

**ACADEMIC INTEGRITY**

Academic Services

**ASSIGNMENT DECLARATION FORM**

**Plagiarism** means the reproduction of someone else’s words, ideas or findings and presenting them as your own ideas without proper acknowledgement and includes:

* direct copying or paraphrasing from someone else’s published work (either electronic or hard copy) without acknowledging the source;
* using facts, information and ideas derived from a source without acknowledgement;
* producing assignments (required to be independent) in collaboration with and/or using the work of other people; and
* assisting another person to commit an act of plagiarism.

**By submitting\* this piece of work and signing this document, I declare that:**

* the work is my own work or the work of the group; and
* I have not previously submitted all or part of this work for assessment in any subject, unless the subject coordinator for the current subject (or my research supervisor, if applicable) has given me written permission to reuse specific material and I have correctly referenced the material taken from my own earlier work; and
* I have read the plagiarism avoidance advice issued to me by my Faculty; and
* I have read and agree to be bound by the Statutes, Regulations and Policies of the University relating to Academic Misconduct – see [http://www.latrobe.edu.au/learning/integrity.html;](http://www.latrobe.edu.au/learning/integrity.html) and
* if I am discovered committing or having committed an act of academic misconduct (including plagiarism or cheating), I will be subject to student discipline processes.

I also agree that the University or any third party authorised by the University has the right to reproduce and/or communicate (make available online or electronically transmit) the work I have submitted, for the purpose of detecting work that has been copied from other sources.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subject Code** | C | S | E | 5 | D | M | I |  |  |
| **Subject Name** | DATA MINING | | | | | | | | |
| **Lecturer’s name** | Professor Phoebe Chen | | | | | | | | |
| **Name of Document** | CSE5DMI Assignment One | | | | | | | | |
| **Student Number** | 1 | 9 | 8 | 5 | 0 | 5 | 5 | 7 |  |
| **Family Name** | TURLAPATI | | | | | | | | |
| **Given Names** | VASISTA | | | | | | | | |
| **Signature of Student (Digital signature is allowed)** | T.Vasista | | | | | | | | |
| **Date of Signature** | 11-09-20218 | | | | | | | | |

Assignment answers

Q1) a) ans)

1. s=input()

Explanation:-Input is a built in module in python which takes the input directly from console and I am storing the value in variable names ‘s’

1. for i in range(0,len(s)):

Explanation:- Starting a for loop which iterates from 0 to the length of the input string

1. print(ord(s[i]))

Explanation:- In the for loop itself I’m printing

OUTPUT:-

hello

104

101

108

108

111

Q1) b) ans)

1. s=input()

Explanation:- Input is a built in module in python which takes the input directly from console and I am storing the value in variable names ‘s’

1. sum1=0;

Explanation:-Initializing a variable named sum1 to 0 which will be used below

1. for i in range( 0,len(s)):

Explanation:-Starting for loop which iterates from 0 to the length of the input string

1. sum1=sum1+ord(s[i])

Explanation:-every time the loop iterates im adding the ASCII value of each character taken from the input string to variable sum1

1. print(sum1)

Explanation:-printing the value of sum1 outside the loop

OUTPUT:-

hello

532

Q1) c) ans)

1. s=input()

Explanation:- Input is a built in module in python which takes the input directly from console and I am storing the value in variable names ‘s’

1. lis=[];

Explanation:-creating an empty list named ‘lis’

1. temp="";

Explanation:-creating a temporary string variable ‘temp’ which is empty

1. for i in range( 0,len(s)):

Explanation:-Starting for loop which iterates from 0 to the length of the input string

1. temp=ord(s[i])

Explanation:- As the loop iterates the ASCII value of each character in the input string is stored in temp

1. lis.append(temp)

Explanation:-As the loop iterates the ASCII value of each character in the input string which is stored in temp variable

1. print(lis)

Explanation:- printing the entire list

OUTPUT:-

hello

[104, 101, 108, 108, 111]

2)a)ans)

The given program is

X= 'HELLO'

def fun():

X='NI'

def nested():

print(X)

nested()

fun()

print(X)

Here the OUTPUT will be HELLO

Reason:- Start with how the machine thinks . Frist it will create a variable X which has value ‘HELLO’.

Now as the program runs further more just focous on the facts the x in the fun() function is limited to that function only that is it is a local variable of function Fun. When the function is called before the last line, the value of X in the fun function is initialized to ‘NI’, but its scope is limited to that function only and that function itself. So the X value which is initialized in the first line doesn’t change.

3) a)ans)

import pandas as p

Explanation:- importing the pandas module as variable p , which helps in using pandas function easy

csv\_path=("C:\\Users\\acer\\Desktop\\friends\\assg1\_room.csv")

Explanation:- creating a variable csv\_path which has the location of the csv file named “assg1\_room.csv”

data=p.read\_csv(csv\_path)

Explanation:- reading the data in the specified path into a variable ‘data’ using the pandas function read\_csv

print(data)

Explanation:-

Printing the data to check if the data imported is the required one

te=data['Tenants']

Explanation:- storing the value in column Tenants of data to a temporary variable

te1=data['Furnished']

Explanation:- storing the value in column Furnished of data to a temporary variable

mean=te.mean()

Explanation:-calicuating the mean of the values in the variable te using inbuilt function mean() and storing the value in a variable mean

te.where(p.notna(te),mean,inplace=True)

Explanation:-the function where traverses the data in te and applies the given condition which is to fill the places were the values in te has nothing in it , that is the values which are missing with mean

te1.where(p.notna(te1),'N/A',inplace=True)

Explanation:-the function where traverses the data in te1 and applies the given condition which is to fill the places were the values in te1 has nothing in it , that is the values which are missing with N/A

print(te.isnull().values.any())

Explanation:- printing to check if there is any missing values in the variable te

data['Tenants']=te

Explanation:-Transferring the changed te values in to the original data

print(data['Tenants'].isnull().values.any())

Explanation:- printing to re check if there is any missing values in the variable te

datae1=p.DataFrame(data)

Explanation:-creating a data frame datae1 which has the values of data so that it is easy to export in to a other file

exportFilePath=("C:\\Users\\acer\\Desktop\\friends\\export.csv")

Explanation:-storing the path of the file where the data need to be exported to

with open(exportFilePath,"w") as output:

datae1.to\_csv(output,header=True,sep=",")

Explanation:-using the function with open() we are writing the data to the output file

OUTPUT:-

False

False

3)ans)b)

import numpy as n

import pandas as p

Explanation:- importing the pandas module as variable p and numpy as n, which helps in using pandas and numpy function easy

from Orange.data import Table,Domain

from Orange.evaluation import CrossValidation,scoring

from Orange.classification import SklTreeLearner

Explanation:- importing the table,domain modules form Orange.data,cross validation and scoring modules from Orange.evaluation and SklTreeLearner from Orange.classification

td=Table.from\_file("C:\\Users\\acer\\Desktop\\friends\\export.csv")

Explanation:-Importing the file export.csv which is the file we stored the output data into in 3a) into variable td

feature\_vars=list(td.domain.variables[1:])

Explanation:-storing the domain variable names in a list called feature\_vars

class\_label\_var=td.domain.variables[7]

Explanation:-storing the domain variable name of 7th column which is the class variable in

class\_label\_var

md=Domain(feature\_vars,class\_label\_var)

Explanation:-creating a domain using the Domain function and the data from Feature\_vars and class\_label\_var into variable md

td=Table.from\_table(domain=md,source=td)

Explanation:- inserting the domain to the data

n1=td.approx\_len()

Explanation:- Calculating the length if the files

n2=n1\*80/100

Explanation:-knowing the value of 80%if the data length and storing ginto n2

train\_data\_set=td[:n2]

Explanation:-creating the train data

test\_data\_set=td[n2:]

Explanation:-creating the test data

tree\_learner=SklTreeLearner()

Explanation:-creating a variable which has the SklTreeLearner properties

decision\_tree=tree\_learner(train\_data\_set)

Explanation:- creating the decision tree using the tree\_learner varible

results=CrossValidation(td,[tree\_learner],k=10)

Explanation:-cross validating it 10 times

print(decision\_tree(test\_data\_set))

Explanation:-printing the decision tree

print("Accuracy",scoring.CA(results)[0])

Explanation:-Printing the accurcy

print("AUC",scoring.AUC(results)[0])

Explanation:-printing the area under curvev

OUTPUT :-

[2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.

2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.

3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.

3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.

3. 3. 3. 3.]

Accuracy 1.0