NAME:THALAPATHY SURYA.K

EXAM NO:422221104041

COLLEGE CODE:4222

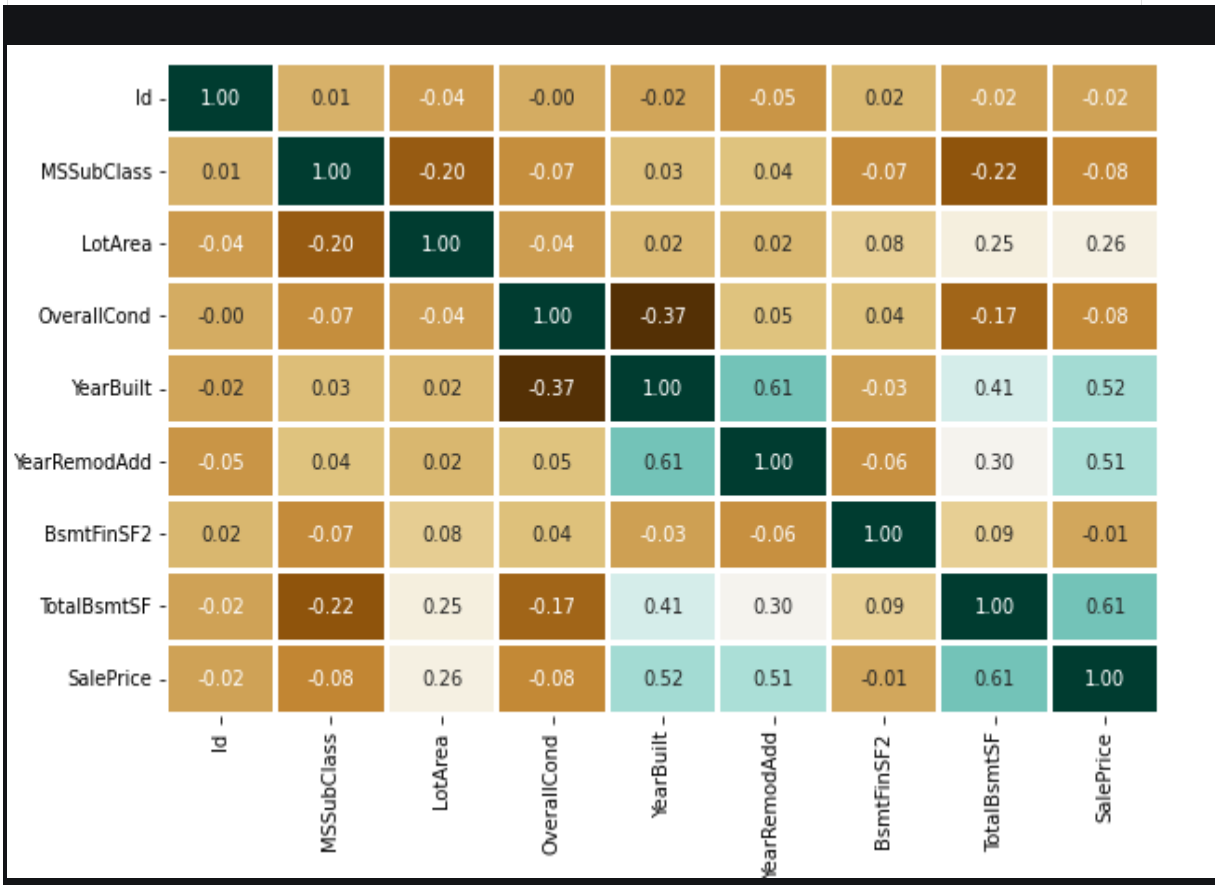
COLLEGE NAME:SURYA GROUP OF INSTITUTIONS

Exploratory Data Analysis

EDA refers to the deep analysis of data so as to discover different patterns and spot anomalies. Before making inferences from data it is essential to examine all your variables.
So here let's make a heatmap using seaborn library.

```
plt.figure(figsize=(12, 6))
```

```
sns.heatmap(dataset.corr(),
            cmap = 'BrBG',
            fmt = '.2f',
            linewidths = 2,            annot = True)
```



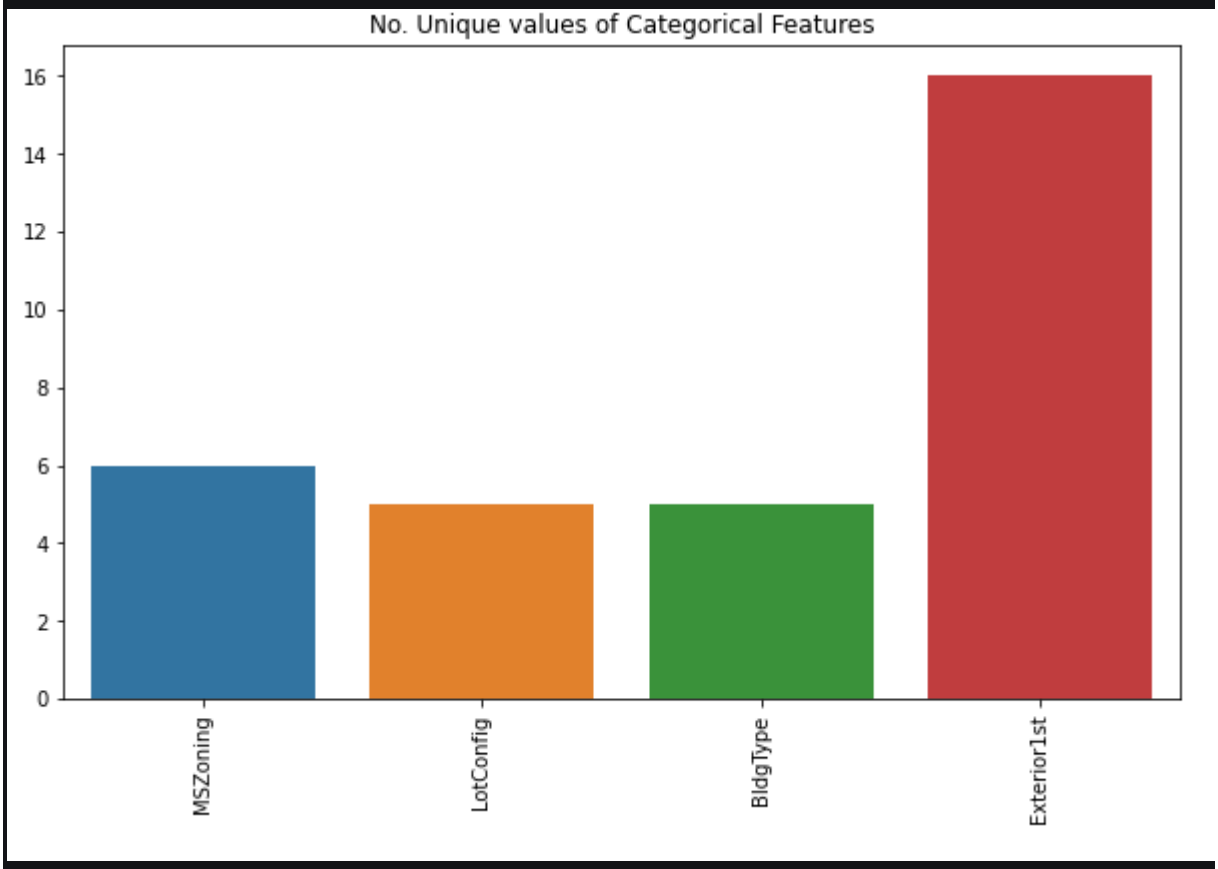
To analyze the different categorical features. Let's draw the barplot.

```
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)
```



The plot shows that Exterior1st has around 16 unique categories and other features have around 6 unique categories. To findout the actual count of each category we can plot the bargraph of each four features separately.

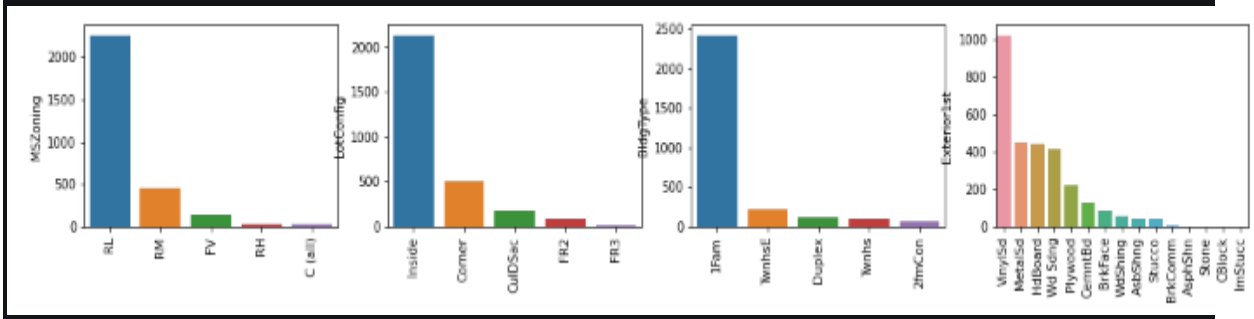
```
plt.figure(figsize=(18, 36))

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)
    index += 1
```



Replacing SalePrice empty values with their mean values to make the data distribution symmetric.

Random Forest Regression

Random Forest is an ensemble technique that uses multiple of decision trees and can be used for both regression and classification tasks. To read more about random forests refer this.

```
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)
```

Linear Regression

Linear Regression predicts the final output-dependent value based on the given independent features. Like, here we have to predict SalePrice depending on features like MSSubClass, YearBuilt, BldgType, Exterior1st etc. To read more about Linear Regression refer this.

```
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))
```