ECT_HW4

2020

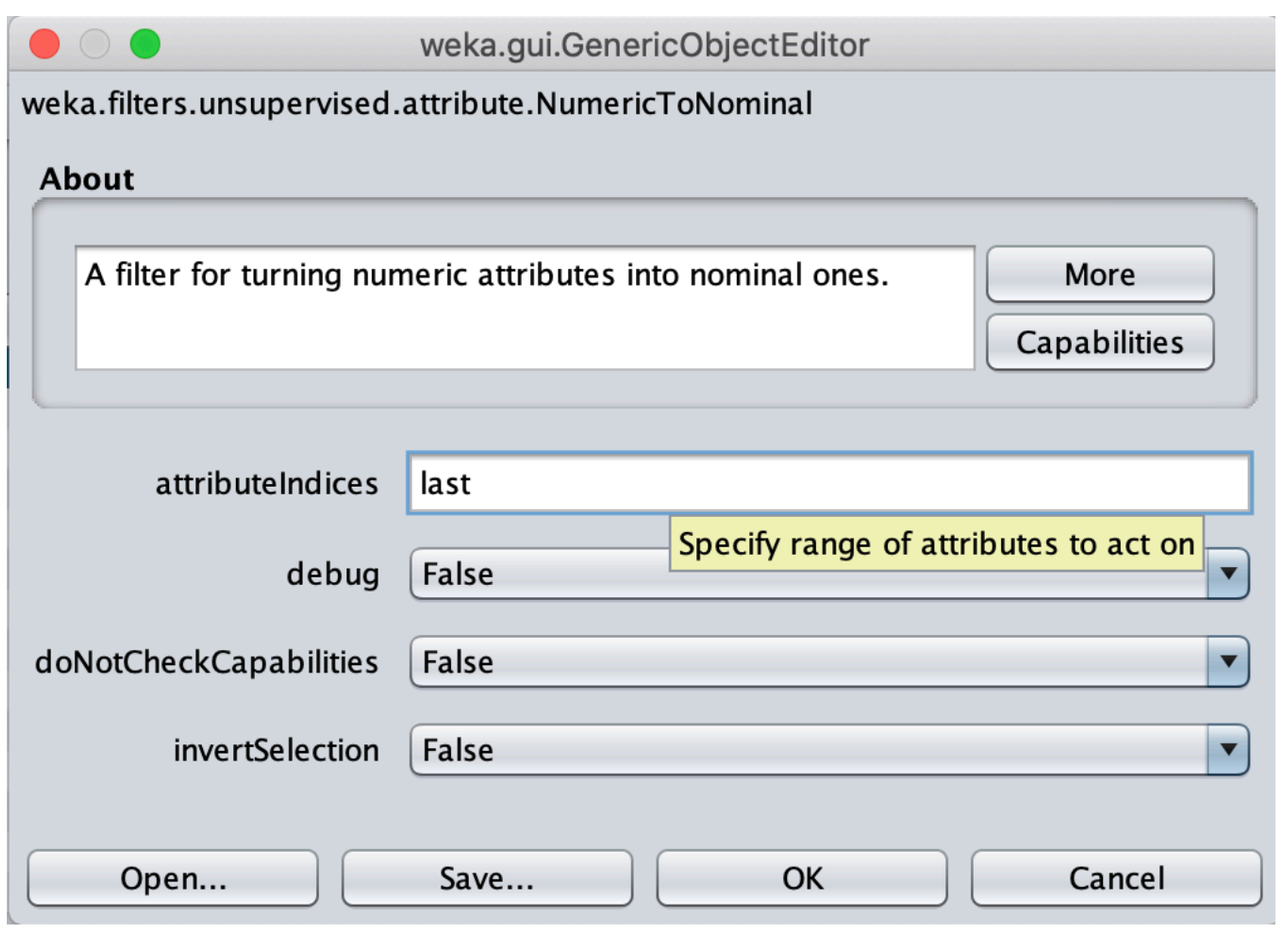
1. 使用weka

(a)題目

- 請嘗試修改 heart.arff，使其可以使用 SMO function 進行 SVM 分析，並說明原本為何無法使用 SMO (5%)

(a)答案

- 選擇NumericToNominal，將最後一項’target’變成nominal
- 因為原先target被當成numeric，但classify只能對nominal做分類



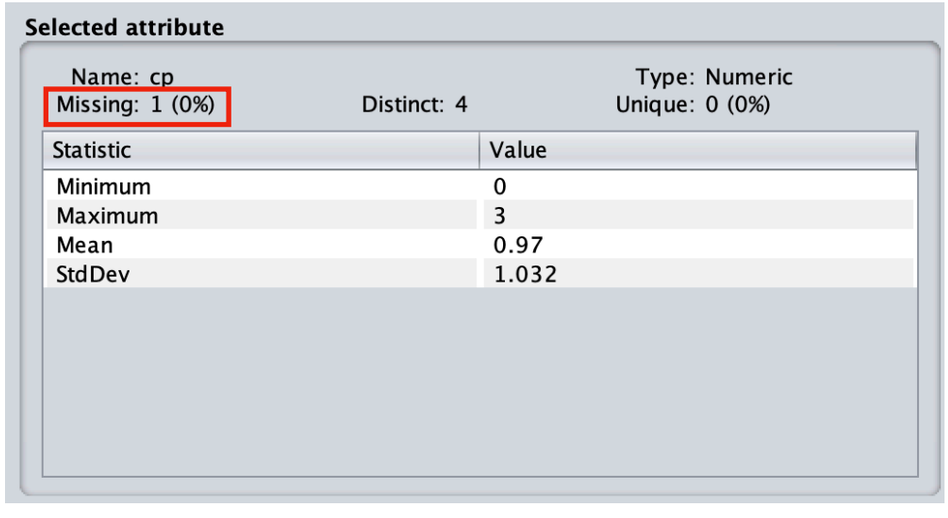
(b)題目

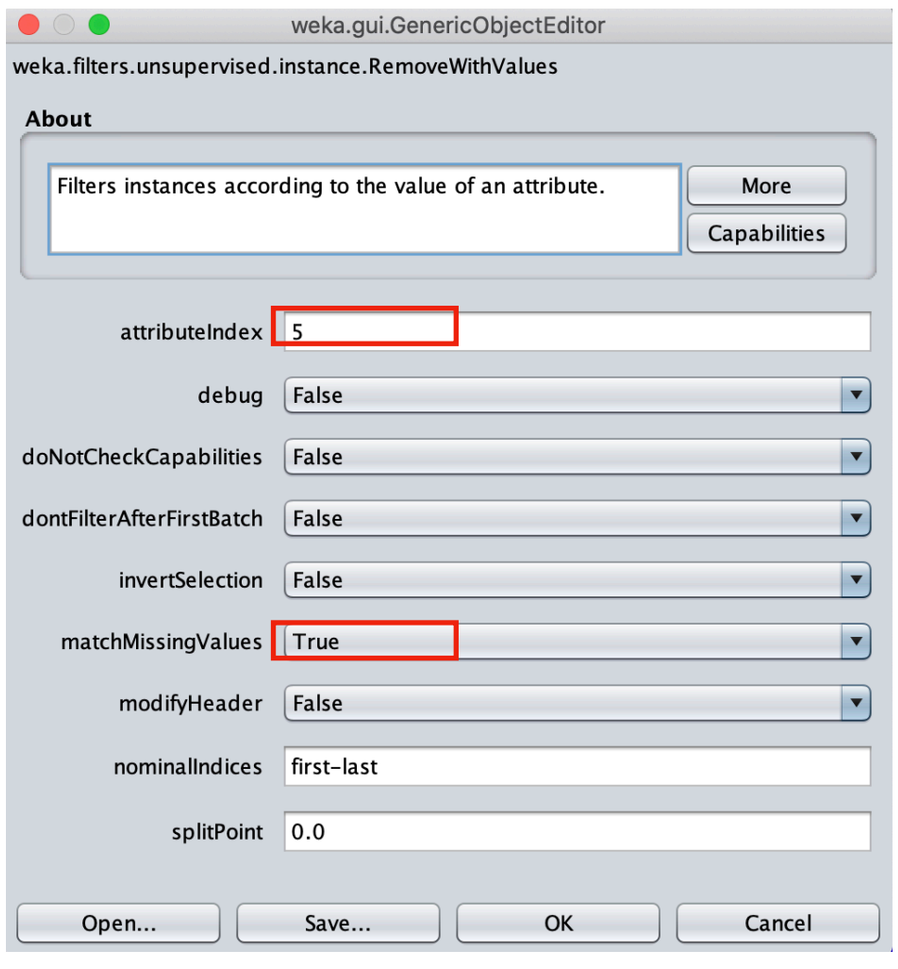
- 請嘗試去除有空值的資料 (5%)

(b)答案

- 可以從selected attribute那邊看到是否有missing value

- 對所有有missing value的屬性使用RemoveWithValue，matchMissingValues設為True，attributeIndex設為有missing value的屬性index



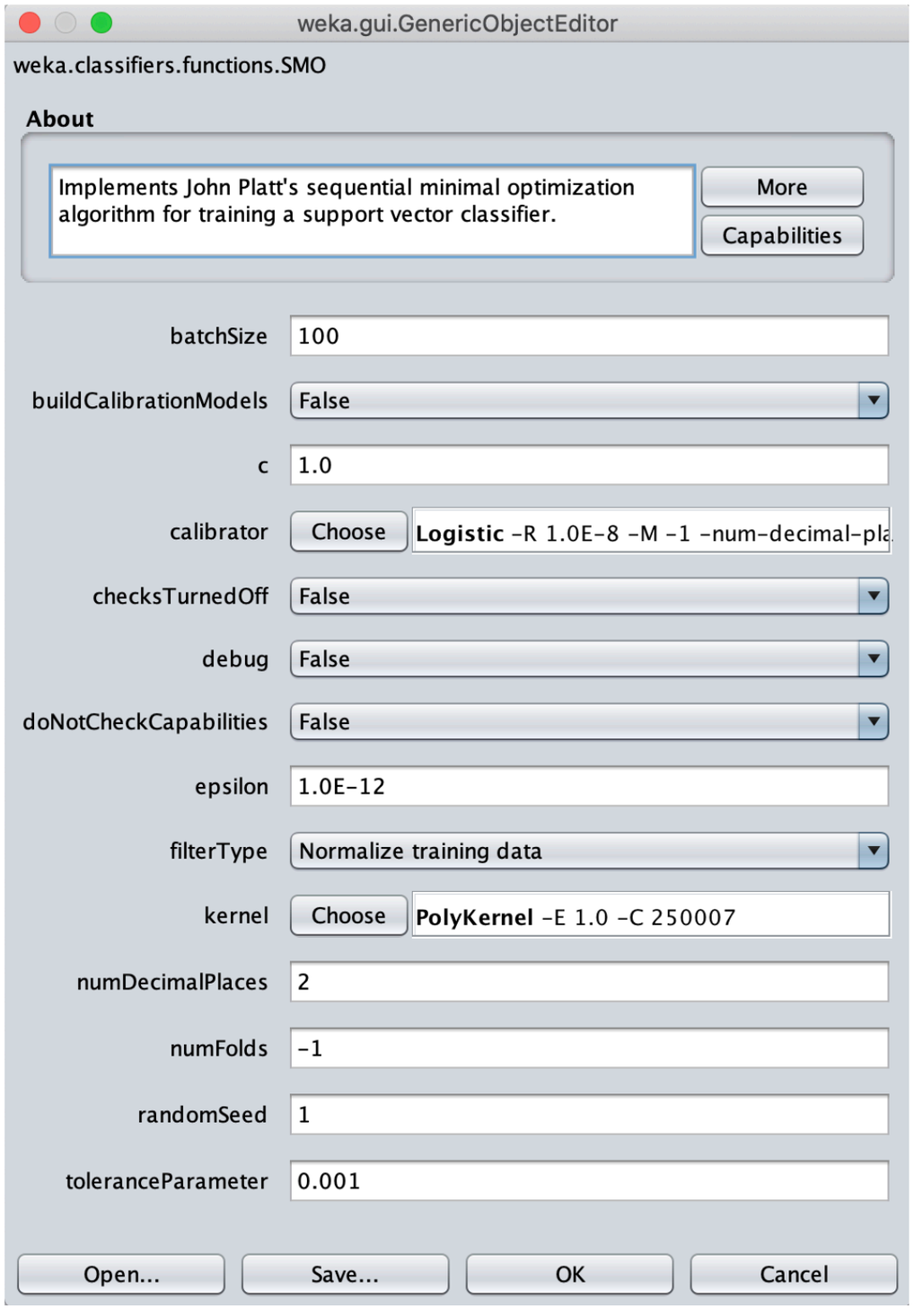


(c)題目

- 用SMO function 對前處理過的 heart.arff 進行 SVM 分析， kernel 設為' linear '，Percentage spilt 設為 66%，截圖並附上過 程及準確率 (30%)

Hint: poly kernel, exponent: 1

(c)答案




```
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

=== Summary ===

Correctly Classified Instances          86               86      %
Incorrectly Classified Instances        14               14      %
Kappa statistic                          0.72
Mean absolute error                      0.14
Root mean squared error                  0.3742
Relative absolute error                 28      %
Root relative squared error             74.2256 %
Total Number of Instances              100
```

2. 使用python

(d)題目

- 請問資料集是否有空值?有幾筆資料含有空值?如有空值即去 掉該筆資料 (5%)

(d)答案

- 有空值，共10筆資料含有空值

```
df[df.isnull().values==True]
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
20	59	1	0.0	135	234.0	0	1.0	161	NaN	0.5	1.0	0	3	1
23	61	1	2.0	150	243.0	1	NaN	137	1.0	1.0	1.0	0	2	1
28	65	0	2.0	140	NaN	1	0.0	157	0.0	0.8	2.0	1	2	1
32	44	1	1.0	130	219.0	0	0.0	188	0.0	NaN	2.0	0	2	1
49	53	0	0.0	138	234.0	0	NaN	160	0.0	0.0	2.0	0	2	1
126	47	1	0.0	112	204.0	0	NaN	143	0.0	0.1	2.0	0	2	1
181	65	0	0.0	150	225.0	0	0.0	114	0.0	NaN	1.0	3	3	0
240	70	1	2.0	160	NaN	0	1.0	112	1.0	2.9	1.0	1	3	0
243	57	1	NaN	152	274.0	0	1.0	88	1.0	1.2	1.0	1	3	0
299	45	1	3.0	110	264.0	0	1.0	132	0.0	1.2	NaN	0	3	0

- 用dropna()即可去除空值資料

```
df = df.dropna()
```

(e)題目

- 將最後一個屬性值”target”切分出來，其餘屬型切分為Feature (5%)

(e)答案

```
x = df.drop('target', axis=1)
y = df['target']
```

(f)題目

- 將Feature用sklearn.preprocessing的StandardScaler進行標準化 (10%)

(f)答案

```
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
```

```
scaler = preprocessing.StandardScaler()
scaler.fit(x)
X = scaler.transform(x)
```

(g)題目

- 切分資料集與測試集，設 test_size=0.33，random_state=1 (10%)

(g)答案

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
```

(h)題目

- 最後，使用 sklearn.svm 裡的 SVC 進行分析，kernel 設為 ' linear ‘，並印出模型最終的準確度 (30%)

(h)答案

```
from sklearn.svm import SVC
from sklearn import metrics
svc = SVC(kernel='linear')
svc.fit(X_train, y_train)
y_pred=svc.predict(X_test)
print('Accuracy score:')
print(metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy score:
0.865979381443299
```