109550198 卜銳凱

Introduction to Data Science

HW3

Report

This report is a combination of HW2 analysis and HW3 analysis, with HW3 focusing on data cleaning. This report describes the analysis of the dataset from the CSV file "2023 June Unemployment Rate by County (Percent).csv." To guarantee data quality and completeness, the dataset was loaded from a CSV file, and different data analysis and manipulation operations were performed.

**Code Summary:**

The provided code accomplishes the following tasks:

Data Loading: The code starts by importing necessary Python libraries such as NumPy, Matplotlib, Pandas, and Seaborn. These libraries are frequently employed in data analysis and visualization.

The dataset is loaded into a Pandas DataFrame named 'dataset' from the file '2023 June Unemployment Rate by County (Percent).csv'. This first step is critical for gaining access to and modifying the data.

A screenshot of a computer

Description automatically generated

**Data Splitting**:

It divides the dataset into two arrays, x, and y, where x contains all except the last column and y contains the last column. This is commonly done to differentiate between features (independent variables) and the desired variable (dependent variable).

A screenshot of a computer

Description automatically generated

**Data Exploration:**

It shows the first ten rows of the Data Frame, providing an overview of the data's structure and content.

A screenshot of a computer

Description automatically generated

Displays the last 5 rows using dataset.tail(5) to check the data's end.

A screenshot of a computer

Description automatically generated

Retrieves the column names using dataset.columns to understand the variables included in the dataset.

Generates summary statistics using dataset.describe(), which provides insights into the central tendencies and distributions of numerical columns.

Uses dataset.info() to obtain information about data types and non-null counts.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Missing Values Handling**:

Missing values are common in real-world datasets and handling them is critical to ensuring that subsequent analyses are accurate. To handle missing values, the code does the following:

Uses 'dataset.isnull().sum()' to check for missing values and reports the number of missing values in each column.

To highlight the pattern of missing values across columns, a heatmap built with Seaborn's sns.heatmap() is used to visualize the missing data.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Calculates and displays the percentage of missing values in each column using round(dataset.isnull().sum() / len(dataset) \* 100, 1).

A screenshot of a computer

Description automatically generated

**Data cleaning:**

The code snippet below demonstrates the location-based replacement of values in the 'Unnamed: 3' column:

A screenshot of a computer

Description automatically generated

At row index 92, the value in the 'Unnamed: 3' column was replaced with 7.51.

At row index 94, the value in the 'Unnamed: 3' column was replaced with 7.62.

At row index 1038, the value in the 'Unnamed: 3' column was replaced with 7.83.

At row index 2419, the value in the 'Unnamed: 3' column was replaced with 7.94.

At row index 2918, the value in the 'Unnamed: 3' column was replaced with 8.15.

Following the replacement, the changed dataset was printed using print(dataset) to display the updated values. To check the changes, the unique values in the 'Unnamed: 3' column were displayed with print(dataset['Unnamed: 3'].unique()).

The provided code snippet demonstrates the process of replacing missing values:

The .fillna(8.643, inplace=True) method is applied to the 'Unnamed: 3' column of the 'dataset' DataFrame. This method replaces any missing values in the specified column with the numeric value 8.643. The inplace=True argument ensures that the changes are made directly to the 'dataset' DataFrame.

A screenshot of a computer

Description automatically generated

**Data Export:**

To store the cleaned and changed dataset for further analysis, the code uses dataset.to\_csv("new\_dataset.csv", index=False) to save the modified DataFrame to a new CSV file named 'new\_dataset.csv'.

A screenshot of a computer

Description automatically generated

**Discuss possible problems you plan to investigate for future studies**.

For future studies, I plan to investigate:

Exploratory Data Analysis (EDA): To acquire deeper insights, try undertaking more complete EDA, such as investigating connections between variables, displaying distributions, and detecting outliers.

Data Visualization: Extend the visualization capabilities to incorporate different sorts of plots, such as histograms, box plots, scatter plots, or time series plots, to expose new insights in the data.

Check weather Simpson's paradox exists in your dataset and explain what you find and why you choose these visualization methods.

Finally, this report highlights the critical procedures involved in preparing a dataset comprising unemployment rates by county. This function prepares the data for future analysis, visualization, or modeling by importing the data, investigating its features, addressing missing values, and exporting the cleaned dataset.