**1.** Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.    
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
attempts name qualify score   
a 1 Anastasia yes 12.5   
b 3 Dima no 9.0  
.... i 2 Kevin no 8.0   
j 1 Jonas yes 19.0 

**2.** Write a Pandas program to display a summary of the basic information about a specified DataFrame and its data.   
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
Summary of the basic information about this DataFrame and its data:  
<class 'pandas.core.frame.DataFrame'>  
Index: 10 entries, a to j  
Data columns (total 4 columns):  
.... dtypes: float64(1), int64(1), object(2)  
memory usage: 400.0+ bytes  
None 

**3.** Write a Pandas program to select the specified columns and rows from a given data frame.   
Sample Python dictionary data and list labels:  
Select 'name' and 'score' columns in rows 1, 3, 5, 6 from the following data frame.  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
Select specific columns and rows:  
name score   
b Dima 9.0   
d James NaN   
f Michael 20.0   
g Matthew 14.5 

**4.** Write a Pandas program to select the rows where the score is missing, i.e. is NaN.   
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
Rows where score is missing:  
attempts name qualify score  
d 3 James no NaN  
h 1 Laura no NaN

**5.** Write a Pandas program to select the rows the score is between 15 and 20 (inclusive).   
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
Rows where score between 15 and 20 (inclusive):  
attempts name qualify score   
c 2 Katherine yes 16.5   
f 3 Michael yes 20.0   
j 1 Jonas yes 19.0 

**6.** Write a Pandas program to calculate the sum of the examination attempts by the students.   
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
Sum of the examination attempts by the students:  
19 

**7.** Write a Pandas program to sort the DataFrame first by 'name' in descending order, then by 'score' in ascending order.   
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Values for each column will be:  
name : "Suresh", score: 15.5, attempts: 1, qualify: "yes", label: "k"  
Expected Output:  
Sort the data frame first by 'name' in descending order, then by 'score' in ascending order:   
attempts name qualify score   
a 1 Anastasia yes 12.5  
b 3 Dima no 9.0  
.....   
i 2 Kevin no 8.0   
j 1 Jonas yes 19.0 

**8.** Write a Pandas program to get list from DataFrame column headers.   
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
Expected Output:  
['attempts', 'name', 'qualify', 'score']

**9.** Write a Pandas program to rename columns of a given DataFrame   
Sample data:  
Original DataFrame  
col1 col2 col3  
0 1 4 7  
1 2 5 8  
2 3 6 9  
New DataFrame after renaming columns:  
Column1 Column2 Column3  
0 1 4 7  
1 2 5 8  
2 3 6 9

**10.** Write a Pandas program to select rows from a given DataFrame based on values in some columns.   
Sample data:  
Original DataFrame  
col1 col2 col3  
0 1 4 7  
1 4 5 8  
2 3 6 9  
3 4 7 0  
4 5 8 1  
Rows for colum1 value == 4  
col1 col2 col3  
1 4 5 8  
3 4 7 0

**11.** Write a Pandas program to widen output display to see more columns.   
Sample data:  
Original DataFrame  
col1 col2 col3  
0 1 4 7  
1 4 5 8  
2 3 6 9  
3 4 7 0  
4 5 8 1

**12.** Write a Pandas program to replace all the NaN values with Zero's in a column of a dataframe.   
Sample data:  
Original DataFrame  
attempts name qualify score  
0 1 Anastasia yes 12.5  
1 3 Dima no 9.0  
2 2 Katherine yes 16.5  
3 3 James no NaN  
4 2 Emily no 9.0  
5 3 Michael yes 20.0  
6 1 Matthew yes 14.5  
7 1 Laura no NaN  
8 2 Kevin no 8.0  
9 1 Jonas yes 19.0  
New DataFrame replacing all NaN with 0:  
attempts name qualify score  
0 1 Anastasia yes 12.5  
1 3 Dima no 9.0  
2 2 Katherine yes 16.5  
3 3 James no 0.0  
4 2 Emily no 9.0  
5 3 Michael yes 20.0  
6 1 Matthew yes 14.5  
7 1 Laura no 0.0  
8 2 Kevin no 8.0  
9 1 Jonas yes 19.0

**13.** Write a Pandas program to reset index in a given DataFrame.   
Sample data:  
Original DataFrame  
attempts name qualify score  
0 1 Anastasia yes 12.5  
1 3 Dima no 9.0  
2 2 Katherine yes 16.5  
3 3 James no NaN  
4 2 Emily no 9.0  
5 3 Michael yes 20.0  
6 1 Matthew yes 14.5  
7 1 Laura no NaN  
8 2 Kevin no 8.0  
9 1 Jonas yes 19.0  
After removing first and second rows  
attempts name qualify score  
2 2 Katherine yes 16.5  
3 3 James no NaN  
4 2 Emily no 9.0  
5 3 Michael yes 20.0  
6 1 Matthew yes 14.5  
7 1 Laura no NaN  
8 2 Kevin no 8.0  
9 1 Jonas yes 19.0  
Reset the Index:  
index attempts name qualify score  
0 2 2 Katherine yes 16.5  
1 3 3 James no NaN  
2 4 2 Emily no 9.0  
3 5 3 Michael yes 20.0  
4 6 1 Matthew yes 14.5  
5 7 1 Laura no NaN  
6 8 2 Kevin no 8.0  
7 9 1 Jonas yes 19.0

**14.** Write a Pandas program to devide a DataFrame in a given ratio.  
Sample data:  
Original DataFrame:  
0 1  
0 0.316147 -0.767359  
1 -0.813410 -2.522672  
2 0.869615 1.194704  
3 -0.892915 -0.055133  
4 -0.341126 0.518266  
5 1.857342 1.361229  
6 -0.044353 -1.205002  
7 -0.726346 -0.535147  
8 -1.350726 0.563117  
9 1.051666 -0.441533  
70% of the said DataFrame:  
0 1  
8 -1.350726 0.563117  
2 0.869615 1.194704  
5 1.857342 1.361229  
6 -0.044353 -1.205002  
3 -0.892915 -0.055133  
1 -0.813410 -2.522672  
0 0.316147 -0.767359  
30% of the said DataFrame:  
0 1  
4 -0.341126 0.518266  
7 -0.726346 -0.535147  
9 1.051666 -0.441533

**15.** Write a Pandas program to convert DataFrame column type from string to datetime.  
  
Sample data:  
String Date:  
0 3/11/2000  
1 3/12/2000  
2 3/13/2000  
dtype: object  
Original DataFrame (string to datetime):  
0  
0 2000-03-11  
1 2000-03-12  
2 2000-03-13

**16.** Write a Pandas program to create a DataFrame from a Numpy array and specify the index column and column headers.   
Sample Output:  
Column1 Column2 Column3  
Index1 0 0.0 0.0  
Index2 0 0.0 0.0  
Index3 0 0.0 0.0  
.........  
Index12 0 0.0 0.0  
Index13 0 0.0 0.0  
Index14 0 0.0 0.0  
Index15 0 0.0 0.0

**17.** Write a Pandas program to find the row for where the value of a given column is maximum.   
Sample Output:  
Original DataFrame  
col1 col2 col3  
0 1 4 7  
1 2 5 8  
2 3 6 12  
3 4 9 1  
4 7 5 11  
Row where col1 has maximum value:  
4  
Row where col2 has maximum value:  
3  
Row where col3 has maximum value:  
2

**18.** Write a Pandas program to convert the datatype of a given column (floats to ints).   
Sample data:  
Original DataFrame:  
attempts name qualify score  
0 1 Anastasia yes 12.50  
1 3 Dima no 9.10  
......  
8 2 Kevin no 8.80  
9 1 Jonas yes 19.13  
Data types of the columns of the said DataFrame:  
attempts int64  
name object  
qualify object  
score float64  
dtype: object  
Now change the Data type of 'score' column from float to int:  
attempts name qualify score  
0 1 Anastasia yes 12  
1 3 Dima no 9  
2 2 Katherine yes 16  
3 3 James no 12  
4 2 Emily no 9  
5 3 Michael yes 20  
6 1 Matthew yes 14  
7 1 Laura no 11  
8 2 Kevin no 8  
9 1 Jonas yes 19  
Data types of the columns of the DataFrame now:  
attempts int64  
name object  
qualify object  
score int64  
dtype: object

**19.** Write a Pandas program to convert a given list of lists into a Dataframe.   
Sample data:  
Original list of lists:  
[[2, 4], [1, 3]]  
New DataFrame  
col1 col2  
0 2 4  
1 1 3

**20.** Write a Pandas program to group by the first column and get second column as lists in rows.   
Sample data:  
Original DataFrame  
col1 col2  
0 C1 1  
1 C1 2  
2 C2 3  
3 C2 3  
4 C2 4  
5 C3 6  
6 C2 5  
Group on the col1:  
col1  
C1 [1, 2]  
C2 [3, 3, 4, 5]  
C3 [6]  
Name: col2, dtype: object

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Scikit learn:

1. Please create a Logistic Regression model against the dataset < UCI\_Credit\_Card.csv> using all possible independent variable, dependent variable is: default.payment.next.month. (Please split into Training-80% and Test – 20% Randomly and test your training model onto test data).
2. Visualize the Prediction vs Actual, and also the basic Test output including, but not limit to T-Test, P-value etc.