Code-1

This code performs a simple linear regression analysis to predict the price of cab rides based on the distance and cab type (Lyft or Uber). The code first imports the necessary libraries including pandas, numpy, matplotlib.pyplot, and the scikit-learn modules for data preprocessing, model building, and evaluation.

The code then loads the dataset ('cab\_rides.csv') into a pandas DataFrame and performs data cleaning by removing any missing or duplicate values. The relevant columns for analysis (distance, cab\_type, and price) are extracted and the categorical variable cab\_type is converted to numerical values (0 for Lyft and 1 for Uber).

The data is then split into training and testing sets using the train\_test\_split function from scikit-learn. A simple linear regression model is built using the LinearRegression function and fit to the training data.

The model's performance is evaluated on the testing set using mean squared error (MSE) and the root mean squared error (RMSE) is calculated and printed.

Finally, the code creates a scatter plot to visualize the pricing of Uber and Lyft for different distances. The data is split into two subsets based on cab type and plotted as separate scatter plots. The x-axis represents the distance and the y-axis represents the price. The scatter plot is labeled with a legend to distinguish between Uber and Lyft.

This code performs exploratory data analysis (EDA) and builds a machine learning model to predict the price of cab rides based on the distance, hour of the day, and day of the week.

Code-2

First, the necessary libraries including pandas, numpy, matplotlib.pyplot, seaborn, and scikit-learn modules are imported. The code then reads the dataset 'cab\_rides.csv' into a pandas DataFrame and displays the first 5 rows of the dataset.

Next, the data is cleaned by removing the rows with missing values and converting the timestamp columns to datetime format. The date\_time, hour, day, month, and day\_of\_week columns are created based on the time\_stamp column.

EDA is performed using various visualizations. A histogram is plotted to show the distribution of fare prices, a scatter plot is created to visualize the correlation between price and distance, and box plots are used to show the correlation between price and day of the week and hour of the day

A machine learning model is then built using the relevant features including distance, hour, and day\_of\_week columns. The data is split into training and testing sets using the train\_test\_split function. A linear regression model is trained on the training set using the LinearRegression function, and the model is used to predict the fare prices for the test set. The R-squared value is calculated using the score function of the linear regression model on the test set.

Overall, this code performs data cleaning, EDA, and a simple linear regression model to predict the price of cab rides based on the distance, hour of the day, and day of the week.

This code is an implementation of a simple graphical user interface (GUI) using the tkinter library in Python. The GUI is designed to allow users to select a source and destination from two dropdown menus and calculate the average prices of Lyft and Uber rides between the selected locations. It also displays the best price and the vehicle type that offers the best price.

Code-3

The code begins by importing the required libraries: tkinter and pandas. Then, the cab rides dataset is loaded into a pandas dataframe.

Next, the unique sources and destinations in the dataset are extracted and stored in two arrays: sources and destinations.

Then, a function called get\_prices() is defined. This function is called when the user clicks on the "Submit" button in the GUI. The function first gets the source and destination selected by the user from the dropdown menus. It then filters the dataset to include only rides with the selected source and destination. If the filtered dataframe is empty, an error message is displayed using the messagebox function of the tkinter library.

If the filtered dataframe is not empty, the function calculates the average price for each cab type (Lyft and Uber) using the groupby() function of pandas. It then determines the best cab type (Lyft or Uber) and its average price using the idxmin() and min() functions of pandas. Finally, it gets the name of the best vehicle from the filtered dataframe and displays the results in a message box using the messagebox function of tkinter.

After defining the get\_prices() function, the main window is created using the Tk() function of tkinter. The title of the window is set to "Ride prices".

Then, two dropdown menus are created for the source and destination selection. The options for the dropdown menus are set to the unique sources and destinations extracted earlier. The selected source and destination are stored in source\_var and destination\_var variables, respectively.

A "Submit" button is also created using the Button() function of tkinter. When the button is clicked, it calls the get\_prices() function.

Finally, the main event loop is started using the mainloop() function of tkinter, which waits for user input and updates the GUI as required.