In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('C://Users/Gopi/Desktop/house/train.csv')
```

In [3]:
```
print(df.shape)
```

```
(1460, 81)
```

In [4]:
```
df['LotFrontage']=df['LotFrontage'].fillna(df['LotFrontage'].mean())
```

In [5]:
```
df.drop(['Alley'],axis=1,inplace=True)
```

In [6]:
```
df['FireplaceQu']=df['FireplaceQu'].fillna(df['FireplaceQu'].mode()[0])
df['GarageType']=df['GarageType'].fillna(df['GarageType'].mode()[0])
```

In [7]:
```
df.drop(['GarageYrBlt'],axis=1,inplace=True)
```

In [8]:
```
df['GarageFinish']=df['GarageFinish'].fillna(df['GarageFinish'].mode()[0])
df['GarageQual']=df['GarageQual'].fillna(df['GarageQual'].mode()[0])
df['GarageCond']=df['GarageCond'].fillna(df['GarageCond'].mode()[0])
```

In [9]:
```
df.drop(['PoolQC','Fence','MiscFeature'],axis=1,inplace=True)
```

In [10]:
```
df.drop(['Id'],axis=1,inplace=True)
```

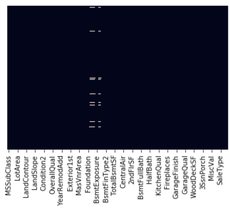
In [11]:
```
df['MasVnrType']=df['MasVnrType'].fillna(df['MasVnrType'].mode()[0])
df['MasVnrArea']=df['MasVnrArea'].fillna(df['MasVnrArea'].mode()[0])
```

In [12]:
```
df['BsmtExposure']=df['BsmtExposure'].fillna(df['BsmtExposure'].mode()[0])
```

In [13]:
```
df['BsmtFinType2']=df['BsmtFinType2'].fillna(df['BsmtFinType2'].mode()[0])
```

In [28]:
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False)
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x520f308>
```



In [14]:
```
df.dropna(inplace=True)
```

In [15]:
```
print(df.shape)
```

```
(1422, 75)
```

In [16]:
```
columns=['MSZoning','Street','LotShape','LandContour','Utilities','LotConfig','LandSlope','Neighborh
ood',
       'Condition2','BldgType','Condition1','HouseStyle','SaleType',
       'SaleCondition','ExterCond',
       'ExterQual','Foundation','BsmtQual','BsmtCond','BsmtExposure','BsmtFinType1','BsmtFinType2'
,
       'RoofStyle','RoofMatl','Exterior1st','Exterior2nd','MasVnrType','Heating','HeatingQC',
       'CentralAir',
       'Electrical','KitchenQual','Functional',
       'FireplaceQu','GarageType','GarageFinish','GarageQual','GarageCond','PavedDrive']
```

In [17]:
```
print(len(columns))
```

```
39
```

In [18]:
```
def category_onehot_multcols(multcolumns):
    df_final=final_df
    i=0
    for fields in multcolumns:

        print(fields)
        df1=pd.get_dummies(final_df[fields],drop_first=True)

        final_df.drop([fields],axis=1,inplace=True)
        if i==0:
            df_final=df1.copy()
        else:

            df_final=pd.concat([df_final,df1],axis=1)
        i=i+1


    df_final=pd.concat([final_df,df_final],axis=1)

    return df_final
```

In [19]:
```
main_df=df.copy()
```

In [20]:
```
###Combining Test Data
test_df=pd.read_csv('formulatedtest.csv')
```

In [21]:
```
final_df=pd.concat([df,test_df],axis=0)
```

```
C:\Users\Gopi\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: Sorting becaus
e non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  """Entry point for launching an IPython kernel.
```

In [22]:
```
print(final_df.shape)
```

```
(2881, 75)
```

In [23]:
```
final_df=category_onehot_multcols(columns)
```

```
MSZoning
Street
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition2
BldgType
Condition1
HouseStyle
SaleType
SaleCondition
ExterCond
ExterQual
Foundation
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrType
Heating
HeatingQC
CentralAir
Electrical
KitchenQual
Functional
FireplaceQu
GarageType
GarageFinish
GarageQual
GarageCond
PavedDrive
```

In [24]:
```
print(final_df.shape)
```

```
(2881, 235)
```

In [25]:
```
final_df =final_df.loc[:,~final_df.columns.duplicated()]
```

In [26]:
```
print(final_df.shape)
```

```
(2881, 175)
```

In [27]:
```
df_Train=final_df.iloc[:1422,:]
df_Test=final_df.iloc[1422:,:]
```

In [28]:
```
df_Test.drop(['SalePrice'],axis=1,inplace=True)
```

```
C:\Users\Gopi\Anaconda3\lib\site-packages\pandas\core\frame.py:4102: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/ind
exing.html#returning-a-view-versus-a-copy
  errors=errors,
```

In [29]:
```
X_train=df_Train.drop(['SalePrice'],axis=1)
y_train=df_Train['SalePrice']
```

MODEL BUILDING

In [31]:
```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
tree.fit(X_train,y_train)
```

Out[31]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=0, splitter='best')
```

In [33]:
```
y_pred=tree.predict(df_Test)
print(y_pred)
```

```
[ 94750. 158000. 216500. ... 168000. 135500. 275500.]
```

In [106]:
```
pred=pd.DataFrame(y_pred)
sub_df=pd.read_csv('C://Users/Gopi/Desktop/house/final_sub.csv')
datasets=pd.concat([sub_df['Id'],pred],axis=1)
```