Documentation of Data Mining

By:

Manar Qadi

Khawla Al_Areeqi

Atrab Al_Dubai

Safa'a Almiri

The Dataset Is Jobs and Salaries in Data Science

About Dataset:

work_year: The year in which the data was recorded. This field indicates the temporal context of the data, important for understanding salary trends over time.

job_title: The specific title of the job role, like 'Data Scientist', 'Data Engineer', or 'Data Analyst'. This column is crucial for understanding the salary distribution across various specialized roles within the data field.

job_category: A classification of the job role into broader categories for easier analysis. This might include areas like 'Data Analysis', 'Machine Learning', 'Data Engineering', etc.

salary_currency: The currency in which the salary is paid, such as USD, EUR, etc. This is important for currency conversion and understanding the actual value of the salary in a global context.

salary: The annual gross salary of the role in the local currency. This raw salary figure is key for direct regional salary comparisons.

salary_in_usd: The annual gross salary converted to United States Dollars (USD). This uniform currency conversion aids in global salary comparisons and analyses.

employee_residence: The country of residence of the employee. This data point can be used to explore geographical salary differences and cost-of-living variations.

experience_level: Classifies the professional experience level of the employee. Common categories might include 'Entry-level', 'Mid-level', 'Senior', and 'Executive', providing insight into how experience influences salary in data-related roles.

employment_type: Specifies the type of employment, such as 'Full-time', 'Part-time', 'Contract', etc. This helps in analyzing how different employment arrangements affect salary structures.

work_setting: The work setting or environment, like 'Remote', 'In-person', or 'Hybrid'. This column reflects the impact of work settings on salary levels in the data industry.

company_location: The country where the company is located. It helps in analyzing how the location of the company affects salary structures.

company_size: The size of the employer company, often categorized into small (S), medium (M), and large (L) sizes. This allows for analysis of how company size influences salary.

As Code:

1-
```
dataset=pd.read_csv("dataset.csv")
dataset
```

This code is used to read a CSV file and store it in a variable called "dataset". It is assumed that the "dataset.csv" file is located in the same directory from which the code is being run.

After executing this code, you will be able to access the data in the CSV file by using the "dataset" variable.

2- `dataset.info()`:

in this context is used to display information about the structure and content of the data stored in the dataset variable.

3- `missing_values_count = dataset.isnull().sum()` :

calculates the number of missing values in each column of the dataset and stores the result in the variable missing_values_count.

4- dataset['job_category'].fillna(dataset['job_category'].mode()[0], inplace=True)

fills the missing values in the 'job_category' column of the dataset with the mode (most frequent value) of that column.

The fillna() function is used to replace missing values with a specified value. In this case, the mode of the 'job_category' column is calculated using the mode() function, which returns a Series containing the most frequent value(s) in the column. The [0] indexing is used to select the first value from the mode Series. Finally, the inplace=True argument is used to modify the dataset in-place, meaning the changes are applied directly to the dataset without creating a new copy.

5- job_categories = dataset['job_category'].unique()
for category in job_categories:
mode_value = dataset[dataset['job_category'] == category]['job_title'].mode()[0]
dataset.loc[(dataset['job_category'] == category) & (dataset['job_title'].isnull()), 'job_title'] = mode_value

the code iterates over each unique job category in the dataset, calculates the mode of the 'job_title' column for that category, and then fills the missing values in the 'job_title' column with the corresponding mode value for each category. This ensures that missing 'job_title' values are replaced with the most frequent 'job_title' value observed for each job category.

6- The code dataset['experience_level'].fillna(method='ffill', inplace=True)

fills the missing values in the 'experience_level' column of the dataset using the previous available value (forward fill).
The method='ffill' parameter indicates the use of the "forward fill" technique, which copies the previous available value in the column to fill the missing values. This method is used when we assume that the previous values are relevant and can be used to fill the missing values.

The inplace=True parameter indicates that the modification will be done in-place, meaning that the changes will be applied directly to the original dataset without creating a new copy.

```
7- for i in range(len(dataset)):
    if pd.isnull(dataset.loc[i, 'salary_currency']):
        if dataset.loc[i, 'salary'] == dataset.loc[i, 'salary_in_usd']:
            dataset.loc[i, 'salary_currency'] = 'USD'
        else:
            dataset.loc[i, 'salary_currency'] = dataset.loc[i-1, 'salary_currency']
```

this code checks each row in the dataset. If the 'salary_currency' is null, it compares the 'salary' and 'salary_in_usd' values. If they are equal, it assigns the value 'USD' to the 'salary_currency'. Otherwise, it fills in the missing 'salary_currency' value with the currency value from the previous row.

```
8- dataset.to_csv('dataset.csv', index=False)
```

this code saves the dataset as a CSV file, allowing you to export and share it with others or use it for further analysis.

```
9- for i in range(len(dataset)):
    if pd.isnull(dataset.loc[i, 'salary']):
        if dataset.loc[i, 'salary_currency'] == 'USD':
            dataset.loc[i, 'salary'] = dataset.loc[i, 'salary_in_usd']
        else:
            dataset['salary'].fillna(dataset['salary'].mean(), inplace=True)
```

this code checks each row in the dataset. If the 'salary' value is null, it checks if the 'salary_currency' is 'USD'. If it is, it assigns the value of 'salary_in_usd' to 'salary'. If the 'salary_currency' is not 'USD', it fills the missing 'salary' values with the mean of the 'salary' column.