Practice Project - Pandas based : Missing Values Treatment Problem

Table of Contents

Problem Statement:

Problem on data analysis or data engineering where we should never lose any data, ideally there are two ways with which you can deal with missing values one by filling it with mean median and mode and second by removing the rows contating missing values.

1. Importing Libraries

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

2. Python Implementation

Reading the csv file uing the read_csv function

In [ ]:
```
data = pd.read_csv(r'data/tested.csv')
```

The head() function allows us to see the first 5 rows

In [ ]:
```
data.head()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

`Is null function` returns us a boolean table which shows True in all the places where there was a null value.

Places where a value is already present will be marked as False.

In [ ]:
```
data.isnull()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	True	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	True	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
413	False	False	False	False	False	True	False	False	False	False	True	False
414	False	False	False	False	False	False	False	False	False	False	False	False
415	False	False	False	False	False	False	False	False	False	False	True	False
416	False	False	False	False	False	True	False	False	False	False	True	False
417	False	False	False	False	False	True	False	False	False	False	True	False

418 rows × 12 columns

The above output does not give us much of an insight into the data. So we can use the sum() function to add up all the null values in each column.

In [ ]:
```
data.isnull().sum()
```

Out[ ]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

A very important part of data analysis or data engineering is we should never lose any data, ideally there are two ways with which you can deal with missing values one by filling it with mean median and mode and second by removing the rows contating missing values.

In [ ]:
```
data['Age'].mean()
```

Out[ ]:
```
30.272590361445783
```

In [ ]:
```
data['Age'].median()
```

Out[ ]:
```
27.0
```

In [ ]:
```
data['Age'].mode()
```

Out[ ]:
```
0    21.0
1    24.0
dtype: float64
```

In [ ]:
```
data['Age'].fillna(data['Age'].mean(),inplace=True)
```

In [ ]:
```
data.isnull().sum()
```

Out[ ]:
```
PassengerId      0
Survived         0
```

```
Pclass          0
Name            0
Sex             0
Age             0
SibSp           0
Parch           0
Ticket          0
Fare            1
Cabin         327
Embarked        0
dtype: int64
```

In [ ]:
```
data['Cabin'].fillna(data['Cabin'].mode()[0],inplace=True)
```

In [ ]:
```
data_1=data.dropna(subset=["Age"])
data_1.shape
```

Out[ ]:
```
(418, 12)
```

In [ ]:
```
data.isnull().sum()
```

Out[ ]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           1
Cabin          0
Embarked       0
dtype: int64
```
