Credential's

Name : Mayank Anand

Email : 21410010345

Exploratory data analysis (EDA)

```
# importing libraries
import pandas as pd
import numpy as np
from matplotlib.pyplot import plot
```

```
# to read the file
df = pd.read_csv("Train_Data.csv")
```

```
# to view the first few data's of the dataframe
df.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	2
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	1
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	2
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnormal	2
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	2

5 rows × 81 columns

```
# to view the last few data's of the dataframe
df.tail()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD	Normal	2
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	Normal	2
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	Normal	2
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD	Normal	1
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD	Normal	2

5 rows × 81 columns

```
# to display about the total number of rows and columns in the dataset
df.shape
```

```
(1460, 81)
```

```
# to get info of dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrArea     1452 non-null   float64
 26  ExterQual      1460 non-null   object
 27  ExterCond      1460 non-null   object
 28  Foundation     1460 non-null   object
 29  BsmtQual       1423 non-null   object
 30  BsmtCond       1423 non-null   object
 31  BsmtExposure   1422 non-null   object
 32  BsmtFinType1   1423 non-null   object
 33  BsmtFinSF1     1460 non-null   int64
 34  BsmtFinType2   1422 non-null   object
 35  BsmtFinSF2     1460 non-null   int64
 36  BsmtUnfSF      1460 non-null   int64
 37  TotalBsmtSF    1460 non-null   int64
 38  Heating        1460 non-null   object
 39  HeatingQC      1460 non-null   object
 40  CentralAir     1460 non-null   object
 41  Electrical     1459 non-null   object
 42  1stFlrSF       1460 non-null   int64
 43  2ndFlrSF       1460 non-null   int64
 44  LowQualFinSF   1460 non-null   int64
 45  GrLivArea      1460 non-null   int64
 46  BsmtFullBath   1460 non-null   int64
 47  BsmtHalfBath   1460 non-null   int64
 48  FullBath       1460 non-null   int64
 49  HalfBath       1460 non-null   int64
 50  BedroomAbvGr   1460 non-null   int64
 51  KitchenAbvGr   1460 non-null   int64
 52  KitchenQual    1460 non-null   object
 53  TotRmsAbvGrd   1460 non-null   int64
 54  Functional     1460 non-null   object
 55  Fireplaces     1460 non-null   int64
 56  FireplaceQu    770 non-null    object
 57  GarageType     1379 non-null   object
 58  GarageYrBlt    1379 non-null   float64
 59  GarageFinish   1379 non-null   object
 60  GarageCars     1460 non-null   int64
 61  GarageArea     1460 non-null   int64
 62  GarageQual     1379 non-null   object
 63  GarageCond     1379 non-null   object
 64  PavedDrive     1460 non-null   object
 65  WoodDeckSF     1460 non-null   int64
 66  OpenPorchSF    1460 non-null   int64
 67  EnclosedPorch  1460 non-null   int64
 68  3SsnPorch      1460 non-null   int64
 69  ScreenPorch    1460 non-null   int64
 70  PoolArea       1460 non-null   int64
 71  PoolQC         7 non-null      object
 72  Fence          281 non-null    object
 73  MiscFeature    54 non-null     object
 74  MiscVal        1460 non-null   int64
 75  MoSold         1460 non-null   int64
 76  YrSold         1460 non-null   int64
 77  SaleType       1460 non-null   object
 78  SaleCondition  1460 non-null   object
 79  SalePrice      1460 non-null   int64
dtypes: float64(11), int64(34), object(36)
memory usage: 1.1+ MB
```

```
# to get statistical information about the dataset
df.describe()
```

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	MoSold	YrSold	SalePrice
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	...	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	...	94.244521	46.660274	21.954110	3.409589	15.060959	2.758904	43.489041	6.321918	2007.815753	180921.195890
std	421.610009	42.300571	24.284752	9981.264449	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	...	125.338794	66.256028	61.119149	29.317331	55.757415	40.177307	496.123024	2.703626	1.328095	79442.502883
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	2006.000000	34900.000000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	5.000000	2007.000000	129975.000000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	164.000000	383.500000	...	0.000000	25.000000	0.000000	0.000000	0.000000	0.000000	0.000000	6.000000	2008.000000	163000.000000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	335.000000	712.250000	...	168.000000	68.000000	0.000000	0.000000	0.000000	0.000000	0.000000	8.000000	2009.000000	214000.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.000000	547.000000	552.000000	508.000000	480.000000	738.000000	15500.000000	12.000000	2010.000000	755000.000000

8 rows × 38 columns

```
# to get statistical information about the dataset using with the categorical column
df.describe(include=['object'])
```

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	...	GarageType	GarageFinish	GarageQual	GarageCond	PavedDrive	PoolQC	Fence	MiscFeature	SaleType	SaleCondition
count	1460	1460	91	1460	1460	1460	1460	1460	1460	1460	...	1379	1379	1379	1379	1460	7	281	54	1460	1460
unique	5	2	2	4	4	2	5	3	25	9	...	6	3	5	5	3	3	4	4	9	6
top	RL	Pave	Grvl	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	...	Attchd	Unf	TA	TA	Y	Gd	MnPrv	Shed	WD	Normal
freq	1151	1454	50	925	1311	1459	1052	1382	225	1260	...	870	605	1311	1326	1340	3	157	54	1267	1198

4 rows × 43 columns

```
# to know about number of duplicate row's in the dataframe
df.duplicated().sum()
```

```
0
```

```
# to know about the missing value in the dataframe
df.isnull.sum()
```

```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage    259
LotArea          0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 81, dtype: int64
```

Handling Missing Values

```
# to drop the Id column which is unnecessary for the dataset
df.drop('Id', axis=1, inplace = True)
```

```
# to view all the columns present in the dataset
df.columns
```

```
Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrArea', 'ExterQual',
       'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea',
       'GarageQual', 'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', 'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence',
       'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

```
# to check the missing value in the dataset
threshold = len(df)*0.20
colum = df.isnull().sum() > threshold
columnsToBeDropped = df.columns[colum]
```

```
# To know how many columns follow the above condition
columnsToBeDropped.shape
```

```
(5,)
```

```
# To drop the columns that all follow the above condition
df.drop(columnsToBeDropped, axis=1, inplace = True)
df.shape
```

```
(1460, 75)
```

```
# To know how the names of columns that are remaing after removing the above columns
df.columns
```

```
Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'GarageType', 'GarageYrBlt',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold',
       'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

```
# To know the count of different type of data types present in the dataset
df.dtypes.value_counts()
```

```
object  38
int64   34
float64  3
dtype: int64
```

```
# To check how many null values exist in the dataframe
df.isnull.sum()
```

```
MSSubClass       0
MSZoning         0
LotFrontage    259
LotArea          0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 75, dtype: int64
```

```
# To check the percentage of how many data are null in the dataset
df.isnull().sum()/len(df)
```

```
MSSubClass    0.000000
MSZoning      0.000000
LotFrontage   0.177397
LotArea       0.000000
               ...
MoSold        0.000000
YrSold        0.000000
SaleType      0.000000
SaleCondition 0.000000
SalePrice     0.000000
Length: 75, dtype: float64
```

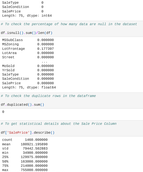
```
# To drop the duplicate rows in the dataframe
df.duplicated().sum()
```

```
0
```

```
# To get statistical details about the Sale Price Column
df["SalePrice".]describe()
```

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

```
# To get a visual view about the Sale Price Column
df["SalePrice".]hist(bins=50);
```




```
# To drop the columns due to low variability:
df.drop(
    [
        'Street',
        'Utilities',
        'LandSlope',
        'MasVnrArea',
        'ExterCond',
        'BsmtCond',
        'Heating',
        'GarageCond',
        'PoolArea',
        'MiscVal',
        'SaleType',
        'SaleCondition',
    ],
    axis=1,
    inplace=True,
)
```

```
# To know the shape of the dataframe after removing certain columns
df.shape
```

```
(1460, 67)
```

```
# To replace the missing data in numeric columns by median
for col in df select_dtypes(include=['int64','float64']):
    if df[col].isnull().sum() > 0:
        median_value = df[col].median()
        df[col].fillna(median_value, inplace=True)
```

```
# To check how many null values exist in the dataframe after performing the above replacement operation
df.isnull().sum()
```

```
MSSubClass    0
MSZoning      0
LotFrontage   0
LotArea       0
LotShape      0
              ..
MoSold        0
YrSold        0
SaleType      0
SaleCondition 0
SalePrice     0
Length: 67, dtype: int64
```

```
# To handle missing data in categorical columns by mode
for col in df select_dtypes(include=['object']):
    if df[col].isnull().sum() > 0:
        mode_value = df[col].mode()[0]
        df[col].fillna(mode_value, inplace=True)
```

```
# To check whether we have any null values present or not
df.isnull().sum()
```

```
MSSubClass    0
MSZoning      0
LotFrontage   0
LotArea       0
LotShape      0
              ..
MoSold        0
YrSold        0
SaleType      0
SaleCondition 0
SalePrice     0
Length: 67, dtype: int64
```

Data Transformation & Reduction & Feature Engineering

```
# To combine the both column in one
df["Total_Floors"] = df["1stFlrSF"] + df["2ndFlrSF"]
df["Total_Floors"].head()
```

```
0    1710
1    1262
2    1786
3    1717
4    2198
Name: Total_Floors, dtype: int64
```

```
# To combine the both column to one
df["House_quality_score"] = df["OverallQual"] * df["OverallQual"]
df["House_quality_score".]head()
```

```
0    49
1    36
2    64
3    49
4    36
Name: House_quality_score, dtype: int64
```

```
# To combine the three column to one
df["Total_porch_area"] =
    df["OpenPorchSF"] + df["EnclosedPorch"] + df["ScreenPorch"] + df["WoodDeckSF"]
df["Total_porch_area".]head()
```

```
0    61
1    43
2    80
3    304
4    84
Name: Total_porch_area, dtype: int64
```

```
# To combine the both column to one
df["YearBuiltRemod"] = df["YearBuilt"] + df["YrSold"]
df["YearBuiltRemod".]head()
```

```
# To drop the irrelevant column
df.drop(
    [
        'BsmtFinSF1',
        'BsmtFinSF2',
        'BsmtUnfSF',
        'FullBath',
        'HalfBath',
        'BedroomAbvGr',
        'KitchenAbvGr',
        'EnclosedPorch',
        'ScreenPorch',
        'WoodDeckSF',
        'OpenPorchSF',
        'YearBuilt',
        'YrSold',
    ],
    axis=1,
    inplace=True,
)
```

```
# To know the shape of the dataframe after performing the cleaning operations
df.shape
```

```
(1460, 61)
```

```
# To know the count of different type of data types present in the dataset
df.dtypes.value_counts()
```

```
object  38
int64   20
float64  3
dtype: int64
```

```
# To view the first few data's of the dataset
df.head()
```

	MSSubClass	MSZoning	LotFrontage	LotArea	LotShape	LandContour	LotConfig	LandSlope	Neighborhood	Condition1	...	YrBuilt	SaleType	SaleCondition	SalePrice	Total_Floors	House_quality_score
0	60	RL	65.0	8450	Reg	Lvl	Inside	Gtl	CollgCr	Norm	...	2003	WD	Normal	208500	1710	49
1	20	RL	80.0	9600	Reg	Lvl	FR2	Gtl	Veenker	Feedr	...	1976	WD	Normal	181500	1262	36
2	60	RL	68.0	11250	IR1	Lvl	Inside	Gtl	CollgCr	Norm	...	2001	WD	Normal	223500	1786	64
3	70	RL	60.0	9550	IR1	Lvl	Corner	Gtl	Crawfor	Norm	...	2006	WD	Abnormal	140000	1717	49
4	60	RL	84.0	14260	IR1	Lvl	FR2	Gtl	NoRidge	Norm	...	2008	WD	Normal	250000	2198	36

5 rows × 61 columns

Encoding Categorical Columns

```
# To load the Preprocessing
from sklearn.preprocessing import LabelEncoder
```

```
label_encoder = LabelEncoder()
```

```
# To iterate over all columns in the dataframe
for col in df select_dtypes(include=['object']):
    df[col] = label_encoder.fit_transform(df[col])
```

```
df.head()
```

	MSSubClass	MSZoning	LotFrontage	LotArea	LotShape	LandContour	LotConfig	LandSlope	Neighborhood	Condition1	...	YrBuilt	SaleType	SaleCondition	SalePrice	Total_Floors	House_quality_score
0	60	3	65.0	8450	3	3	4	0	5	2	...	2003	8	4	208500	1710	49
1	20	3	80.0	9600	3	3	2	0	24	1	...	1976	8	4	181500	1262	36
2	60	3	68.0	11250	0	3	4	0	5	2	...	2001	8	4	223500	1786	64
3	70	3	60.0	9550	0	3	0	0	6	2	...	2006	8	0	140000	1717	49
4	60	3	84.0	14260	0	3	2	0	15	2	...	2008	8	4	250000	2198	36

5 rows × 61 columns

```
# To know the count of different type of data types present in the dataset
df.dtypes.value_counts()
```

```
int32   38
int64   20
float64  3
dtype: int64
```