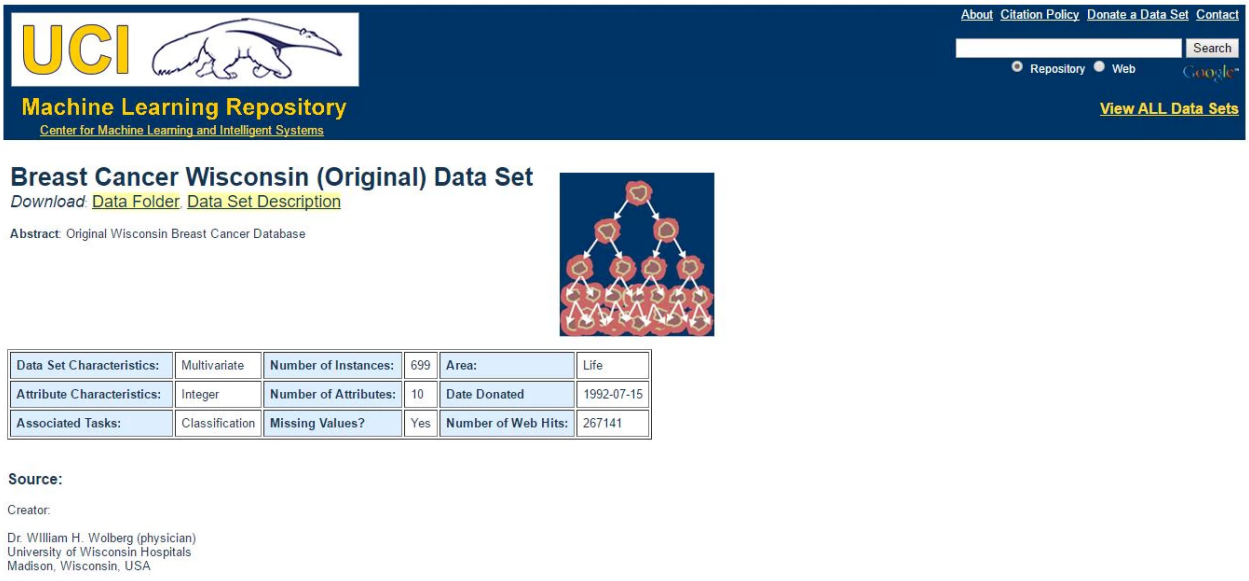
# 物聯網 Final project 教案

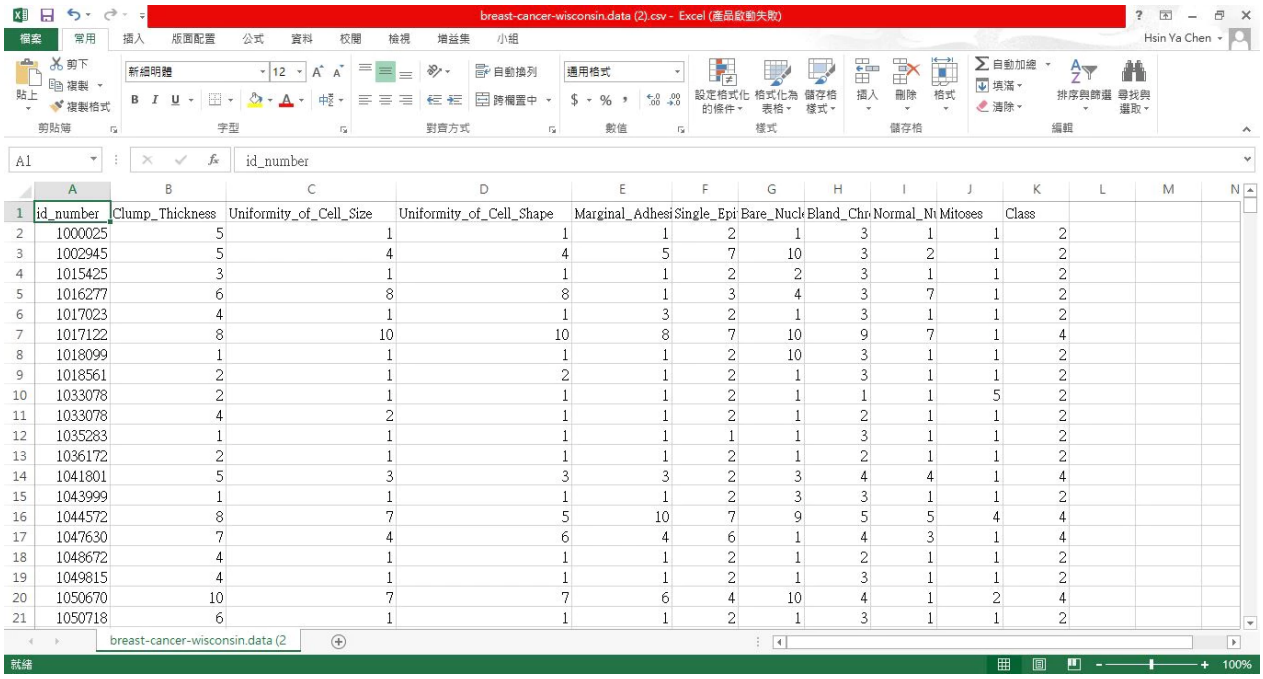
作者：4102056040陳薪雅、4102056044黃筱真

## 題目：乳癌預測分析

1. 從 UCI Data set 下載 Breast Cancer Wisconsin (Original) Data Set




2. 將 Data set 轉換成 .csv檔，並加上Column name名稱以辨別不同Attributes




| id_number | Clump_Thickness | Uniformity_of_Cell_Size | Uniformity_of_Cell_Shape | Marginal_Adhesi | Single_Epi | Bare_Nucl | Bland_Chr | Normal_Nu | Mitoses | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| 1000025 | 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| 1002945 | 5 | 4 | 4 | 5 | 7 | 10 | 3 | 2 | 1 | 2 |
| 1015425 | 3 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 2 |
| 1016277 | 6 | 8 | 8 | 1 | 3 | 4 | 3 | 7 | 1 | 2 |
| 1017023 | 4 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 1 | 2 |
| 1017122 | 8 | 10 | 10 | 8 | 7 | 10 | 9 | 7 | 1 | 4 |
| 1018099 | 1 | 1 | 1 | 1 | 2 | 10 | 3 | 1 | 1 | 2 |
| 1018561 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| 1033078 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 5 | 2 |
| 1033078 | 4 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 |
| 1035283 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 2 |
| 1036172 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 |
| 1041801 | 5 | 3 | 3 | 3 | 2 | 3 | 4 | 4 | 1 | 4 |
| 1043999 | 1 | 1 | 1 | 1 | 2 | 3 | 3 | 1 | 1 | 2 |
| 1044572 | 8 | 7 | 5 | 10 | 7 | 9 | 5 | 5 | 4 | 4 |
| 1047630 | 7 | 4 | 6 | 4 | 6 | 1 | 4 | 3 | 1 | 4 |
| 1048672 | 4 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 |
| 1049815 | 4 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| 1050670 | 10 | 7 | 7 | 6 | 4 | 10 | 4 | 1 | 2 | 4 |
| 1050718 | 6 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |

3. 此資料集的 missing value 原以 "?" 表示，但我們發現 "?" 在分析時無法判斷成NULL，因此將資料集中所有的 "?" 改成空白，才能正確找出missing value

4. 使用 pandas 載入 Breast Cancer Wisconsin (Original) Data Set

```
import pandas as pd
data=pd.read_csv("breast-cancer-wisconsin.data.csv")
```

5. 觀察 data 的整體概況以及相關性

```
from pandas import DataFrame as df
import matplotlib.pylab as plt

print(data.info())
print(data.describe())
mat = data.ix[:,1:10]
correlation = mat.corr()
print(correlation)
correMatrix = df(correlation)
plt.pcolor(correMatrix)
plt.show()
```

6. 找出 data 的 missing value並排除

```
print(pd.isnull(data.Bare_Nuclei).value_counts())
data = data[np.isfinite(data['Bare_Nuclei'])]
```

7. 將 attributes 與 class 分別取出並分割 training data & testing data size

```
from sklearn.cross_validation import train_test_split as tts

X = data.values
Y = data.loc[:,'Class'].values
X_train,X_test,Y_train,Y_test = tts(X, Y, test_size=0.5, random_state=0)
```

8. 將 attributes 進行正規化

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train_std = sc.fit_transform(X_train)
X_test_std = sc.fit_transform(X_test)
```

9. 分別載入 Perceptron、Support Vector Machine、Logistic Regression、Naive Bayes 四種不同的 model 進行分類及預測，再算出準確率

```
from sklearn.metrics import accuracy_score
#---------------Perceptron-----------------
```

```
from sklearn.linear_model import Perceptron

ppn = Perceptron()
ppn.fit(X_train_std, Y_train)
Y_pred = ppn.predict(X_test_std)
print(accuracy_score(Y_test,y_predict))

#-----------------SVM-----------------------
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

tuned_parameters = [{'kernel': ['rbf'], 'gamma': [1e-3, 1e-4],
                     'C': [1, 10, 100, 1000]},
                    {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}]
clf = GridSearchCV(SVC(C=1.0),tuned_parameters,cv=5)
clf = SVC()
clf.fit(X_train_std,Y_train)
y_predict = clf.predict(X_test_std)
print(accuracy_score(Y_test,y_predict))

#-----------Logistic Regression-------------
from sklearn.linear_model import LogisticRegression as lr

clf=lr(C=1,tol=1e-4)
clf.fit(X_train_std,Y_train)
y_predict = clf.predict(X_test_std)
print(accuracy_score(Y_test,y_predict))

#--------------Naive Bayes------------------
from sklearn.naive_bayes import BernoulliNB

clf = BernoulliNB()
clf.fit(X_train_std, Y_train)
Y_pred = clf.predict(X_test_std)
print(accuracy_score(Y_test,y_predict))
```

10. 反覆重複上一步驟10次後，可得平均準確率，並report 出 precision、recall、f1-score、support

```
from sklearn.metrics import classification_report as clf_report

report = clf_report(Y_test,y_predict,digits=5)
```

```
print(report)
```

11. 之後我們利用ABS公式，找出屬性重要程度 top 5

$$ABS\ 公式 = \frac{良性腫瘤平均值 - 惡性腫瘤平均值}{(良性腫瘤標準差 + 惡性腫瘤標準差)/2}$$

```
from operator import itemgetter

dic = {}
for i in range(1,10):
    sub = data.ix[:,[i,10]]
    good = sub[sub.Class==2]
    bad = sub[sub.Class==4]
    abss = (good.ix[:,0].mean()-bad.ix[:,0].mean())/((good.ix[:,0].std()+bad.ix[:,0].std())/2)
    dic[sub.columns[0]] = abss

newDic = sorted(dic.items(), key=itemgetter(1))
top = df(newDic, columns=['Attr', 'score'])
print(top.ix[0:4,0])
```

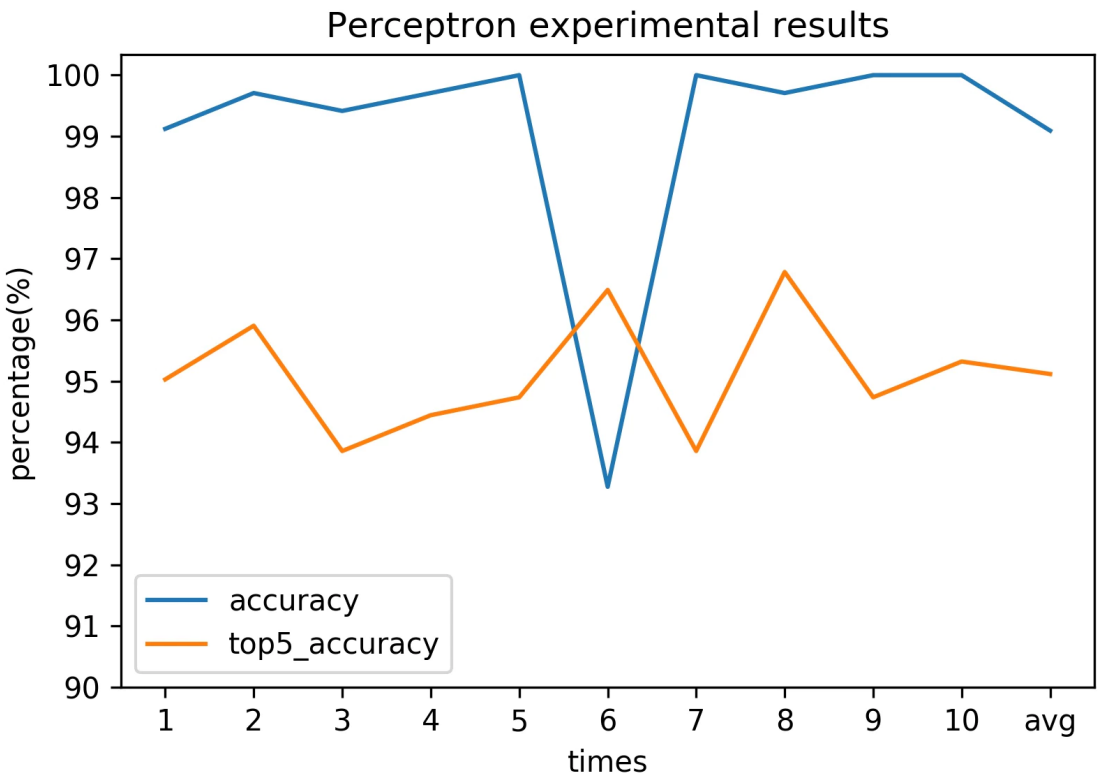
| Attribute | value | rank |
|---|---|---|
| Uniformity_of_Cell_Size 細胞大小的均勻性 | 2.944829558780585 | 1 |
| Bare_Nuclei 裸細胞核 | 2.9250103336912967 | 2 |
| Uniformity_of_Cell_Shape 細胞形狀的均勻性 | 2.9189208207504 | 3 |
| Bland_Chromatin 染色質 | 2.326987111848247 | 4 |
| Normal_Nucleoli 細胞核正常程度 | 2.136167539950982 | 5 |
| Marginal_Adhesion 邊緣粘著性 | 2.0608731337771795 | 6 |
| Clump_Thickness 腫塊厚度 | 2.0553461783927625 | 7 |
| Single_Epithelial_Cell_Size 單上皮細胞大小 | 1.938590187375081 | 8 |
| Mitoses 細胞有絲分裂 | 1.0000513712812957 | 9 |

12. 取出屬性重要程度 top 5，再重複步驟 7 ~ 10，即可得出準確率

13. 比較 全部資料 與 top 5 的準確率差異

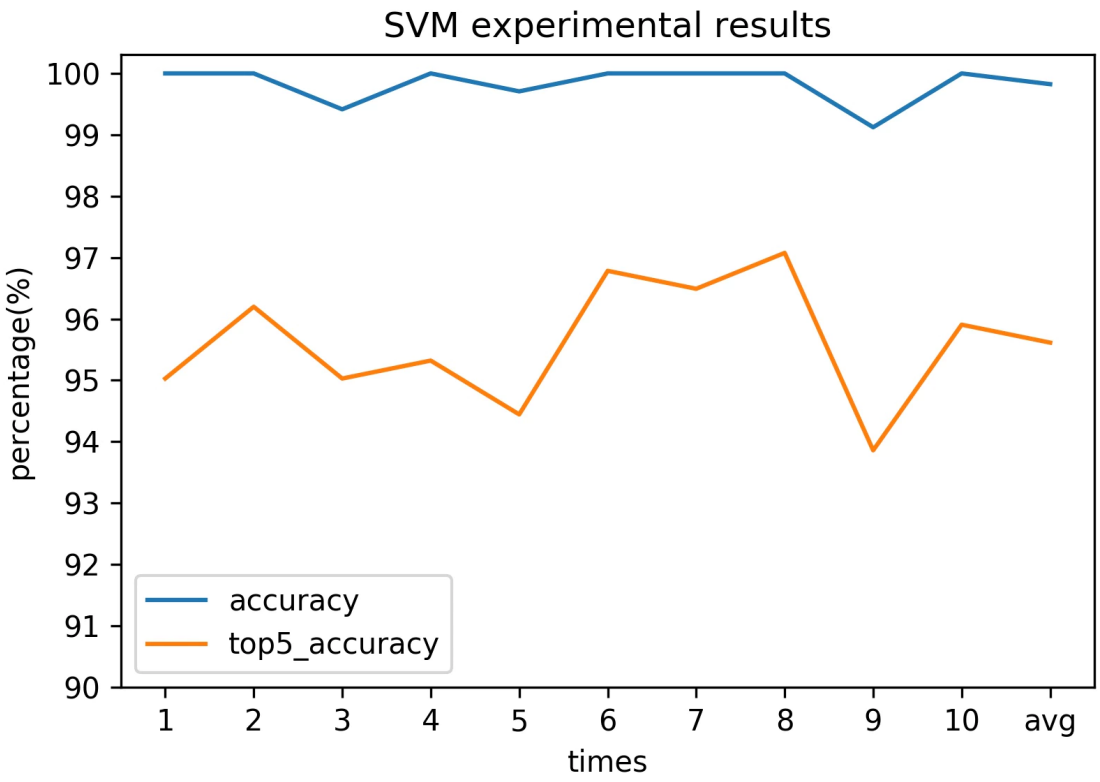
## Perceptron

| | 全部資料 | top 5 |
|---|---|---|
| 十次平均準確率 | 99.09% | 95.12% |



SVM

| | 全部資料 | top 5 |
|---|---|---|
| 十次平均準確率 | 99.82% | 95.61% |

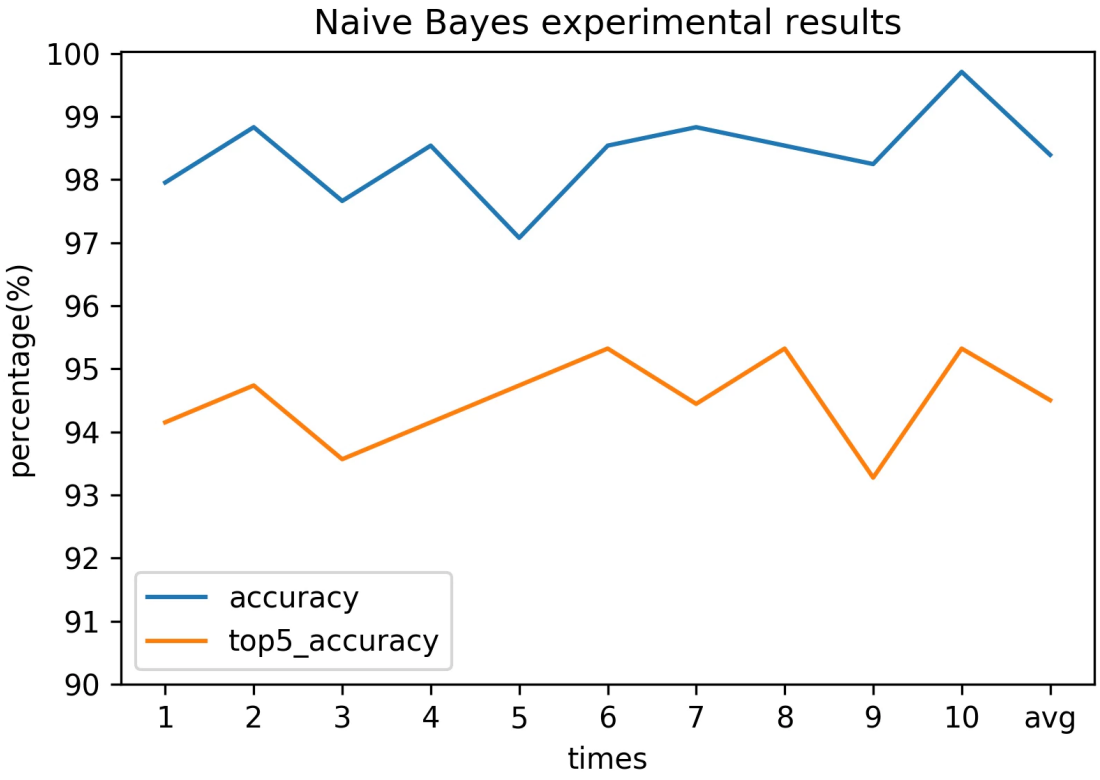


## Logistic Regression

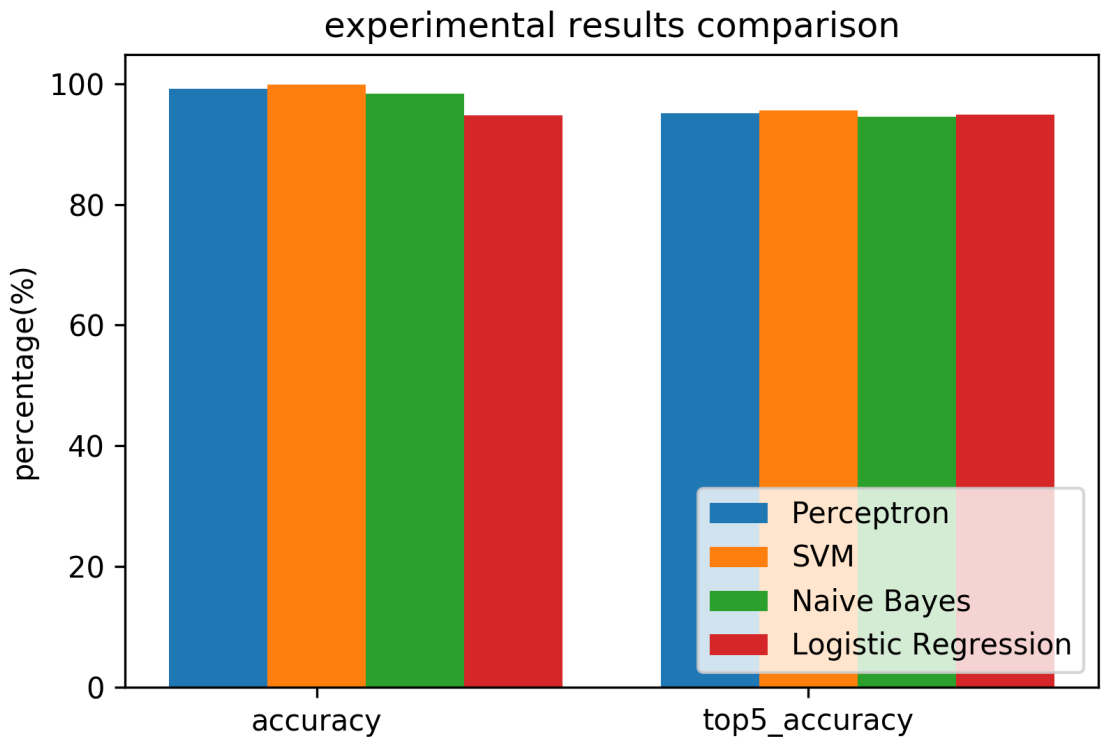
| | 全部資料 | top 5 |
|---|---|---|
| 十次平均準確率 | 94.78% | 94.83% |

## Naive Bayes

| | 全部資料 | top 5 |
|---|---|---|
| 十次平均準確率 | 98.39% | 94.50% |



綜合比較




實驗結果分析：

1. 在 全部資料 與 top 5 的準確率中，都是 SVM 勝出
2. 在 全部資料 的準確率中，Logistic Regression的表現最差
3. 在 top 5 的準確率中，4種分類器的表現並無太大差異
4. 以 Logistic Regression 分類，top 5 的準確度 高於 全部資料的準確度
5. 由實驗結果可得，在時間急迫的情況下，可優先檢查以下5個屬性，便可大致判斷腫瘤為良性或惡性，而分類器可採用 SVM 來做判斷

| Attribute | value | rank |
|---|---|---|
| Uniformity_of_Cell_Size 細胞大小的均勻性 | 2.944829558780585 | 1 |
| Bare_Nuclei 裸細胞核 | 2.9250103336912967 | 2 |
| Uniformity_of_Cell_Shape 細胞形狀的均勻性 | 2.9189208207504 | 3 |
| Bland_Chromatin 染色質 | 2.326987111848247 | 4 |
| Normal_Nucleoli 細胞核正常程度 | 2.136167539950982 | 5 |

參考文獻：


1. UCI Breast Cancer Wisconsin (Original) Data Set
2. 賴琴文，以資料探勘與模糊邏輯技術建置乳癌疾病診斷系統，義守大學資訊管理系研究所碩士論文，2010
3. 李俊宏、古清仁，類神經網路與資料探勘技術在醫療診斷之應用研究，國立高雄應用科技大學電機工程系碩士論文，2009
4. scikit-learn Machine Learning in Python
5. Matplotlib: Python plotting
6. pandas: powerful Python data analysis toolkit