Artificial Intelligence 101 : Assignment 1

```
Student Name:  Armielyn C. Obinguar
Instructor: Jefferson Costales
```

DATAFRAMES

In [2]:

```
# program starts from here
# import pandas library
#import numpy library
import pandas as pd
import numpy as np
```

Question 1 ¶

In [5]:

```
#Set the array or value per column
array = np.array([["a",2,1,3,3.0,"h","2020-01-01 00:00:00-05:00","2020-01-01

#INDEX gets a value at a given location in a range of cells based on numeric
index_values = [0,1,2,3]

# Column Names
column_values = ["column-a","column-b","column-c","column-d","column-e","col

Armielyn = pd.DataFrame(data = array, index = index_values,columns = column_

print(Armielyn)
```

```
  column-a column-b column-c column-d column-e column-f  \
0        a        2        1        3      3.0        h
1        b        5        2        4      4.0        i
2        c        8        3        5      5.0        j
3        d       11        4        6      6.0        k

                   column-g                       column-h column-i
0  2020-01-01 00:00:00-05:00  2020-01-01 00:00:00.000000000     True
1  2020-01-02 00:00:00-05:00  2020-01-01 00:00:00.000000001    False
2  2020-01-03 00:00:00-05:00  2020-01-01 00:00:00.000000002    False
3  2020-01-04 00:00:00-05:00  2020-01-01 00:00:00.000000003     True
```

In [6]:

```
Armielyn
```

Out[6]:

	column-a	column-b	column-c	column-d	column-e	column-f	column-g	column-h	column-i
0	a	2	1	3	3.0	h	2020-01-01 00:00:00-05:00	2020-01-01 00:00:00.000000000	True
1	b	5	2	4	4.0	i	2020-01-02 00:00:00-05:00	2020-01-01 00:00:00.000000001	False
2	c	8	3	5	5.0	j	2020-01-03 00:00:00-05:00	2020-01-01 00:00:00.000000002	False
3	d	11	4	6	6.0	k	2020-01-04 00:00:00-05:00	2020-01-01 00:00:00.000000003	True

Question 2: Data for this question can be found from "tweets" sheet in assignment-data.xlsx.“ABC Company” has collected “tweets” from tweet.com and instructed its junior data scientist “Mr. Jo Jo” to mask sensitive data so that they can use the masked data for testing.

Task-1: Mr.Jo Jo likes to do his experiment on small amount of data thus decided to play with only 10 rows.Read only the rows 3-12 from tweets sheet and name it as "df" and display the type of df. The output should be shown as follows

In [7]:

```
# read_csv file and store in df
Armielyn1 = pd.read_csv("Tweets.csv", skiprows=[1, 2], nrows=10)
```

In [8]:
```python
# call the dataframe
Armielyn1
```

Out[8]:
```
              tweet_id          created_at                                                tweet
0   832516558903730176   2017-02-17 9:06:29         @comark yes check the flux capacitor in our lo...
1   832293187670704128  2017-02-16 18:18:53        RT @iafrikan: .@88mph_Africa stopped running i...
2   831880802489217024  2017-02-15 15:00:13           88mph invest in Ahoy - a business travel app f...
3   946063916068634631  2017-12-27 17:03:09     Ce samedi 30 décembre 2017 à @ActivSpaces, se ...
4   945973955847999488  2017-12-27 11:05:41        RT @OIFfrancophonie: Retour en vidéo sur la vi...
5   945972004548726785  2017-12-27 10:57:55       RT @OIFfrancophonie: L’OIF a organisé un ateli...
6   945606558930690048  2017-12-26 10:45:46             Plus que deux jours et les inscriptions seront...
7   945595453256687616  2017-12-26 10:01:39       RT @nlend_nyounai: Transform your idea into a ...
8   945595401729724416  2017-12-26 10:01:26          RT @ElongWilliam: Si tu as une idée et tu es u...
9   944304498461347840  2017-12-22 20:31:51   RT @chantaledie: AWESOME https://t.co/pTu3k5LA81
```

In [9]:
```python
# display the type of df
type(Armielyn1)
```

Out[9]:
```
pandas.core.frame.DataFrame
```

For this task we use use Python's Pandas library

At Pandas's read_csv method, pass following attributes as parameters -

file_name = this is the name of your csv file. In this question, name of csv file is tweets_sheet.csv and it store at same location where program code file is stored.

This CSV file has header name of all columns in row 1 (or index 0) skip_rows = this parameter takes list as value. This list contains index of rows which we do not want to read in our dataframe. In this question we start read this file from index 3, so skip only index 1, 2. Do not skip index 0, because it contains header information.

nrows = This parameter takes an integer as value. It decided number of rows to be read from file and store in df. In this question 10 rows needed to be read, so pass value 10 to this parameter. So finally, read_csv method looks like this - df = pandas.read_csv("tweets_sheet.csv", skiprows=[1,2], nrows=10)

Question 2

Task-2: Mr.Jo Jo found that “tweetid, created-at and username” columns are sensitive thus decided to mask the values from those columns. He created new columns “new-tweet-id”,”created-at4” to store the masked “tweet-ids” and masked “created-at” values. He decided to use “username” column to store masked usernames.

In [10]:

```python
import pandas as pd
Armielyn11 = pd.DataFrame({"tweet_id":[3017, 412, 1702, 3463, 9948, 2678, 90
                           "created_at":['Wednesday/February/16', 'Tuesday/February/
                           "username":['yyy', 'yyy', 'yyy', 'yyy', 'yyy', 'yyy', 'yy
Armielyn11
```

Out[10]:

	tweet_id	created_at	username
0	3017	Wednesday/February/16	yyy
1	412	Tuesday/February/16	yyy
2	1702	Monday/February/16	yyy
3	3463	Tuesday/December/16	yyy
4	9948	Tuesday/December/16	yyy
5	2678	Tuesday/December/16	yyy
6	9004	Monday/December/16	yyy
7	8761	Monday/December/16	yyy
8	2441	Monday/December/16	yyy
9	4784	Thursday/December/16	yyy

In [11]:

```python
new_tweet_id = ['', '', '', '', '', '', '', '', '', '']
created_at4 = ['', '', '', '', '', '', '', '', '', '']
usernames = ['', '', '', '', '', '', '', '', '', '']
Armielyn11['new_tweet_id'] = new_tweet_id
Armielyn11['created_at4'] = created_at4
Armielyn11['usernames'] = usernames
Armielyn11
```

Out[11]:

	tweet_id	created_at	username	new_tweet_id	created_at4	usernames
0	3017	Wednesday/February/16	yyy			
1	412	Tuesday/February/16	yyy			
2	1702	Monday/February/16	yyy			
3	3463	Tuesday/December/16	yyy			
4	9948	Tuesday/December/16	yyy			
5	2678	Tuesday/December/16	yyy			
6	9004	Monday/December/16	yyy			
7	8761	Monday/December/16	yyy			
8	2441	Monday/December/16	yyy			
9	4784	Thursday/December/16	yyy			

In [12]:

```
Armielyn11['new_tweet_id'] = Armielyn11['tweet_id']
Armielyn11['created_at4'] = Armielyn11['created_at']
Armielyn11['usernames'] = Armielyn11['username']
mask_len = 3
Armielyn11['usernames'] = (
    Armielyn11['usernames'].astype(str).str[:-mask_len]+"y" * mask_len)

Armielyn11

```

Out[12]:

	tweet_id	created_at	username	new_tweet_id	created_at4	usernames
0	3017	Wednesday/February/16	yyy	3017	Wednesday/February/16	yyy
1	412	Tuesday/February/16	yyy	412	Tuesday/February/16	yyy
2	1702	Monday/February/16	yyy	1702	Monday/February/16	yyy
3	3463	Tuesday/December/16	yyy	3463	Tuesday/December/16	yyy
4	9948	Tuesday/December/16	yyy	9948	Tuesday/December/16	yyy
5	2678	Tuesday/December/16	yyy	2678	Tuesday/December/16	yyy
6	9004	Monday/December/16	yyy	9004	Monday/December/16	yyy
7	8761	Monday/December/16	yyy	8761	Monday/December/16	yyy
8	2441	Monday/December/16	yyy	2441	Monday/December/16	yyy
9	4784	Thursday/December/16	yyy	4784	Thursday/December/16	yyy

Question 3

Question 3: Data for this question can be found in "online-retail" sheet from assignment-data.xlsx. Since it is a big data, load first 200 rows and keep it in the data frame called "dataset". This “dataset” is used for all tasks in this question

Assume that you are a data scientist in Amazon. Since the company is celebrating Silver Jubilee this year, it has decided to reward their customers. Your Manager handed over last 2 years retail data and asked you to do certain tasks. The tasks are as follows:

Task1-1:When you started working with data, you’ve realized that it needs cleaning to produce better results. Do essential data cleaning. The final output should be the one as follows

In [13]:

```
Data1 = pd.read_csv('Online Retail_200.csv')
```

In [14]:

```
Data1
```

Out[14]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01/12/2010 8:26	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01/12/2010 8:26	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01/12/2010 8:26	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01/12/2010 8:26	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01/12/2010 8:26	3.39	17850	United Kingdom
...	...	...	...	...	...	...	...	...
194	536388	21115	ROSE CARAVAN DOORSTOP	4	01/12/2010 9:59	6.75	16250	United Kingdom
195	536388	22469	HEART OF WICKER SMALL	12	01/12/2010 9:59	1.65	16250	United Kingdom
196	536388	22242	5 HOOK HANGER MAGIC TOADSTOOL	12	01/12/2010 9:59	1.65	16250	United Kingdom
197	536389	22941	CHRISTMAS LIGHTS 10 REINDEER	6	01/12/2010 10:03	8.50	12431	Australia
198	536389	21622	VINTAGE UNION JACK CUSHION COVER	8	01/12/2010 10:03	4.95	12431	Australia

199 rows × 8 columns

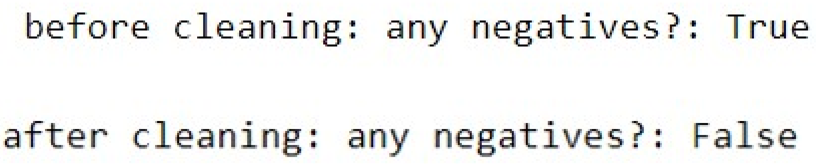
Type *Markdown* and LaTeX: α^2

Task 1

In [15]:

```
from IPython.display import Image
Image(filename = 'Screenshot_76.jpg')
```

Out[15]:



```
 before cleaning: any negatives?: True

after cleaning: any negatives?: False
```

In [16]:

```
# Import library
import numpy as np
import pandas as pd

# Define the dataframe
Data1 = pd.DataFrame(np.array([[1,2],[np.nan,3]]),columns = ['A','B'])

if pd.isnull(Data1).sum().sum() ==1:
  print('before clearning , any negatives? : True')

Data1.fillna(0,inplace = True)
if pd.isnull(Data1).sum().sum() ==0:
  print('after clearning , any negatives? : False')
```

```
before clearning , any negatives? : True
after clearning , any negatives? : False
```

In [17]:

```python
import pandas
# Reading csv file using pandas
df = pandas.read_csv('Online Retail_200.csv')

print(df)
print()

for columns in df:
    # Checking values of each column
    for rows in range(len(df[columns])):
        try:
            # If number is negative coonvert to positive
            df[columns][rows] = abs(int(df[columns][rows]))
        except:
            pass

```

```
    InvoiceNo StockCode                          Description  Quantity  \
0      536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6   
1      536365     71053                  WHITE METAL LANTERN         6   
2      536365    84406B       CREAM CUPID HEARTS COAT HANGER         8   
3      536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6   
4      536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6   
..        ...       ...                                  ...       ...   
194    536388     21115                 ROSE CARAVAN DOORSTOP         4   
195    536388     22469                 HEART OF WICKER SMALL        12   
196    536388     22242           5 HOOK HANGER MAGIC TOADSTOOL        12   
197    536389     22941          CHRISTMAS LIGHTS 10 REINDEER         6   
198    536389     21622         VINTAGE UNION JACK CUSHION COVER         8   

          InvoiceDate  UnitPrice  CustomerID         Country  
0     01/12/2010 8:26       2.55       17850  United Kingdom  
1     01/12/2010 8:26       3.39       17850  United Kingdom  
2     01/12/2010 8:26       2.75       17850  United Kingdom  
3     01/12/2010 8:26       3.39       17850  United Kingdom  
4     01/12/2010 8:26       3.39       17850  United Kingdom  
..                ...        ...         ...             ...  
194   01/12/2010 9:59       6.75       16250  United Kingdom  
195   01/12/2010 9:59       1.65       16250  United Kingdom  
196   01/12/2010 9:59       1.65       16250  United Kingdom  
197  01/12/2010 10:03       8.50       12431       Australia  
198  01/12/2010 10:03       4.95       12431       Australia  

[199 rows x 8 columns]


<ipython-input-17-a4a6b0c89f23>:13: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df[columns][rows] = abs(int(df[columns][rows]))
```

In [18]:

```
df
```

Out[18]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01/12/2010 8:26	2.0	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01/12/2010 8:26	3.0	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01/12/2010 8:26	2.0	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01/12/2010 8:26	3.0	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01/12/2010 8:26	3.0	17850	United Kingdom
...	...	...	...	...	...	...	...	...
194	536388	21115	ROSE CARAVAN DOORSTOP	4	01/12/2010 9:59	6.0	16250	United Kingdom
195	536388	22469	HEART OF WICKER SMALL	12	01/12/2010 9:59	1.0	16250	United Kingdom
196	536388	22242	5 HOOK HANGER MAGIC TOADSTOOL	12	01/12/2010 9:59	1.0	16250	United Kingdom
197	536389	22941	CHRISTMAS LIGHTS 10 REINDEER	6	01/12/2010 10:03	8.0	12431	Australia
198	536389	21622	VINTAGE UNION JACK CUSHION COVER	8	01/12/2010 10:03	4.0	12431	Australia

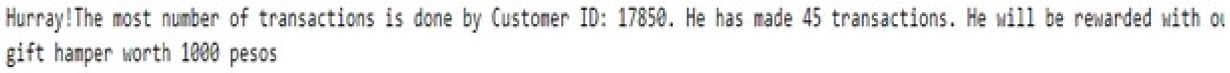
199 rows × 8 columns

Data Cleaning

Task 2

In [19]:
```python
from IPython.display import Image
Image(filename = 'Screenshot_74.jpg')
```

Out[19]:



```
Hurray!The most number of transactions is done by Customer ID: 17850. He has made 45 transactions. He will be rewarded with ou
gift hamper worth 1000 pesos
```

Task 3

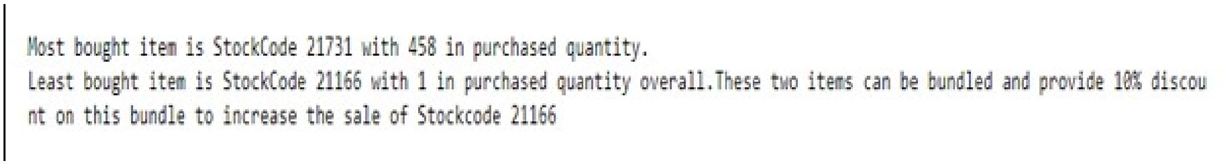
In [ ]:
```python
from IPython.display import Image
Image(filename = 'Screenshot_72.jpg')
```

In [ ]:
```python
 multiply the Quantity and the UnitPrice and create a new #column called values w









ustomer since he has spent a lot on our business. His total purchase amount is ",
```

Task 4

In [25]:
```python
from IPython.display import Image
Image(filename = 'Screenshot_67.jpg')
```

Out[25]:



```
Most bought item is StockCode 21731 with 458 in purchased quantity.
Least bought item is StockCode 21166 with 1 in purchased quantity overall.These two items can be bundled and provide 10% discou
nt on this bundle to increase the sale of Stockcode 21166
```

In [30]:

```
#here we find the maximum quantity and the low quantity purchase
Data1 = pd.read_csv('Online Retail_200.csv')

max_value = Data1['Quantity'].max()
min_value = Data1['Quantity'].min()

#finding the row which thaat value contains
Data2 = Data1[Data1['Quantity'] == max_value]
Data3 = Data1[Data1['Quantity'] == min_value ]

#print the required results
print("Most bought item is StockCode ",Data2['StockCode']," with ",max_value
print("Least bought item is StockCode ",Data3['StockCode']," with ",min_valu

```

```
Most bought item is StockCode  181    22466
182    21731
Name: StockCode, dtype: object  with  432  in purchase quantity.
Least bought item is StockCode  141         D
154    35004C
Name: StockCode, dtype: object  with  -1  in purchased quantity overall. These
two items can be bundled and provide 10% discount on this bundle to increase th
e sale pf StockCode  141         D
154    35004C
Name: StockCode, dtype: object
```

Task 5

In [ ]:

```
from IPython.display import Image
Image(filename = 'Screenshot_68.jpg')
```

In [31]:

```
# program starts from here

# import Pandas library
import pandas as pd
# read csv file store in same location where this program file is stored
# name of csv file = amazon.csv
# read this csv file and store in dataframe object data1

data1 = pd.read_csv("Online Retail_200.csv") # make sure file is store with

# print head
data1.head(200)

# add a column in dataframe data1 for purchase price
# Purchase Price = Quantity * Unit Price

data1[['PurchasePrice']] = data1['Quantity']*data1['UnitPrice']

# calculate average purchase price and store it in a variable
average_purchase_price = data1.groupby('InvoiceNo')['PurchasePrice'].sum().m

```

In [32]:

```
# print the result
print("Overall average purchase amount among all transaction is : {:.2f}".fo
```

```
Overall average purchase amount among all transaction is : 351.95
```

Task 6

In [22]:

```
import pandas as pd
data=[[15100,350.40,1],[15291,328.80,2],[15311,454.63,36],[16029,3702.12,8],
      [16098,430.60,12],[16250,226.14,14],[17420,130.85,7],[17809,34.80,1],
      [17850,725.44,45],[18074,489.60,13]]
df=pd.DataFrame(data, columns=['CustomerID','TotalPurchaseAmount','NumberOfT
print(df)
```

```
   CustomerID  TotalPurchaseAmount  NumberOfTransactions
0       15100               350.40                     1
1       15291               328.80                     2
2       15311               454.63                    36
3       16029              3702.12                     8
4       16098               430.60                    12
5       16250               226.14                    14
6       17420               130.85                     7
7       17809                34.80                     1
8       17850               725.44                    45
9       18074               489.60                    13
```

In [23]:

```
df
```

Out[23]:

	CustomerID	TotalPurchaseAmount	NumberOfTransactions
0	15100	350.40	1
1	15291	328.80	2
2	15311	454.63	36
3	16029	3702.12	8
4	16098	430.60	12
5	16250	226.14	14
6	17420	130.85	7
7	17809	34.80	1
8	17850	725.44	45
9	18074	489.60	13

Task 7

In [23]:

```

# import pandas with alias pd
import pandas as pd

# define a list for each column
customer_id = [12341, 12583, 13047, 13748, 14527]
total_purchase_amount = [105.60, 855.86, 366.63, 204.00, 27.50]
num_txns = [3, 20, 17, 1, 1]
avg_purchase = [35.200000, 42.793000, 21.566471, 204.000000, 27.500000]

# create a dictionary using above defined lists
# here keys are the column names and values are the corresponding lists
data = {
    'CustomerID': customer_id,
    'TotalPurchaseAmount': total_purchase_amount,
    'NumberOfTransactions': num_txns,
    'AveragePurchase': avg_purchase
}

# create dataframe
df = pd.DataFrame(data)

# print dataframe
df
```

Out[23]:

	CustomerID	TotalPurchaseAmount	NumberOfTransactions	AveragePurchase
0	12341	105.60	3	35.200000
1	12583	855.86	20	42.793000
2	13047	366.63	17	21.566471
3	13748	204.00	1	204.000000
4	14527	27.50	1	27.500000