Lesson 6: Exploratory Data Analysis (EDA)

Welcome to Lesson 6, where we dive into **Exploratory Data Analysis (EDA)**, an essential step in the machine learning process. EDA helps us understand our data better, uncover patterns, and identify relationships between variables before building any models.

Topics Covered

1. **What is EDA and Why is it Important?**
2. **Techniques in EDA:**
   - Descriptive statistics.
   - Visualizing data with plots (scatter plots, histograms, box plots, and heatmaps).
   - Correlation analysis.
3. **Activities:**
   - Perform EDA on a simple dataset.
   - Identify relationships between features and target variables.

---

1. What is EDA and Why is it Important?

Exploratory Data Analysis (EDA) is the process of analyzing datasets to:

- Summarize their main characteristics.
- Find patterns and relationships in the data.
- Detect missing data, outliers, or anomalies.
- Ensure that the data is ready for preprocessing and modeling.

EDA helps us answer important questions like:

- What is the range of values in the dataset?
- Are there any correlations between features?
- What kind of distributions do features have?

---

2. Techniques in EDA

Descriptive Statistics

Descriptive statistics provide a summary of the data, including:

- **Mean:** Average value.
- **Median:** Middle value.
- **Standard Deviation:** How spread out the data is.

Let’s explore with an example:

Example Dataset: Student Exam Scores

Name	Math	Science	English
Alice	85	90	88
Bob	75	80	70
Charlie	95	85	92
Diana	70	75	65

Code: Descriptive Statistics

```
import pandas as pd

# Dataset
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Diana'],
        'Math': [85, 75, 95, 70],
        'Science': [90, 80, 85, 75],
        'English': [88, 70, 92, 65]}
df = pd.DataFrame(data)

# Summary statistics
print(df.describe())
```

Explanation:

1. `describe()` gives you key statistics like mean, median, min, max, and standard deviation for each column.
2. Look for any unusually high or low values.

Distributions

A **distribution** shows how data is spread across different values. Common visualizations include histograms and box plots.

Code: Visualizing Distributions

```
import matplotlib.pyplot as plt

# Plot histograms
df[['Math', 'Science', 'English']].hist(bins=5, figsize=(8, 4))
plt.show()
```

Explanation:

- Histograms show the frequency of values for each subject.
- Look for patterns like normal distribution or skewed data.

Scatter Plots

Scatter plots show relationships between two variables.

Code: Scatter Plot Example

```
# Scatter plot between Math and Science scores
plt.scatter(df['Math'], df['Science'], color='blue')
plt.title('Math vs Science Scores')
plt.xlabel('Math Scores')
plt.ylabel('Science Scores')
plt.show()
```

Explanation:

- Points close together suggest a strong relationship.
- No clear pattern might indicate no relationship.

### Heatmaps

A **heatmap** visualizes correlations between multiple features.

Code: Correlation Heatmap

```
import seaborn as sns

# Correlation matrix
corr_matrix = df[['Math', 'Science', 'English']].corr()

# Heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

Explanation:

- Values closer to 1 indicate strong positive correlation.
- Values closer to -1 indicate strong negative correlation.

Visualizing Missing Data

```
import seaborn as sns
import numpy as np

# Titanic dataset
data = {
    'Passenger': [1, 2, 3, 4, 5],
    'Age': [22, 38, 26, 35, np.nan],
    'Gender': ['Male', 'Female', 'Female', 'Male', 'Female'],
    'Survived': ['Yes', 'Yes', 'No', 'No', 'Yes'],
    'Fare': [7.25, 71.28, np.nan, 8.05, 53.10]
}
titanic_df = pd.DataFrame(data)

# Heatmap of missing data
sns.heatmap(titanic_df.isnull(), cbar=False, cmap="viridis")
plt.title("Missing Data Heatmap")
plt.show()
```

Explanation:

- The heatmap clearly shows where data is missing.
- Yellow lines represent missing values in the dataset.

3. Activities: Perform EDA on a Dataset

Dataset: Car Prices

Car	Price	Engine Size (L)	Mileage (MPG)
Car A	20000	2.0	30
Car B	25000	2.5	28
Car C	18000	1.8	35
Car D	22000	2.2	32

Activity 1: Perform Descriptive Statistics

```
# Car Price dataset
car_data = {'Car': ['Car A', 'Car B', 'Car C', 'Car D'],
            'Price': [20000, 25000, 18000, 22000],
            'Engine_Size': [2.0, 2.5, 1.8, 2.2],
            'Mileage': [30, 28, 35, 32]}
car_df = pd.DataFrame(car_data)

# Descriptive statistics
print(car_df.describe())
```

Explanation:

1. The mean price helps identify the average car price.
2. Standard deviation shows how prices vary.

Activity 2: Analyze Relationships

```
# Scatter plot: Engine Size vs Price
plt.scatter(car_df['Engine_Size'], car_df['Price'], color='green')
plt.title('Engine Size vs Price')
plt.xlabel('Engine Size (L)')
plt.ylabel('Price ($)')
plt.show()
```

Explanation:

- A positive slope suggests that larger engine sizes lead to higher prices.

Activity 3: Correlation Analysis

```
# Correlation heatmap
car_corr_matrix = car_df[['Price', 'Engine_Size', 'Mileage']].corr()
sns.heatmap(car_corr_matrix, annot=True, cmap='coolwarm')
plt.title('Car Dataset Correlation Heatmap')
plt.show()
```

Explanation:

- Negative correlation between Mileage and Price suggests that cars with higher mileage are cheaper.

---

4. Identifying Relationships Between Features and Target Variables

Example: Titanic Dataset

Passenger	Age	Gender	Survived
1	22	Male	No
2	38	Female	Yes
3	26	Female	Yes
4	35	Male	No

Code: Analyze Relationships

```
# Titanic dataset
titanic_data = {'Passenger': [1, 2, 3, 4],
                'Age': [22, 38, 26, 35],
                'Gender': ['Male', 'Female', 'Female', 'Male'],
                'Survived': ['No', 'Yes', 'Yes', 'No']}
titanic_df = pd.DataFrame(titanic_data)

# Encoding Gender
titanic_df['Gender_Encoded'] = label_encoder.fit_transform(titanic_df['Gender'])

# Scatter plot: Age vs Survived
sns.scatterplot(data=titanic_df, x='Age', y='Gender_Encoded', hue='Survived',
style='Survived')
plt.title('Age vs Gender with Survival')
plt.xlabel('Age')
plt.ylabel('Gender (0 = Female, 1 = Male)')
plt.show()
```

Explanation:

- This plot helps visualize survival patterns based on age and gender.

Conclusion

In this session, we explored:

- Descriptive statistics to summarize data.
- Visualizations (scatter plots, heatmaps) to identify relationships.
- Correlation analysis to understand feature relationships.

EDA is a critical step in understanding your data before modeling. In the next session, we'll delve into feature engineering and selection!

1. Analyzing Student Grades

Dataset: Student Grades

Name	Math	Science	English
Alice	85	90	88
Bob	75	80	70
Charlie	95	85	92
Diana	70	75	65

Code: Descriptive Statistics

```
import pandas as pd

# Dataset
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'Diana'],
    'Math': [85, 75, 95, 70],
    'Science': [90, 80, 85, 75],
    'English': [88, 70, 92, 65]
}
df = pd.DataFrame(data)

# Descriptive statistics
print("Summary Statistics:")
print(df.describe())
```

Explanation:

1. `describe()` provides basic descriptive statistics such as count, mean, and standard deviation for numerical columns.
2. Use this to quickly summarize key metrics.

Code: Correlation Analysis

```
# Correlation between subjects
corr_matrix = df[['Math', 'Science', 'English']].corr()

# Print correlation
print("Correlation Matrix:")
print(corr_matrix)
```

Explanation:

- Correlation values range from -1 to 1:
  - Close to 1: Strong positive correlation.
  - Close to -1: Strong negative correlation.
  - Close to 0: No correlation.

Code: Visualizing Grade Distributions

```
import matplotlib.pyplot as plt

# Histogram of grades
df[['Math', 'Science', 'English']].hist(bins=5, figsize=(8, 4))
plt.suptitle("Grade Distributions")
plt.show()
```

Explanation:

- Histograms show how grades are distributed across students.
- Peaks indicate common grade ranges.

---

2. Analyzing a Car Dataset

Dataset: Car Prices

Car	Price	Mileage (MPG)	Engine Size (L)
Car A	20000	30	2.0
Car B	25000	28	2.5
Car C	18000	35	1.8
Car D	22000	32	2.2

Code: Scatter Plot (Price vs Mileage)

```
# Scatter plot for price and mileage
plt.scatter(df['Price'], df['Mileage'], color='blue')
plt.title('Price vs Mileage')
plt.xlabel('Price ($)')
plt.ylabel('Mileage (MPG)')
plt.show()
```

Explanation:

- Scatter plots help visualize trends. For example, cars with higher mileage may have lower prices.

---

3. Visualizing Missing Data in Titanic Dataset

Dataset: Titanic

Passenger	Age	Gender	Survived	Fare
1	22	Male	Yes	7.25
2	38	Female	Yes	71.28
3	26	Female	No	NaN
4	35	Male	No	8.05
5	NaN	Female	Yes	53.10

Code: Visualizing Missing Data

```
import seaborn as sns
import numpy as np

# Titanic dataset
data = {
    'Passenger': [1, 2, 3, 4, 5],
    'Age': [22, 38, 26, 35, np.nan],
    'Gender': ['Male', 'Female', 'Female', 'Male', 'Female'],
    'Survived': ['Yes', 'Yes', 'No', 'No', 'Yes'],
    'Fare': [7.25, 71.28, np.nan, 8.05, 53.10]
}
titanic_df = pd.DataFrame(data)

# Heatmap of missing data
sns.heatmap(titanic_df.isnull(), cbar=False, cmap="viridis")
```

```
plt.title("Missing Data Heatmap")
plt.show()
```

Explanation:

- The heatmap clearly shows where data is missing.
- Yellow lines represent missing values in the dataset.

Code: Filling Missing Values

```
# Fill missing Age with mean
titanic_df['Age'].fillna(titanic_df['Age'].mean(), inplace=True)

# Fill missing Fare with median
titanic_df['Fare'].fillna(titanic_df['Fare'].median(), inplace=True)

print("Updated Titanic Dataset:")
print(titanic_df)
```

Explanation:

- Filling missing values with the mean or median prevents losing rows with incomplete data.

---

4. Correlation Analysis with Heatmaps

Dataset: House Prices

House	Area (sq ft)	Bedrooms	Price ($)
House A	1200	3	200000
House B	1500	4	250000
House C	800	2	120000
House D	2000	5	300000

Code: Heatmap for Correlation

```
# House dataset
house_data = {
    'House': ['A', 'B', 'C', 'D'],
    'Area': [1200, 1500, 800, 2000],
    'Bedrooms': [3, 4, 2, 5],
    'Price': [200000, 250000, 120000, 300000]
}
house_df = pd.DataFrame(house_data)

# Correlation heatmap
sns.heatmap(house_df.corr(), annot=True, cmap='coolwarm')
plt.title('House Price Correlation Heatmap')
plt.show()
```

Explanation:

- Strong correlation between area and price suggests that larger houses tend to be more expensive.

---

5. Pairplot Analysis

Dataset: Iris (Preloaded in Seaborn)

Iris dataset includes measurements of flowers and their species.

Code: Pairplot

```
# Load Iris dataset
from seaborn import load_dataset
iris = load_dataset('iris')

# Pairplot
sns.pairplot(iris, hue='species')
plt.show()
```

Explanation:

- Pairplots display relationships between all numeric columns.
- Different colors distinguish between species.

---

6. Box Plot to Identify Outliers

Dataset: Car Prices

Car	Price	Mileage (MPG)	Engine Size (L)
Car A	20000	30	2.0
Car B	25000	28	2.5
Car C	18000	35	1.8
Car D	22000	32	2.2
Car E	40000	20	4.5

Code: Box Plot Example

```
# Box plot for Price
sns.boxplot(x=car_df['Price'])
plt.title("Car Price Box Plot")
plt.show()
```

Explanation:

- Outliers are points that lie far outside the box.
- For example, Car E might be flagged as an outlier.