Weekly Assignment Report – 1

Summary of the report :

- Machine learning
- Machine learning steps
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Overfitting
- Vectors
- Magnitude of a vector
- Distance between vectors
- Matrix
- Arithmetic Operations on matrix
- Matrix manipulation
- Python Datatypes
- Iterations
- Functions
- NumPy Package

References for the report:

- https://www.geeksforgeeks.org/python-programming-language/
- https://www.w3schools.com/python/python_intro.asp
- https://www.kaggle.com/
- https://scholar.google.com

Algorithm:

Algorithms are a set of instructions for the computer.

Defining machine learning:

Machine learning (ML) is the study of approaches and algorithms for automating solutions to complex problems that are difficult to address using conventional programming techniques.

Why Machine learning?

A huge quantity of data has been produced since the invention of computers, and it seems to be expanding constantly. This data may be very helpful with efficient learning techniques. Finding new patterns and ideas while learning from the data has proven to be the researchers' most challenging assignment. Machine learning uncovers the data's hidden patterns, learns from them, and successfully predicts the unseen data.

Real world applications of Machine learning:

- Robotics
- Computer vision
- Voice recognition
- Facial recognition
- Business analytics

Machine learning steps:

- Data manipulation
    - Data acquisition
    - Data storage
    - Data cleaning
- Analytics
    - Finding relationships and correlations
    - Exploratory data analysis
    - Predictive machine learning
- Evaluation and visualization

Supervised learning:

Supervised learning is simply a formalization of the idea of learning from examples. In supervised learning, the learner (computer program) is provided with two sets of data, a training set and a test set. The idea is for the learner to "learn" from a set of labeled examples in the training set so that it can identify unlabeled examples in the test set with the highest possible accuracy.

Unsupervised learning:

Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of the input patterns.

- **Clustering:**
  Grouping similar examples to understand a subject (dataset) in a machine learning system is called clustering.

Reinforcement learning:

A computer learns how to perform actions based on learning from interactions. The computer receives some evaluation(reward) based on the performed action.

Model Evaluation:

For a ML problem, we must evaluate whether the model can predict the label for new data and future data. To perform the evaluation, we must:

- Randomly split the data into training and testing data.
- Using the training data to train the model.
- Evaluate the model using the test data.
- Check the result and try again.

Model Selection:

Model selection is one of the challenging tasks because there is not easy way to know if a certain model offers the best result. Hence, we need to try out different model.

Overfitting:

When a ML model learns the details and noise of a training dataset, the gained knowledge might negatively affect the performance of the new dataset, this is known as overfitting.

Vector:

A vector is a data structure with two components: magnitude and direction.

Vector Operations:

The three main operations on vectors are:

- Transpose
- Add
- Inner product

Transpose

$$\mathbf{x}^T = [x_1, x_2, \ldots, x_n]$$

Add

$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

Inner product

$$\mathbf{x}^T\mathbf{y} = x_1y_1 + x_2y_2 + \ldots + x_ny_n$$

Magnitude of the vector:

To find the magnitude of the vector or the length of the vector:

$$\text{length}((x)) = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

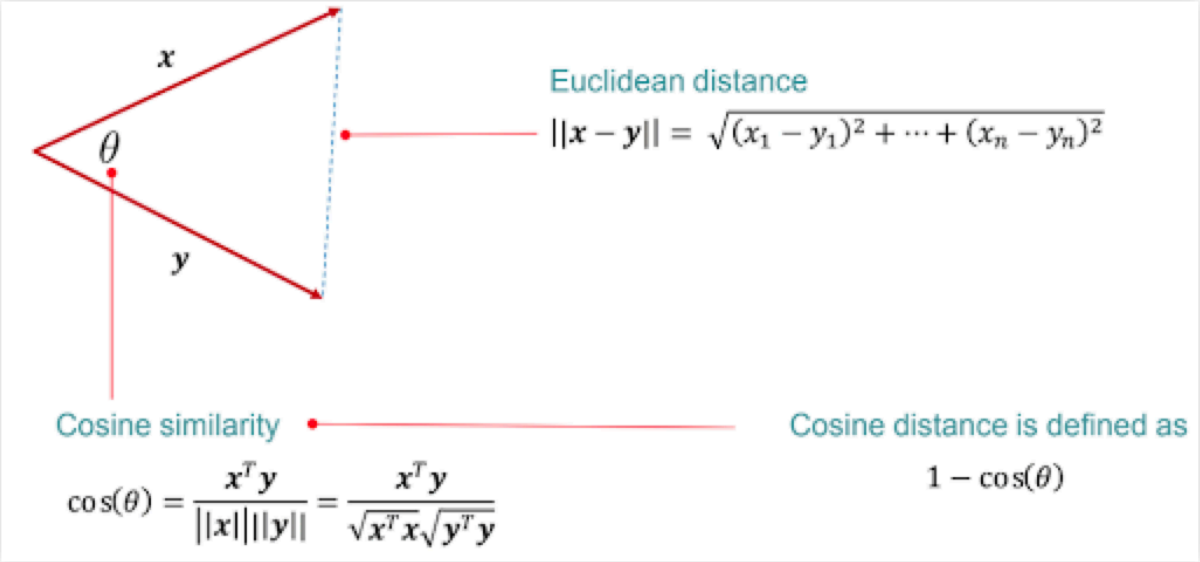
The above length is also called **2-norm** of a vector.

$$||\mathbf{x}||_2 = \sqrt{x_1^2 + \ldots + x_n^2}$$

Distance between vectors:

Cosine similarity measure the cosine angle between two vectors.

Cosine distance measures angular difference between two vectors.



Matrix:

Matrix is a 2-dimensional array consisting of numbers.

Basic operations on matrix:

- Addition/Subtraction: Addition and subtraction can be performed only if both matrices are of same size.
- Scalar Multiplication/Division
- Elementwise matrix multiplication
- Matrix to matrix multiplication
- Transpose of a matrix
- Symmetric matrix
- Diagonal matrix
- Identity matrix
- Inverse matrix
- Orthogonal matrix

Setting up Python:

- Installing Anaconda, Jupyter Notebook.
- Open Anaconda navigator and then open jupyter Notebook.

Printing Hello, World!

```
print("Hello, World!")
print('Hello, World!')
```

Upon Execution, below is the result obtained:

```
Hello, World!
Hello, World!
```

Inserting Comments:

Comments are used to understand what each line of code does.

```
#One Line Comment

'''
Multi
Line
Comment
'''

"""
Another
Multi
Line
Comment
"""
```

Variables:

Variables are containers used to store values.

```
  x = 10
  type(x)
```

List:

List is a common datatype in python and can be used to store different type of values. The syntax is list_name = [item_0,item_1,…,item_n]

```
list1 = [1,2,3,'XYZ',4.92]
list1
```

[1, 2, 3, 'XYZ', 4.92]

Tuple:

Tuple is like list, but the items cannot be changed on added. The syntax is:

tuple_name = (item_0, item_1,…, item_n)

```
tup1 = (1,2,3,'XYZ',4.92)
tup1
```

(1, 2, 3, 'XYZ', 4.92)

Dictionaries:

Dictionaries are also like lists, but the elements will be in a key-value pair. The syntax is:

Dictionary_name = {key_1 : value_1,…, key_n : value_n}

```
dict1 = {'one' : 1, 2 : 'Two'}
dict1
```

{'one': 1, 2: 'Two'}

Strings:

Strings are used to store text in python.

```
str1 = "This is Sparta"
str1
```

Conditions and Decision Trees:

Conditions or Decisions can be made by checking the conditions of the variable. For example, is the cup empty? The result can only be Yes or No, True or False

```
isCupEmpty = True

if isCupEmpty == True:
  print("True")
else:
  print("False")
```

Iterations:

Loops are Iterations can be defined using the for or while loop.

- For loops:

  The for loop runs for a fixed number of iterations.

  ```
  list2 = [1,2,3,4,5]
  for items in list2:
    print(items)
  ```

- While loops:

  While loop is like for loops, but the number of iterations is unknown.

  ```
  i = 0
  while(i < 5):
    print(i, end = " ")
    i += 1
  ```

Functions:

Functions are used to store a block of code and user can call them whenever it is required.

```
def add(a,b):
  return a + b

add(10,20)
```

Anonymous Function:

Anonymous functions are functions which are not stored in a program file, but is associated with a variable.

```
sub = lambda x,y: x - y
print(sub(20,10))
```

NumPy Package:

NumPy package is used to perform various mathematical operations over vectors, matrices, random numbers.

```
import numpy as np

vec1 = np.array([1,2,3])
print(vec1)
mat1 = np.array([(1,2,3), (4,5,6)])
print(mat1)
```

Matrix Addition and Subtraction:

- Matrix Addition:

```
import numpy as np
mat1 = np.array([(1,2,3), (4,5,6)])
print("The values of Matrix 1 before Addition is \n", mat1)
print("The values of Matrix 2 after Addition is \n", mat1 + 3)
mat2 = np.array([(1,3,5), (2,4,6)])
print("The sum of Matrix 1 and Matrix 2 is \n", mat1 + mat2)
```

```
The values of Matrix 1 before Addition is
 [[1 2 3]
 [4 5 6]]
The values of Matrix 2 after Addition is
 [[4 5 6]
 [7 8 9]]
The sum of Matrix 1 and Matrix 2 is
 [[ 2  5  8]
 [ 6  9 12]]
```

- Matrix Subtraction:

```
import numpy as np
mat1 = np.array([(1,2,3), (4,5,6)])
print("The values of Matrix 1 before Subtraction is \n", mat1)
print("The values of Matrix 2 after Subtraction is \n", mat1 - 3)
mat2 = np.array([(1,3,5), (2,4,6)])
print("The difference of Matrix 1 and Matrix 2 is \n", mat1 - mat2)
```

The values of Matrix 1 before Subtraction is
[[1 2 3]
[4 5 6]]
The values of Matrix 2 after Subtraction is
[[-2 -1 0]
[1 2 3]]
The difference of Matrix 1 and Matrix 2 is
[[0 -1 -2]
[2 1 0]]

Matrix multiplication:

```
import numpy as np
mat1 = np.array([(1,2,3), (4,5,6)])
print("The values of Matrix 1 before Multiplication is \n", mat1)
print("The values of Matrix 1 after Multiplication is \n", mat1 * 3)
mat2 = np.array([(1,3,5), (2,4,6)])
print("The product of Matrix 1 and Matrix 2 is \n", mat1 * mat2)
```

The values of Matrix 1 before Multiplication is
[[1 2 3]
[4 5 6]]
The values of Matrix 1 after Multiplication is
[[3 6 9]
[12 15 18]]
The product of Matrix 1 and Matrix 2 is
[[1 6 15]
[8 20 36]]

Matrix Division:

Dividing a matrix is equal to inverse of a matrix.

```
import numpy as np
mat1 = np.array([(1,2), (4,5)])
print("The values of Matrix 1 before Inverse is \n", mat1)
print("The values of Matrix 1 after Inverse is \n", mat1_inv)
mat1_inv = np.linalg.inv(mat1)
```

```
The values of Matrix 1 before Inverse is
 [[1 2]
 [4 5]]
The values of Matrix 1 after Inverse is
 [[-1.66666667  0.66666667]
 [ 1.33333333 -0.33333333]]
```

Useful matrix manipulation:

- **Slicing:**

  Slicing refers accessing matrix or vector elements.

  ```
  import numpy as np
  mat1 = np.array([(1,2), (4,5)])
  mat1[:,1]
  ```

  ```
  array([2, 5])
  ```

- **Logical checks:**

  We can verify the elements of a matrix or vector against any conditions.

  ```
  import numpy as np
  mat1 = np.array([(1,2), (4,5)])
  print("Boolean-->",mat1[:,1]>4)
  ```

  ```
  Boolean--> [False  True]
  ```

Week 1 Quiz Screenshot: