

School of Computer Science, UPES, Dehradun.

Elements of AI ML

Lab File

B.TECH. -III Semester

**Submitted to:**

Ms. Pooja Sarin

**Submitted by: Amanjot Singh Saini**

Batch: 35

SAP ID: 500120674 Roll No.: R214223117

Expirement 4

Write a python program to import and export data using Pandas and show the details of the dataset like number of rows, columns, first five rows, size, number of missing values, sum, average, min and max values from the numerical columns.

Code Snippet :

import pandas as pd

df = pd.read\_csv("C:/Users/Amanjot Singh/Downloads/Uber-Jan-Feb-FOIL.csv")

print("Number of rows:", df.shape[0])

print("Number of columns:", df.shape[1])

print("\nFirst five rows of the dataset:")

print(df.head())

print("\nDataset size (number of elements):", df.size)

print("\nNumber of missing values in each column:")

print(df.isnull().sum())

print("\nSum of numerical columns:")

print(df.select\_dtypes(include=['number']).sum())

print("\nAverage (mean) of numerical columns:")

print(df.select\_dtypes(include=['number']).mean())

print("\nMinimum values of numerical columns:")

print(df.select\_dtypes(include=['number']).min())

print("\nMaximum values of numerical columns:")

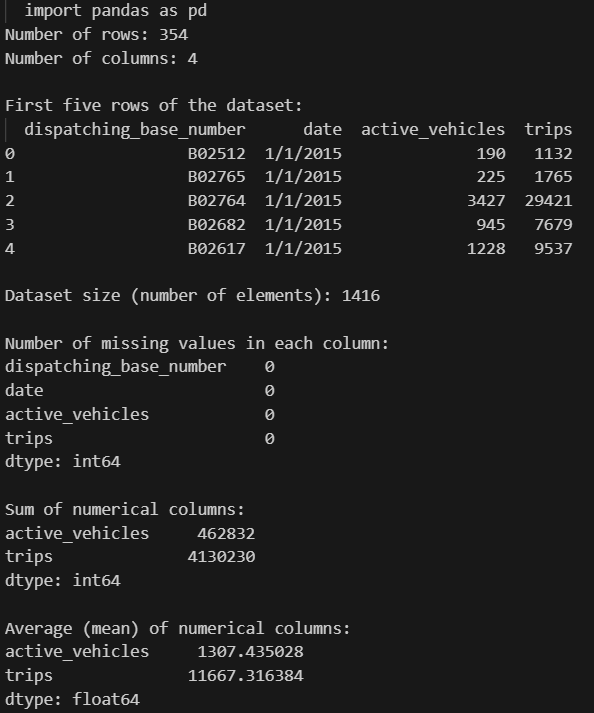
print(df.select\_dtypes(include=['number']).max())

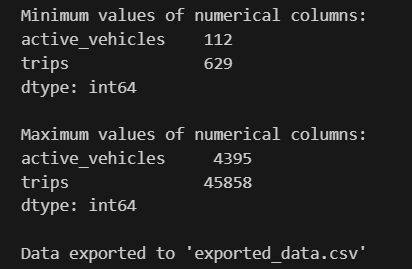
df.to\_csv("exported\_data.csv", index=False)

print("\nData exported to 'exported\_data.csv'")

Output:

Output:





Experiment 5

Using Python language do the exploratory data analysis of dataset imported in the lab 4.

Code snippet:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

df = pd.read\_csv("C:/Users/Amanjot Singh/Downloads/Uber-Jan-Feb-FOIL.csv")

# Basic information about the dataset

print("Dataset Information:")

df.info()

print("\nSummary Statistics:")

print(df.describe())

# Display first five rows

print("\nFirst five rows:")

print(df.head())

# Check for missing values

print("\nMissing Values:")

print(df.isnull().sum())

# Data distribution and summary for numerical columns

numerical\_cols = df.select\_dtypes(include=['float64', 'int64']).columns

print("\nStatistical Summary for Numerical Columns:")

print(df[numerical\_cols].describe())

# Count of unique values for categorical columns (if any)

categorical\_cols = df.select\_dtypes(include=['object']).columns

print("\nUnique Values in Categorical Columns:")

for col in categorical\_cols:

    print(f"{col}: {df[col].nunique()} unique values")

# Visualize distributions of numerical columns

print("\nVisualizing Distributions of Numerical Columns")

for col in numerical\_cols:

    plt.figure(figsize=(6, 4))

    sns.histplot(df[col].dropna(), kde=True)

    plt.title(f"Distribution of {col}")

    plt.show()

# Visualize missing values heatmap

plt.figure(figsize=(10, 6))

sns.heatmap(df.isnull(), cbar=False, cmap="viridis")

plt.title("Missing Values Heatmap")

plt.show()

# Plotting correlations

plt.figure(figsize=(10, 8))

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm", square=True)

plt.title("Correlation Matrix")

plt.show()

# Pair plot for pairwise relationships in a dataset

print("\nVisualizing Pairwise Relationships")

sns.pairplot(df[numerical\_cols].dropna())

plt.show()

# Exporting the cleaned data if any cleaning was done (optional)

df\_cleaned = df.dropna()  # Example cleaning step if needed

df\_cleaned.to\_csv("cleaned\_data.csv", index=False)

print("\nCleaned data exported to 'cleaned\_data.csv'")

# Data distribution and summary for numerical columns

numerical\_cols = df.select\_dtypes(include=['float64', 'int64']).columns

print("\nStatistical Summary for Numerical Columns:")

print(df[numerical\_cols].describe())

# Count of unique values for categorical columns (if any)

categorical\_cols = df.select\_dtypes(include=['object']).columns

print("\nUnique Values in Categorical Columns:")

for col in categorical\_cols:

    print(f"{col}: {df[col].nunique()} unique values")

# Visualize distributions of numerical columns

print("\nVisualizing Distributions of Numerical Columns")

for col in numerical\_cols:

    plt.figure(figsize=(6, 4))

    sns.histplot(df[col].dropna(), kde=True)

    plt.title(f"Distribution of {col}")

    plt.show()

# Visualize missing values heatmap

plt.figure(figsize=(10, 6))

sns.heatmap(df.isnull(), cbar=False, cmap="viridis")

plt.title("Missing Values Heatmap")

plt.show()

# Plotting correlations

plt.figure(figsize=(10, 8))

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm", square=True)

plt.title("Correlation Matrix")

plt.show()

# Pair plot for pairwise relationships in a dataset

print("\nVisualizing Pairwise Relationships")

sns.pairplot(df[numerical\_cols].dropna())

plt.show()

# Exporting the cleaned data if any cleaning was done (optional)

df\_cleaned = df.dropna()  # Example cleaning step if needed

df\_cleaned.to\_csv("cleaned\_data.csv", index=False)

print("\nCleaned data exported to 'cleaned\_data.csv'")

# Pair plot for pairwise relationships in a dataset

print("\nVisualizing Pairwise Relationships")

sns.pairplot(df[numerical\_cols].dropna())

plt.show()

# Exporting the cleaned data if any cleaning was done (optional)

df\_cleaned = df.dropna()  # Example cleaning step if needed

df\_cleaned.to\_csv("cleaned\_data.csv", index=False)

print("\nCleaned data exported to 'cleaned\_data.csv'")

Output:

