

การทำนายภาพโฆษณาบนเว็บไซต์

Internet advertisements

บทนำ

หลักการและเหตุผล

เนื่องจากปัจจุบันในหน้าเว็บไซต์หนึ่งมีโฆษณาแฝงอยู่จำนวนมาก เราจึงอยากจำแนกโฆษณาออกจากหน้าเว็บไซต์ เพื่อนำไปใช้ประโยชน์ต่อไปได้ในอนาคต เช่น พัฒนาแอปพลิเคชันการลบโฆษณาออกจากหน้าเว็บไซต์ เป็นต้น

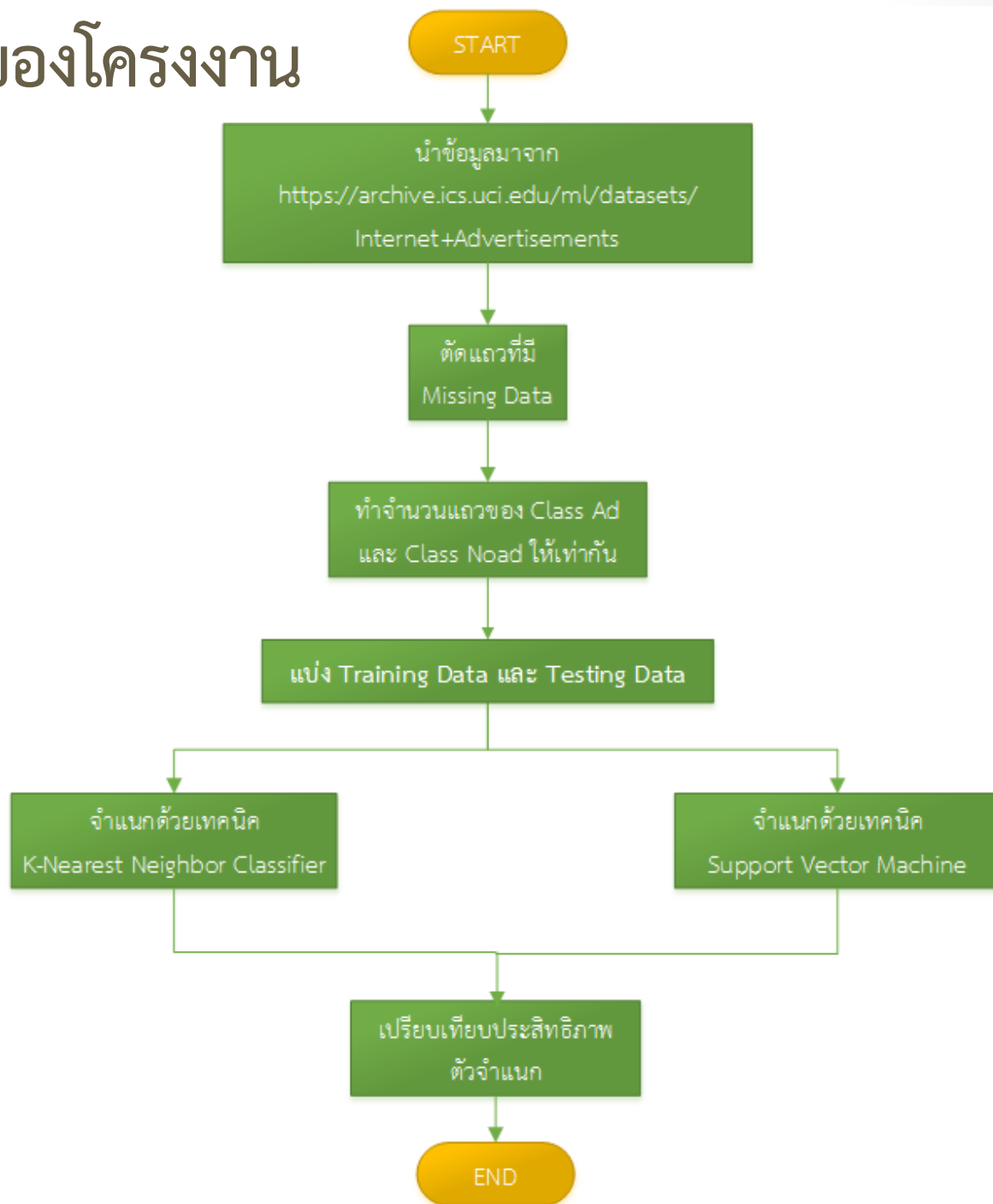
วัตถุประสงค์

- สามารถนำข้อมูลมาจำแนกคาสได้อย่างถูกต้อง
- สามารถวัดประสิทธิภาพตัวจำแนกได้
- สามารถแสดงผลลัพธ์ที่ทำนายผิดได้

ขอบเขตโครงการ

- จำนวน Attribute 1558 ตัว จำนวนหน้าเว็บไซต์ที่นำมาทดสอบ 3279 ตัว
- จำนวน Class ที่มีโฆษณาจำนวน 458 ตัว และไม่มีโฆษณาจำนวน 2821 ตัว

ภาพรวมของโครงการ



ข้อมูล

	A	B	C	D	E	F	G	BGY
1	height	width.	aratio	local	url*images+buttons	url*likesbooks.com	url*www.slake.com	ad.
2	125	125	1	1	0	0	0	ad.
3	57	468	8.2105	1	0	0	0	ad.
4	33	230	6.9696	1	0	0	0	ad.
5	60	468	7.8	1	0	0	0	ad.
6	60	468	7.8	1	0	0	0	ad.
7	60	468	7.8	1	0	0	0	ad.
8	59	460	7.7966	1	0	0	0	ad.
9	60	234	3.9	1	0	0	0	ad.
10	60	468	7.8	1	0	0	0	ad.
11	60	468	7.8	1	0	0	0	nonad.
12	?	?	?	1	0	0	0	nonad.
13	90	52	0.5777	1	0	0	0	nonad.
14	90	60	0.6666	1	0	0	0	nonad.
15	90	60	0.6666	1	0	0	0	nonad.
16	33	230	6.9696	1	0	0	0	nonad.
17	60	468	7.8	1	0	0	0	nonad.
18	60	468	7.8	0	0	0	0	nonad.
19	125	125	1	1	0	0	0	nonad.
20	60	468	7.8	1	0	0	0	nonad.

ตัดแถวที่มี Missing Data

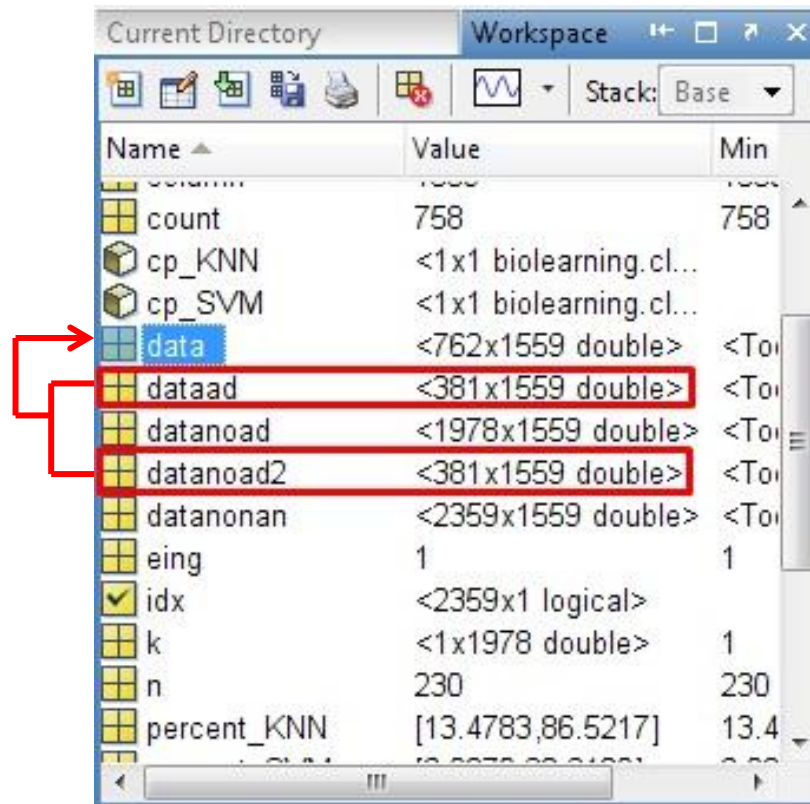
	1	2	3	4	5
61	60	234	3.9000	1	0
62	60	468	7.8000	0	0
63	NaN	NaN	NaN	0	0
64	NaN	NaN	NaN	0	0
65	NaN	NaN	NaN	0	0
66	NaN	NaN	NaN	0	0
67	60	468	7.8000	1	0
68	2	2	1	1	0
69	60	120	2	1	0
70	65	125	1.9230	0	0
71	125	125	1	0	0
72	125	125	1	0	0
73	NaN	NaN	NaN	0	0
74	60	468	7.8000	1	0
75	60	468	7.8000	0	0
76	60	468	7.8000	1	0
77	60	234	3.9000	1	0
78	44	127	2.8863	1	0
79	60	468	7.8000	1	0
80	80	80	1	1	0



	1	2	3	4	5
61	60	468	7.8000	1	0
62	2	2	1	1	0
63	60	120	2	1	0
64	65	125	1.9230	0	0
65	125	125	1	0	0
66	125	125	1	0	0
67	60	468	7.8000	1	0
68	60	468	7.8000	0	0
69	60	468	7.8000	1	0
70	60	234	3.9000	1	0
71	44	127	2.8863	1	0
72	60	468	7.8000	1	0
73	80	80	1	1	0
74	80	80	1	1	0
75	80	80	1	1	0
76	60	468	7.8000	1	0
77	80	80	1	1	0
78	80	80	1	1	0
79	80	80	1	1	0
80	125	125	1	1	0

ตัดแถวที่มี Missing Data ออกโดยใช้คำสั่ง
`datanonan = data(~any(isnan(data),2),:);`

ทำจำนวนแถวของ Class Ad และ Class Noad ให้เท่ากัน
และรวมข้อมูล



Name	Value	Min
count	758	758
cp_KNN	<1x1 biolearning.cl...>	
cp_SVM	<1x1 biolearning.cl...>	
data	<762x1559 double>	<Toi
dataad	<381x1559 double>	<Toi
datanoad	<1978x1559 double>	<Toi
datanoad2	<381x1559 double>	<Toi
datanonan	<2359x1559 double>	<Toi
eing	1	1
idx	<2359x1 logical>	
k	<1x1978 double>	1
n	230	230
percent_KNN	[13.4783,86.5217]	13.4

$data = dataad + datanoad2$

แบ่ง Training Data และ Testing Data

✓ train <762x1 logical>		
	1	2
1	1	
2	0	
3	1	
4	0	
5	0	
6	0	
7	1	
8	0	
9	0	
10	0	
11	0	
12	1	
13	1	
14	0	
15	1	
16	0	
17	1	
18	1	
19	1	
20	0	



data <762x1559 double>					
	1	2	3	4	5
1	125	125	1	1	0
2	57	468	8.2105	1	0
3	33	230	6.9696	1	0
4	60	468	7.8000	1	0
5	60	468	7.8000	1	0
6	60	468	7.8000	1	0
7	59	460	7.7966	1	0
8	60	234	3.9000	1	0
9	60	468	7.8000	1	0
10	60	468	7.8000	1	0
11	90	52	0.5777	1	0
12	90	60	0.6666	1	0
13	90	60	0.6666	1	0
14	33	230	6.9696	1	0
15	60	468	7.8000	1	0
16	60	468	7.8000	0	0
17	125	125	1	1	0
18	60	468	7.8000	1	0
19	30	585	19.5000	1	0
20	90	60	0.6666	1	0



✓ test <762x1 logical>		
	1	2
1	0	
2	1	
3	0	
4	1	
5	1	
6	1	
7	0	
8	1	
9	1	
10	1	
11	1	
12	0	
13	0	
14	1	
15	0	
16	1	
17	0	
18	0	
19	0	
20	1	

จำแนกด้วยเทคนิค

K-Nearest Neighbor Classifier

```
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%KNN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 - nub=0;
24 - class_KNN = knnclassify(data(test,1:1558),data(train,1:1558),data(train,1559),5);
```

Code K-NN

class_KNN <230x1 double>		
	1	2
20	1	
21	0	
22	1	
23	1	
24	1	
25	1	
26	0	
27	0	
28	0	
29	0	
30	1	
31	1	
32	1	
33	1	
34	1	
35	1	

ผลลัพธ์ในตาราง

percent_KNN =

86.5217 13.4783

ผลลัพธ์ที่แสดงออกมา

จำแนกด้วยเทคนิค

Support Vector Machine

```
71 %////////////////////////////////SVM////////////////////////////////////  
72 - nub=0;  
73 - svmStruct = svmtrain(data(train,1:1558),data(train,1559),'kernel_function',...  
74     'rbf','boxconstraint',1);  
75 - class_SVM = svmclassify(svmStruct,data(test,1:1558));
```

class_SVM <230x1 double>		
	1	2
20	1	
21	1	
22	1	
23	1	
24	1	
25	1	
26	1	
27	0	
28	1	
29	1	
30	1	
31	1	
32	1	
33	1	
34	1	
35	1	

ผลลัพธ์ในตาราง

Code SVM

percent_SVM =

92.6087 7.3913

ผลลัพธ์ที่แสดงออกมา

แสดงผลลัพธ์ที่ทำนายผิดด้วยเทคนิค K-Nearest Neighbor Classifier

wrongpoint_KNN <31x1559 double>									
	1	2	3	4	5	6	7	8	
1	2	2	1	1	0	0	0	0	
2	31	88	2.8387	1	0	0	0	0	
3	31	96	3.0967	0	0	0	0	0	
4	100	100	1	1	0	0	0	0	
5	20	58	2.9000	0	0	0	0	0	
6	136	93	0.6838	1	0	0	0	0	
7	124	120	0.9677	1	0	0	0	0	
8	20	83	4.1500	1	0	0	0	0	
9	45	100	2.2222	1	0	0	0	0	
10	45	100	2.2222	1	0	0	0	0	
11	45	100	2.2222	1	0	0	0	0	
12	31	88	2.8387	0	0	0	0	0	
13	93	261	2.8064	1	0	0	0	0	
14	33	270	8.1818	1	0	0	0	0	
15	124	120	0.9677	1	0	0	0	0	
16	45	345	7.6666	1	0	0	0	0	
17	240	120	0.5000	0	0	0	0	0	
18	90	215	2.3888	1	0	0	0	0	
19	171	227	1.3274	0	0	0	0	0	
20	74	78	1.0540	1	0	0	0	0	
21	174	100	0.5747	0	0	0	0	0	
22	80	86	1.0750	1	0	0	0	0	
23	29	230	7.9310	1	0	0	0	0	
24	60	95	1.5833	0	0	0	0	0	
25	52	144	2.7692	1	0	0	0	0	
26	46	109	2.3695	1	0	0	0	0	
27	60	234	3.9000	1	0	0	0	0	
28	24	236	9.8333	1	0	0	0	0	

แสดงผลลัพธ์ที่ทำนายผิดด้วยเทคนิค Support Vector Machine

[illegible]

ผลการทดลองและวิจารณ์

ครั้งที่	ผลการจำแนก (จาก testing 30%)			
	K-NN (K=5)		SVM	
	จำแนกถูก	จำแนกผิด	จำแนกถูก	จำแนกผิด
1	85.97	14.03	92.55	7.45
2	88.69	11.30	92.17	7.82
3	86.52	13.47	92.60	7.39
4	87.39	12.60	91.73	8.26
5	87.82	12.17	94.78	5.21
เฉลี่ยรวม	87.28	12.72	92.77	7.22

เปรียบเทียบประสิทธิภาพด้วยการทำ Accuracy

สรุปผลที่ได้จากการทำโครงการ

เทคนิค K-Nearest Neighbor Classifier จำแนกเฉลี่ยได้ถูกต้อง 87.28%

เทคนิค Support Vector Machine จำแนกเฉลี่ยได้ถูกต้อง 92.77%

จากสรุปผลเห็นว่าผลที่ได้จากการดำเนินงานโดยใช้เทคนิควิธี Support Vector Machine ทำนายภาพโฆษณาบนได้ดีว่าเทคนิควิธี K-Nearest Neighbor Classifier ซึ่งมีวิธีการดำเนินการนั้นขึ้นอยู่กับค่า k ซึ่งหากเลือกค่า k ที่ไม่เหมาะสมก็จะทำให้ความถูกต้องนั้นน้อยลงไป

ปัญหาและอุปสรรคที่พบ

- ระยะเวลาในการทำโครงงานน้อยเกินไป
- ถ้าทำการลดมิติข้อมูลด้วย PCA ก่อน attribute ที่น้อยไปจะไม่สามารถจำแนกข้อมูลให้ถูกต้องตรงตาม class ได้
- เวลาในการประมวลผลโปรแกรมที่ข้อมูลมีจำนวนมากเกินไปทำให้ประมวลผลใช้เวลานาน

จบการนำเสนอ