Assignment - 2.

1. Calculate the z ~~lear~~ 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 point,
μ is the mean,
σ is standard deviation.

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

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

Now calculate z-score for each datapoint.

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-scores for the data set are approximately,
-0.333, 0.333, -1, 0.333, -0.333, 1.

2) What is one hot encoding? Name the pandas function which perform OHE.

→ 1) One-hot enconding is a data preprocessing step to convert categorical values into compatible numerical representations

2) One-Hot Encoding can be implemented with pandas using the get_dummies function that takes the following parameters:

data : array, Series, or DataFrame - The data containing categorical variables of which to get dummy ~~var~~ indicators.

3) List all the transformers (function & power)?

→ 1) Function Transformer :-

i) Log Transformer

ii) Reciprocal Transformer

iii) Square Transformer

iv) Square root Transformer

v) Custom Transformer.

2) Power Transformer :-

i) Box cox

ii) Yeo Johnson.

4) Explain all the assumptions of linear regression in 2 lines?

→ 1) Linearity - Assumes a linear relationship between the independent & dependent variables, implying that the change in the mean of the dependent variable is proportional to a change in independent variable

2) Independence & Homoscedascity :-

Assumes that errors are independent & have constant variance across all levels of the independent variable, ensuring the reliability of model predictions.

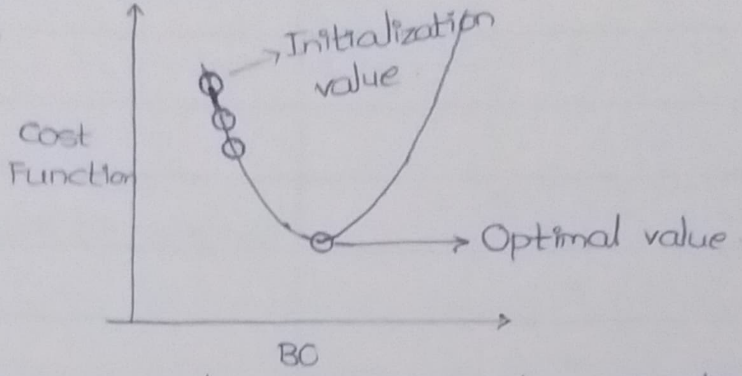
Equal Variance	Includes noautocorrelation.
Homoscedascity	Independence.

3) Lack of Multicollinearity :- (Predictors are not correlated with each other).

✓ $x_1 \not\sim x_2$. ⊗ $X_1 \sim X_2$.

5) What is the gradient descent algorithm ?Explain with a diagram:-

⇒ Gradient Descent is an optimization algorithm for finding the local minimum of a function 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 the cost function



- A regression model optimizes the gradient descent algorithm to update the coefficients of the line by reducing the cost function by randomly selecting coefficient values & then iteratively updating the values to reach the minimum cost function

6) What is pandas profiling ? Write a suitable syntax

⇒ Pandas profiling is a Python library that generates comprenhensive reports for datasets with numerous features. These reports can be customized according to specific requirements.

The syntax for this method is as follows :-

▷ pandas_profiling.ProfileReport (df, **kwargs)

Here, df is data to be analyzed & kwargs are optional arguments that can be passed to the method.

Some of the important arguments are :

1) Bins : Number of bins in histogram. The default is 1

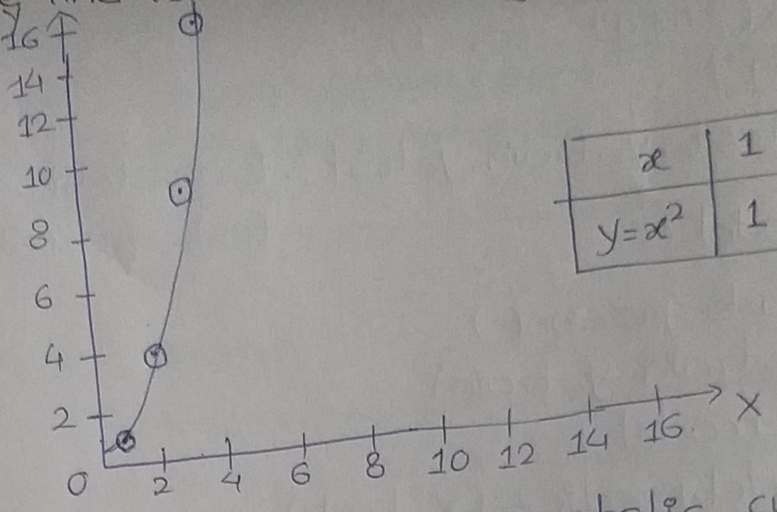
2) check_correlation : Whether or not to check correlat It's True by default

3) correlation_threshold : Threshold to determine if the variable pair is correlated. The default is 0.9.

4) correlation_overrides : Variable names not to be rejected because they are correlated. There is no variable in the list None by default

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

⇒



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

The eq^n $y=x^2$ represents a parabolic curve

8) Build the regression model for the "mpg" dataset (present in seaborn library).

import all the necessary libraries
import dataset from seaborn library.
check for missing value.
split the data into train & test.
fit the model.
predict the model

⇒

1) Import all the necessary libraries :-

```
import pandas as pd.
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt.
from sklearn.model_selection import train_test
from sklearn.linear_model import LinearRegressi
```

2) Load dataset from seaborn library :-

```
df = sns.load_dataset ("mpg")
```

3) Check for missing value :-

```
df.isnull().sum()
```

4) Split the data into train & test :-

```
X = df.drop ("mpg" ,axis=1)
y=df ["mpg"]
X_train, X_test ,y_train,y_test = train_test_split(X,y,
                                    test_size=0.2,random_state
```

5) Fit the model :-

```
model = LinearRegression( )
model. fit (X_train ,y_train)
```

) Predict the model :-

```
y_pred = model.predict, (X_test)
```