Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

Amazon Web Services

MLOps with AWS

Masterclass

aws

Machine Learning Operations with AWS

Day -10

Amazon Web Services

MLOps with AWS

Masterclass

Pantech e Learning

DIGITAL LEARNING SIMPLIFIED

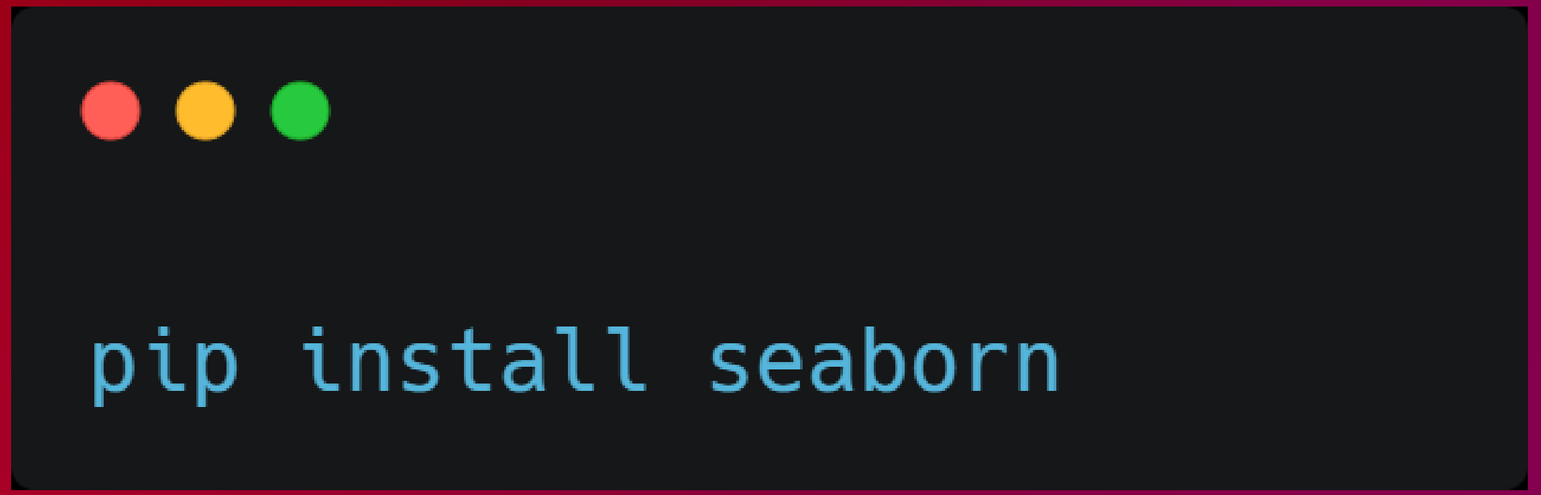
Data Visualization Library

Seaborn

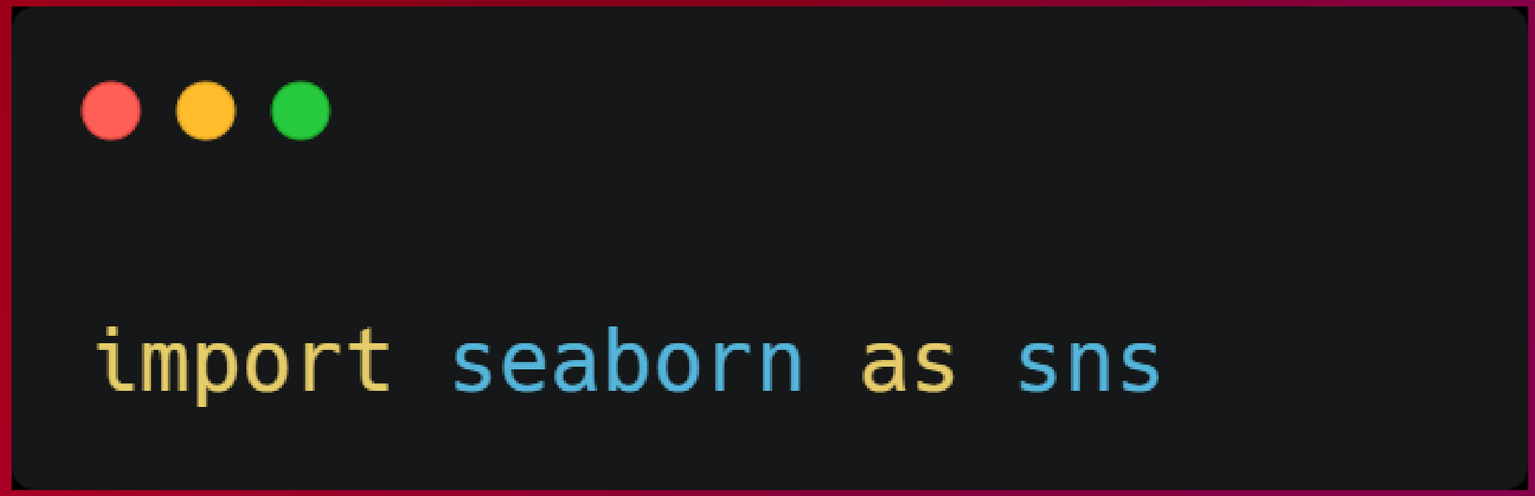
Seaborn

- Seaborn is a popular Python data visualization library that is built on top of the matplotlib library.
- Seaborn offers a variety of functions to generate different types of visualizations, such as scatter plots, bar plots, heatmaps, violin plots, and many others.
- It provides a high-level interface for creating informative and visually attractive statistical graphics.

Installation



Import



Scatter plot

```
sns.scatterplot(x, y, data)
```

Scatter plot

```
sns.scatterplot(x, y, data, hue, size)
```

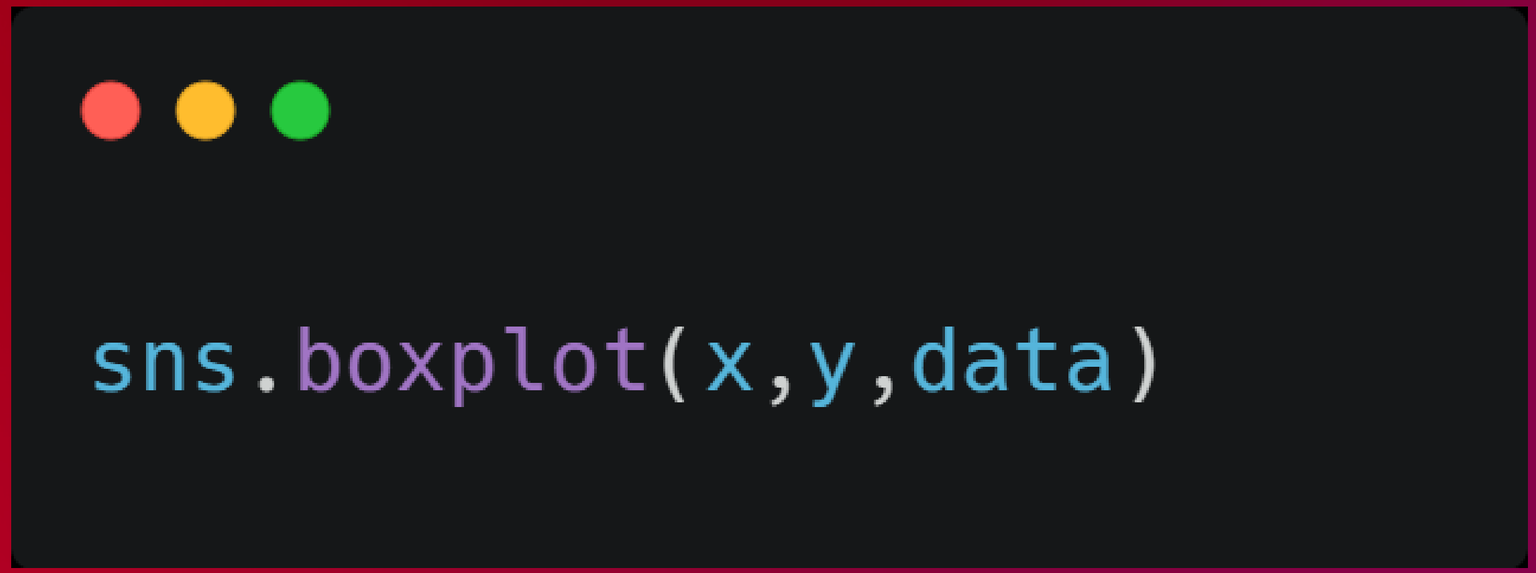
Bar plot

```
sns.barplot(x,y,hue,data,palette)
```

Count plot

```
sns.countplot(x, data)
```

Box plot



Pair plot

```
sns.pairplot(data,hue)
```

Heat maps

```
dataset.corr()

sns.heatmap(corr, cbar, linewidth, annot, fmt, cmap)
```

Scikit-Learn

Scikit-Learn

- Scikit-learn, also known as sklearn, is a popular open-source machine learning library in Python that provides a wide range of tools for data analysis, modeling, and evaluation.

- Sklearn is built on top of NumPy, SciPy, and Matplotlib, and supports integration with Pandas, which makes it easy to use in data science workflows.

- Sklearn is widely used in the data science community for various applications such as predictive modeling, natural language processing, computer vision, and time series forecasting, among others.

Installation

Import

```
from sklearn import
```

Preprocessing

- Feature Scaling
- Encoding
- Imputing null values
- Outlier - detection & Handling

Feature-Scaling

- Feature scaling is a method used to normalize the range of features of data.
- Feature Scaling involves modifying values by methods like Normalization or Standardization.
- It helps to avoid bias in machine learning model.

Why Scaling ?

- When dataset has numerical features and each of them are in different scale.
- ML model can put weight on features with larger scale.
- Scaling helps to contribute all features equally.

Age	Weight	Length
2 Years	26.5 lb. (12.02 kg)	33.7" (85.5 cm)
3 Years	31.5 lb. (14.29 kg)	37.0" (94 cm)
4 Years	34.0 lb. (15.42 kg)	39.5" (100.3 cm)
5 Years	39.5 lb. (17.92 kg)	42.5" (107.9 cm)
6 Years	44.0 lb. (19.96 kg)	45.5" (115.5 cm)
7 Years	49.5 lb. (22.45 kg)	47.7" (121.1 cm)
8 Years	57.0 lb. (25.85 kg)	50.5" (128.2 cm)
9 Years	62.0 lb. (28.12 kg)	52.5" (133.3 cm)
10 Years	70.5 lb. (31.98 kg)	54.5" (138.4 cm)
11 Years	81.5 lb. (36.97 kg)	56.7" (144 cm)
12 Years	91.5 lb. (41.5 kg)	59.0" (149.8 cm)

Normalization

It is the method of scaling the data by fitting the data points between a range of 0 to 1.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

MinMaxScaler

MinMaxScaler from sklearn perform normalization

```
from sklearn.preprocessing import MinMaxScaler


scaler = MinMaxScaler()


scaler.fit_transform(data)
```

Standardization

This converts all the data points

to have a mean value of 0 and

standard deviation of 1

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

μ = Mean

σ = Standard Deviation

Standard Scaler

StandardScaler from sklearn perform standardization

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()



scaler.fit_transform(data)
```

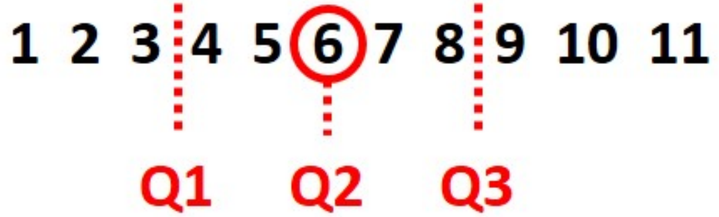
Robust Scaler

This uses interquartile range so

that it is robust to outliers

$$X_{new} = \frac{X - X_{median}}{IQR}$$

$$IQR = Q3 - Q1$$
$$= 8.5 - 3.5$$
$$= 5$$



Robust Scaler

```
from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()


scaler.fit_transform(data)
```

Use Case

Normalization:

- Useful when the data doesn't follow gaussian(normal) distrubution
- Useful in algorithms like KNN, and Neural networks like CNN, ANN

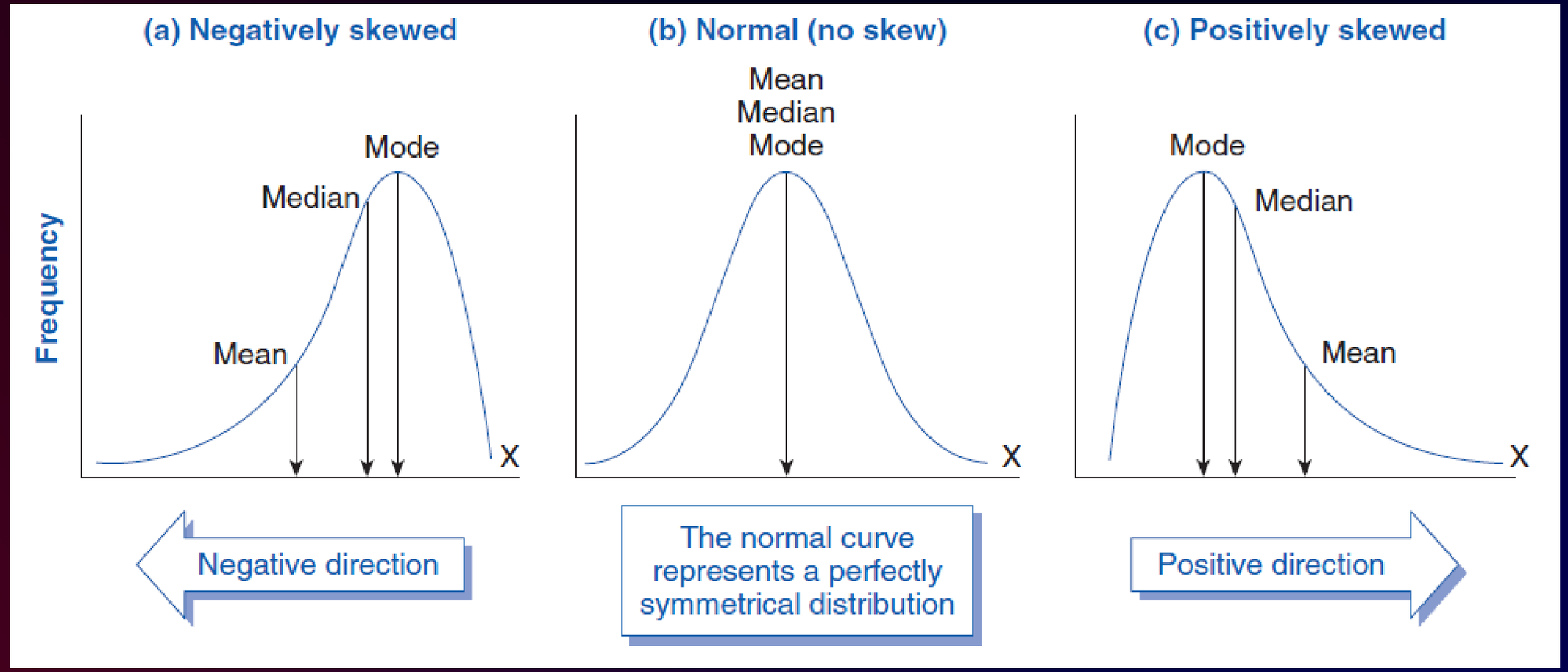
Standardization:

- When your data follows gaussian distribution

Robust Scaler:

- When your data has outliers

Data Distribution



Thank you