**Step-by-Step Guide**

1. **Set up the Google Colab environment**:
   * Open Google Colab: Google Colab
   * Create a new notebook by selecting File → New Notebook.
2. **Upload the CSV file**:
   * Before running the code, you'll need to upload the uber\_rides.csv file to Colab.
     + In Colab, click on the file icon in the left sidebar.
     + Click the "Upload" button and select your CSV file.

**Cell 1: Import Libraries and Load Data**

python

Copy code

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Load the CSV file

df = pd.read\_csv("/content/uber\_rides.csv")

# View the first few rows of the DataFrame

df.head()

**Cell 2: Check Data Info**

python

Copy code

# Check the info of the DataFrame (data types, non-null counts)

df.info()

**Cell 3: Check Shape of the DataFrame**

python

Copy code

# View the shape (rows and columns) of the DataFrame

df.shape

**Cell 4: Check for Missing Values**

python

Copy code

# Check for missing values in each column

df.isnull().sum()

**Cell 5: Remove Missing Values**

python

Copy code

# Drop rows with missing values

df.dropna(inplace=True)

# Check again for missing values after drop

df.isnull().sum()

**Cell 6: Drop Unwanted Columns**

python

Copy code

# Drop unnecessary columns like 'Unnamed: 0' and 'key'

df.drop(labels='Unnamed: 0', axis=1, inplace=True)

df.drop(labels='key', axis=1, inplace=True)

# View the DataFrame after dropping columns

df.head()

**Cell 7: Convert 'pickup\_datetime' to Datetime**

python

Copy code

# Convert 'pickup\_datetime' to datetime format

df["pickup\_datetime"] = pd.to\_datetime(df["pickup\_datetime"])

# Check the data types after conversion

df.dtypes

**Cell 8: Get Descriptive Statistics**

python

Copy code

# View summary statistics for numerical columns

df.describe()

**Cell 9: Plot Distribution of Fare Amount**

python

Copy code

# Plot the distribution of the 'fare\_amount' column

sns.distplot(df['fare\_amount'])

plt.show()

**Cell 10: Plot Distribution of Pickup Latitude**

python

Copy code

# Plot the distribution of 'pickup\_latitude'

sns.distplot(df['pickup\_latitude'])

plt.show()

**Cell 11: Plot Distribution of Pickup Longitude**

python

Copy code

# Plot the distribution of 'pickup\_longitude'

sns.distplot(df['pickup\_longitude'])

plt.show()

**Cell 12: Plot Distribution of Dropoff Longitude**

python

Copy code

# Plot the distribution of 'dropoff\_longitude'

sns.distplot(df['dropoff\_longitude'])

plt.show()

**Cell 13: Plot Distribution of Dropoff Latitude**

python

Copy code

# Plot the distribution of 'dropoff\_latitude'

sns.distplot(df['dropoff\_latitude'])

plt.show()

**Cell 14: Find Outliers using IQR**

python

Copy code

# Define a function to find outliers using IQR (Interquartile Range)

def find\_outliers\_IQR(df):

q1 = df.quantile(0.25)

q3 = df.quantile(0.75)

IQR = q3 - q1

outliers = df[((df < (q1 - 1.5 \* IQR)) | (df > (q3 + 1.5 \* IQR)))]

return outliers

# Find outliers in the 'fare\_amount' column

outliers = find\_outliers\_IQR(df["fare\_amount"])

# Display the number and max/min outliers

print("Number of outliers: " + str(len(outliers)))

print("Max outlier value: " + str(outliers.max()))

print("Min outlier value: " + str(outliers.min()))

outliers

**Cell 15: Find Outliers in Multiple Columns**

python

Copy code

# Find outliers in 'passenger\_count' and 'fare\_amount' columns

outliers = find\_outliers\_IQR(df[["passenger\_count", "fare\_amount"]])

outliers

**Cell 16: Calculate Fare Amount Limits**

python

Copy code

# Calculate upper and lower limits using mean + 3 \* std for 'fare\_amount'

upper\_limit = df['fare\_amount'].mean() + 3 \* df['fare\_amount'].std()

lower\_limit = df['fare\_amount'].mean() - 3 \* df['fare\_amount'].std()

# Display the limits

print(f"Upper limit: {upper\_limit}")

print(f"Lower limit: {lower\_limit}")

**Cell 17: Correlation Heatmap**

python

Copy code

# Create a correlation heatmap of the DataFrame

corrMatrix = df.corr()

sns.heatmap(corrMatrix, annot=True)

plt.show()

**Cell 18: Extract Features from Pickup Datetime**

python

Copy code

import calendar

# Extract day, hour, month, year, and weekday from 'pickup\_datetime'

if 'pickup\_datetime' in df.columns:

df['pickup\_datetime'] = pd.to\_datetime(df['pickup\_datetime'], errors='coerce')

df['day'] = df['pickup\_datetime'].dt.day

df['hour'] = df['pickup\_datetime'].dt.hour

df['month'] = df['pickup\_datetime'].dt.month

df['year'] = df['pickup\_datetime'].dt.year

df['weekday'] = df['pickup\_datetime'].dt.day\_name()

df.drop('pickup\_datetime', axis=1, inplace=True)

df['weekday'] = df['weekday'].map({

'Sunday': 0, 'Monday': 1, 'Tuesday': 2, 'Wednesday': 3,

'Thursday': 4, 'Friday': 5, 'Saturday': 6

})

else:

print("Column 'pickup\_datetime' not found in DataFrame.")

# View the first few rows of the DataFrame

df.head()

**Cell 19: Check DataFrame Info After Modifications**

python

Copy code

# Check the updated info of the DataFrame

df.info()

**Cell 20: Prepare Feature Set for Model**

python

Copy code

# Prepare feature set by dropping 'fare\_amount' column

X = df.drop("fare\_amount", axis=1)

X.head()

**Cell 21: Split Data into Training and Test Sets**

python

Copy code

from sklearn.model\_selection import train\_test\_split

# Prepare target variable (fare\_amount)

y = df["fare\_amount"]

# Split the data into training and testing sets (80/20 split)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

# View the first few rows of the training feature set (x\_train)

print(x\_train.head())

**Cell 22: View Test Set Features**

python

Copy code

# View the first few rows of the test feature set (x\_test)

print(x\_test.head())

**Cell 23: View Training Target**

python

Copy code

# View the first few values of the training target variable (y\_train)

print(y\_train.head())

**Cell 24: View Test Target**

python

Copy code

# View the first few values of the test target variable (y\_test)

print(y\_test.head())

**How to Run Each Cell**

* **Execute each cell one by one** by clicking on the **play button** to the left of each cell.
* After running the code in each cell, you'll see the output (such as DataFrame prints, plots, etc.) below the cell.