

University of Sunderland
ISMT College
Kathmandu, Nepal
E-Portfolio Documentation

Student name: Pradeep Khadka

Module Code & Name: CET313 Artificial Intelligence

Student Id: 219311481

E-Portfolio Canvas Link:

<https://canvas.sunderland.ac.uk/eportfolios/7244?verifier=hjT0GavB0nbyb18FkhhbUUM7MjULrMArjUIOkmbXx>

WEEK 1:

In first week of AI & Machine learning module, I was introduced about the most popular programming language python, where actually used it? How actually used it? & Why actually used it? Not only this I was introduced about the overall all basic concepts of python including data structure, list, dictionary, tuples, files, OOP, etc. including some famous python library like numpy and pandas.

Similarly, I was introduced about the AI and machine learning. Before jumping into AI and Machine learning, we require some foundation of some programming language. So, at first Python was introduce.

I also introduced about the where actually AI and machine learning used like weather forecasting, datamining, predicting price and stock market, etc. Also I was introduce about the awesome platform or IDE to Build project and write code like Google Colab Rearch and Jupiter, etc. where we can learn online also. In first week I have did more number of lab exercise & task in order to be familiar with python program.

Furthermore, I have provided about different reference websites to learn and research about python and its library. Also different book to learn about the AI and Machine learning.

Lab exercise what I did in week 1, I have pasted in this canvas e-portfolio, here for some evidence I have paste screenshot what I did in my lab and week 1 self-research period.

Lab Screen shot:

1. Variable, datatypes, conditions & Operators

```
counter = 100 # An integer assignment
miles = 1000.0 # A floating point assignment
count = counter * 2
print(count)
travel = miles / 2.5
print(travel)
print(name)
200
40.0
Kate
```

```
In [4]: # variables
a = 2 # a is here variable which store value 2
b = a * a
print(b)
name = "jo" #single or double quotes allowed
print(name) #print statement

4
jo
```

```
In [6]: x = int(input("Enter number1: ")) # variable store int type data
#by prompting user
y = int(input("Enter number2: ")) #same
if (x>y): #conditions implementation
    print(x, " is greater")
else:
    print(y, " is geraten")

Enter number1: 45
Enter number2: 20
45 is greater
```

1. Python Array & data Structure

```
In [32]: #Declaring Array and Data structure
z = [1,2,3]
print(z)
z.extend([4, 5, 6])
print(z)
z.append(0)
print(z)
z.pop(2) #poiping item 2 from array list
print(z)

[1, 2, 3]
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6, 0]
[1, 2, 4, 5, 6, 0]
```

```
In [9]: # Lists can be modified
integer_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]
print(integer_list[1]) # indices of a list indicated by []
3 in integer_list #checks if 23 is in the integer_list

2
Out[9]: True
```

2. List

```
In [11]: # all lists have a sort method
x = [4, 2, 3, 1, 5]
print(sorted(x)) # [1, 2, 3, 4, 5]
print(x) #[4, 2, 3, 1, 5]
y = sorted(x)
print(y) # [1, 2, 3, 4, 5]
print(x) # [4, 2, 3, 1, 5]
x.sort() # sort() is the method print(x) # [1, 2, 3, 4, 5]

[1, 2, 3, 4, 5]
[4, 2, 3, 1, 5]
[1, 2, 3, 4, 5]
[4, 2, 3, 1, 5]
```

```
In [12]: # transforming lists
x= [1, 2, 3, 4, 5]
evens = [x for x in range(5) if x % 2 == 0]
odds = [x for x in range (5) if x % 2 > 0]
print(odds) # [0, 2, 4]

[1, 3]
```

3. Tuple

```
In [35]: #tuples - like lists but cant be changed. use () or no braces
mytuple1 = (1,2)
mytuple2 = 3, 4

tuple = mytuple1 + mytuple2 # adding two tuple

print(mytuple1)
print (mytuple2)

print(tuple)

(1, 2)
(3, 4)
(1, 2, 3, 4)
```

4. Dictionaries

```
In [14]: #dictionaries - associate things like a DB
emptydict = {}
emptydict2 = {}
grades = { "jo" : 80, "tim" : 90 }
print(grades["jo"])
s={}
s[1] = {'date':1/1/16, 'symbol':'fox','price':4.44}
s[2] = {'date':2/1/16, 'symbol':'2ppl','price':24.44}
```

5. Files handling & exceptions handling

```
line_list= []      #list
email_dict = {}    #dictionaries
f_name = input("Enter a file name: ") #prompting user for file
try:               #exception handling to make the program simple and user friendly
    f_handle = open(f_name) #file will open and store it in f_handle variable
except:
    print(f"Sorry I couldn't find the file {f_name}")
    exit()         # if input file is wrong message will display and program will terminate
count = 0
for line in f_handle: #implementation of for loops (at here while line of file can be handle by line)
    count += 1
    line_list = line.split() #Split() built in method will allows to make or convert string of the file in list form
    if line.startswith('From'): #in this if condition days of the list should be less than 3 i.e tuesday
        #& Line.StartsWith() bulit in method checks whether a sting start with str
        email = line_list[1] #FIRST LINE EMAIL WILL CHECKS IN LIST
        if email not in email_dict: #if email not in email_dict then else will statement will checks
            email_dict[email] = 1
        else:
            email_dict[email] += 1
print(f"The number of line in {f_name} is {count}\n")
print(email_dict) #email_dict will display
# /content/drive/MyDrive/mbox-short.txt Note enter thiis to gain output
--INSERT--
Enter a file name: /content/drive/MyDrive/mbox-short.txt
The number of line in /content/drive/MyDrive/mbox-short.txt is 1911

{'stephen.marquard@uct.ac.za': 4, 'louis@media.berkeley.edu': 6, 'zqian@umich.edu': 8, 'rjlowe@iupui.edu': 4, 'cwen@iupui.edu': 10, 'gsilver@umich.edu': 6,
```

6. Functions

```
In [37]: # functions
def mysquare(x):
    return(x*x)
mn =3
print(mysquare(mn))

#one-off 'Lamda' functions
add_five = lambda number: number + 5
print(add_five(number =4))

9
9
```

```
[ ] hours = int(input("Enter some hours: "))
rate = int(input("Enter some rate of works: "))

def computedPay(hours, rate):
    if hours <=40: #when hours is less than or equal to 40 this statement will perform
        pay= hours * rate
        return pay

    else: #When a hours is greater than 40 this stament will perform
        overTimes = hours -40 #function computedPay hanving parameter overTimes and rate
        pay =((overTimes*1.5*rate)) +(40 *rate) #operation for returing aspected output
        return pay

x = computedPay(hours, rate) #calling function (computedPay)
print ("Pay : ", x)

Enter some hours: 50
Enter some rate of works: 10
Pay : 550.0
```

7. OOPS

```
In [40]: #Classes

class Employee: #name of class is Employee
    'Common base class for all employees'
    empCount = 0
    def __init__(self, name, salary): # constructor class
        self.name = name #instance of classs
        self.salary = salary #same
        Employee.empCount += 1

    def displayCount(self):
        print("The employee %d" % Employee.empCount)
    def displayEmployee(self):
        print("Name: ", self.name, ", Salary: ", self.salary)

"This would create first object of employee class"
emp1 = Employee("Zarra", 2000)
"This would create second object of employee class"
emp2 = Employee("Manni", 5000)

emp1.displayEmployee()
emp2.displayEmployee()

print("The Employee %d " % Employee.empCount)

Name: Zarra , Salary: 2000
Name: Manni , Salary: 5000
The Employee 2
```

8. Random Regular Expression

```
In [19]: # 1 Random
import random
random.seed(10)
print(random.random())

0.5714025946899135
```

```
In [20]: # 2 Regular expressions
import re
str = 'an example word:cat !!'
match = re.search(r'word:\w\w\w', str)
# If-statement after search() tests if it succeeded
if match:
    print ('found'), match.group() ## 'found word:cat'
else:
    print ('did not find')

found
```

```
In [21]: ## Search for pattern 'iii' in string 'piiig'.
## All of the pattern must match, but it may appear anywhere.
## On success, match.group() is matched text.
match = re.search(r'iii', 'piiig') #=> found, match.group() == "iii"
print(match)
match = re.search(r'igs', 'piiig') #=> not found, match == None
print(match)

<re.Match object; span=(1, 4), match='iii'>
None
```

9. Python Numpy Libraries


```
import numpy as np
```

```
arr = np.array([1,2,3,4,5,6], ndmin=5)
```

```
print(arr)
```

```
print('numbers of dimensions :', arr.ndim)
```

```
[[[[[1 2 3 4 5 6]]]]]
```

```
numbers of dimensions : 5
```

```
import numpy as np
```

```
arr = np.array([[7,8,9,6], [1,2,3,4]], ndmin=20)
```

```
print(arrr)
```

```
print('numbers of dimensions :', arrr.ndim)
```

```
[[[[[[[[[[[[[[[[[[[[[7 8 9 6]
```

```
[1 2 3 4]]]]]]]]]]]]]]]]]]]]]]]]]
```

```
numbers of dimensions : 20
```

```
[ ] import numpy as nd
```

```
arr = np.array([1,2,3,4,5,6,7])
```

```
print(arr[1:5])
```

```
print(arr[:5])
```

```
print(arr[3:])
```

```
#negative slicing
```

```
print(arr[-1])
```

```
print(arr[-3:-1])
```

```
print(arr[1:5:2])
```

```
print(arr[::-2])
```

[2 3 4 5]

[1 2 3 4 5]

[4 5 6 7]

7

[5 6]

[2 4]

[1 3 5 7]

```

#generate a 3-d array with 5 rows 4 column
from numpy import random

x = random.randint(500, size = (3,5,4))

y = random.rand(5,3)
print(x)
print('\n')
# print(y)

--NORMAL--
[[[348 325 251 274]
  [136 141 373 70]
  [393 205 260 442]
  [402 4 59 475]
  [130 290 385 163]]

  [[ 15 412 243 178]
  [276 298 131 227]
  [453 116 332 49]
  [ 80 7 297 7]
  [323 490 496 28]]

  [[438 350 125 239]
  [ 9 40 270 26]
  [417 331 60 204]
  [230 451 462 128]
  [ 97 226 4 20]]]

```

10. Python Pandas library

```

import pandas as pd

mydataset = {
    'vehicles': ["Car", "Jeep", "van", "Truck", "Bus"],
    'Quantities': [5,7,8,10,9]
}

myvar = pd.DataFrame(mydataset)

print(myvar)

--VISUAL--

```

	vehicles	Quantities
0	Car	5
1	Jeep	7
2	van	8
3	Truck	10
4	Bus	9

```

import pandas as pd

data = {
    'employees' : [200, 150, 300, 500],
    'works' : [10, 15, 25, 30]
}

mydata = pd.DataFrame(data, index = ["Day1", "Day2", "Day3", "Day4"])

print(" The wage table of employee and weeks: ")
print(mydata)
print("\n")

print(mydata.loc["Day2"])

```

The wage table of employee and weeks:		
	employees	works
Day1	200	10
Day2	150	15
Day3	300	25
Day4	500	30

```

employees    150
works        15
Name: Day2, dtype: int64

```

```
[ ] import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/data.csv')

print(df.head(10))

print(df.tail(1))

print(df.info())
```

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0
5	60	102	127	300.0
6	60	110	136	374.0
7	45	104	134	253.3
8	30	109	133	195.1
9	60	98	124	269.0
168	75	125	150	330.4

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Duration    169 non-null    int64
1   Pulse       169 non-null    int64
2   Maxpulse    169 non-null    int64
3   Calories    164 non-null    float64
dtypes: float64(1), int64(3)
```

Also I have given exercise to create temperature conversion program and simple Average calculation program. Respective exercise I was completed in week 1. I have uploaded my lab work as well as exercise file in e-portfolio canvas. As a proof I have pasted here screen shot code of respective exercise code.

Temperature conversions,

```
In [44]: #temperation Conversion program between Fahrenheit & Centigrade

Centigrade = float(input("Enter tenperature in Centigrade: "))

#calculate degree-celcius to fahrenheit
fahrenheit = (Centigrade *1.8) + 32

print('%0.1f Centigrade is equal to %0.1f degree Fahrenheit'%(Centigrade, fahrenheit))

fahrenheit = float(input("Enter tenperature in Fahrenheit: "))

#calculate fahrenheit to degree-Celcius
centigrade = (fahrenheit -32) /1.8
print('%0.1f Fahrenheit is equal to %0.1f degree Celcius'%(fahrenheit, centigrade))
```

```
Enter temperature in Centigrade: 64
64.0 Centigrade is equal to 147.2 degree Fahrenheit
Enter temperature in Fahrenheit: 310
310.0 Fahrenheit is equal to 154.4 degree Celcius
```

Average calculator,

```
In [50]: #program to calculate average of any number

number = int(input("How many numbers: "))
sum_total = 0
for n in range(number):
    numbers = float(input('Enter number : '))
    sum_total += numbers

average = sum_total/number

print('Average of ', number, 'numbers is: ', average)
```

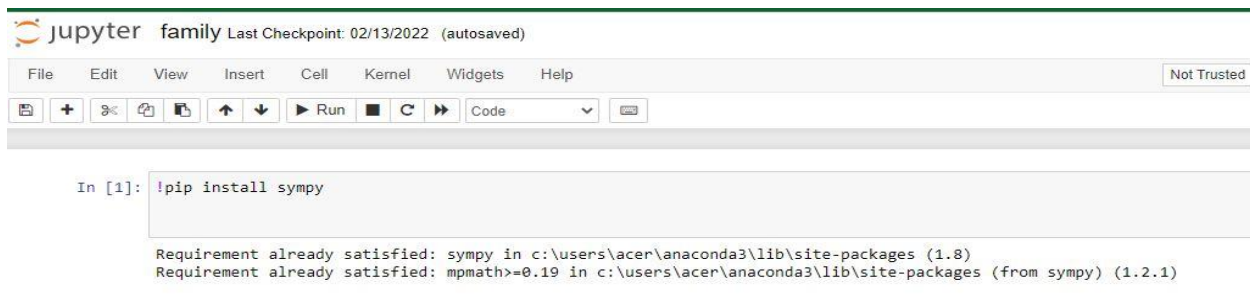
```
How many numbers: 3
Enter number : 20
Enter number : 10
Enter number : 30
Average of 3 numbers is: 20.0
```


All these are the work what we did in week 1, furthermore about the week one task I have uploaded it in canvas e-portfolio. On task, we have given to discuss and research over possible technique are being used to applicable in my chosen application area including impacts on the society and any associated ethical implications.

WEEK 2:

In week 2 of machine learning module I was introduced about the artificial intelligence. And similarly, we have learn different AI logic, semantic and relationships. In week we have did lab exercise and as well as task what was given. It was a fun moment for me. In this lab I was able install logpy.py and sympy.py. While installing smypy and logic I wasn't face problem but in while installing I have face problem error was exit. The lab exercise what I did I have already uploaded in canvas e-portfolio.

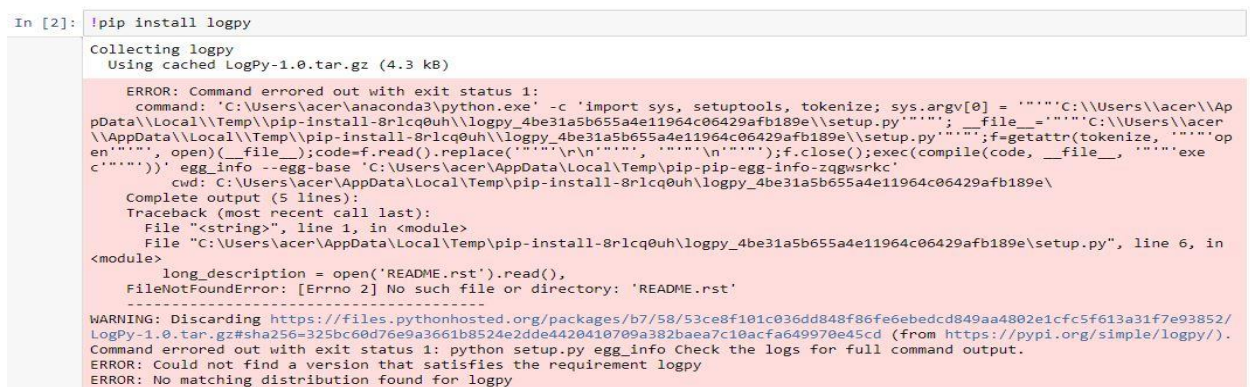
Sympy installing in Jupiter Notebook:



```
jupyter family Last Checkpoint: 02/13/2022 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
In [1]: !pip install sympy

Requirement already satisfied: sympy in c:\users\acer\anaconda3\lib\site-packages (1.8)
Requirement already satisfied: mpmath>=0.19 in c:\users\acer\anaconda3\lib\site-packages (from sympy) (1.2.1)
```

Error what I face during installing logpy but it function well,



```
In [2]: !pip install logpy

Collecting logpy
  Using cached LogPy-1.0.tar.gz (4.3 kB)

ERROR: Command errored out with exit status 1:
 command: 'C:\Users\acer\anaconda3\python.exe' -c 'import sys, setuptools, tokenize; sys.argv[0] = '"'"'C:\Users\acer\AppData\Local\Temp\pip-install-8rlcq0uh\logpy_4be31a5b655a4e11964c06429afb189e\setup.py'"'"'; __file__='"'"'C:\Users\acer\AppData\Local\Temp\pip-install-8rlcq0uh\logpy_4be31a5b655a4e11964c06429afb189e\setup.py'"'"';f=getattr(tokenize, '"'"'open'"'"', open)(__file__);code=f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' egg_info --egg-base 'C:\Users\acer\AppData\Local\Temp\pip-pip-egg-info-zqgwsrkc'
 cwd: C:\Users\acer\AppData\Local\Temp\pip-install-8rlcq0uh\logpy_4be31a5b655a4e11964c06429afb189e\
Complete output (5 lines):
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "C:\Users\acer\AppData\Local\Temp\pip-install-8rlcq0uh\logpy_4be31a5b655a4e11964c06429afb189e\setup.py", line 6, in <module>
    long_description = open('README.rst').read(),
  FileNotFound error: [Errno 2] No such file or directory: 'README.rst'

WARNING: Discarding https://files.pythonhosted.org/packages/b7/58/53ce8f101c036dd848f86fe6ebcd849aa4802e1cfc5f613a31f7e93852/LogPy-1.0.tar.gz#sha256=325bc60d76e9a3661b8524e2dde4420410709a382baae7c10acfa649970e45cd (from https://pypi.org/simple/logpy/).
Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
ERROR: Could not find a version that satisfies the requirement logpy
ERROR: No matching distribution found for logpy
```

I have install & import here logic.

```
In [4]: !pip install logic
Requirement already satisfied: logic in c:\users\acer\anaconda3\lib\site-packages (0.2.3)
Requirement already satisfied: multipledispatch in c:\users\acer\anaconda3\lib\site-packages (from logic) (0.6.0)
Requirement already satisfied: toolz in c:\users\acer\anaconda3\lib\site-packages (from logic) (0.11.1)
Requirement already satisfied: unification in c:\users\acer\anaconda3\lib\site-packages (from logic) (0.2.2)
Requirement already satisfied: six in c:\users\acer\anaconda3\lib\site-packages (from multipledispatch->logic) (1.15.0)

In [5]: import json
from logpy import Relation, facts, run, conde, var, eq
```

I was able to define here some rule,

For family relationships:

```
In [37]: #check if x is the parent of y
def parent(x, y):
    return conde([father(x,y)], [mother(x,y)])

#check if x is the grandparent of y
def grandparent(x,y):
    temp = var()
    return conde((parent(x, temp), parent(temp, y)))

#check for sibling relationships between 'a' and 'b'
def sibling(x,y):
    temp = var()
    return conde((parent(temp, x), parent(temp,y)))

#check if x is y's uncle
def uncle(x, y):
    temp = var()
    return conde((father(temp,x), grandparent(temp, y)))
```

I have implemented here some OOP logic to access the parental relationships,

```
#main method
if __name__ == '__main__':
    father = Relation()
    mother = Relation()

    with open('relationships.json') as f:
        d = json.loads(f.read())

    for item in d['father']:
        facts(father, (list(item.keys())[0], list(item.values())[0]))

    for item in d['mother']:
        facts(mother, (list(item.keys())[0], list(item.values())[0]))

    x = var()
```

At here we have check here relationships of the family.

```
# Jhons children
name = 'Jhon'
output = run(0, x, father(name, x))
print("\n List of " + name + "'s children: " )
for item in output:
    print(item)

# William's Mother
name = 'William'
output = run(0, x, mother(x, name))[0]
print("\n " + name + "'s mother:\n " + output)

#Adam's parents
name = 'Adam'
output = run(0, x, parent(name, x))
print("\n List of " + name + "'s parents: ")
for item in output:
    print(item)

#Wayne's grandparents
name = 'Wayne'
output = run(0, x, grandparent(x, name))
print("\nList of " + name + "'s grandparents: ")
for item in output:
    print(item)
```

```
#Megan's grandchildren
name = 'Megan'
output = run(0, x, grandparent(name,x))
print("\nList of " + name + "'s grandchildren:")
for item in output:
    print(item)
```

List of Jhon's children:

William's mother:
Megan

List of Adam's parents:
Sophia

List of Wayne's grandparents:
John
Megan

List of Megan's grandchildren:
Julie
Stephanie
Sophia
Tiffany
Chris
Neil
Wayne
Peter

```
In [34]: #David's siblings
name = 'David'
output = run(0, x, sibling(x, name))
siblings = [x for x in output if x != name]
print("\nList of " + name + "'s siblings: ")
for item in siblings:
    print (item)
```

List of David's siblings:
William
Adam

```
In [35]: #Tiffany's uncles
name = "Tiffany"
name_father = run(0, x, father (x, name))[0]
output = run (0, x, uncle(x, name))
output = [x for x in output if x != name_father]
print("\nList of " + name + "'s uncles:")
for item in output:
    print (item)
```

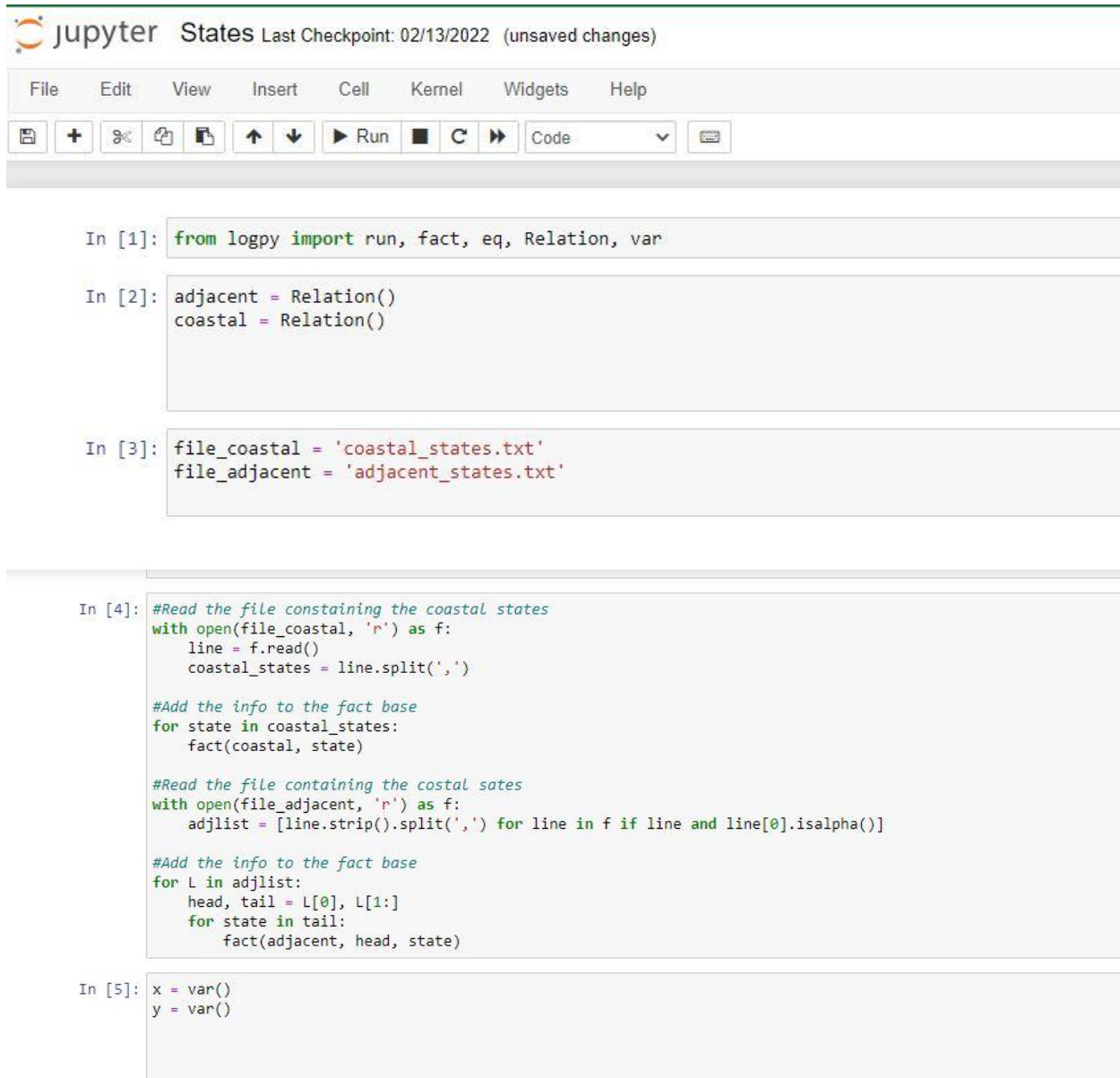
List of Tiffany's uncles:
William
Adam

```
In [36]: #All spouses

a, b, c = var(), var(), var()
output = run (0, (a,b), (father, a, c), (mother, b,c))
print ("\nList of all spouses: ")
for item in output:
    print('Husband:', item[0], '<==> Wife:', item[1])
```

List of all spouses:
Husband: William <==> Wife: Emma
Husband: John <==> Wife: Megan
Husband: David <==> Wife: Olivia
Husband: Adam <==> Wife: Lily

And similarly, I have did lab exercise of state the code view was this,



The image shows a JupyterLab interface with the title bar "jupyter States Last Checkpoint: 02/13/2022 (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar contains icons for saving, adding, deleting, copying, pasting, undo, redo, running, and a dropdown menu currently set to "Code".

```
In [1]: from logpy import run, fact, eq, Relation, var

In [2]: adjacent = Relation()
        coastal = Relation()

In [3]: file_coastal = 'coastal_states.txt'
        file_adjacent = 'adjacent_states.txt'

In [4]: #Read the file containing the coastal states
        with open(file_coastal, 'r') as f:
            line = f.read()
            coastal_states = line.split(',')

        #Add the info to the fact base
        for state in coastal_states:
            fact(coastal, state)

        #Read the file containing the costal sates
        with open(file_adjacent, 'r') as f:
            adjlist = [line.strip().split(',') for line in f if line and line[0].isalpha()]

        #Add the info to the fact base
        for L in adjlist:
            head, tail = L[0], L[1:]
            for state in tail:
                fact(adjacent, head, state)

In [5]: x = var()
        y = var()
```

```
In [6]: #Is Nevada adjacent tp Louisiana
output = run(0, x, adjacent('Nevada', 'Louisiana'))
print('\nIs Nevada adjacent to Louisiana?:')
print('Yes' if len(output) else 'No')
```

```
Is Nevada adjacent to Louisiana?:
No
```

```
In [7]: #states adjacent to oregon
output = run(0, x, adjacent('Oregon', x))
print('\nList of states adjacent to Oregon:')
for item in output:
    print(item)
```

```
List of states adjacent to Oregon:
Washington
Idaho
California
Nevada
```

```
In [8]: #States adjacent to Mississippi that are coastal
output = run(0, x, adjacent('Mississippi', x), coastal(x))
print('\nList of coastal states adjacent to Mississippi: ')
for item in output:
    print(item)
```

```
List of coastal states adjacent to Mississippi:
Alabama
Louisiana
```

```
In [9]: #List of 'n' states that border a coastal state
n = 7
output = run(n, x, coastal(y), adjacent(x, y))
print('\nList of ' +str(n) +' states that border a coastal state: ')
for item in output:
    print(item)
```

```
List of 7 states that border a coastal state:
Texas
Florida
South Dakota
New York
Pennsylvania
Connecticut
Maryland
```

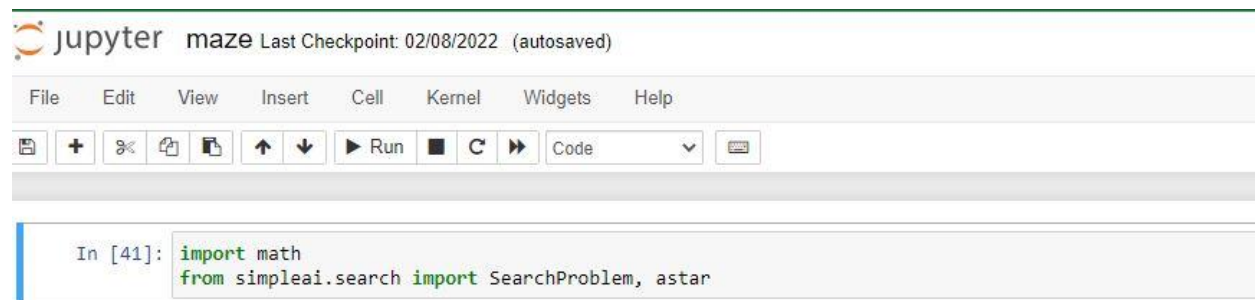
```
In [10]: #List of state that adjacent to the two givem states
output = run(0, x, adjacent('Arkansas', x), adjacent('Kentucky', x))
print('\nList pf states that are adjacent to Arkansas and Kentucky:')
for item in output:
    print (item)
```

```
List pf states that are adjacent to Arkansas and Kentucky:
Missouri
Tennessee
```


WEEK 3

In third week we have discuss about several others AI path finding algorithms. As per the lab exercise, we have did here lab exercise of maze solver, path finding algorithms using heuristic and popular A* algorithms. Similarly, I have also did next path finding project. A* algorithms lab what I did was look, likes.

At first I have install here necessary package



The screenshot shows a Jupyter Notebook window titled "maze" with a last checkpoint of "02/08/2022 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. A code cell is active, showing the following code:

```
In [41]: import math
from simpleai.search import SearchProblem, astar
```

Then I have define here maze problem solving functions

```
In [3]: class MazeSolver(SearchProblem):
        #initilize the constructor class of the MazeSolver
        def __init__(self, board):
            self.board = board
            self.goal = (0, 0)

        #Extract the inital file and final positions
        for y in range(len(self.board)):
            for x in range(len(self.board[y])):
                if self.board[y][x].lower() == "o":
                    self.initial = (x, y)
                elif self.board[y][x].lower() == "x":
                    self.goal = (x, y)
            super(MazeSolver, self).__init__(initial_state=self.initial)

        #Define the method that takes the actions
        #to arrive at the solutions
        def actions(self, state):
            actions = []
            for action in COSTS.keys():
                newx, newy = self.result(state, action)
                if self.board[newy][newx] != "#":
                    actions.append(action)
            return actions

        #Update the sate based on the actions
        def result(self, state, action):
            x, y = state
```

```

        if action.count("up"):
            y -=1
        if action.count("down"):
            y +=1
        if action.count("left"):
            x -=1
        if action.count("right"):
            x +=1

        new_state = (x, y)

        return new_state

    #Check if we have reached at goal
    def is_goal(self, state):
        return state == self.goal

    #compute the cost of an taking actions
    def cost(self, state, action, state2):
        return COSTS[action]

    #heuristic that arrive at the solutions
    def heuristic(self, state):
        x, y = state
        gx, gy = self.goal

        return math.sqrt((x-gx)**2 + (y-gy)**2)

```

At above I have define here different function. Just above function I have applied heuristic path finding rule.

At here I have define Map as well as initial and final point of path finding,

```

In [107]: #main fucntion or method

if __name__ == "__main__":

    #Defining MAP
    MAP = """
    #####
    #           #
    # #####   #
    # 0 #     #
    #   ##   #####
    #   #   ##   #
    #   #   #   #   #
    #   #####   # x
    #           #
    #####
    """

    #Conver MAP To List
    print(MAP)
    MAP = [list(x) for x in MAP.split("\n") if x]

    # Dfne cost of moving around the map
    cost_regular = 1.0
    cost_diagonal = 1.7

```

I have create here dictionary here I have set some diagonal cost,

```
# Create the cost Dictionary
COSTS = {
    "up":cost_regular,
    "down":cost_regular,
    "left":cost_regular,
    "right":cost_regular,
    "up left":cost_diagonal,
    "down left":cost_diagonal,
    "up right":cost_diagonal,
    "down right":cost_diagonal,
}
```

```
#####
#           #           #
# #####           #
# o #           #
#   ##           #####
#   #           #
#   #           #
#   #           #
#   #####       # # x
#           #   #
#           #   #
#####
```

```
In [112]: #creat maze solver object

problem = MazeSolver(MAP)
```

```
In [114]: #run the solver
result = astar(problem, graph_search=True)

#Extract the path
path = [x[1] for x in result.path()]
```

Final output of path finding algorithms.

```
In [115]: #Final output
print()
for y in range(len(MAP)):
    for x in range(len(MAP[y])):
        if (x,y) == problem.initial:
            print('o', end='')
        elif (x,y) == problem.goal:
            print('x', end='')
        elif (x,y) in path:
            print('*', end='')
        else:
            print(MAP[y][x], end='')
    print()
```

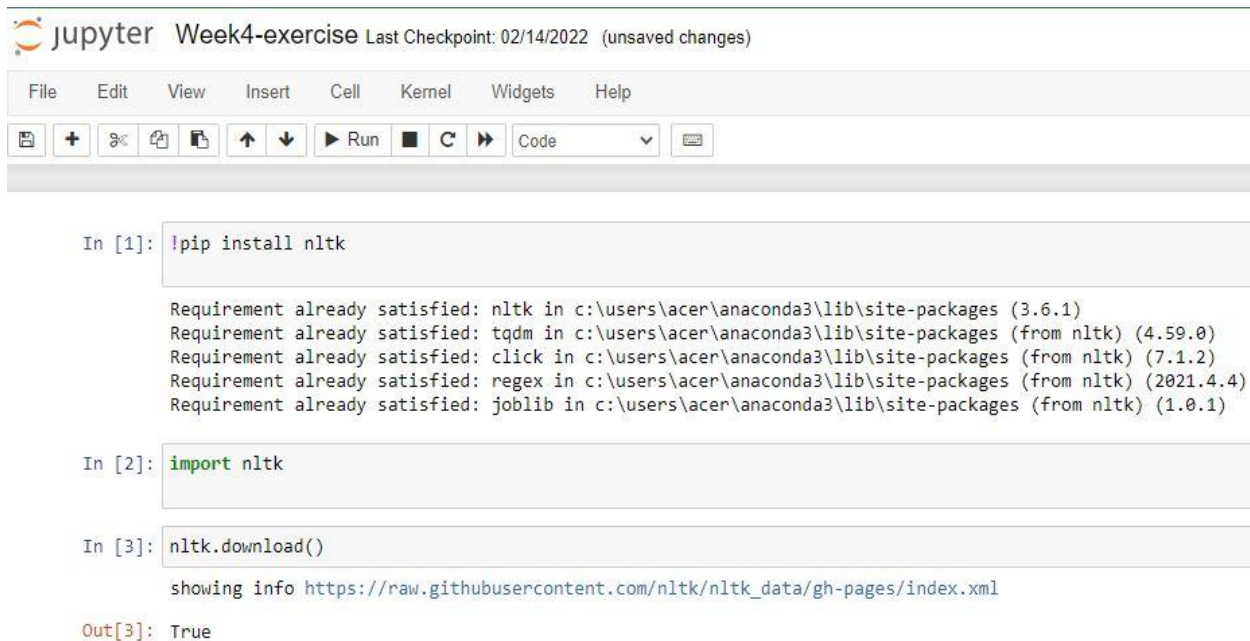
```
#####
#           #           #
# #####           #
# o #           #
# * ##           #####
# * #           #
# * #           #
# * #           * * *
# * ##### * * * # x
#           * * *
#           * * *
#####
```

In task provided I have did here several research where I have find various path finding algorithms. I have uploaded task what I did in canvas.

WEEK 4

In week 4 we have taught about the python AI Natural Language processing. Where I was gain chance to learn & research about the NLP. In this week I have did differ lab exercise and group tasks? I have uploaded what I did in lab exercise and as per tasks given in canvas e-portfolio. Some lab exercise what I did was this,

At first I have downloaded Natural language process package like nltk, import and downloaded in Jupiter Notebook



The screenshot shows a Jupyter Notebook window titled "Week4-exercise" with a last checkpoint of "02/14/2022 (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and code execution. The notebook contains three input cells:

```
In [1]: !pip install nltk
```

Requirement already satisfied: nltk in c:\users\acer\anaconda3\lib\site-packages (3.6.1)
Requirement already satisfied: tqdm in c:\users\acer\anaconda3\lib\site-packages (from nltk) (4.59.0)
Requirement already satisfied: click in c:\users\acer\anaconda3\lib\site-packages (from nltk) (7.1.2)
Requirement already satisfied: regex in c:\users\acer\anaconda3\lib\site-packages (from nltk) (2021.4.4)
Requirement already satisfied: joblib in c:\users\acer\anaconda3\lib\site-packages (from nltk) (1.0.1)

```
In [2]: import nltk
```

```
In [3]: nltk.download()
```

showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml

Out[3]: True

I have imported NLTK book,

```
In [*]: from nltk.book import*

*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
```

I have checks text and concordance

```
In [6]: text1
Out[6]: <Text: Moby Dick by Herman Melville 1851>

In [7]: text1.concordance("monstrous")

Displaying 11 of 11 matches:
ong the former , one was of a most monstrous size . ... This came towards us ,
ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have r
ll over with a heathenish array of monstrous clubs and spears . Some were thick
d as you gazed , and wondered what monstrous cannibal and savage could ever hav
that has survived the flood ; most monstrous and most mountainous ! That Himmal
they might scout at Moby Dick as a monstrous fable , or still worse and more de
th of Radney ." CHAPTER 55 Of the Monstrous Pictures of Whales . I shall ere l
ing Scenes . In connexion with the monstrous pictures of whales , I am strongly
ere to enter upon those still more monstrous stories of them which are to be fo
ght have been rummaged out of this monstrous cabinet there is no telling . But
of Whale - Bones ; for Whales of a monstrous size are oftentimes cast up dead u
```

I have checks here length of text also I have checks text similarity and how we define function in nltk & finally I have define lexical diversity function too

```
In [8]: len(text1)
Out[8]: 260819

In [9]: text1.similar("monstrous")

true contemptible christian abundant few part mean careful puzzled
mystifying passing curious loving wise doleful gamesome singular
delightfully perilous fearless

In [10]: def lexical_diversity(text):
          return len(text) / len(set(text))

          def percentage(total, count):
              return 100 * count/total

          lexical_diversity("Hello my name is pradeep")

Out[10]: 1.7142857142857142
```

Then similarly, I have did several exercise

```
In [13]: #Exercise 1
          print(len(text6))

          #number of items in Monty python and Holly grails

          count = 0
          for item in (text6):
              count += 1
          print("The total number of item in Monty python and Holly grails is: " , count)

16967
The total number of item in Monty python and Holly grails is: 16967
```



```
In [10]: #Exercise 2
text5.concordance("lol")
print("\n")
#text5.similar("lol")
```

Displaying 25 of 822 matches:

```
ast PART 24 / m boo . 26 / m and sexy lol U115 boo . JOIN PART he drew a girl w
ope he didnt draw a penis PART ewwww lol & a head between her legs JOIN JOIN s
a bowl i got a blunt an a bong ..... lol JOIN well , glad it worked out my cha
e " PART Hi U121 in ny . ACTION would lol @ U121 . . . but apparently she does
30 make sure u buy a nice ring for U6 lol U7 Hi U115 . ACTION isnt falling for
didnt ya hear !!!! PART JOIN geeshhh lol U6 PART hes deaf ppl here dont get it
es nobody here i wanna misbeahve with lol JOIN so read it . thanks U7 .. Im hap
ies want to chat can i talk to him !! lol U121 !!! forwards too lol JOIN ALL PE
k to him !! lol U121 !!! forwards too lol JOIN ALL PERvs ... redirect to U121 '
loves ME the most i love myself JOIN lol U44 how do u know that what ? jerkett
ng wrong ... i can see it in his eyes lol U20 = fiance Jerketts lmao wtf yah I
cooler by the minute what 'd I miss ? lol noo there too much work ! why not ??
that mean I want you ? U6 hello room lol U83 and this .. has been the grammar
the rule he 's in PM land now though lol ah ok i wont bug em then someone wann
flight to hell :) lmao bbl maybe PART LOL lol U7 it was me , U83 hahah U83 ! 80
ht to hell :) lmao bbl maybe PART LOL lol U7 it was me , U83 hahah U83 ! 808265
082653953 K-Fed got his ass kicked .. Lol . ACTION laughs . i got a first class
. i got a first class ticket to hell lol U7 JOIN any texas girls in here ? any
. whats up U155 i was only kidding . lol he 's a douchebag . Poor U121 i 'm bo
??? sits with U30 Cum to my shower . lol U121 . ACTION U1370 watches his nads
ur nad with a stick . ca u U23 ewwww lol *sniffs* ewwwwww PART U115 ! owww spl
ACTION is resisting . ur female right lol U115 beeeehave Remember the LAST tim
pm's me . charge that is 1.99 / min . lol @ innocent hahah lol .... yeah LOLOLO
is 1.99 / min . lol @ innocent hahah lol .... yeah LOLLOLLL U12 thats not nic
```

```
In [15]: #Exercise 3
```

```
def my_text(text):

    print("the toal length of the text:", len(text))

my_text("Hello my name is pradeep Khadka")
|
```

the toal length of the text: 31

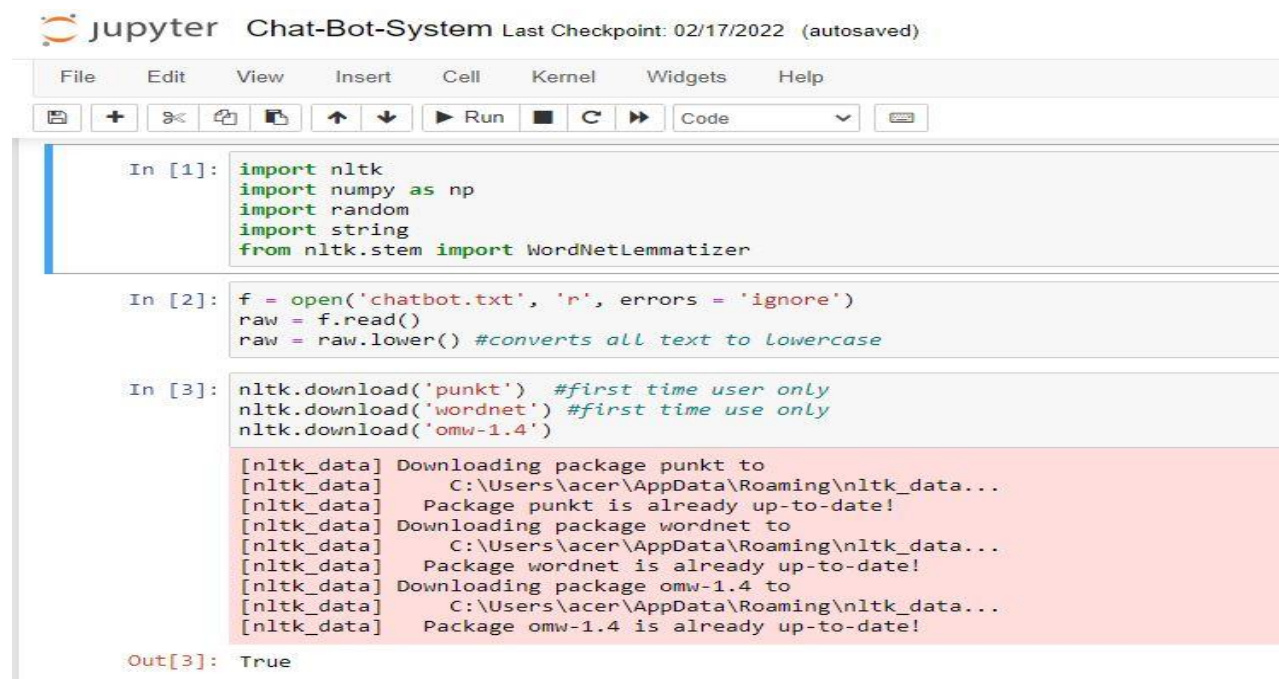
```
In [16]: print(text1)
```

<Text: Moby Dick by Herman Melville 1851>

WEEK 5:

In fifth week I have made Chat-bot system, for developing the chat-bot I have used here python programming language and I have imported here necessary python library. What I did to build the chat-bot system

First I have imported necessary python library, I have and read file. & I have converted all text file in to lower cases and I have did ntk file download.



```
In [1]: import nltk
import numpy as np
import random
import string
from nltk.stem import WordNetLemmatizer

In [2]: f = open('chatbot.txt', 'r', errors = 'ignore')
raw = f.read()
raw = raw.lower() #converts all text to lowercase

In [3]: nltk.download('punkt') #first time user only
nltk.download('wordnet') #first time use only
nltk.download('omw-1.4')

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\acer\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\acer\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data] C:\Users\acer\AppData\Roaming\nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!

Out[3]: True
```

I have converted list of token in to sentence and list into word

```
In [4]: sent_tokens = nltk.sent_tokenize(raw) #converts to list if sentence
word_tokens = nltk.word_tokenize(raw) #converts to list of words
```

```
In [5]: len(sent_tokens)
```

```
Out[5]: 271
```

```
In [6]: len(word_tokens)
```

```
Out[6]: 5317
```

In this steps I have created word lemmatized function which allows to remove the noise word and for converting possible uppercase to lowercase.

```
In [7]: lemmmer = nltk.stem.WordNetLemmatizer()

#WordNet is a semantically-oriented dictionary of
#English included in nltk

def lem_tokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)

def lem_normalize(text):
    return lem_tokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))
```

Then I have sets inputs and response

```
In [8]: GREETING_INPUTS = ("hello", 'hi', 'greeting', 'sup', "what's up", 'hey', "thank you robo", 'robo')
GREETING_RESPONSES = ["hi", "hey", "**nods**", "hi there", "hello", 'I am glad! You are talking to me',
                       "welcome there", "hello there!"]

def greeting(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
```

Then covert text into vector matric forms using cosine. So, for this I have used here python Sklearn library. Which is prebuild library also.

```
In [9]: # From scikit Learn Library, import the Tfidf vectorizer to convert a collections of raw documents
#to a matrix of TF-IDF features
from sklearn.feature_extraction.text import TfidfVectorizer

#import cosine similarity module from scikit Learn library
from sklearn.metrics.pairwise import cosine_similarity

#this will be used to find the similarity between words and entered
#by the user and the words in the corps
#this is the simplest possible implementation of a chatbot..
```

I have define here response function,

```
In [10]: def response(user_response):
    robo_response = ''
    sent_tokens.append(user_response)

    TfidfVec = TfidfVectorizer(tokenizer = lem_normalize, stop_words = 'english', analyzer = 'word')
    tfidf = TfidfVec.fit_transform(sent_tokens)

    vals = cosine_similarity(tfidf[-1], tfidf) #similarities between Last sentence (user_reponse) and other sentences
    # print("vals.shape = ", vals.shape) #shape = (1, 301)

    idx = vals.argsort()[0][-2] # index of the sentence that is the most similar to user_response. [-2] because
    #second Last of the setence ascending sorted scores (Last one is 1, cosine_similarity with
    # print("vals.argsort() =", vals.argsort())

    vals_flat = vals.flatten() # convert vals to one dinemision
    vals_flat.sort() #inplace sorting of flat

    req_tfidf = vals_flat[-2] #second Last of the ascending sorted vals (Last is 1 - similarity with itself)

    if req_tfidf == 0:
        robo_response = robo_response + "I am Sorry! I do not understand you"
        return robo_response
    else:
        robo_response = robo_response + sent_tokens[idx]
        return robo_response
```

Then finally when flag is true it will return response system

```
In [11]: flag = True

print("ROBO: My name is Robo. I will answer your queries about chatbots. if you want to exit, type Bye.")

while flag:
    user_response = input()
    user_response = user_response.lower()

    if user_response.lower() != 'bye':
        if (user_response == 'thanks' or user_response == 'thank you'):
            flag = False
            print("ROBO: you are Welcome! Bye for now!")
        else:
            if greeting(user_response) != None:
                print("ROBO: " + greeting(user_response))
            else:
                print("ROBO: ", end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag = False
        print("ROBO: Bye! take care ..")
```

```
ROBO: My name is Robo. I will answer your queries about chatbots. if you want to exit, type Bye.
who are you?
ROBO:
```

```
C:\Users\acer\anaconda3\lib\site-packages\sklearn\feature_extraction\text.py:388: UserWarning: Your stop_words may be inconsistent with your preprocessing. Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
warnings.warn('Your stop_words may be inconsistent with '
```

```
bird, jordan j.; ekart, aniko; faria, diego r. (june 2018).
hey
ROBO: hi there
what's up?
ROBO: I am Sorry! I do not understand you
what are the most recent notable programs?
ROBO: [10][11][12][13] more recent notable programs include a.l.i.c.e., jabberwacky and d.u.d.e (agence nationale de la recherche and cnrs 2006).
chatbot competitions focus on what?
ROBO: chatbot competitions focus on the turing test or more specific goals.
thanks
ROBO: you are Welcome! Bye for now!
```

WEEK 6

In week I was learn to install Weka and how we will work in weka for classification and predictions. In week 6, I was introduce about the machine learning model also. In weka, I have run different iris. irff file. Where I get chance to learn about how actual problem how they are classified. Several technique like 1R, zeroR, SVM, Gaussian Naïve, random forest, decision tree, etc. was I learn from this week. In this week we have did exercise on weather forecasting also. The process of how to install data set and search dataset was I learn from this week 6.

WEEK 7

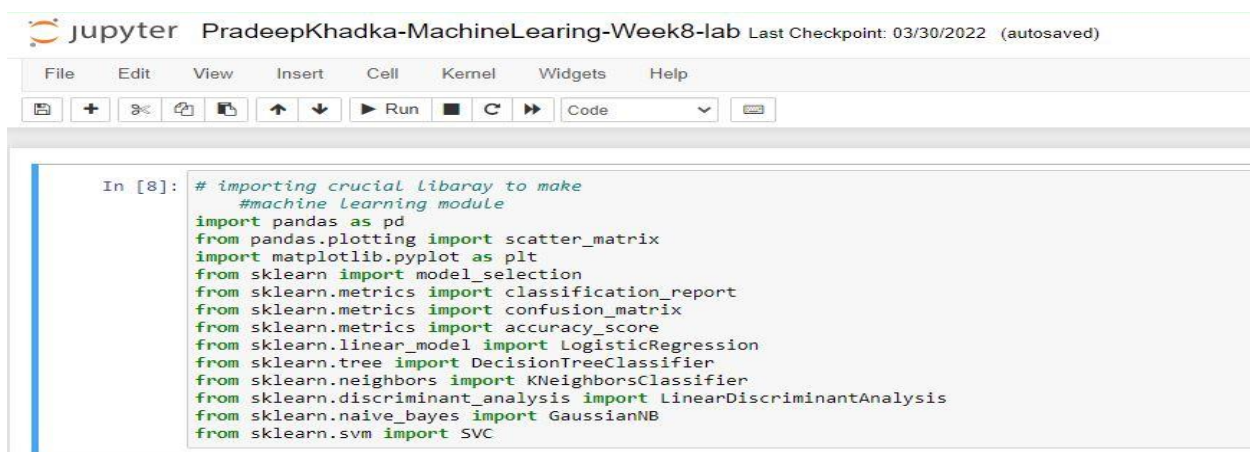
In week seven we have discussed about the machine learning and its types. Whole week was pass on the learning, discussing and researching about machine learning.

WEEK 8

In week 8 we have learn how to implements python crucial library for building machine learning model. In this week we have did lab about how to implement data sets in python machine learning prototype. And we have learn how to visualize data for this we have learn about the different plotting technique like bar-plot, histogram, scatterplot, violin plot, etc.

Similarly, we have learn different machine learning algorithms. On talking about the lab & task whatever I did I have uploaded in my canvas e-portfolio. In this week I have did lab of iris dataset.

At first I have imported necessary python library:



```
In [8]: # importing crucial library to make
        #machine Learning module
import pandas as pd
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

Then second, I have uploaded data sets & I have checks type as well as head of data sets:

```
In [16]: #Loading Datasets

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"
names = ["sepal-length", "sepal-width", "petal-length", "petal-width", "class"]
data = pd.read_csv(url, names=names)

In [17]: type(data)
Out[17]: pandas.core.frame.DataFrame

In [18]: #it will render top five data from the data sets
data.head()
Out[18]:
```

	sepal-length	sepal-width	petal-length	petal-width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Then third, I have describe datasets where I did different statistical analysis:

```
In [19]: data.shape
Out[19]: (150, 5)

In [20]: data.describe()
Out[20]:
```

	sepal-length	sepal-width	petal-length	petal-width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```

In [21]: #number of instaces that belong to each class
data.groupby("class").size()
Out[21]:
```

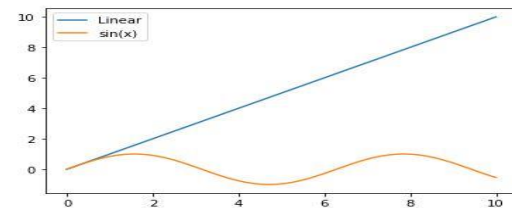
class	
Iris-setosa	50
Iris-versicolor	50
Iris-virginica	50

dtype: int64

Then I have used here data visualization technique

```
In [26]: #Data Vusualizations
import numpy as np
#prepare input data
x = np.linspace(0, 10, 100)
#plot the data
plt.plot(x, x, label = "Linear")
plt.plot(x, np.sin(x), label = "sin(x)")

#Legend
plt.legend()
Out[26]: <matplotlib.legend.Legend at 0x12c2244a6d0>
```

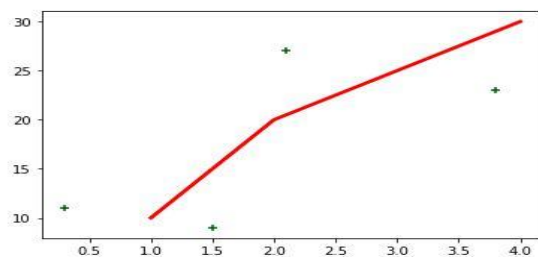


I have learn how to implement scatter plot:

```
In [25]: fig = plt.figure()
ax = fig.add_subplot(111)

x = [1,2,3,4]
y = [10,20,25,30]

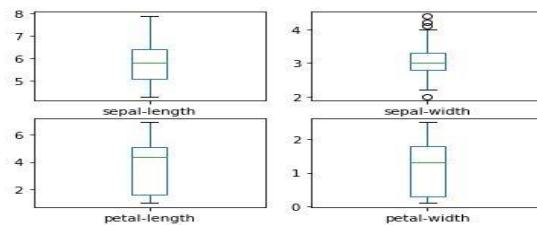
ax.plot(x, y, color = "red", linewidth = 3)
ax.scatter([0.3, 3.8, 1.5, 2.1], [11,23, 9, 27], color = "darkgreen", marker = '+')
Out[25]: <matplotlib.collections.PathCollection at 0x12c223f9ca0>
```



Then I have learn how to plot iris data in whisker plot:

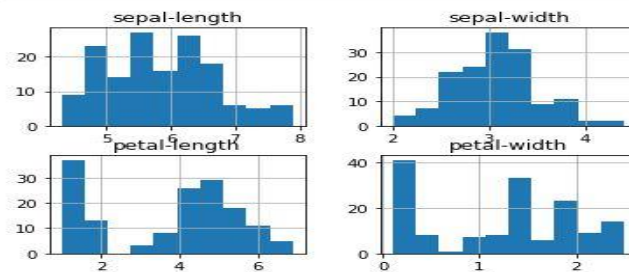
```
In [27]: #box and whisker plots
data.plot(kind = "box", subplots= True, layout=(2,2), sharex =False, sharey=False)
```

```
Out[27]: sepal-length      AxesSubplot(0.125,0.536818;0.352273x0.343182)
sepal-width      AxesSubplot(0.547727,0.536818;0.352273x0.343182)
petal-length      AxesSubplot(0.125,0.125;0.352273x0.343182)
petal-width      AxesSubplot(0.547727,0.125;0.352273x0.343182)
dtype: object
```



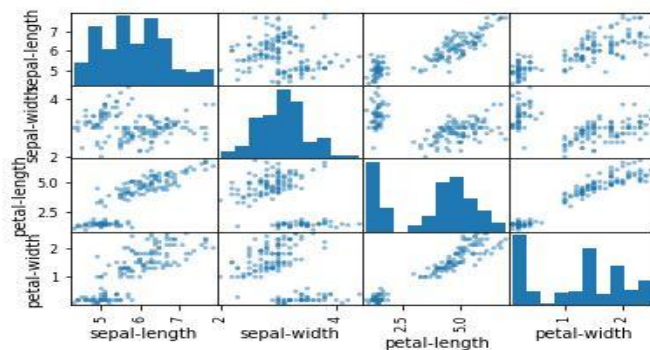
Then, I have plotted iris data in histogram plot

```
In [36]: #histogram
data.hist()
plt.show()
```



Finally, I have plotted iris datasets in scatter matrix plot:

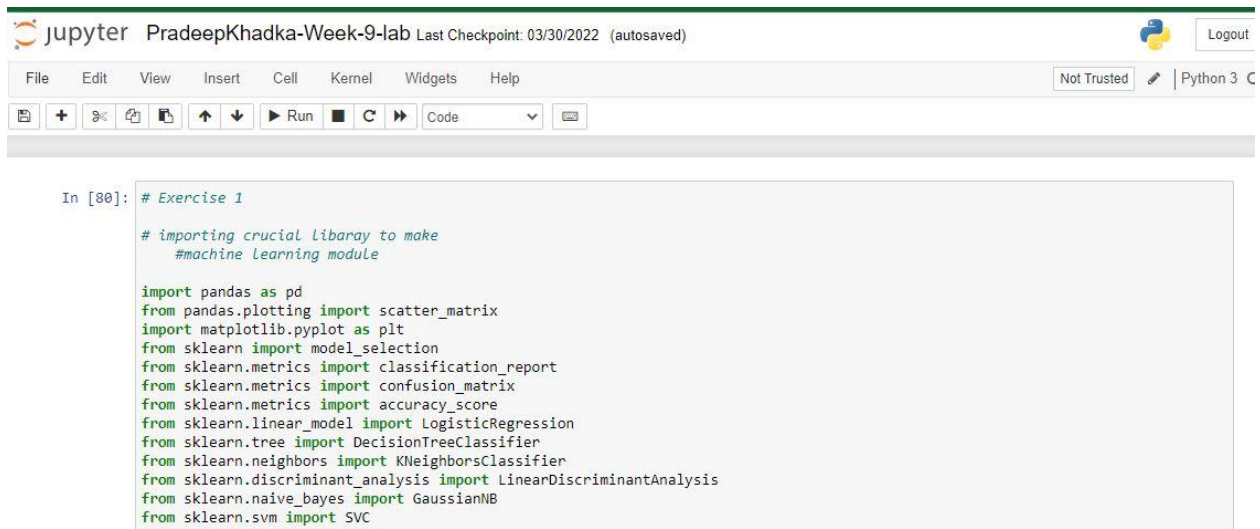
```
In [35]: scatter_matrix(data)
plt.show()
```



WEEK 9

In week 9, I have about the several clustering algorithms. There are various types of clustering algorithms do we have like k-means, k-Neighbors-Networking clustering algorithms. Similarly, in this week I have did lab as well as task as per provided. I have already uploaded in my canvas e-portfolio. Let see what was I did in my lab.

At first I have imported here python libraries



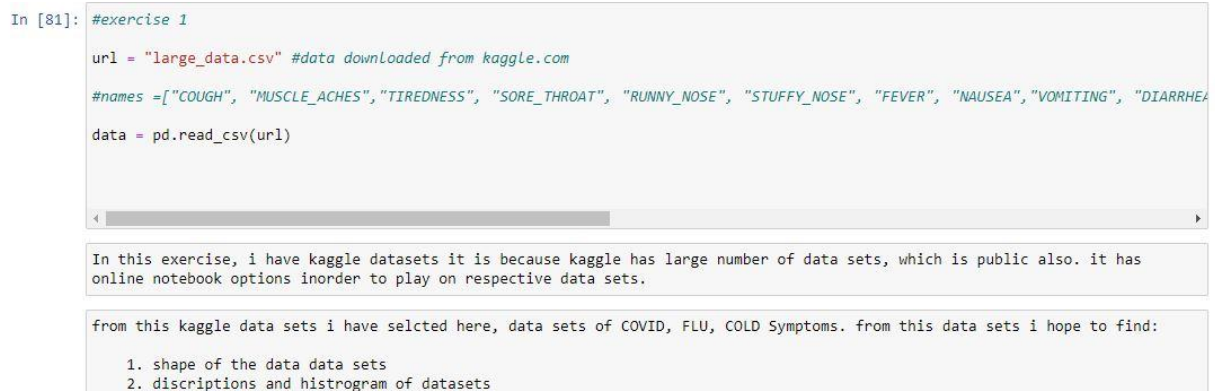
The screenshot shows a Jupyter Notebook interface with the title 'PradeepKhadka-Week-9-lab'. The last checkpoint is '03/30/2022 (autosaved)'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and code execution. The code cell, labeled 'In [80]:', contains the following Python code:

```
# Exercise 1

# importing crucial library to make
# machine learning module

import pandas as pd
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

Then I have uploaded csv datasets file



The screenshot shows a Jupyter Notebook interface with the title 'PradeepKhadka-Week-9-lab'. The last checkpoint is '03/30/2022 (autosaved)'. The code cell, labeled 'In [81]:', contains the following Python code:

```
#exercise 1

url = "large_data.csv" #data downloaded from kaggle.com

#names = ["COUGH", "MUSCLE_ACHES", "TIREDNESS", "SORE_THROAT", "RUNNY_NOSE", "STUFFY_NOSE", "FEVER", "NAUSEA", "VOMITING", "DIARRHEA"]

data = pd.read_csv(url)
```

Below the code cell, there is a text box with the following text:

In this exercise, i have kaggle datasets it is because kaggle has large number of data sets, which is public also. it has online notebook options inorder to play on respective data sets.

from this kaggle data sets i have selcted here, data sets of COVID, FLU, COLD Symptoms. from this data sets i hope to find:

1. shape of the data data sets
2. discriptions and histogram of datasets

Then I have did task as per exercise in first I have check shape of data sets then I have display top thirty rows of datasets including head (Which is second exercise also)

```
In [82]: #Exercise 2
|
| data.shape #this will give size of matrix
|
Out[82]: (44453, 21)
```

```
In [83]: data.head(30) #this will give top thirty data from the data sets
Out[83]:
```

	COUGH	MUSCLE_ACHES	TIREDNESS	SORE_THROAT	RUNNY_NOSE	STUFFY_NOSE	FEVER	NAUSEA	VOMITING	DIARRHEA	...	DIFFICULTY_BREATHI
0	0	0	0	1	0	1	0	0	0	0	0	...
1	0	0	0	1	0	0	0	0	0	0	0	...
2	0	1	1	1	1	0	0	0	0	0	0	...
3	0	0	0	0	1	1	0	0	0	0	0	...
4	0	0	0	1	0	1	0	0	0	0	0	...
5	0	0	0	0	0	0	0	0	0	0	0	...
6	1	0	0	0	0	1	1	0	0	0	0	...
7	0	1	1	1	1	0	0	0	0	0	0	...
8	1	1	0	0	0	1	0	0	0	0	0	...
9	1	0	1	1	1	1	0	0	0	0	0	...
10	0	0	0	0	0	1	1	0	0	0	0	...
11	1	1	1	0	0	1	0	0	0	0	0	...
12	1	0	0	1	1	1	0	0	0	0	0	...
13	1	0	0	1	1	0	0	0	0	0	0	...
14	1	0	0	0	0	0	1	0	0	0	0	...
15	1	0	1	0	0	1	1	0	0	0	0	...
16	1	1	1	1	0	1	0	0	0	0	0	...
17	1	0	0	0	0	1	1	0	0	0	0	...
18	1	1	0	0	0	1	0	0	0	0	0	...
19	1	0	1	1	1	1	0	0	0	0	0	...
20	1	1	0	0	0	0	0	0	0	0	0	...
21	0	1	0	0	1	0	1	0	0	0	0	...
22	0	0	0	0	0	1	0	0	0	0	0	...
23	0	1	0	1	1	1	1	0	0	0	0	...
24	1	0	0	1	1	1	0	0	0	0	0	...
25	1	1	1	0	0	1	0	0	0	0	0	...
26	0	1	1	0	0	0	0	0	0	0	0	...
27	1	0	0	1	1	1	1	0	0	0	0	...
28	0	0	0	0	0	1	0	0	0	0	0	...
29	0	1	1	0	0	0	0	0	0	0	0	...

30 rows x 21 columns

Then I have describe data sets according to different statistical aspects

```
In [84]: data.describe()
```

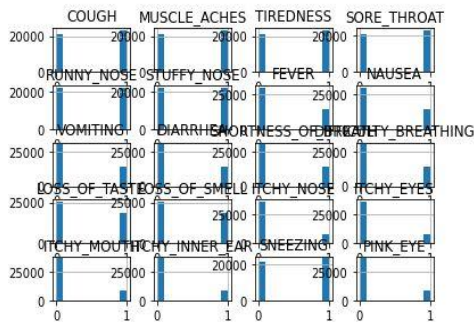
```
Out[84]:
```

	COUGH	MUSCLE_ACHES	TIREDFNESS	SORE_THROAT	RUNNY_NOSE	STUFFY_NOSE	FEVER	NAUSEA	VOMITING	DIARRHEA
count	44453.000000	44453.000000	44453.000000	44453.000000	44453.000000	44453.000000	44453.000000	44453.000000	44453.000000	44453.000000
mean	0.520662	0.519762	0.519897	0.519358	0.496232	0.495850	0.325400	0.324253	0.324523	0.323465
std	0.499579	0.499615	0.499610	0.499631	0.499991	0.499988	0.468529	0.468100	0.468201	0.467804
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

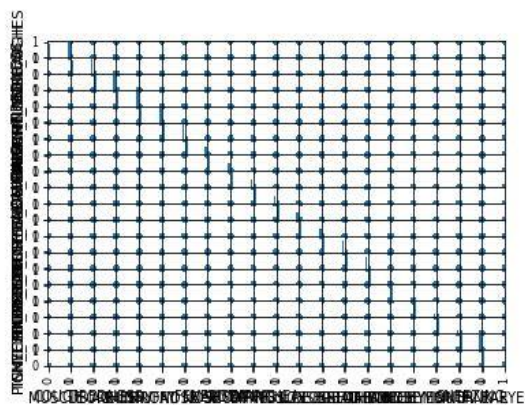
At here I have did third exercise to show data in histogram & scatter matrix plot

```
In [85]: ## Exercise 3
```

```
data.hist()
plt.show()
```



```
In [96]: scatter_matrix(data)
plt.show()
```



Here, I have checks info of the datasets:

```
In [98]: data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 44453 entries, 0 to 44452
Data columns (total 21 columns):
 #   Column                                  Non-Null Count  Dtype
---  -
 0   COUGH                                  44453 non-null  int64
 1   MUSCLE_ACHES                          44453 non-null  int64
 2   TIREDNESS                             44453 non-null  int64
 3   SORE_THROAT                           44453 non-null  int64
 4   RUNNY_NOSE                            44453 non-null  int64
 5   STUFFY_NOSE                           44453 non-null  int64
 6   FEVER                                  44453 non-null  int64
 7   NAUSEA                                44453 non-null  int64
 8   VOMITING                              44453 non-null  int64
 9   DIARRHEA                              44453 non-null  int64
10   SHORTNESS_OF_BREATH                   44453 non-null  int64
11   DIFFICULTY_BREATHING                  44453 non-null  int64
12   LOSS_OF_TASTE                         44453 non-null  int64
13   LOSS_OF_SMELL                         44453 non-null  int64
14   ITCHY_NOSE                            44453 non-null  int64
15   ITCHY_EYES                            44453 non-null  int64
16   ITCHY_MOUTH                           44453 non-null  int64
17   ITCHY_INNER_EAR                       44453 non-null  int64
18   SNEEZING                              44453 non-null  int64
19   PINK_EYE                              44453 non-null  int64
20   TYPE                                  44453 non-null  object
dtypes: int64(20), object(1)
memory usage: 7.5+ MB
```

At here I have use technique to remove duplicate data from the data sets & I have also check here whether a datasets is null or not:

```
In [99]: data=data.drop_duplicates()
```

```
In [100]: data.shape
```

```
Out[100]: (44453, 21)
```

```
In [101]: data.isnull().sum()
```

```
Out[101]: COUGH                                0
MUSCLE_ACHES                                0
TIREDNESS                                  0
SORE_THROAT                               0
RUNNY_NOSE                                0
STUFFY_NOSE                               0
FEVER                                      0
NAUSEA                                    0
VOMITING                                  0
DIARRHEA                                  0
SHORTNESS_OF_BREATH                       0
DIFFICULTY_BREATHING                      0
LOSS_OF_TASTE                             0
LOSS_OF_SMELL                             0
ITCHY_NOSE                                0
ITCHY_EYES                                0
ITCHY_MOUTH                               0
ITCHY_INNER_EAR                           0
SNEEZING                                  0
PINK_EYE                                  0
TYPE                                       0
dtype: int64
```

At here I have implemented technique to check columns of datasets & I have also checks and print here how many data of COUGH & MUSCLE_ACHES have 0 shape data:

```
In [102]: data.columns
```

```
Out[102]: Index(['COUGH', 'MUSCLE_ACHES', 'TIREDNESS', 'SORE_THROAT', 'RUNNY_NOSE',  
                'STUFFY_NOSE', 'FEVER', 'NAUSEA', 'VOMITING', 'DIARRHEA',  
                'SHORTNESS_OF_BREATH', 'DIFFICULTY_BREATHING', 'LOSS_OF_TASTE',  
                'LOSS_OF_SMELL', 'ITCHY_NOSE', 'ITCHY_EYES', 'ITCHY_MOUTH',  
                'ITCHY_INNER_EAR', 'SNEEZING', 'PINK_EYE', 'TYPE'],  
               dtype='object')
```

```
In [103]: print("No of people having cough", data[data['COUGH']==0].shape[0])
```

```
No of people having cough 21308
```

```
In [104]: print("No of people having ,ucle acjes", data[data['MUSCLE_ACHES']==0].shape[0])
```

```
No of people having ,ucle acjes 21348
```

WEEK 10

In week 10 I have learn about the artificial neural network. Where I get chance to learn about CNN, ANN, RNN, etc. In this week I have did several task and lab exercise? Respective task and lab exercise what I did was I have uploaded in my canvas assignment e-portfolio.

Some code view of the lab exercise is: (In this week I have built my own neural network)

```
jupyter Week10-Python-NeuralNetwork-pradeepKhadka Last Checkpoint: 03/31/2022 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted

In [67]: import numpy as np

In [68]: # Each row is a training example, each column is a feature [X1, X2, X3]
X = np.array([[0,0,1],[0,1,1], [1,0,1], [1,1,1]], dtype = float)
y = np.array([[0], [1],[1],[0]], dtype=float)

In [69]: print(X)
print('Shape of X = ', X.shape)
print(10*'-')
print(y)
print('Shape of y =', y.shape)

[[0. 0. 1.]
 [0. 1. 1.]
 [1. 0. 1.]
 [1. 1. 1.]]
Shape of X = (4, 3)
-----
[[0.]
 [1.]
 [1.]
 [0.]]
Shape of y = (4, 1)

In [70]: #Activation functions

def sigmoid(t):
    return 1/(1+np.exp(-t))

#Derivative of sigmoid
def sigmoid_derivative(p):
    return p * (1-p)

#scratch building Neural Network class in python
import numpy as np
#class

class NeuralNetwork:
    def __init__(self, x, y): #initlizzing constructor
        self.input = x
        self.weights1 = np.random.rand(self.input.shape[1],4)
        self.weights2 = np.random.rand(4,1)
        self.y = y
        self.output = np.zeros(y.shape)

    #For Feed-forward
    def feedforward(self):
        self.layer1 = sigmoid(np.dot(self.input, self.weights1))
        self.layer2 = sigmoid(np.dot(self.layer1, self.weights2))
        return self.layer2
```

```

def backprop(self):
    #Application of the chain Rule to find derivative of the Loss functions
    d_weights2 = np.dot(self.layer1.T, 2*(self.y-self.output) * sigmoid_derivative(self.output))
    d_weights1 = np.dot(self.input.T, np.dot(2*(self.y -self.output) * sigmoid_derivative(self.output),
        self.weights2.T)*sigmoid_derivative(self.layer1))

    # update the weights with the derivative (slope) of the Loss function
    self.weights1 += d_weights1
    self.weights2 += d_weights2

def train(self):
    self.output = self.feedforward()
    self.backprop()

```

```

In [71]: print ("Input: \n" +str(X))
        print("\nActual Output: \n" + str(y))

```

```

Input:
[[0. 0. 1.]
 [0. 1. 1.]
 [1. 0. 1.]
 [1. 1. 1.]]

Actual Output:
[[0.]
 [1.]
 [1.]
 [0.]]

```

```

In [72]: NN = NeuralNetwork(X, y)

        loss_lst = []

        for i in range(1000):    #trains the NN 1,000 times
            loss = np.mean(np.square(y - NN.feedforward()))
            loss_lst.append(loss)

            if i % 100 == 0:
                print("For iterations # " + str(i) + "\n")
                print("predicted Output : \n" + str(NN.feedforward()))
                print("Loss: \n" + str(loss))
                print ("-----\n")

            NN.train()

```

```

For iterations # 800

```

```

predicted Output :
[[0.4936785 ]
 [0.51125645]
 [0.49190311]
 [0.50995457]]
Loss:
0.2502012091714111
-----

```

```

For iterations # 900

```

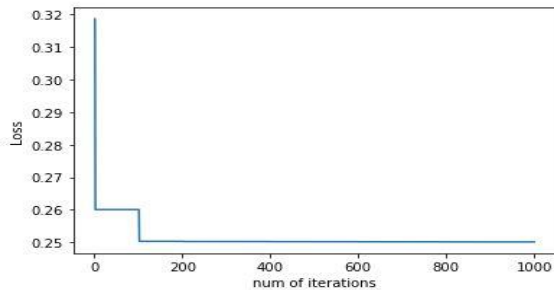
```

predicted Output :

```

```
In [73]: #Loss plot
import matplotlib.pyplot as plt
plt.plot(np.arange(1, len(loss_lst)+1), loss_lst)
plt.xlabel('num of iterations')
plt.ylabel('Loss')
plt.plot()
```

Out[73]: []



WEEK 11

In week 11 I was introduced about the computer vision, where we have learn image processing technique using CNN model. At here I have did lab of time series calculation of Convolution Neural network. For this lab I have chosen here google Colab.Reserarch in order to write the code. It is because at this time I was going to use the Keras library. Which don't support in Jupiter notebook. Rest of the other lab single variate, 1D variate, multi variate I have uploaded in my canvas assignment e-portfolio.

1-D Variate CNN Model:

```
Week11-lab-1D-CNN-Model-PradeepKhadka-.ipynb ☆
File Edit View Insert Runtime Tools Help Last edited on March 31

+ Code + Text

from numpy import array
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
--NORMAL--

[ ] #split a univariate sequence into samples
def split_sequence(sequence, n_steps):
    X, y = list(), list()
    for i in range(len(sequence)):
        #find the end of this patterns
        end_ix = i + n_steps
        #check if we are beyond the sequence
        if end_ix > len(sequence) - 1:
            break;
        #gather input and output parts of the patterns
        seq_x, seq_y = sequence[i: end_ix], sequence[end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)
```



```
[ ] #define input sequence
raw_seq = [10,20,30,40,50,60,70,80,90]

[ ] #choose a number of time steps
n_steps = 3

[ ] #split into samples
X, y = split_sequence(raw_seq, n_steps)

[ ] #reshape from [samples, timesteps] into [samples, timesteps, features]
n_features = 1
X = X.reshape((X.shape[0], X.shape[1], n_features))

[ ] #define models
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(n_steps, n_features)))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(50, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

```
[ ] #fit model
model.fit(X, y, epochs=1000, verbose=0)

<keras.callbacks.History at 0x7faba7970990>

[ ] #demonstrate predictions
x_input = array([70, 80, 90])
x_input = x_input.reshape((1, n_steps, n_features))
yhat = model.predict(x_input, verbose = 0)

print(yhat)

[[101.15428]]
```

Multi-variate CNN Model:

```
Week11-lab-Multivariate-multi-headed-cnn-model-PradeepKhadka.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text

[1] # multivariate multi-headed 1d cnn example
from numpy import array
from numpy import hstack
from keras.models import Model
from keras.layers import Input
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from keras.layers.merge import concatenate

[2] #split a multivariate sequence into samples

def split_sequences(sequences, n_steps):
    X, y = list(), list()
    for i in range(len(sequences)):
        #find the end of this pattern
        end_ix = i + n_steps
        #checks if we are beyond the datasets
        if end_ix > len(sequences):
            break
        #gathering input and output parts of the pattern
        seq_x, seq_y = sequences[i:end_ix, :-1], sequences[end_ix-1, -1]
        X.append(seq_x)
```

```
[2] y.append(seq_y)
    return array(X), array(y)

[3] #define input sequences
in_seq1 = array([10,20,30,40,50,60,70,80,90])
in_seq2 = array([15,25,35,45,55,65,75,85,95])
out_seq = array([in_seq1[i] + in_seq2[i] for i in range(len(in_seq1))])

[4] #convert to [rows, columns] structure
in_seq1 = in_seq1.reshape((len(in_seq1), 1))
in_seq2 = in_seq2.reshape((len(in_seq2), 1))
out_seq = out_seq.reshape((len(out_seq), 1))

[10] dataset = hstack((in_seq1, in_seq2, out_seq))

[5] #choose a number of time steps
n_steps = 3

[11] #convert into input & output
X, y = split_sequences(dataset, n_steps)
```

```
[7] # one time series per head
n_features = 1

[12] # separate input data
X1 = X[:, :, 0].reshape(X.shape[0], X.shape[1], n_features)
X2 = X[:, :, 1].reshape(X.shape[0], X.shape[1], n_features)

[13] # first input model
visible1 = Input(shape=(n_steps, n_features))
cnn1 = Conv1D(filters=64, kernel_size=2, activation='relu')(visible1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Flatten()(cnn1)

[14] # second input model
visible2 = Input(shape=(n_steps, n_features))
cnn2 = Conv1D(filters=64, kernel_size=2, activation='relu')(visible2)
cnn2 = MaxPooling1D(pool_size=2)(cnn2)
cnn2 = Flatten()(cnn2)
```

```
[15] # merge input models
merge = concatenate([cnn1, cnn2])
dense = Dense(50, activation='relu')(merge)
output = Dense(1)(dense)
model = Model(inputs=[visible1, visible2], outputs=output)
model.compile(optimizer='adam', loss='mse')

[16] # fit model
model.fit([X1, X2], y, epochs=1000, verbose=0)

<keras.callbacks.History at 0x7f9c2080ba10>
```

```
# demonstrate prediction
x_input = array([[80, 85], [90, 95], [100, 105]])
x1 = x_input[:, 0].reshape((1, n_steps, n_features))
x2 = x_input[:, 1].reshape((1, n_steps, n_features))
yhat = model.predict([x1, x2], verbose=0)

print(yhat)
--NORMAL--
[[205.7665]]
```

WEEK 12

In week 12 we have did discussed on the future AI as well as research over it.

E-Portfolio Canvas Link:

<https://canvas.sunderland.ac.uk/eportfolios/7244?verifier=hjT0GavB0nbyb18FkhhUUM7MjULrMARjUIOkmbXx>