Pandas

W2 - Practical 2

Learning Objectives

1. Data description
    a. df.info()
    b. df.describe()
    c. df.value_counts()
    d. df.shape

2. Missing Values
    a. df.isnull().sum()
    b. df.isna().sum()

3. Unique values
    a. Df[['column']].unique()
    b. Df[['column']].value_counts()

4. Handling Missing Values
    a. Replace the missing values using df['column'].fillna(new_value)
    b. Delete rows with missing values df.drop([row indexes])

Task 1: Read the "online_store_customer_data.csv" file into a data frame named as fulldata_df.

- Find out the number of rows and columns in your data frame by using the function shape.
- Display the statistical summary of your data frame by using describe() function.
- Use function info to find out columns, their data types, and number of non-null values in each column.
- Use function value_counts() to find out unique values and their frequency in columns Sex, Marital_status, and Payement_method.

Task 2: Find missing values in your dataset.

- Use function isna() to find missing values.
- Use function isnull() to find missing values.
- What's the difference between isna() and isnull()? Google it!!!

Answer: ?

Missing values count

```
Amount_spent        241
Referal             154
Age                  42
Gender               28
Employees_status     26
Transaction_date      0
Marital_status        0
State_names           0
Segment               0
Payment_method        0
dtype: int64
```

Task 3: Handle missing values in your dataset.

Handling missing values in the common task in the data pre-processing part. For many reasons most of the time we will encounter missing values. Without dealing with this we can't do the proper model building. You have already find out the missing value count in Task 2. Now, we decided how to handle them. We can handle this by removing affected columns or rows or replacing appropriate values there.

Remove Columns: If there are a lot of values missing a column then it's a good idea to drop/delete that column.

Drop column "Amount_spent" using df.drop(columns=['column name'], inplace=True)

Or

df.drop('column name', axis=1, inplace=True)

Remove Rows: If there are more missing values then it is better to remove rows.

Remove rows from data frame where Employee_status values are missing.

df.dropna(subset = ["Employees_status"], inplace=True)

Impute/Replace Missing Values: Most of the time, we can't afford to delete rows or columns. It's always better to replace missing values rather than deleting data. We will learn how to replace missing values for both numeric and categorical features.

a) **Numeric:** For numerical features, we can replace the missing values with 0 or mean value.

   Replace Amount_spent missing values with the mean value of amount_spent.

- First find the mean value of column "Amount_spent".

```
mean_amount_spent = df['Amount_spent'].mean()
```

- Replace the missing value by using function fillna.

```
df['Amount_spent'].fillna(mean_amount_spent, inplace=True)
```

- **Try your code to replace missing values in Age with the mean age value.**

b) Categorical Features:

- Missing values in Categorical Features could be replace with either 'Unknown' or mode value.
- Replace missing values in "Employee_status" with the mode value of the column "Employee_status".

  First find the mode value using function mode.

```
# Impute Mode in Employees_status column
mode_emp = df['Employees_status'].mode().iloc[0]
```

  NOTE: If there are multiple mode values, df['Employees_status'].mode() will return a Series containing all the mode values. Using iloc[0] ensures that you're selecting the first value from this Series, which avoids an error. This is because pandas expects a single scalar value for replacing missing data, not a list or Series.

  Replace missing values with mode value using function fillna()

```
df['Employees_status'].fillna(mode_emp, inplace=True)
```

- **Try your code to replace the missing values in Gender.**