

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

dataset = pd.read_csv("/content/HousePricePrediction.xlsx - Sheet1.csv")

# Printing first 5 records of the dataset
print(dataset.head(5))
```

	Id	MSSubClass	MSZoning	LotArea	LotConfig	BldgType	OverallCond	\
0	0	60	RL	8450	Inside	1Fam	5	
1	1	20	RL	9600	FR2	1Fam	8	
2	2	60	RL	11250	Inside	1Fam	5	
3	3	70	RL	9550	Corner	1Fam	5	
4	4	60	RL	14260	FR2	1Fam	5	

	YearBuilt	YearRemodAdd	Exterior1st	BsmtFinSF2	TotalBsmtSF	SalePrice
0	2003	2003	VinylSd	0.0	856.0	208500.0
1	1976	1976	MetalSd	0.0	1262.0	181500.0
2	2001	2002	VinylSd	0.0	920.0	223500.0
3	1915	1970	Wd Sdng	0.0	756.0	140000.0
4	2000	2000	VinylSd	0.0	1145.0	250000.0

```
dataset.shape
```

```
(2919, 13)
```

```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:", len(object_cols))
```

```
int_ = (dataset.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:", len(num_cols))
```

```
fl = (dataset.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:", len(fl_cols))
```

```
Categorical variables: 4
Integer variables: 6
Float variables: 3
```

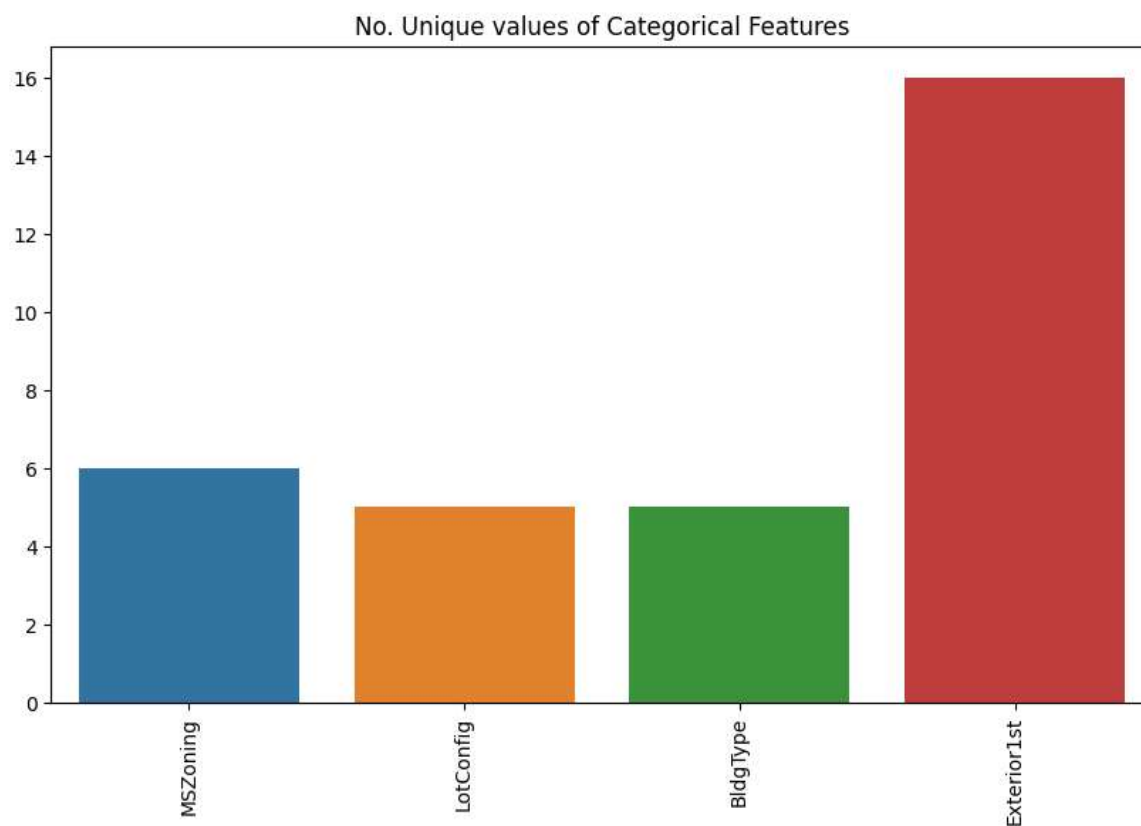
```
plt.figure(figsize=(12, 6))
sns.heatmap(dataset.corr(),
             cmap = 'BrBG',
             fmt = '.2f',
             linewidths = 2,
             annot = True)
```

```
<ipython-input-8-8feaf9d49085>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future
sns.heatmap(dataset.corr()),
<Axes: >
```



```
unique_values = []
for col in object_cols:
    unique_values.append(dataset[col].unique().size)
plt.figure(figsize=(10,6))
plt.title('No. Unique values of Categorical Features')
plt.xticks(rotation=90)
sns.barplot(x=object_cols,y=unique_values)
```

```
<Axes: title={'center': 'No. Unique values of Categorical Features'}>
```



```
plt.figure(figsize=(18, 36))
plt.title('Categorical Features: Distribution')
plt.xticks(rotation=90)
index = 1
```

```
for col in object_cols:
    y = dataset[col].value_counts()
    plt.subplot(11, 4, index)
    plt.xticks(rotation=90)
    sns.barplot(x=list(y.index), y=y)
```



```
AttributeError: 'OneHotEncoder' object has no attribute 'get_feature_names'
Apparently, it has been renamed to get_feature_names_out."""
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:868: FutureWarning:
  warnings.warn(
'\nAttributeError: 'OneHotEncoder' object has no attribute 'get_feature_names'\nApparently, it has been renamed to get_feature_names_out.'
```

```
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
```

```
X = df_final.drop(['SalePrice'], axis=1)
Y = df_final['SalePrice']
```

```
# Split the training set into
# training and validation set
X_train, X_valid, Y_train, Y_valid = train_test_split(
    X, Y, train_size=0.8, test_size=0.2, random_state=0)
```

```
from sklearn import svm
from sklearn.svm import SVC
from sklearn.metrics import mean_absolute_percentage_error
```

```
model_SVR = svm.SVR()
model_SVR.fit(X_train, Y_train)
Y_pred = model_SVR.predict(X_valid)
```

```
print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.1870512931870423
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
model_RFR = RandomForestRegressor(n_estimators=10)
model_RFR.fit(X_train, Y_train)
Y_pred = model_RFR.predict(X_valid)
```

```
mean_absolute_percentage_error(Y_valid, Y_pred)
```

```
0.191021799467832
```

```
from sklearn.linear_model import LinearRegression
```

```
model_LR = LinearRegression()
model_LR.fit(X_train, Y_train)
Y_pred = model_LR.predict(X_valid)
```

```
print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.18741683841599854
```

```
!pip3 install catboost
```

```
Collecting catboost
```

```
  Downloading catboost-1.2.2-cp310-cp310-manylinux2014_x86_64.whl (98.7 MB)
```

```
98.7/98.7 MB 9.4 MB/s eta 0:00:00
```

```
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.7.1)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.23.5)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.5.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.11.3)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.15.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2023.3.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.1.1)
```

```
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (3.1.1)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly->catboost) (8.2.3)
Installing collected packages: catboost
Successfully installed catboost-1.2.2
```

```
# This code is contributed by @amartajisce
from catboost import CatBoostRegressor
cb_model = CatBoostRegressor()
cb_model.fit(X_train, Y_train)
preds = cb_model.predict(X_valid)
from sklearn.metrics import r2_score
cb_r2_score=r2_score(Y_valid, preds)
cb_r2_score
```

