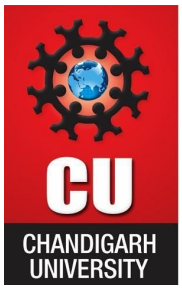


University Institute of Engineering Department of Computer Science & Engineering

Experiment: 1.2

Student Name: UID:

Branch: CSE Section/Group

Semester: 1st Date of Performance:

Subject Name: Disruptive Technologies Subject Code: 21ECP-102

1. Aim of the practical:

Explore, visualize, transform and summarize input datasets for building Classification/regression/prediction models

2. Tool Used:

VS CODE

3. Basic Concept/ Command Description:

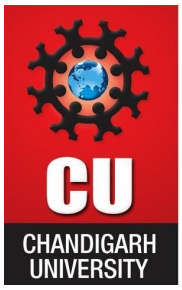
It is a bundle of many Machine Learning algorithms. Only three lines of code is required to compare 20 ML models. Pycaret is available for:

Classification

Regression
Clustering

---

4. Code and Observations, Simulation Screen Shots



University Institute of Engineering Department of Computer Science & Engineering

and Discussions:

(a) Install Pycaret

```
!pip install pycaret &> /dev/null
print("Pycaret installed sucessfully!!")
```

```
Pycaret installed sucessfully!!
```

(b) Get the version of the pycaret

```
from pycaret.utils import version
version()
```

```
'2.3.4'
```

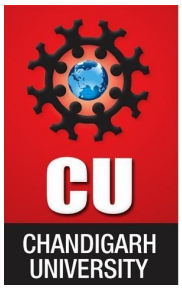
1. Classification: Basics

1.1 Loading Dataset - Loading dataset from pycaret

```
from pycaret.datasets import get_data
#No output
```

1.2 Get the list of datasets available in pycaret (55)

```
#Internet connection is required
datasets = get_data('index')
```



University Institute of Engineering Department of Computer Science & Engineering

]

	Dataset	Data Types	Default Task	Target Variable 1	Target Variable 2	# Instances	# Attributes	Missing Values
0	anomaly	Multivariate	Anomaly Detection	None	None	1000	10	N
1	france	Multivariate	Association Rule Mining	InvoiceNo	Description	8557	8	N
2	germany	Multivariate	Association Rule Mining	InvoiceNo	Description	9495	8	N
3	bank	Multivariate	Classification (Binary)	deposit	None	45211	17	N
4	blood	Multivariate	Classification (Binary)	Class	None	748	5	N
5	cancer	Multivariate	Classification (Binary)	Class	None	683	10	N
6	credit	Multivariate	Classification (Binary)	default	None	24000	24	N
7	diabetes	Multivariate	Classification (Binary)	Class variable	None	768	9	N
8	electrical_grid	Multivariate	Classification (Binary)	stabf	None	10000	14	N
9	employee	Multivariate	Classification (Binary)	left	None	14999	10	N
10	heart	Multivariate	Classification (Binary)	DEATH	None	200	16	N
11	heart_disease	Multivariate	Classification (Binary)	Disease	None	270	14	N
12	hepatitis	Multivariate	Classification (Binary)	Class	None	154	32	Y
13	income	Multivariate	Classification (Binary)	income >50K	None	32561	14	Y
14	juice	Multivariate	Classification (Binary)	Purchase	None	1070	15	N
15	nba	Multivariate	Classification (Binary)	TARGET_5Yrs	None	1340	21	N
16	wine	Multivariate	Classification (Binary)	type	None	6498	13	N
17	telescope	Multivariate	Classification (Binary)	Class	None	19020	11	N
18	titanic	Multivariate	Classification (Binary)	Survived	None	891	11	Y

1.3 Get diabetes dataset

```
diabetesDataSet = get_data("diabetes")
```

	Number of times pregnant	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Diastolic blood pressure (mm Hg)	Triceps skin fold thickness (mm)	2-Hour serum insulin (mu U/ml)	Body mass index (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

University Institute of Engineering Department of Computer Science & Engineering

2.Read data from file

```
#import pandas as pd
#diabetesDataSet = pd.read_csv("diabetes.csv")

diabetesDataSet.columns
```

```
Index(['Number of times pregnant',
       'Plasma glucose concentration a 2 hours in an oral glucose tolerance test',
       'Diastolic blood pressure (mm Hg)', 'Triceps skin fold thickness (mm)',
       '2-Hour serum insulin (mu U/ml)',
       'Body mass index (weight in kg/(height in m)^2)',
       'Diabetes pedigree function', 'Age (years)', 'Class variable'],
      dtype='object')
```

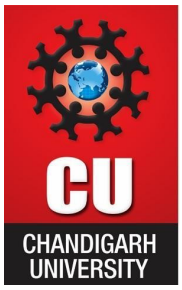
#Get the statistical summary of the dataset diabetesDataSet.describe()

```
#import pandas as pd
#diabetesDataSet = pd.read_csv("diabetes.csv")


#Get the statistical summary of the dataset
diabetesDataSet.describe()
```

	Number of times pregnant	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Diastolic blood pressure (mm Hg)	Triceps skin fold thickness (mm)	2-Hour serum insulin (mu U/ml)	Body mass index (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

#Accessing data from dataset part1



University Institute of Engineering Department of Computer Science & Engineering

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
diabetes = pd.read_csv('/diabetes.csv')
print(diabetes.columns)
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

#print("type(df) ",type(diabetesDataSet))

```
print("type->df",type(diabetesDataSet))
```

```
type->df <class 'pandas.core.frame.DataFrame'>
```

##Get the dimension of datset

```
import pandas as pd
diabetes = pd.read_csv('/diabetes.csv')
print("Diabetes data set dimensions : {}".format(diabetes.shape))
```

```
Diabetes data set dimensions : (768, 9)
```

#Show top 5 rows of Dataset

University Institute of Engineering Department of Computer Science & Engineering

```
import pandas as pd
df = pd.read_csv('/diabetes.csv')
result = df.head(5)
print("First 5 rows of the DataFrame:")
print(result)
```

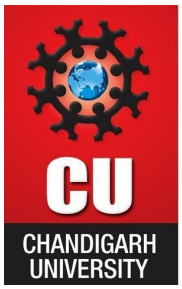
```
First 5 rows of the DataFrame:
   Pregnancies  Glucose  BloodPressure  ...  DiabetesPedigreeFunction  Age  Outcome
0            6      148             72  ...                     0.627   50        1
1            1       85             66  ...                     0.351   31        0
2            8      183             64  ...                     0.672   32        1
3            1       89             66  ...                     0.167   21        0
4            0      137             40  ...                     2.288   33        1

[5 rows x 9 columns]
```

#Get the maximum of each column in the dataset

```
[ ] import pandas as pd
    df = pd.read_csv('/diabetes.csv')
    result= df.max()
    print("max value of all columns")
    print(result)
```

```
max value of all columns
Pregnancies                 17.00
Glucose                    199.00
BloodPressure              122.00
SkinThickness               99.00
Insulin                    846.00
BMI                         67.10
DiabetesPedigreeFunction     2.42
Age                         81.00
Outcome                      1.00
dtype: float64
```



University Institute of Engineering Department of Computer Science & Engineering

#Get the mean of the all the columns present in the dataset

```
[ ]  import pandas as pd
     df = pd.read_csv('/diabetes.csv')
     result = df.mean()
     print("means of all coumns:")
     print(result)
```

```
means of all coumns:
Pregnancies                 3.845052
Glucose                   120.894531
BloodPressure              69.105469
SkinThickness              20.536458
Insulin                    79.799479
BMI                        31.992578
DiabetesPedigreeFunction    0.471876
Age                        33.240885
Outcome                     0.348958
dtype: float64
```

#Drop the duplicates present in the dataset.

```
[ ]  import pandas as pd
     result = df.drop_duplicates()
     print('Result DataFrame:\n', result)
```

```
Result DataFrame:
      Pregnancies  Glucose  ...  Age  Outcome
0               6      148  ...   50        1
1               1       85  ...   31        0
2               8      183  ...   32        1
3               1       89  ...   21        0
4               0      137  ...   33        1
..            ...      ...  ...  ...      ...
763            10      101  ...   63        0
764             2      122  ...   27        0
765             5      121  ...   30        0
766             1      126  ...   47        1
767             1       93  ...   23        0

[768 rows x 9 columns]
```

Drop NA values (delete rows)

Drop NA values (delete rows)

University Institute of Engineering Department of Computer Science & Engineering

```
import pandas as pd
result = df.dropna
print('Result DataFrame:\n', result)
```

```
Result DataFrame:
 <bound method DataFrame.dropna of      Pregnancies  Glucose  ...  Age  Outcome
0              6      148  ...   50        1
1              1       85  ...   31        0
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]>
```

Fill the null values with '0'

```
import pandas as pd
result = df.fillna(0)
print('Result DataFrame:\n', result)
```

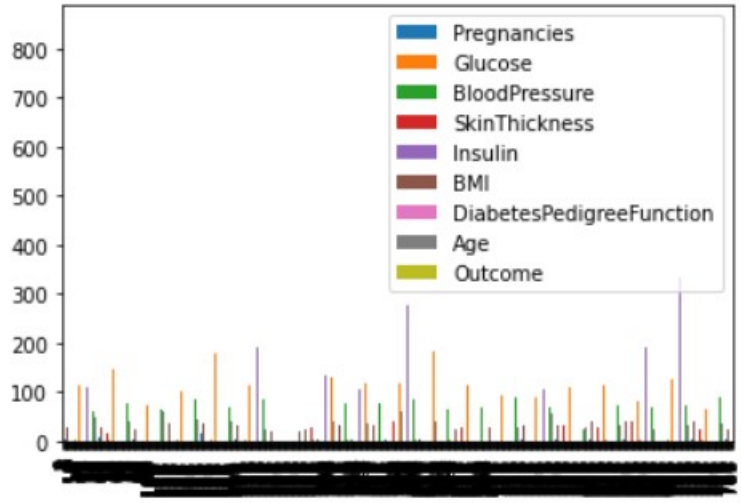
```
Result DataFrame:
      Pregnancies  Glucose  ...  Age  Outcome
0              6      148  ...   50        1
1              1       85  ...   31        0
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]
```

Bar graph

```
[ ]  import matplotlib. pyplot as plt
     import numpy as np
     df =pd.read_csv('/diabetes.csv')
     df. plot(kind='bar')
```

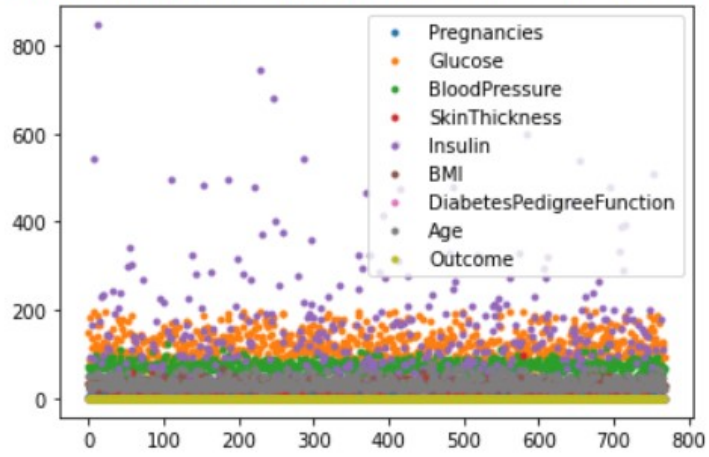
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59f8ea1150>
```



Scatter plot

```
[ ]  import matplotlib. pyplot as plt
     import numpy as np
     df =pd.read_csv('/diabetes.csv')
     df. plot(style=".")
```

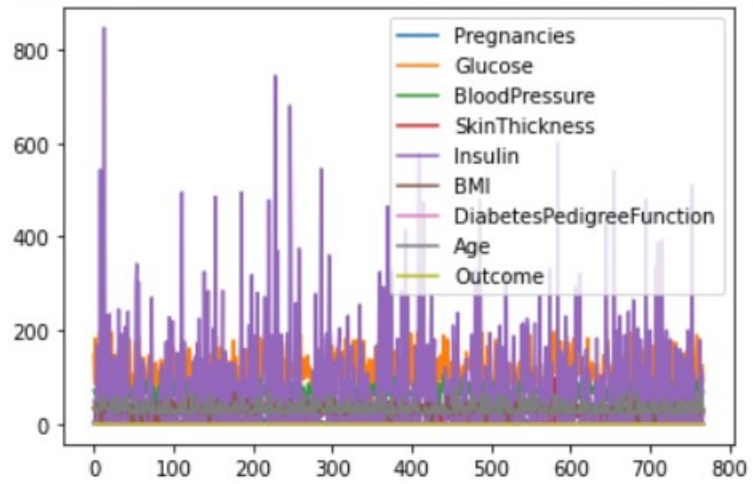
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59f3352410>
```



Subplot

```
[ ]  import matplotlib. pyplot as plt
     import numpy as np
     df =pd.read_csv('/diabetes.csv')
     df. plot(kind='line')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59f28cf1d0>
```



6. Result and Summary:

PyCaret's Classification Module is a supervised machine learning module which is used for classifying elements into groups. The goal is to predict the categorical class labels which are discrete and unordered. Some common use cases include predicting customer default (Yes or No), predicting customer churn (customer will leave or stay), disease found (positive or negative). This module can be used for binary or multiclass problems.

7. Additional Creative Inputs (If Any):

Learning outcomes (What I have learnt):

1. Getting Data: How to import data from PyCaret repository

2. Setting up Environment: How to setup an experiment in PyCaret and get started with building regression models

University Institute of Engineering Department of Computer Science & Engineering

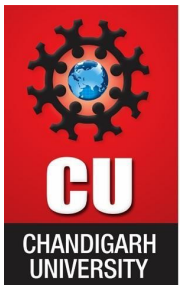
3. Create Model: How to create a model, perform cross validation and evaluate regression metrics

4. Tune Model: How to automatically tune the hyperparameters of a regression model

5. Plot Model: How to analyze model performance using various plots

Evaluation Grid (To be filled by Faculty):

Sr. No.	**Parameters**	**Marks Obtained**	**Maximum Marks**
1.	Worksheet completion including writinglearning objectives/Outcomes.(To besubmitted at the end of the day)		10
2.	Post Lab Quiz Result.		5
3.	Student Engagement in Simulation/Demonstration/Performanc e and Controls/Pre-Lab Questions.		5
	Signature of Faculty (with Date):	**Total Marks Obtained:**	**20**



University Institute of Engineering Department of Computer Science & Engineering