# Assignment-1

1.
```
i=0
n=10
while i<10:
    i=i+1
    print(i)
```

2.
```
n=4
for i in range(4):
    for j in range(i+1):
        print("*", end="")
    for j in range(i,n):
        print("#", end="")
    print()
```

3.
```
def add(n):
    total=0
    for x in n:
        total=total+x
    return total

add([10,5,8])
```
23

4.
```
for i in range(1,11)
    print("5X", i, "=", i*5)
```

5.
```
list=[1,2,3,4,5,6,7,8]
for list in list:
    print(list)
```

6.
```
num=23567
print(len(str(num)))
```

7.
```
print list in
li=['o','n','u','s','h','k','A']
for i in range(len(li)-1,-1,-1):
    print(li[i])
```

8.
```
for in range (-10,0):
    print (i)
```

9.
```
num = 0
if num > 0:
    print ("positive no.")
elif num >= 0:
    print ('zero')
else:
    print ("negative no.")
print ('DONE')
```

10.
```
for ir in range (2,99):
    for j in range (2,99):
        if i % j == 0:
            break
    if i == j:
        print (i, end = " ,")
```

11.
```
A = int(input(" Enter a No." ))
n1 = 0
n2 = 1
for i in range (2,10)
    n3 = n1 + n2
    print( n3)
    n1 = n2
    n2 = n3
```

12.
```
S = int (input ("Enter a No.")
fact = 1
for i in range (1,6)
    Fact = Fact * i
print (Fact)
```

13.
```
a = [21, 22, 25, 30]
for i in range (0,2,3):
    print (a[i])
```

15. for i in range (10):
    print (i * i)

16. n = 5
for x in range (n):
    print (" " * (n-x), "*" * (2*x + 1))
for x in range (n-2, -1, -1):
    print (" " * (n-x), "*" * (2*x + 1))

17 def sum(x):
    if x == 0:
        return 0
    else:
        return ((x * (x+1)) / 2)

18 num (n):
    m = 1
    for i in range (0, n):
        n ~~num~~ = 1
    for j in range (0, i+1):
        print (~~num~~ n, end = " ")
    print ("\r")
    n = 5

Assignment-2

Q.1 Calculate the z-score for the below data set assume sd = 1.5 How do u perform normalization (only formula) data.

2
3
1
3
2
4

→ we have, $z = \frac{x-\mu}{\sigma}$

where, z is the z-score
x is the individual data points.
$\mu$ is the mean.
$\sigma$ is the standard deviation.

Here $\sigma = 1.5$,
and, the dataset is : 2 3 1 3 2 4
The mean ($\mu$) is calculated as,

$$\mu = \frac{2+3+1+3+2+4}{6} = \frac{15}{6} = 2.5$$

Now, calculate z-score for each datapoints

i) for x = 2 :

$$z = \frac{2-2.5}{1.5} = \frac{-0.5}{1.5} = -0.333$$

ii) for x = 3 :

$$z = \frac{3-2.5}{1.5} = \frac{0.5}{1.5} = 0.333$$

iii) for x = 1 :

$$z = \frac{1-2.5}{1.5} = \frac{-1.5}{1.5} = -1$$

iv) for x = 3 :

$$z = \frac{3-2.5}{1.5} = 0.333$$

v) for x = 2 :

$$z = \frac{2-2.5}{1.5} = -0.333$$

vi) for x = 4 :

$$z = \frac{4-2.5}{1.5} = \frac{1.5}{1.5} = 1$$

So, the z score for the data set are approximately,
-0.333, 0.333, -1, 0.333, -0.333, 1

2) one-hot encoding is a technique used to convert categorical variables into a binary matrix, where each category is represented by a binary column. In pandas, the get_dummies function is commonly used for one hot encoding.

3) List all the transformers (function and power)

The question seems incomplete or ambiguous. if you're referring to mathematical transformations, it could include functions like square root, logarithm, exponential, etc, with different powers.

4) Linear regression assumes that the relationship between the independent and dependent variables is linear, and the residuals (the differences between actual and predicted values) are normally distributed with constant variance

5) Gradient desent is an optimization algorithm used to minimize the cost function in machine learning models. it iteratively adjusts model parameters in the direction of steepest decrease of the cost. The diagram typically shows a convergence towards the minimum point of cost function.

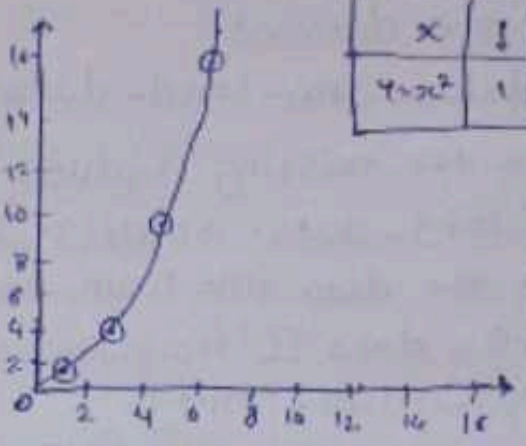
6. Pandas Profiling is a library used for generating exploratory data analysis reports for a pandas DataFrame. To use it:-

```
Python
copy code
import Pandas_Profiling

Profile =
Pandas_Profiling.ProfileReport(df)
Profile.to_file("output.html")
```

7. Draw the line for the following equation:- $y = x^2$

| x | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $y = x^2$ | 1 | 4 | 9 | 16 |




=

2
$y = x$
2
:
since
=
2
$y = x$
2
represents a quadratic function, the graph is a Parabola it. opens upwords if the conffrcient of

2
$x$
2

it positive. The specific shape and location of the parabole depend on the cofficient values.

8]

python
copy code

```
# import necessary libraries
import seaborn as sns
from sklearn.linearmodel-selection import
train_test_split
from sklearn.linear_model import
LinearRegression
from sklearn.metrics import
mean_squared_error
# Load the dataset
mpg_data = sms.load_dataset('mpg')
# check for missing values
print(mpg_data.isnull().sum())
# split the data into train and test sets
x = mpg_data [['horsepower']]
y = mpg_data ['mpg']
x = train, x_test, y_train, y_test = train
train_test_split (x,y,test_size=0,2) random_state=42)
# create and fill the model
model =LinearRegression()
model.fit(x_train, y_train)
# Evaluate the model
mse=mean-squared-error (y-test, y-pred)
printf(f' mean squared Error : {mse}')
```

This code imports the necessary libraries, loads the "mpg" dataset, checks for the missing values, splits the data into traning and testing sets, builds a linear regression model, makes predictions, and evaluates the model using mean squared error.