Lab 1: Data Preprocessing

R Abhijit Srivathsan
2448044

Importing pandas

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

Loading the datasets

In [19]:
```
data = pd.read_csv('Housing_Price.csv')
data.head()
```

Out[19]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	Misc
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	

5 rows × 81 columns

Calculating the total *missing values*

In [22]:
```
missing_values = data.isnull().sum()
missing_percent = (missing_values / len(data)) * 100
missing_data = pd.DataFrame({'Missing Values': missing_values, 'Percentage': missing_percent})
```

```
missing_data = missing_data[missing_data['Missing Values'] > 0]
print(missing_data)
```

```
              Missing Values  Percentage
LotFrontage              259   17.739726
MasVnrType               872   59.726027
MasVnrArea                 8    0.547945
BsmtQual                  37    2.534247
BsmtCond                  37    2.534247
BsmtExposure              38    2.602740
BsmtFinType1              37    2.534247
BsmtFinType2              38    2.602740
Electrical                 1    0.068493
FireplaceQu              690   47.260274
GarageType                81    5.547945
GarageYrBlt               81    5.547945
GarageFinish              81    5.547945
GarageQual                81    5.547945
GarageCond                81    5.547945
```

Note: Dropped columns with more than 80% missing values

In [21]:
```
# Drop columns with more than 80% missing values
data.drop(columns=['PoolQC', 'MiscFeature', 'Alley', 'Fence'], inplace=True)
```

Filling in $median$ values for `LotFrontage` column, and filling 0's for the other numerical columns

In [24]:
```
# Impute numerical missing values
data['LotFrontage'] = data['LotFrontage'].fillna(data['LotFrontage'].median())
data['GarageYrBlt'] = data['GarageYrBlt'].fillna(0)
data['MasVnrArea'] = data['MasVnrArea'].fillna(0)
```

Filling `None` for categorical missing values

In [28]:
```
# Impute categorical missing values
garage_cols = ['GarageType', 'GarageFinish', 'GarageQual', 'GarageCond']
data[garage_cols] = data[garage_cols].fillna('None')

bsmt_cat_cols = ['BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2']
data[bsmt_cat_cols] = data[bsmt_cat_cols].fillna('None')

data['FireplaceQu'] = data['FireplaceQu'].fillna('None')
data['MasVnrType'] = data['MasVnrType'].fillna('None')

# Impute mode for categorical variables
data['Electrical'] = data['Electrical'].fillna(data['Electrical'].mode()[0])
```

Computing *Total missing values*

In [27]:
```
# Verify missing values handled
print("Total missing values left:", data.isnull().sum().sum())
```

```
Total missing values left: 0
```