Steps Needed for Data Cleaning:

* Using Data Dictionary fill out the values for different columns

Syntax: cars["CarName"] = np.where(cars["CarName"] == "vw dasher", "volkswagen dasher", cars["CarName"])

* Column:MSSubClass

house["MSSubClass"]=np.where(house["MSSubClass"]==20,"1-STORY 1946 & NEWER ALL STYLES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==30,"1-STORY 1945 & OLDER", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==40,"1-STORY W/FINISHED ATTIC ALL AGES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==45,"1-1/2 STORY – UNFINISHED ALL AGES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==50,"1-1/2 STORY FINISHED ALL AGES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==60,"1-STORY 1946 & NEWER", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==70,"1-STORY 1945 & OLDER", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==75,"2-1/2 STORY ALL AGES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==80,"SPLIT OR MULTI-LEVEL", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==85,"SPLIT FOYER", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==90,"DUPLEX – ALL STYLES AND AGES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==120,"1-STORY PLANNED UNIT DEVELOPMENT 1946 & NEWER", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==150,"1-1/2 STORY PUD – ALL AGES", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==160,"2-STORY PUD 1946 & NEWER", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==180,"PUD – MULTILEVEL", house["MSSubClass"])

house["MSSubClass"]=np.where(house["MSSubClass"]==190,"2 FAMILY CONVERSION – ALL STYLES AND AGES", house["MSSubClass"])

* Change the column names, datatypes, values for columns
* Fill Null values, remove columns that are not required

Syntax: cars['fueltype'].fillna(value='gas', inplace=True)

* Column: LotFrontage

house[‘LotFrontage’].fillna(value=0.00,inplace=True)

* Column: MasVnrArea

#filling null values with 103.68 as average area

house['MasVnrArea'].fillna(value=103.68,inplace=True)

* Correlations:

Syntax:

corrmat=cars.corr()

top\_corr\_feature=corrmat.index

plt.figure(figsize=(10,10))

g=sns.heatmap(cars[top\_corr\_feature].corr(), annot=True,cmap="RdYlGn")

#correlation matrix

corrmat=house.corr()

top\_corr\_feature=corrmat.index

plt.figure(figsize=(25,25))

g=sns.heatmap(house[top\_corr\_feature].corr(), annot=True,cmap="RdYlGn")

#correlation by salesprice

num = house.select\_dtypes(exclude = 'object')

numcorr = house.corr()

f, ax = plt.subplots(figsize = (19,1)) # set figure size

sns.heatmap(numcorr.sort\_values(by = 'SalePrice', ascending = False).head(1), annot = True, fmt = ".2f")

plt.show()

#If we wish to label the strength of the association, for absolute values of r, **0-0.19 is regarded as very weak, 0.2-0.39 as weak, 0.40-0.59 as moderate, 0.6-0.79 as strong and 0.8-1 as very strong correlation**