Data Pre-processing

May 8, 2021

Data Pre-processing

1 Removing Unused Attributes

[1]:
```
import pandas as pd
```

[2]:
```
df = pd.read_csv('titanic.csv')
df.head()
```

[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

[3]:
```
df = df.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1)
df.head()
```

[3]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S

```
3         1       1  female  35.0      1      0  53.1000        S
4         0       3    male  35.0      0      0   8.0500        S
```

1.0.1 To do the same we can just store input and output data in a variables and we will solve the problem without modifying the original dataset

[4]:
```
x = df[['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']]
x.head()
```

[4]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.0	1	0	7.2500	S
1	1	female	38.0	1	0	71.2833	C
2	3	female	26.0	0	0	7.9250	S
3	1	female	35.0	1	0	53.1000	S
4	3	male	35.0	0	0	8.0500	S

2 Managing Null Values

2.0.1 The info() method will provide the information about the non-null values. By using that data we will come to know that how many null values are there in particular attribute.

[5]:
```
df = pd.read_csv('titanic.csv')
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[7]:
```
len(df)
```

[7]:
```
891
```

[8]:
```
df['Age'] = df.fillna(df['Age'].mean())
```

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          891 non-null    object
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(1), int64(5), object(6)
memory usage: 83.7+ KB
```

[10]:
```
df['Embarked'].value_counts()
```

[10]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

[11]:
```
df['Embarked'] = df.fillna('S')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          891 non-null    object
 6   SibSp        891 non-null    int64
```

```
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     891 non-null    object
dtypes: float64(1), int64(5), object(6)
memory usage: 83.7+ KB
```

3 Converting non-numeric data into numeric

3.0.1 Using map() method

[13]:
```
df = pd.read_csv('titanic.csv')
```

[14]:
```
df['Sex'].value_counts()
```

[14]:
```
male      577
female    314
Name: Sex, dtype: int64
```

[15]:
```
df['Sex'].unique()
```

[15]:
```
array(['male', 'female'], dtype=object)
```

[16]:
```
map_val = {'male':0, 'female':1}
```

[17]:
```
df['Sex'] = df['Sex'].map(map_val)
```

[18]:
```
df.head()
```

[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	1	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0				
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0				

```
3            113803  53.1000  C123        S
4            373450   8.0500   NaN        S
```

3.0.2 Using replace() method

[19]:
```
df = pd.read_csv('titanic.csv')
```

[20]:
```
replace_val = {'male':0, 'female':1}
```

[21]:
```
df['Sex'] = df['Sex'].replace(replace_val)
```

[22]:
```
df.head()
```

[22]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name  Sex   Age  SibSp  Parch  \
0                            Braund, Mr. Owen Harris    0  22.0      1      0
1  Cumings, Mrs. John Bradley (Florence Briggs Th…    1  38.0      1      0
2                             Heikkinen, Miss. Laina    1  26.0      0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)    1  35.0      1      0
4                           Allen, Mr. William Henry    0  35.0      0      0

             Ticket     Fare Cabin Embarked
0         A/5 21171   7.2500   NaN        S
1          PC 17599  71.2833   C85        C
2  STON/O2. 3101282   7.9250   NaN        S
3            113803  53.1000  C123        S
4            373450   8.0500   NaN        S
```

3.0.3 Using LabelEncoder from sklearn library

[23]:
```
df = pd.read_csv('titanic.csv')
```

[24]:
```
from sklearn.preprocessing import LabelEncoder
```

[25]:
```
le = LabelEncoder()
```

[26]:
```
df['Sex'] = le.fit_transform(df['Sex'])
```

[27]:
```
df['Sex']
```

[27]:
```
0      1
1      0
2      0
3      0
4      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 891, dtype: int32
```

3.0.4 Using get_dummies() method

[28]:
```
df = pd.read_csv('titanic.csv')
```

[29]:
```
df = pd.get_dummies(df, columns = ['Sex'])
```

[30]:
```
df.head()
```

[30]:

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Sex_female	Sex_male
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	NaN	S	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	38.0	1	0	PC 17599	71.2833	C85	C	1	0
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	C123	S	1	0
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	NaN	S	0	1

4 Feature Scaling

4.0.1 Using MinMaxScaler from sklearn library

[31]:
```
df = pd.read_csv('titanic.csv')
```

[32]:
```
from sklearn.preprocessing import MinMaxScaler
x = df[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']]
```

[33]:
```
scaler = MinMaxScaler()
df_val = x.values
df_valued = scaler.fit_transform(df_val)
norm_df = pd.DataFrame(df_valued)
norm_df.head()
```

[33]:

	0	1	2	3	4	5
0	0.0	1.0	0.271174	0.125	0.0	0.014151
1	1.0	0.0	0.472229	0.125	0.0	0.139136
2	1.0	1.0	0.321438	0.000	0.0	0.015469
3	1.0	0.0	0.434531	0.125	0.0	0.103644
4	0.0	1.0	0.434531	0.000	0.0	0.015713

4.0.2 Using StandardScaler from sklearn library

[34]:
```
df = pd.read_csv('titanic.csv')
```

[35]:
```
from sklearn.preprocessing import StandardScaler
x = df[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']]
```

[36]:
```
std = StandardScaler()
df_val = x.values
df_std = std.fit_transform(df_val)
std_df = pd.DataFrame(df_std)
std_df.head()
```

[36]:

	0	1	2	3	4	5
0	-0.789272	0.827377	-0.530377	0.432793	-0.473674	-0.502445
1	1.266990	-1.566107	0.571831	0.432793	-0.473674	0.786845
2	1.266990	0.827377	-0.254825	-0.474545	-0.473674	-0.488854
3	1.266990	-1.566107	0.365167	0.432793	-0.473674	0.420730
4	-0.789272	0.827377	0.365167	-0.474545	-0.473674	-0.486337