Lab-4 [Metrics of Evaluation]

Out date: Jun 27, 2022
Due date: July 03, 2022 at 11:59PM

Submission

1. Prepare your solution in Orange and save the workspace for Problem 1 (e.g., Lab-1_1_LastName.ows) **[10 points]**
2. Complete the tables given below and save the file (e.g., Lab-1_LastName.docx). **[80 points]**
3. Upload the files to the Canvas.

Objective: To review and understand metrics of evaluation available in Orange for classification problems.

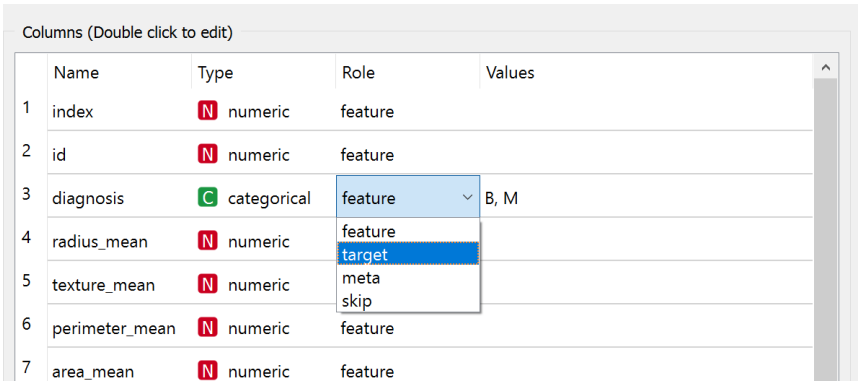
Problem 1/2. [100 points]

Data: For this lab, please download *wisc_bc_data_with_index.csv* from Canvas to your folder. This is breast cancer dataset includes measurements from digitized images of fine-needle aspirate of a breast mass. A clinician examines the cells under a microscope to determine whether the mass is likely to be malignant (M) or benign (B). The values represent characteristics of the cell nuclei present in the digital image. Data is available on UCI ML website (http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

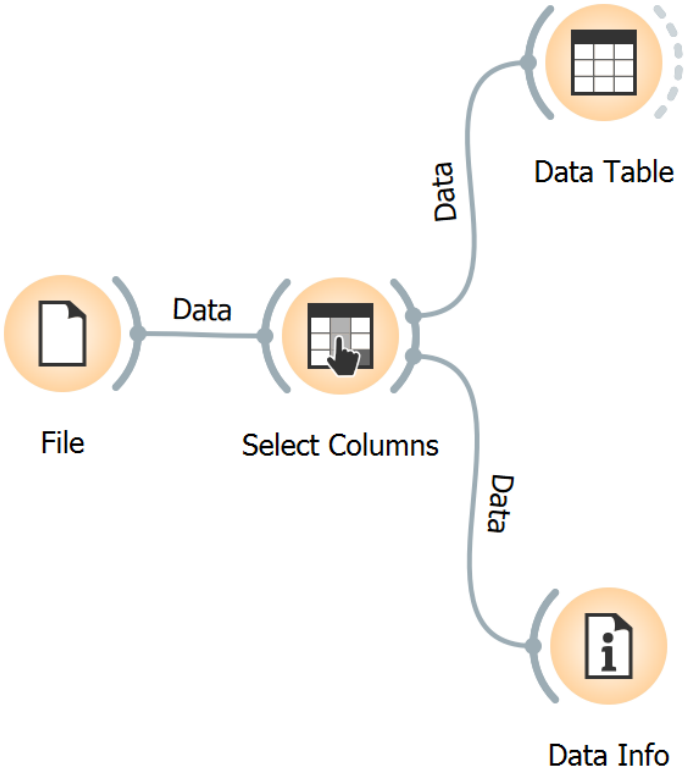
The data has 3 measurements (mean, standard error, worst/largest values) and 10 characteristics (Radius, Texture, Area, Perimeter, etc.).

Lab Instructions

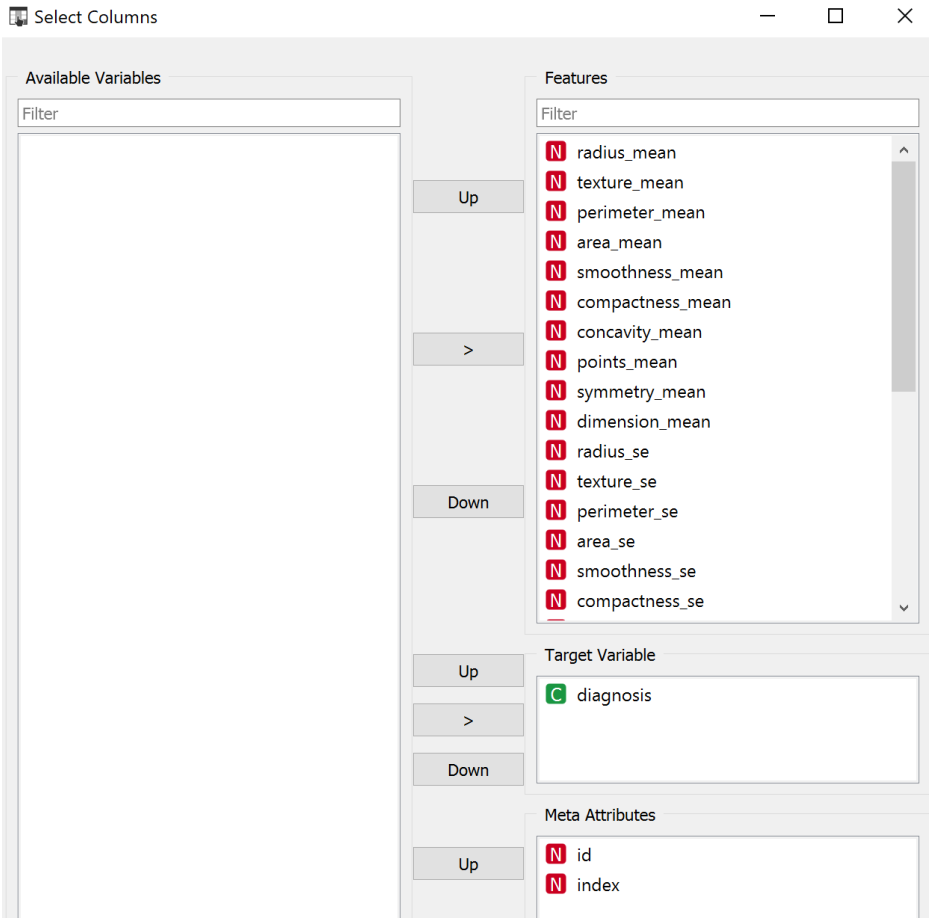
1. Bring in the **File** widget.
2. Load the *wisc_bc_data_with_index.csv*.
3. Open File window by double clicking on **File**.
4. Change the **diagnosis** feature to target as shown below and click **Apply**.



5. Add the **Select Columns**, **Data Table** and **Data Info** widgets as shown below.

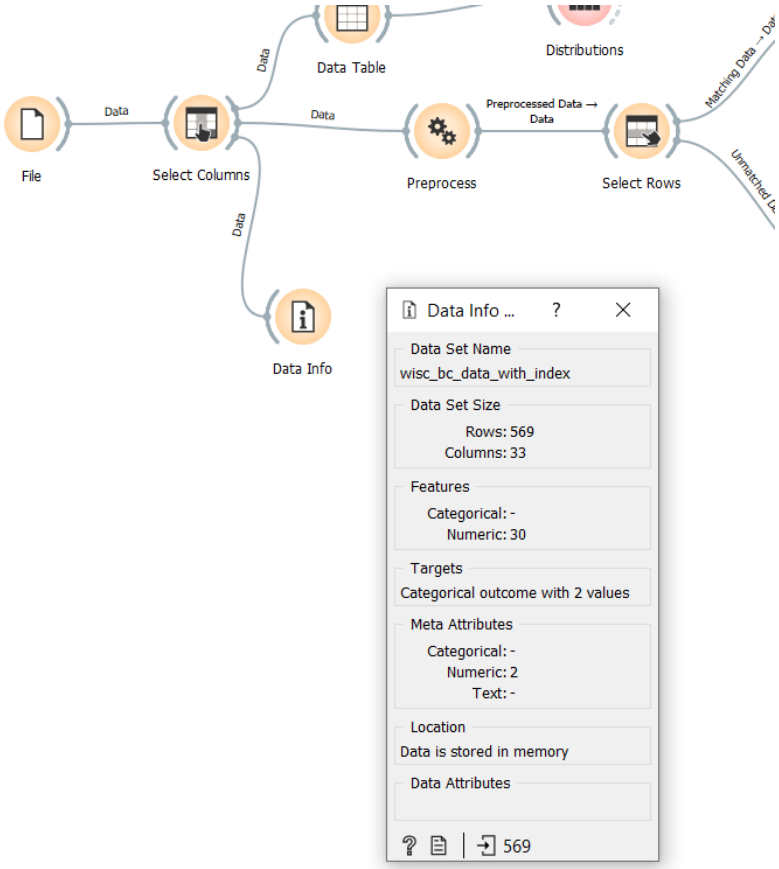


6. Organize the features as shown below (diagnosis → Target Variable, id and index → Meta Attributes).

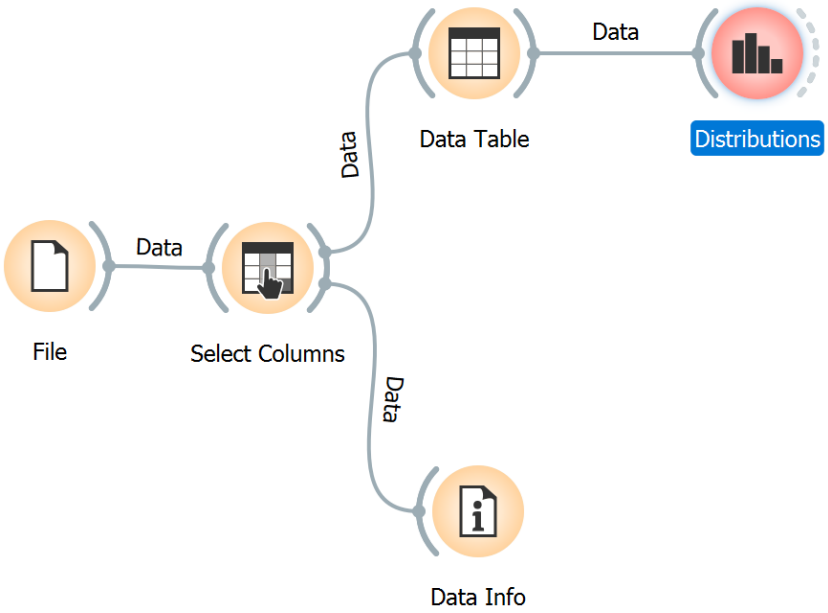


7. Answer the following questions for this data:

How many objects (rows) it has?	~~599~~ **569**
What is the dimensionality (columns) of this data?	~~2~~ **33**
What are the class levels of the target feature?	B, M

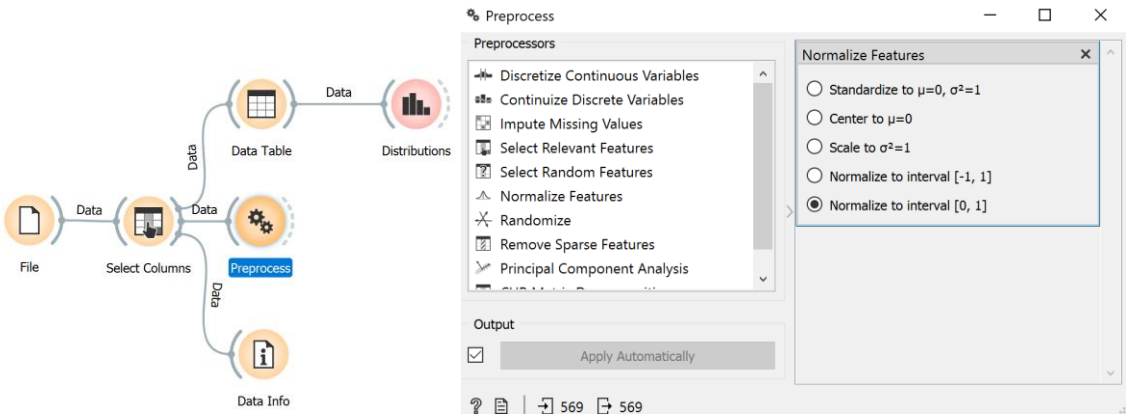


8. Add the **Distribution** widget as shown below.

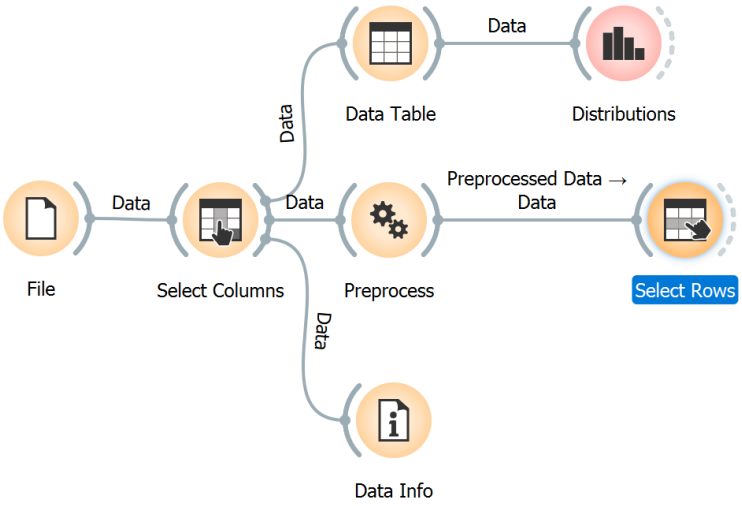


Comment on the class distribution.	Approx. 63% benign and 37% that are actually malignant

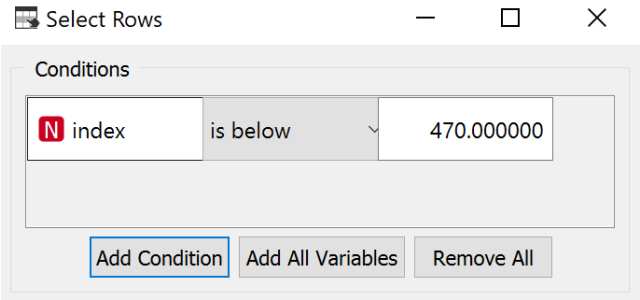
9. Add the **Preprocess** widget and add the **[0, 1] normalization** module as shown below:



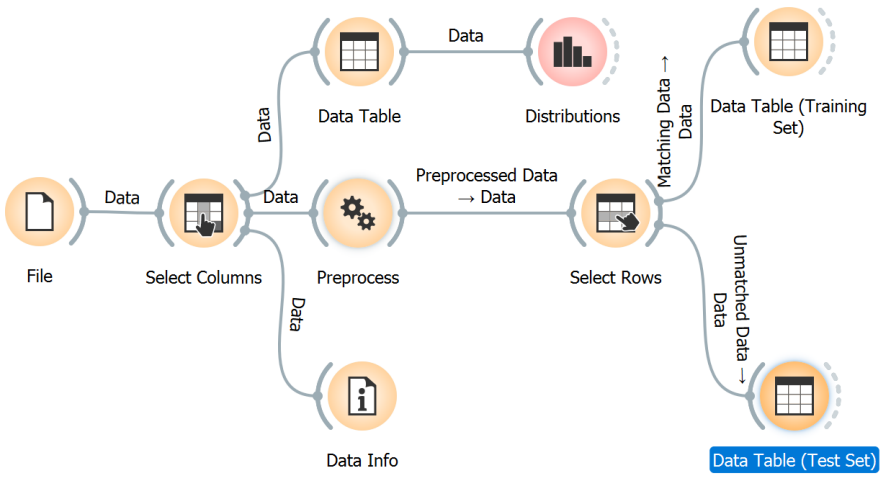
10. Add the **Select Rows** widget as shown below:



11. Double click on **Select Rows** and set **Conditions** as shown below:



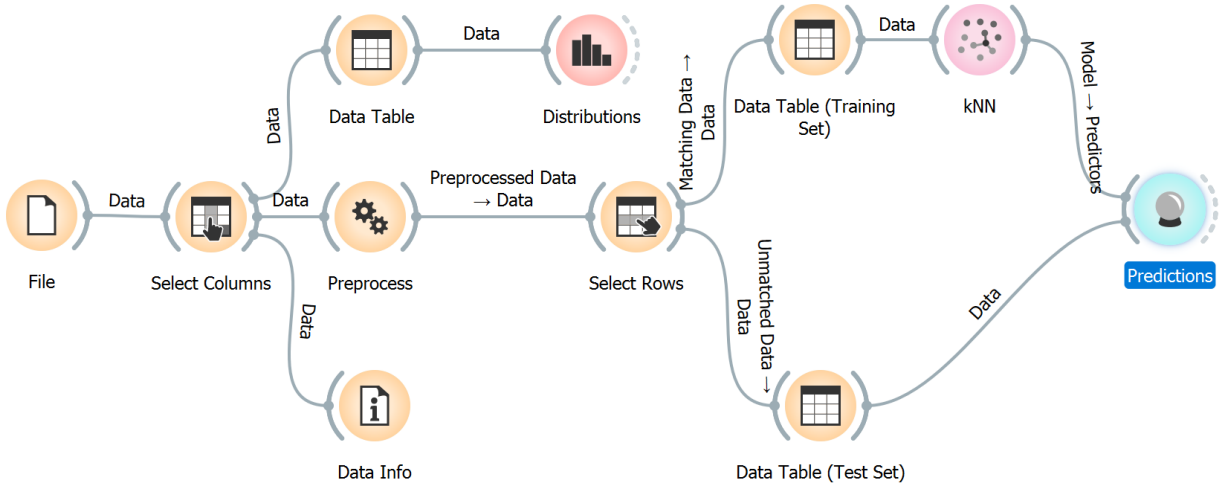
12. Add two instances of **Data Table** as shown below



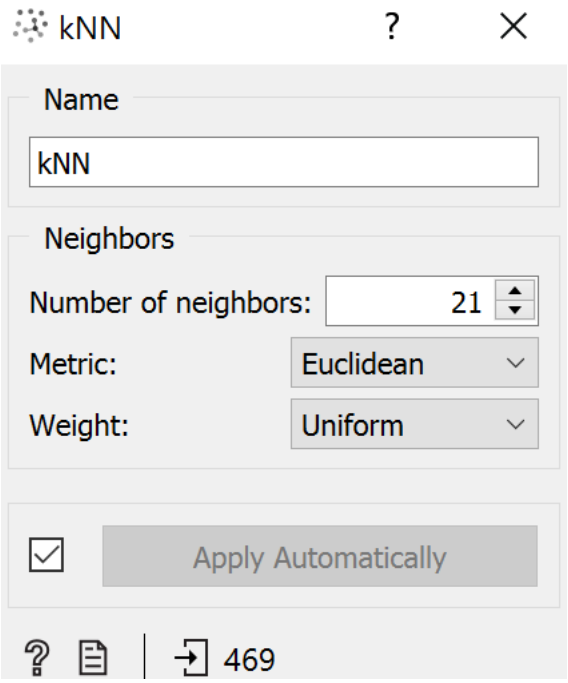
13.Complete the following table.

Size of the training Samples	469
Size of the test Samples	100

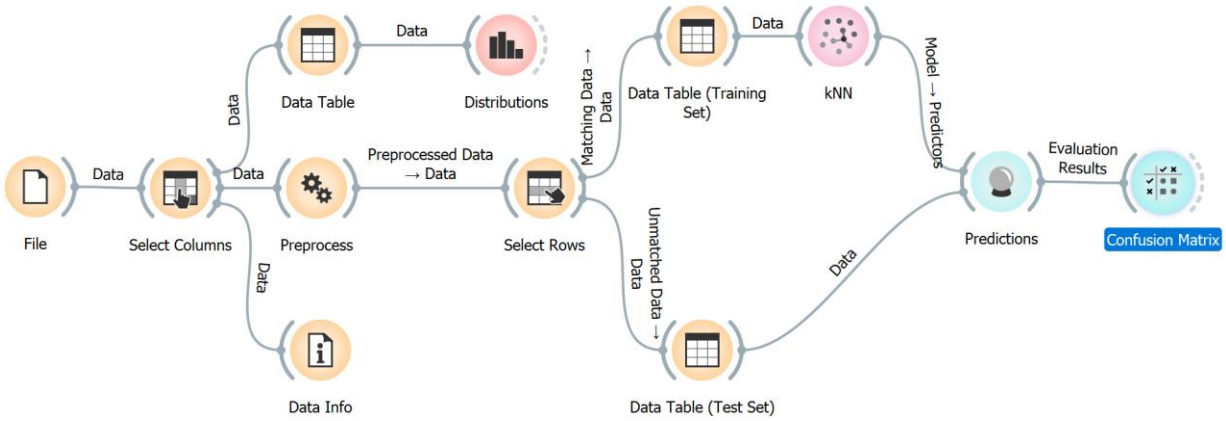
14.Add the **kNN** and **Predictions** widgets as shown below.



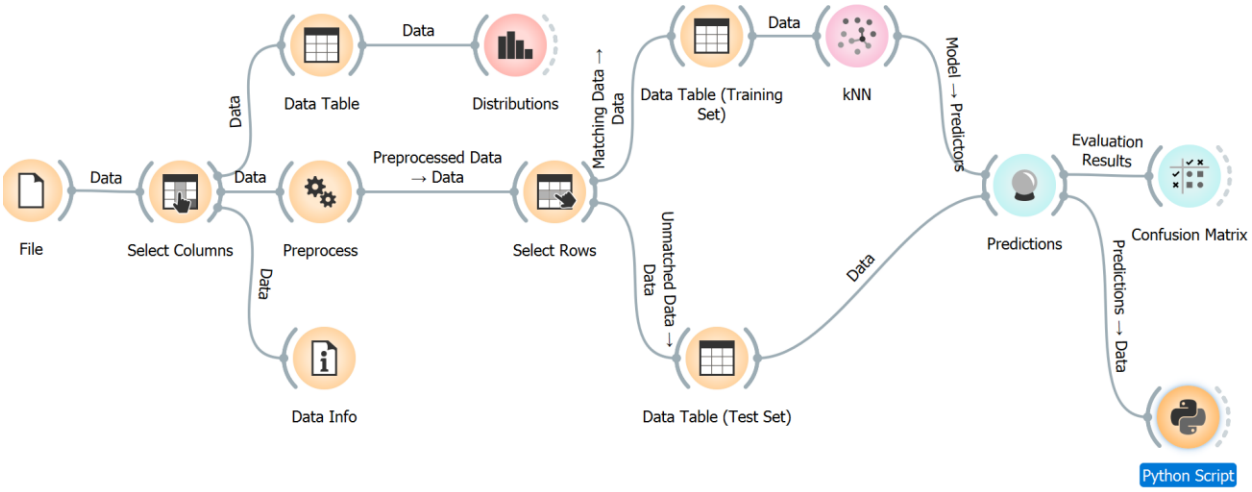
15.Double click on **kNN** and change the settings as shown below.



16.Add the **Confusion Matrix** widget as shown below.



17.Add the Python Script widget as shown below.



18. Double click on **Python Script**. Copy the following code and paste it in the script window.

```
import Orange
import numpy as np
import pandas as pd

test = in_data.copy()

kNN_prediction = test.metas[:,3]
#Tree_prediction = test.metas[:,5]
Actual = test.Y
```

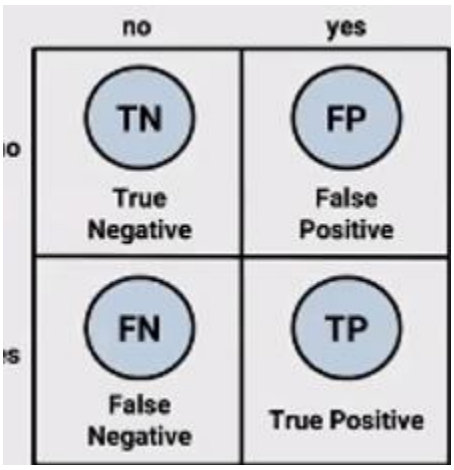
```
from sklearn.metrics import accuracy_score, cohen_kappa_score,
classification_report

#Accuracy
CA_kNN = accuracy_score(Actual, kNN_prediction)
#kappa
kappa_kNN = cohen_kappa_score(Actual, kNN_prediction)

learner = ['kNN'] #, 'Tree'
metrics = {'CA': [CA_kNN], #,CA_Tree
        'kappa': [kappa_kNN] } #,kappa_Tree

performance_summary = pd.DataFrame(metrics, index=learner)

print('Performance summary')
print(performance_summary)

# precision recall and f1
print(classification_report(Actual, kNN_prediction))

#Note that in binary classification,
# 1. recall of the positive class is also known as “sensitivity”;
# 2. recall of the negative class is “specificity”.
```

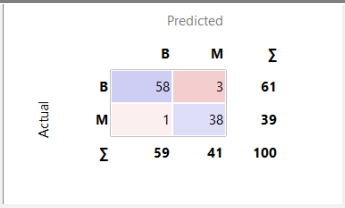
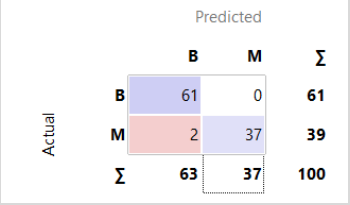
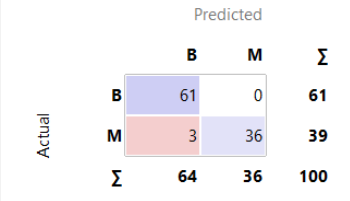
19. Complete the following table.

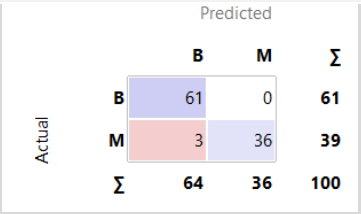
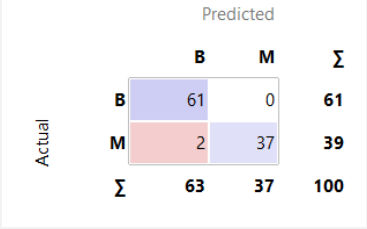
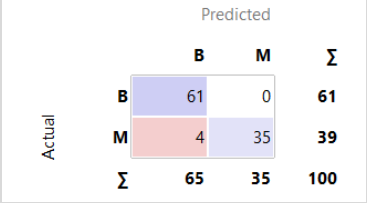
Comment on the CA and kappa values.	CA value 0.98 and kappa value 0.957573; High kappa value indicates almost perfect agreement between the model’s prediction and the true values

XXXXXXXXXX Model is Ready → Now let’s understand the model XXXXXXXXXX

20. Complete the following table. Observe and note in the comment the Accuracy & F1 metrics values for each of the models.



	k	True Negative TN (B→B) Top left	True Positive TP (M→M) Bottom Right	False Positive FP (B→M)	False Negative FN (M→B)	Error (Sum of M's bottom left + top right)	Comments
M1	1	58	38	3	1	3+1 = 4	 58 and 38 correct classifications 38+58 = 96 96 / 100 = 98% accuracy
M2	5	61	37	0	2	2	 (37+61) / 100 = 98%
M3	11	61	36	0	3	3	

							(36+61)/100 = 97%
M4	15	61	36	0	3	3	 (36+61)/100 = 97%
M5	21	61	37	0	2	2	 (37+61)/100 = 98%
M6	27	61	35	0	4	4	 (35+61)/100 = 96%

21.Answer the following question

Which model are you going to put in production? Why?
We would want the one with least amount of false negative, so we would want to use M1 with the k value of 1, for now (less if there is overfitting, etc.)