Day 5: Data Visualization with Matplotlib & Seaborn

Objective

The goal of this day is to learn how to **visualize data** — a crucial skill in Data Science. Good visualizations help:

- Understand patterns and trends
- Detect outliers
- Present data clearly to stakeholders
- Decide how to preprocess features for ML models

Topics Explained

1. What is Matplotlib?

Matplotlib is the most popular and powerful Python plotting library.

- pyplot module mimics MATLAB's plotting functions.
- You can create simple to complex static, animated, and interactive plots.

Basic Plot Types:

- line plot
- bar chart
- scatter plot
- histogram

2. What is Seaborn?

Seaborn is built on top of Matplotlib and makes beautiful, complex plots easy.

- Handles pandas DataFrames directly
- · Adds statistical insights visually

Special Features:

- Categorical plots (barplot, boxplot, violinplot)
- Regression plots (regplot, lmplot)
- Heatmaps & correlation plots

Setup

Install if not already installed # !pip install matplotlib seaborn

import pandas as pd import matplotlib.pyplot as plt import seaborn as sns

Dataset: Titanic (Same as Day 4)

df = sns.load_dataset('titanic')
df.head()

III Examples and Use Cases

1. Bar Plot – Count of Male vs Female

sns.countplot(x='sex', data=df)
plt.title("Passenger Gender Distribution")
plt.show()

2. Histogram – Age Distribution

sns.histplot(df['age'].dropna(), kde=True, bins=30)
plt.title("Age Distribution of Passengers")
plt.xlabel("Age")
plt.show()

3. Box Plot – Age vs Class

sns.boxplot(x='class', y='age', data=df)
plt.title("Age Distribution per Passenger Class")
plt.show()

4. Scatter Plot – Fare vs Age

sns.scatterplot(x='age', y='fare', data=df, hue='survived')
plt.title("Fare Paid vs Passenger Age (Survival Color-coded)")
plt.show()

5. Correlation Heatmap

numeric_df = df.select_dtypes(include=['float64', 'int64'])
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()

6. Pie Chart (Matplotlib) – Survival Percentage

survived_counts = df['survived'].value_counts()
labels = ['Not Survived', 'Survived']
plt.pie(survived_counts, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title("Survival Distribution")
plt.show()

Game-Based Learning Activity

@ "Guess the Plot" Game

Ask interns to guess which plot type is best for questions like:

- "What's the most common passenger class?"
- "Did age affect survival?"
- "Do males/females survive more?"

Interns then:

- 1. Write down their guess
- 2. Try visualizing it using seaborn/matplotlib
- 3. Discuss which plot gave the clearest insight

Mini Challenge: Build a Titanic Dashboard

- Combine **3–4 plots** into one figure using plt.subplot
- Tell a **story** visually (e.g., "Who survived the Titanic?")

Exploratory Data Analysis (EDA)

Objective

EDA is all about:

- Understanding the **structure**, **patterns**, and **anomalies** in your dataset
- Identifying relationships between variables
- Preparing for model building with domain insights
 - EDA is what separates good Data Scientists from average ones.

📚 Topics Covered

- 1. Types of Variables: Numerical vs Categorical
- 2. **Univariate Analysis**: One variable at a time (distributions, counts)
- 3. **Bivariate Analysis**: Two variables (correlations, relationships)
- 4. Outlier Detection: Using boxplots and IQR

- 5. Feature Insights: Detecting skewness, imbalance
- 6. Statistical Summaries

X Setup and Dataset

We will continue using the **Titanic** dataset:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

df = sns.load_dataset("titanic")

df.head()



Step-by-Step Guide with Code

1. Understand Data Types

df.info()

df.dtypes.value_counts()

• 2. Univariate Analysis

➤ Categorical Variables

sns.countplot(x='sex', data=df)

plt.title("Count of Genders")

plt.show()

➤ Numerical Variables

```
sns.histplot(df['age'].dropna(), bins=30, kde=True)
plt.title("Distribution of Age")
plt.show()
```

• 3. Bivariate Analysis

➤ Survival by Gender

```
sns.countplot(x='sex', hue='survived', data=df)
plt.title("Survival Count by Gender")
plt.show()
```

➤ Age vs Fare

```
sns.scatterplot(x='age', y='fare', hue='survived', data=df)
plt.title("Fare vs Age Colored by Survival")
plt.show()
```

4. Correlation Analysis

```
numeric = df.select_dtypes(include='number')
corr = numeric.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

5. Outlier Detection with Boxplot

```
sns.boxplot(y='fare', data=df)
plt.title("Boxplot of Fare (Check for Outliers)")
plt.show()
```

• 6. Missing Data Analysis

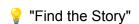
```
df.isnull().sum().sort_values(ascending=False)
sns.heatmap(df.isnull(), cbar=False, cmap="YIGnBu")
plt.title("Missing Value Heatmap")
plt.show()
```

7. Skewness Check

from scipy.stats import skew

```
numeric_cols = df.select_dtypes(include='number')
skew_vals = numeric_cols.apply(lambda x: skew(x.dropna()))
print(skew_vals)
```

Mini EDA Game Challenge



Give interns tasks like:

- Which group had the highest survival rate?
- Are older people more likely to survive?
- What is the relationship between class, fare, and survival?

Let them visualize to answer these questions using plots.