Q1 Answer the following question

1) Calculate Z score for following data

2, 3, 1, 3, 2, 4

Formula $\frac{x-\mu}{\sigma}$

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

For $x = 2$, $\frac{2-2.5}{1.5} = -0.333$

For $X = 3$, $z = \frac{3-1.5}{1.5} = 0.333$

For $X = 1$ $z = \frac{1-2.5}{1.5} = 1.0$

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

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

For $X = 4$ $z = \frac{4-2.5}{1.5} = 0.667$

Z-score for the data set is -0.333, 0.333, 1, 0.333, -0.333, 0.667

2) ~~Ans~~ Normalization formula -

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

2) One hot encoding

→ One hot encoding is used in categorical factors as binary vectors. This is helpful because machine learning algorithm generally act on numerical data

2) Pandas function which perform one hot encoding is get-dummies

3) List all the transformers

→ Function transformer

1) Log transformer
2) Reciprokal transformer
3) Square transformer
4) Square root transformer
5) Custom transformer

Power transformer

1) Box cox
2) Yeo johnson

4) Assumptions of linear regression

→ 1) The relationship between X & Y is linear

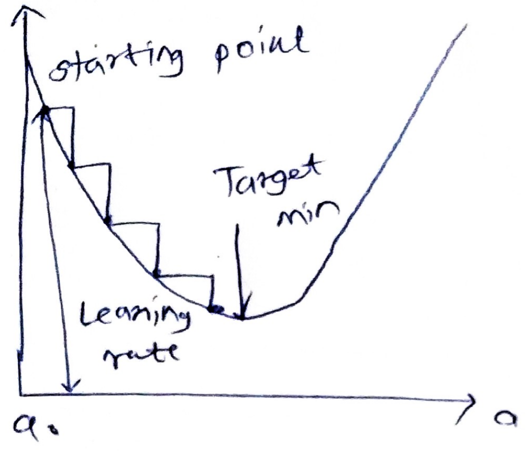
2) The variance of the residual is the same for any value of X

3) observations are independant of each other.

5) Gradient descent Algorithm with diagram.

→ Algorithm used to minimize a cost function by adjusting parameters in the direction of steepest decrease in the loss function.

The algorithm minimizes the loss function that measures the error betn predicted values & actual values.



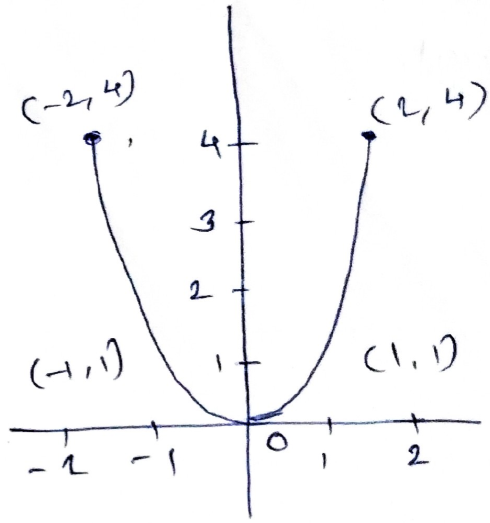
Q.6. Pandas profiling:

→ Pandas profiling is an python library that performs an automated exploratory data analysis (EDA). It automatically generates a datasets profile report that gives valuable insights. The report is generated in an HTML format which makes it easy to analyse

- Syntax for pandas profiling:

```
From pandas_profiling import
                    profilereport
Profile = profilereport (df)
profile to_file (index.html))
```

Q.7 $y = x^2$



Q.8)

```
i) Import pandas as pd
   Import seaborn as sns
   Import matplolib.pyplot as plt
From sklearn.model_selection
import train_test_split
From sklearn.linear_model
import LinearRegression
ii) df = sns.load_dataset('mpg')
iii) (df.isnull().sum())
iv) X = df.features
    y = df.target
 X_train, X_test, Y_train, Y_test =
train_test_split (X, Y, test_size =
                              0.2,
         random_state = 42)
v) model = LinearRegression()
```

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

vi)
```
y_pred = model. predict (x_test)
y pred.
```