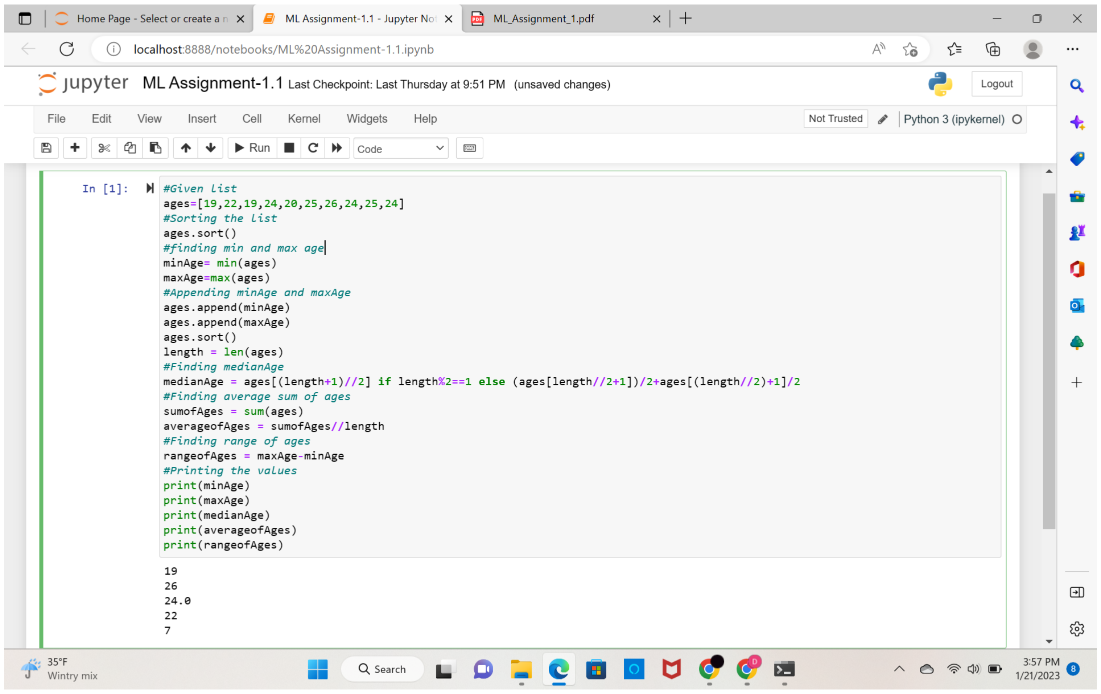
MACHINE LEARNING
ASSIGNMENT-1

NAME - DHARANI PULIMAMIDI **700745377**

Video Link - https://drive.google.com/file/d/1kXRi3SbQ2uoKFy0bSX2alMqR63KKEwWo/view?usp=share_link

1. Given list of 10 students ages as [19,22,19,24,20,25,26,24,25,24]
   Firstly sort the list by using sort function and find min, max age of student . Now append the min age and max age by append() function. Median age of the student can be calculated by length, ages[length+1//2] . Average age can be printed by sum(), Range can be calculated by maxage - minage.

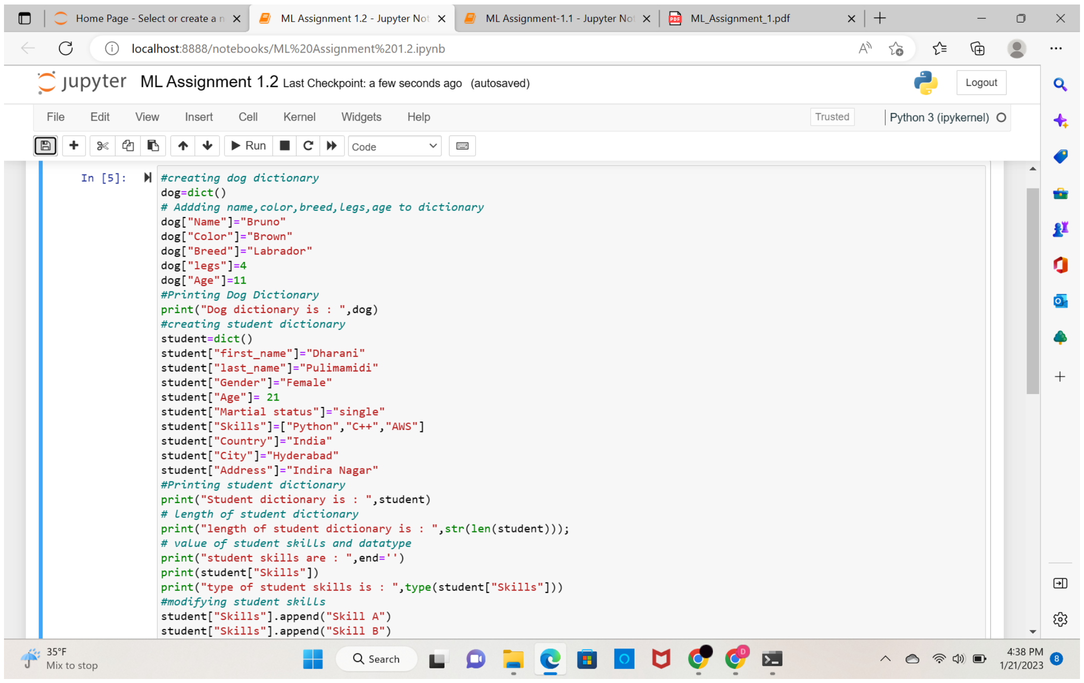


```
In [1]:
#Given list
ages=[19,22,19,24,20,25,26,24,25,24]
#Sorting the list
ages.sort()
#finding min and max age
minAge= min(ages)
maxAge=max(ages)
#Appending minAge and maxAge
ages.append(minAge)
ages.append(maxAge)
ages.sort()
length = len(ages)
#Finding medianAge
medianAge = ages[(length+1)//2] if length%2==1 else (ages[length//2+1])/2+ages[(length//2)+1]/2
#Finding average sum of ages
sumofAges = sum(ages)
averageofAges = sumofAges//length
#Finding range of ages
rangeofAges = maxAge-minAge
#Printing the values
print(minAge)
print(maxAge)
print(medianAge)
print(averageofAges)
print(rangeofAges)
```

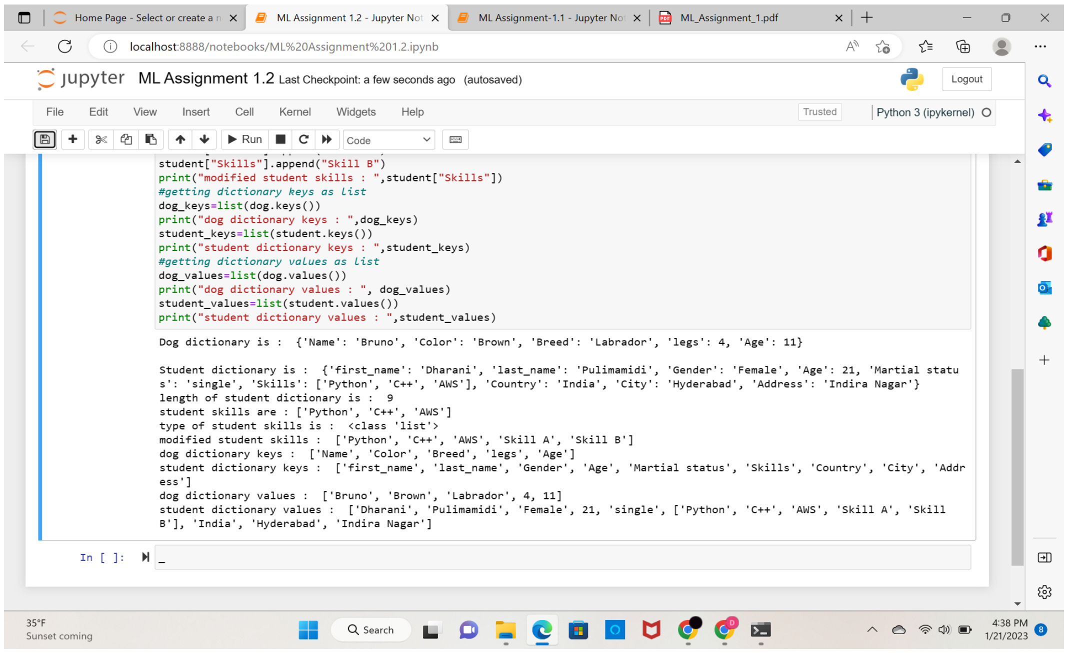
```
19
26
24.0
22
7
```

2. Create an empty dictionary as Dog. Add Name, color,breed,legs,age to the dictionary and print those values. Now create another dictionary as student dictionary and add the given keys to the dictionary. The length of the student dictionary can be printed by str(len(student)). The type of the student is checked by type(student["Skills"]). Modify by

using append(). To get dictionary keys use list(dog.keys()) , for dictionary values use list(dog.values()).



```
In [5]:
#creating dog dictionary
dog=dict()
# Addding name,color,breed,legs,age to dictionary
dog["Name"]="Bruno"
dog["Color"]="Brown"
dog["Breed"]="Labrador"
dog["legs"]=4
dog["Age"]=11
#Printing Dog Dictionary
print("Dog dictionary is : ",dog)
#creating student dictionary
student=dict()
student["first_name"]="Dharani"
student["last_name"]="Pulimamidi"
student["Gender"]="Female"
student["Age"]= 21
student["Martial status"]="single"
student["Skills"]=["Python","C++","AWS"]
student["Country"]="India"
student["City"]="Hyderabad"
student["Address"]="Indira Nagar"
#Printing student dictionary
print("Student dictionary is : ",student)
# length of student dictionary
print("length of student dictionary is : ",str(len(student)));
# value of student skills and datatype
print("student skills are : ",end='')
print(student["Skills"])
print("type of student skills is : ",type(student["Skills"]))
#modifying student skills
student["Skills"].append("Skill A")
student["Skills"].append("Skill B")
```



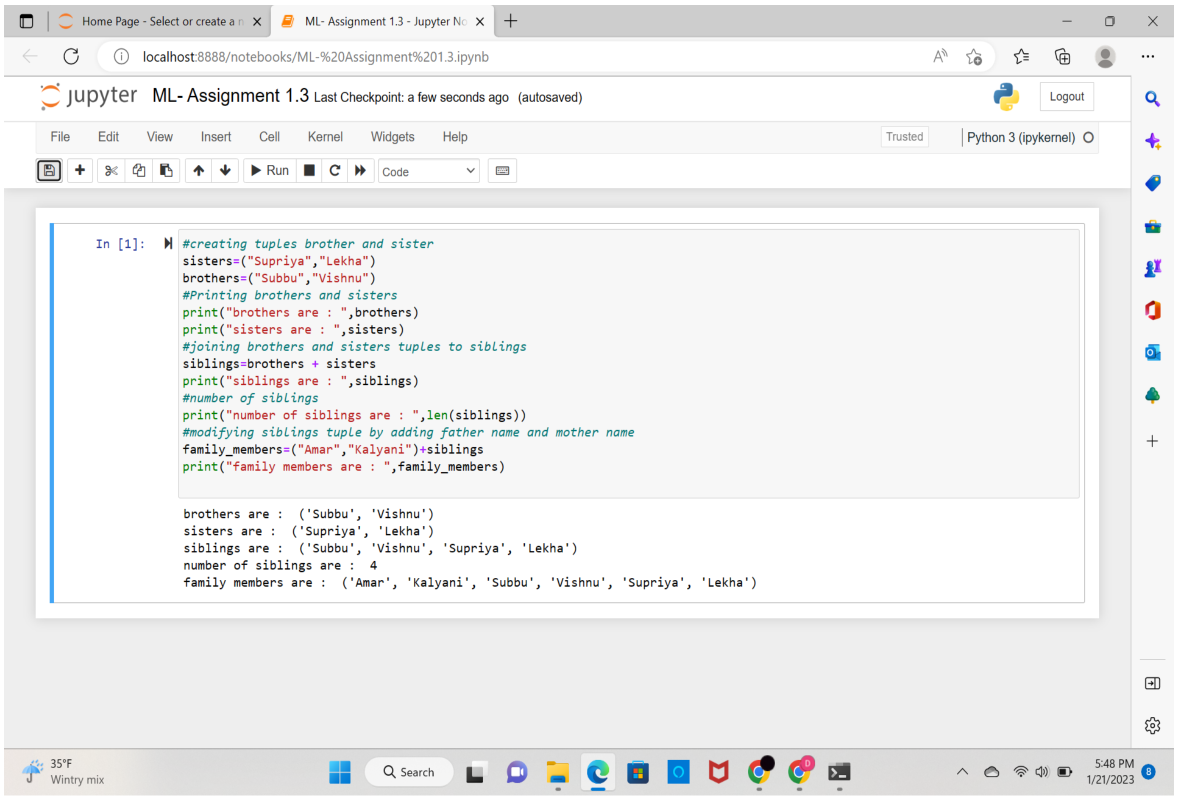
```
student["Skills"].append("Skill B")
print("modified student skills : ",student["Skills"])
#getting dictionary keys as list
dog_keys=list(dog.keys())
print("dog dictionary keys : ",dog_keys)
student_keys=list(student.keys())
print("student dictionary keys : ",student_keys)
#getting dictionary values as list
dog_values=list(dog.values())
print("dog dictionary values : ", dog_values)
student_values=list(student.values())
print("student dictionary values : ",student_values)
```

```
Dog dictionary is :  {'Name': 'Bruno', 'Color': 'Brown', 'Breed': 'Labrador', 'legs': 4, 'Age': 11}

Student dictionary is :  {'first_name': 'Dharani', 'last_name': 'Pulimamidi', 'Gender': 'Female', 'Age': 21, 'Martial statu
s': 'single', 'Skills': ['Python', 'C++', 'AWS'], 'Country': 'India', 'City': 'Hyderabad', 'Address': 'Indira Nagar'}
length of student dictionary is :  9
student skills are : ['Python', 'C++', 'AWS']
type of student skills is :  <class 'list'>
modified student skills :  ['Python', 'C++', 'AWS', 'Skill A', 'Skill B']
dog dictionary keys :  ['Name', 'Color', 'Breed', 'legs', 'Age']
student dictionary keys :  ['first_name', 'last_name', 'Gender', 'Age', 'Martial status', 'Skills', 'Country', 'City', 'Addr
ess']
dog dictionary values :  ['Bruno', 'Brown', 'Labrador', 4, 11]
student dictionary values :  ['Dharani', 'Pulimamidi', 'Female', 21, 'single', ['Python', 'C++', 'AWS', 'Skill A', 'Skill
B'], 'India', 'Hyderabad', 'Indira Nagar']
```

```
In [ ]: _
```

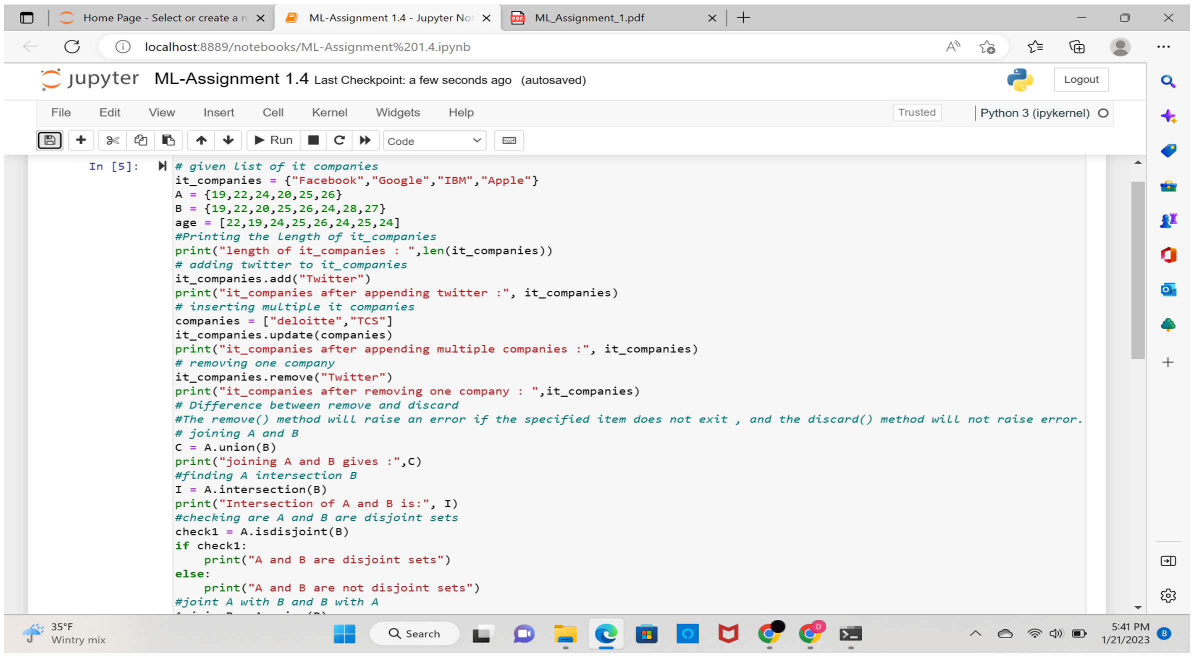
3. Create a tuple as sisters and brothers and print those values.Join brothers and sisters by using arithmetic operation perform Addition (+). The number of siblings can be calculated by len(siblings).Append Father and Mother name to siblings by arithmetic operator and print as family_members.


Home Page - Select or create a n
ML- Assignment 1.3 - Jupyter No
localhost:8888/notebooks/ML-%20Assignment%201.3.ipynb
jupyter ML- Assignment 1.3 Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:

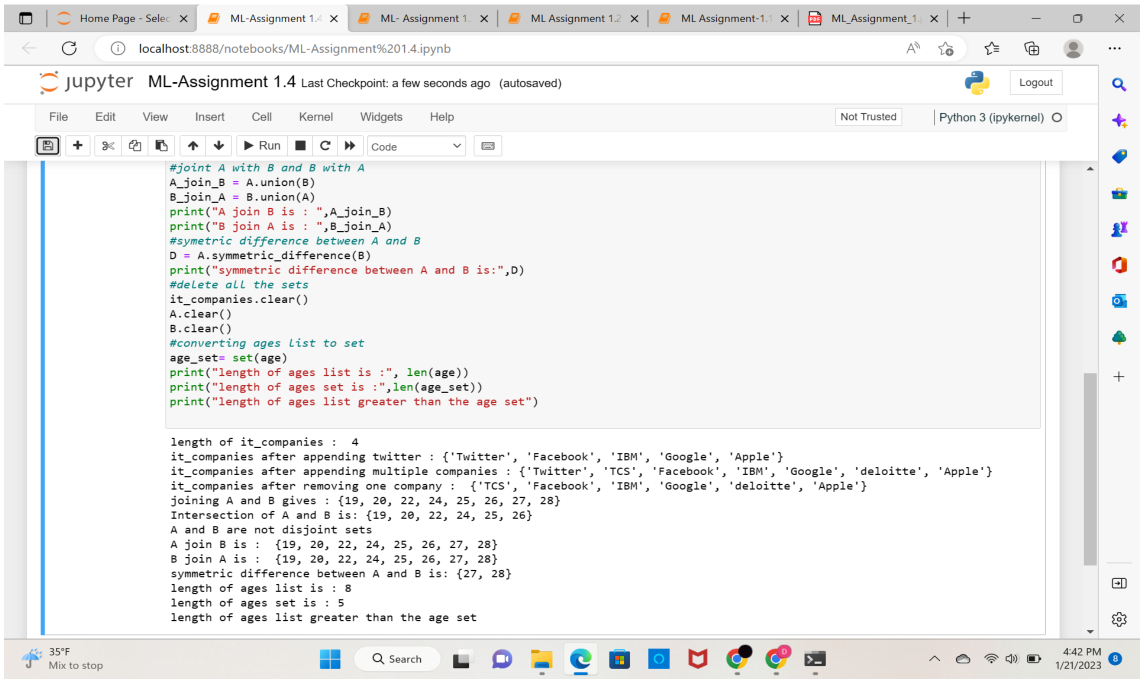
```python
#creating tuples brother and sister
sisters=("Supriya","Lekha")
brothers=("Subbu","Vishnu")
#Printing brothers and sisters
print("brothers are : ",brothers)
print("sisters are : ",sisters)
#joining brothers and sisters tuples to siblings
siblings=brothers + sisters
print("siblings are : ",siblings)
#number of siblings
print("number of siblings are : ",len(siblings))
#modifying siblings tuple by adding father name and mother name
family_members=("Amar","Kalyani")+siblings
print("family members are : ",family_members)
```

```
brothers are :  ('Subbu', 'Vishnu')
sisters are :  ('Supriya', 'Lekha')
siblings are :  ('Subbu', 'Vishnu', 'Supriya', 'Lekha')
number of siblings are :  4
family members are :  ('Amar', 'Kalyani', 'Subbu', 'Vishnu', 'Supriya', 'Lekha')
```

4. The length of the it_companies can printed by len(it_companies). Append “Twitter” by .add(“Twitter”) to it_companies.For inserting multiple values use .update() and to remove values use .remove() to it_comapnies.The difference between remove and discard raises an error so cant execute . For joining A and B use .union(). For intersection use .intersection(). For checking disjoint sets use if else loops.Joining A with B can be done by A_join_B , //ly B with A can be done. The symmetric difference can be done by A.symmetric_difference(B). For deleting sets we use .clear().Finally the convert to set and the length can be calculated by len() function.



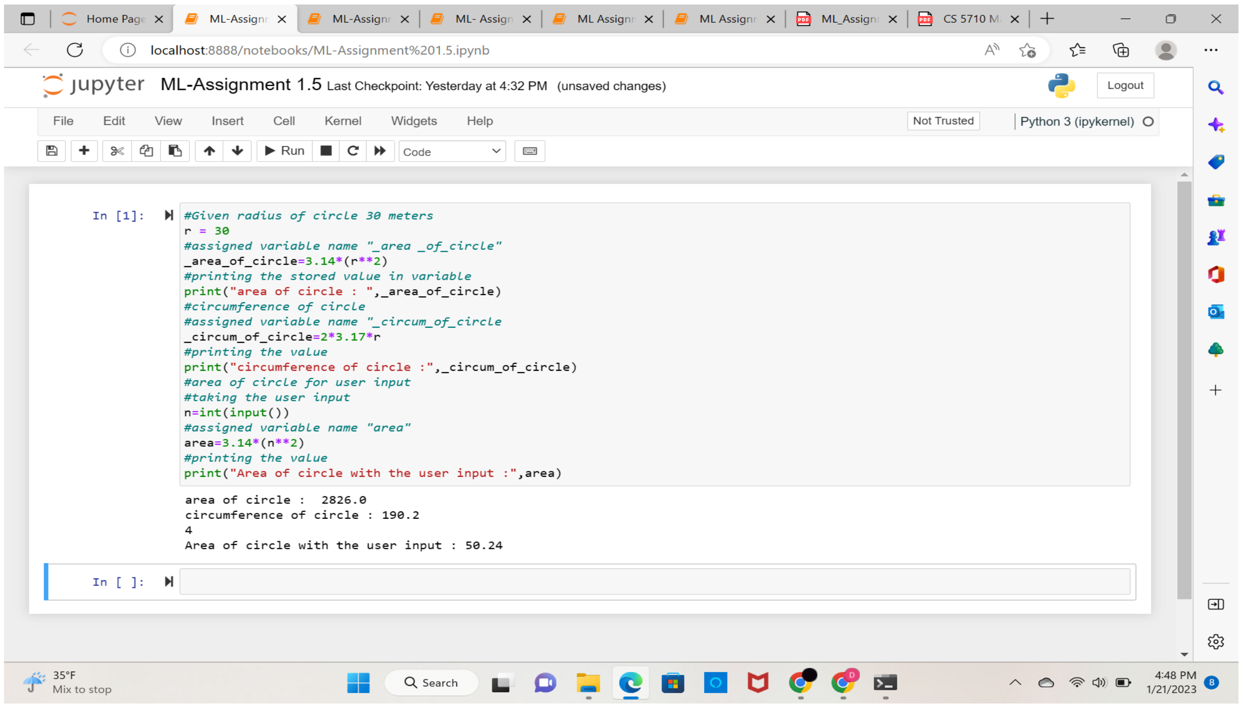
```
In [5]:  # given list of it companies
it_companies = {"Facebook","Google","IBM","Apple"}
A = {19,22,24,20,25,26}
B = {19,22,20,25,26,24,28,27}
age = [22,19,24,25,26,24,25,24]
#Printing the length of it_companies
print("length of it_companies : ",len(it_companies))
# adding twitter to it_companies
it_companies.add("Twitter")
print("it_companies after appending twitter :", it_companies)
# inserting multiple it companies
companies = ["deloitte","TCS"]
it_companies.update(companies)
print("it_companies after appending multiple companies :", it_companies)
# removing one company
it_companies.remove("Twitter")
print("it_companies after removing one company : ",it_companies)
# Difference between remove and discard
#The remove() method will raise an error if the specified item does not exit , and the discard() method will not raise error.
# joining A and B
C = A.union(B)
print("joining A and B gives :",C)
#finding A intersection B
I = A.intersection(B)
print("Intersection of A and B is:", I)
#checking are A and B are disjoint sets
check1 = A.isdisjoint(B)
if check1:
    print("A and B are disjoint sets")
else:
    print("A and B are not disjoint sets")
#joint A with B and B with A
```



```
#joint A with B and B with A
A_join_B = A.union(B)
B_join_A = B.union(A)
print("A join B is : ",A_join_B)
print("B join A is : ",B_join_A)
#symetric difference between A and B
D = A.symmetric_difference(B)
print("symmetric difference between A and B is:",D)
#delete all the sets
it_companies.clear()
A.clear()
B.clear()
#converting ages list to set
age_set= set(age)
print("length of ages list is :", len(age))
print("length of ages set is :",len(age_set))
print("length of ages list greater than the age set")
```

```
length of it_companies :  4
it_companies after appending twitter : {'Twitter', 'Facebook', 'IBM', 'Google', 'Apple'}
it_companies after appending multiple companies : {'Twitter', 'TCS', 'Facebook', 'IBM', 'Google', 'deloitte', 'Apple'}
it_companies after removing one company :  {'TCS', 'Facebook', 'IBM', 'Google', 'deloitte', 'Apple'}
joining A and B gives : {19, 20, 22, 24, 25, 26, 27, 28}
Intersection of A and B is: {19, 20, 22, 24, 25, 26}
A and B are not disjoint sets
A join B is :  {19, 20, 22, 24, 25, 26, 27, 28}
B join A is :  {19, 20, 22, 24, 25, 26, 27, 28}
symmetric difference between A and B is: {27, 28}
length of ages list is : 8
length of ages set is : 5
length of ages list greater than the age set
```

5. The radius of the circle is given as 30 meters.The area of circle is 3.14*(r**2). So the area of circle can be printed by it.The circumference of circle is 2*3.14*r , print it to another assigned variable _circum_of_circle. The user needs to give values to the input to calculate area of circle.

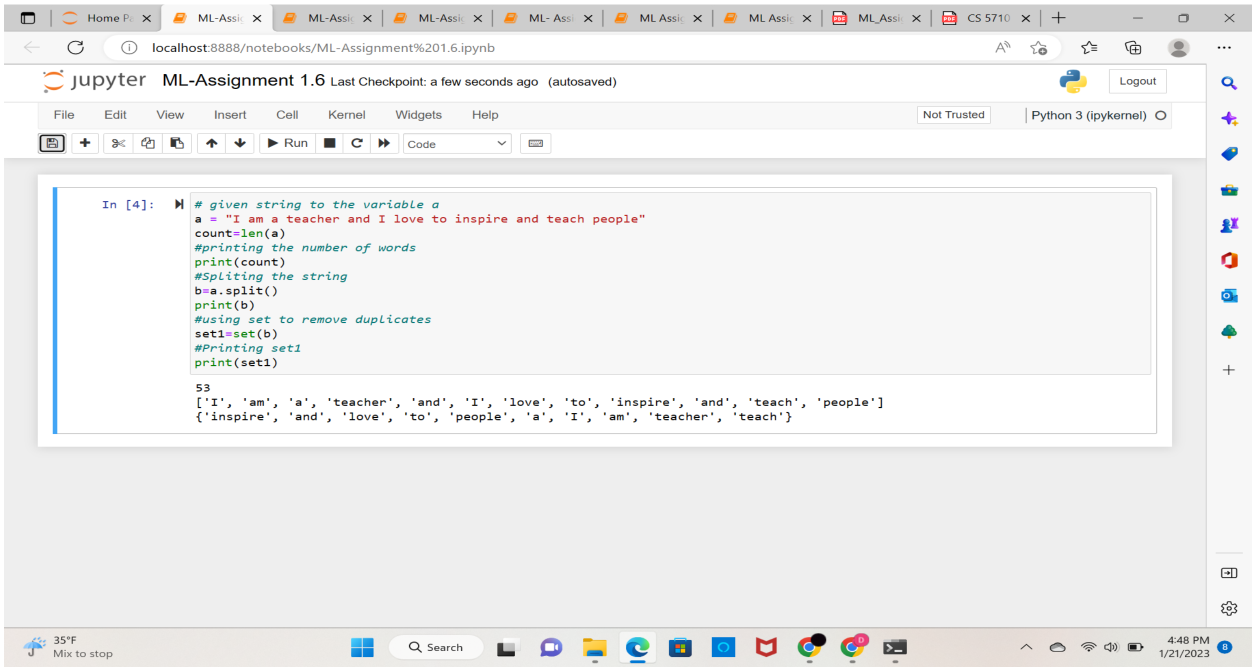

localhost:8888/notebooks/ML-Assignment%201.5.ipynb
jupyter ML-Assignment 1.5 Last Checkpoint: Yesterday at 4:32 PM (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)

```
In [1]:
#Given radius of circle 30 meters
r = 30
#assigned variable name "_area _of_circle"
_area_of_circle=3.14*(r**2)
#printing the stored value in variable
print("area of circle : ",_area_of_circle)
#circumference of circle
#assigned variable name "_circum_of_circle
_circum_of_circle=2*3.17*r
#printing the value
print("circumference of circle :",_circum_of_circle)
#area of circle for user input
#taking the user input
n=int(input())
#assigned variable name "area"
area=3.14*(n**2)
#printing the value
print("Area of circle with the user input :",area)
```

```
area of circle :  2826.0
circumference of circle : 190.2
4
Area of circle with the user input : 50.24
```

```
In [ ]:
```

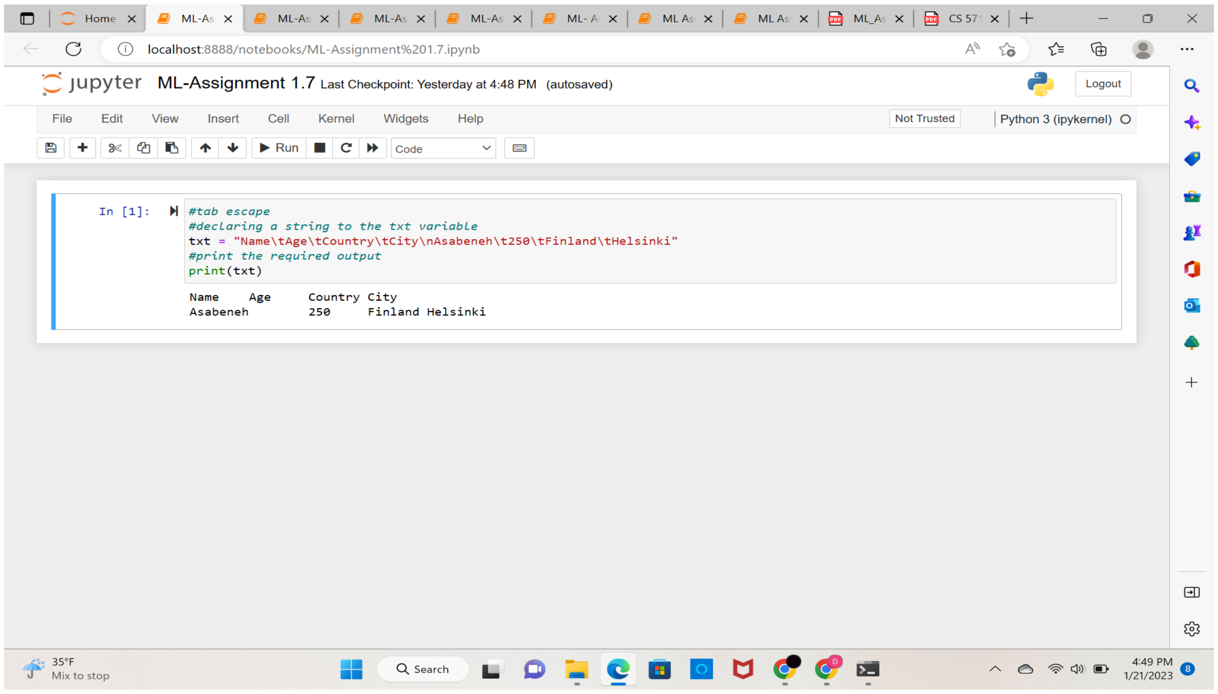
6. Assign the given sentence to variable a . Count the number of words in the variable a by len() and print the words by print(count). For splitting the words use .split() . For unique words delete the duplicates and print by print(set1).


localhost:8888/notebooks/ML-Assignment%201.6.ipynb
jupyter ML-Assignment 1.6 Last Checkpoint: a few seconds ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)

```
In [4]:
# given string to the variable a
a = "I am a teacher and I love to inspire and teach people"
count=len(a)
#printing the number of words
print(count)
#Spliting the string
b=a.split()
print(b)
#using set to remove duplicates
set1=set(b)
#Printing set1
print(set1)
```

```
53
['I', 'am', 'a', 'teacher', 'and', 'I', 'love', 'to', 'inspire', 'and', 'teach', 'people']
{'inspire', 'and', 'love', 'to', 'people', 'a', 'I', 'am', 'teacher', 'teach'}
```

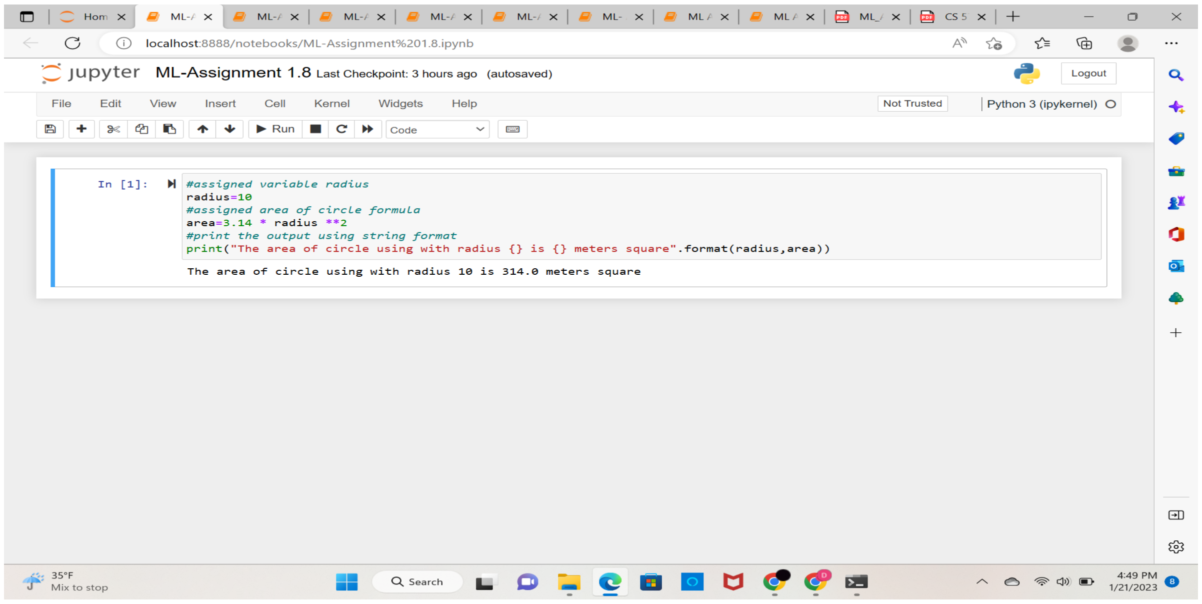
7. Declare a string to the txt variable and the print the txt statement.



```
In [1]:
#tab escape
#declaring a string to the txt variable
txt = "Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki"
#print the required output
print(txt)
```

```
Name    Age     Country City
Asabeneh        250     Finland Helsinki
```

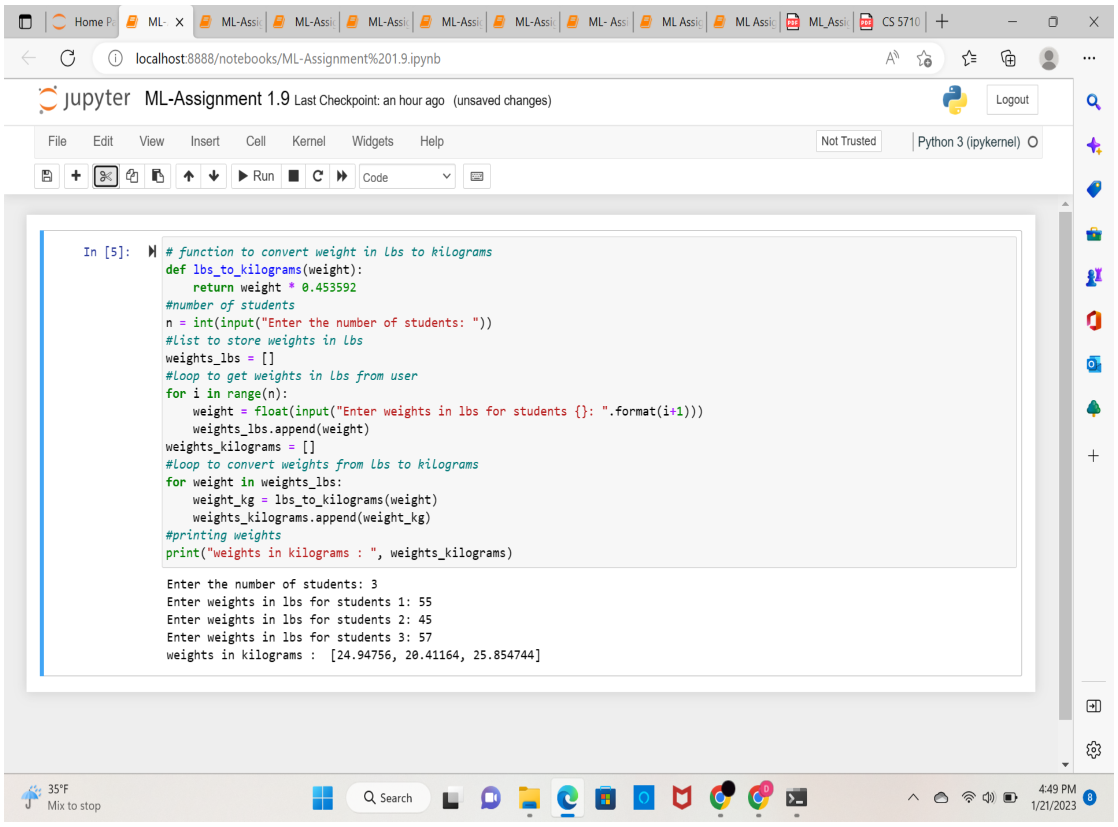
8. Given radius as 10 and area =3.14*radius**2 . Printing the statement : area of circle with radius 10 is 314 meters square by string formatting is done by .format(radius,area).



```
In [1]:
#assigned variable radius
radius=10
#assigned area of circle formula
area=3.14 * radius **2
#print the output using string format
print("The area of circle using with radius {} is {} meters square".format(radius,area))
```

```
The area of circle using with radius 10 is 314.0 meters square
```

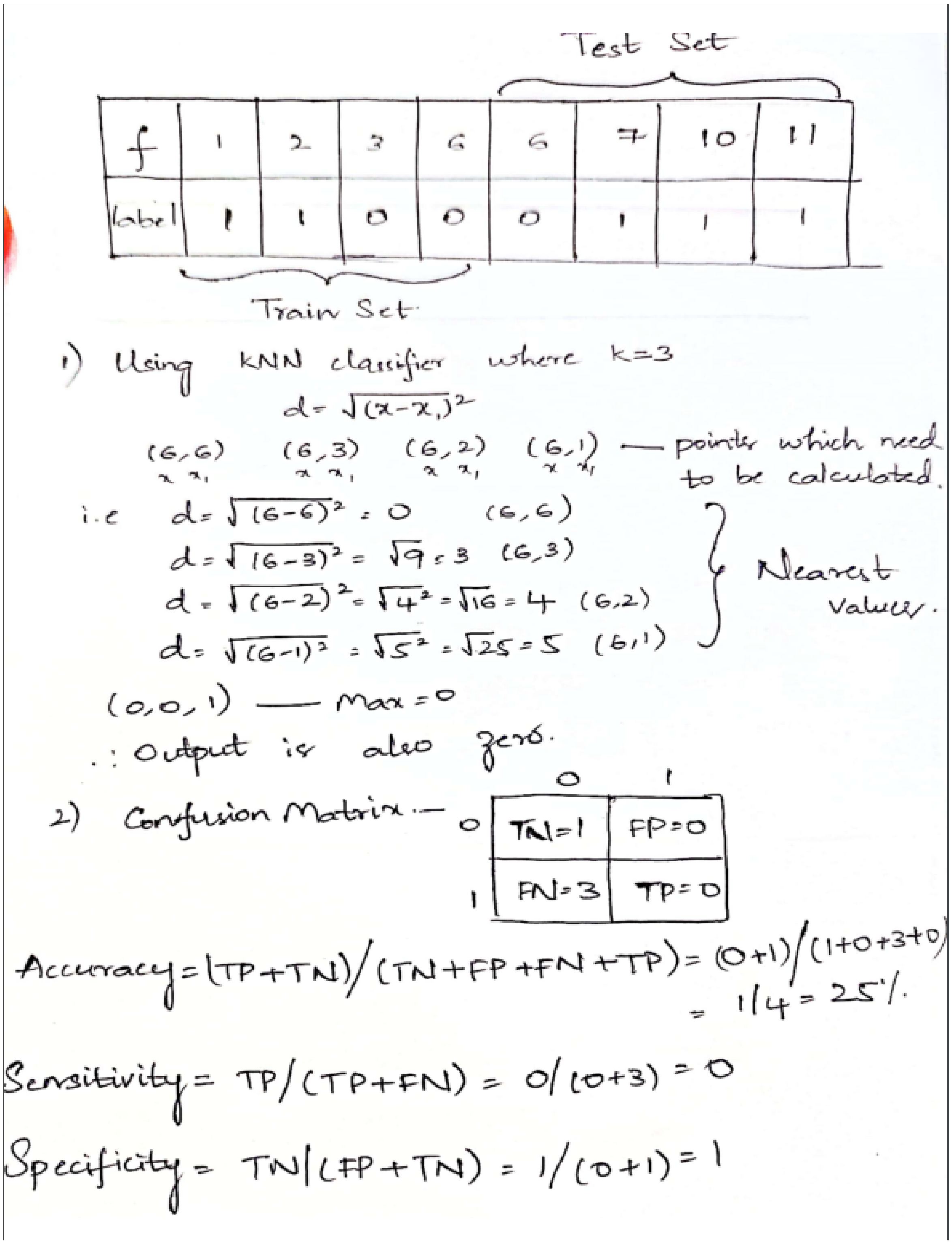
9. N is a string which indicates to "Enter the number of students:" . Create a list to store weight in lbs. Use for loop for taking weights from user and append to add weights.
For printing the weights in kilograms define lbs_to_kilograms(weight) and return the weight.



```
In [5]:  # function to convert weight in lbs to kilograms
         def lbs_to_kilograms(weight):
             return weight * 0.453592
         #number of students
         n = int(input("Enter the number of students: "))
         #list to store weights in lbs
         weights_lbs = []
         #loop to get weights in lbs from user
         for i in range(n):
             weight = float(input("Enter weights in lbs for students {}: ".format(i+1)))
             weights_lbs.append(weight)
         weights_kilograms = []
         #loop to convert weights from lbs to kilograms
         for weight in weights_lbs:
             weight_kg = lbs_to_kilograms(weight)
             weights_kilograms.append(weight_kg)
         #printing weights
         print("weights in kilograms : ", weights_kilograms)
```

```
Enter the number of students: 3
Enter weights in lbs for students 1: 55
Enter weights in lbs for students 2: 45
Enter weights in lbs for students 3: 57
weights in kilograms :  [24.94756, 20.41164, 25.854744]
```

10.



Test Set

f	1	2	3	6	6	7	10	11
label	1	1	0	0	0	1	1	1

Train Set

1) Using KNN classifier where k=3

$$d = \sqrt{(x - x_1)^2}$$

$\underset{x\ x_1}{(6,6)}$ $\underset{x\ x_1}{(6,3)}$ $\underset{x\ x_1}{(6,2)}$ $\underset{x\ x_1}{(6,1)}$ — points which need to be calculated.

i.e
$d = \sqrt{(6-6)^2} = 0$ (6,6)
$d = \sqrt{(6-3)^2} = \sqrt{9} = 3$ (6,3)
$d = \sqrt{(6-2)^2} = \sqrt{4^2} = \sqrt{16} = 4$ (6,2)
$d = \sqrt{(6-1)^2} = \sqrt{5^2} = \sqrt{25} = 5$ (6,1)
} Nearest values.

(0,0,1) — Max = 0

∴ Output is also zero.

2) Confusion Matrix —

$$\begin{array}{c|c|c} & 0 & 1 \\ \hline 0 & TN=1 & FP=0 \\ \hline 1 & FN=3 & TP=0 \end{array}$$

Accuracy = (TP+TN)/(TN+FP+FN+TP) = (0+1)/(1+0+3+0)
= 1/4 = 25%.

Sensitivity = TP/(TP+FN) = 0/(0+3) = 0

Specificity = TN/(FP+TN) = 1/(0+1) = 1