

Pre-processing and EDA (Exploratory Data Analysis)

Preprocessing:

1. Taking care of redundant data – Missing Values, Outliers, Duplicates etc.

Handling Missing Values:

`df.isna().sum()`

Imputing:

From `sklearn.impute` import `SimpleImputer`

`Imputer = SimpleImputer(fill_value = np.nan, strategy = 'median')`

`X = imputer.fit_transform(df)`

B. `df.dropna()`

C. Predict for the missing values

Duplicates:

`Df.duplicated(keep='first')`

2. Train Test Split:

from `sklearn.model_selection` import `train_test_split`

`X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)`

3. Handling Categorical Variables:

Convert that into integers.

↓
Encode them.

Regression cannot take
Categorical values.

outliers →

Mean

Z-score

3

X ₁	X ₂	Y
✓	⊖	✓
⊖	⊖	⊖
⊖	⊖	⊖
⊖	⊖	⊖

Ja

Jump

Dec

missing

S.No.	Name	Marks	Height
	Ben	90	5'11"
	Sam	68	5'10"
	John	82	5'5"

1. Label Encoding – convert cat variables into numerical labels.

Eg : (India -1, China -0, US -2, UK - 4, Italy – 3)

Drawback: it gives the highest priority to any one category due to its label is high.

From sklearn.preprocessing import LabelEncoder

Le = LabelEncoder()

Le.fit(catDf['Country'])

catDf.Country=Le.transform(catDf.Country)

print(catDf)

2. One Hot Encoding: .

	Country	GDP	Continent	Area.
	Ind	2.68	Asia	<input type="checkbox"/>
	China	10.58	Asia	<input type="checkbox"/>
	US	15.48	North America	<input type="checkbox"/>

GDP	Area	Country_china	Country_India	Country_US	Continent_Africa	Continent_NA
2.68	<input type="checkbox"/>	0	1	0	1	0
10.58	<input type="checkbox"/>	1	0	0	1	0
15.48	<input type="checkbox"/>	0	0	1	0	1

1st Method:

`Pd.get_dummies(data=catDf)`

2nd Method:

```
From sklearn.preprocessing import OneHotEncoder  
ohe = OneHotEncoder()  
df1 = pd.DataFrame(ohe.fit_transform(catDf.iloc[:, [0, 3]]))  
pd.concat([catDf, df1], axis = 1)
```

Normalize/Standardize the data

Common scale.

$$Z = \frac{x - \mu}{\sigma}$$

(Standardizing)

→ No matter what was the individual value.
Mean & S.D.

ND
↓
SND

(0, 1, 1)
mean, sd, var

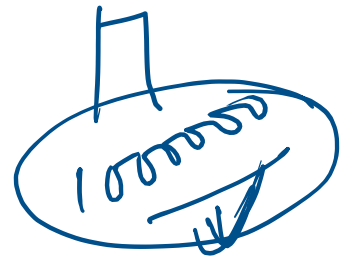
0 ✓

Z

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

How many std. away
any pt. is from mean.

Algorithm



0.000001
✓

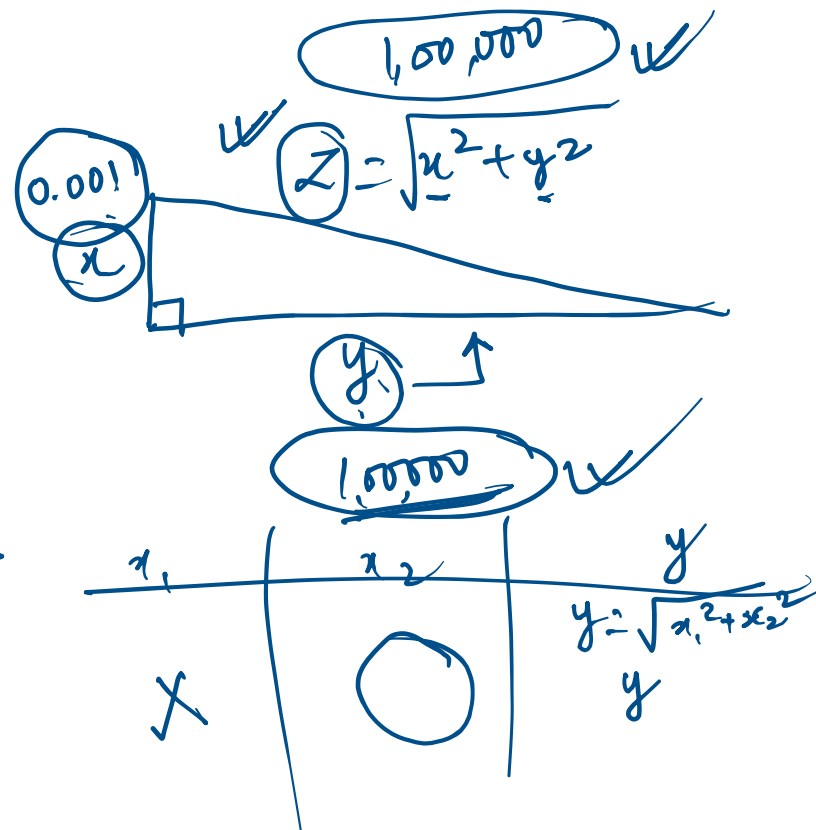
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

Min-Max Scalar Normalization:

scale to a fixed Range
(0, 1)

$$X_{norm} = \frac{X - X_{min}}{(X_{max} - X_{min})}$$

↓



```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)
```

>>>>>>EOF