Manvith S Rao

MIT - CCE-24

(training dataset)

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

df=pd.read_csv('train.csv')

df.head()
```

```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape
\
0   1          60       RL         65.0     8450   Pave   NaN      Reg

1   2          20       RL         80.0     9600   Pave   NaN      Reg

2   3          60       RL         68.0    11250   Pave   NaN      IR1

3   4          70       RL         60.0     9550   Pave   NaN      IR1

4   5          60       RL         84.0    14260   Pave   NaN      IR1


  LandContour Utilities  ... PoolArea PoolQC Fence MiscFeature MiscVal
MoSold  \
0         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
2
1         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
5
2         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
9
3         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
2
4         Lvl    AllPub  ...        0    NaN   NaN         NaN       0
12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000
```

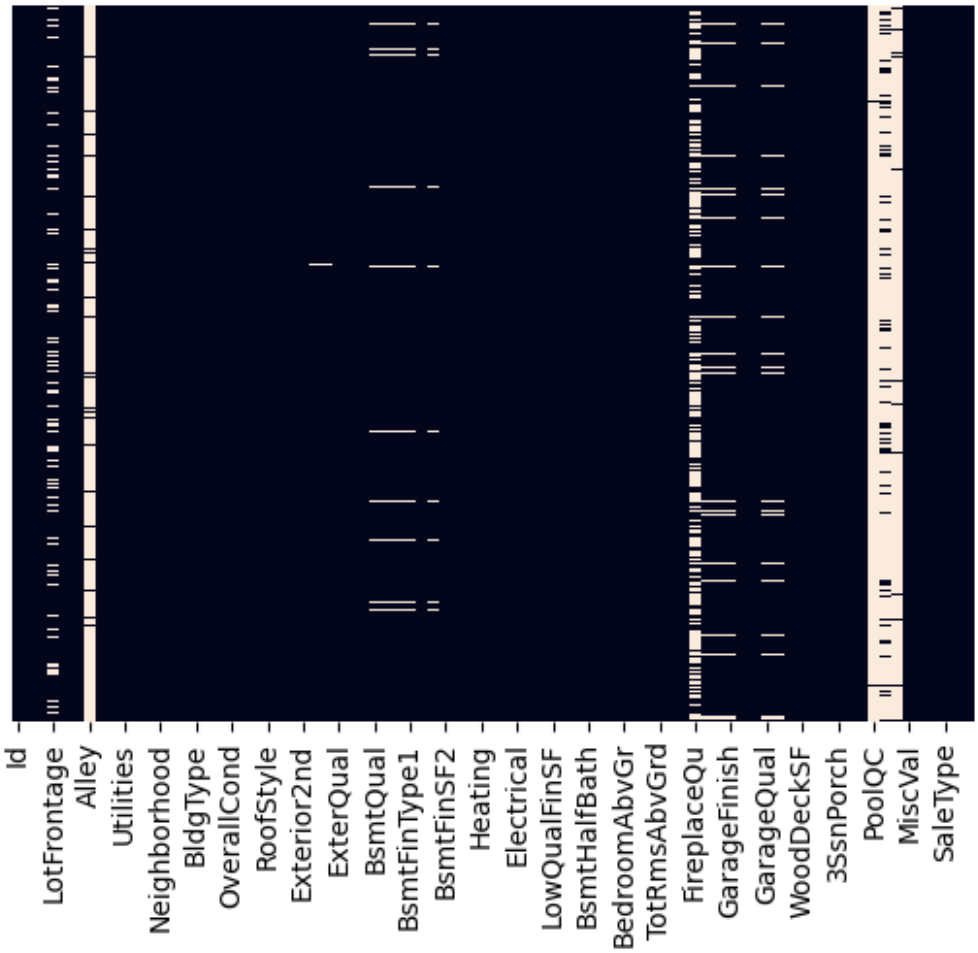
```
[5 rows x 81 columns]
```

```
df.isnull().sum()
```

```
Id                  0
MSSubClass          0
MSZoning            0
LotFrontage       259
LotArea             0
                 ...
MoSold              0
YrSold              0
SaleType            0
SaleCondition       0
SalePrice           0
Length: 81, dtype: int64
```

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False)
```

```
<Axes: >
```



```
df.shape
```

```
(1460, 81)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1460 non-null   int64  
 1   MSSubClass     1460 non-null   int64  
 2   MSZoning       1460 non-null   object 
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64  
 5   Street         1460 non-null   object 
 6   Alley          91 non-null     object 
 7   LotShape       1460 non-null   object 
 8   LandContour    1460 non-null   object 
 9   Utilities      1460 non-null   object 
 10  LotConfig      1460 non-null   object 
 11  LandSlope      1460 non-null   object 
 12  Neighborhood   1460 non-null   object 
 13  Condition1     1460 non-null   object 
 14  Condition2     1460 non-null   object 
 15  BldgType       1460 non-null   object 
 16  HouseStyle     1460 non-null   object 
 17  OverallQual    1460 non-null   int64  
 18  OverallCond    1460 non-null   int64  
 19  YearBuilt      1460 non-null   int64  
 20  YearRemodAdd   1460 non-null   int64  
 21  RoofStyle      1460 non-null   object 
 22  RoofMatl       1460 non-null   object 
 23  Exterior1st    1460 non-null   object 
 24  Exterior2nd    1460 non-null   object 
 25  MasVnrType     1452 non-null   object 
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object 
 28  ExterCond      1460 non-null   object 
 29  Foundation     1460 non-null   object 
 30  BsmtQual       1423 non-null   object 
 31  BsmtCond       1423 non-null   object 
 32  BsmtExposure   1422 non-null   object 
 33  BsmtFinType1   1423 non-null   object 
 34  BsmtFinSF1     1460 non-null   int64  
 35  BsmtFinType2   1422 non-null   object 
 36  BsmtFinSF2     1460 non-null   int64  
 37  BsmtUnfSF      1460 non-null   int64  
 38  TotalBsmtSF    1460 non-null   int64  
```

```
 39  Heating         1460 non-null   object
 40  HeatingQC       1460 non-null   object
 41  CentralAir      1460 non-null   object
 42  Electrical      1459 non-null   object
 43  1stFlrSF        1460 non-null   int64
 44  2ndFlrSF        1460 non-null   int64
 45  LowQualFinSF    1460 non-null   int64
 46  GrLivArea       1460 non-null   int64
 47  BsmtFullBath    1460 non-null   int64
 48  BsmtHalfBath    1460 non-null   int64
 49  FullBath        1460 non-null   int64
 50  HalfBath        1460 non-null   int64
 51  BedroomAbvGr    1460 non-null   int64
 52  KitchenAbvGr    1460 non-null   int64
 53  KitchenQual     1460 non-null   object
 54  TotRmsAbvGrd    1460 non-null   int64
 55  Functional      1460 non-null   object
 56  Fireplaces      1460 non-null   int64
 57  FireplaceQu     770 non-null    object
 58  GarageType      1379 non-null   object
 59  GarageYrBlt     1379 non-null   float64
 60  GarageFinish    1379 non-null   object
 61  GarageCars      1460 non-null   int64
 62  GarageArea      1460 non-null   int64
 63  GarageQual      1379 non-null   object
 64  GarageCond      1379 non-null   object
 65  PavedDrive      1460 non-null   object
 66  WoodDeckSF      1460 non-null   int64
 67  OpenPorchSF     1460 non-null   int64
 68  EnclosedPorch   1460 non-null   int64
 69  3SsnPorch       1460 non-null   int64
 70  ScreenPorch     1460 non-null   int64
 71  PoolArea        1460 non-null   int64
 72  PoolQC          7 non-null      object
 73  Fence           281 non-null    object
 74  MiscFeature     54 non-null     object
 75  MiscVal         1460 non-null   int64
 76  MoSold          1460 non-null   int64
 77  YrSold          1460 non-null   int64
 78  SaleType        1460 non-null   object
 79  SaleCondition   1460 non-null   object
 80  SalePrice       1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

Fill Missing Values

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

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

df['BsmtCond']=df['BsmtCond'].fillna(df['BsmtCond'].mode()[0])
df['BsmtQual']=df['BsmtQual'].fillna(df['BsmtQual'].mode()[0])

df['FireplaceQu']=df['FireplaceQu'].fillna(df['FireplaceQu'].mode()
[0])
df['GarageType']=df['GarageType'].fillna(df['GarageType'].mode()[0])

df.drop(['GarageYrBlt'],axis=1,inplace=True)

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])

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

df.shape
```

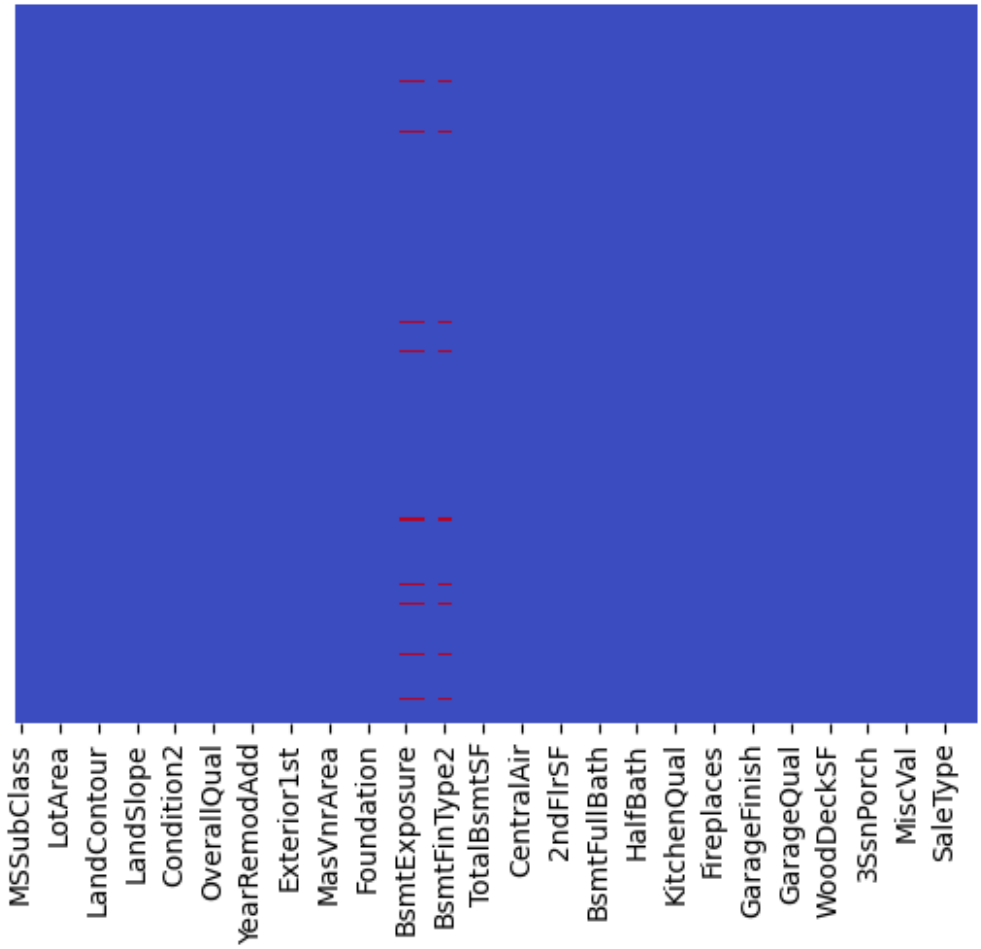
```
(1460, 76)
```

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


df.isnull().sum()
```

```
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
Street           0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 75, dtype: int64
```

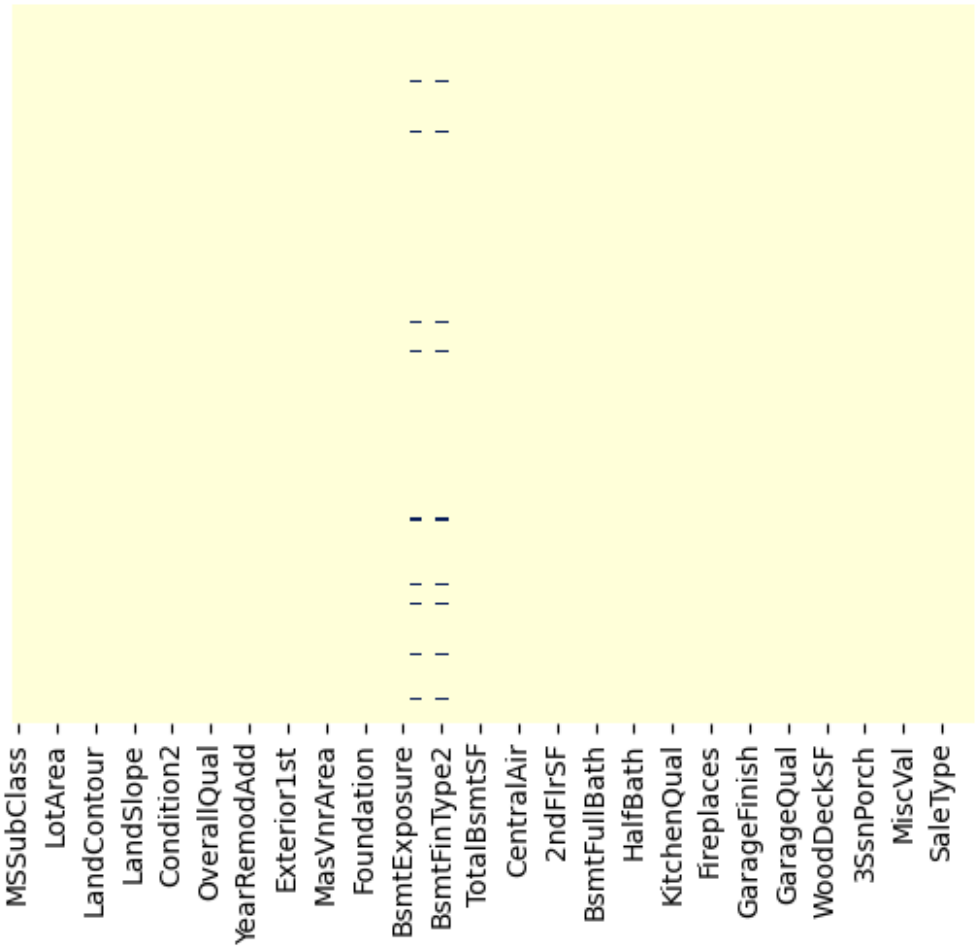
```
df['MasVnrType']=df['MasVnrType'].fillna(df['MasVnrType'].mode()[0])
df['MasVnrArea']=df['MasVnrArea'].fillna(df['MasVnrArea'].mode()[0])

sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='coolwarm')
```

```
<Axes: >
```




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

sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='YlGnBu')
```

```
<Axes: >
```




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

df.dropna(inplace=True)

df.shape

(1422, 75)

df.head()

   MSSubClass MSZoning  LotFrontage  LotArea Street LotShape
LandContour  \
0          60       RL         65.0     8450   Pave      Reg
Lvl
1          20       RL         80.0     9600   Pave      Reg
Lvl
2          60       RL         68.0    11250   Pave      IR1
Lvl
3          70       RL         60.0     9550   Pave      IR1
Lvl
```

```
4          60       RL         84.0    14260   Pave      IR1
Lvl

  Utilities LotConfig LandSlope  ... EnclosedPorch 3SsnPorch
ScreenPorch  \
0    AllPub    Inside       Gtl  ...             0         0
0
1    AllPub       FR2       Gtl  ...             0         0
0
2    AllPub    Inside       Gtl  ...             0         0
0
3    AllPub    Corner       Gtl  ...           272         0
0
4    AllPub       FR2       Gtl  ...             0         0
0

  PoolArea MiscVal  MoSold  YrSold  SaleType  SaleCondition SalePrice

0        0       0       2    2008        WD         Normal    208500

1        0       0       5    2007        WD         Normal    181500

2        0       0       9    2008        WD         Normal    223500

3        0       0       2    2006        WD        Abnorml    140000

4        0       0      12    2008        WD         Normal    250000


[5 rows x 75 columns]
```

##Handle Categorical Features

```
columns=['MSZoning','Street','LotShape','LandContour','Utilities','Lot
Config','LandSlope','Neighborhood',
         'Condition2','BldgType','Condition1','HouseStyle','SaleType',
        'SaleCondition','ExterCond',

'ExterQual','Foundation','BsmtQual','BsmtCond','BsmtExposure','BsmtFin
Type1','BsmtFinType2',

'RoofStyle','RoofMatl','Exterior1st','Exterior2nd','MasVnrType','Heati
ng','HeatingQC',
         'CentralAir',
         'Electrical','KitchenQual','Functional',

'FireplaceQu','GarageType','GarageFinish','GarageQual','GarageCond','P
avedDrive']

len(columns)
```

```
39
```

```
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
```

Combine Test Data

```
test_df=pd.read_csv('ModifiedTest.csv')
```

```
test_df.shape
```

```
(1459, 74)
```

```
test_df.head()
```

```
   MSSubClass MSZoning  LotFrontage  LotArea Street LotShape
LandContour  \
0          20       RH         80.0    11622   Pave      Reg
Lvl
1          20       RL         81.0    14267   Pave      IR1
Lvl
2          60       RL         74.0    13830   Pave      IR1
Lvl
3          60       RL         78.0     9978   Pave      IR1
Lvl
4         120       RL         43.0     5005   Pave      IR1
HLS

  Utilities LotConfig LandSlope  ... OpenPorchSF EnclosedPorch
3SsnPorch  \
0    AllPub    Inside       Gtl  ...           0             0
0
1    AllPub    Corner       Gtl  ...          36             0
```

```
0
2    AllPub    Inside       Gtl  ...          34             0
0
3    AllPub    Inside       Gtl  ...          36             0
0
4    AllPub    Inside       Gtl  ...          82             0
0

  ScreenPorch PoolArea  MiscVal  MoSold  YrSold  SaleType
SaleCondition
0         120        0        0       6    2010        WD
Normal
1           0        0    12500       6    2010        WD
Normal
2           0        0        0       3    2010        WD
Normal
3           0        0        0       6    2010        WD
Normal
4         144        0        0       1    2010        WD
Normal

[5 rows x 74 columns]
```

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

```
final_df['SalePrice']
```

```
0       208500.0
1       181500.0
2       223500.0
3       140000.0
4       250000.0
          ...
1454         NaN
1455         NaN
1456         NaN
1457         NaN
1458         NaN
Name: SalePrice, Length: 2881, dtype: float64
```

```
final_df.shape
```

```
(2881, 75)
```

```
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

final_df.shape

(2881, 235)

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

final_df.shape

(2881, 175)

final_df

      MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond
YearBuilt  \
0             60         65.0     8450            7            5
2003
```

```
1             20         80.0     9600            6            8
1976
2             60         68.0    11250            7            5
2001
3             70         60.0     9550            7            5
1915
4             60         84.0    14260            8            5
2000
...          ...          ...      ...          ...          ...
...
1454         160         21.0     1936            4            7
1970
1455         160         21.0     1894            4            5
1970
1456          20        160.0    20000            5            7
1960
1457          85         62.0    10441            5            5
1992
1458          60         74.0     9627            7            5
1993

      YearRemodAdd  MasVnrArea  BsmtFinSF1  BsmtFinSF2  ...  Min1
Min2  Typ  \
0             2003       196.0       706.0         0.0  ...     0
0    1
1             1976         0.0       978.0         0.0  ...     0
0    1
2             2002       162.0       486.0         0.0  ...     0
0    1
3             1970         0.0       216.0         0.0  ...     0
0    1
4             2000       350.0       655.0         0.0  ...     0
0    1
...            ...         ...         ...         ...  ...   ...   ..
.  ...
1454          1970         0.0         0.0         0.0  ...     0
0    1
1455          1970         0.0       252.0         0.0  ...     0
0    1
1456          1996         0.0      1224.0         0.0  ...     0
0    1
1457          1992         0.0       337.0         0.0  ...     0
0    1
1458          1994        94.0       758.0         0.0  ...     0
0    1

      Attchd  Basment  BuiltIn  CarPort  Detchd  RFn  P
0          1        0        0        0       0    1  0
1          1        0        0        0       0    1  0
```

```
2          1        0        0        0       0    1  0
3          0        0        0        0       1    0  0
4          1        0        0        0       0    1  0
...      ...      ...      ...      ...     ...  ...  ..
1454       1        0        0        0       0    0  0
1455       0        0        0        1       0    0  0
1456       0        0        0        0       1    0  0
1457       1        0        0        0       0    0  0
1458       1        0        0        0       0    0  0

[2881 rows x 175 columns]
```

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

df_Train.head()
```

```
   MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond
YearBuilt  \
0          60         65.0     8450            7            5
2003
1          20         80.0     9600            6            8
1976
2          60         68.0    11250            7            5
2001
3          70         60.0     9550            7            5
1915
4          60         84.0    14260            8            5
2000

   YearRemodAdd  MasVnrArea  BsmtFinSF1  BsmtFinSF2  ...  Min1  Min2
Typ  \
0          2003       196.0       706.0         0.0  ...     0     0
1
1          1976         0.0       978.0         0.0  ...     0     0
1
2          2002       162.0       486.0         0.0  ...     0     0
1
3          1970         0.0       216.0         0.0  ...     0     0
1
4          2000       350.0       655.0         0.0  ...     0     0
1

   Attchd  Basment  BuiltIn  CarPort  Detchd  RFn  P
0       1        0        0        0       0    1  0
1       1        0        0        0       0    1  0
2       1        0        0        0       0    1  0
3       0        0        0        0       1    0  0
4       1        0        0        0       0    1  0
```

```
[5 rows x 175 columns]
```

```
df_Test.head()
```

```
   MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond
YearBuilt  \
0          20         80.0    11622            5            6
1961
1          20         81.0    14267            6            6
1958
2          60         74.0    13830            5            5
1997
3          60         78.0     9978            6            6
1998
4         120         43.0     5005            8            5
1992

   YearRemodAdd  MasVnrArea  BsmtFinSF1  BsmtFinSF2  ...  Min1  Min2
Typ  \
0          1961         0.0       468.0       144.0  ...     0     0
1
1          1958       108.0       923.0         0.0  ...     0     0
1
2          1998         0.0       791.0         0.0  ...     0     0
1
3          1998        20.0       602.0         0.0  ...     0     0
1
4          1992         0.0       263.0         0.0  ...     0     0
1

   Attchd  Basment  BuiltIn  CarPort  Detchd  RFn  P
0       1        0        0        0       0    0  0
1       1        0        0        0       0    0  0
2       1        0        0        0       0    0  0
3       1        0        0        0       0    0  0
4       1        0        0        0       0    1  0

[5 rows x 175 columns]
```

```
df_Train.shape
```

```
(1422, 175)
```

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

```
<ipython-input-48-8fdc58f80b2f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

```
returning-a-view-versus-a-copy
  df_Test.drop(['SalePrice'],axis=1,inplace=True)

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

##Prediciton and selecting the Algorithm

```
from sklearn.linear_model import LinearRegression
Linear = LinearRegression()
Linear.fit(X_train,y_train)
Linear.score(X_train,y_train)

0.9123193308652172

from sklearn.ensemble import RandomForestRegressor
random_model = RandomForestRegressor()
random_model.fit(X_train, y_train)
random_model.score(X_train, y_train)

0.9798560184871163

from xgboost import XGBRegressor
xgb_model = XGBRegressor()
xgb_model.fit(X_train, y_train)
xgb_model.score(X_train,y_train)

0.9994999305968802

import xgboost
classifier=xgboost.XGBRegressor()
classifier.fit(X_train,y_train)

XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None,
early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None,
feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None,
max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan,
monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)

y_pred=classifier.predict(df_Test)
```

```
y_pred
```

```
array([131608.7 , 147968.84, 204317.55, ..., 168148.14, 104032.52,
       231323.73], dtype=float32)
```