***NAME:THALAPATHY SURYA.K***

***EXAM NO:422221104041***

***COLLEGE CODE:4222***

***COLLEGE NAME:SURYA GROUP OF INSTITUTIONS***

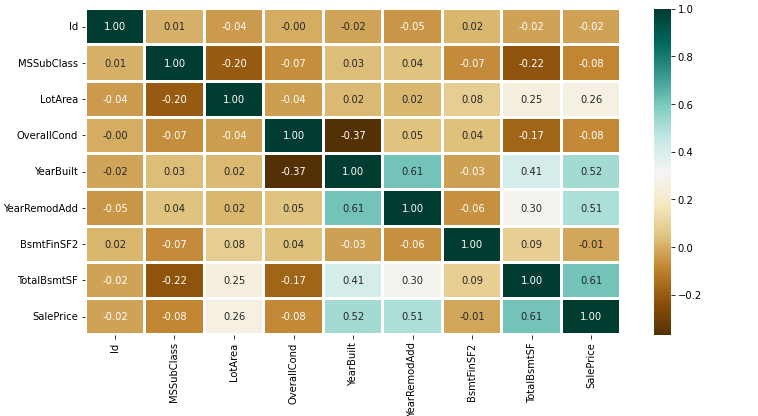


**Exploratory Data Analysis**

[EDA](https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/) 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.

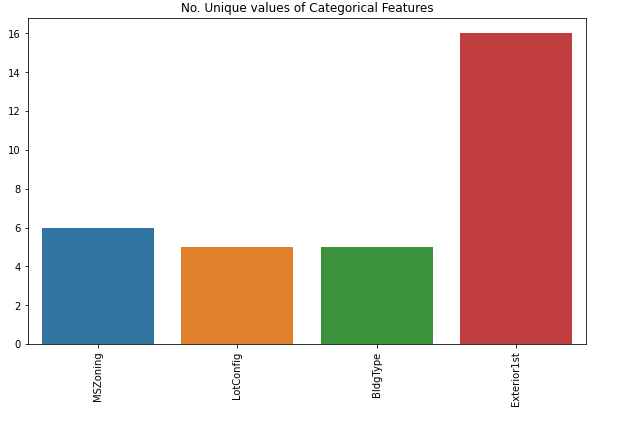
So here let’s make a [heatmap](https://www.geeksforgeeks.org/how-to-create-a-seaborn-correlation-heatmap-in-python/) using seaborn library.

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



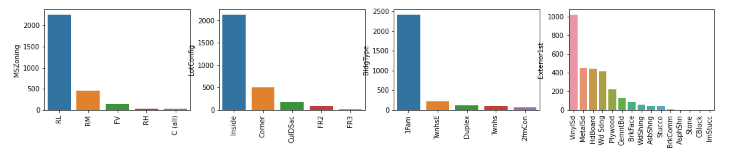
To analyze the different categorical features. Let’s draw the [barplot](https://www.geeksforgeeks.org/bar-plot-in-matplotlib/).

|  |
| --- |
| 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.

0.18705129

**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.](https://www.geeksforgeeks.org/random-forest-regression-in-python/)

|  |
| --- |
| 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) |

0.1929469

**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.](https://www.geeksforgeeks.org/ml-linear-regression/)

|  |
| --- |
| 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)) |

0.187416838