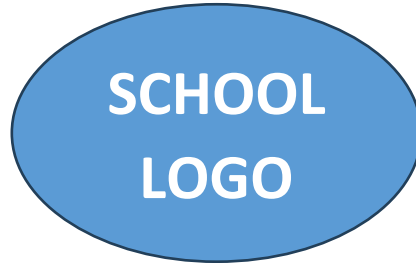


**SCHOOL NAME**

**SCHOOL ADDRESS**



**PRACTICAL FILE 2026-27**

**ARTIFICIAL INTELLIGENCE (417)**

**Submitted to:**

**Teacher's Name**

**Submitted to: -**

**Student Name**

**Class - X**

**Roll no**

# CERTIFICATE

This is to certify that **Student Name** of Class **X – (Section)** has completed his/her practical file of the subject **Artificial Intelligence** under the guidance of Artificial Intelligence Teacher (**Teacher's Name**) during the year **2026-27**.  
This project is submitted as part of the complete fulfillment of the syllabus prescribed by the CBSE Board.

The work done by the student is original and has been carried out with sincerity and dedication.

**Internal Examiner**

**External Examiner**

**Principal's Signature**

# **ACKNOWLEDGEMENT**

I express my sincere gratitude to the Almighty for His blessings and guidance throughout the completion of this project.

I am deeply thankful to our respected Principal, **(Principal's Name)**, and my Artificial Intelligence teacher, **(Teacher's Name)**, for their invaluable guidance, encouragement, and support. Their motivation and constructive suggestions helped me complete this file successfully.

This project, prepared as per the CBSE curriculum, has enhanced my knowledge and understanding of Artificial Intelligence and its practical applications.

I would also like to thank my parents for their constant encouragement and unwavering support. Finally, I extend my gratitude to everyone who contributed directly or indirectly to the successful completion of this project.

Thank you all for making this learning journey meaningful and enriching.

# INDEX

SNO	OBJECTIVE	DATE	T. SIGN
1	Write a program to calculate mean, median, mode using numpy. If the list is follows: [80,100,50,25,65,72,91,101,207,250,132,631,44,77,88]		
2	Write a program to display a line chart from (3,6) to (14,15).		
3	Read the CSV file saved in your system and display the top two records, bottom two records. Also display its information.		
4	Write a program to read an image and display in grey color using Python.		
5	Write a program to create a pie chart using following data: Labels : English, Hindi, Science Marks : 50, 25, 25		
6	Write a program to display a scatter chart for the following points: (1,2) , (3,7), (5,4) , (7,9) , (9,6)		
7	Write a program to add two elements of lists. L1 = [2,4,6,8] L2 = [1,7,2,4]		
8	Write a program to input a string and display it in reverse order.		
9	Write a program to input a string and display the count of consonants and vowels in the string.		
10	Write a program to input three sides of a triangle and display whether the triangle can be formed with the inputs or not. If the triangle can be formed if the triangle can be scalene, isosceles and equilateral.		
11	Write a program to input a number whether it is palindrome or not.		
12	Write a program to input a number whether it is prime or not.		

13	Write a program to input two numbers and swap both the numbers using the third number.		
14	Using Numpy package, a. Create a 4 X 2 array with random integers. b. Create a 3 X 3 array with all zeros.		
15	Write a menu driven program to display sum and product of three numbers as per user's choice.		

## PROJECT QUESTIONS

1	Using the Palmer Penguins dataset in Orange Data Mining, develop and evaluate an AI model that applies statistical analysis and machine learning techniques to predict the species of a penguin based on features such as bill length, bill depth, flipper length, and body mass.		
2	Using Orange Data Mining, design and develop a computer vision-based AI system that analyzes underwater images to identify and classify coral reefs as healthy or bleached, enabling the early detection of coral bleaching for marine ecosystem monitoring and conservation.		
3	Develop an AI-based Natural Language Processing (NLP) model using Orange Data Mining to classify book excerpts into Children's Literature or Adult Literature based on their textual content.		

# PROGRAM 1

Write a program to calculate mean, median, mode using numpy. If the list is follows:

```
[80,100,50,25,65,72,91,101,207,250,132,631,44,77,88]
```

## CODE

```
import numpy as np
from statistics import mode

# Creating the list
numbers=[80,100,50,25,65,72,91,101,207,250,132,631,44,77,88]

# Mean
mean_value = np.mean(numbers)

# Median
median_value = np.median(numbers)

# Mode
mode_value = mode(numbers)

# Displaying results
print("Mean =", mean_value)
print("Median =", median_value)
print("Mode =", mode_value)
```

## OUTPUT

```
Mean = 134.2
```

```
Median = 88.0
```

```
Mode = 80
```

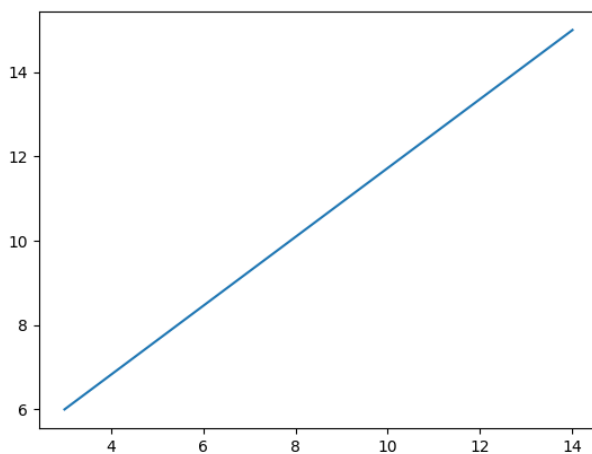
## PROGRAM 2

Write a program to display a line chart from (3,6) to (14,15).

### CODE

```
import matplotlib.pyplot as plt
# X-axis values
x = [3, 14]
# Y-axis values
y = [6, 15]
# Plotting line chart
plt.plot(x, y)
# Displaying chart
plt.show()
```

### OUTPUT



# PROGRAM 3

Read the CSV file saved in your system and display the top two records, bottom two records. Also display its information.

## CODE

```
import pandas as pd
# Reading the CSV file
df = pd.read_csv("STUDENT.csv")
# Display top 2 records
print("Top Two Records")
print(df.head(2))
# Display bottom 2 records
print("\nBottom Two Records")
print(df.tail(2))
print("\nInformation of CSV File")
df.info()
```

## OUTPUT

Top Two Records

	RNo	Name	Marks
0	1	Aman	85
1	2	Riya	90

Bottom Two Records

	RNo	Name	Marks
8	9	Vikas	91
9	10	Anjali	73

Information of CSV File

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 10 entries, 0 to 9
```

```
Data columns (total 3 columns)
```

# PROGRAM 4

Write a program to read an image and display in grey color using Python.

## CODE

```
import cv2

# Read the image
img = cv2.imread("image.png")

# Resize image
img = cv2.resize(img, (500, 400))

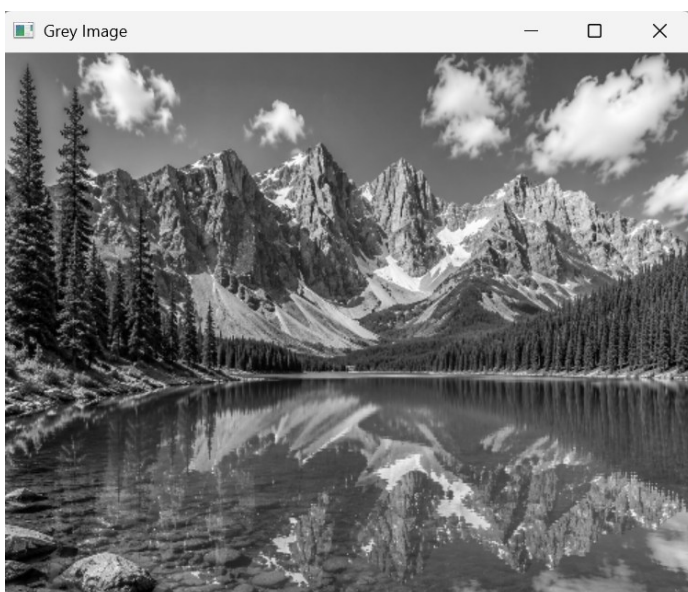
# Convert image into grey color
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Display the grey image
cv2.imshow("Grey Image", gray)

# Wait until a key is pressed
cv2.waitKey(0)

# Close all windows
cv2.destroyAllWindows()
```

## OUTPUT



# PROGRAM 5

Write a program to create a pie chart using following data:

Labels : English, Hindi, Science

Marks : 50, 25, 25

## CODE

```
import matplotlib.pyplot as plt

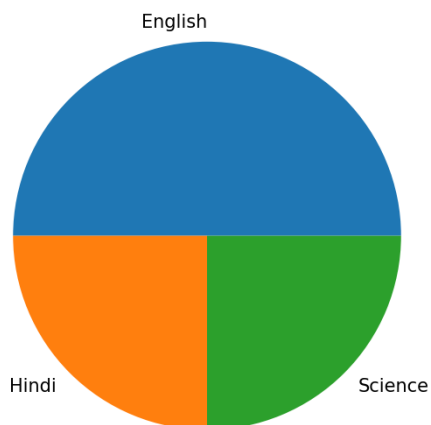
# Data
labels = ["English", "Hindi", "Science"]
marks = [50, 25, 25]

# Create pie chart
plt.pie(marks, labels=labels)

# Display chart
plt.show()
```

## OUTPUT

Figure 1



Navigation icons: Home, Back, Forward, Search, and Print.

# PROGRAM 6

Write a program to display a scatter chart for the following points:

(1,2) , (3,7) , (5,4) , (7,9) , (9,6)

## CODE

```
import matplotlib.pyplot as plt
```

```
# X-axis values  
x = [1, 3, 5, 7, 9]
```

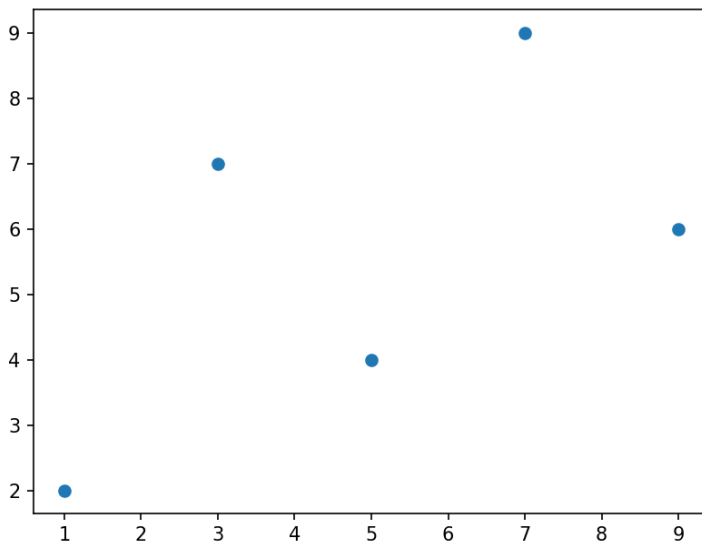
```
# Y-axis values  
y = [2, 7, 4, 9, 6]
```

```
# Create scatter chart  
plt.scatter(x, y)
```

```
# Display chart  
plt.show()
```

## OUTPUT

Figure 1



Home Back Forward Pan Zoom Save

# PROGRAM 7

Write a program to concatenate two elements of lists.

L1 = [2,4,6,8]

L2 = [1,7,2,4]

## CODE

```
# Creating lists
```

```
L1 = [2,4,6,8]
```

```
L2 = [1,7,2,4]
```

```
# Concatenating lists
```

```
result = L1 + L2
```

```
# Display result
```

```
print("Concatenated List =", result)
```

## OUTPUT

Concatenated List = [2, 4, 6, 8, 1, 7, 2, 4]

# PROGRAM 8

Write a program to input a string and display it in reverse order.

## CODE

```
# Input string
text = input("Enter a string: ")
```

## OUTPUT

```
# Display string in reverse order
print("Reversed String =", text[::-1])
```

```
Enter a string: Good Morning
Reversed String = gninroM dooG
```

# **PROGRAM 9**

Write a program to input a string and display the count of consonants and vowels in the string.

## **CODE**

```
# Input string
text = input("Enter a string: ")

# Initialize counters
vowels = 0
consonants = 0

# Check each character
for ch in text.lower():
    if ch.isalpha():
        if ch in "aeiou":
            vowels += 1
        else:
            consonants += 1

# Display result
print("Number of Vowels =", vowels)
print("Number of Consonants =", consonants)
```

## **OUTPUT**

```
Enter a string: Hello, How are you ?
Number of Vowels = 7
Number of Consonants = 7
```

# PROGRAM 10

Write a program to input three sides of a triangle and display whether the triangle can be formed with the inputs or not. If the triangle can be formed if the triangle can be scalene, isosceles and equilateral.

## CODE

```
# Input sides of triangle
a = int(input("Enter first side: "))
b = int(input("Enter second side: "))
c = int(input("Enter third side: "))
# Check whether triangle can be formed
if (a + b > c) and (a + c > b) and (b + c > a):
    print("Triangle can be formed")
    # Check type of triangle
    if a == b == c:
        print("It is an Equilateral Triangle")
    elif a == b or b == c or a == c:
        print("It is an Isosceles Triangle")
    else:
        print("It is a Scalene Triangle")
else:
    print("Triangle cannot be formed")
```

## OUTPUT

Enter first side: 5

Enter second side: 5

Enter third side: 5

Triangle can be formed

It is an Equilateral Triangle

# PROGRAM 11

Write a program to input a number whether it is palindrome or not.

## CODE

```
# Input number
num = input("Enter a number: ")

# Reverse the number
rev = num[::-1]

# Check palindrome
if num == rev:
    print("Palindrome Number")
else:
    print("Not a Palindrome Number")
```

## OUTPUT

Enter a number: 252

Palindrome Number

Enter a number: 234

Not a Palindrome Number

# **PROGRAM 12**

Write a program to input a number whether it is prime or not.

## **CODE**

```
# Input number
num = int(input("Enter a number: "))
# Assume number is prime
prime = True
# Check prime number
if num <= 1:
    prime = False
else:
    for i in range(2, num):
        if num % i == 0:
            prime = False
            break
# Display result
if prime:
    print("Prime Number")
else:
    print("Not a Prime Number")
```

## **OUTPUT**

```
Enter a number: 77
Not a Prime Number
Enter a number: 5
Prime Number
```

# **PROGRAM 13**

Write a program to input two numbers and swap both the numbers using the third number.

## **CODE**

```
# Input two numbers
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
# Display numbers before swapping
print("Before Swapping")
print("a =", a)
print("b =", b)
# Swapping using third variable
temp = a
a = b
b = temp
# Display numbers after swapping
print("After Swapping")
print("a =", a)
print("b =", b)
```

## **OUTPUT**

```
Enter first number: 56
Enter second number: 12
Before Swapping
a = 56
b = 12
After Swapping
a = 12
b = 56
```

# PROGRAM 14

Using Numpy package,

- Create a 4 X 2 array with random integers.
- Create a 3 X 3 array with all zeros.

## CODE

```
import numpy as np

# a. Create a 4 X 2 array with random integers

arr1 = np.random.randint(1, 100, (4, 2))

print("4 X 2 Random Integer Array")

print(arr1)

# b. Create a 3 X 3 array with all zeros

arr2 = np.zeros((3, 3), dtype=int)

print("\n3 X 3 Zero Array")

print(arr2)
```

## OUTPUT

4 X 2 Random Integer Array

```
[[99 38]
```

```
[34 33]
```

```
[86 45]
```

```
[89 18]]
```

3 X 3 Zero Array

```
[[0 0 0]
```

```
[0 0 0]
```

```
[0 0 0]]
```

# PROGRAM 15

Write a menu driven program to display sum and product of three numbers as per user's choice.

## CODE

```
while True:
    # Display menu
    print("\nMENU")
    print("1. Sum")
    print("2. Product")
    print("3. Exit")
    # User choice
    choice = int(input("Enter your choice: "))
    if choice == 3:
        print("Program Ended")
        break
    # Input numbers
    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    c = int(input("Enter third number: "))
    # Perform operation
    if choice == 1:
        print("Sum =", a + b + c)
    elif choice == 2:
        print("Product =", a * b * c)
    # Exit condition
    else:
```

```
print("Invalid Choice")
```

## **OUTPUT**

MENU

1. Sum

2. Product

3. Exit

Enter your choice: 1

Enter first number: 11

Enter second number: 7

Enter third number: 2

Sum = 20

MENU

1. Sum

2. Product

3. Exit

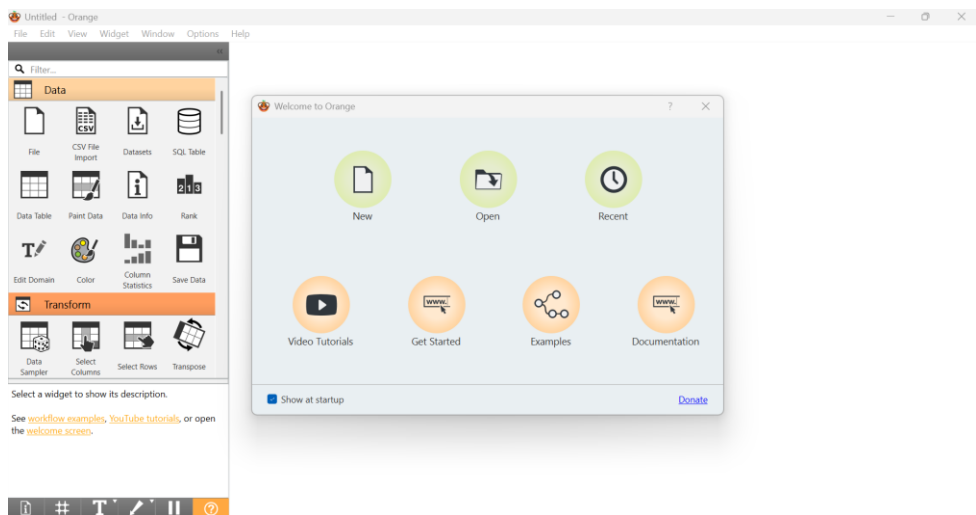
Enter your choice: 3

Program Ended

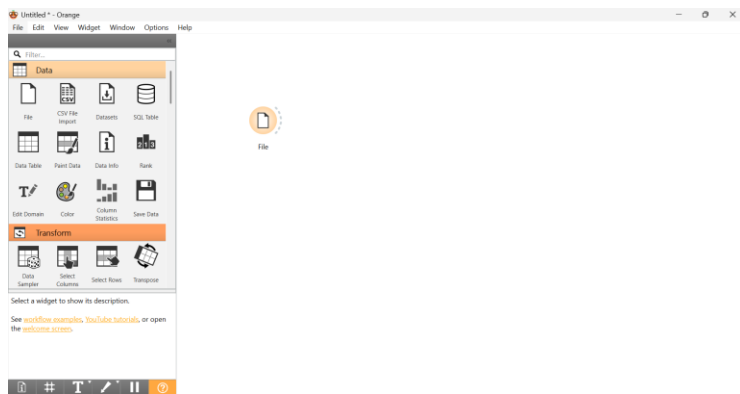
# PROJECT 1

Using the Palmer Penguins dataset in Orange Data Mining, develop and evaluate an AI model that applies statistical analysis and machine learning techniques to predict the species of a penguin based on features such as bill length, bill depth, flipper length, and body mass.

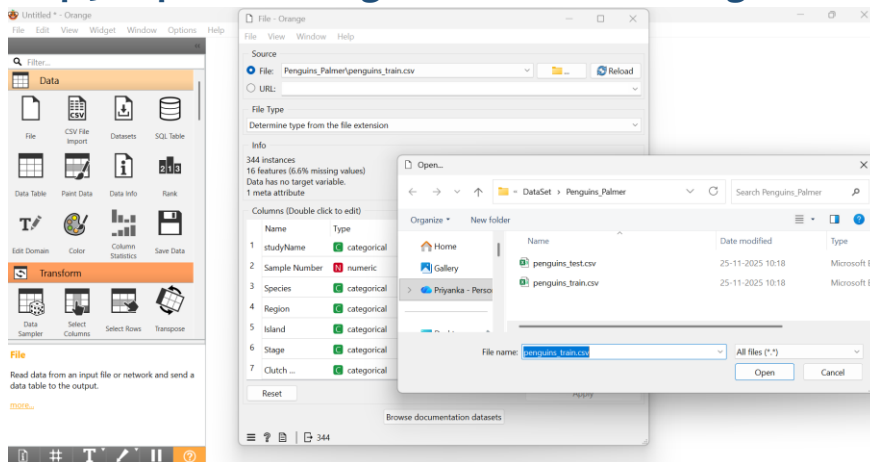
**Step 1 : Open Orange Data mining software. Click on New File.**



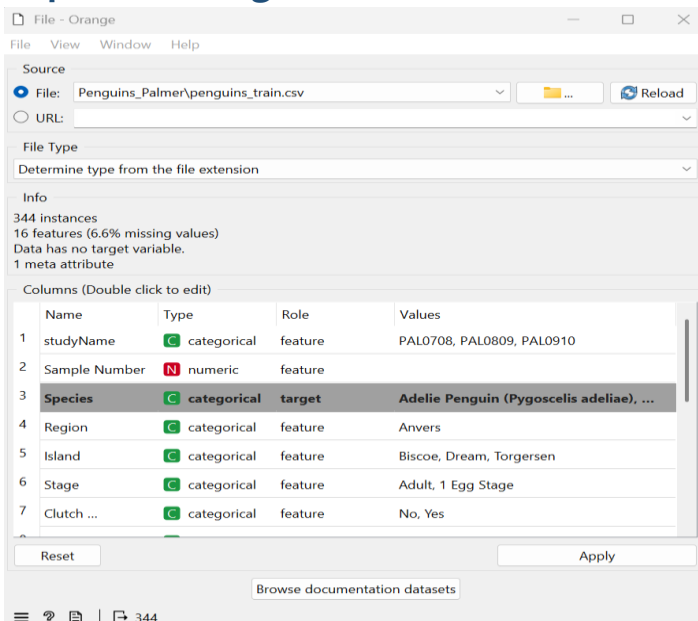
**Step 2 : Click on File widget. Than double click on it to upload the dataset.**



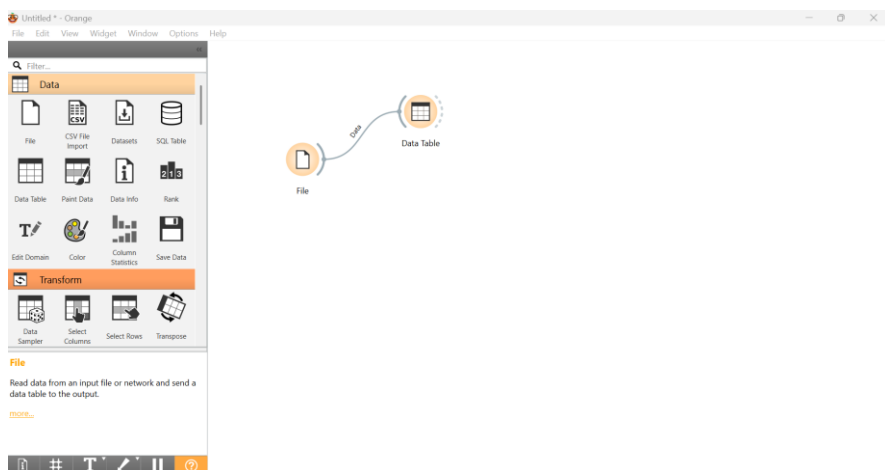
**Step 3 : Upload training dataset from the Penguins-Palmer folder.**



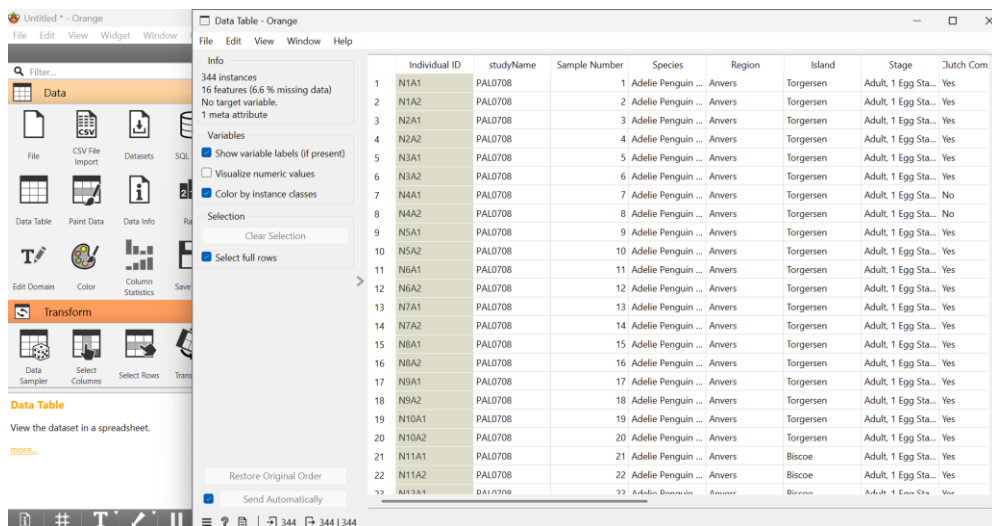
## Step 4 : Change Role of Species to Target.



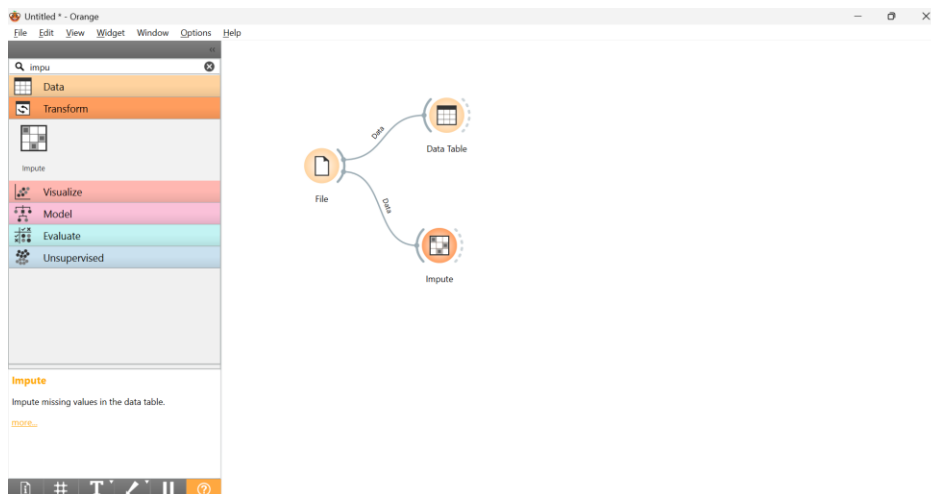
## Step 5 : Drag Data table widget and connect output of File widget to input of data Table widget.



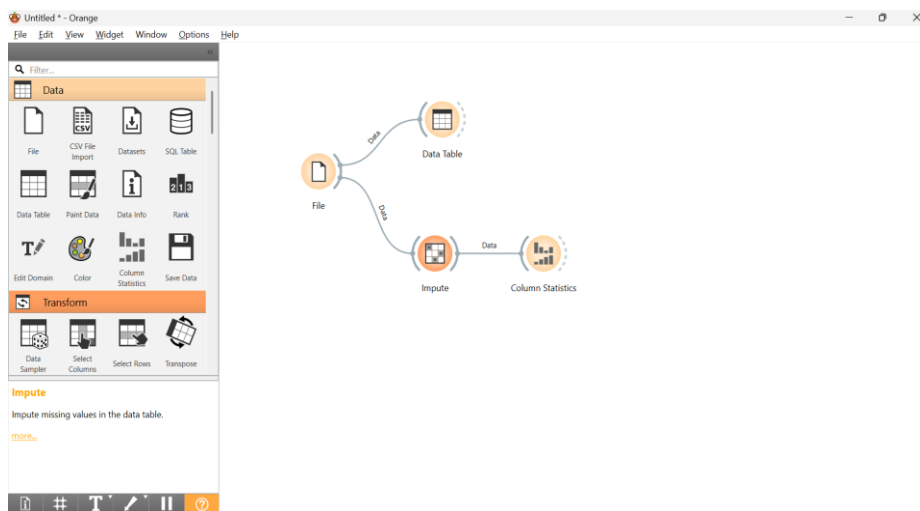
## Step 6 : Double click on Data Table widget and can scroll all three categories of penguins with 344 records



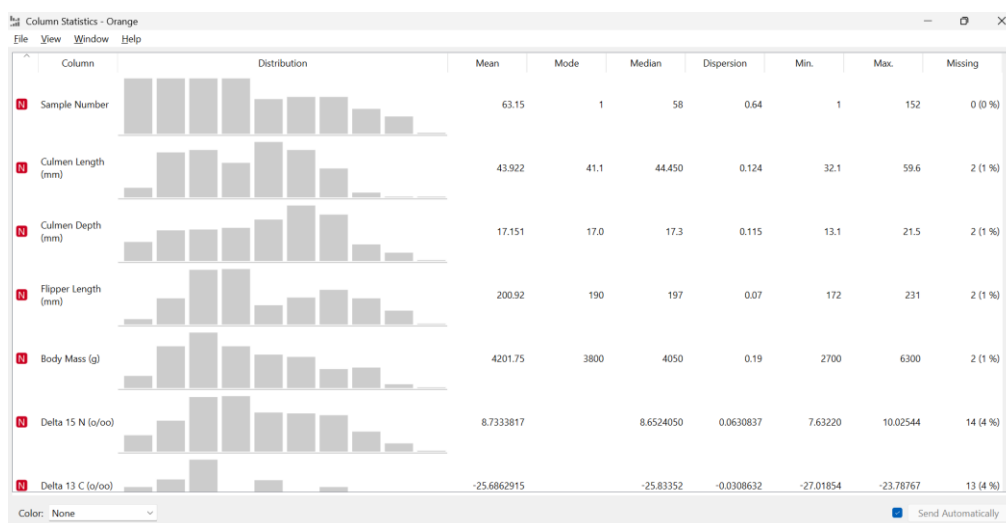
## Step 7 : Drag Impute widget inside Transform menu to fill missing values. Connect the output of File widget to input of Impute widget.



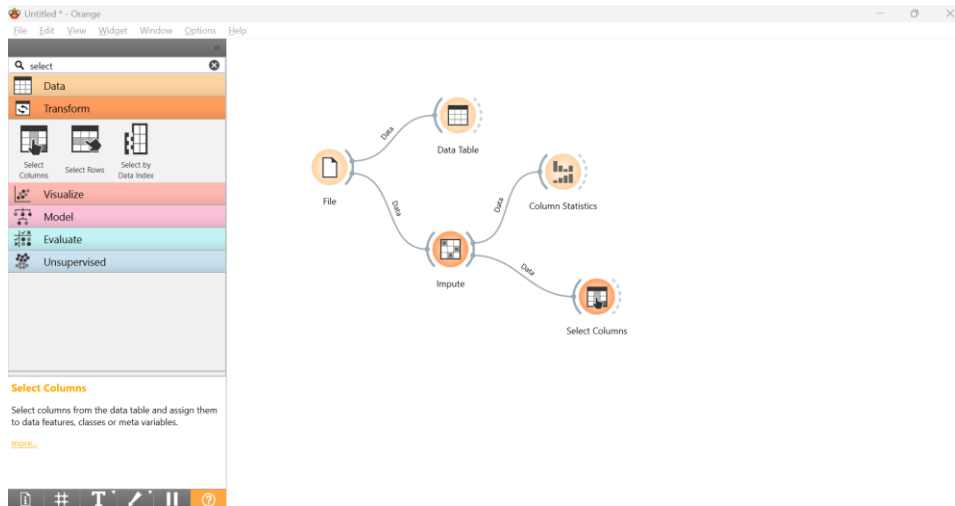
## Step 8 : Connect Column Statistics widget to the output of Impute Widget.



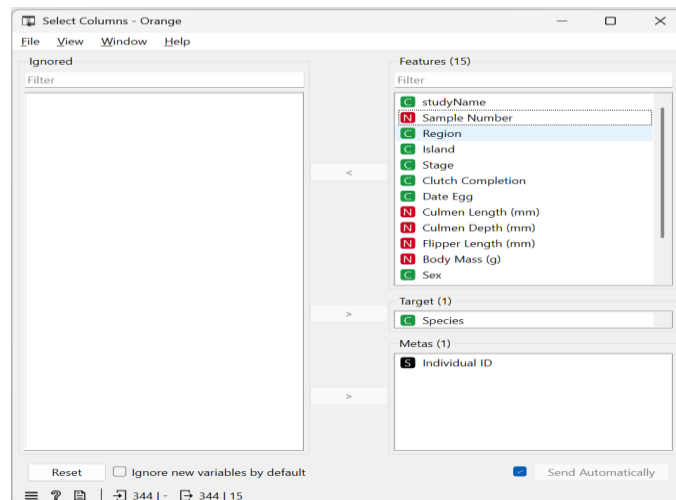
## Step 9 : Click on Column Statistics widget.



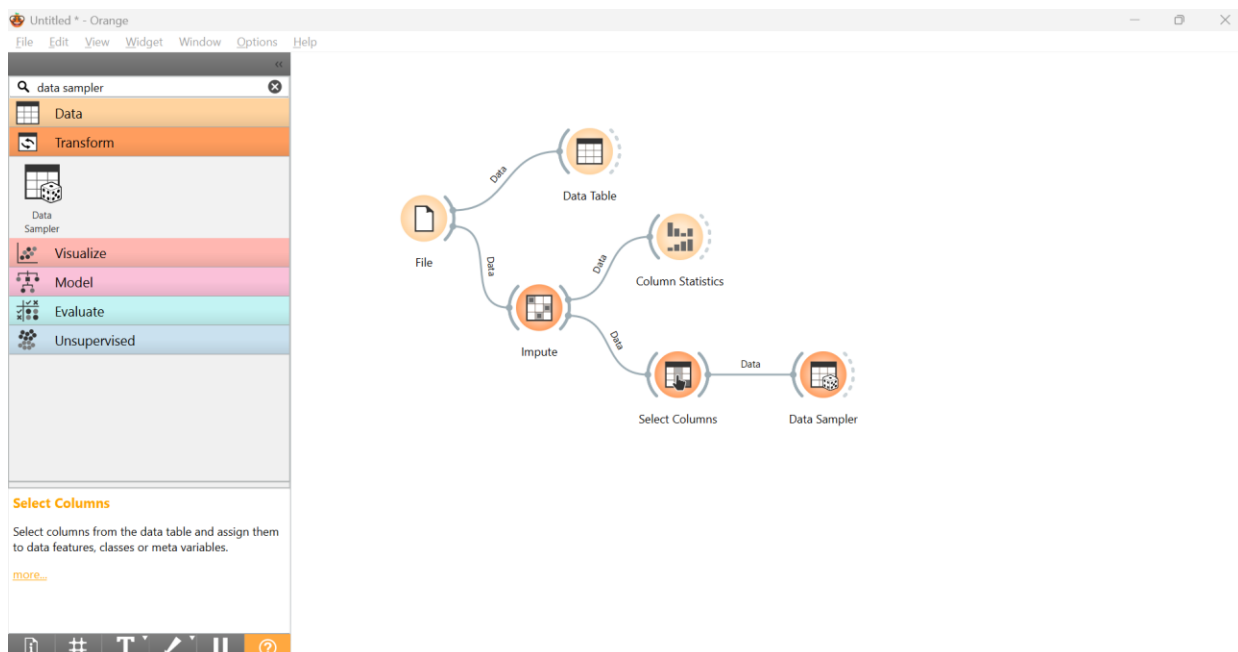
## Step 10 : Drag Select columns widget to connect to the output of Impute widget.



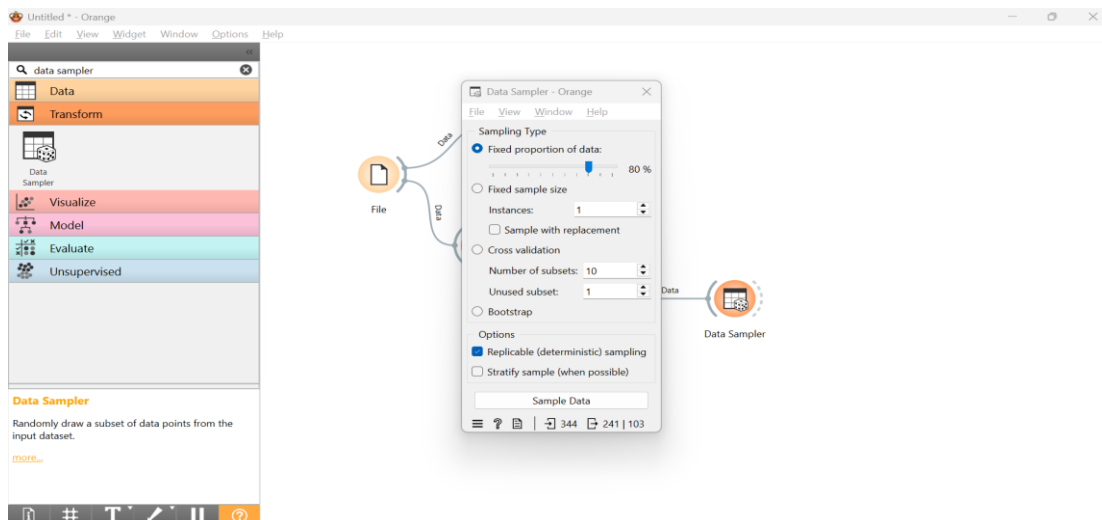
## Step 11: Double click and set Species to Target , if not already done.



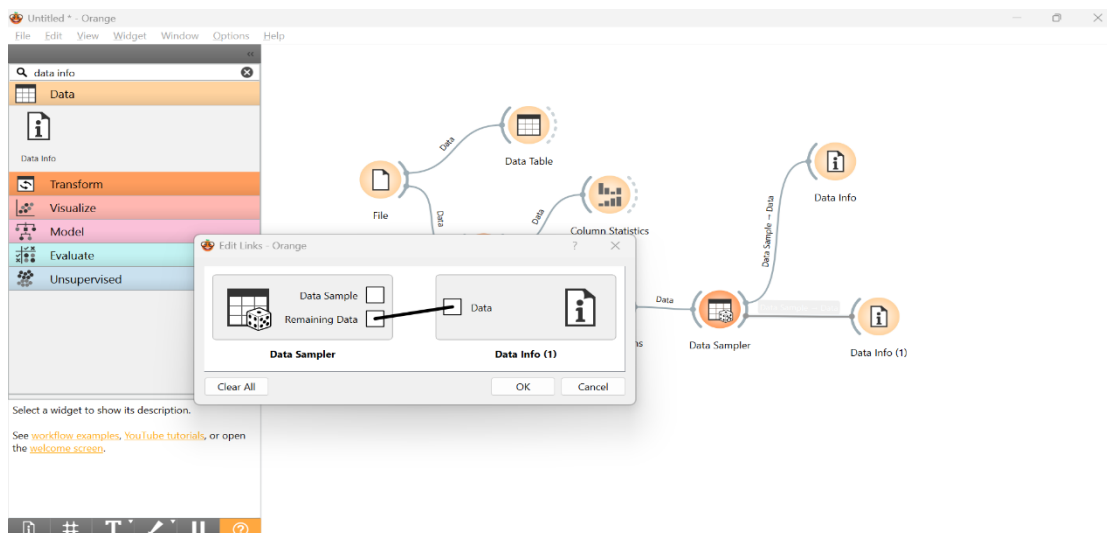
## Step 11: Connect Data Sampler widget to split training and testing data



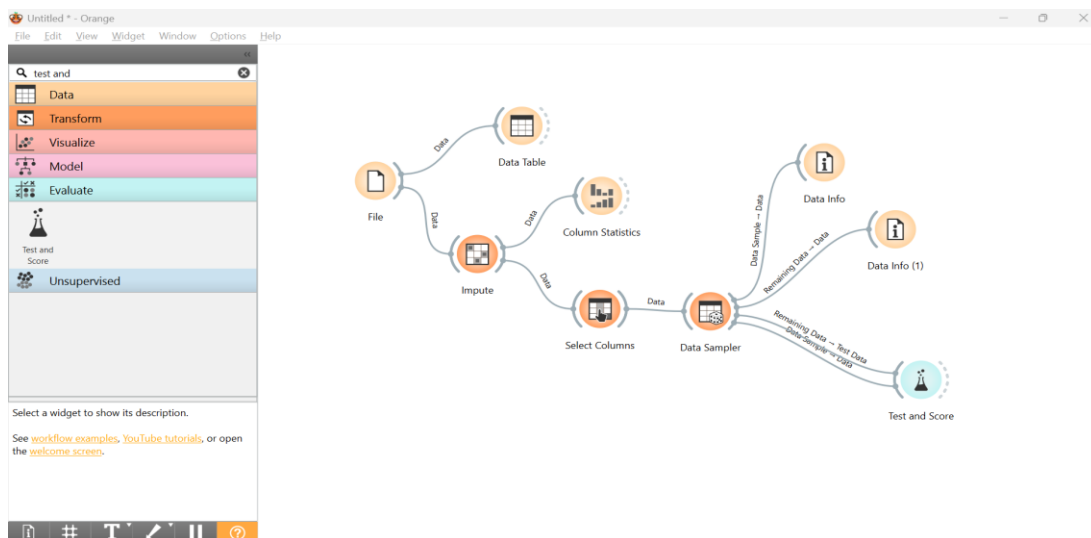
## Step 12: Click on DataSampler to change fixed proportion to 80%



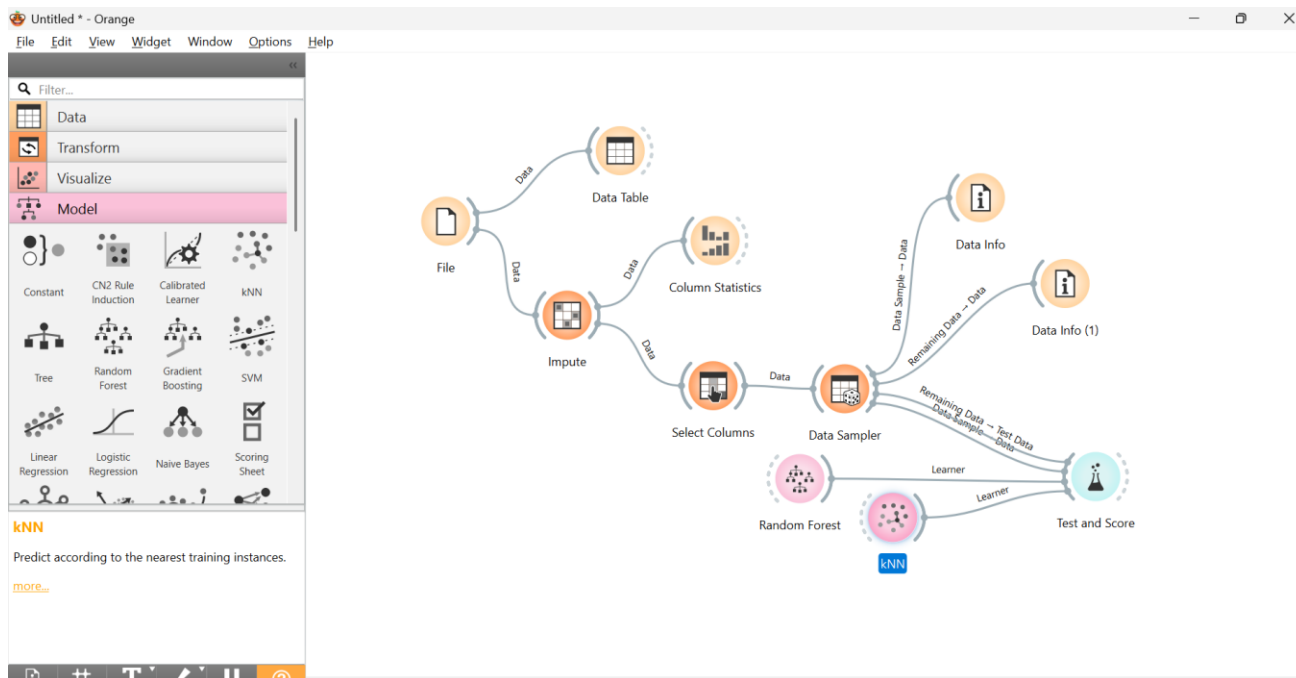
## Step 13: Connect remaining data with data for second Data Info after clicking on its link.



## Step 14 : Drag Test and Score widget and connect twice to the output of Data Sampler widget. Click on any one link and connect data remaining data with test data.



## Step 15 : Train with Random Forest and KNN model by connecting both with test and score widget.



## Step 16 : Double click on Test and Score widget and compare the values of AUC, CA, F1, Precision, Recall, MCC

The 'Test and Score' widget configuration window shows the following settings and results:

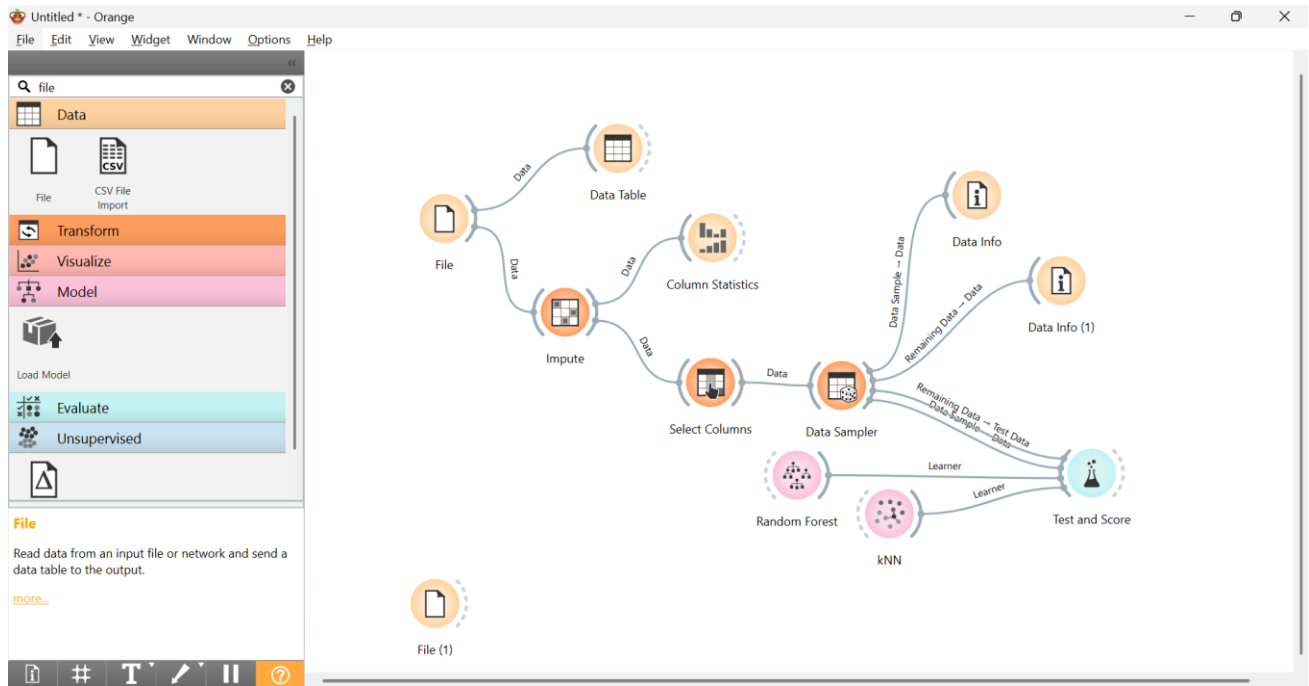
- Cross validation:**
  - Number of folds: 5
  - Stratified:
  - Cross validation by feature:
  - Random sampling:
  - Repeat train/test: 10
  - Training set size: 66 %
  - Stratified:
  - Leave one out:
  - Test on train data:
  - Test on test data:
- Evaluation results for target (None, show average over classes):**

Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest	0.999	0.983	0.983	0.984	0.983	0.974
kNN	0.895	0.755	0.748	0.745	0.755	0.608
- Compare models by:** Area under ROC curve
- Negligible diff.:** 0.1
- Comparison Table:**

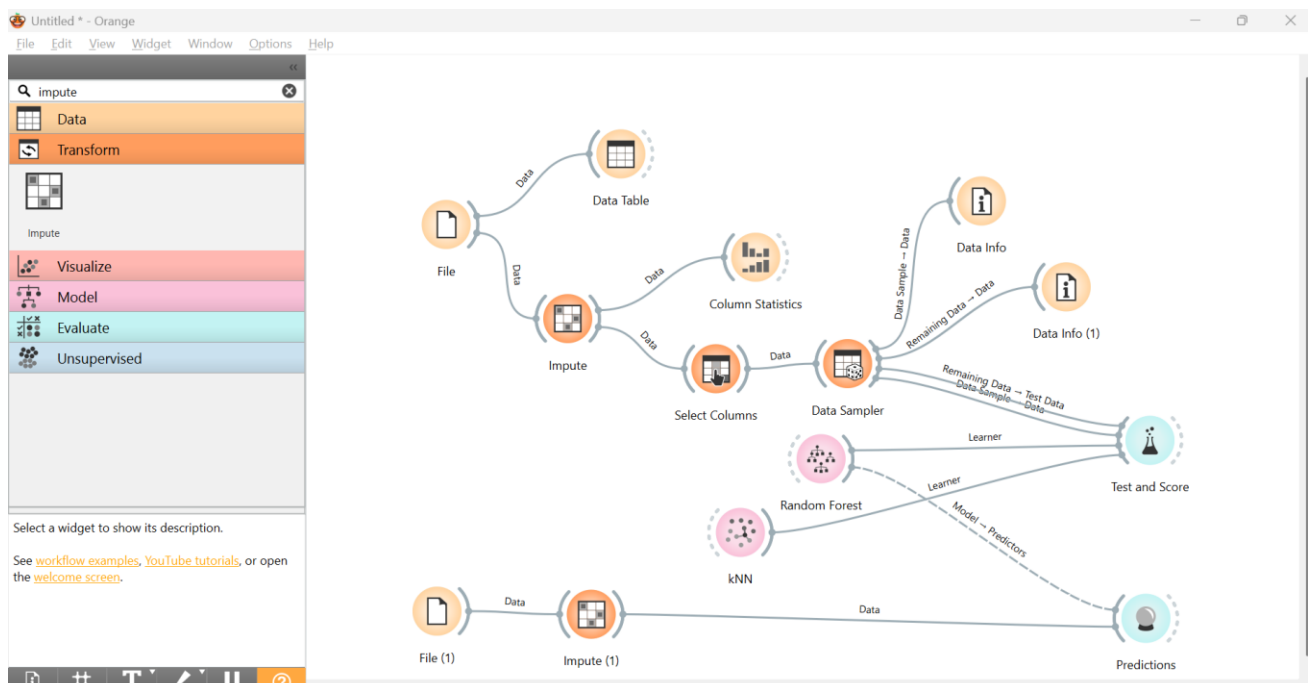
	Random For...	kNN
Random Forest		0.999
kNN	0.001	

Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.

## Step 17 : Add File widget for testing data. Check species column should be target.



## Step 18: Connect the output of second file widget with Impute widget for missing values. Than connect with the Predictions widget. Also, connect any one or most appropriate model with Predictions Widget.



## Step 19: Check the values of Prediction widget by double clicking on it.

The screenshot shows the Orange data mining software interface. On the left is the widget palette with categories: Data, Transform, Impute, Visualize, Model, Evaluate, and Unsupervised. The main workspace contains a workflow with 'File (1)', 'Impute (1)', and 'Predictions' widgets. The 'Predictions' widget window is open, displaying a table of data and options for classification errors and performance scores.

**Predictions - Orange**

File View Window Help

Show probabilities for: Classes in data  Show classification errors [Restore Original Order](#)

actual species	Island	ulmen_length_mm	ulmen_depth_mm	lipper_length_mm	body_mass_g
Adelie	Torgersen	39.1	18.7	181	3750
Adelie	Torgersen	39.5	17.4	186	3800
Adelie	Torgersen	40.3	18.0	195	3250
Adelie	Torgersen	43.922	17.151	200.92	4201.75
Adelie	Torgersen	36.7	19.3	193	3450
Adelie	Torgersen	39.3	20.6	190	3650
Adelie	Torgersen	38.9	17.8	181	3625
Adelie	Torgersen	39.2	19.6	195	4675
Adelie	Torgersen	34.1	18.1	193	3475
Adelie	Torgersen	42.0	20.2	190	4250
Adelie	Torgersen	37.8	17.1	186	3300

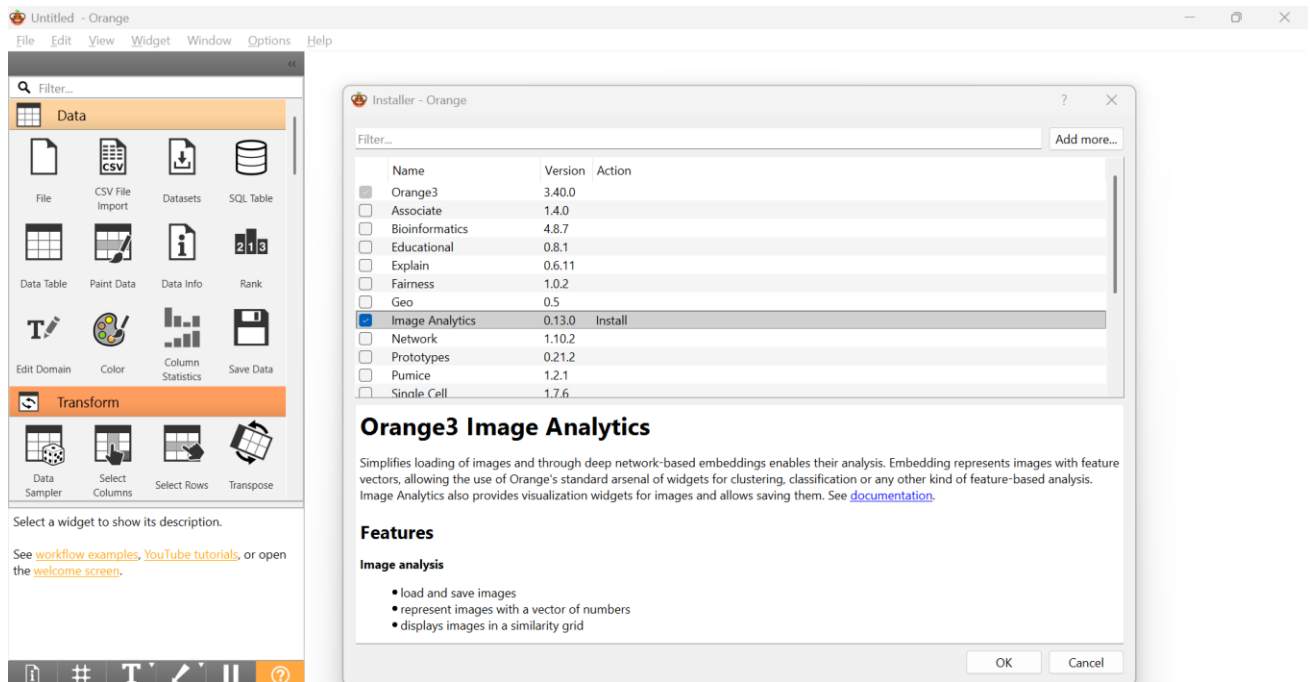
Show performance scores Target class: Chinstrap

File (1) Impute (1) Predictions

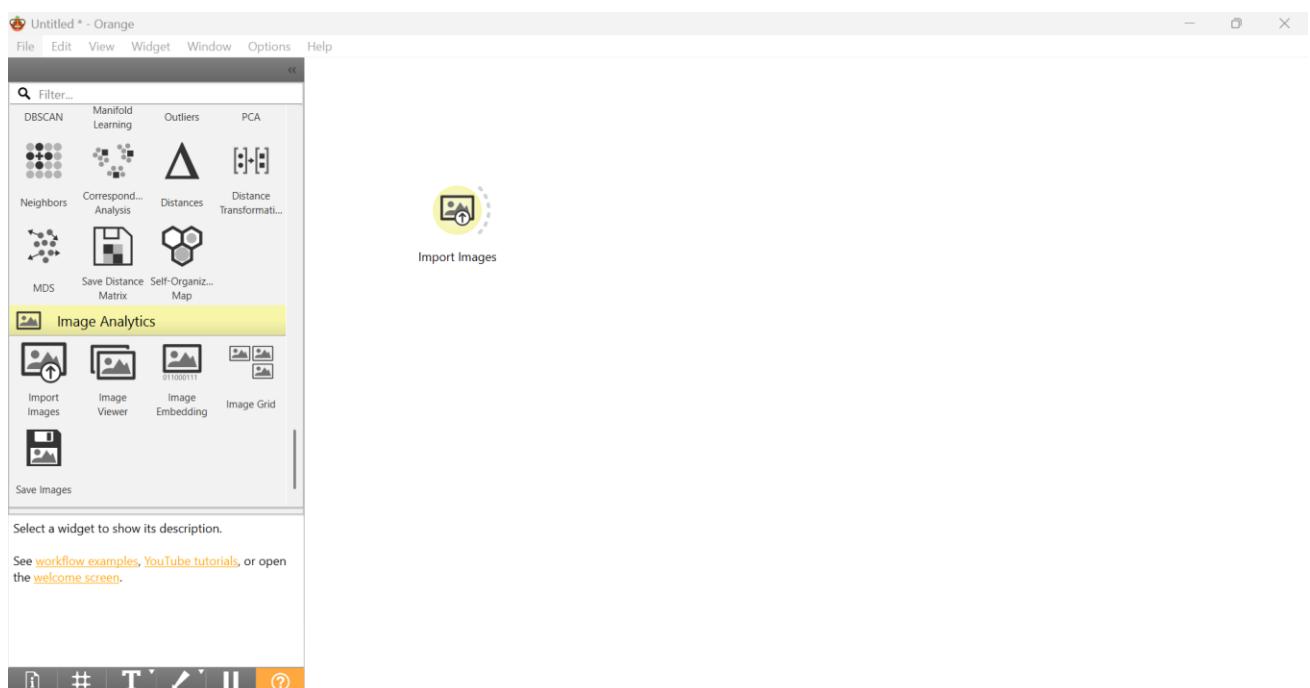
# PROJECT 2

Using Orange Data Mining, design and develop a computer vision-based AI system that analyzes underwater images to identify and classify coral reefs as healthy or bleached, enabling the early detection of coral bleaching for marine ecosystem monitoring and conservation.

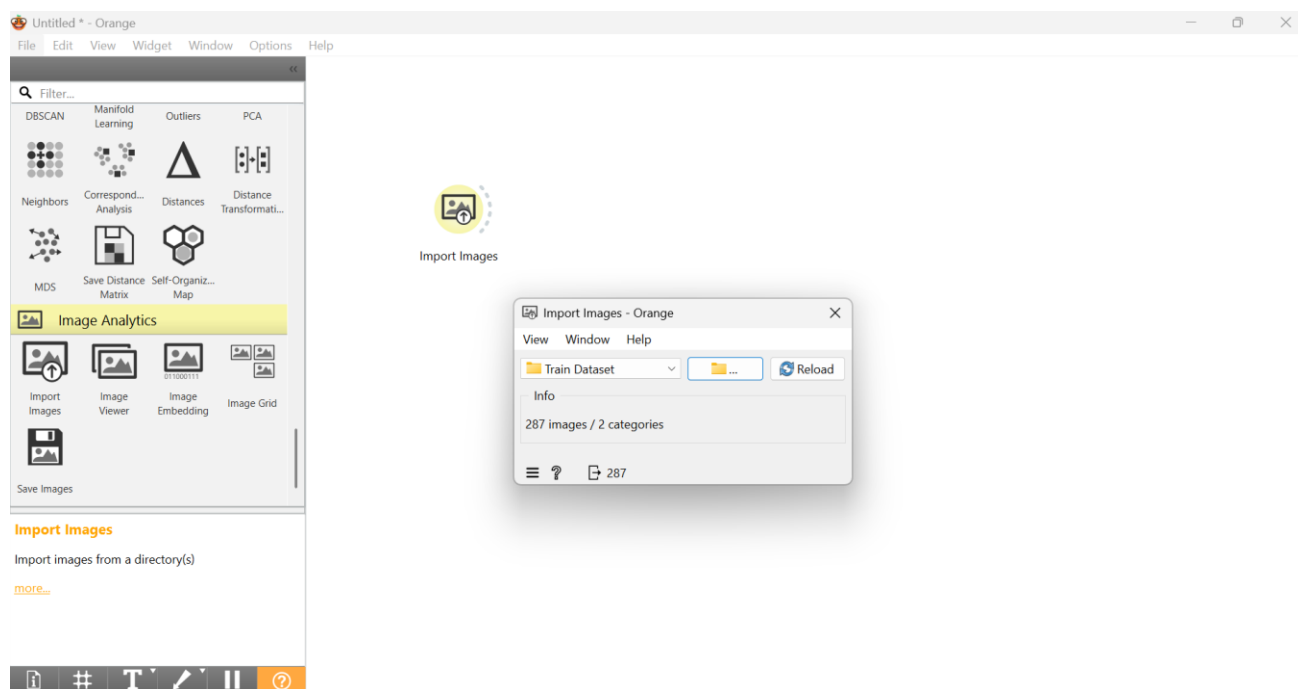
**Step 1: Click on Options -> Add On -> Image Analytics . Restart Orange for the changes to take effect.**



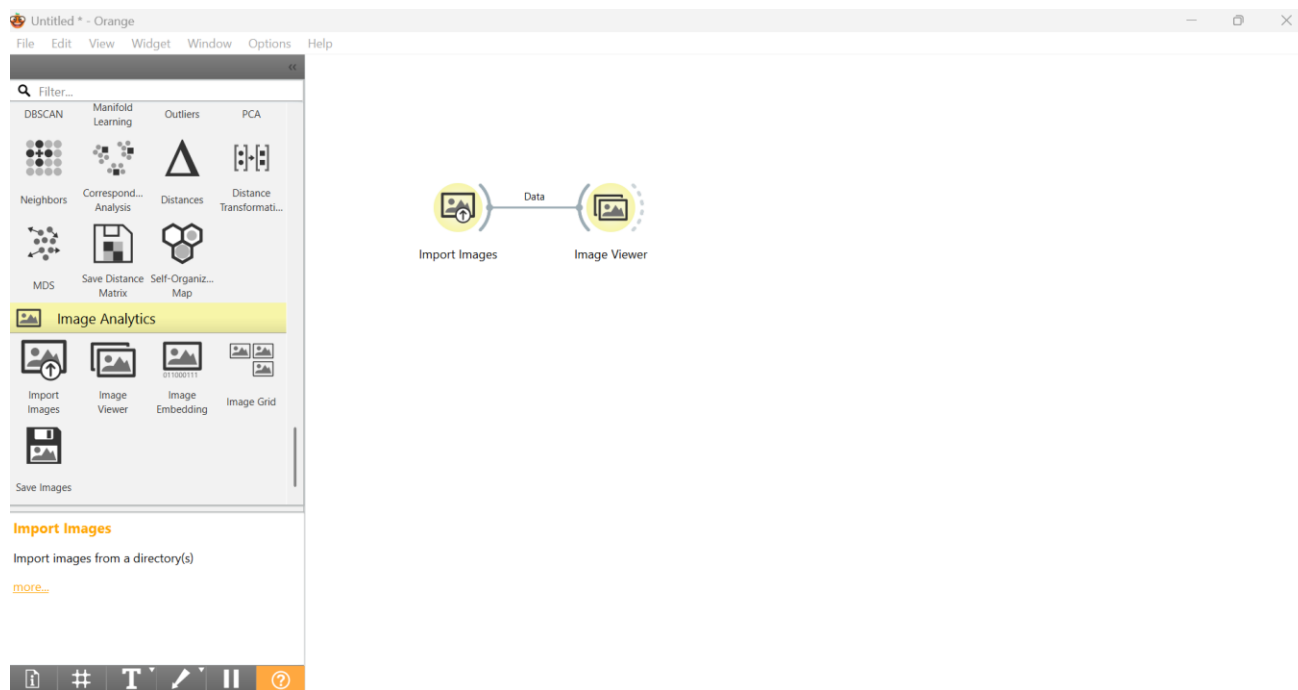
**Step 2: Click on Import Images widget under Image Analytics menu**



### Step 3: Click on Import images widget to upload Train dataset folder which has both bleached and unbleached images of coral.



### Step 4: Connect Image Viewer widget to the output of Import images.



## Step 5: Connect Image Embeddings widget to the output of Import Images widget to convert images in 0 and 1.

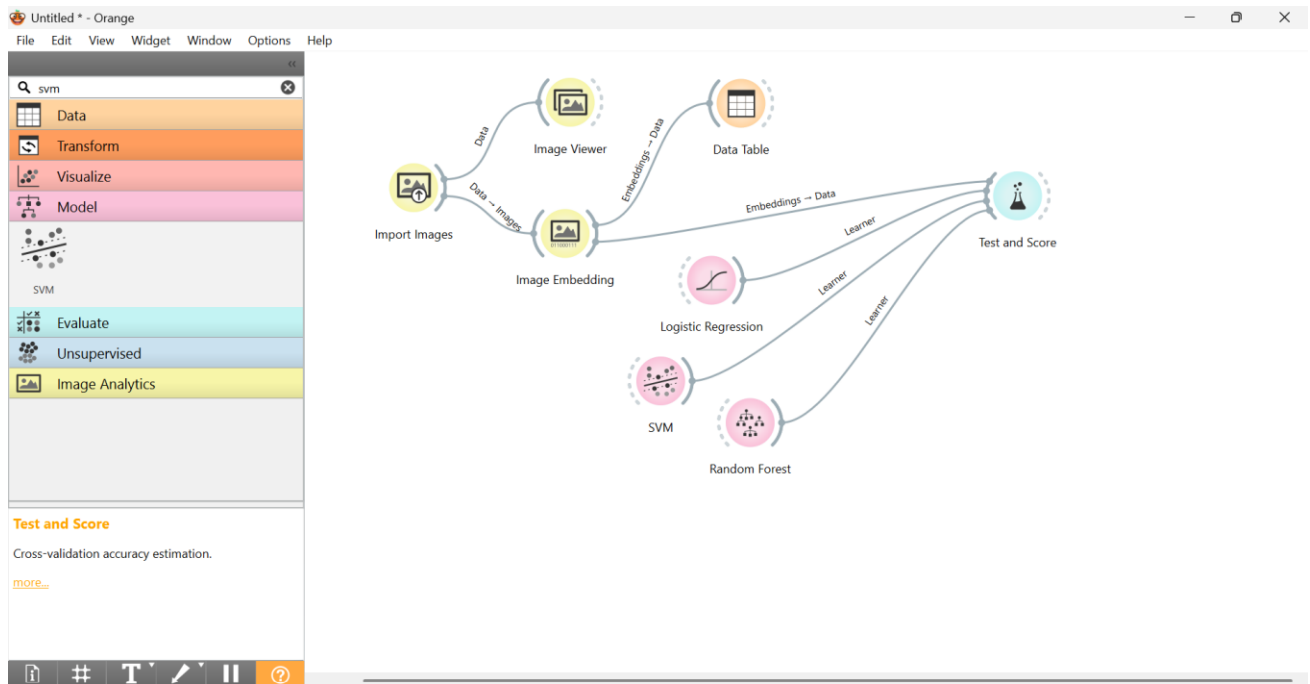
The screenshot shows the Orange3 software interface. On the left, the 'Image Analytics' widget palette is open, displaying various image processing widgets. The 'Image Embedding' widget is highlighted, and its description is shown below: 'Image Embedding through deep neural networks. [more...](#)'. In the main workspace, a workflow is set up: 'Import Images' is connected to 'Image Viewer' (labeled 'Data') and 'Image Embedding' (labeled 'Data - Images').

## Step 6: Connect Data Table widget to the output of Image Embedding and click to check the data instances

The screenshot shows the Orange3 software interface with the workflow from Step 5 updated. The 'Image Embedding' widget is now connected to a 'Data Table' widget (labeled 'Embeddings - Data'). The 'Data Table' widget's data preview is open, showing a table with the following data:

hidden origin type	category	image name	image	size	width	height	n0
	Bleached	1	rflow/Coral Bleac image	108214	227	227	0.167783
1	Bleached	100	Bleached/100.p...	90942	227	227	0.28305
2	Bleached	101	Bleached/101.p...	111510	227	227	0.123964
3	Bleached	102	Bleached/102.p...	92203	227	227	0.322635
4	Bleached	103	Bleached/103.p...	116068	227	227	0.194662
5	Bleached	104	Bleached/104.p...	105082	227	227	0.0605793
6	Bleached	105	Bleached/105.p...	111299	227	227	0.0138477
7	Bleached						

## Step 7: Add Test and Score widget to the output of Image Embedding and algorithms like Logistic Regression, SVM and random Forest.



## Step 8: Open Test and Score widget to compare the values of AUC, CA, F1 Score, Precision, Recall and MCC

The screenshot shows the configuration window for the **Test and Score** widget. The **Cross validation** section is set to **Number of folds: 5**, **Stratified**, **Repeat train/test: 10**, and **Training set size: 66 %**. The **Evaluation results for target** table is as follows:

Model	AUC	CA	F1	Prec	Recall	MCC
Logistic Regression	0.977	0.916	0.916	0.918	0.916	0.834
SVM	0.970	0.889	0.888	0.890	0.889	0.779
Random Forest	0.940	0.895	0.895	0.896	0.895	0.791

The **Compare models by** section is set to **Area under ROC curve**. The comparison table is as follows:

	Logistic Regression	SVM	Random Forest
Logistic Regression		0.863	0.922
SVM	0.137		0.925
Random Forest	0.078	0.075	

Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.

## Step 9: Add Confusion Matrix to the output of Test and Score and double click to view the Actual and Predicted values.

The screenshot shows the Orange 3.12.1 interface. The workflow consists of the following widgets: Import Images, Image Viewer, Image Embedding, Data Table, Logistic Regression, SVM, Random Forest, Test and Score, and Confusion Matrix. The 'Confusion Matrix - Orange' window is open, showing the following data:

		Predicted	
		Bleached	Unbleached
Actual	Bleached	129	16
	Unbleached	8	134
$\Sigma$		137	150

## Step 10: Again add Import Images widget and upload Test Dataset.

The screenshot shows the Orange 3.12.1 interface with an additional 'Import Images (1)' widget added to the workflow. The 'Import Images (1) - Orange' window is open, showing the following information:

- Folder: Test Dataset
- Info: 55 images

## Step 11: Add Image viewer to view the images of Test dataset.

The screenshot shows the Orange 3 software interface. On the left is a widget palette with various machine learning and data analysis tools. The main workspace contains a workflow diagram with the following components and connections:

- Import Images** widget connects to **Image Viewer** (Data - Images) and **Image Embedding** (Data - Images).
- Image Viewer** widget connects to **Data Table** (Embeddings - Data).
- Image Embedding** widget connects to **Test and Score** (Learner).
- Data Table** widget connects to **Test and Score** (Learner).
- Test and Score** widget connects to **Confusion Matrix** (Evaluation Results).

A pop-up window titled "Image Viewer (1) - Orange" is open, displaying a grid of 16 coral reef images. The window includes a "File" menu, a "Filter..." search bar, and several dropdown menus for "Image Filename Attribute" (set to "image"), "Title Attribute" (set to "image name"), and "Image Size". A "Send Automatically" checkbox is checked at the bottom.

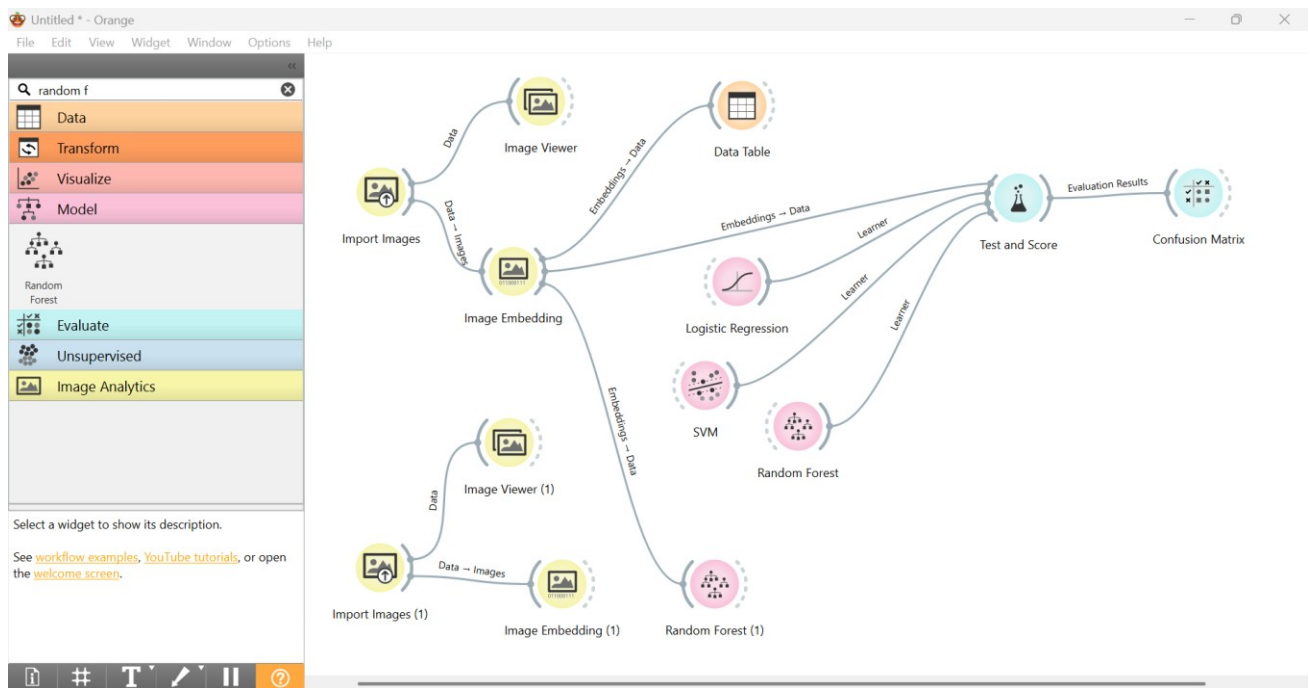
## Step 12: Add Image Embedding widget to convert images in 0 and 1

The screenshot shows the Orange 3 software interface with a different workflow diagram. The components and connections are as follows:

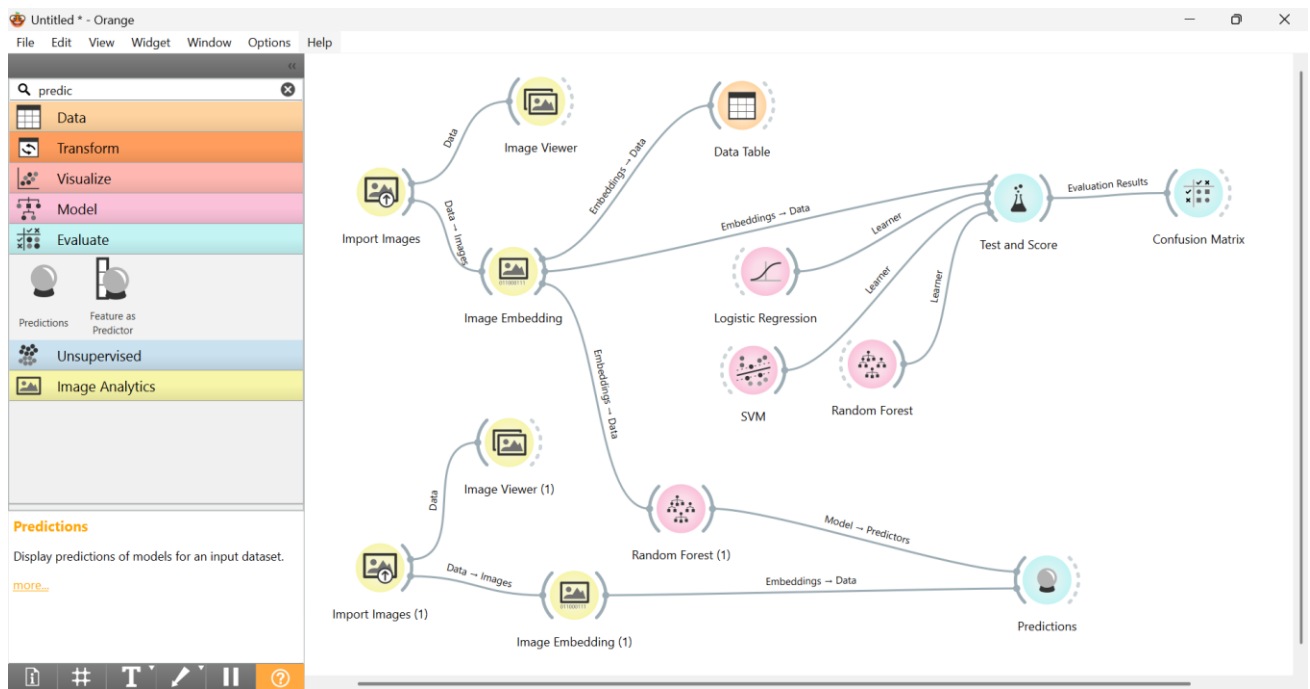
- Import Images** widget connects to **Image Viewer (1)** (Data) and **Image Embedding (1)** (Data - Images).
- Image Embedding (1)** widget connects to **Logistic Regression** (Learner), **SVM** (Learner), and **Random Forest** (Learner).
- Logistic Regression**, **SVM**, and **Random Forest** widgets all connect to **Test and Score** (Learner).
- Test and Score** widget connects to **Confusion Matrix** (Evaluation Results).

The widget palette on the left is the same as in Step 11, but the workflow diagram has been updated to include the new Image Embedding widget and three different classifier widgets.

## Step 13: Connect initial Image Embedding to the newly added Random Forest widget.



## Step 14: Add Predictions widget to the output of Image Embedding widget for Testing data.



## Step 15: Double click on predictions widget to check the Testing Data image is Bleached or Unbleached.

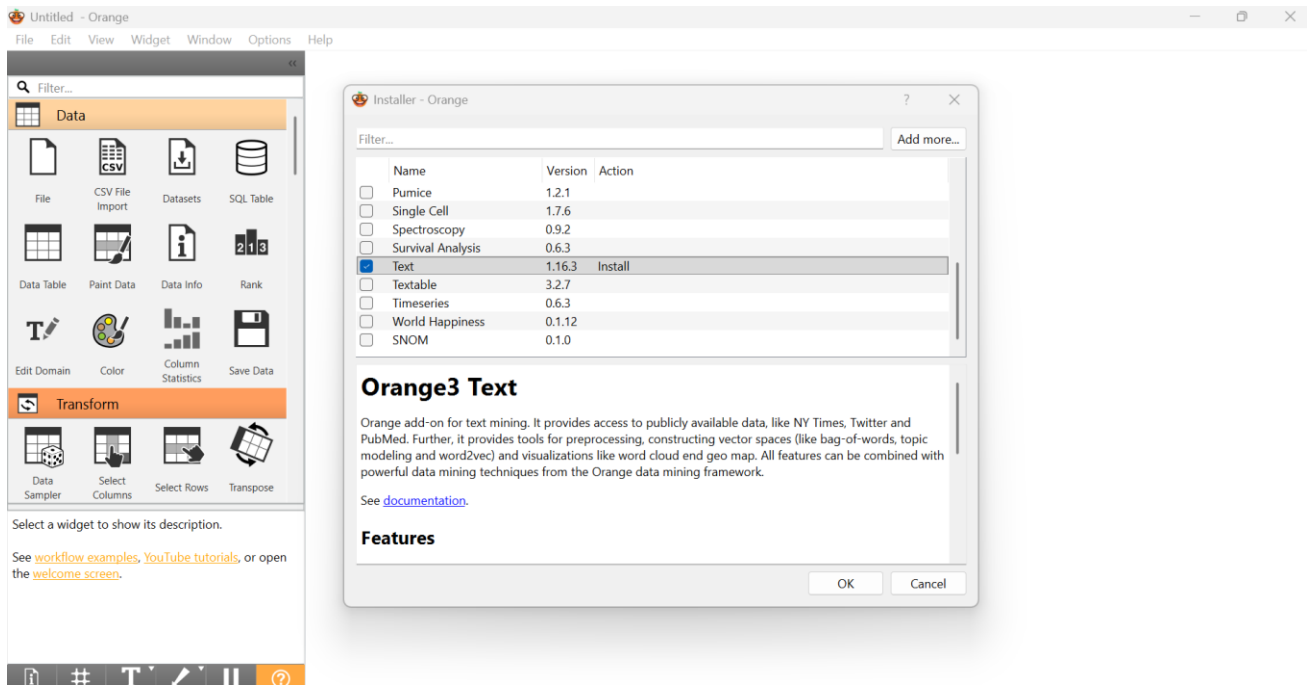
The screenshot shows the Orange data mining software interface. On the left is a widget palette with categories: Data, Transform, Visualize, Model, Evaluate, Predictions, Unsupervised, and Image Analytics. The main workspace contains a workflow with 'Data' and 'Image Viewer' widgets connected by a 'Data' link, and 'Image Viewer' and 'Data Table' widgets connected by an 'Image - Data' link. A 'Predictions' widget window is open in the foreground, displaying a table of results for a 'Random Forest (1)' model. The table has columns for 'image name', 'image', 'size', 'width', 'height', 'n0', and 'n1'. The 'image name' column lists image IDs (1-11) and their predicted status (Unbleached or Bleached). The 'n0' and 'n1' columns show the predicted probabilities for each class.

	Random Forest (1)	image name	image	size	width	height	n0	n1
1	Unbleached	1	1.png	122817	227	227	0.263119	0.0673202
2	Bleached	11	11.png	91892	227	227	0.305015	0.0039056
3	Unbleached	13	13.png	107984	227	227	0.227681	0.0649252
4	Bleached	15	15.png	101215	227	227	0.208008	0.0374342
5	Unbleached	17	17.png	103013	227	227	0.120755	0.0398997
6	Unbleached	19	19.png	111554	227	227	0.200335	0.0690718
7	Bleached	201	201.png	112648	227	227	0.156265	0.0229786
8	Bleached	202	202.png	101644	227	227	0.0940598	0.319403
9	Bleached	203	203.png	102433	227	227	0.0116396	0.166725
10	Bleached	204	204.png	110809	227	227	0.124021	0.280888
11	Bleached	205	205.png	108504	227	227	0.0834972	0.158679

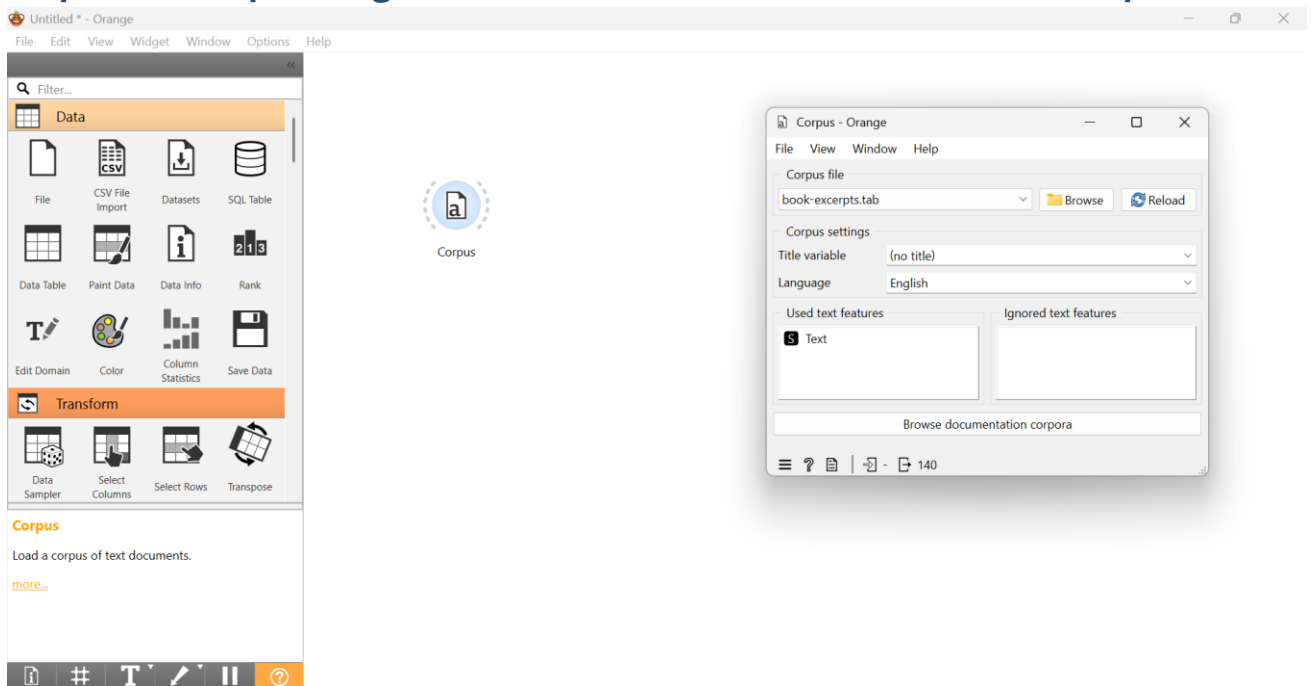
# PROJECT 3

Develop an AI-based Natural Language Processing (NLP) model using Orange Data Mining to classify book excerpts into Children's Literature or Adult Literature based on their textual content.

**Step 1:** Click on Options -> Add On -> Text . Restart Orange for the changes to take effect.



**Step 2:** Add corpus widget and from inbuilt datasets select book excerpts.



### Step 3: Connect Data table with the output of corpus widget to view the dataset.

The screenshot shows the Orange data mining interface. A workflow is visible with a 'Corpus' widget connected to a 'Data Table' widget. The 'Data Table' widget's options panel is open, showing settings for 140 instances with 2 features and 1 meta-attribute. The 'Selection' section is set to 'Select full rows'. A preview window displays the following data:

include	Category	Text
1	children	the house Jim says he rum ; and as he s...
2	children	has lived rough and I'll raise Cain Your ...
3	children	Now boy he said take me in to the capt...
4	children	thanks to you big hulking chicken-hear...
5	children	the empty chest; and the next we had ...
6	children	stood irresolute on the road You have u...
7	children	WE rode hard all the way till we drew u...
8	children	same as the tattoo mark Billy Bones his...
9	children	IT was longer than the squire imagined...
10	children	treasure Long John Silver unearthed a ...
11	children	We are so grateful to you for having kil...
12	children	I am told said the third man that it is th...
13	children	to find the one paved with yellow brick...
14	children	take away the pole I shall be greatly ob...
15	children	Won't you tell me a story while we are ...
16	children	5 The Rescue of the Tin Woodman Wh...
17	children	to a place where the trees and branche...
18	children	All this time Dorothy and her compani...
19	children	I am going to the Great Oz to ask him t...
20	children	measuring the distance carefully in his ...
21	children	and tossing her head I'm going to have...
22	children	up somethin' you eat me out o' house ...
23	children	one of the household creeping in and ...

### Step 4: Add Preprocess Text with the output of Corpus to clean and transform raw text into a structured format.

The screenshot shows the Orange data mining interface with a workflow. A 'Corpus' widget is connected to a 'Preprocess Text' widget, which is then connected to a 'Data Table' widget. The 'Preprocess Text' widget is highlighted in the left sidebar.

### Step 5: Add Bag of Words widget to convert text documents into numerical data.

The screenshot shows the Orange data mining interface with a workflow. A 'Corpus' widget is connected to a 'Preprocess Text' widget, which is then connected to a 'Bag of Words' widget. The 'Bag of Words' widget is highlighted in the left sidebar.

## Step 6: Connect Data Sampler widget to randomly divide the dataset into training and testing machine learning models.

The screenshot shows the Orange 3 software interface. On the left, a widget palette is visible with a search bar containing "data sampl". The "Data Sampler" widget is highlighted under the "Transform" category. Below the palette, a description for "Data Sampler" is shown: "Randomly draw a subset of data points from the input dataset." and a link "more\_".

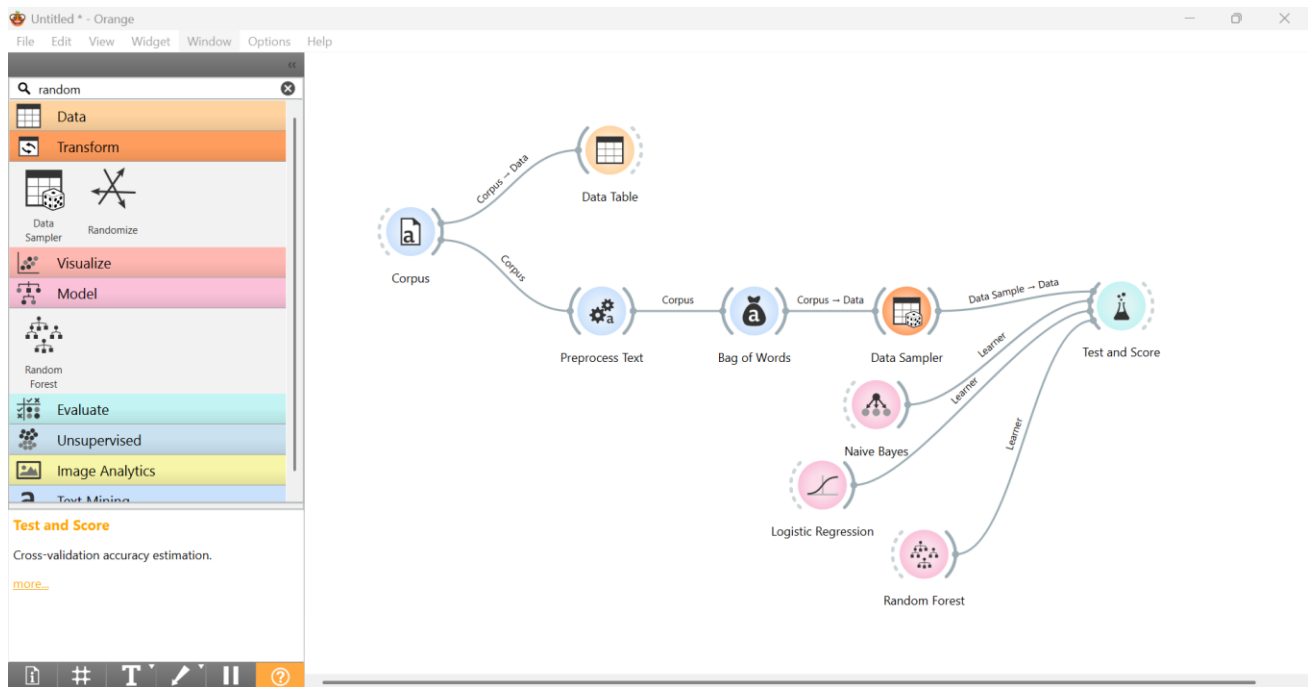
The main workspace displays a workflow diagram. It starts with a "Corpus" widget (represented by a document icon with 'a'). Two arrows labeled "Corpus -> Data" branch from the "Corpus" widget to a "Data Table" widget (represented by a grid icon) and a "Preprocess Text" widget (represented by a gear icon with 'a'). The "Preprocess Text" widget is connected to a "Bag of Words" widget (represented by a bag icon with 'a'), which is then connected to a "Data Sampler" widget (represented by a grid icon with a red circle). The "Data Sampler" widget has two outgoing arrows labeled "Corpus -> Data".

## Step 7: Connect Test and Score with the output of Data Sampler. Train it with Naïve Bayes algorithm.

The screenshot shows the Orange 3 software interface. On the left, a widget palette is visible with a search bar containing "naive". The "Naive Bayes" widget is highlighted under the "Model" category. Below the palette, a description for "Naive Bayes" is shown: "A fast and simple probabilistic classifier based on Bayes' theorem with the assumption of feature independence." and a link "more\_".

The main workspace displays a workflow diagram. It starts with a "Corpus" widget (represented by a document icon with 'a'). Two arrows labeled "Corpus -> Data" branch from the "Corpus" widget to a "Data Table" widget (represented by a grid icon) and a "Preprocess Text" widget (represented by a gear icon with 'a'). The "Preprocess Text" widget is connected to a "Bag of Words" widget (represented by a bag icon with 'a'), which is then connected to a "Data Sampler" widget (represented by a grid icon with a red circle). The "Data Sampler" widget has two outgoing arrows labeled "Corpus -> Data". One arrow goes to a "Naive Bayes" widget (represented by a tree icon) and the other goes to a "Test and Score" widget (represented by a test tube icon). The "Naive Bayes" widget is connected to the "Test and Score" widget via an arrow labeled "Learner".

## Step 8: Train your model with more algorithms like Logistic Regression , Random Forest



## Step 9: Compare the values of AUC, CA, F1, Precision, Recall and MCC.

