

1 Load Dataset & Identify Missing Values

```
import pandas as pd

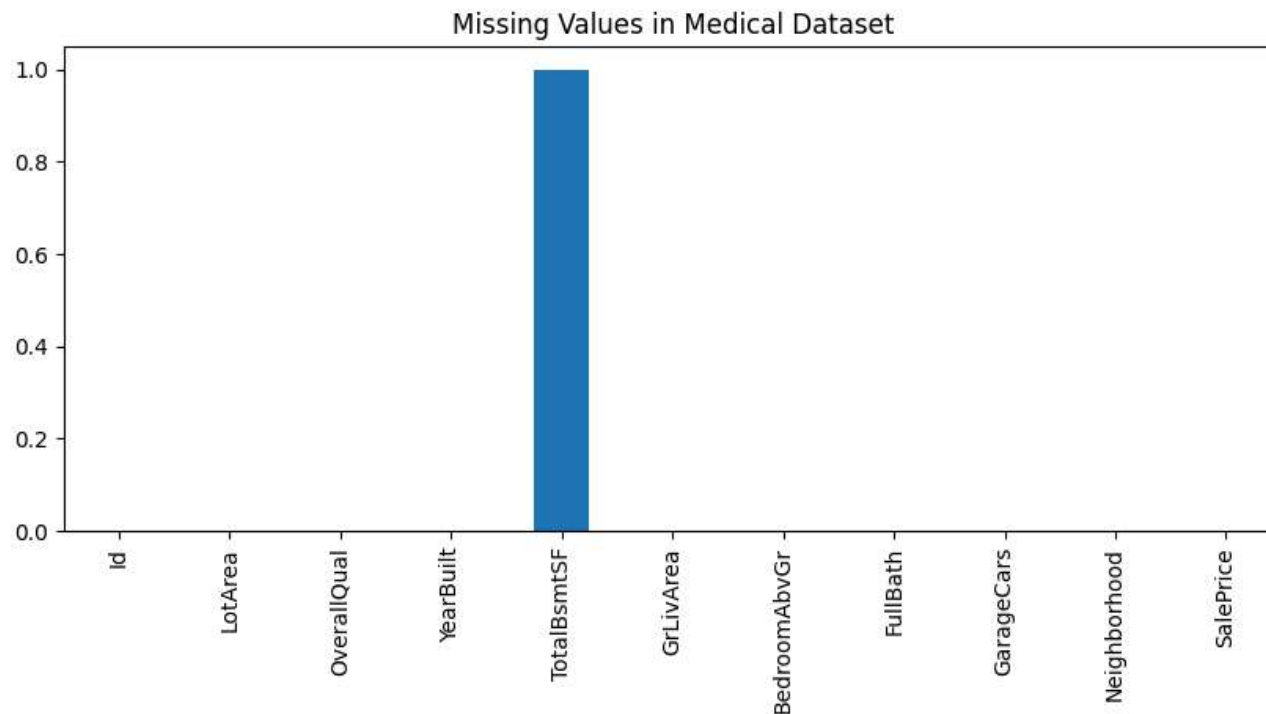
df = pd.read_csv("/content/house_prices.csv")
df.isnull().sum()
```

	0
Id	0
LotArea	0
OverallQual	0
YearBuilt	0
TotalBsmtSF	1
GrLivArea	0
BedroomAbvGr	0
FullBath	0
GarageCars	0
Neighborhood	0
SalePrice	0

dtype: int64

2 Visualize Missing Data

```
import matplotlib.pyplot as plt
df.isnull().sum().plot(kind='bar', figsize=(10,4))
plt.title("Missing Values in Medical Dataset")
plt.show()
```



3 Numerical Imputation

```
num_cols = ['TotalBsmtSF']

for col in num_cols:
    df[col] = df[col].fillna(df[col].median())
```

4 Categorical Imputation

```
cat_cols = df.select_dtypes(include='object').columns

for col in cat_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

/tmp/ipython-input-2422164989.py:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value


```
df[col].fillna(df[col].mode()[0], inplace=True)
```

5 Remove Unnecessary / ID Columns

```
df.drop(columns=['Id'], inplace=True)
```

6 Validate Clean Dataset

```
df.isnull().sum()  
df.head()
```

	LotArea	OverallQual	YearBuilt	TotalBsmntSF	GrLivArea	BedroomAbvGr	FullBath	GarageCars	Neighborhood	SalePrice	
0	8450	7	2003	856.0	1710	3	2	2	CollgCr	208500	
1	9600	6	1976	1262.0	1262	3	2	2	Veenker	181500	
2	11250	7	2001	920.0	1786	3	2	2	CollgCr	223500	
3	9550	7	1915	756.0	1717	3	1	3	Crawfor	140000	
4	14260	8	2000	1145.0	2198	4	2	3	NoRidge	250000	

Next steps:

[Generate code with df](#)[New interactive sheet](#)Start coding or [generate](#) with AI.