**NAME – KATKURI PAVANI**

**STUDENT ID – 700732649**

**MACHINE LEARNING ASSIGNMENT 1**

**Github link :** [**https://github.com/pavanir2530/ML-Assign-1-pk**](https://github.com/pavanir2530/ML-Assign-1-pk)

**Video link :** [**https://drive.google.com/file/d/1Q3lxykqyhr8BRzd7HlPwG11rd0lzpcdx/view?usp=sharing**](https://drive.google.com/file/d/1Q3lxykqyhr8BRzd7HlPwG11rd0lzpcdx/view?usp=sharing)

**QUESTION -1**

Graphical user interface, text

Description automatically generated

In the snippet of code above, we created an empty list called "ages" and then added some members to it. Then we used the built-in method sort(), which by default sorts list entries in ascending order. The other built-in functions min() and max() were also utilized to determine the minimum and maximum elements in the supplied list.

The minimum and maximum ages were added to the list using the append() method in the code sample, and when the new elements were added, the length of the new list was reported.

Graphical user interface, text, application, email

Description automatically generated

Here, we used the if-else conditional statements to find the list's median and displayed the median.

Graphical user interface, text, application, email

Description automatically generated

In this bit of code, the average was determined by using the sum() method to find the sum of the elements in the given list. We determined the list's range in the subsequent snippet.

**QUESTION – 2**

Graphical user interface, text, application

Description automatically generated

This code sample uses the dict() function to initialize a new dictionary before adding keys and values to it. Then we printed a second dictionary that contained student information as keys and values.

Text

Description automatically generated

In this code sample, the length of the dictionary was determined using the len() function, and the datatype of the skills key was determined using the type() function. The skills key was then updated by adding certain values to it. Dictionary values were then printed after the dictionary's keys. Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**QUESTION -3**

Graphical user interface, text, application

Description automatically generated

Here, we formed two tuples with this code and then added the values to the tuples. The other tuple was then given new elements, and it was given back to the original tuple. We added values to the tuples using the compound assignment operator since tuples are immutable.

Graphical user interface, text, application

Description automatically generated

In this code, two additional tuples were formed, added to the family member tuple, and then printed.

**QUESTION -4**

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

In this line of code, we determined the set's length, added a few extra elements, and printed the updated set. The set was then expanded by a few values at once, and it was then printed once more.

The resultant set was then printed after checking the delete () function from sets' built-in functions.

Next, we used the set operation known as union and printed the outcome.

Graphical user interface, text, application, email

Description automatically generated

The intersection, subset, disjoint, and set symmetric difference were used in this line of code, and the results were printed.Graphical user interface, text, application, email

Description automatically generated

Here, we demonstrated that sets will not permit duplicates by entirely deleting the sets, printing the empty sets, and then converting the list into sets using the set() function.

**QUESTION -5**

Graphical user interface, text, application, email

Description automatically generated

In this, By taking the radius as user input and using the input() function, we calculated the circle's area first, followed by its circumference.

**QUESTION -6**

Graphical user interface, text, application, email

Description automatically generated

In this bit of code, the given string was divided at each space character using the split() method.

Next, we compared each word to every other word in the string and recorded how many distinct words in the supplied string

**QUESTION -7**

Graphical user interface, text, application

Description automatically generated The escape character "t" was used in this bit of code to insert three spaces between the words.

**QUESTION -8**

Graphical user interface, text, application, email

Description automatically generated

In this , code we initializes the radius to 10 and finds the area of circle in meters square.

**QUESTION -9**

Graphical user interface, text, application, email

Description automatically generated

With the help of user input, we translated a list of values in pounds to kilograms in this snippet of code.

**Question -10**

Chart, box and whisker chart

Description automatically generated

**Solution:**

Given data elements are taken in the tabular form as below,

|  |  |
| --- | --- |
| **Feature** | **Label** |
| 1 | O |
| 2 | O |
| 3 | X |
| 6 | X |
| 6 | X |
| 7 | O |
| 10 | O |
| 11 | O |

Here, the first four rows of data are considered to be the Training dataset and the next four rows are selected as the Testing dataset.

Now, according to the KNN Classifier we shall now consider K=3 and then the distance between the testing and training data is demonstrated below.

In the below table the columns are the training dataset and rows are the testing dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1(O) | 2(O) | 3(X) | 6(X) |
| 6 | 5 | **4** | **3** | **0** |
| 7 | 6 | **5** | **4** | **1** |
| 10 | 9 | **8** | **7** | **4** |
| 11 | 10 | **9** | **8** | **5** |

The highlighted rows are the distance values.

Let us now assume ‘O’ as negative and ‘X’ as positive values. Now the prediction on testing data is as below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | True label | Predicted label | O/P |
| 6 | X | X | Tp |
| 7 | O | X | Fp |
| 10 | O | X | Fp |
| 11 | O | X | Fp |

Confusion matrix for the above prediction is:

|  |  |
| --- | --- |
| TN | FP |
| FN | TP |

The final confusion matrix is :

# 0 3

**0 1**

Accuracy of the classifier = (TP + TN) / (P + N) = 1 / 4 = 0.25 Sensitivity of the classifier = TP / P = 1/1 = 1

Specificity of the classifier = TN / N = 0/3 = 0