Assignment-1

March 11, 2023

Name - Eliza Sarwat

USN- 21BTRCS235

[11]:
```
import pandas as pd

df = pd.read_excel('Sample Dataset.xlsx')
```

[12]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Name     6 non-null      object
 1   Age      6 non-null      int64
 2   Gender   6 non-null      object
 3   Salary   6 non-null      int64
dtypes: int64(2), object(2)
memory usage: 320.0+ bytes
None
```

[13]:
```
print(df.describe())
```

```
             Age         Salary
count   6.000000      6.000000
mean   34.000000  65833.333333
std     8.461678  19083.151382
min    22.000000  40000.000000
25%    29.000000  52500.000000
50%    34.000000  67500.000000
75%    39.750000  78750.000000
max    45.000000  90000.000000
```

[18]:
```
salary_groups = df.groupby('Salary')
print(salary_groups.describe())
```

```
        Age
```

```
       count  mean std   min   25%   50%   75%   max
Salary
40000    1.0  22.0 NaN  22.0  22.0  22.0  22.0  22.0
50000    1.0  28.0 NaN  28.0  28.0  28.0  28.0  28.0
60000    1.0  32.0 NaN  32.0  32.0  32.0  32.0  32.0
75000    1.0  45.0 NaN  45.0  45.0  45.0  45.0  45.0
80000    1.0  36.0 NaN  36.0  36.0  36.0  36.0  36.0
90000    1.0  41.0 NaN  41.0  41.0  41.0  41.0  41.0
```

[19]:
```
print(df['Salary'].value_counts())
```

```
60000    1
75000    1
50000    1
80000    1
90000    1
40000    1
Name: Salary, dtype: int64
```

Adding Missing values to our dataset

[20]:
```
import numpy as np

# Add some missing values to the Salary column
df.loc[[1, 3, 5], 'Salary'] = np.nan

# Add a missing value to the Gender column
df.loc[2, 'Gender'] = np.nan

# Add a missing value to the Age column
df.loc[4, 'Age'] = np.nan

print(df)
```

```
     Name   Age  Gender    Salary  Gender  Age
0    Alice    32  Female  60000.0     NaN  NaN
1      Bob    45    Male      NaN     NaN  NaN
2  Charlie    28    Male  50000.0     NaN  NaN
3     Dana    36  Female      NaN     NaN  NaN
4    Emily    41  Female  90000.0     NaN  NaN
5   Franky    22    Male      NaN     NaN  NaN
```

Using Dropna()

[21]:
```
df_dropped = df.dropna()
print(df_dropped)
```

```
Empty DataFrame
Columns: [Name , Age , Gender , Salary, Gender, Age]
```

```
Index: []
```

Using Fillna()

[24]:
```
# Fill missing values in the Salary column with the mean
df_mean = df.fillna({'Salary': df['Salary'].mean()})
print(df_mean)

# Fill missing values in the Age column with the median
df_median = df.fillna({'Age': df['Age'].median()})
print(df_median)
```

```
      Name   Age  Gender         Salary  Gender  Age
0    Alice    32  Female  60000.000000     NaN  NaN
1      Bob    45    Male  66666.666667     NaN  NaN
2  Charlie    28    Male  50000.000000     NaN  NaN
3     Dana    36  Female  66666.666667     NaN  NaN
4    Emily    41  Female  90000.000000     NaN  NaN
5   Franky    22    Male  66666.666667     NaN  NaN
      Name   Age  Gender    Salary  Gender  Age
0    Alice    32  Female  60000.0     NaN  NaN
1      Bob    45    Male      NaN     NaN  NaN
2  Charlie    28    Male  50000.0     NaN  NaN
3     Dana    36  Female      NaN     NaN  NaN
4    Emily    41  Female  90000.0     NaN  NaN
5   Franky    22    Male      NaN     NaN  NaN
```

#Using Replace

[26]:
```
# Replace all occurrences of 'Male' with 'M' and 'Female' with 'F' in the␣
 ↪'Gender' column
df['Gender'].replace({'Male': 'M', 'Female': 'F'}, inplace=True)
```

[27]:
```
# Convert the 'Gender' column to category type
df['Gender'] = df['Gender'].astype('category')

# Create new binary variables for each category in the 'Gender' column
gender_dummies = pd.get_dummies(df['Gender'], prefix='Gender')

# Merge the original DataFrame with the new dummy variables DataFrame
df = pd.concat([df, gender_dummies], axis=1)
```

[28]:
```
from sklearn.preprocessing import OrdinalEncoder

# Create an OrdinalEncoder object
oe = OrdinalEncoder()

# Fit the encoder to the 'Gender' column and transform it
df['Gender_encoded'] = oe.fit_transform(df[['Gender']])
```

1 Write short description of encoding & its methods.

In machine learning, encoding is the process of transforming categorical data into numerical data that can be used by algorithms to build predictive models. This is necessary because many machine learning algorithms can only handle numerical data, and cannot directly work with categorical data.

There are several methods of encoding categorical data, including:

1) Label Encoding: This method assigns each unique value in a categorical column with a unique integer. For example, if a column has values 'red', 'green', and 'blue', these might be encoded as 0, 1, and 2. This method is suitable for ordinal categorical data where there is a natural ordering between the categories.

2) One-Hot Encoding: This method creates a new binary column for each unique value in a categorical column. If a row has a certain value for a categorical column, the corresponding binary column will have a value of 1, and all other binary columns will have a value of 0. This method is suitable for nominal categorical data where there is no natural ordering between the categories.

3) Binary Encoding: This method converts each unique value in a categorical column to a binary code. For example, if a column has values 'red', 'green', and 'blue', these might be encoded as 00, 01, and 10. This method is suitable for categorical data with many unique values.

4) Target Encoding: This method replaces each unique value in a categorical column with the mean of the target variable for that value. This method is suitable for categorical data where the target variable is continuous.

5) Frequency Encoding: This method replaces each unique value in a categorical column with its frequency in the dataset. This method is suitable for categorical data where the frequency of the value is informative.

6) Ordinal Encoding: This method assigns each unique value in a categorical column with a numerical value based on its rank or order. This method is suitable for categorical data where there is a natural ordering between the categories, but the categories cannot be assumed to have an equal distance between them.

The choice of encoding method depends on the nature of the categorical data and the specific requirements of the machine learning algorithm being used.

[ ]: