊界的問題

SVM 參數值設定



參數值 gamma	0.1	0.5	1	10
訓練正確率	82.42	86.2	88.66	94.61
測試正確率	78.99	71.71	71.99	66.95
是否過度配飾 (>10%)	否	是	是	是



7. Random Forest



RandomForest 參數值設定

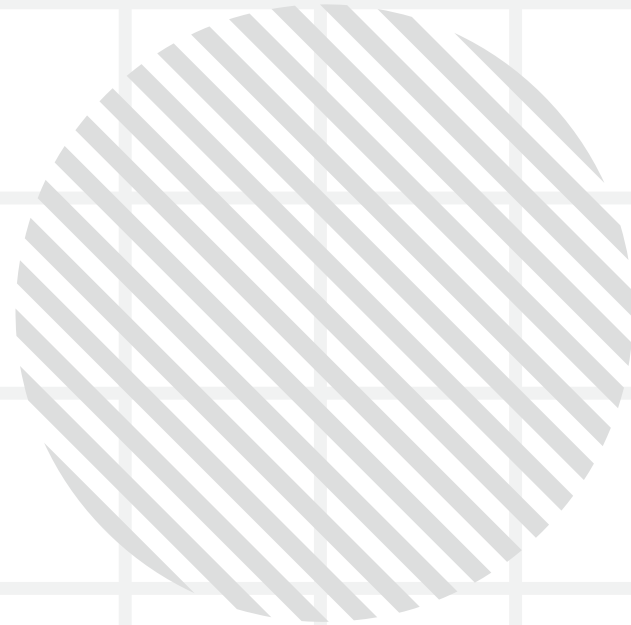


- 🔍 模型過度擬合，減小max_depth，限制樹木的深度
- 🔍 模型的方差較高且準確性不夠，增加n_estimators，增加樹木的數量
- 🔍 資料集較小，限制n_estimators的數量，以避免過度擬合
- 🔍 資料集較大且複雜，增加max_depth和n_estimators，以更好地擬合訓練數據

RandomForest自訂參數值



n_estimators	100	100	500	500
max_depth	5	10	5	10
訓練正確率	80.60	92.02	80.53	92.16
測試正確率	75.91	80.95	76.19	80.95
是否過度配飾 (>10%)	否	是	否	是



8. KNN



找出BEST K及最高正確率



```
175 #knn
176 from sklearn.neighbors import KNeighborsClassifier
177 acc=[]
178 for i in range(2,1429):
179     knn=KNeighborsClassifier(n_neighbors=i)
180     knn.fit(X_std_train,y_train)
181     acc.append(knn.score(X_std_train,y_train))
182 #print(acc)
183 print("max(acc)=",max(acc))
184 print("bestK=",acc.index(max(acc))+2)
```

用KNeighborsClassifier套件
找到最高正確率=0.899
best K =2

```
max(acc)= 0.8998599439775911
bestK= 2
```


找出BEST K



```
175 #knn
176 from sklearn.neighbors import KNeighborsClassifier
177 acc=[]
178 for i in range(2,1429):
179     knn=KNeighborsClassifier(n_neighbors=i)
180     knn.fit(X_std_train,y_train)
181     acc.append(knn.score(X_std_train,y_train))
182 #print(acc)
183 print("max(acc)",max(acc))
184 print("bestK=",acc.index(max(acc))+2)
185
186 clf9=KNeighborsClassifier(n_neighbors=2)#給定參數，knn的結果
187 clf9.fit(X_std_train,y_train)
188 print("KNN訓練資料集的正確率=",clf9.score(X_std_train,y_train))#0.8999
```

使用best K 計算訓練資料集的
正確率=89.99%

KNN訓練資料集的正確率= 0.8998599439775911



9. 各模型之訓練資料集 正確率比較



Voting 模型



hard voting 正確率=82.77%

```
from sklearn.ensemble import VotingClassifier
#voting="hard"=硬投票
clf10=VotingClassifier(estimators=[("SVM",clf1),("RF",clf5),("KNN",clf9)],voting="hard",n_jobs=-1)
clf10.fit(X_std_train,y_train)
print("hard")
print("Voting 訓練資料集的正確率=",clf10.score(X_std_train,y_train))
```

soft voting 正確率=85.5%

```
clf11=VotingClassifier(estimators=[("SVM",clf1),("RF",clf5),("KNN",clf9)],voting="soft",n_jobs=-1)
clf11.fit(X_std_train,y_train)
print("soft")
print("Voting 訓練資料集的正確率=",clf11.score(X_std_train,y_train))
```

五種訓練資料集正確率比較

SVM	82.42%
RandomForest	80.60%
KNN	89.99%
Hard Voting	82.77%
Soft Voting	85.50%





10. HIERARCHICAL CLUSTER



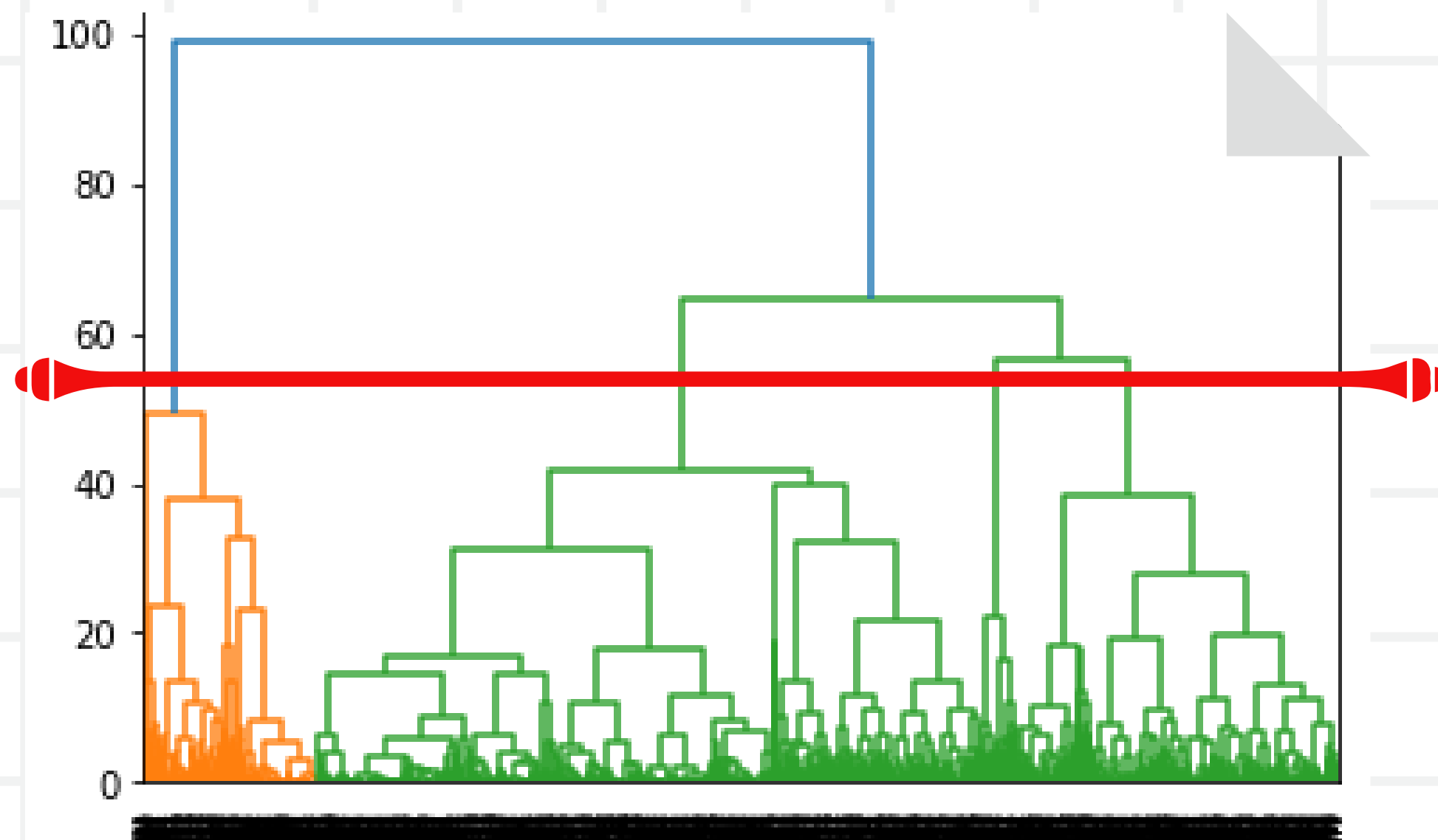
方法1：建議分幾群

```
#階層式  
#方法1: 直接告訴我們要分幾群  
from sklearn.cluster import AgglomerativeClustering  
HC=AgglomerativeClustering(n_clusters=None,affinity="euclidean",linkage="ward",distance_threshold=15)  
HC.fit(X)  
print(HC.n_clusters_) # 要分幾群
```

n_clusters=None
affinity='euclidean'
linkage='ward'
distance_threshold=15

➡ K = 27

方法2：用SCIPY畫圖



metric='euclidean'
method='ward'

➡ K = 4

SSE & 正確率



$K = 4$
正確率 = 70.31%
SSE = 13335.9230

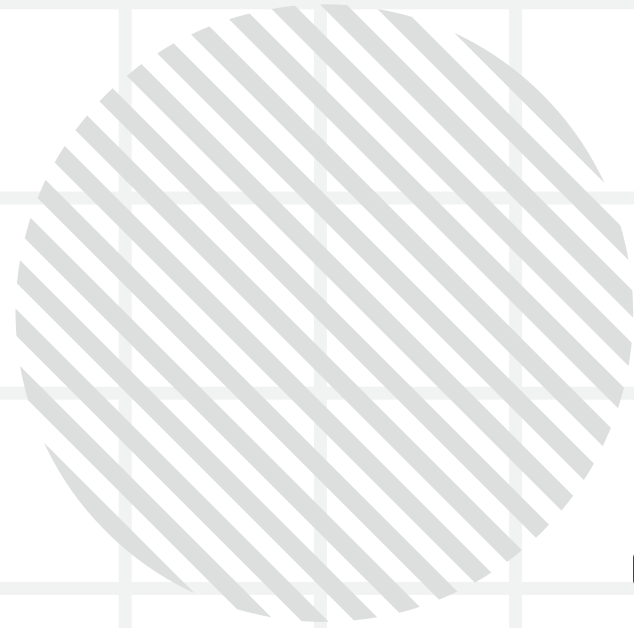
```
col_0    0    1    2    3
rps04
A         6   46    0    0
C        13    1  134    3
I       185    8    9   31
R       324   55  782   26
U        19  118   23    2
正確率= 0.7030812324929971
SSE 13335.923046672247
```

資料落點



```
# 預測每筆資料在哪一群
import numpy as np
y_pred=hc.fit_predict(X)
for i in range(4):
    print("第",i+1,"群有",np.sum(y_pred==i))
```

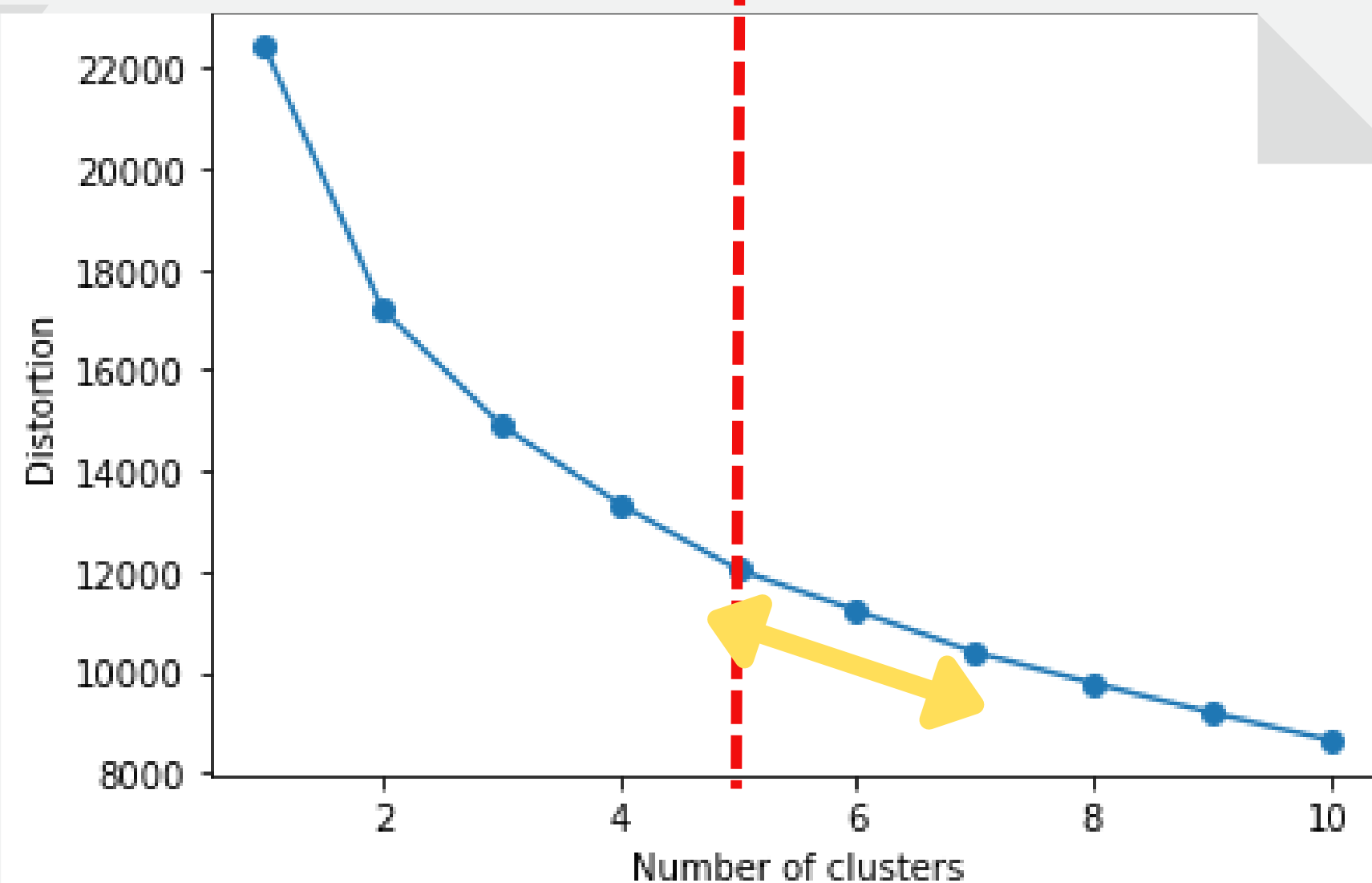
第 1 群有	259
第 2 群有	482
第 3 群有	992
第 4 群有	52



1 1. K-MEANS



K = ?



`init="k-means++"`

`n_init=15`

`max_iter=200`

⇒ K=5 已趨於平緩

SSE & 正确率



$K = 5$

正确率 = 70.42%

SSE = 12031.5273

```
col_0      0      1      2      3      4
rps04
A          6       0       0      45       1
C          13      134       3       1       0
I          184       8      30       8       3
R          323      783      26      55       0
U          19       23       2     118       0
正确率= 0.704201680672269
SSE 12031.527289379781
```



各類別輪廓

	rps01 交易標的	rps08 交易筆數	rps10 總樓層數	rps11 建物型態	rps12 主要用途	rps16 房間數	rps17 客廳數	rps18 衛浴數	rps19 隔間	rps20 管理組織	rps21 總價元	rps28 主建物面積	rps29 附屬建物	rps30 陽台	rps31 電梯	目標變數 rps04 都市土地使用分區
1	Land&Building	土地:多 房子:多 車位:少	低	華夏	住家用	多	多	多	YES	YES	便宜	中	YES	YES	YES	住
2	Land&Building&Parking	土地:少 房子:多 車位:多	高	住宅大樓	住家用	多	多	多	YES	YES	便宜	中	YES	YES	YES	住

	rps01交易標的	rps08交易筆數	rps10總樓層數	rps11建物型態	rps12主要用途	rps16房間數	rps17客廳數	rps18衛浴數	rps19隔間	rps20管理組織	rps21總價元	rps28主建物面積	rps29附屬建物	rps30陽台	rps31電梯	目標變數 rps04都市土地使用分區
3	Land&Building	土地:很少 房子:多 車位:少	低	公寓	其他	少	少	少	YES	NO	中	中	NO	YES	YES	工業
4	Land	土地:多 房子:很少 車位:少	高	其他	住家用	很少	很少	很少	YES	YES	中	小	NO	NO	NO	其他
5	Land&Building	土地:少 房子:很少 車位:少	低	透天	辦公室或工業用	很少	很少	很少	NO	NO	貴	大	NO	NO	NO	工業

比較兩者的SSE



HIERARCHICAL CLUSTER

$K = 4$

正確率 = 70.31%

SSE = 13335.9230



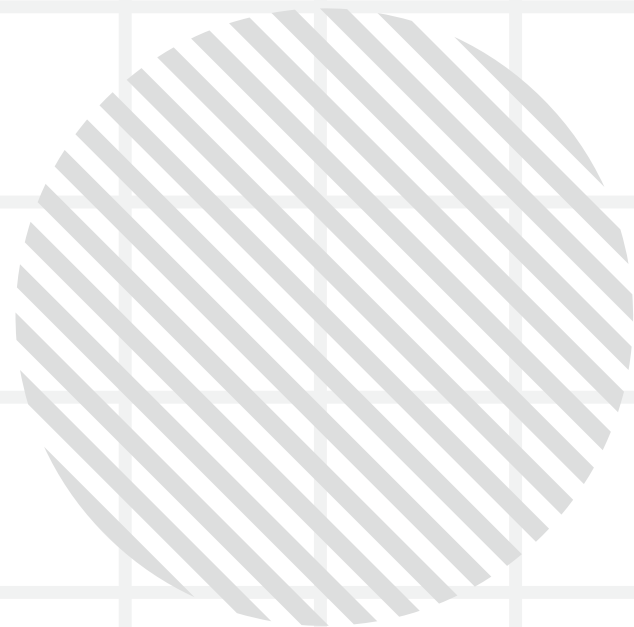
K-MEANS

$K = 5$

正確率 = 70.42%

SSE = 12031.5272





12. 結論



全變數

在所有預測模型中，
Soft Voting擁有最佳測試正確率80.11%



預測模型	測試正確率
C4.5(RWEKA)	76.19%
CART(Python)	76.47%
SVM	78.99%
Random Forest	75.91%
KNN	71.43%
HardVoting	79.27%
Soft Voting	80.11%
階層式	70.31%
K-means	70.42%

結論

🔍 重要變數

- rps01 (交易標的)中的LAND
- rps10 (總樓層數)
- rps11 (建物型態)中的MANSION

🔍 最佳模型

- 這份資料的最適模型為Soft Voting，訓練正確率為85.5%、測試正確率為80.11%。
- 可以根據交易坪數、是否有管理組織、建物型態...等，去預測所在地區為哪種都市使用土地。



THANK YOU

