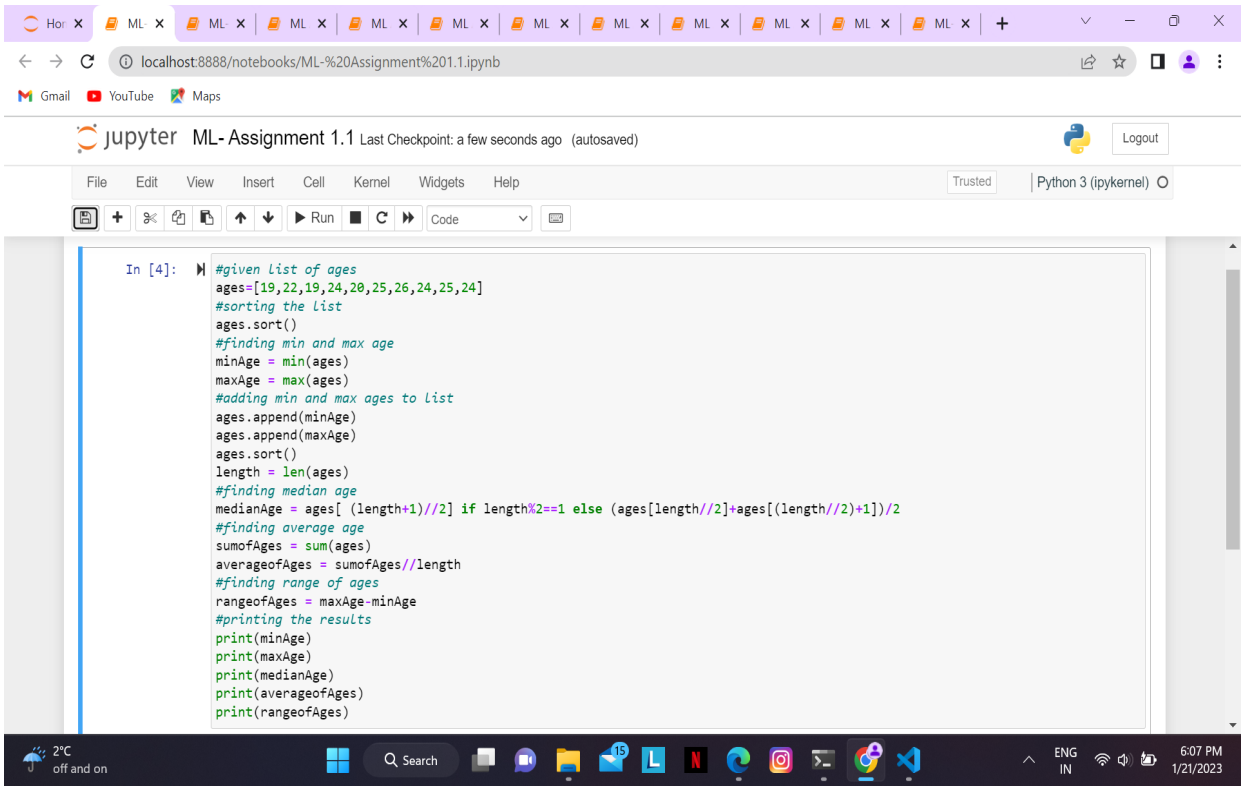
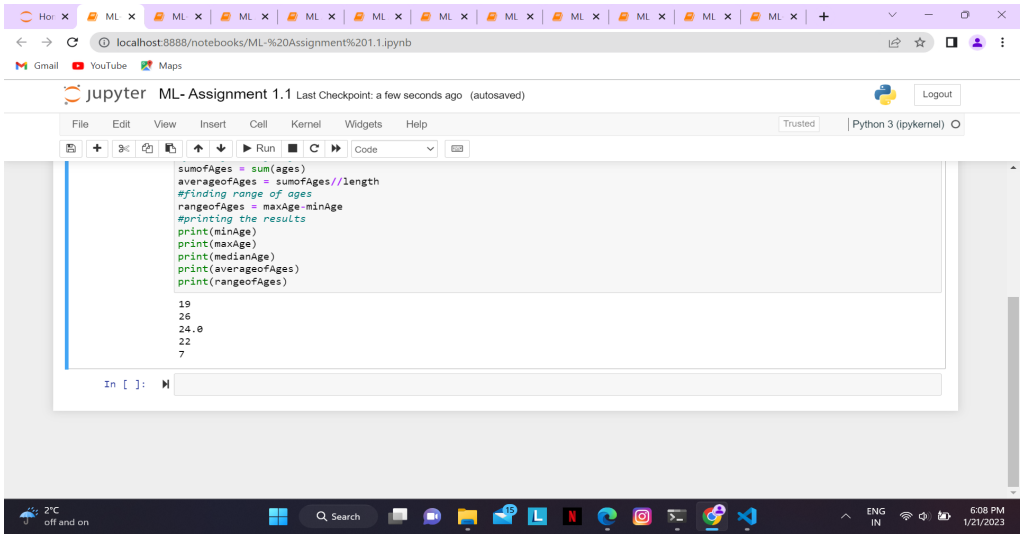
MACHINE LEARNING ASSIGNMENT - 1

NAME : SUPRIYA SAMA 700744510

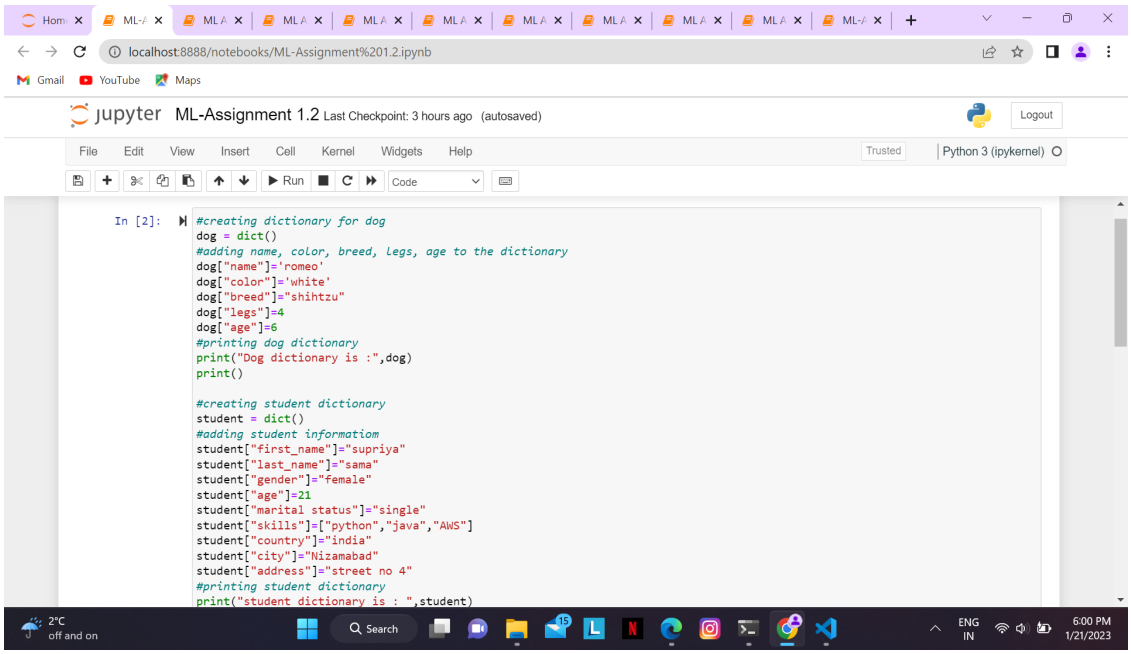
Video link - https://drive.google.com/file/d/1JOBymeNVUTv6n-HmtKmjPVoFPHBABT7o/view?usp=share_link

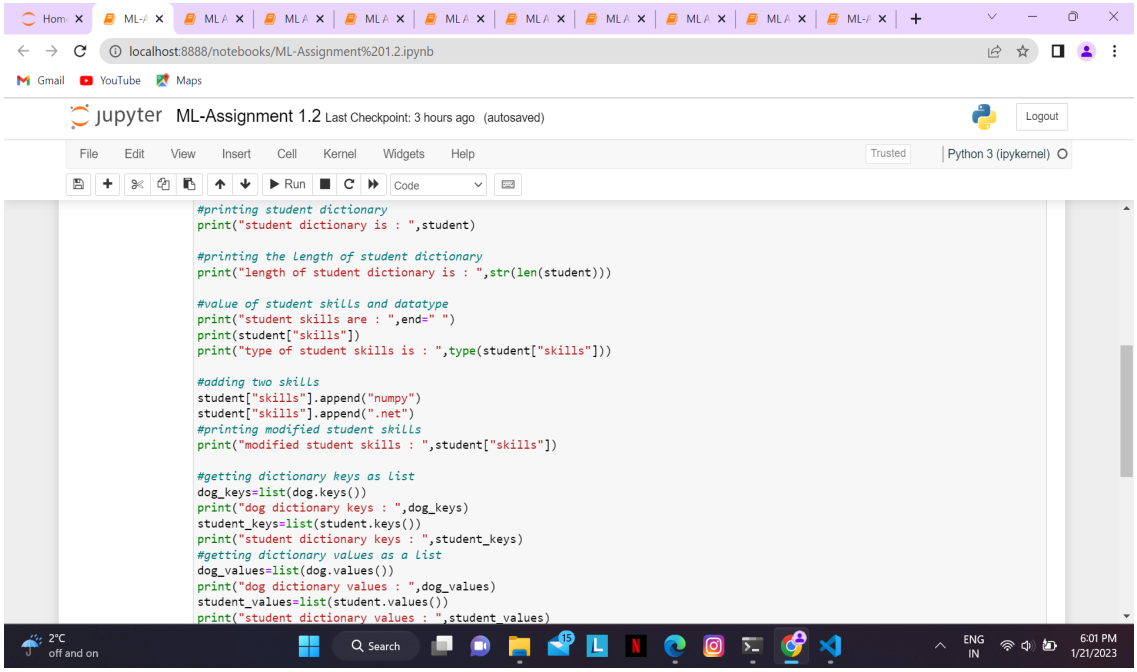
1. For the given list sorting can be done by using sort( ) function in ascending order by default. min( ), max( ), append( ) functions can be used to find minimum, maximum and to add the ages to the list. Median can be find by using ages[(length+1)//2]. Average of the ages can be calculated by dividing sum of all ages by the length. Range can be calculated by subtracting minimum age from maximum age.

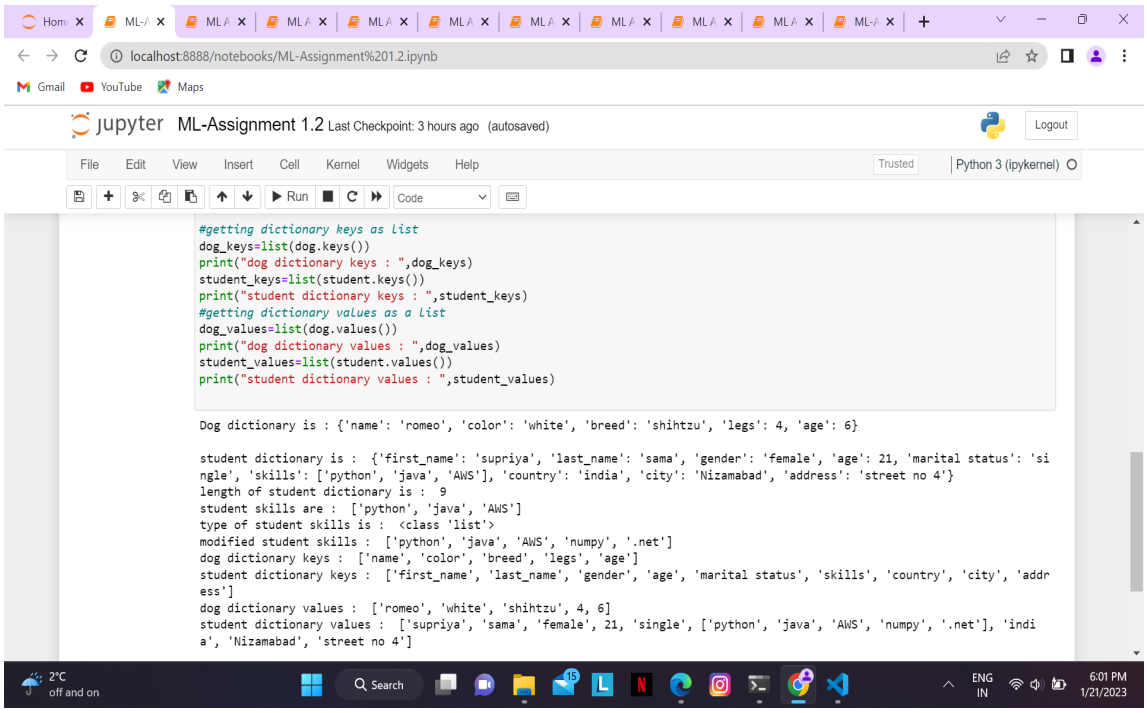




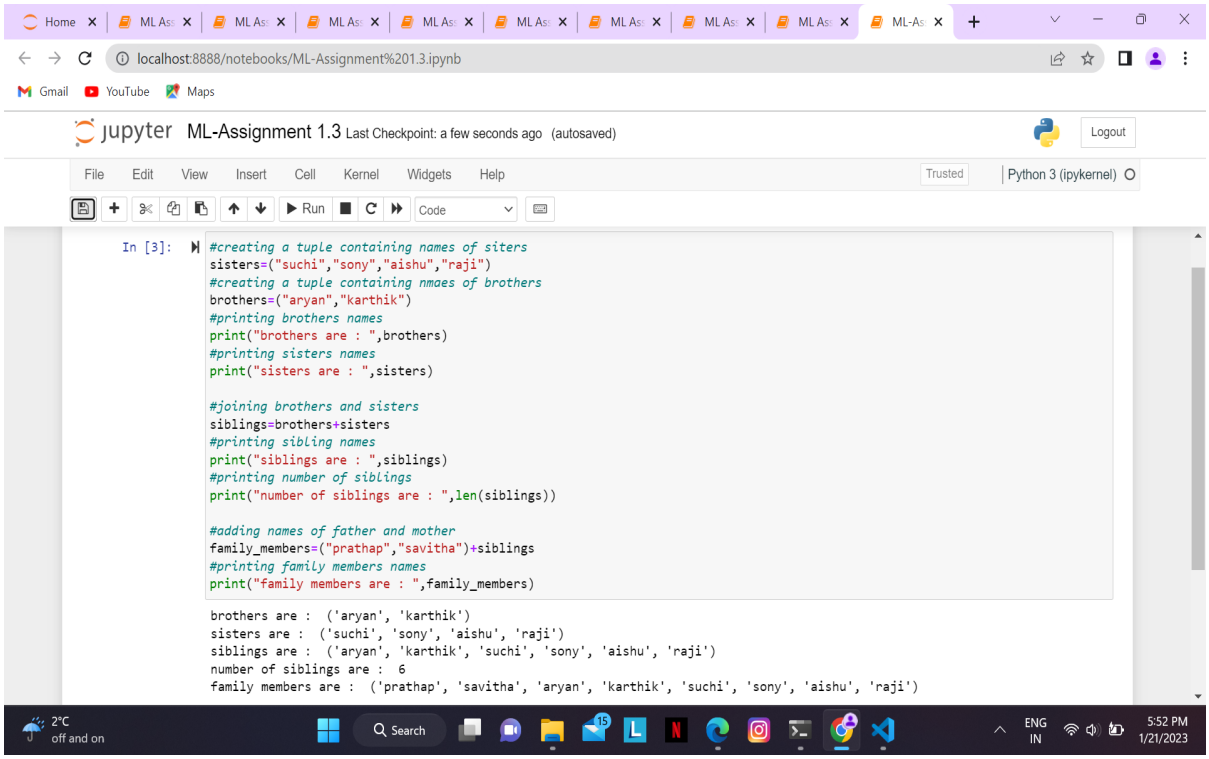
2. Create dog dictionary using dict( ) and add the given features to the dictionary and print those values. Create student dictionary by using dict( ) and add given details to the dictionary and print the results. Length can be calculated by using len(student). Data type can be checked using type( ). Add the skills by using append( ) function. To get dictionary keys as list use list(dog.keys( )) and use list(dog.values( )) for dictionary values.



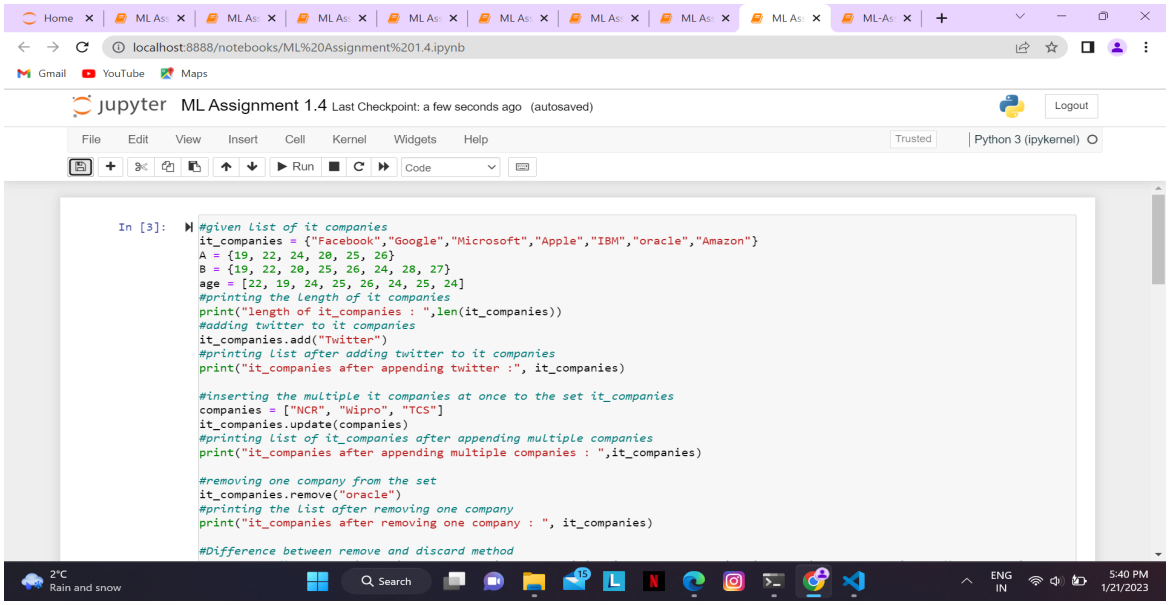


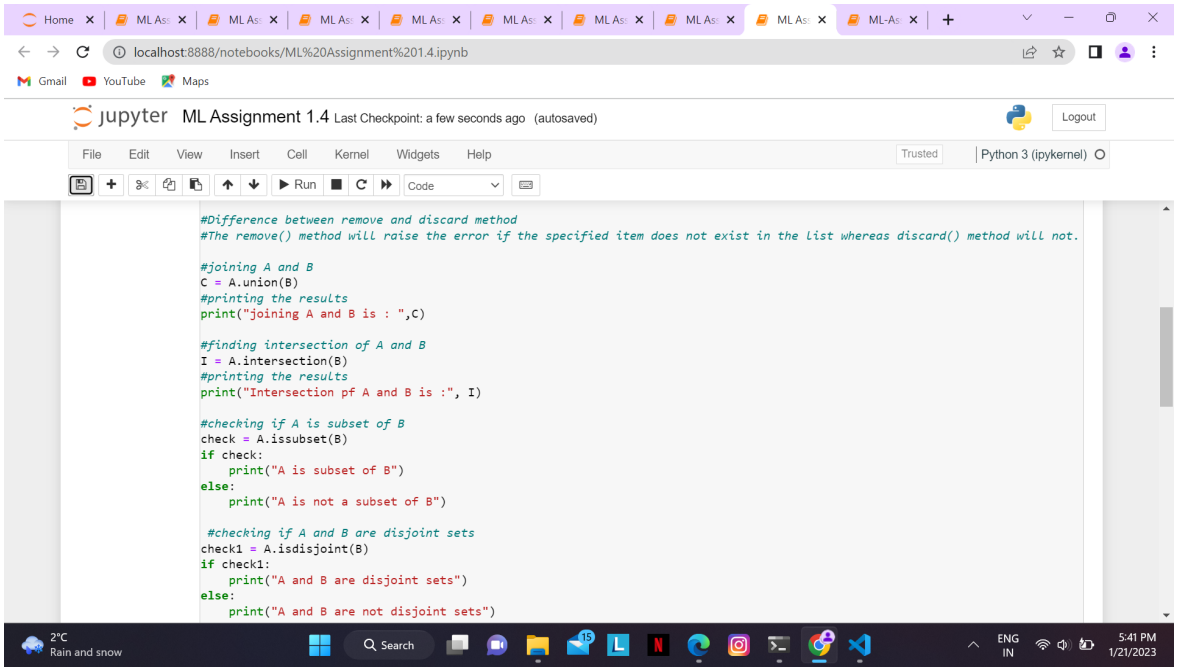


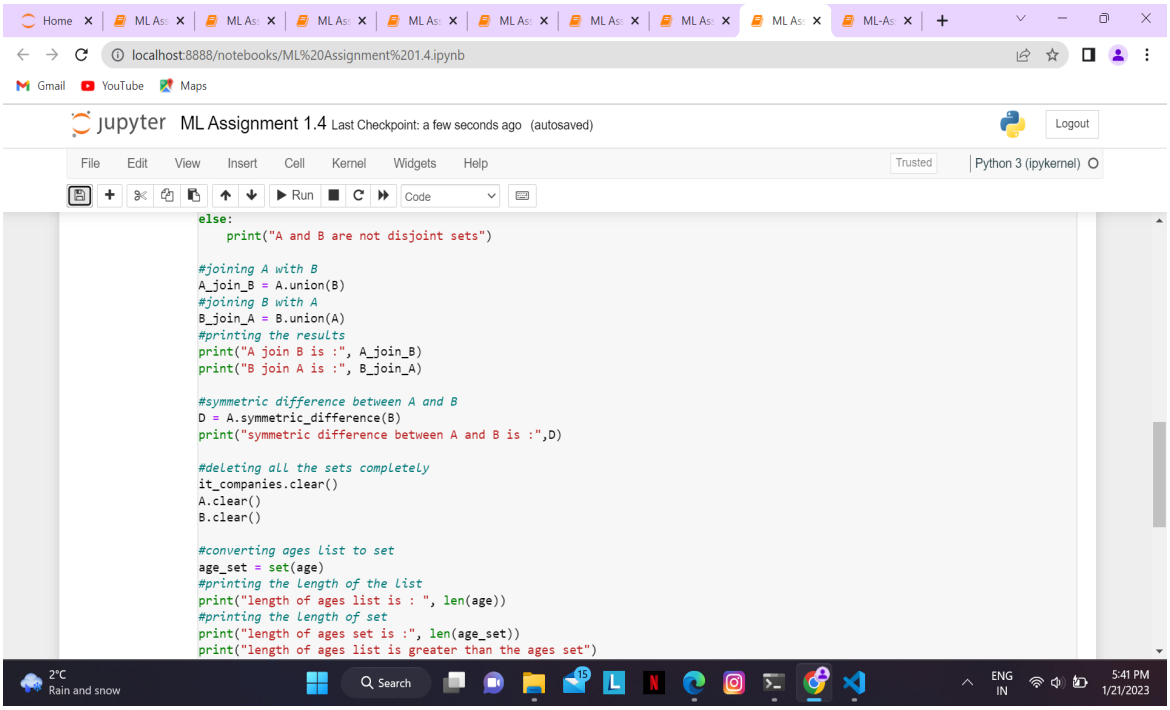
3. Create a separate tuples for brothers and sisters containing names. Join both the tuples and assign it to siblings. Length can be find using len( ) function. Add father and mother names to the siblings by using '+' and assign the result to the family_members.

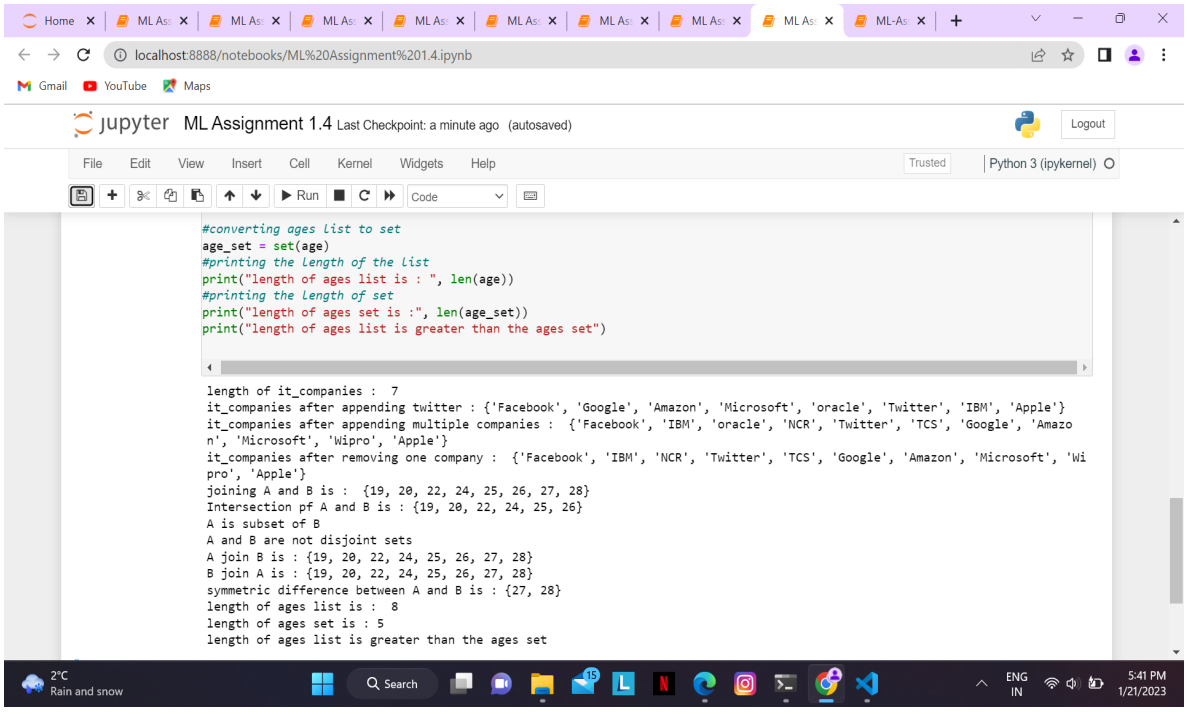


4. Length of it_companies can be find using len( ) function. Add 'Twitter' to the set by using it_companies.add("Twitter"). We can insert and remove companies from the set by using update(companies) and remove(companies). The remove( ) method will raise an error if the specified item does not exist whereas discard( ) method does not. We use union, intersection operators to join and to find intersections. We check if the A and B are subset or disjoint sets by using A.issubset(B) and A.isdisjoint(B). We join the sets using union operator. To delete the sets we use clear( ) method. Convert list to set by using age_set = set(age).

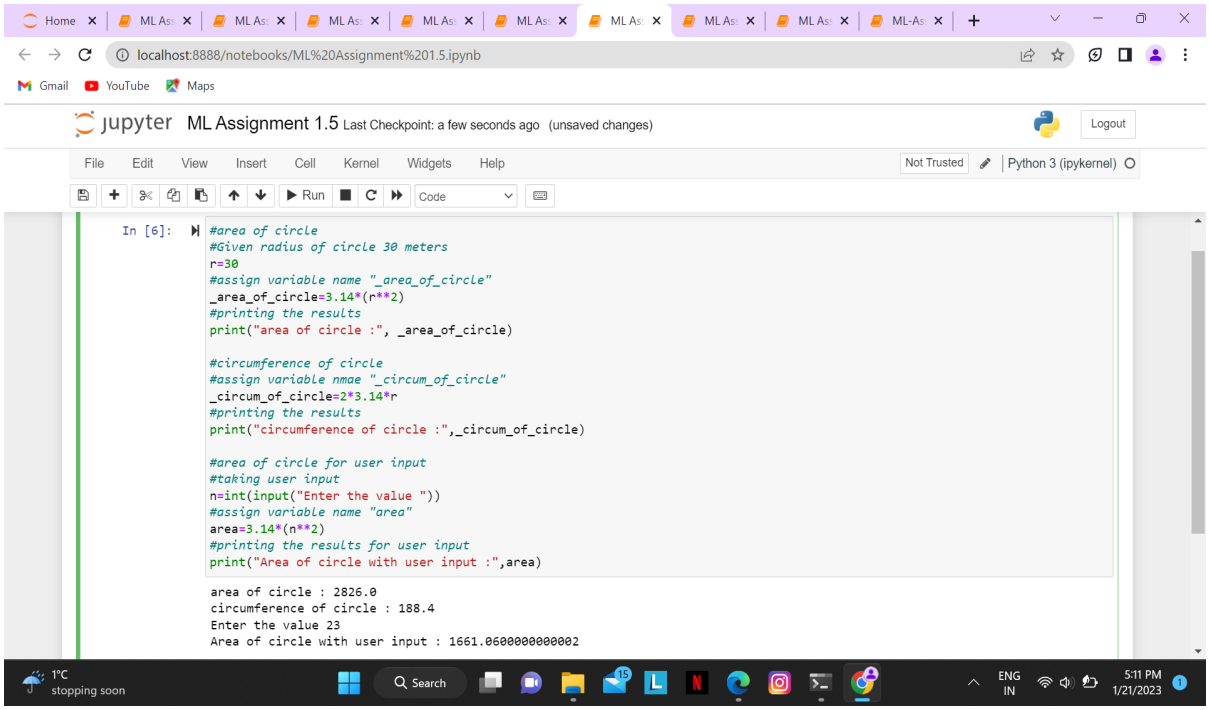

Home
localhost:8888/notebooks/ML%20Assignment%201.4.ipynb
Gmail
YouTube
Maps
jupyter ML Assignment 1.4 Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [3]:
#given list of it companies
it_companies = {"Facebook","Google","Microsoft","Apple","IBM","oracle","Amazon"}
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")
#printing list after adding twitter to it companies
print("it_companies after appending twitter :", it_companies)
#inserting the multiple it companies at once to the set it_companies
companies = ["NCR", "Wipro", "TCS"]
it_companies.update(companies)
#printing list of it_companies after appending multiple companies
print("it_companies after appending multiple companies : ",it_companies)
#removing one company from the set
it_companies.remove("oracle")
#printing the list after removing one company
print("it_companies after removing one company : ", it_companies)
#Difference between remove and discard method
2°C
Rain and snow
Search
ENG
IN
5:40 PM
1/21/2023


Home
localhost:8888/notebooks/ML%20Assignment%201.4.ipynb
Gmail
YouTube
Maps
jupyter ML Assignment 1.4 Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
#Difference between remove and discard method
#The remove() method will raise the error if the specified item does not exist in the list whereas discard() method will not.
#joining A and B
C = A.union(B)
#printing the results
print("joining A and B is : ",C)
#finding intersection of A and B
I = A.intersection(B)
#printing the results
print("Intersection pf A and B is :", I)
#checking if A is subset of B
check = A.issubset(B)
if check:
print("A is subset of B")
else:
print("A is not a subset of B")
#checking if 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")
2°C
Rain and snow
Search
ENG
IN
5:41 PM
1/21/2023

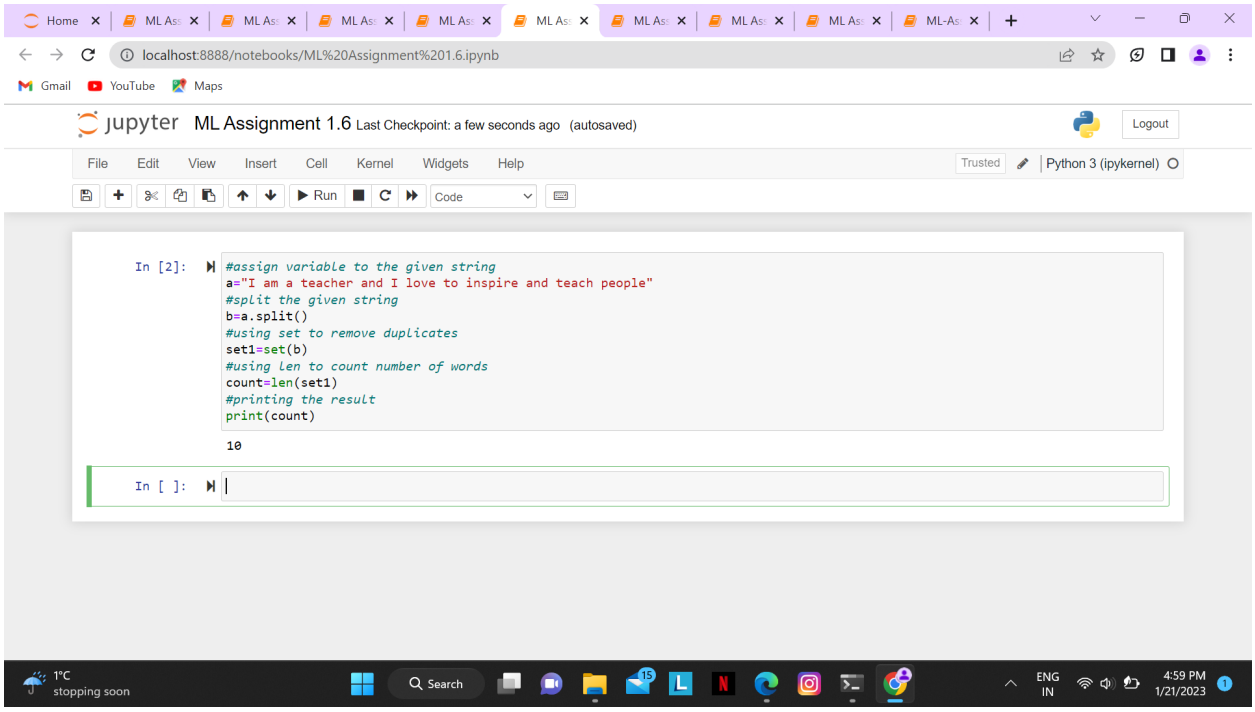




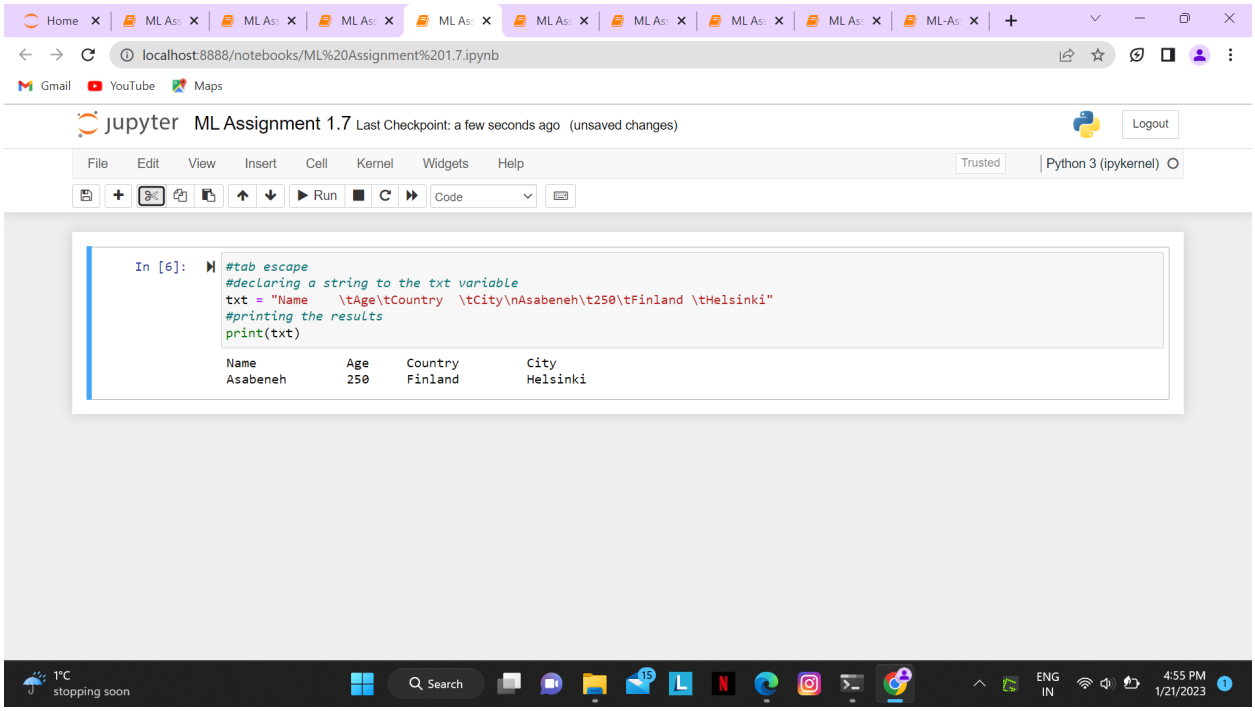
5. Area of circle can be calculated using 3.14*(r**2) and assign this value to variable _area_of_circle_. Circumference can be calculated by using 2*3.14*r and assign this value to variable _circum_of_circle_. We can calculate the area of circle for user input by using 3.14*(n**2).



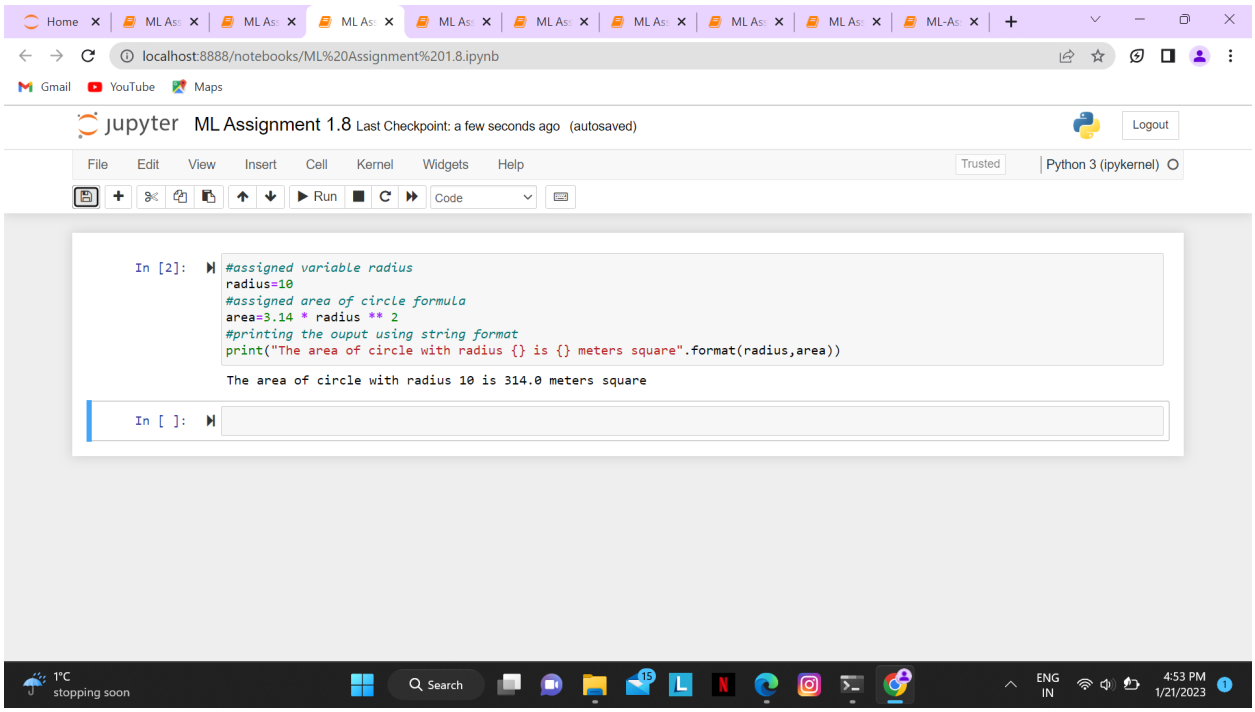
6. Assign given string to the variable ‘a’ and split the given string using a.split( ). Use set to remove duplicates. Count the number of words using len( ) function and print the values.



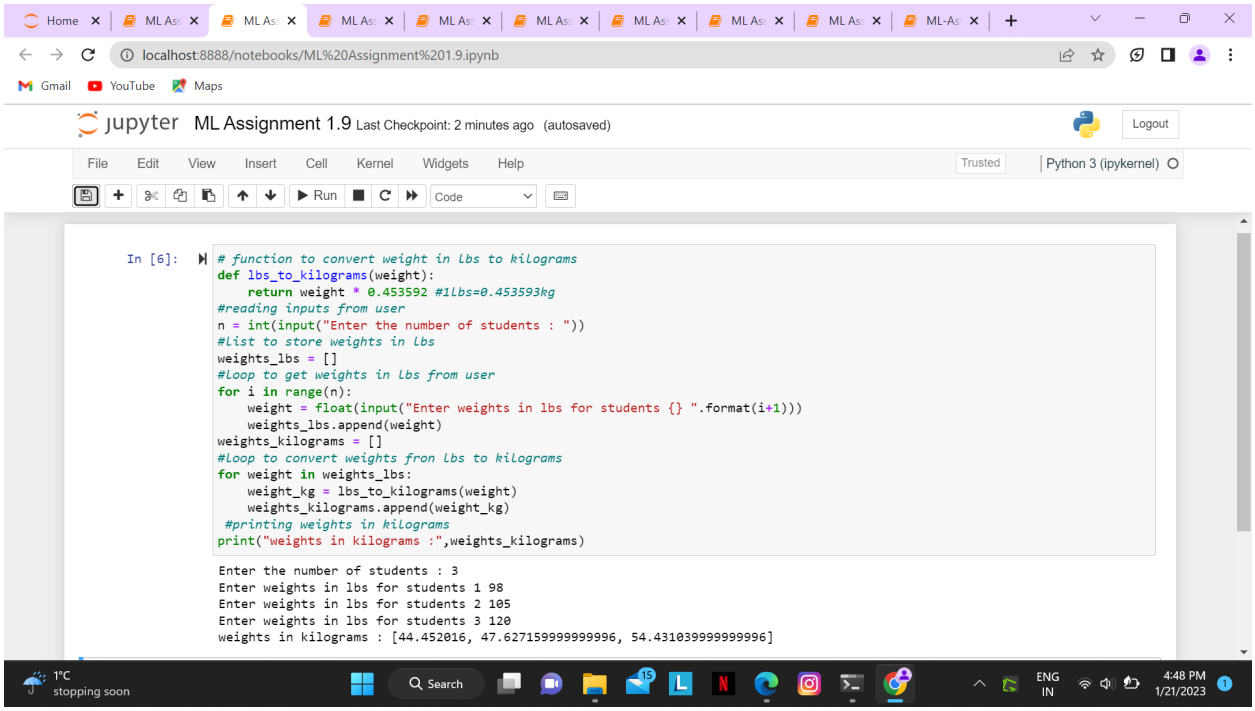
7. Assign given string to the variable ‘txt’. Use ‘/’ to give space between the string. Print the output.



8. Assign given value to the variable ‘radius’. We can find the area of the circle by using 3.14*radius**2. Print the output using format( ).



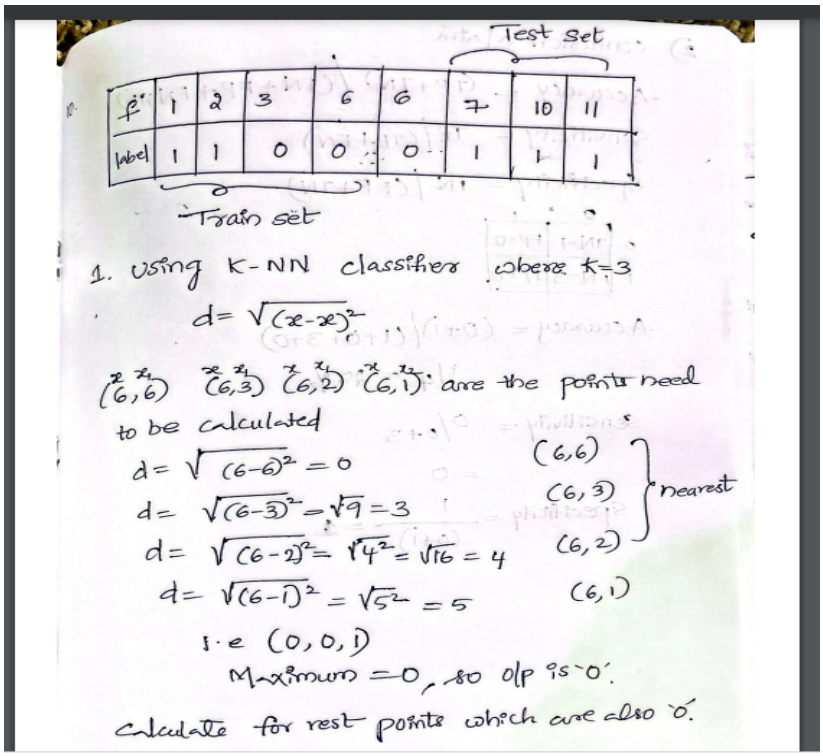
9. Assign input to the variable ‘n’. Use for loop for appending the elements into the list. Convert weights from lbs to kilograms using for loop.



```
In [6]: # function to convert weight in lbs to kilograms
def lbs_to_kilograms(weight):
    return weight * 0.453592 #1lbs=0.453593kg
#reading inputs from user
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 fron lbs to kilograms
for weight in weights_lbs:
    weight_kg = lbs_to_kilograms(weight)
    weights_kilograms.append(weight_kg)
 #printing weights in kilograms
print("weights in kilograms :",weights_kilograms)

Enter the number of students : 3
Enter weights in lbs for students 1 98
Enter weights in lbs for students 2 105
Enter weights in lbs for students 3 120
weights in kilograms : [44.452016, 47.627159999999996, 54.431039999999996]
```

10.



2) Confusion Matrix

Accuracy = (TP+TN) / (TN+FP+FN+TP)

Sensitivity = TP / (TP+FN)

Specificity = TN / (FP+TN)

	0	1
0	TN=1	FP=0
1	FN=3	TP=0

Accuracy = (0+1) / (1+0+3+0)

1/4 = 25%

Sensitivity = 0/0+3

= 0

Specificity = $\frac{1}{(0+1)}$ = 1