1.這次作業我是用 Lubuntu 的 terminal 寫的，直接在 terminal 打上這幾行 code 即可看到結果

>>python3 mushroom.py

>>python3 iris.py

2.

(1)data input:

```
df = pd.read_csv('agaricus-lepiota.data', header = None)
```

把他轉成 dataframe 形式好處理

(2)value frequency:

Mushroom:有上傳檔案，分別是全部資料，和取七成的 e 和 p 的 value frequency，列是各項 feature，行是各個特性出現的表示(e,p,y,u…)，共三個 csv 檔

Iris:第一列是 setosa class 的'sepal_length', 'sepal_width', 'petal_length', 'petal_width'平均，第二是他們的標準差

三四列分別是 versicolor 的平均及標準差

56 就 virginica(同上) ，有上傳檔案，在 mean_dev.csv

(3).drop missing value:

```
for col in df.columns:
    delete = df[df[col] == "?"].index
    df.drop(delete, inplace = True)
```

上面可以 drop 掉有?符號的列

Shuffle:

df_e = df_e.sample(frac=1).reset_index(drop = True) 洗亂資料

(4).model construction:

Mushroom:按照剛剛的 value frequency，分別去計算 P(Xi|Y)的機率，再乘起來(取 log 後)看是 e 還是 p 的可能大

Iris:用 Gaussian distribution 算出 P(Xi|Y)再去比較有可能是哪種 class

(5). Train-Test-Split:

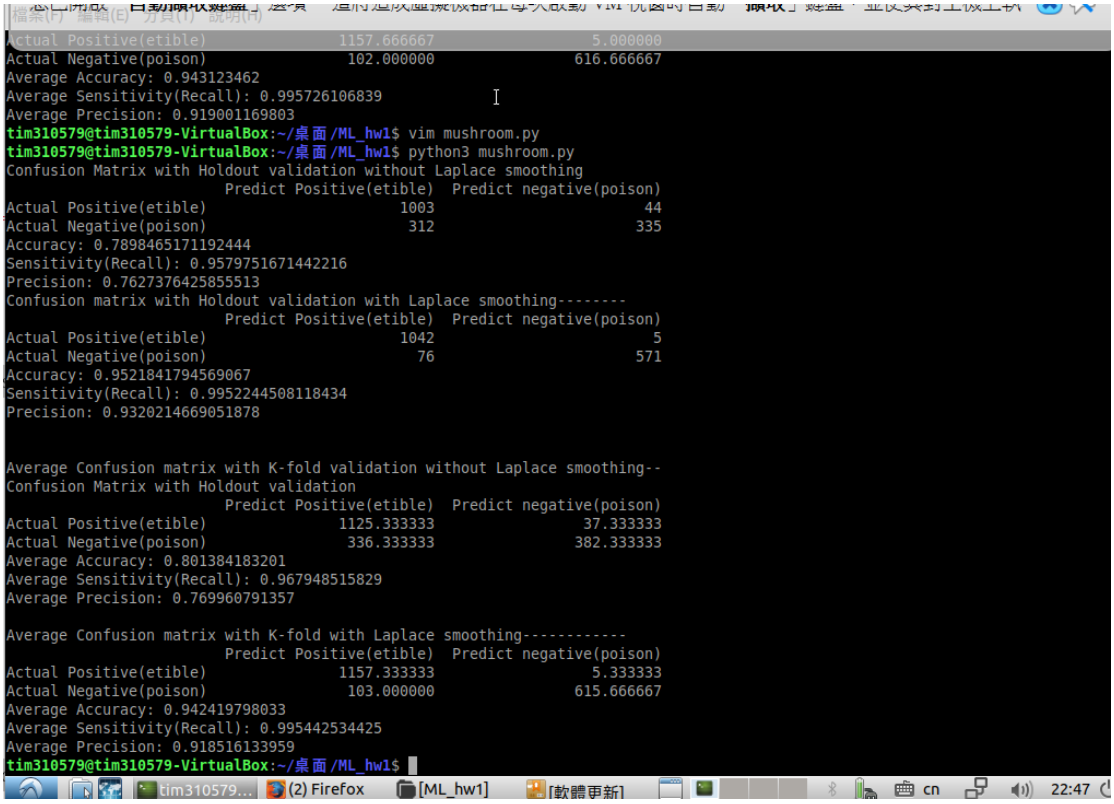
Holdout 先從洗亂的'e'裡面抓出 7/10，'p'也是，計算完 value frequency 後，把剩下的各 3/10 合併起來，變成 test data

Iris 也是差不多

K-fold 將資料洗亂，在隨機分成 3 分，每一份都可以當作 test data，並把另兩份合起來當作 train data，然後做 3 次再取平均就可以了。

(6).Result:

mushroom

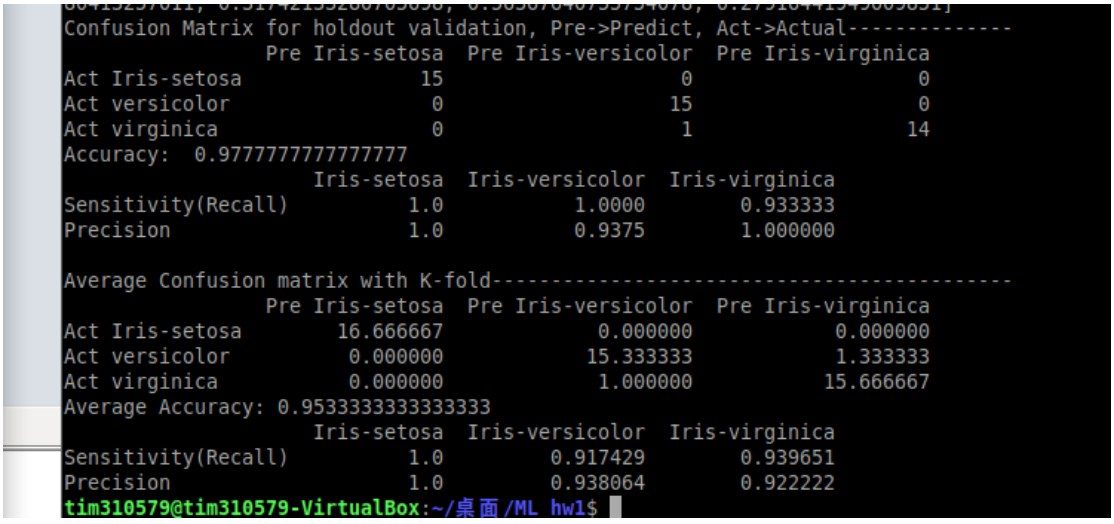


```
Actual Positive(etible)            1157.666667               5.000000
Actual Negative(poison)             102.000000             616.666667
Average Accuracy: 0.943123462
Average Sensitivity(Recall): 0.995726106839
Average Precision: 0.919001169803
tim310579@tim310579-VirtualBox:~/桌面/ML_hw1$ vim mushroom.py
tim310579@tim310579-VirtualBox:~/桌面/ML_hw1$ python3 mushroom.py
Confusion Matrix with Holdout validation without Laplace smoothing
                    Predict Positive(etible)  Predict negative(poison)
Actual Positive(etible)                  1003                        44
Actual Negative(poison)                   312                       335
Accuracy: 0.7898465171192444
Sensitivity(Recall): 0.9579751671442216
Precision: 0.7627376425855513
Confusion matrix with Holdout validation with Laplace smoothing-------
                    Predict Positive(etible)  Predict negative(poison)
Actual Positive(etible)                  1042                         5
Actual Negative(poison)                    76                       571
Accuracy: 0.9521841794569067
Sensitivity(Recall): 0.9952244508118434
Precision: 0.9320214669051878


Average Confusion matrix with K-fold validation without Laplace smoothing--
Confusion Matrix with Holdout validation
                    Predict Positive(etible)  Predict negative(poison)
Actual Positive(etible)             1125.333333                37.333333
Actual Negative(poison)              336.333333               382.333333
Average Accuracy: 0.801384183201
Average Sensitivity(Recall): 0.967948515829
Average Precision: 0.769960791357

Average Confusion matrix with K-fold with Laplace smoothing-----------
                    Predict Positive(etible)  Predict negative(poison)
Actual Positive(etible)             1157.333333                 5.333333
Actual Negative(poison)              103.000000               615.666667
Average Accuracy: 0.942419798033
Average Sensitivity(Recall): 0.995442534425
Average Precision: 0.918516133959
tim310579@tim310579-VirtualBox:~/桌面/ML_hw1$
```

iris



```
Confusion Matrix for holdout validation, Pre->Predict, Act->Actual-------------
                 Pre Iris-setosa  Pre Iris-versicolor  Pre Iris-virginica
Act Iris-setosa               15                    0                   0
Act versicolor                 0                   15                   0
Act virginica                  0                    1                  14
Accuracy:  0.9777777777777777
                      Iris-setosa  Iris-versicolor  Iris-virginica
Sensitivity(Recall)           1.0           1.0000        0.933333
Precision                     1.0           0.9375        1.000000

Average Confusion matrix with K-fold-----------------------------------------
                 Pre Iris-setosa  Pre Iris-versicolor  Pre Iris-virginica
Act Iris-setosa        16.666667             0.000000            0.000000
Act versicolor          0.000000            15.333333            1.333333
Act virginica           0.000000             1.000000           15.666667
Average Accuracy: 0.9533333333333333
                      Iris-setosa  Iris-versicolor  Iris-virginica
Sensitivity(Recall)           1.0         0.917429        0.939651
Precision                     1.0         0.938064        0.922222
tim310579@tim310579-VirtualBox:~/桌面/ML_hw1$
```

(7) Comparison & Conclusion

圖中可以看到，沒有做 laplace 的結果，準確率只有 80%，可能是因為有幾個機率是零的結果被跳過，導致 feature 數變少，進而導致準確度下降，但 laplace 之後，就不會有這個問題，準確率也相對提高。

Iris 的結果好像也是，雖然都滿準確的，但 k-fold 稍微低一點，但是是做了三次的平均，也可以說他是做了多次後的結果，應該滿有可信度的。