Machine Learning Assignment No.2

Que.1 By Taking reference of the Housing Price Dataset plot each independent variable with the dependent variable and store the name of independent variable in a list which show non linear behavior

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

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/housing.csv")

df
```

```
        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
...    ...         ...      ...          ...      ...    ...   ...
...
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
```

```
3            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
...          ...       ...  ...      ...    ...    ...         ...
...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
...     ...    ...       ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[1460 rows x 81 columns]

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.tail()

```
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley
LotShape  \
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
```

```
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[5 rows x 81 columns]
```

```
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

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
```

```
df = df.drop("LotFrontage",axis=1) # Independent variable
```

```
df
```

```
        Id  MSSubClass MSZoning  LotArea Street Alley LotShape
LandContour  \
0        1          60       RL     8450   Pave   NaN      Reg
Lvl
1        2          20       RL     9600   Pave   NaN      Reg
Lvl
2        3          60       RL    11250   Pave   NaN      IR1
Lvl
3        4          70       RL     9550   Pave   NaN      IR1
Lvl
4        5          60       RL    14260   Pave   NaN      IR1
Lvl
...    ...         ...      ...      ...    ...   ...      ...
...
1455  1456          60       RL     7917   Pave   NaN      Reg
Lvl
1456  1457          20       RL    13175   Pave   NaN      Reg
Lvl
1457  1458          70       RL     9042   Pave   NaN      Reg
Lvl
1458  1459          20       RL     9717   Pave   NaN      Reg
Lvl
1459  1460          20       RL     9937   Pave   NaN      Reg
Lvl

     Utilities LotConfig  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0       AllPub    Inside  ...        0    NaN    NaN         NaN
0
1       AllPub       FR2  ...        0    NaN    NaN         NaN
0
2       AllPub    Inside  ...        0    NaN    NaN         NaN
0
3       AllPub    Corner  ...        0    NaN    NaN         NaN
0
4       AllPub       FR2  ...        0    NaN    NaN         NaN
```

```
0
...        ...       ...  ...      ...    ...    ...          ...     .
..
1455    AllPub    Inside  ...        0    NaN    NaN         NaN
0
1456    AllPub    Inside  ...        0    NaN  MnPrv         NaN
0
1457    AllPub    Inside  ...        0    NaN  GdPrv        Shed
2500
1458    AllPub    Inside  ...        0    NaN    NaN         NaN
0
1459    AllPub    Inside  ...        0    NaN    NaN         NaN
0

     MoSold  YrSold  SaleType  SaleCondition  SalePrice
0         2    2008        WD         Normal     208500
1         5    2007        WD         Normal     181500
2         9    2008        WD         Normal     223500
3         2    2006        WD        Abnorml     140000
4        12    2008        WD         Normal     250000
...     ...     ...       ...            ...        ...
1455      8    2007        WD         Normal     175000
1456      2    2010        WD         Normal     210000
1457      5    2010        WD         Normal     266500
1458      4    2010        WD         Normal     142125
1459      6    2008        WD         Normal     147500

[1460 rows x 80 columns]
```

```
df.shape
```

```
(1460, 80)
```

```
X = df.drop("SalePrice",axis = 1) # Independent variable
X
```

```
        Id  MSSubClass MSZoning  LotArea Street Alley LotShape
LandContour  \
0        1          60       RL     8450   Pave   NaN      Reg
Lvl
1        2          20       RL     9600   Pave   NaN      Reg
Lvl
2        3          60       RL    11250   Pave   NaN      IR1
Lvl
3        4          70       RL     9550   Pave   NaN      IR1
Lvl
4        5          60       RL    14260   Pave   NaN      IR1
Lvl
...    ...         ...      ...      ...    ...   ...      ...
...
1455  1456          60       RL     7917   Pave   NaN      Reg
```

```
Lvl
1456  1457          20       RL    13175   Pave   NaN      Reg
Lvl
1457  1458          70       RL     9042   Pave   NaN      Reg
Lvl
1458  1459          20       RL     9717   Pave   NaN      Reg
Lvl
1459  1460          20       RL     9937   Pave   NaN      Reg
Lvl

     Utilities LotConfig  ... ScreenPorch PoolArea PoolQC  Fence
MiscFeature  \
0       AllPub    Inside  ...           0        0    NaN    NaN
NaN
1       AllPub       FR2  ...           0        0    NaN    NaN
NaN
2       AllPub    Inside  ...           0        0    NaN    NaN
NaN
3       AllPub    Corner  ...           0        0    NaN    NaN
NaN
4       AllPub       FR2  ...           0        0    NaN    NaN
NaN
...        ...       ...  ...         ...      ...    ...    ...
...
1455    AllPub    Inside  ...           0        0    NaN    NaN
NaN
1456    AllPub    Inside  ...           0        0    NaN  MnPrv
NaN
1457    AllPub    Inside  ...           0        0    NaN  GdPrv
Shed
1458    AllPub    Inside  ...           0        0    NaN    NaN
NaN
1459    AllPub    Inside  ...           0        0    NaN    NaN
NaN

     MiscVal  MoSold  YrSold  SaleType  SaleCondition
0          0       2    2008        WD         Normal
1          0       5    2007        WD         Normal
2          0       9    2008        WD         Normal
3          0       2    2006        WD        Abnorml
4          0      12    2008        WD         Normal
...      ...     ...     ...       ...            ...
1455       0       8    2007        WD         Normal
1456       0       2    2010        WD         Normal
1457    2500       5    2010        WD         Normal
1458       0       4    2010        WD         Normal
1459       0       6    2008        WD         Normal

[1460 rows x 79 columns]
```

```
Y = df["SalePrice"] #dependent variables only target column will be in
Y
```

```
Y
```

```
0       208500
1       181500
2       223500
3       140000
4       250000
         ...
1455    175000
1456    210000
1457    266500
1458    142125
1459    147500
Name: SalePrice, Length: 1460, dtype: int64
```

```
Y =  df[["SalePrice"]].values # Dependent Variable
```

```
Y
```

```
array([[208500],
       [181500],
       [223500],
       ...,
       [266500],
       [142125],
       [147500]], dtype=int64)
```

```
X.head()
```

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

1   2          20       RL     9600   Pave   NaN      Reg         Lvl

2   3          60       RL    11250   Pave   NaN      IR1         Lvl

3   4          70       RL     9550   Pave   NaN      IR1         Lvl

4   5          60       RL    14260   Pave   NaN      IR1         Lvl


  Utilities LotConfig  ... ScreenPorch PoolArea PoolQC Fence
MiscFeature  \
0    AllPub    Inside  ...           0        0    NaN   NaN
NaN
1    AllPub       FR2  ...           0        0    NaN   NaN
NaN
```

```
2    AllPub    Inside  ...           0        0    NaN   NaN
NaN
3    AllPub    Corner  ...           0        0    NaN   NaN
NaN
4    AllPub       FR2  ...           0        0    NaN   NaN
NaN

  MiscVal  MoSold  YrSold  SaleType  SaleCondition
0       0       2    2008        WD         Normal
1       0       5    2007        WD         Normal
2       0       9    2008        WD         Normal
3       0       2    2006        WD        Abnorml
4       0      12    2008        WD         Normal

[5 rows x 79 columns]
```

```
# separate dataset into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(
    X,
    Y,
    test_size=0.3,
    random_state=50)

X_train.shape, X_test.shape
```

```
((1022, 79), (438, 79))
```

```
X_train
```

```
        Id  MSSubClass MSZoning  LotArea Street Alley LotShape
LandContour  \
175    176          20       RL    12615   Pave   NaN      Reg
Lvl
1408  1409          70       RM     7740   Pave   NaN      Reg
Lvl
1148  1149          50       RM     5700   Pave   NaN      Reg
Lvl
861    862         190       RL    11625   Pave   NaN      Reg
Lvl
220    221          20       RL     8990   Pave   NaN      IR1
Lvl
...    ...         ...      ...      ...    ...   ...      ...
...
229    230         120       RL     3182   Pave   NaN      Reg
Lvl
70      71          20       RL    13651   Pave   NaN      IR1
Lvl
132    133          20       RL     7388   Pave   NaN      Reg
Lvl
1313  1314          60       RL    14774   Pave   NaN      IR1
```

```
Lvl
109    110          20       RL    11751   Pave   NaN      IR1
Lvl

     Utilities LotConfig  ... ScreenPorch PoolArea PoolQC  Fence
MiscFeature  \
175     AllPub    Corner  ...           0        0    NaN  MnPrv
NaN
1408    AllPub    Inside  ...         168        0    NaN    NaN
NaN
1148    AllPub    Inside  ...           0        0    NaN    NaN
NaN
861     AllPub    Inside  ...           0        0    NaN    NaN
NaN
220     AllPub    Inside  ...           0        0    NaN    NaN
NaN
...        ...       ...  ...         ...      ...    ...    ...
...
229     AllPub    Inside  ...           0        0    NaN    NaN
NaN
70      AllPub    Inside  ...           0        0    NaN    NaN
NaN
132     AllPub    Corner  ...           0        0    NaN    NaN
NaN
1313    AllPub    Corner  ...           0        0    NaN    NaN
NaN
109     AllPub    Inside  ...           0        0    NaN  MnPrv
NaN

     MiscVal  MoSold  YrSold  SaleType  SaleCondition
175        0       6    2007        WD         Normal
1408       0       6    2010        WD         Normal
1148       0       8    2008        WD         Normal
861        0       4    2010        WD         Normal
220        0       4    2006       New        Partial
...      ...     ...     ...       ...            ...
229        0       5    2009        WD         Normal
70         0       2    2007        WD         Normal
132        0       7    2007        WD         Normal
1313       0       5    2010        WD         Normal
109        0       1    2010       COD         Normal

[1022 rows x 79 columns]
```

```
X_train.corr()
```

```
                    Id  MSSubClass   LotArea  OverallQual
OverallCond  \
Id             1.000000    0.025180 -0.042861    -0.017525
0.027990
```

```
MSSubClass     0.025180    1.000000 -0.114367     0.018828    -
0.078168
LotArea       -0.042861   -0.114367  1.000000     0.097441    -
0.002465
OverallQual   -0.017525    0.018828  0.097441     1.000000    -
0.111293
OverallCond    0.027990   -0.078168 -0.002465    -0.111293
1.000000
YearBuilt     -0.012989    0.042625  0.015185     0.577101    -
0.387111
YearRemodAdd  -0.013134    0.053252  0.026488     0.555213
0.078987
MasVnrArea    -0.068046    0.025975  0.104614     0.414165    -
0.114867
BsmtFinSF1    -0.006606   -0.079230  0.224503     0.230208    -
0.056106
BsmtFinSF2    -0.022122   -0.088294  0.116479    -0.042107
0.049775
BsmtUnfSF      0.016729   -0.129546 -0.013383     0.307859    -
0.138530
TotalBsmtSF    0.001622   -0.245195  0.266075     0.534580    -
0.179180
1stFlrSF      -0.000038   -0.256829  0.293272     0.470704    -
0.154743
2ndFlrSF      -0.006532    0.292979  0.042529     0.303157
0.005361
LowQualFinSF  -0.048756    0.059712 -0.007210    -0.036297
0.043748
GrLivArea     -0.009664    0.057437  0.251388     0.595486    -
0.106192
BsmtFullBath  -0.000926    0.000866  0.171255     0.085237    -
0.079731
BsmtHalfBath  -0.044217   -0.001635  0.049252    -0.037835
0.121766
FullBath      -0.011373    0.113428  0.140051     0.552090    -
0.216317
HalfBath       0.002378    0.182245  0.001082     0.276522    -
0.071881
BedroomAbvGr   0.008400   -0.033968  0.128956     0.107916    -
0.013135
KitchenAbvGr   0.006342    0.236044 -0.009992    -0.195035    -
0.106891
TotRmsAbvGrd  -0.000633    0.027472  0.182974     0.435719    -
0.101754
Fireplaces    -0.036304   -0.083865  0.280601     0.385690    -
0.045240
GarageYrBlt   -0.003349    0.110443 -0.032688     0.557026    -
0.302170
GarageCars     0.018369   -0.034364  0.140119     0.605606    -
0.202612
```

```
GarageArea     0.015167   -0.093700  0.165381     0.577165    -
0.161514
WoodDeckSF    -0.045645    0.010248  0.169479     0.247537    -
0.002512
OpenPorchSF    0.004472    0.013227  0.069652     0.299579    -
0.026936
EnclosedPorch  0.013902   -0.011002 -0.024140    -0.105834
0.106672
3SsnPorch     -0.075026   -0.064610  0.011642     0.016132
0.024791
ScreenPorch    0.010995   -0.042803  0.028306     0.063606
0.036087
PoolArea       0.020920   -0.007767  0.088009     0.101894    -
0.028996
MiscVal       -0.004467    0.000907  0.035375    -0.035628
0.079555
MoSold         0.024806   -0.006537 -0.013137     0.066265
0.002778
YrSold         0.009276   -0.021893 -0.014502    -0.014671
0.048223

               YearBuilt  YearRemodAdd  MasVnrArea  BsmtFinSF1
BsmtFinSF2  \
Id             -0.012989     -0.013134   -0.068046   -0.006606   -
0.022122
MSSubClass      0.042625      0.053252    0.025975   -0.079230   -
0.088294
LotArea         0.015185      0.026488    0.104614    0.224503
0.116479
OverallQual     0.577101      0.555213    0.414165    0.230208   -
0.042107
OverallCond    -0.387111      0.078987   -0.114867   -0.056106
0.049775
YearBuilt       1.000000      0.598748    0.308525    0.252258   -
0.046764
YearRemodAdd    0.598748      1.000000    0.171563    0.136949   -
0.057883
MasVnrArea      0.308525      0.171563    1.000000    0.263987   -
0.063710
BsmtFinSF1      0.252258      0.136949    0.263987    1.000000   -
0.042417
BsmtFinSF2     -0.046764     -0.057883   -0.063710   -0.042417
1.000000
BsmtUnfSF       0.145947      0.147399    0.114849   -0.502190   -
0.208535
TotalBsmtSF     0.394536      0.270313    0.369768    0.537806
0.113599
1stFlrSF        0.285877      0.232418    0.355018    0.454542
0.117595
2ndFlrSF        0.029176      0.161668    0.176942   -0.141532   -
```

```
0.113736
LowQualFinSF   -0.169594     -0.068087   -0.070230   -0.042055
0.011049
GrLivArea       0.220770      0.299572    0.401767    0.215477   -
0.006091
BsmtFullBath    0.184121      0.110194    0.066960    0.647734
0.165378
BsmtHalfBath   -0.047195     -0.010370    0.042605    0.044331
0.057414
FullBath        0.475584      0.444591    0.283784    0.054317   -
0.083134
HalfBath        0.260997      0.194512    0.190178   -0.003646   -
0.023659
BedroomAbvGr   -0.048401     -0.039360    0.104784   -0.102588   -
0.016352
KitchenAbvGr   -0.159160     -0.164545   -0.035418   -0.082943   -
0.036257
TotRmsAbvGrd    0.117424      0.197051    0.295755    0.065795   -
0.036636
Fireplaces      0.155521      0.116917    0.263209    0.266428
0.054814
GarageYrBlt     0.813789      0.651013    0.231085    0.150479   -
0.097642
GarageCars      0.527617      0.418037    0.346576    0.218608   -
0.028433
GarageArea      0.477330      0.387689    0.362806    0.298668   -
0.017958
WoodDeckSF      0.228017      0.226514    0.207229    0.177238
0.057041
OpenPorchSF     0.201400      0.256033    0.103869    0.110250
0.014051
EnclosedPorch  -0.416450     -0.177672   -0.131871   -0.109477
0.026815
3SsnPorch       0.011016      0.025054    0.032758   -0.004082   -
0.030257
ScreenPorch    -0.009983     -0.026060    0.047905    0.064097
0.091784
PoolArea        0.005994      0.039571    0.035591    0.236978
0.042001
MiscVal        -0.039418     -0.009489   -0.040046   -0.004573   -
0.002470
MoSold          0.025161      0.022081   -0.020837   -0.043953   -
0.012785
YrSold         -0.029227      0.034888   -0.018832    0.019368
0.021047

               ...  GarageArea  WoodDeckSF  OpenPorchSF  EnclosedPorch
\
Id             ...    0.015167   -0.045645     0.004472       0.013902
```

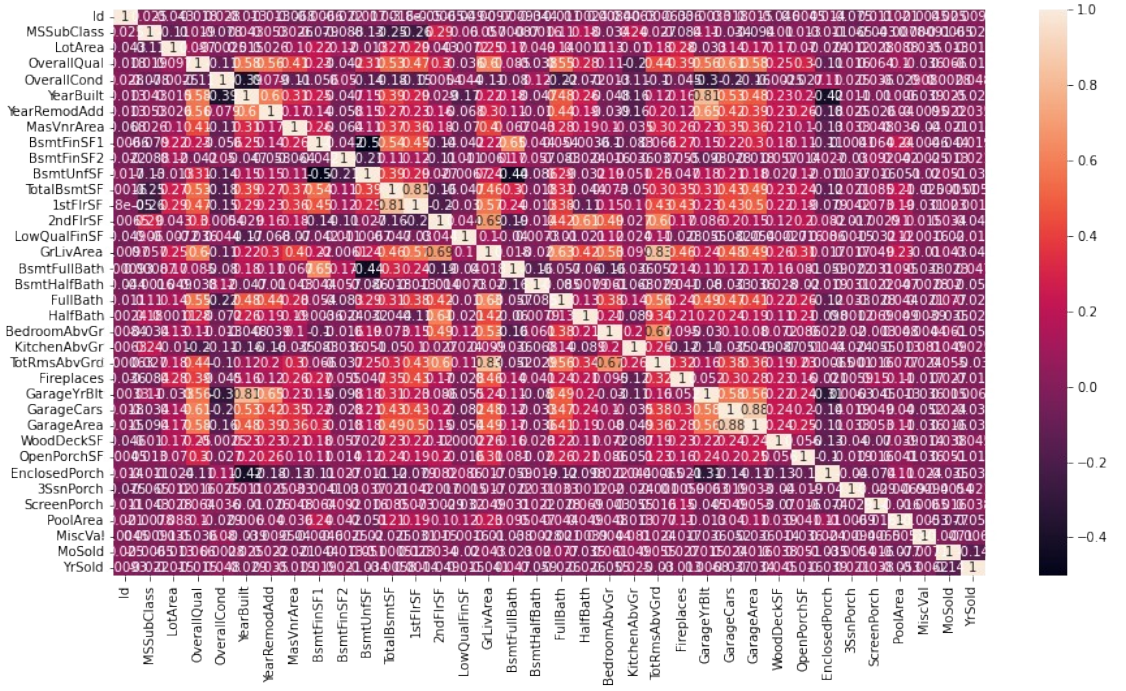
MSSubClass	...	-0.093700	0.010248	0.013227	-0.011002
LotArea	...	0.165381	0.169479	0.069652	-0.024140
OverallQual	...	0.577165	0.247537	0.299579	-0.105834
OverallCond	...	-0.161514	-0.002512	-0.026936	0.106672
YearBuilt	...	0.477330	0.228017	0.201400	-0.416450
YearRemodAdd	...	0.387689	0.226514	0.256033	-0.177672
MasVnrArea	...	0.362806	0.207229	0.103869	-0.131871
BsmtFinSF1	...	0.298668	0.177238	0.110250	-0.109477
BsmtFinSF2	...	-0.017958	0.057041	0.014051	0.026815
BsmtUnfSF	...	0.183632	0.026537	0.118611	-0.011277
TotalBsmtSF	...	0.491643	0.234292	0.239787	-0.116877
1stFlrSF	...	0.497726	0.222607	0.189023	-0.079051
2ndFlrSF	...	0.152533	0.117434	0.204863	0.081864
LowQualFinSF	...	-0.054352	-0.000270	-0.015534	0.085925
GrLivArea	...	0.489400	0.261650	0.307754	0.016677
BsmtFullBath	...	0.169224	0.155536	0.081407	-0.059105
BsmtHalfBath	...	-0.036467	0.028290	-0.020259	-0.019276
FullBath	...	0.410229	0.222691	0.259877	-0.117271
HalfBath	...	0.193076	0.111364	0.212537	-0.097562
BedroomAbvGr	...	0.079784	0.071997	0.085638	0.021797
KitchenAbvGr	...	-0.049193	-0.087300	-0.050643	0.043848
TotRmsAbvGrd	...	0.358110	0.194187	0.229790	0.000650
Fireplaces	...	0.275484	0.228026	0.158219	-0.020807
GarageYrBlt	...	0.562112	0.220969	0.241712	-0.314456

GarageCars	...	0.879030	0.242454	0.198639	-0.136279
GarageArea	...	1.000000	0.239356	0.253631	-0.108896
WoodDeckSF	...	0.239356	1.000000	0.055852	-0.132687
OpenPorchSF	...	0.253631	0.055852	1.000000	-0.104518
EnclosedPorch	...	-0.108896	-0.132687	-0.104518	1.000000
3SsnPorch	...	0.033450	-0.040471	-0.019403	-0.039606
ScreenPorch	...	0.053370	-0.070333	0.016460	-0.074408
PoolArea	...	0.106756	0.039455	0.041268	0.109545
MiscVal	...	-0.035755	-0.014213	-0.036254	0.023738
MoSold	...	0.015672	0.038213	0.051491	-0.035075
YrSold	...	-0.033988	0.045145	-0.016294	-0.039395

	3SsnPorch	ScreenPorch	PoolArea	MiscVal	MoSold	YrSold
Id	-0.075026	0.010995	0.020920	-0.004467	0.024806	0.009276
MSSubClass	-0.064610	-0.042803	-0.007767	0.000907	-0.006537	-0.021893
LotArea	0.011642	0.028306	0.088009	0.035375	-0.013137	-0.014502
OverallQual	0.016132	0.063606	0.101894	-0.035628	0.066265	-0.014671
OverallCond	0.024791	0.036087	-0.028996	0.079555	0.002778	0.048223
YearBuilt	0.011016	-0.009983	0.005994	-0.039418	0.025161	-0.029227
YearRemodAdd	0.025054	-0.026060	0.039571	-0.009489	0.022081	0.034888
MasVnrArea	0.032758	0.047905	0.035591	-0.040046	-0.020837	-0.018832
BsmtFinSF1	-0.004082	0.064097	0.236978	-0.004573	-0.043953	0.019368
BsmtFinSF2	-0.030257	0.091784	0.042001	-0.002470	-0.012785	0.021047
BsmtUnfSF	0.036599	-0.016374	-0.050625	-0.019717	0.050622	-0.034031
TotalBsmtSF	0.021133	0.084885	0.214770	-0.025406	-0.000514	-0.005826

```
1stFlrSF        0.041808     0.072976  0.188889 -0.030553  0.022552
0.001376
2ndFlrSF       -0.016551    -0.002855  0.100149  0.015084  0.034486 -
0.049469
LowQualFinSF    0.001540    -0.031614  0.116828 -0.001580 -0.019975 -
0.015037
GrLivArea       0.017369     0.048856  0.232607 -0.010262  0.043444 -
0.041178
BsmtFullBath   -0.021664     0.030538  0.094942 -0.038425 -0.022647
0.047220
BsmtHalfBath    0.031338     0.021563  0.047085 -0.002763  0.019802 -
0.058886
FullBath        0.033250    -0.028500  0.044232 -0.020825  0.076741 -
0.025906
HalfBath        0.001197     0.069264  0.048676  0.003928 -0.034601 -
0.026238
BedroomAbvGr    0.019734    -0.003015  0.048259  0.004437  0.061194 -
0.054573
KitchenAbvGr   -0.023562    -0.055461 -0.013256  0.080628  0.048652
0.025447
TotRmsAbvGrd   -0.001018     0.015995  0.076767  0.023662  0.054622 -
0.029951
Fireplaces      0.005923     0.149292  0.105683 -0.017496  0.026723 -
0.013111
GarageYrBlt     0.006252    -0.044522 -0.012544 -0.035604  0.014729
0.006770
GarageCars      0.019429     0.049214  0.039868 -0.051857  0.023925 -
0.037423
GarageArea      0.033450     0.053370  0.106756 -0.035755  0.015672 -
0.033988
WoodDeckSF     -0.040471    -0.070333  0.039455 -0.014213  0.038213
0.045145
OpenPorchSF    -0.019403     0.016460  0.041268 -0.036254  0.051491 -
0.016294
EnclosedPorch  -0.039606    -0.074408  0.109545  0.023738 -0.035075 -
0.039395
3SsnPorch       1.000000    -0.028835 -0.006892 -0.009439 -0.005406
0.020883
ScreenPorch    -0.028835     1.000000 -0.016223 -0.006470  0.015864
0.038272
PoolArea       -0.006892    -0.016223  1.000000 -0.005311 -0.077185 -
0.052822
MiscVal        -0.009439    -0.006470 -0.005311  1.000000 -0.007057 -
0.006154
MoSold         -0.005406     0.015864 -0.077185 -0.007057  1.000000 -
0.141790
YrSold          0.020883     0.038272 -0.052822 -0.006154 -0.141790
1.000000

[36 rows x 36 columns]
```

```
import seaborn as sns
import matplotlib.pyplot as plt
#Using Pearson Correlation
plt.figure(figsize=(15,8))
cor = X_train.corr()
sns.heatmap(cor, annot=True)
plt.show()
```




```
# with the following function we can select highly correlated features
# it will remove the first feature that is correlated with anything
other feature

def correlation(dataset, threshold):# X_train,0.5
    col_corr = set()  # Set of all the names of correlated columns
    col_corr_lst = []
    print(f"set initial {col_corr}")
    print(f"list initial {col_corr_lst}")
    corr_arr = dataset.corr() # corr_arr is my correlaion matrix which
is 2d
    for row in range(len(corr_arr)):
        for col in range(row):
            if abs(corr_arr.iloc[row, col]) > threshold: # we are
interested in absolute coeff value
                colname = corr_arr.columns[row]  # getting the name of
column
                col_corr_lst.append(colname)
                col_corr.add(colname)
                print(f"colname name which is correlated is
{colname}")
```

```
                print(f"set {col_corr}")
                print(f"lst {col_corr_lst}")


    print(f"list is {col_corr_lst}")
    return col_corr

corr_features = correlation(X_train, 0.3)#data,threshold
len(set(corr_features))
```

```
set initial set()
list initial []
colname name which is correlated is YearBuilt
set {'YearBuilt'}
lst ['YearBuilt']
colname name which is correlated is YearBuilt
set {'YearBuilt'}
lst ['YearBuilt', 'YearBuilt']
colname name which is correlated is YearRemodAdd
set {'YearRemodAdd', 'YearBuilt'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd']
colname name which is correlated is YearRemodAdd
set {'YearRemodAdd', 'YearBuilt'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd']
colname name which is correlated is MasVnrArea
set {'YearRemodAdd', 'YearBuilt', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea']
colname name which is correlated is MasVnrArea
set {'YearRemodAdd', 'YearBuilt', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea']
colname name which is correlated is BsmtUnfSF
set {'YearRemodAdd', 'YearBuilt', 'BsmtUnfSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF']
colname name which is correlated is BsmtUnfSF
set {'YearRemodAdd', 'YearBuilt', 'BsmtUnfSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF']
colname name which is correlated is TotalBsmtSF
set {'YearBuilt', 'YearRemodAdd', 'BsmtUnfSF', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF']
colname name which is correlated is TotalBsmtSF
set {'YearBuilt', 'YearRemodAdd', 'BsmtUnfSF', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
```

'TotalBsmtSF']
colname name which is correlated is TotalBsmtSF
set {'YearBuilt', 'YearRemodAdd', 'BsmtUnfSF', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF']
colname name which is correlated is TotalBsmtSF
set {'YearBuilt', 'YearRemodAdd', 'BsmtUnfSF', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF']
colname name which is correlated is TotalBsmtSF
set {'YearBuilt', 'YearRemodAdd', 'BsmtUnfSF', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF']
colname name which is correlated is 1stFlrSF
set {'YearBuilt', '1stFlrSF', 'YearRemodAdd', 'BsmtUnfSF',
'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF']
colname name which is correlated is 1stFlrSF
set {'YearBuilt', '1stFlrSF', 'YearRemodAdd', 'BsmtUnfSF',
'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF']
colname name which is correlated is 1stFlrSF
set {'YearBuilt', '1stFlrSF', 'YearRemodAdd', 'BsmtUnfSF',
'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF']
colname name which is correlated is 1stFlrSF
set {'YearBuilt', '1stFlrSF', 'YearRemodAdd', 'BsmtUnfSF',
'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF']
colname name which is correlated is 2ndFlrSF
set {'YearBuilt', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd', 'BsmtUnfSF',
'TotalBsmtSF', 'MasVnrArea'}

```
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF']
colname name which is correlated is GrLivArea
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea']
colname name which is correlated is GrLivArea
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea']
colname name which is correlated is GrLivArea
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea']
colname name which is correlated is GrLivArea
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea']
colname name which is correlated is GrLivArea
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea']
colname name which is correlated is BsmtFullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
```

```
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath']
colname name which is correlated is BsmtFullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath']
colname name which is correlated is BsmtFullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath']
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath']
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath']
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath']
```

```
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath']
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath']
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath']
colname name which is correlated is FullBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'FullBath', 'TotalBsmtSF', 'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath']
colname name which is correlated is HalfBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
```

'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath']
colname name which is correlated is HalfBath
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath']
colname name which is correlated is BedroomAbvGr
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr']
colname name which is correlated is BedroomAbvGr
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr']
colname name which is correlated is BedroomAbvGr
set {'YearBuilt', 'GrLivArea', '1stFlrSF', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',

'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',

```
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd']
colname name which is correlated is TotRmsAbvGrd
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath',
```

'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd']
colname name which is correlated is Fireplaces
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'Fireplaces', '2ndFlrSF', 'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF',
'HalfBath', 'FullBath', 'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces']
colname name which is correlated is Fireplaces
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'Fireplaces', '2ndFlrSF', 'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF',
'HalfBath', 'FullBath', 'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces']
colname name which is correlated is Fireplaces
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'Fireplaces', '2ndFlrSF', 'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF',
'HalfBath', 'FullBath', 'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',

```
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces']
colname name which is correlated is Fireplaces
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'Fireplaces', '2ndFlrSF', 'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF',
'HalfBath', 'FullBath', 'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces']
colname name which is correlated is Fireplaces
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'Fireplaces', '2ndFlrSF', 'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF',
'HalfBath', 'FullBath', 'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces']
colname name which is correlated is GarageYrBlt
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
```

```
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt']
colname name which is correlated is GarageYrBlt
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt']
colname name which is correlated is GarageYrBlt
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt']
colname name which is correlated is GarageYrBlt
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
```

```
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt']
colname name which is correlated is GarageYrBlt
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt']
colname name which is correlated is GarageYrBlt
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
```

```
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
```

```
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
```

'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars']

```
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars']
colname name which is correlated is GarageCars
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'HalfBath', 'FullBath', 'TotalBsmtSF',
'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
```

'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',

```
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
```

```
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
```

```
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
```

'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea']
colname name which is correlated is GarageArea
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'Fireplaces', '2ndFlrSF', 'YearRemodAdd',
'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath', 'FullBath',
'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',

'GarageArea', 'GarageArea']
colname name which is correlated is OpenPorchSF
set {'YearBuilt', 'GrLivArea', 'TotRmsAbvGrd', '1stFlrSF',
'GarageYrBlt', 'GarageCars', 'OpenPorchSF', 'Fireplaces', '2ndFlrSF',
'YearRemodAdd', 'BsmtFullBath', 'BsmtUnfSF', 'GarageArea', 'HalfBath',
'FullBath', 'TotalBsmtSF', 'MasVnrArea', 'BedroomAbvGr'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'OpenPorchSF']
colname name which is correlated is EnclosedPorch
set {'YearBuilt', 'TotRmsAbvGrd', 'GarageCars', '2ndFlrSF',
'TotalBsmtSF', 'BedroomAbvGr', 'GrLivArea', 'GarageYrBlt',
'Fireplaces', 'BsmtFullBath', 'EnclosedPorch', '1stFlrSF',
'OpenPorchSF', 'YearRemodAdd', 'HalfBath', 'MasVnrArea', 'BsmtUnfSF',
'GarageArea', 'FullBath'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'OpenPorchSF', 'EnclosedPorch']
colname name which is correlated is EnclosedPorch
set {'YearBuilt', 'TotRmsAbvGrd', 'GarageCars', '2ndFlrSF',

```
'TotalBsmtSF', 'BedroomAbvGr', 'GrLivArea', 'GarageYrBlt',
'Fireplaces', 'BsmtFullBath', 'EnclosedPorch', '1stFlrSF',
'OpenPorchSF', 'YearRemodAdd', 'HalfBath', 'MasVnrArea', 'BsmtUnfSF',
'GarageArea', 'FullBath'}
lst ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'OpenPorchSF', 'EnclosedPorch',
'EnclosedPorch']
list is ['YearBuilt', 'YearBuilt', 'YearRemodAdd', 'YearRemodAdd',
'MasVnrArea', 'MasVnrArea', 'BsmtUnfSF', 'BsmtUnfSF', 'TotalBsmtSF',
'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF', 'TotalBsmtSF',
'1stFlrSF', '1stFlrSF', '1stFlrSF', '1stFlrSF', '2ndFlrSF',
'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea', 'GrLivArea',
'BsmtFullBath', 'BsmtFullBath', 'BsmtFullBath', 'FullBath',
'FullBath', 'FullBath', 'FullBath', 'FullBath', 'FullBath',
'FullBath', 'HalfBath', 'HalfBath', 'BedroomAbvGr', 'BedroomAbvGr',
'BedroomAbvGr', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd', 'TotRmsAbvGrd',
'TotRmsAbvGrd', 'Fireplaces', 'Fireplaces', 'Fireplaces',
'Fireplaces', 'Fireplaces', 'GarageYrBlt', 'GarageYrBlt',
'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt', 'GarageYrBlt',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars', 'GarageCars',
'GarageCars', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea', 'GarageArea',
'GarageArea', 'GarageArea', 'OpenPorchSF', 'EnclosedPorch',
'EnclosedPorch']

19

corr_features

{'1stFlrSF',
 '2ndFlrSF',
 'BedroomAbvGr',
 'BsmtFullBath',
```

```
 'BsmtUnfSF',
 'EnclosedPorch',
 'Fireplaces',
 'FullBath',
 'GarageArea',
 'GarageCars',
 'GarageYrBlt',
 'GrLivArea',
 'HalfBath',
 'MasVnrArea',
 'OpenPorchSF',
 'TotRmsAbvGrd',
 'TotalBsmtSF',
 'YearBuilt',
 'YearRemodAdd'}
```

```
X_train.drop(corr_features,axis=1,inplace = True)
X_test.drop(corr_features,axis=1,inplace = True)
```

```
X_train
```

```
        Id  MSSubClass MSZoning  LotArea Street Alley LotShape
LandContour  \
175    176          20       RL    12615   Pave   NaN      Reg
Lvl
1408  1409          70       RM     7740   Pave   NaN      Reg
Lvl
1148  1149          50       RM     5700   Pave   NaN      Reg
Lvl
861    862         190       RL    11625   Pave   NaN      Reg
Lvl
220    221          20       RL     8990   Pave   NaN      IR1
Lvl
...    ...         ...      ...      ...    ...   ...      ...
...
229    230         120       RL     3182   Pave   NaN      Reg
Lvl
70      71          20       RL    13651   Pave   NaN      IR1
Lvl
132    133          20       RL     7388   Pave   NaN      Reg
Lvl
1313  1314          60       RL    14774   Pave   NaN      IR1
Lvl
109    110          20       RL    11751   Pave   NaN      IR1
Lvl

     Utilities LotConfig  ... ScreenPorch PoolArea PoolQC  Fence
MiscFeature  \
175     AllPub    Corner  ...           0        0    NaN  MnPrv
NaN
1408    AllPub    Inside  ...         168        0    NaN    NaN
```

```
NaN
1148    AllPub    Inside  ...           0        0    NaN    NaN
NaN
861     AllPub    Inside  ...           0        0    NaN    NaN
NaN
220     AllPub    Inside  ...           0        0    NaN    NaN
NaN
...        ...       ...  ...         ...      ...    ...    ...
...
229     AllPub    Inside  ...           0        0    NaN    NaN
NaN
70      AllPub    Inside  ...           0        0    NaN    NaN
NaN
132     AllPub    Corner  ...           0        0    NaN    NaN
NaN
1313    AllPub    Corner  ...           0        0    NaN    NaN
NaN
109     AllPub    Inside  ...           0        0    NaN  MnPrv
NaN

     MiscVal  MoSold  YrSold SaleType SaleCondition
175        0       6    2007       WD        Normal
1408       0       6    2010       WD        Normal
1148       0       8    2008       WD        Normal
861        0       4    2010       WD        Normal
220        0       4    2006      New       Partial
...      ...     ...     ...      ...           ...
229        0       5    2009       WD        Normal
70         0       2    2007       WD        Normal
132        0       7    2007       WD        Normal
1313       0       5    2010       WD        Normal
109        0       1    2010      COD        Normal

[1022 rows x 60 columns]

X_test

        Id  MSSubClass MSZoning  LotArea Street Alley LotShape
LandContour  \
930    931          20       RL     8925   Pave   NaN      IR1
HLS
530    531          80       RL    10200   Pave   NaN      Reg
Lvl
1291  1292         160       RM     1680   Pave   NaN      Reg
Lvl
1385  1386          50       RM     5436   Pave   NaN      Reg
Lvl
305    306          20       RL    10386   Pave   NaN      Reg
Lvl
...    ...         ...      ...      ...    ...   ...      ...
```

```
...
1307  1308          20       RL     8072   Pave   NaN      Reg
Lvl
1078  1079         120       RM     4435   Pave   NaN      Reg
Lvl
1244  1245          70       RL    11435   Pave   NaN      IR1
HLS
406    407          50       RL    10480   Pave   NaN      Reg
Lvl
1459  1460          20       RL     9937   Pave   NaN      Reg
Lvl

     Utilities LotConfig  ... ScreenPorch PoolArea PoolQC  Fence
MiscFeature  \
930     AllPub    Inside  ...           0        0    NaN    NaN
NaN
530     AllPub    Inside  ...           0        0    NaN    NaN
NaN
1291    AllPub    Inside  ...           0        0    NaN    NaN
NaN
1385    AllPub    Inside  ...           0        0    NaN  MnPrv
NaN
305     AllPub    Inside  ...           0        0    NaN    NaN
NaN
...        ...       ...  ...         ...      ...    ...    ...
...
1307    AllPub    Inside  ...           0        0    NaN    NaN
NaN
1078    AllPub    Inside  ...           0        0    NaN    NaN
NaN
1244    AllPub    Corner  ...           0        0    NaN    NaN
NaN
406     AllPub    Inside  ...           0        0    NaN    NaN
NaN
1459    AllPub    Inside  ...           0        0    NaN    NaN
NaN

     MiscVal  MoSold  YrSold SaleType SaleCondition
930        0       7    2009       WD        Normal
530        0       8    2008       WD       Abnorml
1291       0       2    2009       WD        Normal
1385       0       5    2010       WD        Normal
305        0       7    2007       WD        Normal
...      ...     ...     ...      ...           ...
1307       0       5    2009       WD        Normal
1078       0       5    2006       WD        Normal
1244       0       6    2006       WD        Normal
406        0       3    2008       WD        Normal
1459       0       6    2008       WD        Normal
```

```
[438 rows x 60 columns]
```

```
df =
df.drop(["Alley","1stFlrSF","2ndFlrSF","BedroomAbvGr","BsmtFullBath","
BsmtUnfSF","EnclosedPorch","Fireplaces","FullBath","GarageArea","Garag
eCars","GarageYrBlt","GrLivArea","HalfBath","MasVnrArea","OpenPorchSF"
,"TotRmsAbvGrd","TotalBsmtSF","YearBuilt","YearRemodAdd","MSZoning","L
otConfig","Neighborhood","PoolQC","Fence","MiscFeature","YrSold","Sale
Type","SaleCondition","Condition1","Condition2","BldgType","HouseStyle
","RoofStyle","RoofMatl","Exterior1st","Exterior2nd","MasVnrType","Ext
erQual","ExterCond","Foundation","BsmtQual","BsmtCond","BsmtExposure",
"BsmtFinType1","BsmtFinType2","Heating","HeatingQC","CentralAir","Elec
trical","KitchenQual","Functional","FireplaceQu","GarageType","GarageF
inish","GarageQual","GarageCond","PavedDrive","LandSlope","Street","Lo
tShape","LandContour","Utilities"],axis = 1)
df
```

```
        Id  MSSubClass  LotArea  OverallQual  OverallCond  BsmtFinSF1
\
0        1          60     8450            7            5         706

1        2          20     9600            6            8         978

2        3          60    11250            7            5         486

3        4          70     9550            7            5         216

4        5          60    14260            8            5         655

...    ...         ...      ...          ...          ...         ...

1455  1456          60     7917            6            5           0

1456  1457          20    13175            6            6         790

1457  1458          70     9042            7            9         275

1458  1459          20     9717            5            6          49

1459  1460          20     9937            5            6         830


      BsmtFinSF2  LowQualFinSF  BsmtHalfBath  KitchenAbvGr  WoodDeckSF
\
0              0             0             0             1           0

1              0             0             1             1         298
```

```
2              0             0             0             1           0

3              0             0             0             1           0

4              0             0             0             1         192

...          ...           ...           ...           ...         ...

1455           0             0             0             1           0

1456         163             0             0             1         349

1457           0             0             0             1           0

1458        1029             0             0             1         366

1459         290             0             0             1         736


      3SsnPorch  ScreenPorch  PoolArea  MiscVal  MoSold  SalePrice
0             0            0         0        0       2     208500
1             0            0         0        0       5     181500
2             0            0         0        0       9     223500
3             0            0         0        0       2     140000
4             0            0         0        0      12     250000
...         ...          ...       ...      ...     ...        ...
1455          0            0         0        0       8     175000
1456          0            0         0        0       2     210000
1457          0            0         0     2500       5     266500
1458          0            0         0        0       4     142125
1459          0            0         0        0       6     147500

[1460 rows x 17 columns]

df.isnull().sum()

Id              0
MSSubClass      0
LotArea         0
OverallQual     0
OverallCond     0
BsmtFinSF1      0
BsmtFinSF2      0
LowQualFinSF    0
BsmtHalfBath    0
KitchenAbvGr    0
WoodDeckSF      0
3SsnPorch       0
ScreenPorch     0
```

```
PoolArea        0
MiscVal         0
MoSold          0
SalePrice       0
dtype: int64
```

```
X = df[['MSSubClass',
'LotArea',"OverallQual","OverallCond","BsmtFinSF1","BsmtFinSF2","LowQu
alFinSF","BsmtHalfBath","KitchenAbvGr","WoodDeckSF","3SsnPorch","Scree
nPorch","PoolArea","MiscVal","MoSold"]].values #independent variable
X[0:5]
```

```
array([[   60,  8450,     7,     5,   706,     0,     0,     0,     1,
             0,     0,     0,     0,     0,     2],
       [   20,  9600,     6,     8,   978,     0,     0,     1,     1,
           298,     0,     0,     0,     0,     5],
       [   60, 11250,     7,     5,   486,     0,     0,     0,     1,
             0,     0,     0,     0,     0,     9],
       [   70,  9550,     7,     5,   216,     0,     0,     0,     1,
             0,     0,     0,     0,     0,     2],
       [   60, 14260,     8,     5,   655,     0,     0,     0,     1,
           192,     0,     0,     0,     0,    12]], dtype=int64)
```

```
Y = df[["SalePrice"]]
```

```
Y
```

```
      SalePrice
0        208500
1        181500
2        223500
3        140000
4        250000
...         ...
1455     175000
1456     210000
1457     266500
1458     142125
1459     147500

[1460 rows x 1 columns]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 500)
```

```
X_train.shape
```

```
(1168, 15)
```

```
X_test.shape
```

```
(292, 15)
```

Feature Scaling

```
# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_Y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
Y_train = sc_Y.fit_transform(Y_train)
Y_test = sc_Y.transform(Y_test)

X_train

array([[-0.85469057, -0.0242262 , -0.78256106, ..., -0.07684505,
        -0.11595495,  0.24559339],
       [-0.85469057, -0.18471992,  0.63894954, ..., -0.07684505,
        -0.11595495, -0.49181812],
       [ 0.09911675,  0.29260181,  2.77121545, ..., -0.07684505,
        -0.11595495, -0.12311237],
       ...,
       [-0.85469057, -0.10082087, -0.07180576, ..., -0.07684505,
        -0.11595495, -0.86052387],
       [-0.85469057, -0.28343067, -0.78256106, ..., -0.07684505,
        -0.11595495,  0.24559339],
       [-0.85469057, -0.01966095,  0.63894954, ..., -0.07684505,
        -0.11595495, -0.12311237]])

X_test

array([[ 0.09911675, -0.17731408, -0.07180576, ..., -0.07684505,
        -0.11595495,  0.61429914],
       [ 0.09911675, -0.01154496,  0.63894954, ..., -0.07684505,
        -0.11595495,  0.98300489],
       [ 0.33756858, -0.11299485, -0.78256106, ..., -0.07684505,
        -0.11595495, -0.12311237],
       ...,
       [ 0.09911675, -0.2562421 , -0.07180576, ..., -0.07684505,
        -0.11595495,  2.08912215],
       [ 1.52982773, -0.26192329, -0.07180576, ..., -0.07684505,
        -0.11595495,  0.24559339],
       [ 0.09911675, -0.14353127,  0.63894954, ..., -0.07684505,
        -0.11595495, -0.12311237]])

Y_train

array([[-0.47654967],
       [ 0.04412298],
       [ 3.30843421],
```

```
       ...,
       [-0.52555415],
       [-0.75955057],
       [-0.36628958]])
```

```
Y_test
```

```
array([[-9.79900262e-02],
       [ 2.83019842e-01],
       [-4.33670746e-01],
       [-5.56181958e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 6.07674553e-01],
       [-7.52199896e-01],
       [ 4.91288902e-01],
       [-2.49903929e-01],
       [-2.75214745e-01],
       [ 5.15791144e-01],
       [-5.90485097e-01],
       [-5.13303034e-01],
       [-7.22626718e-02],
       [ 4.07981278e-01],
       [ 1.23755267e-01],
       [-6.66442048e-01],
       [-3.81603481e-01],
       [-9.54343396e-01],
       [-3.41787337e-01],
       [ 4.90234277e-02],
       [ 5.28042265e-01],
       [ 5.26817153e-01],
       [ 2.52392039e-01],
       [-7.38723663e-01],
       [-2.74406171e-01],
       [-1.49339273e+00],
       [-5.99060882e-01],
       [-4.15294064e-01],
       [ 1.48257509e-01],
       [-3.67344204e-02],
       [ 2.70874183e+00],
       [ 8.94350788e-01],
       [-5.13303034e-01],
       [-1.00334788e+00],
       [-5.38859900e-02],
       [ 4.60661099e-01],
       [-4.82675231e-01],
```

```
[-9.42092275e-01],
[-2.65830386e-01],
[-2.92782853e-01],
[ 2.54413474e-01],
[-3.47912898e-01],
[-4.77604295e-02],
[-1.10070660e-02],
[ 5.28042265e-01],
[-7.83882324e-02],
[-1.10070660e-02],
[ 4.36264319e+00],
[ 3.13647645e-01],
[-7.95078821e-01],
[ 1.10384496e+00],
[-9.42092275e-01],
[-2.74406171e-01],
[-5.56181958e-01],
[ 1.89430863e-05],
[ 1.48257509e-01],
[-1.09016035e-01],
[-4.39796307e-01],
[-3.49909831e-01],
[-9.42092275e-01],
[-6.11312003e-01],
[ 2.57462975e-02],
[-5.01051913e-01],
[ 5.63741004e-02],
[-6.35814245e-01],
[ 5.09665584e-01],
[ 1.80093375e+00],
[-7.59550569e-01],
[-4.27545186e-01],
[-4.52047428e-01],
[-4.58172988e-01],
[-1.15036133e+00],
[-3.43012450e-01],
[-4.53272540e-01],
[-4.03042943e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 1.69189878e+00],
[-5.74558640e-01],
[-1.33412815e+00],
[ 1.20185393e+00],
[-1.70271641e-01],
[-2.68280610e-01],
[-1.21774250e+00],
[ 8.34320294e-01],
[ 5.63741004e-02],
[-1.09016035e-01],
```

```
[-4.64298549e-01],
[-6.48065367e-01],
[-9.05338911e-01],
[-4.64298549e-01],
[-1.04010124e+00],
[ 3.44275448e-01],
[-4.76549670e-01],
[-3.84666262e-01],
[ 6.75055719e-01],
[-1.64146081e-01],
[-4.88150538e-03],
[-6.35814245e-01],
[-2.49903929e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.45137930e-02],
[ 7.10583971e-01],
[-2.43778368e-01],
[-7.14221421e-01],
[-8.57389051e-02],
[-2.68280610e-01],
[ 4.60661099e-01],
[-1.13233994e+00],
[ 4.17782175e-01],
[ 1.53263420e+00],
[-7.64451018e-01],
[ 2.39633824e+00],
[ 1.16510057e+00],
[ 4.66786659e-01],
[ 6.86252216e-02],
[ 7.11809083e-01],
[-2.32581871e-02],
[-7.33823215e-01],
[-5.14528146e-01],
[-1.22386806e+00],
[-3.41787337e-01],
[-4.76549670e-01],
[-5.13915590e-01],
[ 8.09818052e-01],
[ 1.69189878e+00],
[ 9.92530245e-02],
[-2.43778368e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.25657619e-01],
[-7.95078821e-01],
[ 4.91288902e-01],
[ 2.79449968e+00],
[ 1.62835221e+00],
[-6.60316488e-01],
```

```
[-2.68280610e-01],
[ 1.96036882e-01],
[-1.33518278e-01],
[-2.19276126e-01],
[-5.13303034e-01],
[-4.70424110e-01],
[-9.35966714e-01],
[-8.45137930e-02],
[-2.68280610e-01],
[-7.70576578e-01],
[-5.31679715e-01],
[-1.49339273e+00],
[-1.08298017e+00],
[ 9.92530245e-02],
[-4.52047428e-01],
[ 1.03033823e+00],
[-7.27697654e-01],
[ 2.50353555e+00],
[ 1.32436514e+00],
[-1.00334788e+00],
[ 1.63064317e+00],
[-1.00334788e+00],
[ 1.75315438e+00],
[-5.75783752e-01],
[ 6.38302356e-01],
[-3.29536216e-01],
[-1.71326265e-02],
[-5.68433079e-01],
[ 9.06601909e-01],
[-5.74558640e-01],
[ 3.30799215e-01],
[-6.05186442e-01],
[-1.08910573e+00],
[-3.05033974e-01],
[ 6.50553477e-01],
[ 3.13647645e-01],
[-7.70576578e-01],
[-6.37039358e-01],
[ 2.34015357e-01],
[-1.45891910e-01],
[-6.66442048e-01],
[ 1.60508630e-01],
[-1.71326265e-02],
[-8.93590086e-01],
[ 4.23907735e-01],
[-5.77008864e-01],
[-6.35814245e-01],
[-5.13303034e-01],
[-1.45769399e-01],
[-3.05033974e-01],
```

```
[-9.78845638e-01],
[-2.69566978e-01],
[-6.11312003e-01],
[-8.45137930e-02],
[-2.31527247e-01],
[ 2.57462975e-02],
[-1.09016035e-01],
[-7.09320972e-01],
[-5.01051913e-01],
[ 2.58517600e-01],
[-4.64298549e-01],
[-1.34025371e+00],
[ 1.65024496e+00],
[ 9.44580385e-01],
[-2.68893166e-01],
[ 6.26051235e-01],
[-9.78845638e-01],
[ 5.89297871e-01],
[ 9.07827021e-01],
[-6.29688685e-01],
[-1.18711470e+00],
[ 8.70019034e-02],
[-9.64144293e-01],
[ 1.01270887e+00],
[-1.23611918e+00],
[ 2.57292488e-01],
[ 1.42131949e-01],
[-8.56334426e-01],
[-4.52047428e-01],
[ 8.33265670e-02],
[-2.93837477e-02],
[ 4.97414462e-01],
[ 1.07321716e+00],
[ 6.99557962e-01],
[-1.24959542e+00],
[-1.33518278e-01],
[-3.78540701e-01],
[-6.48065367e-01],
[ 1.02911312e+00],
[-5.50056397e-01],
[ 7.07962231e-01],
[-9.11464472e-01],
[-2.32581871e-02],
[-8.01204381e-01],
[ 3.48668803e+00],
[-5.25554155e-01],
[-8.80836669e-01],
[-2.32581871e-02],
[ 1.85010873e-01],
[ 1.25085841e+00],
```

```
[ 3.79974187e-02],
[-4.29995410e-01],
[ 4.05531054e-01],
[-4.76549670e-01],
[-1.17486358e+00],
[-2.25401686e-01],
[ 1.12834720e+00],
[-1.71496753e-01],
[ 2.17091761e+00],
[ 5.89297871e-01],
[ 6.87306840e-01],
[ 1.15284944e+00],
[-3.78540701e-01],
[-2.92782853e-01],
[ 3.91960590e+00],
[-3.72415140e-01],
[-8.19581063e-01],
[ 3.44275448e-01],
[ 3.67723066e-02],
[ 2.58517600e-01],
[-4.22644737e-01],
[-5.25554155e-01],
[-5.13303034e-01],
[-6.11312003e-01],
[ 1.23755267e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 2.42696605e+00],
[ 6.21150786e-01],
[-3.94467158e-01],
[-1.15036133e+00],
[-1.55464833e+00],
[-7.39948775e-01],
[-7.76702139e-01],
[ 1.26310954e+00],
[-1.45769399e-01],
[-2.07025005e-01],
[-4.21419625e-01],
[-5.13303034e-01],
[-4.15294064e-01],
[-3.29536216e-01],
[-6.05186442e-01],
[-7.76702139e-01],
[-7.70576578e-01],
[-9.67649141e-02],
[ 6.24996610e-02],
[ 1.13202254e+00],
[-4.76549670e-01],
[-3.99367607e-01],
[-1.00947344e+00],
```

```
       [ 2.21764236e-01],
       [ 1.36006388e-01],
       [ 1.36006388e-01]])
```

Multiple Regression Model

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)#training func question + answers
# The coefficients
print ('Intercept: ',regr.intercept_)
print ('Coefficient : ',regr.coef_)
```

```
Intercept:  [-9.63766304e-17]
Coefficient :  [[-9.02873132e-02  9.87258038e-02  7.27431899e-01 -
2.45253111e-03
   1.50082091e-01  9.31031720e-03  7.51810841e-03 -3.46707655e-03
   4.88886019e-02  1.01349908e-01  3.38783711e-02  5.96773472e-02
   5.10781326e-03  5.93216756e-04 -1.01065850e-02]]
```

```
regr.intercept_
```

```
array([-9.63766304e-17])
```

```
regr.coef_
```

```
array([[-9.02873132e-02,  9.87258038e-02,  7.27431899e-01,
        -2.45253111e-03,  1.50082091e-01,  9.31031720e-03,
         7.51810841e-03, -3.46707655e-03,  4.88886019e-02,
         1.01349908e-01,  3.38783711e-02,  5.96773472e-02,
         5.10781326e-03,  5.93216756e-04, -1.01065850e-02]])
```

```
regr.coef_[0][0]
```

```
-0.09028731320937944
```

```
regr.coef_[0][1]
```

```
0.0987258037893155
```

```
y_pred = regr.predict(X_test)
```

```
y_pred
```

```
array([[-0.33722427],
       [ 0.31952841],
       [-0.60598383],
       [-0.43255507],
       [-0.2066125 ],
       [-0.07160902],
       [ 0.6781571 ],
       [-1.10503171],
```

```
[ 0.21028814],
[-1.23931036],
[-0.814138  ],
[ 0.4244962 ],
[-0.5451187 ],
[-0.21897612],
[-0.46689745],
[ 0.33025668],
[ 0.44355144],
[-0.08914773],
[-0.65128136],
[-0.34497513],
[ 0.07679373],
[ 0.32799796],
[ 0.46826844],
[ 0.31005931],
[ 0.13525266],
[-1.2042709 ],
[ 0.17498732],
[-0.74903129],
[-0.50982496],
[-0.4303928 ],
[-0.44485707],
[-0.31956605],
[ 0.98460265],
[ 0.29931619],
[-0.12868821],
[-0.52960458],
[ 0.02974848],
[ 0.33418911],
[-0.27523327],
[-0.72647106],
[ 0.07658592],
[ 0.10426839],
[ 0.15597823],
[-0.68344642],
[ 0.56527845],
[-0.31353658],
[ 0.43280447],
[-0.09031499],
[ 0.13700916],
[ 2.24160402],
[ 0.45744836],
[-0.02999004],
[ 0.43046149],
[-1.64726142],
[-0.05336739],
[ 0.10097173],
[ 0.17522168],
[ 0.38798755],
```

```
[ 0.03163575],
[-0.56627556],
[-0.26515208],
[-0.69104115],
[-0.03245363],
[ 0.38942499],
[-0.79396253],
[ 0.19121225],
[-0.89236777],
[-0.32129353],
[ 1.54942548],
[-0.68230439],
[-0.3041476 ],
[ 0.15680716],
[-0.17693343],
[-0.60565691],
[-0.36108329],
[-0.625252  ],
[-0.0261397 ],
[ 0.19006291],
[-0.52798787],
[ 0.99621294],
[-0.35212091],
[-0.81840694],
[ 1.61590636],
[ 0.81302414],
[-0.55413843],
[ 0.73116857],
[ 0.60807354],
[-0.31187296],
[-0.04262776],
[-0.33011247],
[-0.52552539],
[-1.30033641],
[ 0.14908033],
[-0.74441845],
[ 0.50556446],
[-0.21782159],
[-0.28690066],
[ 0.97966072],
[-0.00940664],
[ 0.3024453 ],
[-0.67620002],
[ 0.08262464],
[ 0.24298669],
[-0.34124958],
[ 0.70405685],
[ 0.72047602],
[-0.21422159],
[-1.104303  ],
```

```
[-0.24206834],
[-0.93603582],
[ 0.25833981],
[-1.26780733],
[ 0.36339021],
[ 1.44178445],
[-0.89886932],
[ 1.93990215],
[ 1.40125033],
[ 1.08180221],
[ 0.69134642],
[ 0.62626405],
[ 0.3178701 ],
[-0.49942853],
[-0.73199092],
[-1.08795445],
[-0.50991548],
[-0.5287795 ],
[-0.60028323],
[ 0.72585446],
[ 1.34688684],
[-0.10122057],
[-0.2246643 ],
[ 0.19150164],
[-0.38169556],
[-0.42313689],
[-0.85662924],
[ 0.59962032],
[ 1.28761947],
[ 1.62859904],
[-0.44478709],
[-0.68480354],
[-0.11305694],
[-0.08475332],
[-0.21099322],
[ 0.3863821 ],
[-0.57207386],
[-0.31588738],
[-0.30103027],
[ 0.13363087],
[-0.6245721 ],
[-0.24182002],
[-2.33757118],
[-1.73137674],
[ 0.61196944],
[-0.53626757],
[ 1.4997626 ],
[-0.79273903],
[ 2.09032573],
[ 1.18323343],
```

```
[-1.22047336],
[ 1.0241325 ],
[-0.67923585],
[ 1.22813411],
[-0.1785582 ],
[ 0.33440121],
[ 0.47980554],
[ 1.00268185],
[-0.36115082],
[ 0.73208475],
[-0.75159389],
[ 0.23932238],
[-0.5761583 ],
[-0.95300175],
[-0.46454803],
[ 0.89185981],
[ 0.01033378],
[-0.55264001],
[-0.6759019 ],
[ 0.87387675],
[-0.12287569],
[-0.06103652],
[ 0.29288302],
[ 0.48033152],
[-0.66651045],
[ 0.0985792 ],
[-0.51018977],
[-0.52694856],
[-0.16585878],
[ 0.46081645],
[ 0.15920724],
[-1.00175711],
[ 0.30910305],
[ 0.22095195],
[-0.00307097],
[-0.36839809],
[ 0.31487625],
[ 0.08424518],
[-0.86897852],
[-0.15478399],
[ 0.80300416],
[-0.6564087 ],
[-1.82287847],
[ 1.48885577],
[ 0.84617202],
[-0.67168549],
[ 0.42930984],
[-0.2881596 ],
[ 1.0804899 ],
[-0.28413306],
```

```
[-0.10740637],
[-0.69173568],
[-0.70807322],
[-1.35978398],
[ 1.13528322],
[-1.46590868],
[ 0.87499778],
[ 0.28135719],
[-0.16744749],
[-0.63710845],
[-0.22709232],
[-0.4958855 ],
[ 0.31462704],
[ 0.73962659],
[ 0.87080469],
[-1.32949369],
[-0.10851957],
[-0.08476628],
[-0.48278032],
[ 1.00479167],
[-0.58486329],
[ 1.02508065],
[-0.77410236],
[ 0.1738029 ],
[-0.44820476],
[ 2.52769596],
[-0.68428717],
[-0.74762264],
[ 1.06881456],
[ 0.86047993],
[ 0.7355322 ],
[ 0.49234704],
[-0.07374643],
[ 0.29435635],
[-0.04721922],
[-0.80028664],
[-0.52011149],
[ 1.1620312 ],
[-0.02263783],
[ 0.83995179],
[-0.07400944],
[ 0.32020876],
[ 0.69308048],
[-0.0726412 ],
[-0.01250445],
[ 2.00040032],
[-0.01457152],
[-1.28043383],
[ 0.35846698],
[ 0.54553228],
```

```
       [ 0.61861332],
       [-0.08273755],
       [-0.77418483],
       [-0.5952292 ],
       [-0.38737299],
       [ 0.4172817 ],
       [-0.20340189],
       [-0.75876498],
       [ 1.61753864],
       [ 0.81413456],
       [-0.06252453],
       [-1.35475174],
       [-1.3305922 ],
       [-0.21381574],
       [-0.09886488],
       [ 1.87637359],
       [ 0.50861916],
       [-0.21921074],
       [ 0.12846956],
       [-0.52443234],
       [-0.4926669 ],
       [ 0.10661218],
       [-0.29863809],
       [-0.57362189],
       [-0.20341452],
       [-0.32581383],
       [-0.28527058],
       [ 0.71253515],
       [-0.56171185],
       [-0.15478399],
       [-0.43828999],
       [-0.26238376],
       [-0.37192232],
       [ 0.39365457]])
```

```
from sklearn.metrics import r2_score

print(f"R2 Score : {r2_score(Y_test,y_pred)*100} % ")
```

```
R2 Score : 70.281267555182 %
```

```
print(f"Mean absolute error: {np.mean(np.absolute(y_pred - Y_test))}
")#pred - actual
```

```
Mean absolute error: 0.3459234130131204
```

```
print("Residual sum of squares (MSE): %.2f" % np.mean((y_pred -
Y_test) **2))
```

```
Residual sum of squares (MSE): 0.22
```

```
accuracy = (f"accuracy : {r2_score(Y_test,y_pred)*100} % ")
```

```
accuracy
```

```
'accuracy : 70.281267555182 % '
```

Apply Lasso Regression to cover the overfitting and underfitting problem

```
from sklearn.linear_model import Lasso

L = Lasso(alpha = 1)

L.fit(X_train,Y_train)
```

```
Lasso(alpha=1)
```

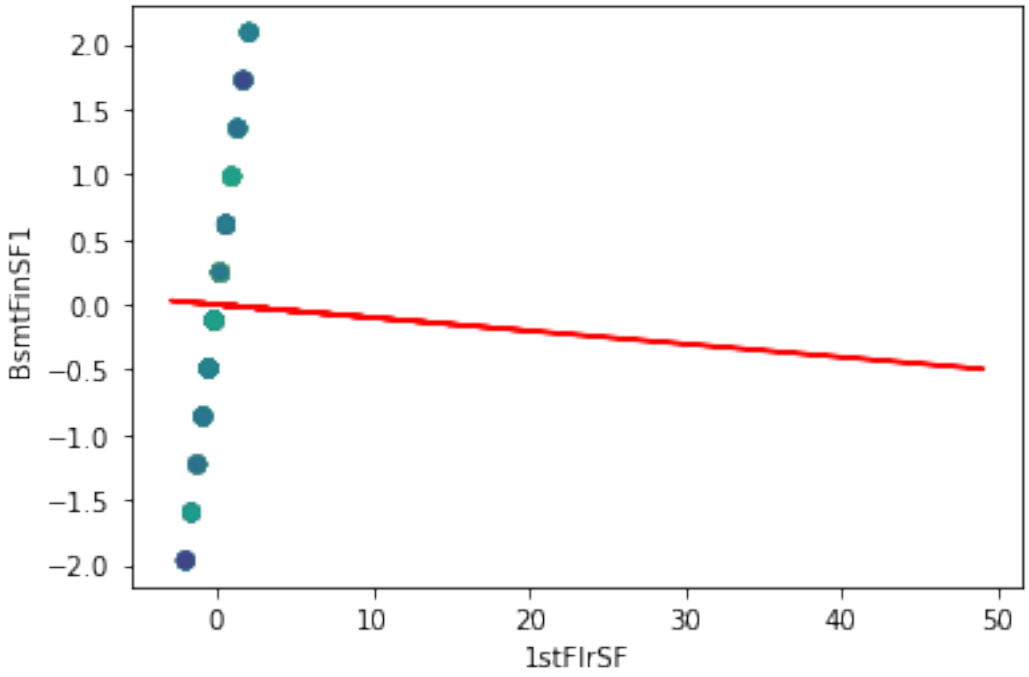
```
ypred1 = L.predict(X_test)

from sklearn.metrics import r2_score
print("R2-score",r2_score(Y_test,ypred1))
print(f"Mean absolute error: {np.mean(np.absolute(ypred1 - Y_test))}
")# pred - actual
```

```
R2-score -0.004908518792945626
Mean absolute error: 0.6243935831081134
```

Plots

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,14],X_test[:,-1], c = y_pred)
plt.plot(X_test, regr.coef_[0][14]*X_test + regr.intercept_[0], 'r') #
y = mx+c
plt.xlabel("1stFlrSF")
plt.ylabel("BsmtFinSF1")
```

```
Text(0, 0.5, 'BsmtFinSF1')
```




```
# This Independent variable shows the Non-Linear Relationship
lst = ["MSSubClass',
'LotArea','OverallQual','OverallCond','BsmtFinSF1','BsmtFinSF2','LowQu
alFinSF','BsmtHalfBath','KitchenAbvGr","WoodDeckSF","3SsnPorch","Scree
nPorch","PoolArea","MiscVal","MoSold"]

lst
```

```
["MSSubClass',
'LotArea','OverallQual','OverallCond','BsmtFinSF1','BsmtFinSF2','LowQu
alFinSF','BsmtHalfBath','KitchenAbvGr",
 'WoodDeckSF',
 '3SsnPorch',
 'ScreenPorch',
 'PoolArea',
 'MiscVal',
 'MoSold']
```

Ans : All the Independent Variable shows the Non-Linear Relationship

Apply Variance method

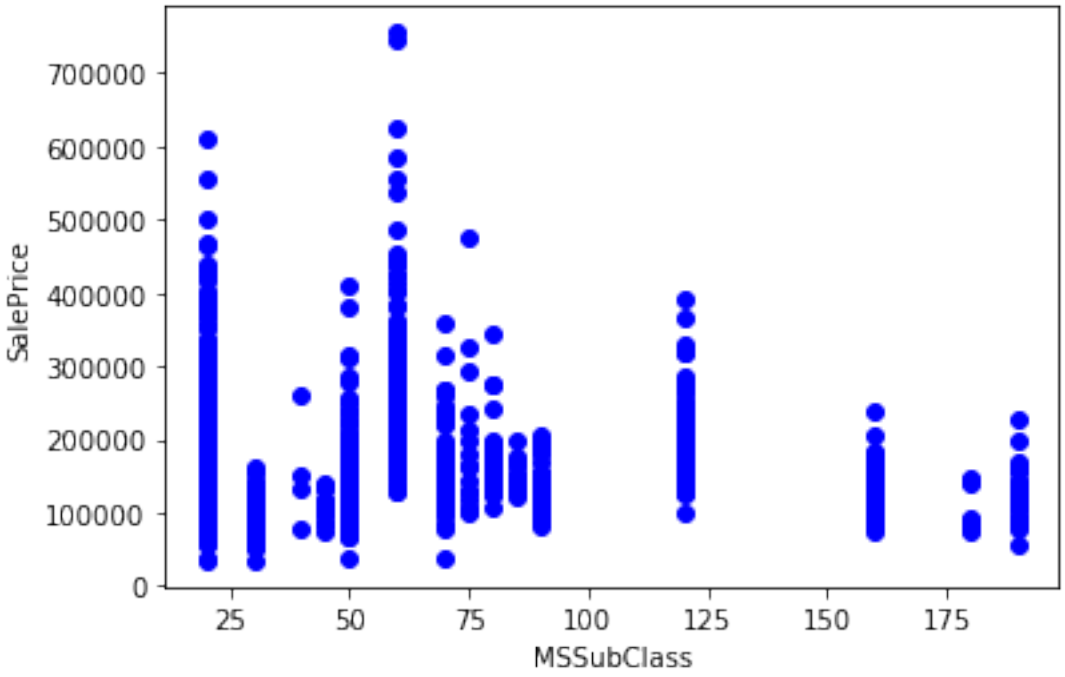
```
import pandas as pd
from sklearn.feature_selection import VarianceThreshold

df
```

```
        Id  MSSubClass MSZoning  LotArea Street Alley LotShape
LandContour  \
0        1          60       RL     8450   Pave   NaN      Reg
Lvl
1        2          20       RL     9600   Pave   NaN      Reg
Lvl
2        3          60       RL    11250   Pave   NaN      IR1
Lvl
3        4          70       RL     9550   Pave   NaN      IR1
Lvl
4        5          60       RL    14260   Pave   NaN      IR1
Lvl
...    ...         ...      ...      ...    ...   ...      ...
...
1455  1456          60       RL     7917   Pave   NaN      Reg
Lvl
1456  1457          20       RL    13175   Pave   NaN      Reg
Lvl
1457  1458          70       RL     9042   Pave   NaN      Reg
Lvl
1458  1459          20       RL     9717   Pave   NaN      Reg
Lvl
1459  1460          20       RL     9937   Pave   NaN      Reg
Lvl

     Utilities LotConfig  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0       AllPub    Inside  ...        0    NaN    NaN         NaN
0
1       AllPub       FR2  ...        0    NaN    NaN         NaN
0
2       AllPub    Inside  ...        0    NaN    NaN         NaN
0
3       AllPub    Corner  ...        0    NaN    NaN         NaN
0
4       AllPub       FR2  ...        0    NaN    NaN         NaN
0
...        ...       ...  ...      ...    ...    ...         ...     .
..
1455    AllPub    Inside  ...        0    NaN    NaN         NaN
0
1456    AllPub    Inside  ...        0    NaN  MnPrv         NaN
0
1457    AllPub    Inside  ...        0    NaN  GdPrv        Shed
2500
1458    AllPub    Inside  ...        0    NaN    NaN         NaN
0
1459    AllPub    Inside  ...        0    NaN    NaN         NaN
0
```

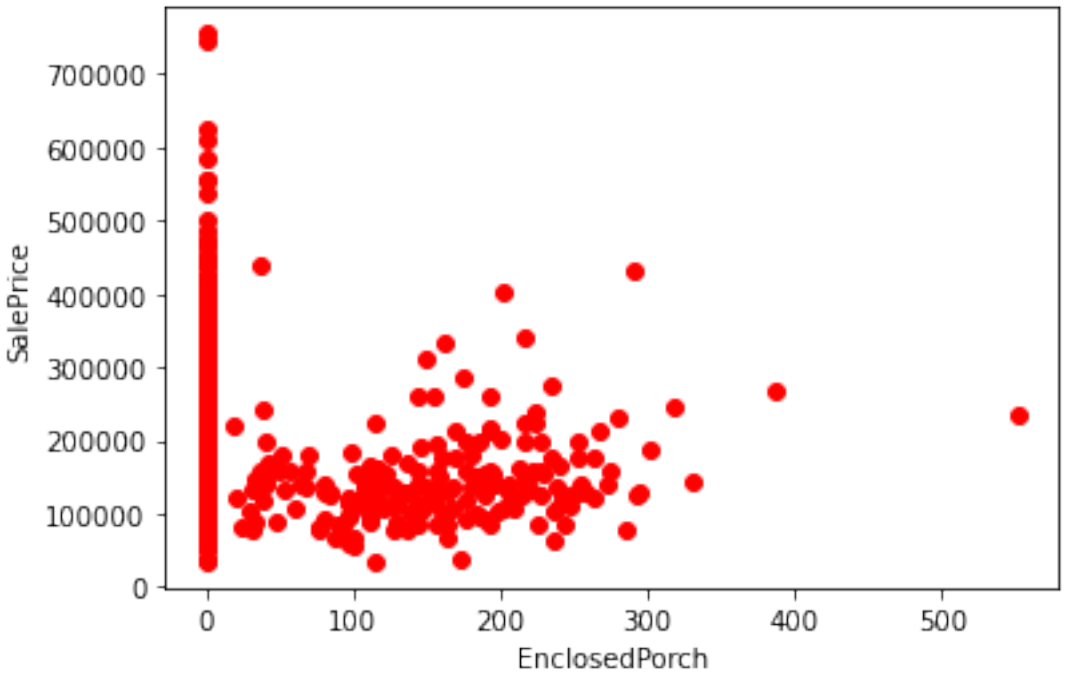
```
     MoSold  YrSold  SaleType SaleCondition SalePrice
0         2    2008        WD        Normal    208500
1         5    2007        WD        Normal    181500
2         9    2008        WD        Normal    223500
3         2    2006        WD       Abnorml    140000
4        12    2008        WD        Normal    250000
...     ...     ...       ...           ...       ...
1455      8    2007        WD        Normal    175000
1456      2    2010        WD        Normal    210000
1457      5    2010        WD        Normal    266500
1458      4    2010        WD        Normal    142125
1459      6    2008        WD        Normal    147500

[1460 rows x 61 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 61 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   LotArea        1460 non-null   int64
 4   Street         1460 non-null   object
 5   Alley          91 non-null     object
 6   LotShape       1460 non-null   object
 7   LandContour    1460 non-null   object
 8   Utilities      1460 non-null   object
 9   LotConfig      1460 non-null   object
 10  LandSlope      1460 non-null   object
 11  Neighborhood   1460 non-null   object
 12  Condition1     1460 non-null   object
 13  Condition2     1460 non-null   object
 14  BldgType       1460 non-null   object
 15  HouseStyle     1460 non-null   object
 16  OverallQual    1460 non-null   int64
 17  OverallCond    1460 non-null   int64
 18  RoofStyle      1460 non-null   object
 19  RoofMatl       1460 non-null   object
 20  Exterior1st    1460 non-null   object
 21  Exterior2nd    1460 non-null   object
 22  MasVnrType     1452 non-null   object
 23  ExterQual      1460 non-null   object
 24  ExterCond      1460 non-null   object
 25  Foundation     1460 non-null   object
 26  BsmtQual       1423 non-null   object
 27  BsmtCond       1423 non-null   object
```

```
 28  BsmtExposure   1422 non-null   object
 29  BsmtFinType1   1423 non-null   object
 30  BsmtFinSF1     1460 non-null   int64
 31  BsmtFinType2   1422 non-null   object
 32  BsmtFinSF2     1460 non-null   int64
 33  Heating        1460 non-null   object
 34  HeatingQC      1460 non-null   object
 35  CentralAir     1460 non-null   object
 36  Electrical     1459 non-null   object
 37  LowQualFinSF   1460 non-null   int64
 38  BsmtHalfBath   1460 non-null   int64
 39  KitchenAbvGr   1460 non-null   int64
 40  KitchenQual    1460 non-null   object
 41  Functional     1460 non-null   object
 42  FireplaceQu    770 non-null    object
 43  GarageType     1379 non-null   object
 44  GarageFinish   1379 non-null   object
 45  GarageQual     1379 non-null   object
 46  GarageCond     1379 non-null   object
 47  PavedDrive     1460 non-null   object
 48  WoodDeckSF     1460 non-null   int64
 49  3SsnPorch      1460 non-null   int64
 50  ScreenPorch    1460 non-null   int64
 51  PoolArea       1460 non-null   int64
 52  PoolQC         7 non-null      object
 53  Fence          281 non-null    object
 54  MiscFeature    54 non-null     object
 55  MiscVal        1460 non-null   int64
 56  MoSold         1460 non-null   int64
 57  YrSold         1460 non-null   int64
 58  SaleType       1460 non-null   object
 59  SaleCondition  1460 non-null   object
 60  SalePrice      1460 non-null   int64
dtypes: int64(18), object(43)
memory usage: 695.9+ KB
```

```
import matplotlib.pyplot as plt
plt.scatter(df.MSSubClass, df.SalePrice,  color='blue')
plt.xlabel("MSSubClass")
plt.ylabel("SalePrice")
plt.show()
```

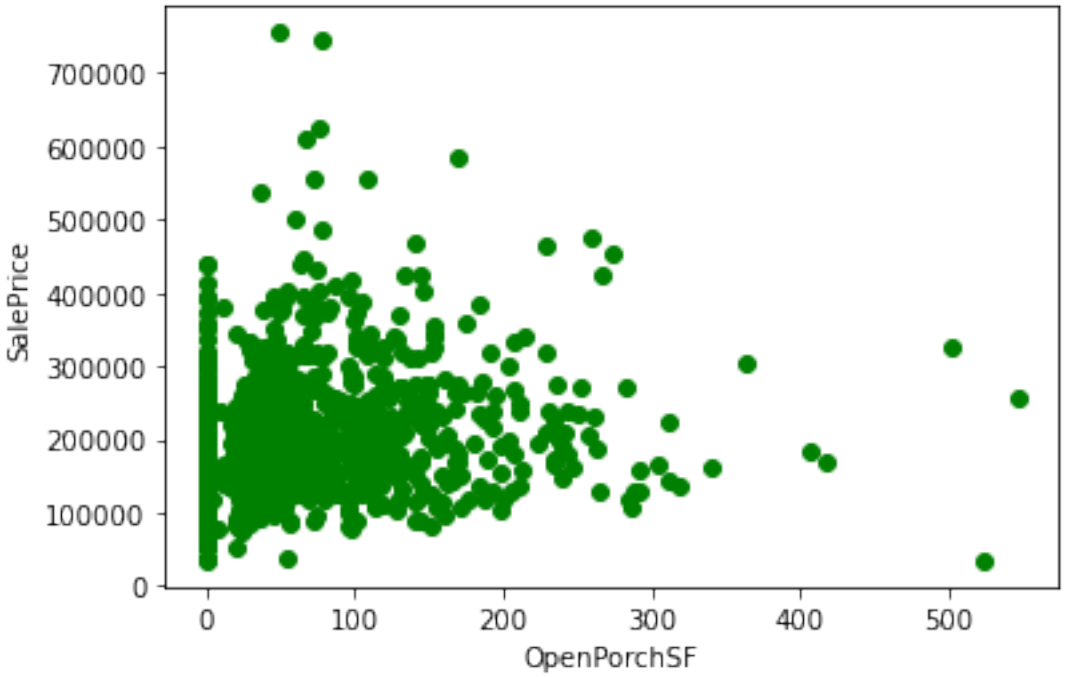



```
import matplotlib.pyplot as plt
plt.scatter(df.EnclosedPorch, df.SalePrice,  color='red')
plt.xlabel("EnclosedPorch")
plt.ylabel("SalePrice")
plt.show()
```

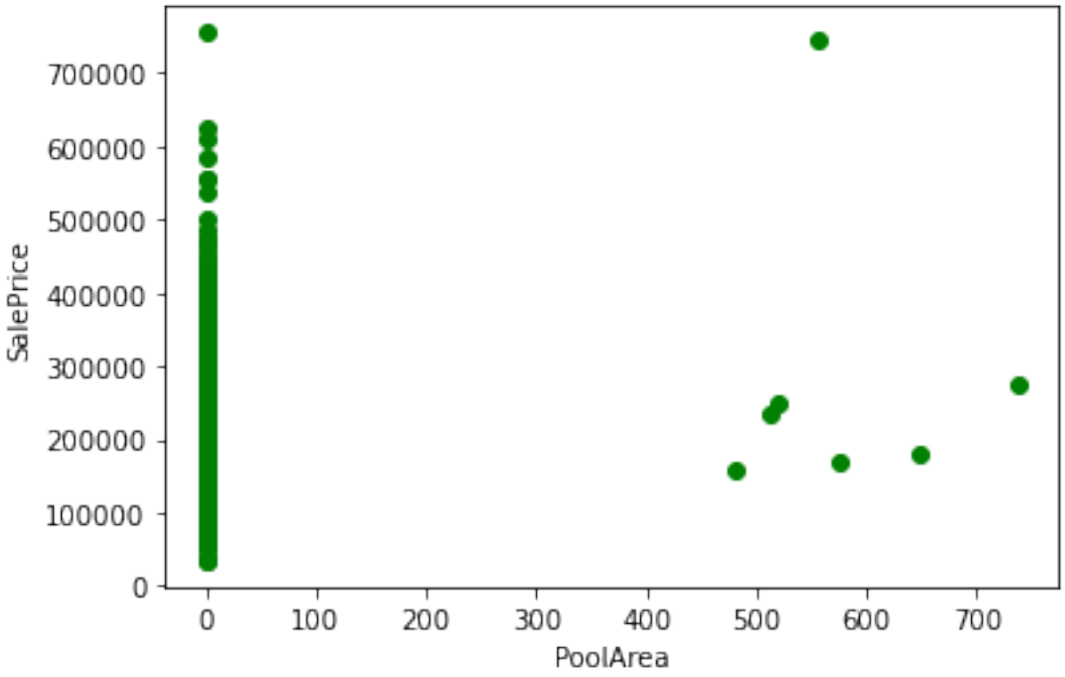



```
import matplotlib.pyplot as plt
plt.scatter(df.OpenPorchSF, df.SalePrice,  color='green')
```

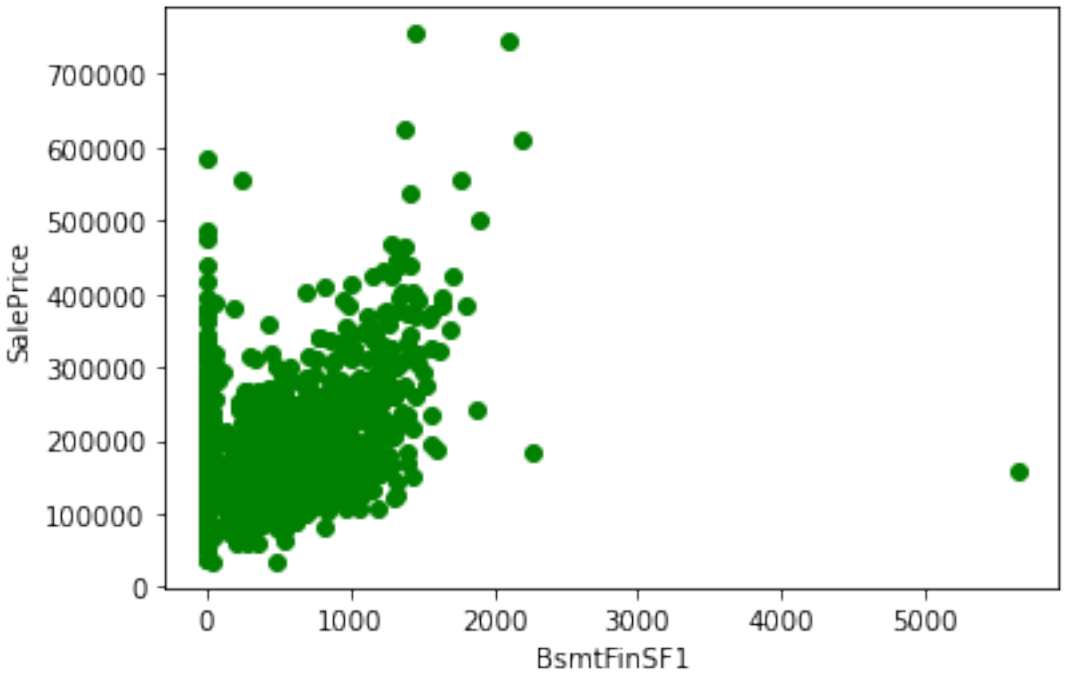
```
plt.xlabel("OpenPorchSF")
plt.ylabel("SalePrice")
plt.show()
```




```
import matplotlib.pyplot as plt
plt.scatter(df.PoolArea, df.SalePrice,  color='green')
plt.xlabel("PoolArea")
plt.ylabel("SalePrice")
plt.show()
```




```
import matplotlib.pyplot as plt
plt.scatter(df.BsmtFinSF1, df.SalePrice,  color='green')
plt.xlabel("BsmtFinSF1")
plt.ylabel("SalePrice")
plt.show()
```




```
lst =
["OverallQual","EnclosedPorch","OpenPorchSF","PoolArea","BsmtFinSF1"]
```

Que.2 Columns which showed non linear behavior apply Polynomial Linear Regression to it ...

```
# Note : If there is None column which is showing Non Linear Behavior
you can take anyone of the column as independent variable and apply
Polynomial Linear Regression to it

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/FuelConsumption.csv")

# take a look at the dataset
df.head()
```

```
   MODELYEAR   MAKE       MODEL VEHICLECLASS  ENGINESIZE  CYLINDERS  \
0       2014  ACURA         ILX      COMPACT         2.0          4
1       2014  ACURA         ILX      COMPACT         2.4          4
2       2014  ACURA  ILX HYBRID      COMPACT         1.5          4
3       2014  ACURA     MDX 4WD  SUV - SMALL         3.5          6
4       2014  ACURA     RDX AWD  SUV - SMALL         3.5          6

  TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  \
0          AS5        Z                   9.9                  6.7
1           M6        Z                  11.2                  7.7
2          AV7        Z                   6.0                  5.8
3          AS6        Z                  12.7                  9.1
4          AS6        Z                  12.1                  8.7

   FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                   8.5                        33           196
1                   9.6                        29           221
2                   5.9                        48           136
3                  11.1                        25           255
4                  10.6                        27           244
```

```
df.tail()
```

```
      MODELYEAR   MAKE     MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
1062       2014  VOLVO  XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO  XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO  XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO  XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO  XC90 AWD  SUV - STANDARD         3.2
```

```
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
1062          AS6        X                  13.4                  9.8

1063          AS6        X                  13.2                  9.5

1064          AS6        X                  13.4                  9.8

1065          AS6        X                  12.9                  9.3

1066          AS6        X                  14.9                 10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

df.shape

(1067, 13)

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1067 entries, 0 to 1066
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   MODELYEAR                 1067 non-null   int64
 1   MAKE                      1067 non-null   object
 2   MODEL                     1067 non-null   object
 3   VEHICLECLASS              1067 non-null   object
 4   ENGINESIZE                1067 non-null   float64
 5   CYLINDERS                 1067 non-null   int64
 6   TRANSMISSION              1067 non-null   object
 7   FUELTYPE                  1067 non-null   object
 8   FUELCONSUMPTION_CITY      1067 non-null   float64
 9   FUELCONSUMPTION_HWY       1067 non-null   float64
 10  FUELCONSUMPTION_COMB      1067 non-null   float64
 11  FUELCONSUMPTION_COMB_MPG  1067 non-null   int64
 12  CO2EMISSIONS              1067 non-null   int64
dtypes: float64(4), int64(4), object(5)
memory usage: 108.5+ KB

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

```
MODELYEAR                   0
MAKE                        0
MODEL                       0
VEHICLECLASS                0
ENGINESIZE                  0
CYLINDERS                   0
TRANSMISSION                0
FUELTYPE                    0
FUELCONSUMPTION_CITY        0
FUELCONSUMPTION_HWY         0
FUELCONSUMPTION_COMB        0
FUELCONSUMPTION_COMB_MPG    0
CO2EMISSIONS                0
dtype: int64
```

```
df =
df.drop(["MODELYEAR","MAKE","MODEL","VEHICLECLASS","TRANSMISSION","FUE
LTYPE"],axis = 1)
```

```
df
```

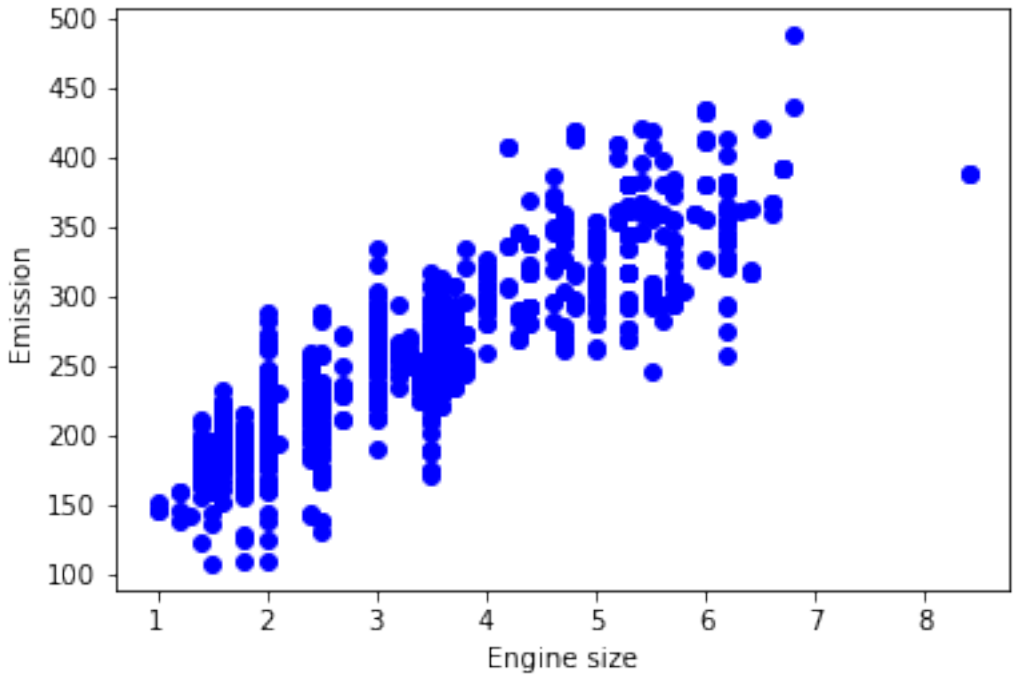
```
      ENGINESIZE  CYLINDERS  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0            2.0          4                   9.9                   6.7

1            2.4          4                  11.2                   7.7

2            1.5          4                   6.0                   5.8

3            3.5          6                  12.7                   9.1

4            3.5          6                  12.1                   8.7

...          ...        ...                   ...                   ...

1062         3.0          6                  13.4                   9.8

1063         3.2          6                  13.2                   9.5

1064         3.0          6                  13.4                   9.8

1065         3.2          6                  12.9                   9.3

1066         3.2          6                  14.9                  10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
```

```
3                    11.1                          25           255
4                    10.6                          27           244
...                   ...                         ...           ...
1062                 11.8                          24           271
1063                 11.5                          25           264
1064                 11.8                          24           271
1065                 11.3                          25           260
1066                 12.8                          22           294

[1067 rows x 7 columns]
```

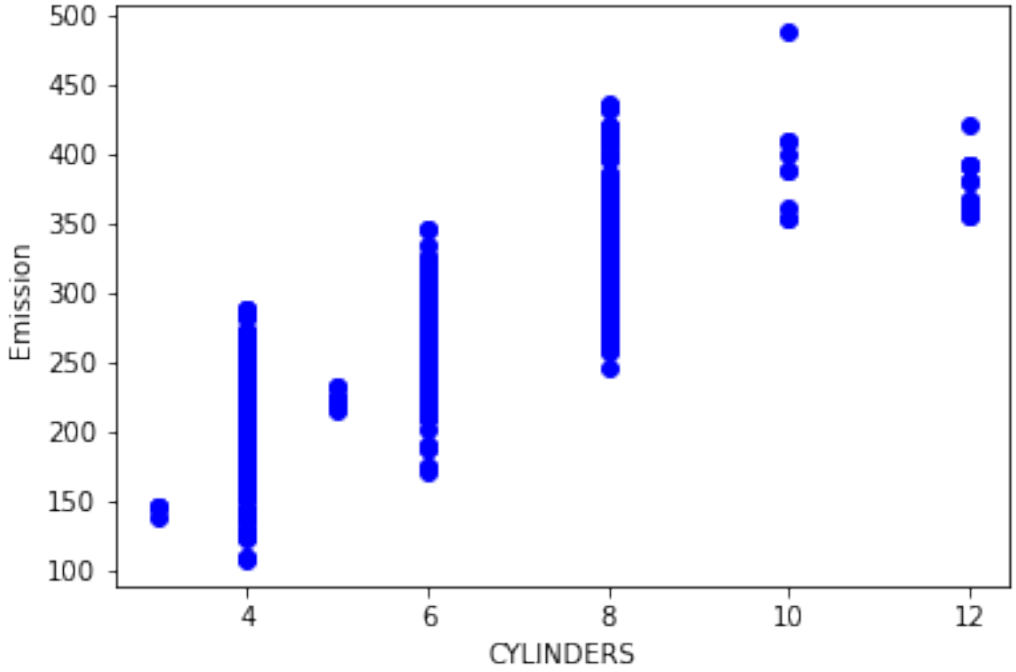
```
cdf =
df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
cdf.head(9)
```

```
   ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
0         2.0          4                   8.5           196
1         2.4          4                   9.6           221
2         1.5          4                   5.9           136
3         3.5          6                  11.1           255
4         3.5          6                  10.6           244
5         3.5          6                  10.0           230
6         3.5          6                  10.1           232
7         3.7          6                  11.1           255
8         3.7          6                  11.6           267
```

```
plt.scatter(cdf.ENGINESIZE, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```

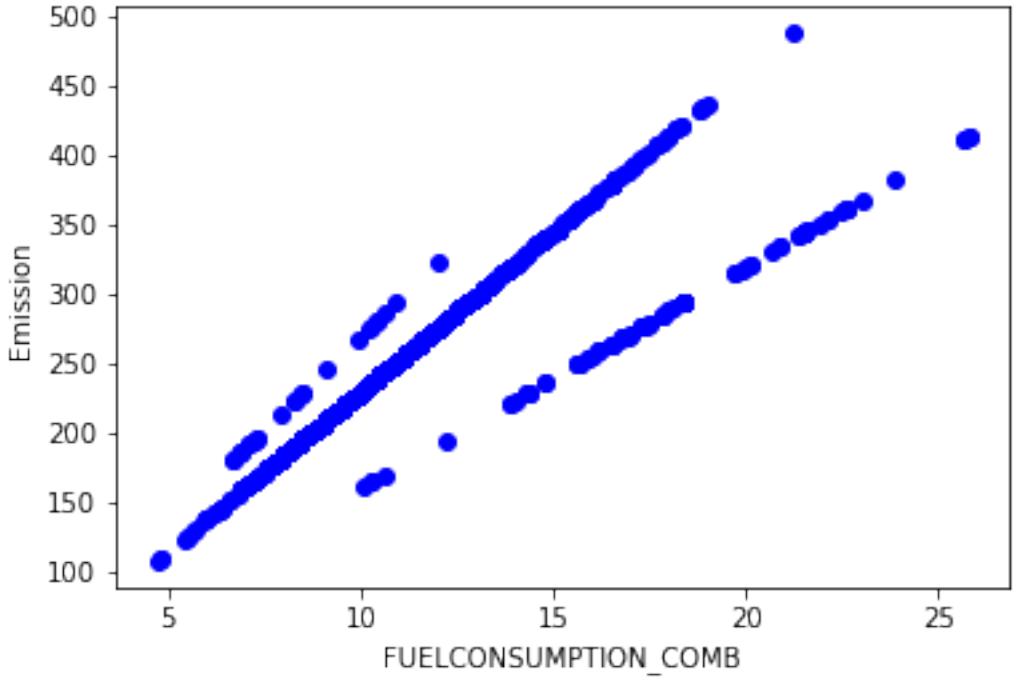



```
plt.scatter(cdf.CYLINDERS, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("CYLINDERS")
plt.ylabel("Emission")
plt.show()
```




```
plt.scatter(cdf.FUELCONSUMPTION_COMB, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("FUELCONSUMPTION_COMB")
```

```
plt.ylabel("Emission")
plt.show()
```



Apply polynomial regression on Fuelconsumption_Comb

Creating train and test dataset

```
len(cdf)
```

```
1067
```

```
cdf
```

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
...	...	...	...	...
1062	3.0	6	11.8	271
1063	3.2	6	11.5	264
1064	3.0	6	11.8	271
1065	3.2	6	11.3	260
1066	3.2	6	12.8	294

```
[1067 rows x 4 columns]
```

```
msk = np.random.rand(len(df)) < 0.8

msk

array([ True,  True,  True, ...,  True,  True,  True])

msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]

train
```

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
4	3.5	6	10.6	244
5	3.5	6	10.0	230
...	...	...	...	...
1062	3.0	6	11.8	271
1063	3.2	6	11.5	264
1064	3.0	6	11.8	271
1065	3.2	6	11.3	260
1066	3.2	6	12.8	294

```
[855 rows x 4 columns]

test
```

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
3	3.5	6	11.1	255
12	5.9	12	15.6	359
23	2.0	4	10.0	230
34	4.0	8	12.5	288
39	3.0	6	11.2	258
...	...	...	...	...
1053	2.0	4	10.7	246
1056	2.5	5	9.7	223
1057	2.5	5	10.1	232
1059	3.2	6	10.2	235
1060	3.0	6	11.5	264

```
[212 rows x 4 columns]
```

Polynomial Regression

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn import linear_model
train_x = np.asanyarray(train[['FUELCONSUMPTION_COMB']]) #independent
train_y = np.asanyarray(train[['CO2EMISSIONS']]) #dependent
#training -> train_x,train_y
```

```
test_x = np.asanyarray(test[['FUELCONSUMPTION_COMB']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
#y_hat=prediction -> test_x
#evaluation -> test_y,y_hat
train_x
```

```
array([[ 8.5],
       [ 9.6],
       [ 5.9],
       [10.6],
       [10. ],
       [10.1],
       [11.1],
       [11.6],
       [ 9.2],
       [ 9.8],
       [10.4],
       [15.6],
       [14.7],
       [15.4],
       [14.7],
       [15.4],
       [15.6],
       [ 8.8],
       [10. ],
       [ 9.3],
       [10. ],
       [ 9.3],
       [10.2],
       [10.9],
       [ 8.3],
       [11.2],
       [ 8.3],
       [11.2],
       [11.3],
       [ 8.4],
       [11.2],
       [15.7],
       [ 8.5],
       [10.5],
       [10.4],
       [ 9.2],
       [13.2],
       [10.9],
       [14.6],
       [17.7],
       [15.4],
       [17.8],
       [13.3],
       [13.4],
```

[12.6],
[11.4],
[12.4],
[11.4],
[12.4],
[11.6],
[12.2],
[12.2],
[12.9],
[12.7],
[9.1],
[9.1],
[10.3],
[10.3],
[15.5],
[13.9],
[16.5],
[14.],
[16.5],
[19.],
[8.4],
[6.7],
[6.7],
[8.4],
[8.7],
[9.6],
[10.3],
[10.1],
[8.4],
[8.7],
[9.1],
[9.6],
[10.3],
[9.1],
[7.9],
[10.1],
[10.7],
[12.7],
[10.1],
[12.7],
[12.2],
[10.7],
[12.7],
[12.7],
[15.5],
[8.3],
[9.3],
[9.2],
[12.7],
[12.7],

[14.7],
[13.8],
[13.8],
[13.8],
[14.7],
[13.8],
[11.3],
[10.],
[11.],
[11.1],
[13.9],
[16.1],
[14.1],
[9.1],
[8.7],
[12.1],
[12.1],
[12.3],
[12.6],
[8.5],
[9.2],
[11.5],
[11.5],
[8.2],
[9.9],
[9.8],
[10.7],
[8.2],
[9.9],
[9.8],
[9.5],
[9.6],
[9.],
[10.9],
[10.1],
[11.3],
[11.4],
[10.1],
[10.8],
[11.3],
[12.8],
[11.],
[11.3],
[11.1],
[14.8],
[16.6],
[14.8],
[14.8],
[14.7],
[20.1],

[16.4],
[22.1],
[12.3],
[11.4],
[11.8],
[11.2],
[11.7],
[10.5],
[12.8],
[12.7],
[16.6],
[14.8],
[12.],
[7.8],
[8.8],
[8.2],
[7.1],
[7.7],
[7.1],
[16.2],
[10.],
[15.1],
[15.9],
[20.9],
[21.5],
[16.5],
[21.5],
[16.5],
[21.5],
[16.5],
[22.6],
[18.],
[18.8],
[25.7],
[18.2],
[25.8],
[9.6],
[10.6],
[8.],
[10.6],
[10.2],
[11.8],
[12.7],
[16.9],
[14.1],
[12.6],
[17.8],
[13.],
[17.3],
[14.3],

```
[ 8. ],
[ 7.5],
[ 8.3],
[ 7.9],
[ 7.5],
[ 7.9],
[ 8.3],
[ 7.9],
[ 8.3],
[ 7.9],
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[13.8],
[18.4],
[12.6],
[ 8.4],
[ 8.1],
[ 9. ],
[10.3],
[10.7],
[10.5],
[14.4],
[ 9.8],
[14.4],
[12.8],
[10.9],
[13.4],
[10.9],
[14.8],
[10.3],
[13.9],
[12. ],
[ 9.8],
[10. ],
[10.5],
[14.4],
[11.2],
[13.1],
[12.8],
[13.8],
[11.2],
[10.3],
[12.8],
[10.9],
[10.9],
[14.8],
```

```
[15.7],
[10.2],
[13.8],
[ 8.6],
[ 8.8],
[ 8.6],
[12.2],
[ 9.2],
[ 8.8],
[ 7.3],
[ 7.3],
[14.4],
[12.1],
[15.6],
[11.1],
[16.2],
[ 7.8],
[ 7.9],
[ 7.2],
[ 7.8],
[ 8.7],
[ 8.4],
[16.2],
[16.8],
[23. ],
[21.2],
[18.3],
[23.9],
[ 9.8],
[10.8],
[11.5],
[ 9.1],
[ 9.6],
[ 9.4],
[ 9.3],
[10. ],
[16. ],
[21.6],
[10.2],
[11.9],
[15.9],
[12.5],
[16.7],
[12.7],
[12.8],
[15.9],
[13.8],
[17.5],
[16.5],
[12.3],
```

[16.5],
[19.7],
[14.6],
[19.7],
[13.4],
[18.],
[13.4],
[18.],
[20.],
[14.8],
[19.9],
[18.],
[7.6],
[8.],
[11.5],
[12.1],
[7.7],
[10.3],
[7.8],
[10.3],
[7.2],
[10.1],
[8.5],
[8.1],
[8.9],
[8.1],
[9.4],
[5.5],
[10.6],
[11.5],
[12.7],
[13.2],
[10.6],
[9.1],
[11.8],
[14.3],
[11.2],
[15.7],
[9.5],
[9.9],
[10.1],
[10.4],
[12.3],
[12.9],
[15.1],
[15.9],
[20.9],
[16.5],
[21.5],
[16.5],

[22.6],
[16.5],
[16.5],
[22.6],
[18.],
[18.8],
[25.7],
[18.2],
[25.8],
[16.8],
[12.7],
[16.9],
[14.1],
[12.6],
[13.],
[17.3],
[14.3],
[9.1],
[12.],
[16.2],
[10.],
[12.6],
[13.8],
[18.4],
[13.8],
[18.4],
[14.7],
[20.1],
[22.1],
[13.8],
[18.4],
[8.1],
[8.7],
[9.1],
[9.6],
[10.9],
[4.8],
[7.2],
[7.6],
[9.1],
[9.4],
[6.3],
[6.9],
[7.5],
[12.],
[7.7],
[8.6],
[8.5],
[13.2],
[11.1],

[11.2],
[11.6],
[10.3],
[11.5],
[11.1],
[9.6],
[8.5],
[6.2],
[6.3],
[9.3],
[10.5],
[10.6],
[8.],
[8.5],
[10.2],
[10.8],
[7.5],
[8.2],
[11.7],
[11.1],
[11.6],
[12.3],
[11.6],
[12.8],
[11.8],
[10.6],
[8.5],
[14.5],
[15.],
[10.3],
[10.6],
[12.8],
[10.5],
[12.2],
[12.4],
[13.3],
[17.5],
[12.4],
[13.3],
[17.5],
[13.3],
[13.3],
[13.1],
[12.6],
[13.7],
[13.5],
[13.7],
[13.5],
[9.5],
[10.5],

[10.],
[10.8],
[11.],
[10.9],
[11.6],
[9.1],
[9.8],
[9.4],
[9.4],
[11.1],
[9.9],
[12.1],
[15.6],
[15.8],
[9.8],
[9.1],
[9.8],
[9.4],
[10.2],
[9.4],
[11.1],
[12.8],
[12.8],
[13.4],
[13.4],
[10.9],
[8.3],
[8.],
[8.4],
[8.4],
[9.7],
[9.8],
[8.4],
[9.4],
[8.4],
[8.6],
[9.9],
[8.7],
[7.6],
[7.7],
[7.6],
[9.2],
[12.],
[10.3],
[11.4],
[11.],
[9.],
[9.1],
[8.8],
[10.],

[11.3],
[11.4],
[10.7],
[18.3],
[15.7],
[17.4],
[11.9],
[14.5],
[9.8],
[9.8],
[15.4],
[15.1],
[12.4],
[15.],
[14.7],
[20.7],
[12.3],
[12.3],
[15.4],
[21.4],
[5.6],
[6.],
[9.6],
[10.4],
[13.9],
[9.7],
[10.4],
[9.7],
[10.6],
[12.6],
[12.9],
[12.9],
[11.4],
[16.5],
[11.7],
[11.1],
[8.2],
[11.8],
[11.3],
[12.8],
[10.2],
[12.2],
[12.2],
[9.],
[6.3],
[16.],
[21.6],
[12.7],
[12.7],
[15.1],

[15.3],
[14.5],
[13.1],
[8.2],
[8.],
[8.6],
[8.5],
[8.9],
[12.6],
[7.7],
[6.9],
[7.],
[7.4],
[7.3],
[7.2],
[7.7],
[7.3],
[9.7],
[9.8],
[7.6],
[7.3],
[10.2],
[9.6],
[8.],
[9.4],
[9.4],
[10.5],
[10.5],
[10.6],
[15.1],
[15.3],
[13.4],
[7.8],
[9.2],
[11.5],
[13.1],
[13.1],
[10.],
[10.4],
[10.5],
[10.4],
[10.2],
[8.8],
[11.6],
[11.7],
[11.4],
[12.9],
[13.5],
[13.5],
[17.7],

[12.],
[15.2],
[15.6],
[15.8],
[8.5],
[11.3],
[12.1],
[16.],
[10.3],
[14.2],
[15.5],
[12.1],
[12.],
[14.2],
[8.9],
[9.],
[9.8],
[16.5],
[15.7],
[15.7],
[7.8],
[7.8],
[8.7],
[7.8],
[7.6],
[8.7],
[7.8],
[7.8],
[7.8],
[8.3],
[8.],
[8.3],
[8.],
[9.],
[8.5],
[8.3],
[8.],
[9.],
[8.5],
[8.3],
[8.],
[8.3],
[8.],
[8.3],
[8.],
[8.5],
[8.3],
[8.],
[8.5],
[8.3],

```
[ 8. ],
[ 8.2],
[ 8.3],
[12.5],
[11.6],
[ 8.6],
[ 8.6],
[ 6. ],
[ 6.4],
[ 8.6],
[ 9.1],
[10.2],
[ 8.7],
[ 8.8],
[ 9. ],
[11. ],
[11.5],
[11.7],
[ 9.3],
[16.5],
[12.3],
[11.3],
[12.7],
[13.7],
[13.1],
[12.9],
[ 8.1],
[ 8.6],
[ 8.7],
[11.8],
[10.6],
[11. ],
[ 8.8],
[ 8.3],
[ 7. ],
[ 7.8],
[15.6],
[17.3],
[ 6.6],
[ 7.7],
[14.2],
[13.7],
[10.1],
[10.7],
[10.3],
[10.7],
[10.5],
[11. ],
[10.8],
[11. ],
```

```
[10.1],
[10.7],
[10.6],
[10.8],
[10.7],
[11.9],
[11.9],
[11.9],
[11.9],
[ 9.2],
[ 9.9],
[10.4],
[12. ],
[13.6],
[10.4],
[13.9],
[13. ],
[10.9],
[13.7],
[ 9.2],
[ 9.9],
[ 9.8],
[10.4],
[10.9],
[11.1],
[12.8],
[12.9],
[14.8],
[13.6],
[14.1],
[10.6],
[12.6],
[17.8],
[10.2],
[11.9],
[16.8],
[11.3],
[11.4],
[16. ],
[17.1],
[17.1],
[17.1],
[17.1],
[15.6],
[ 8.4],
[ 9.5],
[ 6.2],
[ 9. ],
[ 9. ],
[ 9.9],
```

```
[ 9.9],
[ 8. ],
[ 7.9],
[ 6.4],
[ 6.4],
[16.9],
[16.9],
[ 8.5],
[ 9.6],
[ 9.5],
[ 8.6],
[ 9.7],
[ 7.8],
[ 8.5],
[12.5],
[ 8.5],
[11.2],
[12.5],
[ 8.6],
[ 9.9],
[11.6],
[ 9. ],
[11.7],
[13.1],
[ 8.5],
[ 9.2],
[ 7.7],
[12.9],
[12.9],
[ 9.6],
[ 8.2],
[ 9.6],
[ 5.7],
[ 6. ],
[ 7.7],
[ 7.3],
[ 7.1],
[ 7.4],
[ 6.8],
[ 8.5],
[12.9],
[13.9],
[10.3],
[11.2],
[11.5],
[ 8.3],
[ 8.2],
[ 4.7],
[ 5.5],
[ 8.9],
```

[9.4],
[9.5],
[16.7],
[11.3],
[12.4],
[10.9],
[10.2],
[13.1],
[13.3],
[14.],
[13.5],
[14.3],
[15.5],
[15.1],
[16.2],
[10.],
[11.],
[10.2],
[7.3],
[7.1],
[8.5],
[8.7],
[9.1],
[9.4],
[9.6],
[8.7],
[9.3],
[9.4],
[10.1],
[7.3],
[7.3],
[12.3],
[9.6],
[9.5],
[9.4],
[7.2],
[8.2],
[8.2],
[9.4],
[9.],
[9.],
[6.9],
[6.9],
[5.4],
[8.6],
[9.7],
[9.8],
[7.1],
[6.8],
[12.1],

```
       [10.7],
       [11.6],
       [12.2],
       [10.4],
       [11.5],
       [11.2],
       [11.8],
       [11.5],
       [11.8],
       [11.3],
       [12.8]])
```

```
train_x[0:5]
```

```
array([[ 8.5],
       [ 9.6],
       [ 5.9],
       [10.6],
       [10. ]])
```

```
poly = PolynomialFeatures(degree=2)
train_x_poly = poly.fit_transform(train_x)
train_x_poly
```

```
array([[  1.  ,   8.5 ,  72.25],
       [  1.  ,   9.6 ,  92.16],
       [  1.  ,   5.9 ,  34.81],
       ...,
       [  1.  ,  11.8 , 139.24],
       [  1.  ,  11.3 , 127.69],
       [  1.  ,  12.8 , 163.84]])
```

```
clf = linear_model.LinearRegression()
train_y_ = clf.fit(train_x_poly, train_y)
# The coefficients
print ('Coefficients: ', clf.coef_)
print ('Intercept: ',clf.intercept_)
```

```
Coefficients:  [[ 0.         39.81542284 -0.87453348]]
Intercept:  [-76.35808794]
```

```
import numpy as np
```

```
np.arange(0,10,1)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
np.arange(0,10,0.5)
```

```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5,
6. ,
       6.5, 7. , 7.5, 8. , 8.5, 9. , 9.5])
```

```
print(clf.intercept_,"----",clf.coef_)
```

```
[-76.35808794] ---- [[ 0.         39.81542284 -0.87453348]]
```

```
clf.intercept_[0]
```

```
-76.35808794149165
```

```
clf.coef_[0][0]
```

```
0.0
```

```
clf.coef_[0][1]
```

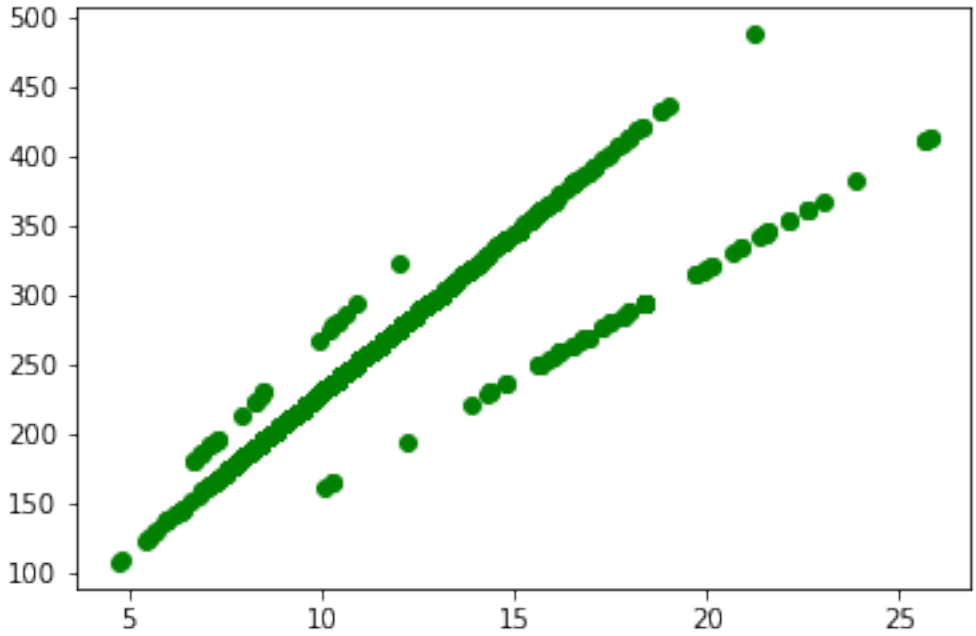
```
39.81542284160467
```

```
clf.coef_[0][2]
```

```
-0.8745334755747078
```

```
plt.scatter(train.FUELCONSUMPTION_COMB, train.CO2EMISSIONS,
color='green')
```

```
<matplotlib.collections.PathCollection at 0x19e677ff4f0>
```




```
XX = np.arange(0, 10, .1)
```

```
XX
```

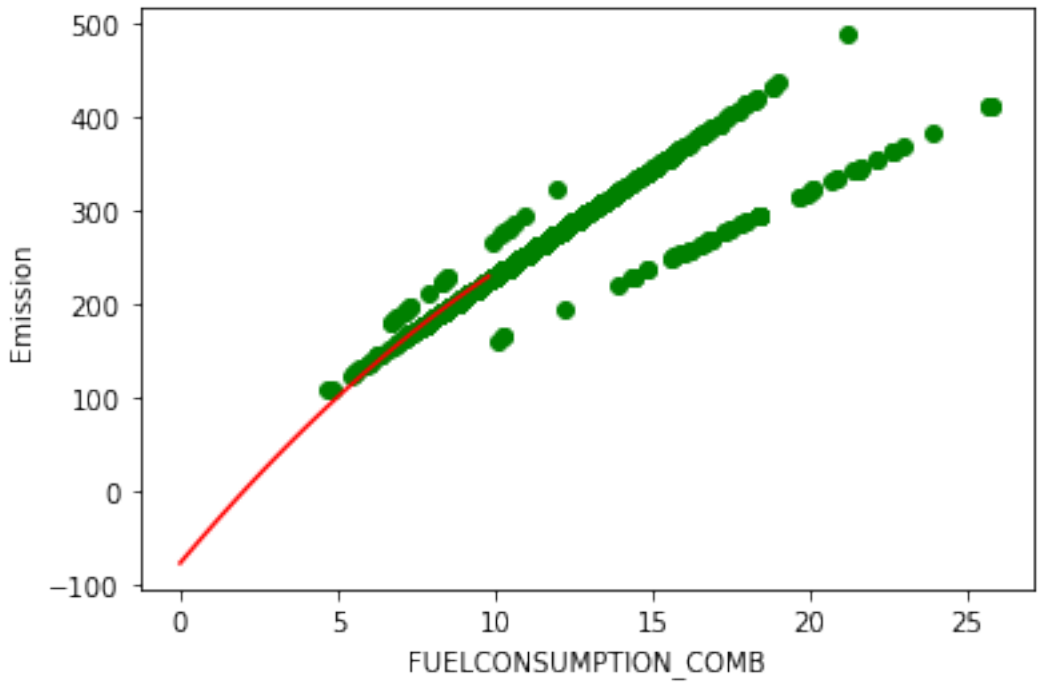
```
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1,
1.2,
       1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1, 2.2, 2.3, 2.4,
2.5,
```

```
       2.6, 2.7, 2.8, 2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7,
3.8,
       3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5. ,
5.1,
       5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6. , 6.1, 6.2, 6.3,
6.4,
       6.5, 6.6, 6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3, 7.4, 7.5, 7.6,
7.7,
       7.8, 7.9, 8. , 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9,
9. ,
       9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9])

#y = b + m1xx + m2xx**2

yy = clf.intercept_[0]+ clf.coef_[0][1]*XX+ clf.coef_[0]
[2]*np.power(XX, 2)

yy

array([-7.63580879e+01, -7.23852910e+01, -6.84299847e+01, -
6.44921691e+01,
       -6.05718442e+01, -5.66690099e+01, -5.27836663e+01, -
4.89158134e+01,
       -4.50654511e+01, -4.12325795e+01, -3.74171986e+01, -
3.36193083e+01,
       -2.98389087e+01, -2.60759998e+01, -2.23305816e+01, -
1.86026540e+01,
       -1.48922171e+01, -1.11992709e+01, -7.52381529e+00, -
3.86585039e+00,
       -2.25376161e-01,  3.39760740e+00,  7.00310029e+00,
1.05911025e+01,
        1.41616141e+01,  1.77146349e+01,  2.12501652e+01,
2.47682047e+01,
        2.82687536e+01,  3.17518118e+01,  3.52173793e+01,
3.86654562e+01,
        4.20960424e+01,  4.55091379e+01,  4.89047427e+01,
5.22828569e+01,
        5.56434804e+01,  5.89866133e+01,  6.23122555e+01,
6.56204070e+01,
        6.89110678e+01,  7.21842380e+01,  7.54399175e+01,
7.86781063e+01,
        8.18988045e+01,  8.51020120e+01,  8.82877288e+01,
9.14559549e+01,
        9.46066904e+01,  9.77399352e+01,  1.00855689e+02,
1.03953953e+02,
        1.07034726e+02,  1.10098008e+02,  1.13143799e+02,
1.16172100e+02,
        1.19182910e+02,  1.22176230e+02,  1.25152058e+02,
1.28110397e+02,
        1.31051244e+02,  1.33974601e+02,  1.36880467e+02,
1.39768842e+02,
```

```
        1.42639727e+02,  1.45493121e+02,  1.48329025e+02,
1.51147437e+02,
        1.53948359e+02,  1.56731791e+02,  1.59497732e+02,
1.62246182e+02,
        1.64977141e+02,  1.67690610e+02,  1.70386588e+02,
1.73065075e+02,
        1.75726072e+02,  1.78369578e+02,  1.80995594e+02,
1.83604118e+02,
        1.86195152e+02,  1.88768696e+02,  1.91324748e+02,
1.93863311e+02,
        1.96384382e+02,  1.98887963e+02,  2.01374053e+02,
2.03842652e+02,
        2.06293761e+02,  2.08727379e+02,  2.11143506e+02,
2.13542143e+02,
        2.15923289e+02,  2.18286944e+02,  2.20633109e+02,
2.22961783e+02,
        2.25272966e+02,  2.27566659e+02,  2.29842861e+02,
2.32101572e+02])
```

```
import matplotlib.pyplot as plt
plt.scatter(train.FUELCONSUMPTION_COMB, train.CO2EMISSIONS,
color='green')
XX = np.arange(0, 10 , 0.2)
yy = clf.intercept_[0]+ clf.coef_[0][1]*XX+ clf.coef_[0]
[2]*np.power(XX, 2)
plt.plot(XX, yy, 'r' )
plt.xlabel("FUELCONSUMPTION_COMB")
plt.ylabel("Emission")
```

```
Text(0, 0.5, 'Emission')
```



Evaluation

```
from sklearn.metrics import r2_score

test_x_poly = poly.fit_transform(test_x)

test_x

array([[11.1],
       [15.6],
       [10. ],
       [12.5],
       [11.2],
       [ 9.1],
       [14.6],
       [17.7],
       [15.4],
       [17.8],
       [12.9],
       [16.5],
       [ 8.7],
       [ 8.8],
       [ 9.1],
       [ 9.1],
       [ 9.1],
       [10. ],
       [ 9.9],
       [10. ],
       [ 9.9],
```

```
[ 8.8],
[12.2],
[12.7],
[14.7],
[14.7],
[ 9.1],
[11.9],
[11. ],
[15.6],
[16.5],
[10.2],
[11.2],
[10.8],
[11.4],
[16.6],
[16.6],
[12.8],
[12.6],
[11.2],
[ 7.8],
[ 9.1],
[12. ],
[12.6],
[17. ],
[16.5],
[22.6],
[16.5],
[18.9],
[14.3],
[ 8.7],
[ 9.9],
[16.8],
[ 8. ],
[ 8.3],
[ 6.9],
[ 6.9],
[12.3],
[10.1],
[10. ],
[10.5],
[13.8],
[16.8],
[13.9],
[13.4],
[11.2],
[13.9],
[ 8.1],
[11.8],
[16.5],
[12.4],
```

```
[12. ],
[ 7.8],
[ 7.8],
[ 7.9],
[ 6.8],
[ 6. ],
[21.9],
[17.2],
[22.5],
[12.2],
[13. ],
[13.4],
[12.3],
[14.6],
[14.9],
[ 7.6],
[ 6.6],
[ 7. ],
[12.8],
[ 8.9],
[ 8. ],
[10.6],
[ 9. ],
[10.2],
[10.3],
[21.5],
[16.5],
[21.5],
[18.9],
[11.8],
[17.8],
[17. ],
[16.4],
[18.4],
[13.8],
[ 7.9],
[ 7. ],
[10.7],
[ 7.7],
[10.6],
[11.5],
[13.4],
[ 7.7],
[10.1],
[10. ],
[ 7.7],
[ 8.8],
[11.5],
[12.3],
[11. ],
```

```
[ 7.6],
[11. ],
[12.8],
[13.3],
[18.1],
[13.3],
[10.1],
[ 9.8],
[10.2],
[14.8],
[ 8.5],
[ 9.6],
[ 9.3],
[11.8],
[ 8.9],
[21.4],
[12.4],
[17.4],
[17.3],
[15.1],
[11. ],
[ 7.6],
[10.6],
[11. ],
[12.3],
[11.4],
[12.3],
[ 7.4],
[ 7. ],
[ 7.1],
[ 7.2],
[ 8.1],
[ 9.9],
[14. ],
[10.4],
[10.5],
[13.2],
[ 7.2],
[10.2],
[12.9],
[18.2],
[13.2],
[11.7],
[12.7],
[10.7],
[ 7.8],
[ 7.8],
[ 7.3],
[ 9. ],
[ 9. ],
```

```
       [ 9.5],
       [12.3],
       [11.4],
       [ 7.6],
       [10.9],
       [ 9.3],
       [11.1],
       [ 8.6],
       [11. ],
       [11.2],
       [10.9],
       [14. ],
       [ 9.8],
       [11.3],
       [11.5],
       [12.7],
       [11.3],
       [15.4],
       [15.9],
       [16. ],
       [11.2],
       [ 7.8],
       [ 9.8],
       [ 6.9],
       [ 8.2],
       [ 4.8],
       [11.9],
       [11.8],
       [11.3],
       [ 9.8],
       [ 9.7],
       [ 9.8],
       [ 6.9],
       [ 9.2],
       [ 8.6],
       [10.8],
       [10.7],
       [ 9.7],
       [10.1],
       [10.2],
       [11.5]])
```

```
test_x_poly
```

```
array([[  1.  ,  11.1 , 123.21],
       [  1.  ,  15.6 , 243.36],
       [  1.  ,  10.  , 100.  ],
       [  1.  ,  12.5 , 156.25],
       [  1.  ,  11.2 , 125.44],
       [  1.  ,   9.1 ,  82.81],
       [  1.  ,  14.6 , 213.16],
```

```
[  1.  ,  17.7 , 313.29],
[  1.  ,  15.4 , 237.16],
[  1.  ,  17.8 , 316.84],
[  1.  ,  12.9 , 166.41],
[  1.  ,  16.5 , 272.25],
[  1.  ,   8.7 ,  75.69],
[  1.  ,   8.8 ,  77.44],
[  1.  ,   9.1 ,  82.81],
[  1.  ,   9.1 ,  82.81],
[  1.  ,   9.1 ,  82.81],
[  1.  ,  10.  , 100.  ],
[  1.  ,   9.9 ,  98.01],
[  1.  ,  10.  , 100.  ],
[  1.  ,   9.9 ,  98.01],
[  1.  ,   8.8 ,  77.44],
[  1.  ,  12.2 , 148.84],
[  1.  ,  12.7 , 161.29],
[  1.  ,  14.7 , 216.09],
[  1.  ,  14.7 , 216.09],
[  1.  ,   9.1 ,  82.81],
[  1.  ,  11.9 , 141.61],
[  1.  ,  11.  , 121.  ],
[  1.  ,  15.6 , 243.36],
[  1.  ,  16.5 , 272.25],
[  1.  ,  10.2 , 104.04],
[  1.  ,  11.2 , 125.44],
[  1.  ,  10.8 , 116.64],
[  1.  ,  11.4 , 129.96],
[  1.  ,  16.6 , 275.56],
[  1.  ,  16.6 , 275.56],
[  1.  ,  12.8 , 163.84],
[  1.  ,  12.6 , 158.76],
[  1.  ,  11.2 , 125.44],
[  1.  ,   7.8 ,  60.84],
[  1.  ,   9.1 ,  82.81],
[  1.  ,  12.  , 144.  ],
[  1.  ,  12.6 , 158.76],
[  1.  ,  17.  , 289.  ],
[  1.  ,  16.5 , 272.25],
[  1.  ,  22.6 , 510.76],
[  1.  ,  16.5 , 272.25],
[  1.  ,  18.9 , 357.21],
[  1.  ,  14.3 , 204.49],
[  1.  ,   8.7 ,  75.69],
[  1.  ,   9.9 ,  98.01],
[  1.  ,  16.8 , 282.24],
[  1.  ,   8.  ,  64.  ],
[  1.  ,   8.3 ,  68.89],
[  1.  ,   6.9 ,  47.61],
[  1.  ,   6.9 ,  47.61],
```

```
[  1.  ,  12.3 , 151.29],
[  1.  ,  10.1 , 102.01],
[  1.  ,  10.  , 100.  ],
[  1.  ,  10.5 , 110.25],
[  1.  ,  13.8 , 190.44],
[  1.  ,  16.8 , 282.24],
[  1.  ,  13.9 , 193.21],
[  1.  ,  13.4 , 179.56],
[  1.  ,  11.2 , 125.44],
[  1.  ,  13.9 , 193.21],
[  1.  ,   8.1 ,  65.61],
[  1.  ,  11.8 , 139.24],
[  1.  ,  16.5 , 272.25],
[  1.  ,  12.4 , 153.76],
[  1.  ,  12.  , 144.  ],
[  1.  ,   7.8 ,  60.84],
[  1.  ,   7.8 ,  60.84],
[  1.  ,   7.9 ,  62.41],
[  1.  ,   6.8 ,  46.24],
[  1.  ,   6.  ,  36.  ],
[  1.  ,  21.9 , 479.61],
[  1.  ,  17.2 , 295.84],
[  1.  ,  22.5 , 506.25],
[  1.  ,  12.2 , 148.84],
[  1.  ,  13.  , 169.  ],
[  1.  ,  13.4 , 179.56],
[  1.  ,  12.3 , 151.29],
[  1.  ,  14.6 , 213.16],
[  1.  ,  14.9 , 222.01],
[  1.  ,   7.6 ,  57.76],
[  1.  ,   6.6 ,  43.56],
[  1.  ,   7.  ,  49.  ],
[  1.  ,  12.8 , 163.84],
[  1.  ,   8.9 ,  79.21],
[  1.  ,   8.  ,  64.  ],
[  1.  ,  10.6 , 112.36],
[  1.  ,   9.  ,  81.  ],
[  1.  ,  10.2 , 104.04],
[  1.  ,  10.3 , 106.09],
[  1.  ,  21.5 , 462.25],
[  1.  ,  16.5 , 272.25],
[  1.  ,  21.5 , 462.25],
[  1.  ,  18.9 , 357.21],
[  1.  ,  11.8 , 139.24],
[  1.  ,  17.8 , 316.84],
[  1.  ,  17.  , 289.  ],
[  1.  ,  16.4 , 268.96],
[  1.  ,  18.4 , 338.56],
[  1.  ,  13.8 , 190.44],
[  1.  ,   7.9 ,  62.41],
```

```
[  1.  ,   7.  ,  49.  ],
[  1.  ,  10.7 , 114.49],
[  1.  ,   7.7 ,  59.29],
[  1.  ,  10.6 , 112.36],
[  1.  ,  11.5 , 132.25],
[  1.  ,  13.4 , 179.56],
[  1.  ,   7.7 ,  59.29],
[  1.  ,  10.1 , 102.01],
[  1.  ,  10.  , 100.  ],
[  1.  ,   7.7 ,  59.29],
[  1.  ,   8.8 ,  77.44],
[  1.  ,  11.5 , 132.25],
[  1.  ,  12.3 , 151.29],
[  1.  ,  11.  , 121.  ],
[  1.  ,   7.6 ,  57.76],
[  1.  ,  11.  , 121.  ],
[  1.  ,  12.8 , 163.84],
[  1.  ,  13.3 , 176.89],
[  1.  ,  18.1 , 327.61],
[  1.  ,  13.3 , 176.89],
[  1.  ,  10.1 , 102.01],
[  1.  ,   9.8 ,  96.04],
[  1.  ,  10.2 , 104.04],
[  1.  ,  14.8 , 219.04],
[  1.  ,   8.5 ,  72.25],
[  1.  ,   9.6 ,  92.16],
[  1.  ,   9.3 ,  86.49],
[  1.  ,  11.8 , 139.24],
[  1.  ,   8.9 ,  79.21],
[  1.  ,  21.4 , 457.96],
[  1.  ,  12.4 , 153.76],
[  1.  ,  17.4 , 302.76],
[  1.  ,  17.3 , 299.29],
[  1.  ,  15.1 , 228.01],
[  1.  ,  11.  , 121.  ],
[  1.  ,   7.6 ,  57.76],
[  1.  ,  10.6 , 112.36],
[  1.  ,  11.  , 121.  ],
[  1.  ,  12.3 , 151.29],
[  1.  ,  11.4 , 129.96],
[  1.  ,  12.3 , 151.29],
[  1.  ,   7.4 ,  54.76],
[  1.  ,   7.  ,  49.  ],
[  1.  ,   7.1 ,  50.41],
[  1.  ,   7.2 ,  51.84],
[  1.  ,   8.1 ,  65.61],
[  1.  ,   9.9 ,  98.01],
[  1.  ,  14.  , 196.  ],
[  1.  ,  10.4 , 108.16],
[  1.  ,  10.5 , 110.25],
```

```
[  1.  ,  13.2 , 174.24],
[  1.  ,   7.2 ,  51.84],
[  1.  ,  10.2 , 104.04],
[  1.  ,  12.9 , 166.41],
[  1.  ,  18.2 , 331.24],
[  1.  ,  13.2 , 174.24],
[  1.  ,  11.7 , 136.89],
[  1.  ,  12.7 , 161.29],
[  1.  ,  10.7 , 114.49],
[  1.  ,   7.8 ,  60.84],
[  1.  ,   7.8 ,  60.84],
[  1.  ,   7.3 ,  53.29],
[  1.  ,   9.  ,  81.  ],
[  1.  ,   9.  ,  81.  ],
[  1.  ,   9.5 ,  90.25],
[  1.  ,  12.3 , 151.29],
[  1.  ,  11.4 , 129.96],
[  1.  ,   7.6 ,  57.76],
[  1.  ,  10.9 , 118.81],
[  1.  ,   9.3 ,  86.49],
[  1.  ,  11.1 , 123.21],
[  1.  ,   8.6 ,  73.96],
[  1.  ,  11.  , 121.  ],
[  1.  ,  11.2 , 125.44],
[  1.  ,  10.9 , 118.81],
[  1.  ,  14.  , 196.  ],
[  1.  ,   9.8 ,  96.04],
[  1.  ,  11.3 , 127.69],
[  1.  ,  11.5 , 132.25],
[  1.  ,  12.7 , 161.29],
[  1.  ,  11.3 , 127.69],
[  1.  ,  15.4 , 237.16],
[  1.  ,  15.9 , 252.81],
[  1.  ,  16.  , 256.  ],
[  1.  ,  11.2 , 125.44],
[  1.  ,   7.8 ,  60.84],
[  1.  ,   9.8 ,  96.04],
[  1.  ,   6.9 ,  47.61],
[  1.  ,   8.2 ,  67.24],
[  1.  ,   4.8 ,  23.04],
[  1.  ,  11.9 , 141.61],
[  1.  ,  11.8 , 139.24],
[  1.  ,  11.3 , 127.69],
[  1.  ,   9.8 ,  96.04],
[  1.  ,   9.7 ,  94.09],
[  1.  ,   9.8 ,  96.04],
[  1.  ,   6.9 ,  47.61],
[  1.  ,   9.2 ,  84.64],
[  1.  ,   8.6 ,  73.96],
[  1.  ,  10.8 , 116.64],
```

```
       [  1.  ,  10.7 , 114.49],
       [  1.  ,   9.7 ,  94.09],
       [  1.  ,  10.1 , 102.01],
       [  1.  ,  10.2 , 104.04],
       [  1.  ,  11.5 , 132.25]])

pred = clf.predict(test_x_poly)

print(f"Mean absolute error: {np.mean(np.absolute(pred - test_y))}")
print(f"Residual sum of squares (MSE): {np.mean((pred - test_y) **
2)}")
print(f"R2-score: {r2_score(pred , test_y)}")

Mean absolute error: 13.976264079815174
Residual sum of squares (MSE): 699.408567190187
R2-score: 0.8066327979825534
```

New predictions get value of Engine Size transform it and then predict

```
transformed_data = poly.fit_transform([[3.25]])

transformed_data

array([[ 1.    ,  3.25  , 10.5625]])

clf.predict(transformed_data)
#clf.predict([[3.15]])

array([[43.80477646]])
```

Que.3 : Apply multi Linear regression to the Housing Price Data Set Note : You can take any number of Independent Variable Note : You need to make 3 models atleast with different number of indepent variable Note : Try to get the best possible accuracy

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/housing.csv")

df

        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
...    ...         ...      ...          ...      ...    ...   ...
...
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
3            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
...          ...       ...  ...      ...    ...    ...         ...
...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
```

```
...     ...    ...        ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[1460 rows x 81 columns]
```

```
cdf = df[["1stFlrSF","2ndFlrSF","OpenPorchSF","SalePrice"]]
```

```
cdf
```

```
      1stFlrSF  2ndFlrSF  OpenPorchSF  SalePrice
0          856       854           61     208500
1         1262         0            0     181500
2          920       866           42     223500
3          961       756           35     140000
4         1145      1053           84     250000
...        ...       ...          ...        ...
1455       953       694           40     175000
1456      2073         0            0     210000
1457      1188      1152           60     266500
1458      1078         0            0     142125
1459      1256         0           68     147500

[1460 rows x 4 columns]
```

```
cdf.head()
```

```
   1stFlrSF  2ndFlrSF  OpenPorchSF  SalePrice
0       856       854           61     208500
1      1262         0            0     181500
2       920       866           42     223500
3       961       756           35     140000
4      1145      1053           84     250000
```

```
cdf.tail()
```

```
      1stFlrSF  2ndFlrSF  OpenPorchSF  SalePrice
1455       953       694           40     175000
1456      2073         0            0     210000
1457      1188      1152           60     266500
1458      1078         0            0     142125
1459      1256         0           68     147500
```

```
cdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
```

```
---  ------       --------------  -----
 0   1stFlrSF     1460 non-null   int64
 1   2ndFlrSF     1460 non-null   int64
 2   OpenPorchSF  1460 non-null   int64
 3   SalePrice    1460 non-null   int64
dtypes: int64(4)
memory usage: 45.8 KB
```

```
cdf.describe()
```

```
          1stFlrSF     2ndFlrSF  OpenPorchSF      SalePrice
count  1460.000000  1460.000000  1460.000000    1460.000000
mean   1162.626712   346.992466    46.660274  180921.195890
std     386.587738   436.528436    66.256028   79442.502883
min     334.000000     0.000000     0.000000   34900.000000
25%     882.000000     0.000000     0.000000  129975.000000
50%    1087.000000     0.000000    25.000000  163000.000000
75%    1391.250000   728.000000    68.000000  214000.000000
max    4692.000000  2065.000000   547.000000  755000.000000
```

```
cdf.isnull().sum()
```

```
1stFlrSF       0
2ndFlrSF       0
OpenPorchSF    0
SalePrice      0
dtype: int64
```

```
X = np.asarray(df[["1stFlrSF","2ndFlrSF","OpenPorchSF"]]) #
Independent Variable
```

```
X
```

```
array([[ 856,  854,   61],
       [1262,    0,    0],
       [ 920,  866,   42],
       ...,
       [1188, 1152,   60],
       [1078,    0,    0],
       [1256,    0,   68]], dtype=int64)
```

```
set(df["1stFlrSF"])
```

```
{334,
 372,
 438,
 480,
 483,
 495,
 520,
 525,
 526,
```

536,
546,
551,
561,
572,
575,
576,
581,
596,
600,
605,
612,
616,
624,
625,
626,
630,
649,
658,
660,
661,
663,
664,
672,
673,
676,
679,
680,
682,
684,
686,
689,
691,
693,
694,
696,
697,
698,
702,
703,
707,
708,
713,
716,
720,
725,
728,
729,
734,
735,

736,
738,
741,
742,
747,
750,
751,
752,
753,
754,
755,
756,
757,
760,
764,
765,
767,
768,
769,
770,
772,
773,
774,
778,
779,
780,
783,
784,
786,
788,
789,
790,
792,
793,
794,
796,
798,
799,
800,
802,
803,
804,
806,
807,
808,
810,
811,
812,
813,
814,

815,
816,
818,
820,
822,
824,
825,
827,
829,
831,
832,
833,
835,
838,
840,
841,
842,
845,
846,
847,
848,
849,
851,
854,
855,
856,
858,
859,
860,
861,
864,
865,
866,
869,
872,
874,
875,
876,
877,
879,
880,
882,
884,
885,
886,
887,
888,
889,
892,
893,

894,
896,
897,
899,
900,
901,
902,
904,
905,
907,
908,
909,
910,
912,
913,
914,
915,
916,
918,
920,
923,
924,
925,
926,
927,
928,
929,
930,
932,
933,
935,
936,
938,
939,
941,
943,
944,
948,
949,
950,
951,
952,
953,
954,
955,
956,
958,
959,
960,
961,

962,
963,
964,
965,
966,
968,
969,
970,
971,
972,
974,
975,
976,
977,
979,
980,
981,
983,
984,
985,
986,
988,
989,
990,
991,
992,
993,
996,
997,
998,
999,
1001,
1002,
1003,
1004,
1005,
1006,
1007,
1008,
1010,
1012,
1013,
1014,
1015,
1020,
1021,
1022,
1024,
1026,
1028,

1029,
1032,
1034,
1035,
1036,
1038,
1039,
1040,
1041,
1042,
1044,
1047,
1048,
1050,
1051,
1052,
1053,
1054,
1055,
1056,
1057,
1058,
1060,
1061,
1062,
1063,
1064,
1065,
1067,
1068,
1069,
1071,
1072,
1073,
1074,
1075,
1077,
1078,
1079,
1080,
1082,
1085,
1086,
1088,
1089,
1090,
1091,
1092,
1094,
1095,

1096,
1097,
1098,
1099,
1100,
1103,
1104,
1105,
1107,
1108,
1110,
1112,
1113,
1114,
1116,
1117,
1118,
1120,
1121,
1122,
1124,
1125,
1126,
1127,
1128,
1130,
1131,
1132,
1133,
1134,
1136,
1137,
1138,
1140,
1141,
1142,
1143,
1144,
1145,
1146,
1148,
1149,
1150,
1152,
1153,
1154,
1155,
1156,
1158,
1159,

1160,
1161,
1163,
1164,
1165,
1166,
1167,
1168,
1170,
1172,
1173,
1175,
1178,
1180,
1181,
1182,
1184,
1186,
1187,
1188,
1190,
1192,
1194,
1195,
1196,
1199,
1200,
1203,
1204,
1207,
1208,
1210,
1211,
1212,
1214,
1215,
1216,
1217,
1218,
1220,
1221,
1222,
1223,
1224,
1225,
1226,
1228,
1229,
1232,
1234,

1235,
1236,
1238,
1240,
1241,
1242,
1244,
1246,
1247,
1248,
1249,
1251,
1252,
1253,
1256,
1258,
1260,
1261,
1262,
1264,
1265,
1266,
1268,
1269,
1272,
1274,
1276,
1277,
1279,
1281,
1282,
1283,
1284,
1285,
1287,
1288,
1291,
1294,
1296,
1297,
1298,
1299,
1301,
1302,
1304,
1306,
1307,
1309,
1310,
1314,

1316,
1318,
1319,
1320,
1324,
1325,
1327,
1328,
1332,
1334,
1336,
1337,
1338,
1339,
1340,
1342,
1344,
1349,
1350,
1352,
1357,
1358,
1360,
1361,
1362,
1363,
1364,
1368,
1370,
1372,
1375,
1377,
1378,
1381,
1382,
1383,
1389,
1390,
1391,
1392,
1394,
1402,
1405,
1407,
1411,
1412,
1414,
1416,
1419,
1422,

1423,
1425,
1426,
1428,
1429,
1430,
1431,
1432,
1434,
1436,
1437,
1440,
1442,
1444,
1445,
1453,
1454,
1456,
1459,
1462,
1464,
1465,
1466,
1468,
1470,
1472,
1473,
1476,
1478,
1479,
1482,
1484,
1486,
1489,
1490,
1493,
1494,
1496,
1498,
1500,
1501,
1502,
1504,
1505,
1506,
1507,
1509,
1512,
1516,
1518,

1520,
1521,
1523,
1525,
1526,
1530,
1532,
1533,
1535,
1536,
1537,
1541,
1542,
1547,
1548,
1549,
1552,
1553,
1554,
1555,
1557,
1559,
1560,
1561,
1563,
1566,
1567,
1569,
1571,
1572,
1573,
1574,
1575,
1576,
1578,
1580,
1582,
1584,
1586,
1588,
1593,
1600,
1601,
1602,
1604,
1610,
1614,
1616,
1617,
1619,

1620,
1621,
1622,
1624,
1625,
1629,
1630,
1632,
1634,
1636,
1640,
1644,
1646,
1647,
1651,
1652,
1654,
1656,
1657,
1659,
1660,
1661,
1663,
1664,
1668,
1670,
1675,
1679,
1680,
1682,
1684,
1686,
1687,
1689,
1690,
1694,
1696,
1698,
1699,
1700,
1701,
1702,
1704,
1707,
1709,
1710,
1712,
1713,
1717,
1718,

1719,
1720,
1721,
1724,
1726,
1728,
1733,
1734,
1742,
1746,
1752,
1766,
1768,
1771,
1776,
1779,
1787,
1788,
1792,
1795,
1800,
1801,
1803,
1811,
1824,
1826,
1828,
1831,
1836,
1838,
1839,
1842,
1844,
1848,
1850,
1856,
1867,
1869,
1872,
1888,
1898,
1902,
1905,
1922,
1932,
1940,
1944,
1959,
1966,
1968,

```
 1973,
 1976,
 1980,
 1987,
 1992,
 2000,
 2018,
 2020,
 2028,
 2036,
 2042,
 2046,
 2053,
 2069,
 2073,
 2076,
 2084,
 2097,
 2110,
 2113,
 2117,
 2121,
 2129,
 2136,
 2156,
 2158,
 2196,
 2207,
 2217,
 2223,
 2234,
 2259,
 2364,
 2392,
 2402,
 2411,
 2444,
 2515,
 2524,
 2633,
 2898,
 3138,
 3228,
 4692}
```

```
set(df['GarageCars'])
```

```
{0, 1, 2, 3, 4}
```

```
Y = np.asarray(df[["SalePrice"]].values) # Dependent Variable
```

```
Y
```

```
array([[208500],
       [181500],
       [223500],
       ...,
       [266500],
       [142125],
       [147500]], dtype=int64)
```

Creating Train and Test dataset

```
# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 500)

X_train.shape
```

```
(1168, 3)
```

```
X_test.shape
```

```
(292, 3)
```

```
Y_train.shape
```

```
(1168, 1)
```

```
Y_test.shape
```

```
(292, 1)
```

Feature Scaling

```
# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_Y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
Y_train = sc_Y.fit_transform(Y_train)
Y_test = sc_Y.transform(Y_test)

X_train
```

```
array([[ 0.10235513, -0.78522929,  0.94757715],
       [ 0.60747084, -0.78522929,  0.85855721],
       [ 1.40050251,  0.52565775,  3.36595228],
       ...,
```

```
       [ 1.15804697, -0.78522929, -0.69929179],
       [-0.52651394, -0.78522929, -0.69929179],
       [ 0.49634538, -0.78522929, -0.69929179]])
```

```
X_test
```

```
array([[-1.06951333e+00,  9.45695502e-01, -1.80008792e-01],
       [-1.90611989e-01,  1.21341187e+00,  3.24437552e-01],
       [-7.20983489e-01,  9.59542900e-01, -6.99291794e-01],
       [-3.11839760e-01, -7.85229289e-01,  4.03360185e+00],
       [-1.23620152e+00,  1.23649087e+00, -1.35498820e-01],
       [-8.62415890e-01,  1.24110667e+00,  3.39274210e-01],
       [-5.31565096e-01,  1.12801958e+00,  7.25027297e-01],
       [-7.81597375e-01, -7.85229289e-01, -6.99291794e-01],
       [-4.55797739e-01,  2.08349006e+00, -6.99291794e-01],
       [-1.75458517e-01,  7.93374120e-01, -6.99291794e-01],
       [-8.22006632e-01,  7.65679324e-01, -6.99291794e-01],
       [ 4.66038441e-01, -7.85229289e-01, -1.80008792e-01],
       [-3.37095546e-01, -7.85229289e-01, -6.99291794e-01],
       [ 5.08973277e-01, -7.85229289e-01, -2.54192078e-01],
       [-1.12255048e+00,  8.97229608e-01, -2.24518764e-01],
       [-1.03415523e+00,  1.02185619e+00, -3.16422201e-02],
       [ 6.50405677e-01, -7.85229289e-01, -1.80008792e-01],
       [-5.41667411e-01,  1.56393797e-01, -6.99291794e-01],
       [ 3.87745505e-01, -7.85229289e-01, -6.99291794e-01],
       [-6.17434768e-01,  1.35880955e+00,  2.23836633e+00],
       [-1.03415523e+00,  1.20418027e+00, -2.98702049e-01],
       [-1.01900176e+00,  1.08878528e+00, -3.16422201e-02],
       [ 3.29657198e-01,  1.48343614e+00, -3.87721993e-01],
       [ 1.23128875e+00, -7.85229289e-01, -1.94845449e-01],
       [ 4.91294227e-01, -7.85229289e-01, -1.80008792e-01],
       [-1.00889945e+00,  2.67172984e-01,  1.46397666e-01],
       [ 1.01661457e+00,  1.47189664e+00,  1.28677515e-02],
       [ 2.58940998e-01, -7.85229289e-01, -6.99291794e-01],
       [-3.37095546e-01, -7.85229289e-01, -6.99291794e-01],
       [-4.66540097e-02, -7.85229289e-01, -6.99291794e-01],
       [ 6.52931256e-01, -7.85229289e-01,  1.97130650e+00],
       [-9.86169240e-01,  8.32608416e-01,  3.68947524e-01],
       [ 7.99414813e-01,  2.22427195e+00,  1.46686016e+00],
       [-4.60848896e-01,  1.42112284e+00,  1.31849358e+00],
       [-2.63853767e-01, -7.85229289e-01,  1.16724352e-01],
       [-1.74384281e+00,  3.77952170e-01, -6.99291794e-01],
       [-1.06193660e+00,  9.24924404e-01, -6.99291794e-01],
       [-5.39141832e-01,  1.01493249e+00, -6.99291794e-01],
       [-7.05830018e-01, -7.85229289e-01, -6.99291794e-01],
       [-7.53816011e-01, -7.85229289e-01, -6.99291794e-01],
       [-7.19097955e-02, -7.85229289e-01, -4.61905279e-01],
       [-3.90772740e-02, -7.85229289e-01,  4.25410659e-02],
       [ 4.91294227e-01, -7.85229289e-01, -1.80008792e-01],
       [ 9.73679735e-01, -7.85229289e-01,  1.02176044e+00],
       [ 9.73039691e-02, -7.85229289e-01, -6.99291794e-01],
```

```
[-6.25011504e-01,  7.79526722e-01, -1.65172135e-01],
[-4.25490796e-01,  1.47189664e+00, -6.99291794e-01],
[-1.03415523e+00,  1.20418027e+00, -2.98702049e-01],
[ 8.34772913e-01, -7.85229289e-01, -6.99291794e-01],
[ 1.30453053e+00,  2.88202337e+00, -1.65172135e-01],
[ 3.54912984e-01, -7.85229289e-01,  2.20580952e-01],
[-4.35593110e-01,  6.77979134e-01, -6.99291794e-01],
[ 9.30744899e-01,  2.28427734e+00,  3.48464553e+00],
[-1.21094573e+00,  4.14878566e-01, -6.99291794e-01],
[ 5.18435547e-02, -7.85229289e-01, -6.99291794e-01],
[ 5.46856956e-01, -8.36277739e-02, -6.99291794e-01],
[ 6.45354520e-01, -7.85229289e-01,  9.32740497e-01],
[ 1.68020169e-01, -7.85229289e-01,  6.80517325e-01],
[-1.63470668e-02, -7.85229289e-01, -3.13538707e-01],
[ 6.69970262e-02, -7.85229289e-01,  2.50254266e-01],
[ 1.27610912e-01, -7.85229289e-01, -6.99291794e-01],
[-5.18937203e-01,  1.56393797e-01, -6.99291794e-01],
[-5.08834889e-01, -7.85229289e-01, -6.99291794e-01],
[ 1.20603296e+00, -7.85229289e-01, -6.99291794e-01],
[-1.85560831e-01, -7.85229289e-01, -6.99291794e-01],
[-1.05435986e+00,  1.05416679e+00, -3.16422201e-02],
[-3.77504803e-01,  1.38419645e+00,  2.14934639e+00],
[-2.56277031e-01,  3.77952170e-01,  3.92974525e+00],
[ 1.35251652e+00, -7.85229289e-01,  2.65090924e-01],
[-9.60913454e-01, -3.51618798e-02, -6.99291794e-01],
[-8.04327582e-01,  2.92559881e-01, -6.99291794e-01],
[-7.48764854e-01,  5.99510544e-01,  4.25410659e-02],
[-8.22006632e-01, -7.85229289e-01, -6.99291794e-01],
[-1.63776851e+00,  5.23349853e-01, -6.99291794e-01],
[ 6.44714476e-02, -7.85229289e-01,  2.35417609e-01],
[-2.20918931e-01, -7.85229289e-01, -6.99291794e-01],
[-4.28016375e-01, -7.85229289e-01, -6.99291794e-01],
[-1.40794086e+00,  1.68883921e+00,  9.92087126e-01],
[-9.02825147e-01, -7.85229289e-01, -6.99291794e-01],
[ 1.32726074e+00,  2.72970199e+00, -6.99291794e-01],
[-3.16890917e-01, -7.85229289e-01, -4.02558650e-01],
[-1.03415523e+00, -7.85229289e-01, -6.99291794e-01],
[-2.20918931e-01,  1.14879068e+00,  4.13457496e-01],
[-6.50267289e-01,  1.12109588e+00,  1.80810327e+00],
[ 1.49647450e+00,  7.10289730e-01, -6.99291794e-01],
[ 5.99894106e-01, -7.85229289e-01, -6.99291794e-01],
[ 3.14503727e-01,  2.02117677e+00, -6.99291794e-01],
[-5.34090675e-01,  1.13032748e+00, -6.99291794e-01],
[-5.41667411e-01,  8.57995313e-01, -1.35498820e-01],
[ 4.30680341e-01, -7.85229289e-01, -2.83865392e-01],
[-5.99755718e-01, -7.85229289e-01, -6.99291794e-01],
[-1.05688544e+00,  9.57235001e-01, -6.99291794e-01],
[-9.05350725e-01,  1.23418297e+00, -6.99291794e-01],
[-7.36136961e-01, -7.85229289e-01, -3.16422201e-02],
[-6.07332454e-01,  1.46497294e+00,  1.27398361e+00],
```

```
[-6.32588239e-01, -7.85229289e-01, -6.99291794e-01],
[-3.37095546e-01, -7.85229289e-01, -6.99291794e-01],
[-1.14844631e-01,  1.29880416e+00,  2.43124287e+00],
[ 4.45833813e-01, -7.85229289e-01, -6.99291794e-01],
[ 2.23582898e-01, -7.85229289e-01, -6.99291794e-01],
[ 4.50884970e-01, -7.85229289e-01, -6.99291794e-01],
[ 4.25629184e-01, -7.85229289e-01, -6.99291794e-01],
[-2.00714303e-01, -7.85229289e-01, -6.99291794e-01],
[-3.09314182e-01,  1.03339569e+00,  1.14045370e+00],
[-1.12255048e+00,  8.69534811e-01, -3.72885335e-01],
[-6.88150968e-01,  1.29188046e+00,  1.02176044e+00],
[ 2.86722362e-01, -7.85229289e-01, -4.47068621e-01],
[-4.63374475e-01, -7.85229289e-01, -6.99291794e-01],
[-2.13982240e-02, -7.85229289e-01,  1.21463698e+00],
[-5.69448775e-01,  1.47651244e+00, -6.99291794e-01],
[-8.24532211e-01,  1.70961031e+00,  1.28677515e-02],
[-1.07267896e-01, -7.85229289e-01,  3.68947524e-01],
[-9.07876304e-01,  1.19956447e+00,  7.25027297e-01],
[ 1.31463284e+00, -7.85229289e-01,  1.46397666e-01],
[-1.37258276e+00,  7.65679324e-01, -6.99291794e-01],
[ 1.44596293e+00, -7.85229289e-01, -1.20662163e-01],
[ 6.83238199e-01,  8.30300516e-01,  7.54700611e-01],
[-6.22485925e-01,  1.49497564e+00,  1.52620678e+00],
[ 3.41645047e-02,  4.37957563e-01, -1.65172135e-01],
[ 3.37233934e-01,  1.64037332e+00, -1.05825506e-01],
[-8.42211261e-01,  1.12571168e+00,  3.09600895e-01],
[-5.16411625e-01, -7.85229289e-01, -6.99291794e-01],
[ 6.27675470e-01, -7.85229289e-01, -6.99291794e-01],
[-8.39685682e-01, -7.85229289e-01, -6.99291794e-01],
[-4.91795883e-02, -7.85229289e-01,  1.61234323e-01],
[ 1.15804697e+00, -7.85229289e-01, -6.99291794e-01],
[-7.44353740e-02, -7.85229289e-01, -6.99291794e-01],
[-4.45695425e-01,  2.18734555e+00,  1.31561009e-01],
[ 1.84752992e+00, -7.85229289e-01,  2.13450973e+00],
[ 1.03176804e+00, -7.85229289e-01,  2.23836633e+00],
[ 7.70993405e-02, -7.85229289e-01,  2.96536253e+00],
[-6.50267289e-01,  1.31495946e+00, -6.99291794e-01],
[-1.14844631e-01, -7.85229289e-01, -4.02558650e-01],
[-1.23620152e+00,  7.93374120e-01,  1.64490004e+00],
[-7.31085804e-01,  2.85636182e-01, -6.99291794e-01],
[ 1.27927474e+00, -7.85229289e-01, -4.64788773e-02],
[ 2.22131555e+00,  1.94963188e+00,  5.76660725e-01],
[ 1.65306037e+00, -7.85229289e-01,  9.03067183e-01],
[-5.44192989e-01, -7.85229289e-01,  2.34222293e+00],
[-4.98732575e-01, -1.90065817e-02,  3.39274210e-01],
[ 1.39040020e+00, -7.85229289e-01, -6.99291794e-01],
[-7.96750847e-01,  1.19494867e+00,  5.46987410e-01],
[ 1.07406283e-01, -7.85229289e-01,  1.15529035e+00],
[-9.46400027e-02, -7.85229289e-01, -6.99291794e-01],
[ 2.38736369e-01, -7.85229289e-01, -6.99291794e-01],
```

```
[-8.62415890e-01, -7.85229289e-01, -6.99291794e-01],
[-1.19895788e-01, -7.85229289e-01, -6.99291794e-01],
[-4.81053525e-01,  9.84929797e-01, -6.99291794e-01],
[-9.33132090e-01, -7.85229289e-01, -6.99291794e-01],
[-7.05830018e-01, -7.85229289e-01,  1.43718684e+00],
[-9.43234404e-01, -7.85229289e-01, -6.99291794e-01],
[-3.37095546e-01, -7.85229289e-01, -6.99291794e-01],
[-2.84058396e-01,  1.20418027e+00, -1.20662163e-01],
[-4.63374475e-01, -7.85229289e-01, -6.99291794e-01],
[ 1.28937706e+00, -7.85229289e-01,  3.68947524e-01],
[-1.03162965e+00,  5.99510544e-01, -1.65172135e-01],
[ 3.07748669e+00, -7.85229289e-01,  8.58557211e-01],
[ 7.45737619e-02,  2.26812204e+00,  4.57967467e-01],
[-9.73541347e-01, -7.85229289e-01, -6.99291794e-01],
[ 7.48903242e-01,  1.89193439e+00,  2.79927581e-01],
[ 1.23886549e+00, -7.85229289e-01, -6.99291794e-01],
[ 8.24670599e-01,  1.80423420e+00, -2.24518764e-01],
[-9.78592504e-01,  1.14851602e-01, -6.99291794e-01],
[ 2.58940998e-01,  1.07032209e+00, -6.99291794e-01],
[-1.51148958e+00,  5.07194555e-01, -6.99291794e-01],
[ 6.78187041e-01, -7.85229289e-01, -1.94845449e-01],
[-7.61392747e-01,  4.17186466e-01, -6.99291794e-01],
[-6.18074812e-02,  1.84808429e+00, -6.99291794e-01],
[-7.08355597e-01, -7.85229289e-01, -6.99291794e-01],
[ 8.06991549e-01, -7.85229289e-01, -1.20662163e-01],
[-4.40644268e-01, -7.85229289e-01,  1.25914696e+00],
[-9.15453040e-01,  5.44120950e-01, -6.99291794e-01],
[-2.10816617e-01, -7.85229289e-01,  2.77044087e-02],
[ 8.87810063e-01, -7.85229289e-01, -1.65172135e-01],
[-7.81597375e-01,  1.20879607e+00,  1.31561009e-01],
[ 1.40050251e+00, -7.85229289e-01, -6.99291794e-01],
[-2.86583974e-01, -7.85229289e-01,  9.32740497e-01],
[ 7.38800927e-01, -7.85229289e-01, -4.32231964e-01],
[ 3.59964141e-01, -7.85229289e-01, -6.99291794e-01],
[-7.81597375e-01, -7.85229289e-01, -6.99291794e-01],
[-8.24532211e-01,  1.75576830e+00,  1.28677515e-02],
[ 6.35252206e-01, -7.85229289e-01, -6.99291794e-01],
[ 2.13480584e-01, -7.85229289e-01, -6.99291794e-01],
[-7.91699690e-01,  1.75346040e+00,  1.28677515e-02],
[-6.90676547e-01, -7.85229289e-01, -2.24518764e-01],
[-7.05830018e-01, -7.85229289e-01, -6.99291794e-01],
[-8.22006632e-01, -7.85229289e-01, -6.99291794e-01],
[-1.19359535e-03, -7.85229289e-01,  1.08110707e+00],
[-4.43169846e-01,  9.78006098e-01, -6.99291794e-01],
[ 1.17508598e-01, -7.85229289e-01, -6.99291794e-01],
[ 2.33685212e-01, -7.85229289e-01, -4.61905279e-01],
[-8.06853161e-01, -7.85229289e-01,  9.62413812e-01],
[-3.34569967e-01,  8.34916315e-01, -6.99291794e-01],
[ 1.20034176e-01,  8.09529418e-01, -3.28375364e-01],
[-1.02216738e-01,  8.00297820e-01,  7.84373925e-01],
```

```
[-1.03415523e+00,  1.20418027e+00, -2.98702049e-01],
[-2.13342196e-01,  7.40292427e-01, -6.99291794e-01],
[-8.22006632e-01, -7.85229289e-01, -6.99291794e-01],
[ 2.53889841e-01, -7.85229289e-01,  7.54700611e-01],
[-7.56341589e-01,  7.14905530e-01, -6.99291794e-01],
[-1.14528069e+00, -7.85229289e-01, -6.99291794e-01],
[ 8.14568285e-01,  6.82594934e-01, -1.05825506e-01],
[ 9.00437956e-01, -7.85229289e-01, -1.65172135e-01],
[-2.20918931e-01, -7.85229289e-01, -6.99291794e-01],
[-5.49244146e-01,  1.81808160e+00, -3.16422201e-02],
[-4.73476789e-01, -7.85229289e-01,  1.19980033e+00],
[-4.28016375e-01,  1.17186967e+00,  8.73393868e-01],
[ 1.23633991e+00, -7.85229289e-01,  1.33333024e+00],
[-1.13265280e+00,  9.54927101e-01, -4.64788773e-02],
[-3.82555960e-01, -7.85229289e-01, -2.54192078e-01],
[-4.63374475e-01,  2.93279716e+00, -6.99291794e-01],
[-7.36136961e-01, -7.85229289e-01, -6.99291794e-01],
[ 9.93884363e-01, -7.85229289e-01, -6.99291794e-01],
[-1.37258276e+00, -7.85229289e-01, -6.99291794e-01],
[-4.55797739e-01,  1.32649896e+00, -6.99291794e-01],
[ 1.27674916e+00, -7.85229289e-01, -7.61521916e-02],
[ 3.65015298e-01,  5.62584148e-01, -6.99291794e-01],
[-4.35593110e-01,  1.52497833e+00, -6.99291794e-01],
[-8.19481054e-01,  1.08647739e+00, -6.99291794e-01],
[-3.06788603e-01,  8.76458510e-01, -3.43212021e-01],
[-7.31085804e-01,  1.87116329e+00,  1.28677515e-02],
[-1.00889945e+00,  1.97040298e+00,  4.13457496e-01],
[ 1.00398668e+00, -7.85229289e-01, -1.65172135e-01],
[-1.07709007e+00, -7.85229289e-01, -6.99291794e-01],
[-7.20983489e-01,  7.74910923e-01,  4.25410659e-02],
[-1.44835012e+00,  5.99510544e-01,  1.85261324e+00],
[-7.44353740e-02, -7.85229289e-01, -6.99291794e-01],
[ 7.84261342e-01, -7.85229289e-01, -2.09682106e-01],
[-4.17914060e-01, -7.85229289e-01, -6.99291794e-01],
[ 8.70131013e-01,  6.71055435e-01, -6.13155345e-02],
[-8.75043783e-01, -7.85229289e-01, -6.99291794e-01],
[-9.25555354e-01,  8.34916315e-01,  4.13457496e-01],
[-7.96750847e-01, -7.85229289e-01, -6.99291794e-01],
[ 3.68615113e+00, -7.85229289e-01,  1.37784021e+00],
[-1.29428983e+00,  5.74123647e-01, -6.99291794e-01],
[-1.24125268e+00,  4.79499758e-01, -6.99291794e-01],
[ 4.13001291e-01, -7.85229289e-01, -6.99291794e-01],
[ 1.85699219e-01, -7.85229289e-01,  5.46987410e-01],
[-4.22965217e-01,  1.61498642e+00,  2.20580952e-01],
[-7.28560225e-01,  1.13725118e+00,  6.95353982e-01],
[-2.81532817e-01, -7.85229289e-01, -6.99291794e-01],
[-8.24532211e-01,  1.78115520e+00,  1.28677515e-02],
[ 3.90271084e-01, -7.85229289e-01, -3.13538707e-01],
[-1.21852247e+00, -7.85229289e-01, -4.02558650e-01],
[-5.29039518e-01, -7.85229289e-01, -6.99291794e-01],
```

```
       [ 8.70131013e-01, -7.85229289e-01,  1.37784021e+00],
       [ 4.05424555e-01, -7.85229289e-01, -4.61905279e-01],
       [ 1.93592517e+00,  2.10887696e+00, -6.99291794e-01],
       [ 1.61012554e+00, -7.85229289e-01, -6.99291794e-01],
       [-6.60369604e-01,  2.01886887e+00,  2.14934639e+00],
       [ 7.79210185e-01,  1.73499721e+00,  1.49653347e+00],
       [-1.65039641e+00,  5.99510544e-01,  1.76359330e+00],
       [ 3.24606041e-01, -7.85229289e-01, -6.99291794e-01],
       [ 2.67844527e+00, -7.85229289e-01,  1.90907638e-01],
       [-5.44192989e-01, -7.85229289e-01, -6.99291794e-01],
       [-4.25490796e-01, -7.85229289e-01, -6.99291794e-01],
       [ 2.27182712e+00, -7.85229289e-01, -6.99291794e-01],
       [ 6.69970262e-02, -7.85229289e-01, -9.09888488e-02],
       [ 8.34772913e-01, -7.85229289e-01,  2.79927581e-01],
       [-1.65039641e+00,  6.52592237e-01,  1.76359330e+00],
       [-1.50896401e+00,  4.44881262e-01, -6.99291794e-01],
       [-5.99755718e-01, -7.85229289e-01, -6.99291794e-01],
       [-6.19960346e-01, -7.85229289e-01, -6.99291794e-01],
       [ 7.21121877e-01, -7.85229289e-01, -1.20662163e-01],
       [ 6.73135884e-01,  2.53814631e+00,  9.32740497e-01],
       [-9.46400027e-02,  9.87237697e-01, -6.99291794e-01],
       [ 2.26172481e+00,  5.39505151e-01,  5.46987410e-01],
       [ 7.96889235e-01, -7.85229289e-01,  1.37784021e+00],
       [ 1.57917855e-01, -7.85229289e-01,  1.64490004e+00],
       [-1.58473136e+00,  4.74883959e-01, -6.99291794e-01],
       [-9.53336718e-01, -7.85229289e-01, -6.99291794e-01],
       [ 5.51908113e-01, -7.85229289e-01, -6.99291794e-01],
       [-9.02825147e-01, -7.85229289e-01, -2.24518764e-01],
       [ 1.24896780e+00, -7.85229289e-01, -3.16422201e-02],
       [-2.15867774e-01, -7.85229289e-01,  3.98620838e-01],
       [ 1.27610912e-01, -7.85229289e-01,  3.81105199e+00],
       [ 2.08429426e-01, -7.85229289e-01,  3.09600895e-01],
       [ 1.02355126e-01, -7.85229289e-01, -6.99291794e-01],
       [-2.56277031e-01, -7.85229289e-01, -2.83865392e-01],
       [-1.20589458e+00,  8.76458510e-01, -4.64788773e-02],
       [-8.80094940e-01,  5.69507847e-01, -6.99291794e-01],
       [-8.62415890e-01,  1.01004204e-01, -6.99291794e-01],
       [-3.57300175e-01, -2.77491351e-01, -6.99291794e-01],
       [-1.19074110e+00,  1.01262459e+00, -1.65172135e-01],
       [-2.00714303e-01,  9.71082399e-01,  7.84373925e-01],
       [-1.65356203e-01,  1.49728354e+00,  1.82293993e+00],
       [-7.44353740e-02, -7.85229289e-01, -6.99291794e-01],
       [-8.22006632e-01, -7.85229289e-01, -6.99291794e-01],
       [-6.45216132e-01, -7.85229289e-01, -6.99291794e-01],
       [-4.12862903e-01,  2.11580066e+00, -1.68055629e-02],
       [ 1.12016329e+00, -7.85229289e-01,  3.24437552e-01],
       [-7.36136961e-01,  1.31034366e+00,  6.06334039e-01]])

Y_train
```

```
array([[-0.47654967],
       [ 0.04412298],
       [ 3.30843421],
       ...,
       [-0.52555415],
       [-0.75955057],
       [-0.36628958]])
```

```
Y_test
```

```
array([[-9.79900262e-02],
       [ 2.83019842e-01],
       [-4.33670746e-01],
       [-5.56181958e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 6.07674553e-01],
       [-7.52199896e-01],
       [ 4.91288902e-01],
       [-2.49903929e-01],
       [-2.75214745e-01],
       [ 5.15791144e-01],
       [-5.90485097e-01],
       [-5.13303034e-01],
       [-7.22626718e-02],
       [ 4.07981278e-01],
       [ 1.23755267e-01],
       [-6.66442048e-01],
       [-3.81603481e-01],
       [-9.54343396e-01],
       [-3.41787337e-01],
       [ 4.90234277e-02],
       [ 5.28042265e-01],
       [ 5.26817153e-01],
       [ 2.52392039e-01],
       [-7.38723663e-01],
       [-2.74406171e-01],
       [-1.49339273e+00],
       [-5.99060882e-01],
       [-4.15294064e-01],
       [ 1.48257509e-01],
       [-3.67344204e-02],
       [ 2.70874183e+00],
       [ 8.94350788e-01],
       [-5.13303034e-01],
       [-1.00334788e+00],
       [-5.38859900e-02],
       [ 4.60661099e-01],
       [-4.82675231e-01],
       [-9.42092275e-01],
       [-2.65830386e-01],
```

```
[-2.92782853e-01],
[ 2.54413474e-01],
[-3.47912898e-01],
[-4.77604295e-02],
[-1.10070660e-02],
[ 5.28042265e-01],
[-7.83882324e-02],
[-1.10070660e-02],
[ 4.36264319e+00],
[ 3.13647645e-01],
[-7.95078821e-01],
[ 1.10384496e+00],
[-9.42092275e-01],
[-2.74406171e-01],
[-5.56181958e-01],
[ 1.89430863e-05],
[ 1.48257509e-01],
[-1.09016035e-01],
[-4.39796307e-01],
[-3.49909831e-01],
[-9.42092275e-01],
[-6.11312003e-01],
[ 2.57462975e-02],
[-5.01051913e-01],
[ 5.63741004e-02],
[-6.35814245e-01],
[ 5.09665584e-01],
[ 1.80093375e+00],
[-7.59550569e-01],
[-4.27545186e-01],
[-4.52047428e-01],
[-4.58172988e-01],
[-1.15036133e+00],
[-3.43012450e-01],
[-4.53272540e-01],
[-4.03042943e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 1.69189878e+00],
[-5.74558640e-01],
[-1.33412815e+00],
[ 1.20185393e+00],
[-1.70271641e-01],
[-2.68280610e-01],
[-1.21774250e+00],
[ 8.34320294e-01],
[ 5.63741004e-02],
[-1.09016035e-01],
[-4.64298549e-01],
[-6.48065367e-01],
```

```
[-9.05338911e-01],
[-4.64298549e-01],
[-1.04010124e+00],
[ 3.44275448e-01],
[-4.76549670e-01],
[-3.84666262e-01],
[ 6.75055719e-01],
[-1.64146081e-01],
[-4.88150538e-03],
[-6.35814245e-01],
[-2.49903929e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.45137930e-02],
[ 7.10583971e-01],
[-2.43778368e-01],
[-7.14221421e-01],
[-8.57389051e-02],
[-2.68280610e-01],
[ 4.60661099e-01],
[-1.13233994e+00],
[ 4.17782175e-01],
[ 1.53263420e+00],
[-7.64451018e-01],
[ 2.39633824e+00],
[ 1.16510057e+00],
[ 4.66786659e-01],
[ 6.86252216e-02],
[ 7.11809083e-01],
[-2.32581871e-02],
[-7.33823215e-01],
[-5.14528146e-01],
[-1.22386806e+00],
[-3.41787337e-01],
[-4.76549670e-01],
[-5.13915590e-01],
[ 8.09818052e-01],
[ 1.69189878e+00],
[ 9.92530245e-02],
[-2.43778368e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.25657619e-01],
[-7.95078821e-01],
[ 4.91288902e-01],
[ 2.79449968e+00],
[ 1.62835221e+00],
[-6.60316488e-01],
[-2.68280610e-01],
[ 1.96036882e-01],
```

```
[-1.33518278e-01],
[-2.19276126e-01],
[-5.13303034e-01],
[-4.70424110e-01],
[-9.35966714e-01],
[-8.45137930e-02],
[-2.68280610e-01],
[-7.70576578e-01],
[-5.31679715e-01],
[-1.49339273e+00],
[-1.08298017e+00],
[ 9.92530245e-02],
[-4.52047428e-01],
[ 1.03033823e+00],
[-7.27697654e-01],
[ 2.50353555e+00],
[ 1.32436514e+00],
[-1.00334788e+00],
[ 1.63064317e+00],
[-1.00334788e+00],
[ 1.75315438e+00],
[-5.75783752e-01],
[ 6.38302356e-01],
[-3.29536216e-01],
[-1.71326265e-02],
[-5.68433079e-01],
[ 9.06601909e-01],
[-5.74558640e-01],
[ 3.30799215e-01],
[-6.05186442e-01],
[-1.08910573e+00],
[-3.05033974e-01],
[ 6.50553477e-01],
[ 3.13647645e-01],
[-7.70576578e-01],
[-6.37039358e-01],
[ 2.34015357e-01],
[-1.45891910e-01],
[-6.66442048e-01],
[ 1.60508630e-01],
[-1.71326265e-02],
[-8.93590086e-01],
[ 4.23907735e-01],
[-5.77008864e-01],
[-6.35814245e-01],
[-5.13303034e-01],
[-1.45769399e-01],
[-3.05033974e-01],
[-9.78845638e-01],
[-2.69566978e-01],
```

```
[-6.11312003e-01],
[-8.45137930e-02],
[-2.31527247e-01],
[ 2.57462975e-02],
[-1.09016035e-01],
[-7.09320972e-01],
[-5.01051913e-01],
[ 2.58517600e-01],
[-4.64298549e-01],
[-1.34025371e+00],
[ 1.65024496e+00],
[ 9.44580385e-01],
[-2.68893166e-01],
[ 6.26051235e-01],
[-9.78845638e-01],
[ 5.89297871e-01],
[ 9.07827021e-01],
[-6.29688685e-01],
[-1.18711470e+00],
[ 8.70019034e-02],
[-9.64144293e-01],
[ 1.01270887e+00],
[-1.23611918e+00],
[ 2.57292488e-01],
[ 1.42131949e-01],
[-8.56334426e-01],
[-4.52047428e-01],
[ 8.33265670e-02],
[-2.93837477e-02],
[ 4.97414462e-01],
[ 1.07321716e+00],
[ 6.99557962e-01],
[-1.24959542e+00],
[-1.33518278e-01],
[-3.78540701e-01],
[-6.48065367e-01],
[ 1.02911312e+00],
[-5.50056397e-01],
[ 7.07962231e-01],
[-9.11464472e-01],
[-2.32581871e-02],
[-8.01204381e-01],
[ 3.48668803e+00],
[-5.25554155e-01],
[-8.80836669e-01],
[-2.32581871e-02],
[ 1.85010873e-01],
[ 1.25085841e+00],
[ 3.79974187e-02],
[-4.29995410e-01],
```

```
[ 4.05531054e-01],
[-4.76549670e-01],
[-1.17486358e+00],
[-2.25401686e-01],
[ 1.12834720e+00],
[-1.71496753e-01],
[ 2.17091761e+00],
[ 5.89297871e-01],
[ 6.87306840e-01],
[ 1.15284944e+00],
[-3.78540701e-01],
[-2.92782853e-01],
[ 3.91960590e+00],
[-3.72415140e-01],
[-8.19581063e-01],
[ 3.44275448e-01],
[ 3.67723066e-02],
[ 2.58517600e-01],
[-4.22644737e-01],
[-5.25554155e-01],
[-5.13303034e-01],
[-6.11312003e-01],
[ 1.23755267e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 2.42696605e+00],
[ 6.21150786e-01],
[-3.94467158e-01],
[-1.15036133e+00],
[-1.55464833e+00],
[-7.39948775e-01],
[-7.76702139e-01],
[ 1.26310954e+00],
[-1.45769399e-01],
[-2.07025005e-01],
[-4.21419625e-01],
[-5.13303034e-01],
[-4.15294064e-01],
[-3.29536216e-01],
[-6.05186442e-01],
[-7.76702139e-01],
[-7.70576578e-01],
[-9.67649141e-02],
[ 6.24996610e-02],
[ 1.13202254e+00],
[-4.76549670e-01],
[-3.99367607e-01],
[-1.00947344e+00],
[ 2.21764236e-01],
```

```
       [ 1.36006388e-01],
       [ 1.36006388e-01]])
```

Multiple Regression Model

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)#training func question + answers
# The coefficients
print ('Intercept: ',regr.intercept_)
print ('Coefficient : ',regr.coef_)
```

```
Intercept:  [-1.22690901e-16]
Coefficient :  [[0.67069037 0.42959836 0.07104694]]
```

```
regr.intercept_
```

```
array([-1.22690901e-16])
```

```
regr.coef_
```

```
array([[0.67069037, 0.42959836, 0.07104694]])
```

```
regr.coef_[0][0]
```

```
0.6706903674798578
```

```
regr.coef_[0][1]
```

```
0.4295983638014665
```

```
regr.coef_[0][2]
```

```
0.07104693987752371
```

```
y_pred = regr.predict(X_test)
```

```
y_pred
```

```
array([[-0.32383212],
       [ 0.41648842],
       [-0.12102116],
       [-0.25990607],
       [-0.30754077],
       [-0.02113224],
       [ 0.17959075],
       [-0.91122559],
       [ 0.53968223],
       [ 0.17347134],
       [-0.27205989],
       [-0.0375548 ],
       [-0.6131025 ],
       [-0.01402931],
```

```
[-0.3833868 ],
[-0.25685829],
[ 0.08609853],
[-0.34578714],
[-0.12695858],
[ 0.32866389],
[-0.19750595],
[-0.21794237],
[ 0.83083318],
[ 0.47463711],
[-0.02061599],
[-0.55148096],
[ 1.3150722 ],
[-0.21334653],
[-0.6131025 ],
[-0.41830615],
[ 0.24063678],
[-0.2775144 ],
[ 1.59591933],
[ 0.39510007],
[-0.50600449],
[-1.05689348],
[-0.36456718],
[ 0.02473356],
[-0.86040915],
[-0.8925929 ],
[-0.41837938],
[-0.36051956],
[-0.02061599],
[ 0.38829735],
[-0.32175493],
[-0.09604077],
[ 0.29726927],
[-0.19750595],
[ 0.17285839],
[ 2.10131361],
[-0.0836249 ],
[-0.05057192],
[ 1.85313685],
[-0.68362103],
[-0.35224479],
[ 0.2811628 ],
[ 0.1617682 ],
[-0.17629504],
[-0.370573  ],
[-0.27461916],
[-0.30142835],
[-0.33054221],
[-0.72828642],
[ 0.42185893],
```

```
[-0.51146962],
[-0.25652876],
[ 0.49416418],
[ 0.26968147],
[ 0.58862048],
[-0.70926343],
[-0.46345406],
[-0.24161821],
[-0.93832769],
[-0.92328787],
[-0.27736714],
[-0.53518396],
[-0.67408222],
[-0.14828506],
[-0.99253189],
[ 2.01317396],
[-0.57846946],
[-1.08061371],
[ 0.37472529],
[ 0.17395315],
[ 1.2591278 ],
[ 0.01532744],
[ 1.02954631],
[ 0.07769482],
[-0.00432451],
[-0.06864783],
[-0.78926614],
[-0.34729884],
[-0.12668957],
[-0.83330127],
[ 0.31253059],
[-0.8112866 ],
[-0.6131025 ],
[ 0.65367132],
[-0.08799932],
[-0.23706086],
[-0.08461155],
[-0.10155037],
[-0.52163291],
[ 0.3175168 ],
[-0.40582543],
[ 0.16604646],
[-0.17679415],
[-0.69779656],
[-0.26538856],
[ 0.20270098],
[ 0.18235419],
[-0.38306417],
[-0.04206199],
[ 0.55477947],
```

```
[-0.64132599],
[ 0.62388751],
[ 0.86855619],
[ 0.33317609],
[ 0.19932468],
[ 0.92336267],
[-0.05926289],
[-0.73336806],
[ 0.03396013],
[-0.95018486],
[-0.35886229],
[ 0.38967519],
[-0.43693885],
[ 0.65010345],
[ 1.05343769],
[ 0.51369275],
[-0.0749435 ],
[ 0.07909388],
[-0.44295897],
[-0.37141111],
[-0.41730591],
[ 0.51736185],
[ 2.36834359],
[ 0.83551861],
[-0.53591044],
[-0.31855594],
[ 0.54551226],
[ 0.01783666],
[-0.18321701],
[-0.4504899 ],
[-0.22689758],
[-0.96542979],
[-0.46742871],
[ 0.05080372],
[-1.01285846],
[-0.70861888],
[-1.01963399],
[-0.6131025 ],
[ 0.31822597],
[-0.69779656],
[ 0.55365215],
[-0.4460903 ],
[ 1.78770532],
[ 1.05693461],
[-1.03996056],
[ 1.33494211],
[ 0.44387939],
[ 1.31224332],
[-0.65667505],
[ 0.58379531],
```

```
[-0.8455341 ],
[ 0.10367713],
[-0.3811187 ],
[ 0.70279776],
[-0.86210304],
[ 0.19533556],
[-0.54341055],
[-0.42991461],
[-0.47675758],
[ 0.24637746],
[ 0.00443399],
[ 0.55228779],
[-0.46327397],
[ 0.12746469],
[-0.14559128],
[-0.91122559],
[ 0.20218359],
[ 0.03904178],
[-0.24383639],
[ 0.22321258],
[-0.8165147 ],
[-0.86040915],
[-0.93832769],
[-0.2613244 ],
[ 0.07323753],
[-0.30820388],
[-0.21341975],
[-0.8101053 ],
[ 0.08460329],
[ 0.40494821],
[ 0.33097822],
[-0.19750595],
[ 0.12525932],
[-0.93832769],
[-0.11343258],
[-0.24983131],
[-1.15514449],
[ 0.83204619],
[ 0.25484687],
[-0.53518396],
[ 0.41042404],
[-0.5696474 ],
[ 0.2784188 ],
[ 0.58659708],
[-0.35272638],
[-0.61196938],
[ 0.89946152],
[-0.88073573],
[ 0.27957291],
[-1.3075938 ],
```

```
[ 0.21448009],
[ 0.51355977],
[ 0.43681493],
[ 0.31329755],
[-0.13255168],
[ 0.14638082],
[ 0.3144307 ],
[ 0.19919764],
[ 0.324296  ],
[-1.10940969],
[-0.1476338 ],
[-0.58222322],
[-0.43693885],
[ 0.17376604],
[-0.66730669],
[ 0.86751652],
[-0.9738992 ],
[-0.23270749],
[-0.92138888],
[ 2.23282417],
[-0.67110768],
[-0.67618644],
[-0.11001977],
[-0.17392476],
[ 0.42578843],
[ 0.04932569],
[-0.57583711],
[ 0.21308976],
[-0.09785813],
[-1.18318506],
[-0.74183747],
[ 0.3441466 ],
[-0.09823583],
[ 2.15469392],
[ 0.69287993],
[ 0.57710372],
[ 1.37428485],
[-0.72405832],
[-0.16930562],
[ 1.47263763],
[-0.75200076],
[-0.67238834],
[ 1.13667681],
[-0.29886344],
[ 0.24242893],
[-0.70125451],
[-0.8706099 ],
[-0.78926614],
[-0.80281719],
[ 0.1377436 ],
```

```
       [ 1.60811761],
       [ 0.31095902],
       [ 1.78754935],
       [ 0.29502405],
       [-0.11455412],
       [-0.90853723],
       [-1.02640951],
       [-0.0168563 ],
       [-0.95880072],
       [ 0.49808937],
       [-0.45379286],
       [ 0.01901777],
       [-0.17554541],
       [-0.31836716],
       [-0.52938352],
       [-0.43555892],
       [-0.3952941 ],
       [-0.58470533],
       [-0.40853016],
       [-0.37533169],
       [ 0.33828563],
       [ 0.66184205],
       [-0.43693885],
       [-0.93832769],
       [-0.819756  ],
       [ 0.63084735],
       [ 0.43699981],
       [ 0.1122797 ]])
```

```
from sklearn.metrics import r2_score
```

```
print(f"R2 Score : {r2_score(Y_test,y_pred)*100} % ")
```

```
R2 Score : 65.71391638689818 %
```

```
print(f"Mean absolute error: {np.mean(np.absolute(y_pred - Y_test))}
")#pred - actual
```

```
Mean absolute error: 0.3503742187252111
```

```
print("Residual sum of squares (MSE): %.2f" % np.mean((y_pred -
Y_test) **2))
```

```
Residual sum of squares (MSE): 0.25
```

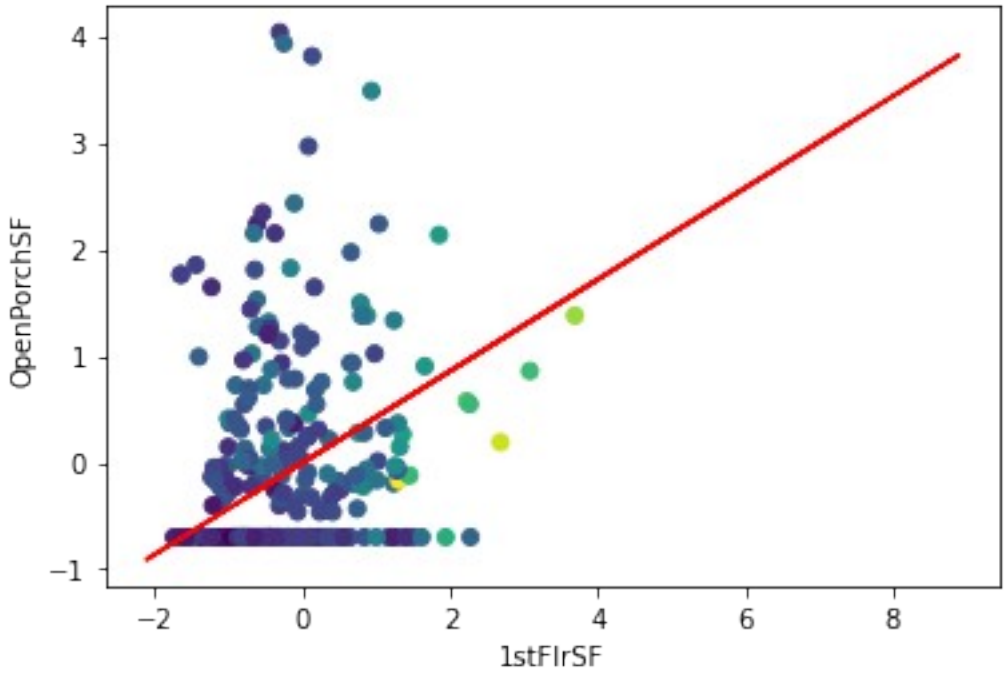
```
accuracy = print(f"R2 Score : {r2_score(Y_test,y_pred)*100} % ")
```

```
R2 Score : 65.71391638689818 %
```

Plot Outputs

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,0],X_test[:,2],c=Y_test)
plt.plot(X_train, regr.coef_[0][1]*X_train + regr.intercept_[0], 'r')
# y = mx+c
plt.xlabel("1stFlrSF")
plt.ylabel("OpenPorchSF")
```

```
Text(0, 0.5, 'OpenPorchSF')
```




```
from sklearn.linear_model import Lasso

L = Lasso(alpha = 1)

L.fit(X_train,Y_train)

Lasso(alpha=1)

ypred2 = L.predict(X_test)

from sklearn.metrics import r2_score
print("R2-score",r2_score(Y_test,ypred2))
print(f"Mean absolute error : {np.mean(np.absolute(ypred2 - Y_test))}
")# pred - actual
```

```
R2-score -0.004908518792945626
Mean absolute error: 0.6243935831081134
```

2nd Model

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/housing.csv")

df
```

```
        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
...    ...         ...      ...          ...      ...    ...   ...
...
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
3            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
...          ...       ...  ...      ...    ...    ...         ...
...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
```

```
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
...     ...    ...       ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[1460 rows x 81 columns]
```

```
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.tail()

        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley
LotShape  \
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[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
```

```
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
```

```
cdf1 = df[["MSSubClass","MoSold","LotArea","SalePrice"]]
```

```
cdf1
```

```
      MSSubClass  MoSold  LotArea  SalePrice
0             60       2     8450     208500
1             20       5     9600     181500
2             60       9    11250     223500
3             70       2     9550     140000
4             60      12    14260     250000
...          ...     ...      ...        ...
1455          60       8     7917     175000
1456          20       2    13175     210000
1457          70       5     9042     266500
1458          20       4     9717     142125
1459          20       6     9937     147500

[1460 rows x 4 columns]
```

```
cdf1.head()
```

```
   MSSubClass  MoSold  LotArea  SalePrice
0          60       2     8450     208500
1          20       5     9600     181500
2          60       9    11250     223500
3          70       2     9550     140000
4          60      12    14260     250000
```

```
cdf1.tail()
```

```
      MSSubClass  MoSold  LotArea  SalePrice
1455          60       8     7917     175000
1456          20       2    13175     210000
1457          70       5     9042     266500
1458          20       4     9717     142125
1459          20       6     9937     147500
```

```
cdf1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   MSSubClass  1460 non-null   int64
 1   MoSold      1460 non-null   int64
 2   LotArea     1460 non-null   int64
 3   SalePrice   1460 non-null   int64
```

```
dtypes: int64(4)
memory usage: 45.8 KB
```

```
cdf1.describe()
```

```
        MSSubClass       MoSold        LotArea      SalePrice
count  1460.000000  1460.000000    1460.000000    1460.000000
mean     56.897260     6.321918   10516.828082  180921.195890
std      42.300571     2.703626    9981.264932   79442.502883
min      20.000000     1.000000    1300.000000   34900.000000
25%      20.000000     5.000000    7553.500000  129975.000000
50%      50.000000     6.000000    9478.500000  163000.000000
75%      70.000000     8.000000   11601.500000  214000.000000
max     190.000000    12.000000  215245.000000  755000.000000
```

```
cdf1.isnull().sum()
```

```
MSSubClass    0
MoSold        0
LotArea       0
SalePrice     0
dtype: int64
```

```
X = df[["MSSubClass","MoSold","LotArea"]]
```

```
X
```

```
      MSSubClass  MoSold  LotArea
0             60       2     8450
1             20       5     9600
2             60       9    11250
3             70       2     9550
4             60      12    14260
...          ...     ...      ...
1455          60       8     7917
1456          20       2    13175
1457          70       5     9042
1458          20       4     9717
1459          20       6     9937

[1460 rows x 3 columns]
```

```
Y = df[["SalePrice"]]
```

```
Y
```

```
      SalePrice
0        208500
1        181500
2        223500
3        140000
4        250000
```

```
...         ...
1455     175000
1456     210000
1457     266500
1458     142125
1459     147500

[1460 rows x 1 columns]
```

Creating Train and Test Dataset

```
# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 500)

X_train.shape
```

```
(1168, 3)
```

```
X_test.shape
```

```
(292, 3)
```

```
Y_train.shape
```

```
(1168, 1)
```

```
Y_test.shape
```

```
(292, 1)
```

Feature Scaling

```
# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_Y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
Y_train = sc_Y.fit_transform(Y_train)
Y_test = sc_Y.transform(Y_test)

X_train
```

```
array([[-0.85469057,  0.24559339, -0.0242262 ],
       [-0.85469057, -0.49181812, -0.18471992],
       [ 0.09911675, -0.12311237,  0.29260181],
       ...,
```

```
       [-0.85469057, -0.86052387, -0.10082087],
       [-0.85469057,  0.24559339, -0.28343067],
       [-0.85469057, -0.12311237, -0.01966095]])
```

```
X_test
```

```
array([[ 9.91167493e-02,  6.14299139e-01, -1.77314083e-01],
       [ 9.91167493e-02,  9.83004891e-01, -1.15449627e-02],
       [ 3.37568579e-01, -1.23112366e-01, -1.12994853e-01],
       [-8.54690568e-01, -1.96664113e+00,  6.61456757e-03],
       [ 9.91167493e-02, -1.22922962e+00,  1.33629830e-01],
       [ 9.91167493e-02, -1.23112366e-01, -1.32168882e-01],
       [ 9.91167493e-02, -4.91818118e-01,  1.86789572e-01],
       [-8.54690568e-01,  1.72041640e+00, -3.68648575e-01],
       [ 9.91167493e-02,  1.72041640e+00,  2.44717459e-01],
       [-1.39335080e-01,  1.72041640e+00,  6.65714526e-02],
       [-1.39335080e-01,  2.45593386e-01, -6.18215305e-01],
       [ 1.52982773e+00, -1.23112366e-01, -4.28402561e-01],
       [-8.54690568e-01,  2.45593386e-01, -3.07778642e-01],
       [-8.54690568e-01, -1.23112366e-01,  7.26584460e-02],
       [ 2.48363504e+00, -1.23112366e-01, -7.47056665e-01],
       [ 9.91167493e-02, -1.23112366e-01, -1.98212760e-01],
       [-8.54690568e-01,  9.83004891e-01, -1.59966152e-01],
       [-1.39335080e-01,  1.72041640e+00, -4.41692496e-01],
       [-8.54690568e-01,  2.45593386e-01, -3.25228023e-01],
       [ 4.56794493e-01,  2.45593386e-01, -5.14736417e-01],
       [ 2.48363504e+00,  1.35171064e+00, -8.26390479e-01],
       [ 9.91167493e-02,  9.83004891e-01, -2.30676725e-01],
       [-1.39335080e-01,  2.45593386e-01,  5.07168325e-01],
       [-8.54690568e-01, -4.91818118e-01,  1.56557505e-01],
       [ 1.52982773e+00,  9.83004891e-01, -3.69155825e-01],
       [-1.39335080e-01,  6.14299139e-01, -5.96707928e-01],
       [ 3.37568579e-01,  2.45593386e-01,  2.09190021e-02],
       [-8.54690568e-01, -1.23112366e-01, -1.03762913e-01],
       [-8.54690568e-01, -1.23112366e-01, -3.13865635e-01],
       [ 6.95246323e-01, -1.23112366e-01,  1.20542794e-01],
       [ 5.76020408e-01, -1.23112366e-01,  5.64386063e-01],
       [ 9.91167493e-02,  2.45593386e-01, -3.61972860e-02],
       [ 9.91167493e-02,  2.45593386e-01,  4.61008625e-01],
       [ 9.91167493e-02, -4.91818118e-01,  1.98659209e-01],
       [ 1.52982773e+00,  1.35171064e+00, -6.64577905e-01],
       [ 2.48363504e+00,  1.72041640e+00, -9.04303995e-01],
       [ 3.37568579e-01,  1.72041640e+00, -3.13865635e-01],
       [ 3.37568579e-01,  2.45593386e-01,  2.56789996e-01],
       [-8.54690568e-01, -1.23112366e-01,  2.79616222e-01],
       [-6.16238739e-01,  1.72041640e+00, -4.53866483e-01],
       [ 1.52982773e+00,  9.83004891e-01, -7.51926260e-01],
       [-8.54690568e-01, -4.91818118e-01,  2.51413152e-01],
       [ 1.52982773e+00, -1.22922962e+00, -3.85692157e-01],
       [-6.16238739e-01, -1.23112366e-01, -2.52995701e-01],
       [-8.54690568e-01, -1.23112366e-01, -1.51545811e-01],
```

```
[ 9.91167493e-02,  2.45593386e-01,  2.48755478e-02],
[ 9.91167493e-02, -4.91818118e-01,  1.70639062e-02],
[ 2.48363504e+00, -1.23112366e-01, -8.08129499e-01],
[ 1.52982773e+00, -4.91818118e-01, -7.51926260e-01],
[ 9.91167493e-02, -1.23112366e-01,  4.35323510e+00],
[ 1.52982773e+00, -8.60523870e-01, -5.61099017e-01],
[-1.39335080e-01, -1.23112366e-01, -5.36040894e-01],
[ 9.91167493e-02,  2.45593386e-01,  1.65687995e-01],
[-1.39335080e-01, -1.96664113e+00, -1.86464550e-02],
[-8.54690568e-01,  2.45593386e-01, -1.70821290e-01],
[-1.39335080e-01, -1.23112366e-01, -2.22560734e-01],
[-8.54690568e-01, -1.23112366e-01,  4.04906836e-01],
[-8.54690568e-01, -4.91818118e-01, -2.06734551e-01],
[-8.54690568e-01,  2.45593386e-01, -1.36531227e-01],
[-8.54690568e-01,  1.35171064e+00, -1.38357325e-01],
[ 8.14472237e-01, -1.23112366e-01, -3.62764482e-01],
[-1.39335080e-01, -8.60523870e-01, -1.11777454e-01],
[-8.54690568e-01, -4.91818118e-01, -2.63140690e-01],
[-8.54690568e-01,  1.72041640e+00,  1.84862024e-01],
[-8.54690568e-01, -8.60523870e-01,  1.92673666e-01],
[ 9.91167493e-02, -1.22922962e+00, -2.50459454e-01],
[ 3.37568579e-01,  1.72041640e+00, -2.09879498e-01],
[ 3.37568579e-01, -4.91818118e-01,  1.09466464e+00],
[ 1.52982773e+00, -1.96664113e+00, -5.55417823e-01],
[ 3.19899053e+00, -4.91818118e-01, -2.83430668e-01],
[ 2.48363504e+00, -1.23112366e-01, -8.40593464e-01],
[-1.39335080e-01,  6.14299139e-01, -4.36417102e-01],
[ 1.52982773e+00, -1.23112366e-01, -6.24505198e-01],
[ 2.48363504e+00, -1.22922962e+00, -9.04303995e-01],
[ 1.52982773e+00, -1.22922962e+00, -2.74097278e-01],
[-8.54690568e-01, -4.91818118e-01, -3.99509319e-02],
[-8.54690568e-01, -1.22922962e+00,  6.56584035e-02],
[ 9.91167493e-02, -4.91818118e-01,  2.23615882e-01],
[-8.54690568e-01, -1.23112366e-01, -4.05170536e-01],
[ 9.91167493e-02, -4.91818118e-01,  1.80804029e-01],
[-8.54690568e-01, -1.23112366e-01, -4.48205267e-02],
[-6.16238739e-01, -1.23112366e-01, -5.43953985e-01],
[ 9.91167493e-02, -1.23112366e-01,  1.43774819e-01],
[ 3.37568579e-01,  1.72041640e+00,  7.17981196e-01],
[ 3.19899053e+00, -4.91818118e-01,  2.09190021e-02],
[-8.54690568e-01, -8.60523870e-01,  1.32513881e-01],
[ 9.91167493e-02, -8.60523870e-01,  5.84554614e-02],
[ 9.91167493e-02, -4.91818118e-01, -3.37624886e-02],
[ 9.91167493e-02, -1.59793537e+00,  2.61943963e-02],
[-8.54690568e-01, -1.23112366e-01, -1.10661505e-01],
[-8.54690568e-01, -4.91818118e-01, -4.79431855e-01],
[ 3.37568579e-01,  1.35171064e+00, -1.53169009e-01],
[ 4.56794493e-01,  2.45593386e-01, -1.76908283e-01],
[-8.54690568e-01,  2.08912215e+00,  2.13876693e-01],
[ 9.91167493e-02,  2.45593386e-01, -1.87053272e-01],
```

```
[ 1.52982773e+00, -4.91818118e-01, -2.61923291e-01],
[-8.54690568e-01,  9.83004891e-01, -1.81980778e-01],
[ 9.91167493e-02, -1.22922962e+00, -1.13705002e-01],
[ 5.76020408e-01, -1.23112366e-01, -1.59661802e-01],
[-8.54690568e-01, -4.91818118e-01, -1.00820866e-01],
[ 5.76020408e-01,  2.45593386e-01, -3.71692072e-01],
[-8.54690568e-01,  9.83004891e-01, -8.15453868e-02],
[-8.54690568e-01, -1.23112366e-01, -1.43855597e-02],
[ 9.91167493e-02, -1.23112366e-01, -2.93575657e-01],
[ 9.91167493e-02,  6.14299139e-01,  1.14417218e+00],
[ 9.91167493e-02, -1.23112366e-01, -1.32696109e-02],
[-8.54690568e-01, -1.22922962e+00, -7.74873912e-02],
[-8.54690568e-01,  6.14299139e-01, -2.56039198e-01],
[ 5.76020408e-01, -4.91818118e-01, -1.59661802e-01],
[ 3.19899053e+00,  2.45593386e-01, -3.56271689e-01],
[ 9.91167493e-02,  2.45593386e-01, -2.42850712e-01],
[-6.16238739e-01, -1.22922962e+00, -4.73649212e-01],
[ 9.91167493e-02, -1.96664113e+00, -3.13865635e-01],
[-8.54690568e-01,  6.14299139e-01, -5.17191192e-02],
[ 2.48363504e+00, -1.59793537e+00, -9.04303995e-01],
[-8.54690568e-01,  1.35171064e+00,  4.36051952e-01],
[-1.39335080e-01, -1.23112366e-01,  1.50557927e+01],
[ 9.91167493e-02, -1.59793537e+00, -5.60814644e-02],
[-1.39335080e-01, -8.60523870e-01,  3.21007777e-01],
[-1.39335080e-01,  9.83004891e-01,  3.55137517e+00],
[ 9.91167493e-02,  9.83004891e-01,  7.33685952e-02],
[-8.54690568e-01, -1.96664113e+00, -2.41247491e-02],
[-8.54690568e-01,  6.14299139e-01,  1.77354732e-01],
[-8.54690568e-01,  9.83004891e-01, -2.22560734e-01],
[-8.54690568e-01,  1.72041640e+00,  4.12089801e-02],
[ 8.14472237e-01, -1.23112366e-01,  1.46150744e+00],
[-8.54690568e-01,  9.83004891e-01, -2.75821926e-01],
[ 9.91167493e-02, -1.23112366e-01,  4.12089801e-02],
[-8.54690568e-01, -1.22922962e+00,  1.09180406e-01],
[-8.54690568e-01, -1.23112366e-01,  2.62818117e+00],
[ 1.52982773e+00, -1.23112366e-01, -6.18215305e-01],
[ 3.37568579e-01, -1.23112366e-01, -8.21115085e-01],
[-8.54690568e-01,  2.08912215e+00, -1.92125767e-01],
[ 3.37568579e-01,  1.35171064e+00, -7.09520206e-01],
[-1.39335080e-01, -4.91818118e-01, -4.66040470e-01],
[-8.54690568e-01,  2.08912215e+00,  1.12621555e+00],
[-1.39335080e-01, -1.22922962e+00,  4.18602571e-01],
[-8.54690568e-01,  2.45593386e-01,  1.11615204e-01],
[ 1.52982773e+00, -1.59793537e+00, -5.74794752e-01],
[-1.39335080e-01, -1.23112366e-01, -1.00820866e-01],
[ 5.76020408e-01, -1.23112366e-01,  7.38271174e-01],
[ 9.91167493e-02, -1.22922962e+00, -1.36632677e-01],
[ 1.52982773e+00, -4.91818118e-01, -6.18215305e-01],
[-8.54690568e-01,  2.45593386e-01, -1.41400822e-01],
[-8.54690568e-01,  1.35171064e+00, -1.32676131e-01],
```

```
[-2.58560995e-01, -1.23112366e-01, -3.41561455e-01],
[-8.54690568e-01, -1.22922962e+00,  5.08994423e-01],
[-1.39335080e-01,  2.45593386e-01,  1.94195414e-01],
[-6.16238739e-01, -1.59793537e+00, -3.44300602e-01],
[-8.54690568e-01, -8.60523870e-01, -2.17488239e-01],
[-6.16238739e-01, -1.96664113e+00, -4.53866483e-01],
[ 8.14472237e-01,  2.45593386e-01, -2.07951950e-01],
[ 9.91167493e-02, -1.23112366e-01, -4.13712304e-02],
[-8.54690568e-01, -1.59793537e+00, -1.36835577e-01],
[ 1.52982773e+00,  2.45593386e-01, -1.77009733e-01],
[ 2.48363504e+00,  6.14299139e-01, -8.79956021e-01],
[-8.54690568e-01, -1.23112366e-01,  2.91993108e-01],
[ 9.91167493e-02,  2.08912215e+00, -4.80669231e-02],
[-6.16238739e-01, -8.60523870e-01, -3.19343929e-01],
[ 9.91167493e-02, -1.23112366e-01, -1.41400822e-01],
[ 8.14472237e-01, -4.91818118e-01, -7.79946407e-02],
[ 9.91167493e-02, -8.60523870e-01,  1.68630042e-01],
[-1.39335080e-01,  2.45593386e-01, -4.66040470e-01],
[ 3.37568579e-01, -1.23112366e-01,  3.28900891e-02],
[ 3.37568579e-01, -8.60523870e-01, -3.44300602e-01],
[-8.54690568e-01, -4.91818118e-01,  1.27542836e-01],
[-1.39335080e-01, -8.60523870e-01,  5.95734079e-01],
[ 9.91167493e-02, -4.91818118e-01,  6.65714526e-02],
[-2.58560995e-01, -4.91818118e-01, -1.00820866e-01],
[-8.54690568e-01, -8.60523870e-01,  2.56789996e-01],
[-8.54690568e-01, -1.22922962e+00, -2.87691563e-01],
[ 3.19899053e+00,  1.72041640e+00, -4.66040470e-01],
[-8.54690568e-01, -1.23112366e-01, -3.62155783e-01],
[-8.54690568e-01,  9.83004891e-01, -1.51545811e-01],
[ 9.91167493e-02, -4.91818118e-01, -1.82893827e-01],
[ 8.14472237e-01,  1.72041640e+00,  3.06382142e-03],
[ 1.52982773e+00, -1.23112366e-01, -6.46215475e-01],
[-8.54690568e-01,  2.45593386e-01, -1.69299542e-01],
[-8.54690568e-01,  1.35171064e+00, -3.61972860e-02],
[-8.54690568e-01,  1.35171064e+00, -1.51545811e-01],
[ 9.91167493e-02,  9.83004891e-01,  1.82528677e-01],
[-8.54690568e-01,  2.45593386e-01, -4.24871792e-02],
[ 8.14472237e-01, -1.96664113e+00,  2.33963771e-01],
[ 9.91167493e-02, -4.91818118e-01, -2.78256724e-01],
[-8.54690568e-01, -4.91818118e-01, -3.44300602e-01],
[-8.54690568e-01, -4.91818118e-01, -8.41830840e-02],
[ 1.52982773e+00, -4.91818118e-01, -6.24809548e-01],
[ 6.95246323e-01, -1.96664113e+00,  3.16527131e-03],
[ 3.37568579e-01,  1.35171064e+00, -3.44300602e-01],
[-8.54690568e-01, -8.60523870e-01, -1.74169136e-01],
[-8.54690568e-01, -1.22922962e+00, -7.51926260e-01],
[-2.58560995e-01,  2.45593386e-01, -4.53866483e-01],
[ 9.91167493e-02, -1.23112366e-01,  1.12223903e-01],
[ 3.19899053e+00, -8.60523870e-01,  2.09190021e-02],
[-1.39335080e-01, -1.23112366e-01, -1.49516813e-01],
```

```
[ 2.48363504e+00,  2.45593386e-01, -8.44651459e-01],
[ 3.37568579e-01, -1.23112366e-01, -1.61690800e-01],
[ 1.52982773e+00, -4.91818118e-01, -6.25722597e-01],
[-8.54690568e-01, -1.96664113e+00, -1.80560479e-01],
[-1.39335080e-01,  2.45593386e-01,  7.61077422e-02],
[-8.54690568e-01,  2.45593386e-01, -4.02431389e-01],
[-1.39335080e-01,  9.83004891e-01,  1.75832984e-01],
[-8.54690568e-01,  1.72041640e+00,  4.03587987e-01],
[ 5.76020408e-01, -1.23112366e-01,  2.84688716e-01],
[ 9.91167493e-02, -1.96664113e+00, -4.58350256e-02],
[-6.16238739e-01, -4.91818118e-01, -4.47779490e-01],
[ 9.91167493e-02,  2.08912215e+00, -1.54995107e-01],
[-8.54690568e-01, -1.22922962e+00,  4.32513348e+00],
[ 9.91167493e-02, -1.22922962e+00,  2.96558353e-01],
[-6.16238739e-01, -1.23112366e-01, -3.44300602e-01],
[ 3.37568579e-01, -4.91818118e-01, -7.62699926e-02],
[-2.58560995e-01, -8.60523870e-01, -5.16765415e-01],
[ 1.52982773e+00,  1.35171064e+00, -6.70766348e-01],
[ 2.96053870e+00, -1.23112366e-01, -8.82999518e-01],
[ 9.91167493e-02, -8.60523870e-01,  3.64935579e-01],
[-8.54690568e-01,  1.35171064e+00, -2.50459454e-01],
[ 8.14472237e-01, -8.60523870e-01, -1.68792292e-01],
[ 8.14472237e-01,  6.14299139e-01, -4.38649000e-01],
[ 3.37568579e-01,  6.14299139e-01,  1.41441471e-01],
[ 4.56794493e-01,  2.45593386e-01, -2.22560734e-01],
[ 9.91167493e-02, -4.91818118e-01, -1.10458605e-01],
[ 9.91167493e-02, -4.91818118e-01,  1.42963220e-01],
[-8.54690568e-01,  2.08912215e+00,  4.02959310e-02],
[-6.16238739e-01, -1.23112366e-01, -4.83794201e-01],
[ 9.91167493e-02, -8.60523870e-01, -2.22560734e-01],
[ 2.48363504e+00, -1.23112366e-01, -7.51926260e-01],
[-8.54690568e-01, -1.23112366e-01, -1.03762913e-01],
[-8.54690568e-01,  2.08912215e+00, -1.57937154e-01],
[-8.54690568e-01,  2.45593386e-01,  8.43251833e-02],
[-1.39335080e-01,  6.14299139e-01,  2.26050680e-01],
[-8.54690568e-01, -8.60523870e-01, -3.44300602e-01],
[ 9.91167493e-02,  2.45593386e-01, -2.75821926e-01],
[-8.54690568e-01, -1.59793537e+00, -3.43590453e-01],
[-8.54690568e-01, -1.22922962e+00,  8.17889361e-02],
[-1.39335080e-01,  1.72041640e+00, -5.30765500e-01],
[-1.39335080e-01, -4.91818118e-01, -1.56029583e-02],
[ 1.52982773e+00,  1.35171064e+00, -5.27417653e-01],
[ 1.52982773e+00, -8.60523870e-01, -7.38331975e-01],
[ 9.91167493e-02,  6.14299139e-01,  3.16527131e-03],
[ 9.91167493e-02,  2.08912215e+00, -4.59364754e-02],
[ 5.76020408e-01, -4.91818118e-01,  7.49917934e-02],
[ 9.91167493e-02,  2.45593386e-01,  1.21963092e-01],
[-8.54690568e-01, -1.23112366e-01, -8.15453868e-02],
[-6.16238739e-01, -1.96664113e+00, -5.26910404e-01],
[-6.16238739e-01,  6.14299139e-01,  2.80630720e-01],
```

```
       [ 1.52982773e+00, -8.60523870e-01,  2.11340446e-01],
       [-8.54690568e-01, -8.60523870e-01, -9.57483714e-02],
       [ 3.37568579e-01,  9.83004891e-01,  1.59702452e-01],
       [-8.54690568e-01,  2.45593386e-01,  3.57225387e-01],
       [ 9.91167493e-02,  1.72041640e+00,  2.56789996e-01],
       [ 9.91167493e-02,  1.72041640e+00, -1.02038265e-01],
       [ 2.48363504e+00, -1.23112366e-01, -7.52129160e-01],
       [ 5.76020408e-01,  1.35171064e+00, -1.26183338e-01],
       [-8.54690568e-01,  2.45593386e-01,  6.92821623e-01],
       [-8.54690568e-01, -8.60523870e-01,  2.20978185e-01],
       [-8.54690568e-01,  1.72041640e+00, -1.62116577e-02],
       [-8.54690568e-01, -1.59793537e+00,  2.61862491e-01],
       [-8.54690568e-01,  6.14299139e-01, -5.18205691e-02],
       [-8.54690568e-01, -1.23112366e-01,  1.30789233e-01],
       [ 2.48363504e+00,  2.45593386e-01, -8.16651290e-01],
       [ 5.76020408e-01,  2.08912215e+00, -9.51596494e-03],
       [ 8.14472237e-01,  6.14299139e-01, -3.44300602e-01],
       [-8.54690568e-01, -1.23112366e-01,  2.74543727e-01],
       [-8.54690568e-01,  9.83004891e-01, -1.87154722e-01],
       [ 3.19899053e+00, -1.22922962e+00,  3.06581915e-02],
       [-1.39335080e-01, -4.91818118e-01, -3.44300602e-01],
       [ 9.91167493e-02, -1.22922962e+00,  3.14413534e-01],
       [-8.54690568e-01, -1.23112366e-01,  6.31221563e-02],
       [-8.54690568e-01, -8.60523870e-01, -3.23300475e-01],
       [ 2.48363504e+00,  6.14299139e-01, -8.49825404e-01],
       [-6.16238739e-01, -1.96664113e+00, -7.05868010e-01],
       [-8.54690568e-01, -4.91818118e-01,  7.58033925e-02],
       [ 8.14472237e-01, -4.91818118e-01,  1.26528338e-01],
       [-8.54690568e-01,  2.45593386e-01,  6.63685528e-02],
       [ 1.52982773e+00,  1.35171064e+00, -6.99781016e-01],
       [ 1.52982773e+00, -1.23112366e-01, -6.18215305e-01],
       [-8.54690568e-01, -1.23112366e-01, -6.66322530e-02],
       [-8.54690568e-01, -8.60523870e-01, -1.44444319e-01],
       [ 2.96053870e+00, -1.23112366e-01, -7.01911464e-01],
       [ 2.48363504e+00, -4.91818118e-01, -8.13810693e-01],
       [-1.39335080e-01, -4.91818118e-01, -4.66040470e-01],
       [-1.39335080e-01,  1.35171064e+00, -2.14444743e-01],
       [-1.39335080e-01, -8.60523870e-01, -1.00820866e-01],
       [ 9.91167493e-02, -1.22922962e+00, -2.50459454e-01],
       [ 5.76020408e-01,  6.14299139e-01, -1.00820866e-01],
       [ 9.91167493e-02, -1.23112366e-01, -8.56033824e-02],
       [ 5.76020408e-01, -1.22922962e+00, -2.46604358e-01],
       [ 1.52982773e+00, -4.91818118e-01, -6.25722597e-01],
       [ 1.52982773e+00, -8.60523870e-01, -2.74097278e-01],
       [ 9.91167493e-02,  2.08912215e+00, -2.56242097e-01],
       [ 1.52982773e+00,  2.45593386e-01, -2.61923291e-01],
       [ 9.91167493e-02, -1.23112366e-01, -1.43531270e-01]])

Y_train
```

```
array([[-0.47654967],
       [ 0.04412298],
       [ 3.30843421],
       ...,
       [-0.52555415],
       [-0.75955057],
       [-0.36628958]])
```

```
Y_test
```

```
array([[-9.79900262e-02],
       [ 2.83019842e-01],
       [-4.33670746e-01],
       [-5.56181958e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 6.07674553e-01],
       [-7.52199896e-01],
       [ 4.91288902e-01],
       [-2.49903929e-01],
       [-2.75214745e-01],
       [ 5.15791144e-01],
       [-5.90485097e-01],
       [-5.13303034e-01],
       [-7.22626718e-02],
       [ 4.07981278e-01],
       [ 1.23755267e-01],
       [-6.66442048e-01],
       [-3.81603481e-01],
       [-9.54343396e-01],
       [-3.41787337e-01],
       [ 4.90234277e-02],
       [ 5.28042265e-01],
       [ 5.26817153e-01],
       [ 2.52392039e-01],
       [-7.38723663e-01],
       [-2.74406171e-01],
       [-1.49339273e+00],
       [-5.99060882e-01],
       [-4.15294064e-01],
       [ 1.48257509e-01],
       [-3.67344204e-02],
       [ 2.70874183e+00],
       [ 8.94350788e-01],
       [-5.13303034e-01],
       [-1.00334788e+00],
       [-5.38859900e-02],
       [ 4.60661099e-01],
       [-4.82675231e-01],
       [-9.42092275e-01],
       [-2.65830386e-01],
```

```
[-2.92782853e-01],
[ 2.54413474e-01],
[-3.47912898e-01],
[-4.77604295e-02],
[-1.10070660e-02],
[ 5.28042265e-01],
[-7.83882324e-02],
[-1.10070660e-02],
[ 4.36264319e+00],
[ 3.13647645e-01],
[-7.95078821e-01],
[ 1.10384496e+00],
[-9.42092275e-01],
[-2.74406171e-01],
[-5.56181958e-01],
[ 1.89430863e-05],
[ 1.48257509e-01],
[-1.09016035e-01],
[-4.39796307e-01],
[-3.49909831e-01],
[-9.42092275e-01],
[-6.11312003e-01],
[ 2.57462975e-02],
[-5.01051913e-01],
[ 5.63741004e-02],
[-6.35814245e-01],
[ 5.09665584e-01],
[ 1.80093375e+00],
[-7.59550569e-01],
[-4.27545186e-01],
[-4.52047428e-01],
[-4.58172988e-01],
[-1.15036133e+00],
[-3.43012450e-01],
[-4.53272540e-01],
[-4.03042943e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 1.69189878e+00],
[-5.74558640e-01],
[-1.33412815e+00],
[ 1.20185393e+00],
[-1.70271641e-01],
[-2.68280610e-01],
[-1.21774250e+00],
[ 8.34320294e-01],
[ 5.63741004e-02],
[-1.09016035e-01],
[-4.64298549e-01],
[-6.48065367e-01],
```

```
[-9.05338911e-01],
[-4.64298549e-01],
[-1.04010124e+00],
[ 3.44275448e-01],
[-4.76549670e-01],
[-3.84666262e-01],
[ 6.75055719e-01],
[-1.64146081e-01],
[-4.88150538e-03],
[-6.35814245e-01],
[-2.49903929e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.45137930e-02],
[ 7.10583971e-01],
[-2.43778368e-01],
[-7.14221421e-01],
[-8.57389051e-02],
[-2.68280610e-01],
[ 4.60661099e-01],
[-1.13233994e+00],
[ 4.17782175e-01],
[ 1.53263420e+00],
[-7.64451018e-01],
[ 2.39633824e+00],
[ 1.16510057e+00],
[ 4.66786659e-01],
[ 6.86252216e-02],
[ 7.11809083e-01],
[-2.32581871e-02],
[-7.33823215e-01],
[-5.14528146e-01],
[-1.22386806e+00],
[-3.41787337e-01],
[-4.76549670e-01],
[-5.13915590e-01],
[ 8.09818052e-01],
[ 1.69189878e+00],
[ 9.92530245e-02],
[-2.43778368e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.25657619e-01],
[-7.95078821e-01],
[ 4.91288902e-01],
[ 2.79449968e+00],
[ 1.62835221e+00],
[-6.60316488e-01],
[-2.68280610e-01],
[ 1.96036882e-01],
```

```
[-1.33518278e-01],
[-2.19276126e-01],
[-5.13303034e-01],
[-4.70424110e-01],
[-9.35966714e-01],
[-8.45137930e-02],
[-2.68280610e-01],
[-7.70576578e-01],
[-5.31679715e-01],
[-1.49339273e+00],
[-1.08298017e+00],
[ 9.92530245e-02],
[-4.52047428e-01],
[ 1.03033823e+00],
[-7.27697654e-01],
[ 2.50353555e+00],
[ 1.32436514e+00],
[-1.00334788e+00],
[ 1.63064317e+00],
[-1.00334788e+00],
[ 1.75315438e+00],
[-5.75783752e-01],
[ 6.38302356e-01],
[-3.29536216e-01],
[-1.71326265e-02],
[-5.68433079e-01],
[ 9.06601909e-01],
[-5.74558640e-01],
[ 3.30799215e-01],
[-6.05186442e-01],
[-1.08910573e+00],
[-3.05033974e-01],
[ 6.50553477e-01],
[ 3.13647645e-01],
[-7.70576578e-01],
[-6.37039358e-01],
[ 2.34015357e-01],
[-1.45891910e-01],
[-6.66442048e-01],
[ 1.60508630e-01],
[-1.71326265e-02],
[-8.93590086e-01],
[ 4.23907735e-01],
[-5.77008864e-01],
[-6.35814245e-01],
[-5.13303034e-01],
[-1.45769399e-01],
[-3.05033974e-01],
[-9.78845638e-01],
[-2.69566978e-01],
```

```
[-6.11312003e-01],
[-8.45137930e-02],
[-2.31527247e-01],
[ 2.57462975e-02],
[-1.09016035e-01],
[-7.09320972e-01],
[-5.01051913e-01],
[ 2.58517600e-01],
[-4.64298549e-01],
[-1.34025371e+00],
[ 1.65024496e+00],
[ 9.44580385e-01],
[-2.68893166e-01],
[ 6.26051235e-01],
[-9.78845638e-01],
[ 5.89297871e-01],
[ 9.07827021e-01],
[-6.29688685e-01],
[-1.18711470e+00],
[ 8.70019034e-02],
[-9.64144293e-01],
[ 1.01270887e+00],
[-1.23611918e+00],
[ 2.57292488e-01],
[ 1.42131949e-01],
[-8.56334426e-01],
[-4.52047428e-01],
[ 8.33265670e-02],
[-2.93837477e-02],
[ 4.97414462e-01],
[ 1.07321716e+00],
[ 6.99557962e-01],
[-1.24959542e+00],
[-1.33518278e-01],
[-3.78540701e-01],
[-6.48065367e-01],
[ 1.02911312e+00],
[-5.50056397e-01],
[ 7.07962231e-01],
[-9.11464472e-01],
[-2.32581871e-02],
[-8.01204381e-01],
[ 3.48668803e+00],
[-5.25554155e-01],
[-8.80836669e-01],
[-2.32581871e-02],
[ 1.85010873e-01],
[ 1.25085841e+00],
[ 3.79974187e-02],
[-4.29995410e-01],
```

```
[ 4.05531054e-01],
[-4.76549670e-01],
[-1.17486358e+00],
[-2.25401686e-01],
[ 1.12834720e+00],
[-1.71496753e-01],
[ 2.17091761e+00],
[ 5.89297871e-01],
[ 6.87306840e-01],
[ 1.15284944e+00],
[-3.78540701e-01],
[-2.92782853e-01],
[ 3.91960590e+00],
[-3.72415140e-01],
[-8.19581063e-01],
[ 3.44275448e-01],
[ 3.67723066e-02],
[ 2.58517600e-01],
[-4.22644737e-01],
[-5.25554155e-01],
[-5.13303034e-01],
[-6.11312003e-01],
[ 1.23755267e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 2.42696605e+00],
[ 6.21150786e-01],
[-3.94467158e-01],
[-1.15036133e+00],
[-1.55464833e+00],
[-7.39948775e-01],
[-7.76702139e-01],
[ 1.26310954e+00],
[-1.45769399e-01],
[-2.07025005e-01],
[-4.21419625e-01],
[-5.13303034e-01],
[-4.15294064e-01],
[-3.29536216e-01],
[-6.05186442e-01],
[-7.76702139e-01],
[-7.70576578e-01],
[-9.67649141e-02],
[ 6.24996610e-02],
[ 1.13202254e+00],
[-4.76549670e-01],
[-3.99367607e-01],
[-1.00947344e+00],
[ 2.21764236e-01],
```

```
       [ 1.36006388e-01],
       [ 1.36006388e-01]])
```

Multiple Regression Model

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)#training func question + answers
# The coefficients
print ('Intercept: ',regr.intercept_)
print ('Coefficient : ',regr.coef_)
```

```
Intercept:  [3.82889473e-17]
Coefficient :  [[-0.05383768  0.04488339  0.24819609]]
```

```
regr.intercept_
```

```
array([3.82889473e-17])
```

```
regr.coef_
```

```
array([[-0.05383768,  0.04488339,  0.24819609]])
```

```
regr.coef_[0][1]
```

```
0.044883387796239835
```

```
regr.coef_[0][2]
```

```
0.2481960930581826
```

```
y_pred = regr.predict(X_test)
```

```
y_pred
```

```
array([[-2.17730518e-02],
       [ 3.59189594e-02],
       [-5.17444895e-02],
       [-4.06132510e-02],
       [-2.73418038e-02],
       [-4.36657158e-02],
       [ 1.89497631e-02],
       [ 3.17355357e-02],
       [ 1.32619818e-01],
       [ 1.01242368e-01],
       [-1.34914083e-01],
       [-1.94215914e-01],
       [-1.93518376e-02],
       [ 5.85223979e-02],
       [-3.24655389e-01],
       [-6.00575484e-02],
       [ 5.04321714e-02],
```

```
[-2.49067585e-02],
[-2.36827058e-02],
[-1.41325259e-01],
[-2.78150679e-01],
[-1.84686878e-02],
[ 1.44401737e-01],
[ 6.27970533e-02],
[-1.29864816e-01],
[-1.13027273e-01],
[-1.95883065e-03],
[ 1.47353060e-02],
[-3.74113688e-02],
[-1.30378972e-02],
[ 1.03541114e-01],
[-3.29717740e-03],
[ 1.20107387e-01],
[ 2.18957607e-02],
[-1.86638659e-01],
[-2.80939746e-01],
[-1.88560165e-02],
[ 5.65834286e-02],
[ 1.09888509e-01],
[-2.25290886e-03],
[-2.24866943e-01],
[ 8.63398543e-02],
[-2.33261649e-01],
[-3.51413818e-02],
[ 2.87577732e-03],
[ 1.18608613e-02],
[-2.31754841e-02],
[-3.39813428e-01],
[-2.91061996e-01],
[ 1.06959403e+00],
[-2.60248183e-01],
[-1.31067478e-01],
[ 4.68099606e-02],
[-8.53960165e-02],
[ 1.46404420e-02],
[-5.32629275e-02],
[ 1.40985150e-01],
[-2.73706156e-02],
[ 2.31511016e-02],
[ 7.23441610e-02],
[-1.39411721e-01],
[-5.88644768e-02],
[-4.13703989e-02],
[ 1.69114704e-01],
[ 5.52121801e-02],
[-1.22671263e-01],
[ 6.95293648e-03],
```

```
[ 2.31443115e-01],
[-3.08484422e-01],
[-2.64647070e-01],
[-3.47870857e-01],
[-7.32437160e-02],
[-2.42887823e-01],
[-4.13329852e-01],
[-2.05564236e-01],
[ 1.40244270e-02],
[ 7.13872496e-03],
[ 2.80899094e-02],
[-6.00728885e-02],
[ 1.74641746e-02],
[ 2.93645759e-02],
[-1.07356091e-01],
[ 2.48224326e-02],
[ 2.37244336e-01],
[-1.89108671e-01],
[ 4.02807565e-02],
[-2.94510251e-02],
[-3.57903967e-02],
[-7.05556219e-02],
[ 1.30231023e-02],
[-9.50530211e-02],
[ 4.47949490e-03],
[-5.74776364e-02],
[ 1.92864795e-01],
[-4.07390438e-02],
[-1.69445173e-01],
[ 4.49682272e-02],
[-8.87293427e-02],
[-7.61647368e-02],
[-1.08325277e-03],
[-1.12241058e-01],
[ 6.98958989e-02],
[ 3.69184158e-02],
[-8.37262468e-02],
[ 3.06214677e-01],
[-1.41553813e-02],
[-2.83895020e-02],
[ 1.00384535e-02],
[-9.27135000e-02],
[-2.49628400e-01],
[-5.45877503e-02],
[-1.39553011e-01],
[-1.71505956e-01],
[ 6.07498987e-02],
[-4.29878615e-01],
[ 2.14910299e-01],
[ 3.73876470e+00],
```

```
[-9.09761691e-02],
[ 4.85511266e-02],
[ 9.33059510e-01],
[ 5.69941728e-02],
[-4.82426293e-02],
[ 1.17605134e-01],
[ 3.48964407e-02],
[ 1.33460580e-01],
[ 3.13365442e-01],
[ 2.16772208e-02],
[-6.34007841e-04],
[ 1.79407160e-02],
[ 6.92793155e-01],
[-2.41326696e-01],
[-2.27497164e-01],
[ 9.20965703e-02],
[-1.33604698e-01],
[-1.30242410e-01],
[ 4.19303735e-01],
[ 5.62250099e-02],
[ 8.47400763e-02],
[-2.96744937e-01],
[-2.30475679e-02],
[ 1.46698720e-01],
[-9.44199021e-02],
[-2.57875459e-01],
[ 2.19424872e-02],
[ 7.37542111e-02],
[-7.63795951e-02],
[ 1.17172993e-01],
[ 6.67230835e-02],
[-1.23997955e-01],
[-4.65884023e-02],
[-1.67740541e-01],
[-8.44390923e-02],
[-2.11300934e-02],
[-5.96682531e-02],
[-1.15272433e-01],
[-3.24542964e-01],
[ 1.12960404e-01],
[ 7.65006413e-02],
[-8.47062793e-02],
[-4.59570472e-02],
[-8.52817224e-02],
[-2.10612464e-03],
[-9.71448835e-02],
[-1.55364169e-02],
[-1.42251199e-01],
[ 5.55957259e-02],
[ 1.16737121e-01],
```

```
[-1.08879045e-02],
[-3.31774848e-02],
[ 7.11256028e-02],
[-8.05613563e-02],
[-2.10677530e-01],
[-4.93967948e-02],
[ 5.25220671e-02],
[-7.28042122e-02],
[ 3.41292507e-02],
[-2.48276229e-01],
[ 1.50181340e-02],
[ 9.76998836e-02],
[ 6.90708304e-02],
[ 8.40872785e-02],
[ 4.64924669e-02],
[-7.40499165e-02],
[-9.64729106e-02],
[-6.15139720e-02],
[ 3.04617971e-03],
[-2.59512124e-01],
[-1.24914356e-01],
[-4.29586197e-02],
[-3.58367702e-02],
[-1.95782594e-01],
[-8.77045011e-02],
[ 1.69916186e-02],
[-2.05657434e-01],
[-3.51337118e-02],
[-3.32329273e-01],
[-6.38306333e-02],
[-2.59738740e-01],
[-8.70693664e-02],
[ 3.74141846e-02],
[-4.28442797e-02],
[ 9.52631266e-02],
[ 2.23401633e-01],
[ 3.41213258e-02],
[-1.04981806e-01],
[-1.00034720e-01],
[ 4.99614838e-02],
[ 1.06432380e+00],
[ 1.30964191e-02],
[-5.78029015e-02],
[-5.91782859e-02],
[-1.52962060e-01],
[-1.88174606e-01],
[-3.84071260e-01],
[ 4.66161427e-02],
[ 4.45208507e-02],
[-1.24366108e-01],
```

```
[-1.25148435e-01],
[ 4.45031386e-02],
[-6.88083963e-02],
[-5.48260733e-02],
[ 8.07223363e-03],
[ 1.49782728e-01],
[-9.24246677e-02],
[-9.91981468e-02],
[-3.25864004e-01],
[ 1.47353060e-02],
[ 1.00582051e-01],
[ 7.79667998e-02],
[ 9.11781991e-02],
[-7.80627352e-02],
[-6.27710769e-02],
[-1.10984005e-01],
[ 1.11422601e-02],
[-4.70143299e-02],
[-1.84455795e-02],
[-1.52596020e-01],
[-3.04236710e-01],
[ 2.30212188e-02],
[ 7.70294101e-02],
[-3.44733944e-02],
[ 3.59576106e-02],
[ 2.02496091e-02],
[-1.85869757e-01],
[ 1.30400138e-01],
[-6.85317261e-02],
[-1.63730427e-02],
[ 6.55842058e-02],
[ 1.45699564e-01],
[ 1.35616174e-01],
[ 4.65564020e-02],
[-3.25914363e-01],
[-1.66045985e-03],
[ 2.28993239e-01],
[ 6.22372512e-02],
[ 1.19209002e-01],
[ 3.92870496e-02],
[ 6.07247192e-02],
[ 7.29502321e-02],
[-3.25379740e-01],
[ 6.03934530e-02],
[-1.01731532e-01],
[ 1.08629536e-01],
[ 4.36840744e-02],
[-2.19788969e-01],
[-1.00027050e-01],
[ 1.75280052e-02],
```

```
        [ 5.61555281e-02],
        [-7.28505857e-02],
        [-3.17064662e-01],
        [-2.30286336e-01],
        [ 4.27541981e-02],
        [-3.45199183e-02],
        [ 7.35100343e-02],
        [-1.95375934e-01],
        [-2.41326696e-01],
        [ 2.39509906e-02],
        [-2.84591866e-02],
        [-3.39125912e-01],
        [-3.57772241e-01],
        [-1.30242410e-01],
        [ 1.49464828e-02],
        [-5.61450944e-02],
        [-1.22671263e-01],
        [-2.84631198e-02],
        [-3.21083408e-02],
        [-1.47389829e-01],
        [-2.59738740e-01],
        [-1.89015473e-01],
        [ 2.48323764e-02],
        [-1.36347647e-01],
        [-4.64858160e-02]])
```

```
from sklearn.metrics import r2_score
```

```
print(f"accuracy : {r2_score(Y_test,y_pred)*100} % ")
```

```
accuracy : 9.092114450354227 %
```

```
print(f"Mean absolute error: {np.mean(np.absolute(y_pred - Y_test))}
")#pred - actual
```

```
Mean absolute error: 0.5837743155131746
```

```
print("Residual sum of squares (MSE): %.2f" % np.mean((y_pred -
Y_test) **2))
```

```
Residual sum of squares (MSE): 0.66
```

```
accuracy = print(f"accuracy : {r2_score(Y_test,y_pred)*100} % ")
```

```
accuracy : 9.092114450354227 %
```

Apply Lasso Regression to cover overfitting and undefitting Problem

```
from sklearn.linear_model import Lasso
```

```
L = Lasso(alpha = 1)
```

```
L.fit(X_train,Y_train)
```

```
Lasso(alpha=1)
```

```
ypred = L.predict(X_test)
```

```
from sklearn.metrics import r2_score
print("R2-score",r2_score(Y_test,ypred))
print(f"Mean absolute error: {np.mean(np.absolute(ypred - Y_test))}
")# pred - actual
```

```
R2-score -0.004908518792945626
Mean absolute error: 0.6243935831081134
```

regr.fit (Independent and Dependent Variable)

```
regr.fit(X_train, Y_train)
```

```
LinearRegression()
```

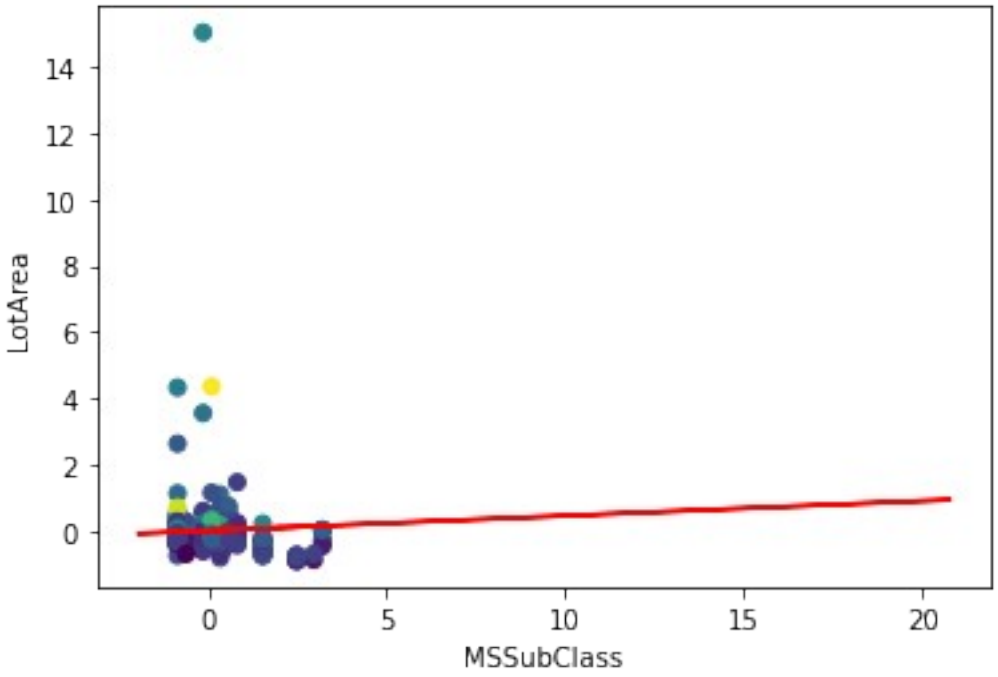
```
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

```
Coefficients:  [[-0.05383768  0.04488339  0.24819609]]
Intercept:  [3.82889473e-17]
```

Plot Outputs

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,0],X_test[:,2],c=Y_test)
plt.plot(X_train, regr.coef_[0][1]*X_train + regr.intercept_[0], 'r')
#y = mx+c
plt.xlabel("MSSubClass")
plt.ylabel("LotArea")
```

```
Text(0, 0.5, 'LotArea')
```



3rd Model

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing

df = pd.read_csv("C:/Users/Lenovo/Documents/Data Set/housing.csv")

df
```

```
        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
...    ...         ...      ...          ...      ...    ...   ...
...
1455  1456          60       RL         62.0     7917   Pave   NaN
Reg
1456  1457          20       RL         85.0    13175   Pave   NaN
Reg
```

```
1457  1458          70       RL         66.0     9042   Pave   NaN
Reg
1458  1459          20       RL         68.0     9717   Pave   NaN
Reg
1459  1460          20       RL         75.0     9937   Pave   NaN
Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature
MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
3            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN
0
...          ...       ...  ...      ...    ...    ...         ...
...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN
0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed
2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN
0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN
0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
...     ...    ...       ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[1460 rows x 81 columns]

df.describe()
```

```
                Id   MSSubClass  LotFrontage        LotArea
OverallQual  \
count  1460.000000  1460.000000  1201.000000    1460.000000
1460.000000
mean    730.500000    56.897260    70.049958   10516.828082
6.099315
std     421.610009    42.300571    24.284752    9981.264932
1.382997
min       1.000000    20.000000    21.000000    1300.000000
1.000000
25%     365.750000    20.000000    59.000000    7553.500000
5.000000
50%     730.500000    50.000000    69.000000    9478.500000
6.000000
75%    1095.250000    70.000000    80.000000   11601.500000
7.000000
max    1460.000000   190.000000   313.000000  215245.000000
10.000000

       OverallCond    YearBuilt  YearRemodAdd   MasVnrArea
BsmtFinSF1  ...  \
count  1460.000000  1460.000000   1460.000000  1452.000000
1460.000000  ...
mean      5.575342  1971.267808   1984.865753   103.685262
443.639726  ...
std       1.112799    30.202904     20.645407   181.066207
456.098091  ...
min       1.000000  1872.000000   1950.000000     0.000000
0.000000  ...
25%       5.000000  1954.000000   1967.000000     0.000000
0.000000  ...
50%       5.000000  1973.000000   1994.000000     0.000000
383.500000  ...
75%       6.000000  2000.000000   2004.000000   166.000000
712.250000  ...
max       9.000000  2010.000000   2010.000000  1600.000000
5644.000000  ...

        WoodDeckSF  OpenPorchSF  EnclosedPorch    3SsnPorch
ScreenPorch  \
count  1460.000000  1460.000000    1460.000000  1460.000000
1460.000000
mean     94.244521    46.660274      21.954110     3.409589
15.060959
std     125.338794    66.256028      61.119149    29.317331
55.757415
min       0.000000     0.000000       0.000000     0.000000
0.000000
25%       0.000000     0.000000       0.000000     0.000000
0.000000
```

```
50%       0.000000    25.000000       0.000000     0.000000
0.000000
75%     168.000000    68.000000       0.000000     0.000000
0.000000
max     857.000000   547.000000     552.000000   508.000000
480.000000

          PoolArea       MiscVal       MoSold       YrSold
SalePrice
count  1460.000000   1460.000000  1460.000000  1460.000000
1460.000000
mean      2.758904     43.489041     6.321918  2007.815753
180921.195890
std      40.177307    496.123024     2.703626     1.328095
79442.502883
min       0.000000      0.000000     1.000000  2006.000000
34900.000000
25%       0.000000      0.000000     5.000000  2007.000000
129975.000000
50%       0.000000      0.000000     6.000000  2008.000000
163000.000000
75%       0.000000      0.000000     8.000000  2009.000000
214000.000000
max     738.000000  15500.000000    12.000000  2010.000000
755000.000000

[8 rows x 38 columns]

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
```

```
cdf2 = df[["OverallCond","OverallQual","MiscVal","SalePrice"]]
```

```
cdf2
```

```
      OverallCond  OverallQual  MiscVal  SalePrice
0               5            7        0     208500
1               8            6        0     181500
2               5            7        0     223500
3               5            7        0     140000
4               5            8        0     250000
...           ...          ...      ...        ...
1455            5            6        0     175000
1456            6            6        0     210000
1457            9            7     2500     266500
1458            6            5        0     142125
1459            6            5        0     147500

[1460 rows x 4 columns]
```

```
X = df[["OverallCond","OverallQual","MiscVal"]] # Independent Variable
```

```
X
```

```
      OverallCond  OverallQual  MiscVal
0               5            7        0
1               8            6        0
2               5            7        0
3               5            7        0
4               5            8        0
...           ...          ...      ...
1455            5            6        0
1456            6            6        0
```

```
1457            9            7     2500
1458            6            5        0
1459            6            5        0

[1460 rows x 3 columns]
```

```
Y = df[["SalePrice"]]
```

```
Y
```

```
      SalePrice
0        208500
1        181500
2        223500
3        140000
4        250000
...         ...
1455     175000
1456     210000
1457     266500
1458     142125
1459     147500

[1460 rows x 1 columns]
```

```
# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 500)
```

```
X_train.shape
```

```
(1168, 3)
```

```
Y_train.shape
```

```
(1168, 1)
```

```
X_test.shape
```

```
(292, 3)
```

```
Y_test.shape
```

```
(292, 1)
```

```
# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_Y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
```

```
X_test = sc_X.transform(X_test)
Y_train = sc_Y.fit_transform(Y_train)
Y_test = sc_Y.transform(Y_test)
```

```
X_train
```

```
array([[-0.5039999 , -0.78256106, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-0.5039999 ,  2.77121545, -0.11595495],
       ...,
       [ 0.39063822, -0.07180576, -0.11595495],
       [-0.5039999 , -0.78256106, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495]])
```

```
Y_train
```

```
array([[-0.47654967],
       [ 0.04412298],
       [ 3.30843421],
       ...,
       [-0.52555415],
       [-0.75955057],
       [-0.36628958]])
```

```
X_test
```

```
array([[-0.5039999 , -0.07180576, -0.11595495],
       [ 0.39063822,  0.63894954, -0.11595495],
       [ 1.28527634, -0.78256106, -0.11595495],
       [ 0.39063822, -0.78256106, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [ 0.39063822, -1.49331636,  1.46830692],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-0.5039999 , -1.49331636, -0.11595495],
       [-0.5039999 , -0.78256106, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-0.5039999 , -0.78256106, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [ 0.39063822, -0.07180576, -0.11595495],
       [ 0.39063822, -0.78256106, -0.11595495],
       [ 0.39063822, -0.07180576, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-1.39863801,  0.63894954, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
```

```
[ 1.28527634, -1.49331636, -0.11595495],
[ 2.17991445,  0.63894954, -0.11595495],
[-2.29327613, -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106,  1.15145454],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[ 0.39063822,  0.63894954, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 2.17991445, -0.07180576, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[-1.39863801, -2.20407167, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[ 2.17991445, -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -1.49331636, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 1.28527634,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
```

```
[ 1.28527634, -0.78256106, -0.11595495],
[ 1.28527634, -0.07180576,  2.10201166],
[-0.5039999 , -0.07180576,  1.15145454],
[ 2.17991445, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 3.07455257,  1.34970485, -0.11595495],
[-1.39863801, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -1.49331636, -0.11595495],
[ 0.39063822,  0.63894954, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 2.17991445, -0.07180576, -0.11595495],
[-1.39863801, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -1.49331636, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[ 2.17991445, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -1.49331636, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[ 1.28527634, -0.07180576,  1.46830692],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 2.17991445, -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106,  1.78515929],
[-1.39863801, -1.49331636, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
```

```
[-1.39863801, -0.78256106, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 2.17991445,  0.63894954, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[ 2.17991445, -0.78256106, -0.11595495],
[ 3.07455257, -0.07180576, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 1.28527634, -0.07180576,  4.0031259 ],
[ 1.28527634, -0.78256106, -0.11595495],
[ 2.17991445, -0.78256106, -0.11595495],
[-2.29327613, -2.91482697, -0.11595495],
[-1.39863801, -2.20407167, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  2.77121545, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -1.49331636, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 3.07455257, -0.07180576, -0.11595495],
[ 3.07455257,  0.63894954, -0.11595495],
[ 3.07455257,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-1.39863801, -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
```

```
[-1.39863801, -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822, -0.07180576,  1.30988073],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-1.39863801, -1.49331636, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822,  0.63894954, -0.11595495],
[ 1.28527634,  0.63894954, -0.11595495],
[-0.5039999 , -1.49331636, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 2.17991445,  0.63894954, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[ 0.39063822,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[ 0.39063822, -2.20407167, -0.11595495],
[ 2.17991445,  1.34970485, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 0.39063822, -1.49331636, -0.11595495],
[ 2.17991445, -0.07180576, -0.11595495],
[-1.39863801, -0.78256106, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 0.39063822, -1.49331636, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 1.28527634, -1.49331636, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[ 2.17991445, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -1.49331636, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
```

```
[-0.5039999 ,  0.63894954, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  2.77121545, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[ 0.39063822,  1.34970485, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[ 2.17991445, -0.07180576, 48.99616286],
[-0.5039999 , -1.49331636, -0.11595495],
[ 0.39063822, -0.07180576, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 , -0.78256106, -0.11595495],
[ 1.28527634, -0.78256106, -0.11595495],
[-0.5039999 ,  0.63894954, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[-1.39863801, -0.78256106, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
[ 1.28527634, -0.07180576, -0.11595495],
[-1.39863801, -1.49331636, -0.11595495],
[-1.39863801, -1.49331636, -0.11595495],
[ 0.39063822, -0.78256106, -0.11595495],
[-0.5039999 , -0.07180576, -0.11595495],
[-0.5039999 ,  2.06046015, -0.11595495],
[-0.5039999 ,  1.34970485, -0.11595495],
```

```
       [-0.5039999 , -0.07180576, -0.11595495],
       [ 0.39063822, -0.78256106, -0.11595495],
       [ 1.28527634, -0.78256106,  1.15145454],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495],
       [ 0.39063822, -0.07180576, -0.11595495],
       [-0.5039999 , -0.78256106, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [ 1.28527634, -0.07180576, -0.11595495],
       [ 0.39063822,  0.63894954, -0.11595495],
       [ 0.39063822, -0.78256106, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 , -0.07180576, -0.11595495],
       [-0.5039999 ,  0.63894954, -0.11595495]])
```

```
Y_test
```

```
array([[-9.79900262e-02],
       [ 2.83019842e-01],
       [-4.33670746e-01],
       [-5.56181958e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 6.07674553e-01],
       [-7.52199896e-01],
       [ 4.91288902e-01],
       [-2.49903929e-01],
       [-2.75214745e-01],
       [ 5.15791144e-01],
       [-5.90485097e-01],
       [-5.13303034e-01],
       [-7.22626718e-02],
       [ 4.07981278e-01],
       [ 1.23755267e-01],
       [-6.66442048e-01],
       [-3.81603481e-01],
       [-9.54343396e-01],
       [-3.41787337e-01],
       [ 4.90234277e-02],
       [ 5.28042265e-01],
       [ 5.26817153e-01],
       [ 2.52392039e-01],
       [-7.38723663e-01],
       [-2.74406171e-01],
       [-1.49339273e+00],
       [-5.99060882e-01],
       [-4.15294064e-01],
       [ 1.48257509e-01],
```

```
[-3.67344204e-02],
[ 2.70874183e+00],
[ 8.94350788e-01],
[-5.13303034e-01],
[-1.00334788e+00],
[-5.38859900e-02],
[ 4.60661099e-01],
[-4.82675231e-01],
[-9.42092275e-01],
[-2.65830386e-01],
[-2.92782853e-01],
[ 2.54413474e-01],
[-3.47912898e-01],
[-4.77604295e-02],
[-1.10070660e-02],
[ 5.28042265e-01],
[-7.83882324e-02],
[-1.10070660e-02],
[ 4.36264319e+00],
[ 3.13647645e-01],
[-7.95078821e-01],
[ 1.10384496e+00],
[-9.42092275e-01],
[-2.74406171e-01],
[-5.56181958e-01],
[ 1.89430863e-05],
[ 1.48257509e-01],
[-1.09016035e-01],
[-4.39796307e-01],
[-3.49909831e-01],
[-9.42092275e-01],
[-6.11312003e-01],
[ 2.57462975e-02],
[-5.01051913e-01],
[ 5.63741004e-02],
[-6.35814245e-01],
[ 5.09665584e-01],
[ 1.80093375e+00],
[-7.59550569e-01],
[-4.27545186e-01],
[-4.52047428e-01],
[-4.58172988e-01],
[-1.15036133e+00],
[-3.43012450e-01],
[-4.53272540e-01],
[-4.03042943e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 1.69189878e+00],
[-5.74558640e-01],
```

```
[-1.33412815e+00],
[ 1.20185393e+00],
[-1.70271641e-01],
[-2.68280610e-01],
[-1.21774250e+00],
[ 8.34320294e-01],
[ 5.63741004e-02],
[-1.09016035e-01],
[-4.64298549e-01],
[-6.48065367e-01],
[-9.05338911e-01],
[-4.64298549e-01],
[-1.04010124e+00],
[ 3.44275448e-01],
[-4.76549670e-01],
[-3.84666262e-01],
[ 6.75055719e-01],
[-1.64146081e-01],
[-4.88150538e-03],
[-6.35814245e-01],
[-2.49903929e-01],
[-3.29536216e-01],
[-3.29536216e-01],
[-8.45137930e-02],
[ 7.10583971e-01],
[-2.43778368e-01],
[-7.14221421e-01],
[-8.57389051e-02],
[-2.68280610e-01],
[ 4.60661099e-01],
[-1.13233994e+00],
[ 4.17782175e-01],
[ 1.53263420e+00],
[-7.64451018e-01],
[ 2.39633824e+00],
[ 1.16510057e+00],
[ 4.66786659e-01],
[ 6.86252216e-02],
[ 7.11809083e-01],
[-2.32581871e-02],
[-7.33823215e-01],
[-5.14528146e-01],
[-1.22386806e+00],
[-3.41787337e-01],
[-4.76549670e-01],
[-5.13915590e-01],
[ 8.09818052e-01],
[ 1.69189878e+00],
[ 9.92530245e-02],
[-2.43778368e-01],
```

[-3.29536216e-01],
[-3.29536216e-01],
[-8.25657619e-01],
[-7.95078821e-01],
[4.91288902e-01],
[2.79449968e+00],
[1.62835221e+00],
[-6.60316488e-01],
[-2.68280610e-01],
[1.96036882e-01],
[-1.33518278e-01],
[-2.19276126e-01],
[-5.13303034e-01],
[-4.70424110e-01],
[-9.35966714e-01],
[-8.45137930e-02],
[-2.68280610e-01],
[-7.70576578e-01],
[-5.31679715e-01],
[-1.49339273e+00],
[-1.08298017e+00],
[9.92530245e-02],
[-4.52047428e-01],
[1.03033823e+00],
[-7.27697654e-01],
[2.50353555e+00],
[1.32436514e+00],
[-1.00334788e+00],
[1.63064317e+00],
[-1.00334788e+00],
[1.75315438e+00],
[-5.75783752e-01],
[6.38302356e-01],
[-3.29536216e-01],
[-1.71326265e-02],
[-5.68433079e-01],
[9.06601909e-01],
[-5.74558640e-01],
[3.30799215e-01],
[-6.05186442e-01],
[-1.08910573e+00],
[-3.05033974e-01],
[6.50553477e-01],
[3.13647645e-01],
[-7.70576578e-01],
[-6.37039358e-01],
[2.34015357e-01],
[-1.45891910e-01],
[-6.66442048e-01],
[1.60508630e-01],

```
[-1.71326265e-02],
[-8.93590086e-01],
[ 4.23907735e-01],
[-5.77008864e-01],
[-6.35814245e-01],
[-5.13303034e-01],
[-1.45769399e-01],
[-3.05033974e-01],
[-9.78845638e-01],
[-2.69566978e-01],
[-6.11312003e-01],
[-8.45137930e-02],
[-2.31527247e-01],
[ 2.57462975e-02],
[-1.09016035e-01],
[-7.09320972e-01],
[-5.01051913e-01],
[ 2.58517600e-01],
[-4.64298549e-01],
[-1.34025371e+00],
[ 1.65024496e+00],
[ 9.44580385e-01],
[-2.68893166e-01],
[ 6.26051235e-01],
[-9.78845638e-01],
[ 5.89297871e-01],
[ 9.07827021e-01],
[-6.29688685e-01],
[-1.18711470e+00],
[ 8.70019034e-02],
[-9.64144293e-01],
[ 1.01270887e+00],
[-1.23611918e+00],
[ 2.57292488e-01],
[ 1.42131949e-01],
[-8.56334426e-01],
[-4.52047428e-01],
[ 8.33265670e-02],
[-2.93837477e-02],
[ 4.97414462e-01],
[ 1.07321716e+00],
[ 6.99557962e-01],
[-1.24959542e+00],
[-1.33518278e-01],
[-3.78540701e-01],
[-6.48065367e-01],
[ 1.02911312e+00],
[-5.50056397e-01],
[ 7.07962231e-01],
[-9.11464472e-01],
```

```
[-2.32581871e-02],
[-8.01204381e-01],
[ 3.48668803e+00],
[-5.25554155e-01],
[-8.80836669e-01],
[-2.32581871e-02],
[ 1.85010873e-01],
[ 1.25085841e+00],
[ 3.79974187e-02],
[-4.29995410e-01],
[ 4.05531054e-01],
[-4.76549670e-01],
[-1.17486358e+00],
[-2.25401686e-01],
[ 1.12834720e+00],
[-1.71496753e-01],
[ 2.17091761e+00],
[ 5.89297871e-01],
[ 6.87306840e-01],
[ 1.15284944e+00],
[-3.78540701e-01],
[-2.92782853e-01],
[ 3.91960590e+00],
[-3.72415140e-01],
[-8.19581063e-01],
[ 3.44275448e-01],
[ 3.67723066e-02],
[ 2.58517600e-01],
[-4.22644737e-01],
[-5.25554155e-01],
[-5.13303034e-01],
[-6.11312003e-01],
[ 1.23755267e-01],
[ 2.21764236e-01],
[-5.37805276e-01],
[ 2.42696605e+00],
[ 6.21150786e-01],
[-3.94467158e-01],
[-1.15036133e+00],
[-1.55464833e+00],
[-7.39948775e-01],
[-7.76702139e-01],
[ 1.26310954e+00],
[-1.45769399e-01],
[-2.07025005e-01],
[-4.21419625e-01],
[-5.13303034e-01],
[-4.15294064e-01],
[-3.29536216e-01],
[-6.05186442e-01],
```

```
       [-7.76702139e-01],
       [-7.70576578e-01],
       [-9.67649141e-02],
       [ 6.24996610e-02],
       [ 1.13202254e+00],
       [-4.76549670e-01],
       [-3.99367607e-01],
       [-1.00947344e+00],
       [ 2.21764236e-01],
       [ 1.36006388e-01],
       [ 1.36006388e-01]])
```

Multiple Regression Model

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)#training func question + answers
# The coefficients
print ('Intercept: ',regr.intercept_)
print ('Coefficient : ',regr.coef_)
```

```
Intercept:  [-5.45294703e-17]
Coefficient :  [[-0.00535786  0.79887333  0.01851957]]
```

```
regr.intercept_
```

```
array([-5.45294703e-17])
```

```
regr.coef_
```

```
array([[-0.00535786,  0.79887333,  0.01851957]])
```

```
regr.coef_[0][0]
```

```
-0.005357859174442641
```

```
regr.coef_[0][1]
```

```
0.7988733281398422
```

```
regr.coef_[0][2]
```

```
0.01851956792706759
```

```
y_pred = regr.predict(X_test)
```

```
y_pred
```

```
array([[-0.05681078],
       [ 0.50619933],
       [-0.63420092],
       [-0.62940758],
       [-0.05681078],
```

```
[-0.05681078],
[ 0.51099267],
[-1.16787119],
[ 0.51099267],
[-1.19241769],
[-0.62461423],
[ 0.51099267],
[-0.62461423],
[-0.05681078],
[-0.05681078],
[ 0.51099267],
[ 0.51099267],
[-0.06160413],
[-0.62940758],
[-0.06160413],
[ 0.51099267],
[ 0.51099267],
[ 0.51578602],
[ 0.51099267],
[ 0.51099267],
[-1.20200438],
[ 0.49661264],
[-0.61502754],
[-0.62461423],
[-0.61072905],
[-0.63420092],
[-0.05681078],
[ 1.07879613],
[ 0.51099267],
[-0.06160413],
[-0.05681078],
[-0.06639747],
[ 0.50619933],
[-0.62461423],
[-0.63420092],
[ 0.51099267],
[-0.05681078],
[ 0.51099267],
[-0.62940758],
[ 0.51099267],
[-0.05681078],
[ 0.51099267],
[ 0.51099267],
[ 0.51099267],
[ 1.07879613],
[ 0.51099267],
[-0.07119082],
[-0.06639747],
[-1.7554278 ],
[-0.06160413],
```

```
[-0.05681078],
[-0.06160413],
[ 0.51099267],
[-0.06160413],
[-0.63899427],
[-0.62461423],
[-0.62461423],
[-0.06160413],
[-0.05681078],
[-1.19241769],
[ 0.51099267],
[-0.62940758],
[-0.62940758],
[ 1.64659958],
[-0.63420092],
[-0.05681078],
[ 0.50140598],
[-0.05681078],
[-0.05681078],
[-0.05681078],
[-0.63420092],
[-0.02532169],
[-0.0333389 ],
[-0.63899427],
[ 0.51099267],
[-0.62461423],
[-0.62461423],
[ 1.07879613],
[ 1.05962275],
[-0.61982089],
[ 0.51099267],
[ 0.51099267],
[-0.05681078],
[-0.05681078],
[-0.62461423],
[-0.63420092],
[-1.19241769],
[ 0.50619933],
[-0.62461423],
[ 0.51099267],
[-0.05681078],
[-0.63420092],
[ 1.07879613],
[-0.05681078],
[-0.07119082],
[-0.61982089],
[-0.05681078],
[-0.05681078],
[-0.05681078],
[ 0.51099267],
```

```
[ 1.07879613],
[-0.63420092],
[-1.19241769],
[-0.06639747],
[-0.63899427],
[ 0.51099267],
[-1.20200438],
[ 0.51099267],
[ 1.07879613],
[-0.63420092],
[ 1.64659958],
[-0.03705763],
[ 1.07879613],
[ 0.51099267],
[-0.07119082],
[ 0.51099267],
[-0.62940758],
[-0.58940642],
[-1.18762434],
[-0.62940758],
[-0.61982089],
[-0.62940758],
[ 1.07879613],
[ 1.64659958],
[-0.62461423],
[-0.05681078],
[ 0.49661264],
[-0.62940758],
[-0.06639747],
[-0.63899427],
[-0.07598416],
[ 1.07879613],
[ 1.64659958],
[-0.05681078],
[-0.63420092],
[-0.05681078],
[-0.06160413],
[-0.05681078],
[-0.06160413],
[-0.63420092],
[-0.06160413],
[-0.62940758],
[ 0.00988613],
[-0.63420092],
[-0.63899427],
[-2.31843791],
[-1.7554278 ],
[ 0.51099267],
[-0.62461423],
[ 1.64659958],
```

```
[-0.62461423],
[ 2.21440304],
[ 1.07879613],
[-1.19241769],
[ 1.07879613],
[-0.62461423],
[ 1.07879613],
[-0.07598416],
[ 0.49181929],
[ 0.49181929],
[ 0.51099267],
[-0.63420092],
[ 1.07879613],
[-0.63420092],
[-0.05681078],
[-0.63420092],
[-0.61982089],
[-0.62461423],
[ 0.51099267],
[-0.05681078],
[-0.61982089],
[-0.63420092],
[ 1.07879613],
[-0.05681078],
[-0.03519826],
[ 0.51099267],
[-0.05681078],
[-1.18762434],
[-0.05681078],
[-0.63420092],
[-0.63420092],
[-0.05681078],
[ 0.50619933],
[ 0.50140598],
[-1.19241769],
[ 0.51099267],
[ 0.49661264],
[-0.06160413],
[-0.06639747],
[ 0.50619933],
[ 0.51099267],
[-0.62940758],
[-0.05681078],
[ 0.51099267],
[-0.62461423],
[-1.76501449],
[ 1.06441609],
[ 0.51099267],
[-0.62461423],
[ 0.51099267],
```

```
[-0.05681078],
[ 1.07879613],
[-1.19721103],
[-0.07119082],
[-0.61982089],
[-0.62940758],
[-1.19721103],
[ 1.07879613],
[-1.20200438],
[ 1.07879613],
[-0.05681078],
[-0.05681078],
[-0.62940758],
[-0.06639747],
[-0.63899427],
[ 0.51099267],
[ 1.07879613],
[ 1.07879613],
[-1.19241769],
[-0.06160413],
[ 0.51099267],
[-0.62940758],
[ 1.07879613],
[-0.62940758],
[ 0.51099267],
[-0.63420092],
[ 0.51099267],
[-0.63420092],
[ 2.21440304],
[-0.62940758],
[-0.62940758],
[ 1.07879613],
[ 1.07879613],
[ 0.51099267],
[ 0.51099267],
[-0.05681078],
[ 0.51099267],
[-0.05681078],
[-0.62940758],
[-0.63420092],
[ 1.07879613],
[-0.05681078],
[ 1.07400278],
[-0.06639747],
[ 0.51099267],
[ 1.07879613],
[ 0.51099267],
[-0.06639747],
[ 1.64659958],
[ 0.83834439],
```

```
        [-1.19241769],
        [-0.06160413],
        [ 0.51099267],
        [ 0.51099267],
        [ 0.51099267],
        [-0.63420092],
        [-0.62461423],
        [-0.63420092],
        [ 0.51099267],
        [-0.06639747],
        [-0.61982089],
        [ 1.64659958],
        [ 1.07879613],
        [-0.06639747],
        [-1.18762434],
        [-1.18762434],
        [-0.62940758],
        [-0.05681078],
        [ 1.64659958],
        [ 1.07879613],
        [-0.05681078],
        [-0.62940758],
        [-0.61072905],
        [-0.05681078],
        [ 0.51099267],
        [-0.06160413],
        [-0.62461423],
        [-0.05681078],
        [-0.05681078],
        [-0.06639747],
        [ 0.50619933],
        [-0.62940758],
        [-0.05681078],
        [-0.05681078],
        [-0.05681078],
        [-0.05681078],
        [ 0.51099267]])

from sklearn.metrics import r2_score

print(f"R2 Score : {r2_score(Y_test,y_pred)*100} % ")

R2 Score : 55.15103204321362 %

print(f"Mean absolute error: {np.mean(np.absolute(y_pred - Y_test))}
")#pred - actual

Mean absolute error: 0.4076871161048974

print("Residual sum of squares (MSE): %.2f" % np.mean((y_pred -
Y_test) **2))
```

```
Residual sum of squares (MSE): 0.33
```

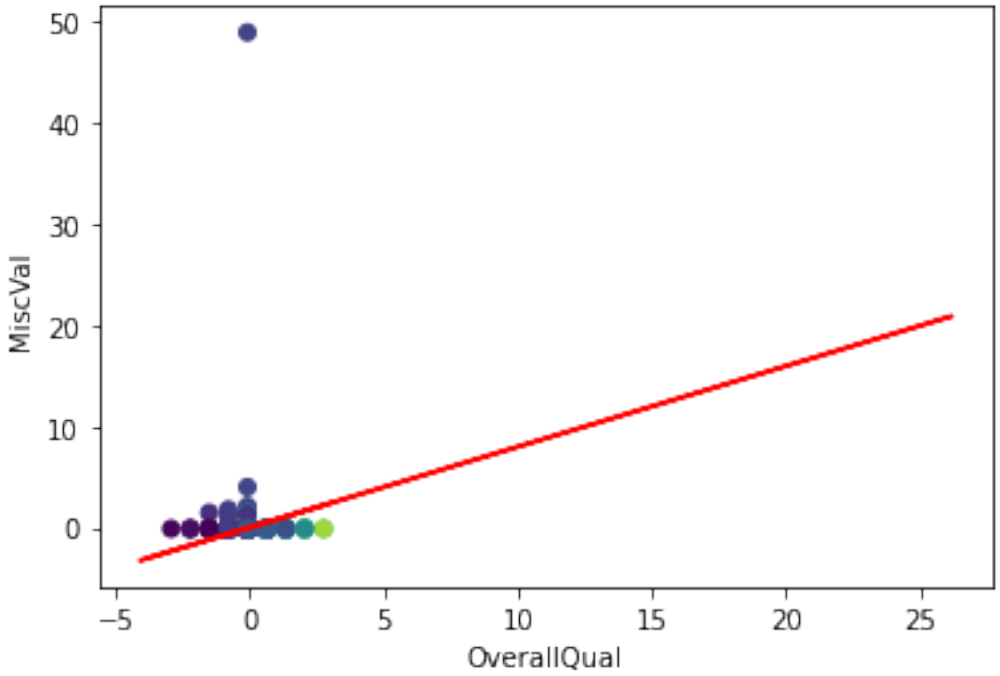
```
accuracy = print(f"accuracy : {r2_score(Y_test,y_pred)*100} % ")
```

```
accuracy : 55.15103204321362 %
```

Plot Outputs

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,2],c=Y_test)
plt.plot(X_train, regr.coef_[0][1]*X_train + regr.intercept_[0], 'r')
#y = mx+c
plt.xlabel("OverallQual")
plt.ylabel("MiscVal")
```

```
Text(0, 0.5, 'MiscVal')
```



Ans : The best possible accuracy is having the 2nd model that is R2 Score is 9.09%.

```
  # The lower value of R2 Score then it is a accurate model
```

Que.4 We are providing you churn dataset and we expect you to apply logistic regression on it and try to change the hyperparameters so that you can get the best possible acuracy

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt

churn_df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/ChurnData.csv")
churn_df.head()
```

```
   tenure   age  address  income   ed  employ  equip  callcard
wireless  \
0    11.0  33.0      7.0   136.0  5.0     5.0    0.0       1.0
1.0
1    33.0  33.0     12.0    33.0  2.0     0.0    0.0       0.0
0.0
2    23.0  30.0      9.0    30.0  1.0     2.0    0.0       0.0
0.0
3    38.0  35.0      5.0    76.0  2.0    10.0    1.0       1.0
1.0
4     7.0  35.0     14.0    80.0  2.0    15.0    0.0       1.0
0.0

   longmon  ...  pager  internet  callwait  confer  ebill  loglong
logtoll  \
0     4.40  ...    1.0       0.0       1.0     1.0    0.0    1.482
3.033
1     9.45  ...    0.0       0.0       0.0     0.0    0.0    2.246
3.240
2     6.30  ...    0.0       0.0       0.0     1.0    0.0    1.841
3.240
3     6.05  ...    1.0       1.0       1.0     1.0    1.0    1.800
3.807
4     7.10  ...    0.0       0.0       1.0     1.0    0.0    1.960
3.091

   lninc  custcat  churn
0  4.913      4.0    1.0
1  3.497      1.0    1.0
2  3.401      3.0    0.0
3  4.331      4.0    0.0
4  4.382      3.0    0.0

[5 rows x 28 columns]
```

Data Pre-Processing and Selection

```
churn_df = churn_df[['tenure', 'age', 'address', 'income', 'ed',
'employ', 'equip',   'callcard', 'wireless','churn']]
churn_df.head()
```

```
   tenure   age  address  income   ed  employ  equip  callcard
wireless  \
0    11.0  33.0      7.0   136.0  5.0     5.0    0.0       1.0
1.0
1    33.0  33.0     12.0    33.0  2.0     0.0    0.0       0.0
0.0
2    23.0  30.0      9.0    30.0  1.0     2.0    0.0       0.0
0.0
3    38.0  35.0      5.0    76.0  2.0    10.0    1.0       1.0
1.0
4     7.0  35.0     14.0    80.0  2.0    15.0    0.0       1.0
0.0

   churn
0    1.0
1    1.0
2    0.0
3    0.0
4    0.0
```

```
churn_df['churn'] = churn_df['churn'].astype('int')
churn_df.head()
```

```
   tenure   age  address  income   ed  employ  equip  callcard
wireless  \
0    11.0  33.0      7.0   136.0  5.0     5.0    0.0       1.0
1.0
1    33.0  33.0     12.0    33.0  2.0     0.0    0.0       0.0
0.0
2    23.0  30.0      9.0    30.0  1.0     2.0    0.0       0.0
0.0
3    38.0  35.0      5.0    76.0  2.0    10.0    1.0       1.0
1.0
4     7.0  35.0     14.0    80.0  2.0    15.0    0.0       1.0
0.0

   churn
0      1
1      1
2      0
3      0
4      0
```

```
churn_df.shape # No. of rows nd no. of cols
```

```
(200, 10)
```

```
churn_df.columns
```

```
Index(['tenure', 'age', 'address', 'income', 'ed', 'employ', 'equip',
       'callcard', 'wireless', 'churn'],
      dtype='object')
```

```
churn_df.info() #how many null/empty values present
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   tenure    200 non-null    float64
 1   age       200 non-null    float64
 2   address   200 non-null    float64
 3   income    200 non-null    float64
 4   ed        200 non-null    float64
 5   employ    200 non-null    float64
 6   equip     200 non-null    float64
 7   callcard  200 non-null    float64
 8   wireless  200 non-null    float64
 9   churn     200 non-null    int32  
dtypes: float64(9), int32(1)
memory usage: 15.0 KB
```

```
churn_df.isnull().sum()
```

```
tenure      0
age         0
address     0
income      0
ed          0
employ      0
equip       0
callcard    0
wireless    0
churn       0
dtype: int64
```

```
#independent variable
#asarray is used to concert columns to same data type to the array
X = np.asarray(churn_df[['tenure', 'age', 'address', 'income', 'ed',
'employ', 'equip']]) # Independent variable
X[0:1] # 0,1,2,3,4,5,6
```

```
array([[ 11.,  33.,   7., 136.,   5.,   5.,   0.]])
```

```
# Dependent variable
y = np.asarray(churn_df['churn']) # Dependent variable
y [0:10]
```

```
array([1, 1, 0, 0, 0, 0, 0, 0, 0, 0])
```

Also we Normalize the Dataset

```
# For the values are for 1 to 10 and some are the 10 to 100 range
that's why we have to convert that values in the range
X[0:5]
```

```
array([[ 11.,  33.,   7., 136.,   5.,   5.,   0.],
       [ 33.,  33.,  12.,  33.,   2.,   0.,   0.],
       [ 23.,  30.,   9.,  30.,   1.,   2.,   0.],
       [ 38.,  35.,   5.,  76.,   2.,  10.,   1.],
       [  7.,  35.,  14.,  80.,   2.,  15.,   0.]])
```

```
from sklearn import preprocessing
X = preprocessing.StandardScaler().fit(X).transform(X)
```

```
X[0:1]
```

```
array([[-1.13518441, -0.62595491, -0.4588971 ,  0.4751423 ,  1.6961288
,
        -0.58477841, -0.85972695]])
```

Train Test Dataset

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y,
test_size=0.2, random_state=200) # Trainingsize = 80%, Test = 20%
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (160, 7) (160,)
Test set: (40, 7) (40,)
```

Modeling (Logistic Regression with Scikit-learn)

Lets build our model using LogisticRegression from Scikit-learn package. This function implements logistic regression and can use different numerical optimizers to find parameters, including 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga' solvers. You can find extensive information about the pros and cons of these optimizers if you search it in internet.

```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(solver='sag')
LR.fit(X_train,y_train) # Training
LR
```

```
LogisticRegression(solver='sag')
```

```
yhat = LR.predict(X_test)#only questions passed and answers are saved
for evaluation
yhat[:5]
```

```
array([1, 0, 0, 0, 0])
```

```
yhat_prob = LR.predict_proba(X_test)
yhat_prob[:5]
```

```
array([[0.25580103, 0.74419897],
       [0.79835828, 0.20164172],
       [0.97099194, 0.02900806],
       [0.96104168, 0.03895832],
       [0.78135768, 0.21864232]])
```

Evaluation

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat) #actualvale,predvalue
```

```
0.75
```

```
churn_df
```

```
     tenure   age  address  income   ed  employ  equip  callcard
wireless  \
0      11.0  33.0      7.0   136.0  5.0     5.0    0.0       1.0
1.0
1      33.0  33.0     12.0    33.0  2.0     0.0    0.0       0.0
0.0
2      23.0  30.0      9.0    30.0  1.0     2.0    0.0       0.0
0.0
```

```
3      38.0  35.0      5.0    76.0  2.0    10.0    1.0       1.0
1.0
4       7.0  35.0     14.0    80.0  2.0    15.0    0.0       1.0
0.0
..      ...   ...      ...     ...  ...     ...    ...       ...
...
195    55.0  44.0     24.0    83.0  1.0    23.0    0.0       1.0
0.0
196    34.0  23.0      3.0    24.0  1.0     7.0    0.0       1.0
0.0
197     6.0  32.0     10.0    47.0  1.0    10.0    0.0       1.0
0.0
198    24.0  30.0      0.0    25.0  4.0     5.0    0.0       1.0
1.0
199    61.0  50.0     16.0   190.0  2.0    22.0    1.0       1.0
1.0

     churn
0        1
1        1
2        0
3        0
4        0
..     ...
195      0
196      0
197      0
198      1
199      0

[200 rows x 10 columns]
```

X_test

```
array([[-1.18150902e+00, -1.00927084e+00, -2.61522001e-01,
        -1.01476095e-03,  9.16299467e-01, -1.03245566e+00,
         1.16316000e+00],
       [-2.08692340e-01,  3.70666503e-01,  1.12010367e+00,
         5.37589130e-01,  9.16299467e-01,  3.10576093e-01,
        -8.59726954e-01],
       [ 9.95747358e-01,  1.44395110e+00,  8.24041023e-01,
        -4.77171824e-01, -6.43359200e-01,  1.87744647e+00,
        -8.59726954e-01],
       [ 1.69061641e+00,  2.59389889e+00,  3.58729236e+00,
        -4.77171824e-01, -6.43359200e-01, -4.72859097e-01,
        -8.59726954e-01],
       [ 1.61904490e-01, -1.31592358e+00, -9.52334835e-01,
        -3.83501582e-01, -6.43359200e-01, -6.96697723e-01,
        -8.59726954e-01],
       [ 9.03098150e-01,  8.30645618e-01,  1.02141612e+00,
```

```
  4.04889621e-01, -1.42318853e+00,  1.98936579e+00,
 -8.59726954e-01],
[ 4.39852113e-01, -1.46924996e+00, -9.52334835e-01,
 -3.05443047e-01,  9.16299467e-01, -1.14437497e+00,
  1.16316000e+00],
[ 1.69061641e+00,  2.59389889e+00,  2.50172933e+00,
 -3.28860608e-01, -1.42318853e+00,  3.78007479e+00,
 -8.59726954e-01],
[ 1.41266879e+00,  1.59727748e+00,  1.90960405e+00,
 -2.19578659e-01, -6.43359200e-01, -6.96697723e-01,
 -8.59726954e-01],
[ 1.08839657e+00,  1.06063518e+00,  1.71222895e+00,
 -2.58607926e-01,  1.36470133e-01,  1.20593060e+00,
 -8.59726954e-01],
[ 1.55164260e+00,  3.70666503e-01,  6.26665928e-01,
 -7.12674424e-02, -6.43359200e-01,  9.82091970e-01,
 -8.59726954e-01],
[-1.59843045e+00, -8.55944470e-01, -8.53647287e-01,
  4.67336448e-01,  9.16299467e-01, -8.08617036e-01,
  1.16316000e+00],
[-7.64587585e-01, -7.02618098e-01,  3.45406417e-02,
 -3.22381749e-02,  9.16299467e-01, -5.84778410e-01,
  1.16316000e+00],
[ 1.64429181e+00,  5.23992875e-01,  1.31747876e+00,
  1.66163203e+00,  1.36470133e-01,  9.82091970e-01,
 -8.59726954e-01],
[ 3.93527509e-01,  2.94003318e-01,  7.25353475e-01,
  1.00461334e-01, -1.42318853e+00,  1.20593060e+00,
 -8.59726954e-01],
[-9.96210604e-01, -1.16259721e+00, -1.05102238e+00,
 -3.91307435e-01, -6.43359200e-01, -1.14437497e+00,
 -8.59726954e-01],
[-5.79289170e-01,  6.77319247e-01, -1.05102238e+00,
  5.92230104e-01,  9.16299467e-01, -2.49020471e-01,
 -8.59726954e-01],
[ 1.50531800e+00,  8.30645618e-01,  5.27978380e-01,
  3.50248646e-01, -1.42318853e+00,  1.54168853e+00,
 -8.59726954e-01],
[ 4.39852113e-01, -1.08593403e+00, -8.53647287e-01,
 -2.82025487e-01,  1.36470133e-01, -9.20536348e-01,
 -8.59726954e-01],
[-1.08885981e+00, -1.31592358e+00, -9.52334835e-01,
 -4.38142556e-01, -1.42318853e+00, -1.14437497e+00,
 -8.59726954e-01],
[ 1.59796721e+00, -4.72628541e-01, -7.54959740e-01,
 -2.11772805e-01, -6.43359200e-01, -1.37101158e-01,
 -8.59726954e-01],
[ 1.69061641e+00,  1.06063518e+00,  1.21879121e+00,
  5.36262135e-02,  1.36470133e-01,  1.65360785e+00,
  1.16316000e+00],
```

```
[-8.10912189e-01, -5.49291726e-01, -7.54959740e-01,
 -2.58607926e-01, -6.43359200e-01,  4.22495406e-01,
 -8.59726954e-01],
[-1.36680743e+00, -9.32607656e-01, -7.54959740e-01,
 -4.38142556e-01, -6.43359200e-01, -5.84778410e-01,
 -8.59726954e-01],
[-1.27415823e+00, -1.46924996e+00, -8.53647287e-01,
 -3.91307435e-01,  9.16299467e-01, -1.14437497e+00,
 -8.59726954e-01],
[ 1.32001958e+00, -1.23926040e+00, -7.54959740e-01,
  6.79109254e-03,  1.36470133e-01, -9.20536348e-01,
  1.16316000e+00],
[ 3.47202905e-01, -9.32607656e-01, -7.54959740e-01,
 -3.28860608e-01, -1.42318853e+00,  3.10576093e-01,
 -8.59726954e-01],
[-4.40315359e-01, -8.55944470e-01, -7.54959740e-01,
  6.79109254e-03,  1.36470133e-01, -3.60939784e-01,
  1.16316000e+00],
[-2.33939249e-02, -5.49291726e-01, -4.58897097e-01,
  2.24027995e-02,  9.16299467e-01, -2.51818454e-02,
 -8.59726954e-01],
[-1.16043132e-01, -6.25954912e-01,  3.45406417e-02,
 -2.58607926e-01,  9.16299467e-01, -3.60939784e-01,
  1.16316000e+00],
[ 8.10448943e-01,  1.21396155e+00,  1.31747876e+00,
 -2.97637194e-01, -1.42318853e+00, -3.60939784e-01,
 -8.59726954e-01],
[ 3.47202905e-01,  1.40676946e-01, -8.53647287e-01,
 -1.57131831e-01,  9.16299467e-01,  8.70172658e-01,
  1.16316000e+00],
[ 7.64124339e-01,  6.40137602e-02,  5.27978380e-01,
 -3.75695728e-01,  1.36470133e-01, -2.49020471e-01,
 -8.59726954e-01],
[ 1.22737038e+00,  2.67056208e+00,  8.24041023e-01,
 -3.13248901e-01,  1.36470133e-01,  8.70172658e-01,
 -8.59726954e-01],
[ 5.32501320e-01, -8.93126115e-02,  4.29290832e-01,
  4.04889621e-01,  9.16299467e-01,  1.98656780e-01,
  1.16316000e+00],
[ 1.61904490e-01,  4.47329689e-01, -1.05102238e+00,
 -5.56557354e-02,  9.16299467e-01, -2.51818454e-02,
 -8.59726954e-01],
[ 1.69061641e+00,  1.52061429e+00,  2.20566669e+00,
 -1.10296710e-01, -1.42318853e+00, -2.49020471e-01,
 -8.59726954e-01],
[-1.62367736e-01, -1.08593403e+00, -8.53647287e-01,
  1.23878895e-01,  9.16299467e-01, -1.03245566e+00,
  1.16316000e+00],
[-1.22783362e+00, -1.26494257e-02,  3.45406417e-02,
 -2.82025487e-01,  9.16299467e-01, -8.08617036e-01,
```

```
         1.16316000e+00],
       [ 4.86176716e-01,  2.94003318e-01,  3.45406417e-02,
         1.62908162e-01,  1.36470133e-01,  7.58253345e-01,
         1.16316000e+00]])
```

```
X_test[:,1] #1 index col values
```

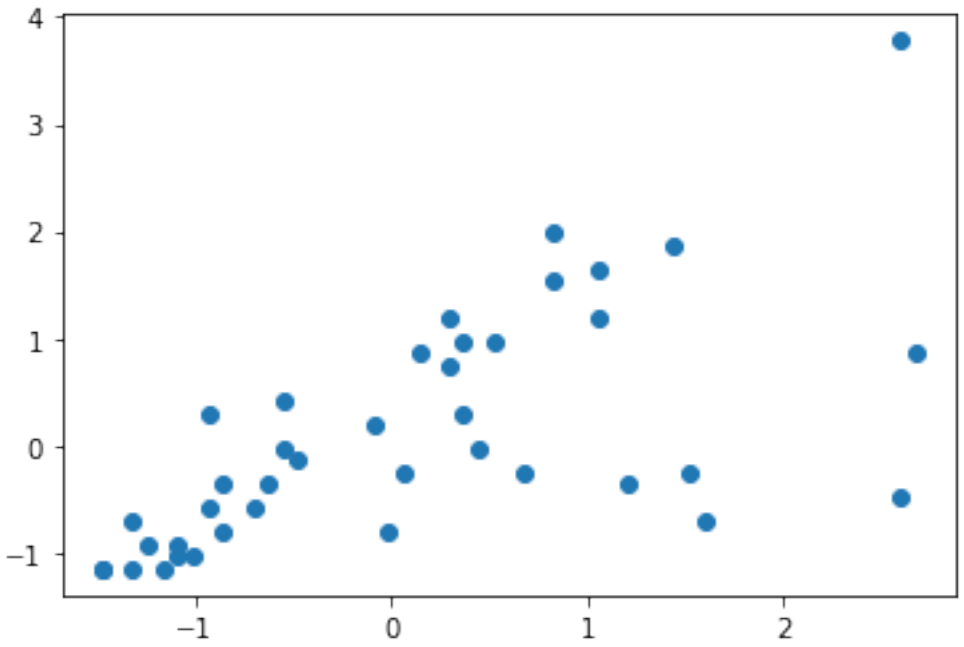
```
array([-1.00927084,  0.3706665 ,  1.4439511 ,  2.59389889, -
1.31592358,
        0.83064562, -1.46924996,  2.59389889,  1.59727748,
1.06063518,
        0.3706665 , -0.85594447, -0.7026181 ,  0.52399288,
0.29400332,
       -1.16259721,  0.67731925,  0.83064562, -1.08593403, -
1.31592358,
       -0.47262854,  1.06063518, -0.54929173, -0.93260766, -
1.46924996,
       -1.2392604 , -0.93260766, -0.85594447, -0.54929173, -
0.62595491,
        1.21396155,  0.14067695,  0.06401376,  2.67056208, -
0.08931261,
        0.44732969,  1.52061429, -1.08593403, -0.01264943,
0.29400332])
```

```
X_test[:,-2] #2nd last colunm values
```

```
array([-1.03245566,  0.31057609,  1.87744647, -0.4728591 , -
0.69669772,
        1.98936579, -1.14437497,  3.78007479, -0.69669772,
1.2059306 ,
        0.98209197, -0.80861704, -0.58477841,  0.98209197,
1.2059306 ,
       -1.14437497, -0.24902047,  1.54168853, -0.92053635, -
1.14437497,
       -0.13710116,  1.65360785,  0.42249541, -0.58477841, -
1.14437497,
       -0.92053635,  0.31057609, -0.36093978, -0.02518185, -
0.36093978,
       -0.36093978,  0.87017266, -0.24902047,  0.87017266,
0.19865678,
       -0.02518185, -0.24902047, -1.03245566, -0.80861704,
0.75825334])
```

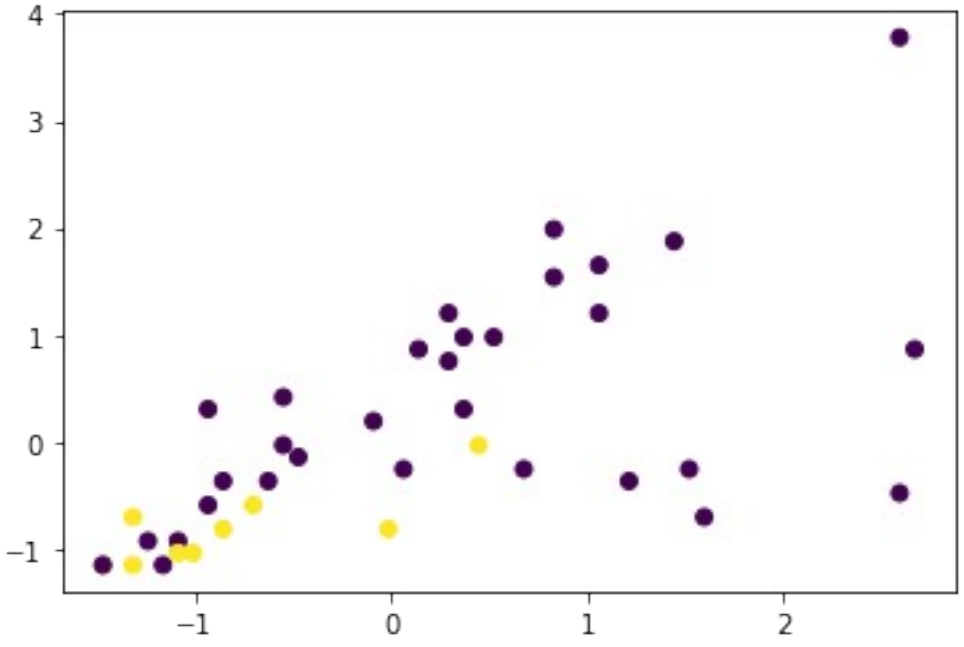
```
plt.scatter(X_test[:,1],X_test[:,-2])
```

```
<matplotlib.collections.PathCollection at 0x1c2aaec3550>
```



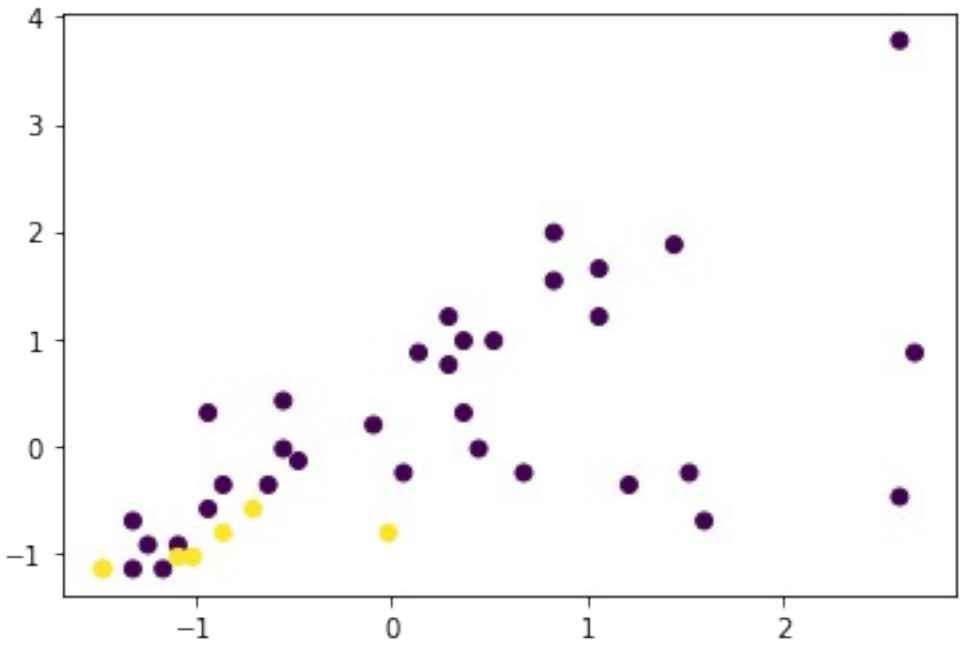

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c = y_test) #coloring based on
actual values
```

```
<matplotlib.collections.PathCollection at 0x1c2aafc2910>
```




```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c = yhat)
```

```
<matplotlib.collections.PathCollection at 0x1c2ab032430>
```




```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(solver='newton-cg')
LR.fit(X_train,y_train) # Training
LR
```

```
LogisticRegression(solver='newton-cg')
```

```
yhat = LR.predict(X_test)#only questions passed and answers are saved
for evaluation
yhat[:5]
```

```
array([1, 0, 0, 0, 0])
```

```
yhat_prob = LR.predict_proba(X_test)
yhat_prob[:5]
```

```
array([[0.25573477, 0.74426523],
       [0.79831414, 0.20168586],
       [0.97100876, 0.02899124],
       [0.96102166, 0.03897834],
       [0.78136144, 0.21863856]])
```

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat) #actualvale,predvalue
```

```
0.75
```

```
churn_df
```

```
     tenure   age  address  income   ed  employ  equip  callcard
wireless  \
0      11.0  33.0      7.0   136.0  5.0     5.0    0.0       1.0
```

```
1.0
1      33.0  33.0     12.0    33.0  2.0     0.0    0.0       0.0
0.0
2      23.0  30.0      9.0    30.0  1.0     2.0    0.0       0.0
0.0
3      38.0  35.0      5.0    76.0  2.0    10.0    1.0       1.0
1.0
4       7.0  35.0     14.0    80.0  2.0    15.0    0.0       1.0
0.0
..      ...   ...      ...     ...  ...     ...    ...       ...
...
195    55.0  44.0     24.0    83.0  1.0    23.0    0.0       1.0
0.0
196    34.0  23.0      3.0    24.0  1.0     7.0    0.0       1.0
0.0
197     6.0  32.0     10.0    47.0  1.0    10.0    0.0       1.0
0.0
198    24.0  30.0      0.0    25.0  4.0     5.0    0.0       1.0
1.0
199    61.0  50.0     16.0   190.0  2.0    22.0    1.0       1.0
1.0

     churn
0        1
1        1
2        0
3        0
4        0
..     ...
195      0
196      0
197      0
198      1
199      0

[200 rows x 10 columns]
```

X_test

```
array([[-1.18150902e+00, -1.00927084e+00, -2.61522001e-01,
        -1.01476095e-03,  9.16299467e-01, -1.03245566e+00,
         1.16316000e+00],
       [-2.08692340e-01,  3.70666503e-01,  1.12010367e+00,
         5.37589130e-01,  9.16299467e-01,  3.10576093e-01,
        -8.59726954e-01],
       [ 9.95747358e-01,  1.44395110e+00,  8.24041023e-01,
        -4.77171824e-01, -6.43359200e-01,  1.87744647e+00,
        -8.59726954e-01],
       [ 1.69061641e+00,  2.59389889e+00,  3.58729236e+00,
        -4.77171824e-01, -6.43359200e-01, -4.72859097e-01,
```

```
 -8.59726954e-01],
[ 1.61904490e-01, -1.31592358e+00, -9.52334835e-01,
 -3.83501582e-01, -6.43359200e-01, -6.96697723e-01,
 -8.59726954e-01],
[ 9.03098150e-01,  8.30645618e-01,  1.02141612e+00,
  4.04889621e-01, -1.42318853e+00,  1.98936579e+00,
 -8.59726954e-01],
[ 4.39852113e-01, -1.46924996e+00, -9.52334835e-01,
 -3.05443047e-01,  9.16299467e-01, -1.14437497e+00,
  1.16316000e+00],
[ 1.69061641e+00,  2.59389889e+00,  2.50172933e+00,
 -3.28860608e-01, -1.42318853e+00,  3.78007479e+00,
 -8.59726954e-01],
[ 1.41266879e+00,  1.59727748e+00,  1.90960405e+00,
 -2.19578659e-01, -6.43359200e-01, -6.96697723e-01,
 -8.59726954e-01],
[ 1.08839657e+00,  1.06063518e+00,  1.71222895e+00,
 -2.58607926e-01,  1.36470133e-01,  1.20593060e+00,
 -8.59726954e-01],
[ 1.55164260e+00,  3.70666503e-01,  6.26665928e-01,
 -7.12674424e-02, -6.43359200e-01,  9.82091970e-01,
 -8.59726954e-01],
[-1.59843045e+00, -8.55944470e-01, -8.53647287e-01,
  4.67336448e-01,  9.16299467e-01, -8.08617036e-01,
  1.16316000e+00],
[-7.64587585e-01, -7.02618098e-01,  3.45406417e-02,
 -3.22381749e-02,  9.16299467e-01, -5.84778410e-01,
  1.16316000e+00],
[ 1.64429181e+00,  5.23992875e-01,  1.31747876e+00,
  1.66163203e+00,  1.36470133e-01,  9.82091970e-01,
 -8.59726954e-01],
[ 3.93527509e-01,  2.94003318e-01,  7.25353475e-01,
  1.00461334e-01, -1.42318853e+00,  1.20593060e+00,
 -8.59726954e-01],
[-9.96210604e-01, -1.16259721e+00, -1.05102238e+00,
 -3.91307435e-01, -6.43359200e-01, -1.14437497e+00,
 -8.59726954e-01],
[-5.79289170e-01,  6.77319247e-01, -1.05102238e+00,
  5.92230104e-01,  9.16299467e-01, -2.49020471e-01,
 -8.59726954e-01],
[ 1.50531800e+00,  8.30645618e-01,  5.27978380e-01,
  3.50248646e-01, -1.42318853e+00,  1.54168853e+00,
 -8.59726954e-01],
[ 4.39852113e-01, -1.08593403e+00, -8.53647287e-01,
 -2.82025487e-01,  1.36470133e-01, -9.20536348e-01,
 -8.59726954e-01],
[-1.08885981e+00, -1.31592358e+00, -9.52334835e-01,
 -4.38142556e-01, -1.42318853e+00, -1.14437497e+00,
 -8.59726954e-01],
[ 1.59796721e+00, -4.72628541e-01, -7.54959740e-01,
```

```
 -2.11772805e-01, -6.43359200e-01, -1.37101158e-01,
 -8.59726954e-01],
[ 1.69061641e+00,  1.06063518e+00,  1.21879121e+00,
  5.36262135e-02,  1.36470133e-01,  1.65360785e+00,
  1.16316000e+00],
[-8.10912189e-01, -5.49291726e-01, -7.54959740e-01,
 -2.58607926e-01, -6.43359200e-01,  4.22495406e-01,
 -8.59726954e-01],
[-1.36680743e+00, -9.32607656e-01, -7.54959740e-01,
 -4.38142556e-01, -6.43359200e-01, -5.84778410e-01,
 -8.59726954e-01],
[-1.27415823e+00, -1.46924996e+00, -8.53647287e-01,
 -3.91307435e-01,  9.16299467e-01, -1.14437497e+00,
 -8.59726954e-01],
[ 1.32001958e+00, -1.23926040e+00, -7.54959740e-01,
  6.79109254e-03,  1.36470133e-01, -9.20536348e-01,
  1.16316000e+00],
[ 3.47202905e-01, -9.32607656e-01, -7.54959740e-01,
 -3.28860608e-01, -1.42318853e+00,  3.10576093e-01,
 -8.59726954e-01],
[-4.40315359e-01, -8.55944470e-01, -7.54959740e-01,
  6.79109254e-03,  1.36470133e-01, -3.60939784e-01,
  1.16316000e+00],
[-2.33939249e-02, -5.49291726e-01, -4.58897097e-01,
  2.24027995e-02,  9.16299467e-01, -2.51818454e-02,
 -8.59726954e-01],
[-1.16043132e-01, -6.25954912e-01,  3.45406417e-02,
 -2.58607926e-01,  9.16299467e-01, -3.60939784e-01,
  1.16316000e+00],
[ 8.10448943e-01,  1.21396155e+00,  1.31747876e+00,
 -2.97637194e-01, -1.42318853e+00, -3.60939784e-01,
 -8.59726954e-01],
[ 3.47202905e-01,  1.40676946e-01, -8.53647287e-01,
 -1.57131831e-01,  9.16299467e-01,  8.70172658e-01,
  1.16316000e+00],
[ 7.64124339e-01,  6.40137602e-02,  5.27978380e-01,
 -3.75695728e-01,  1.36470133e-01, -2.49020471e-01,
 -8.59726954e-01],
[ 1.22737038e+00,  2.67056208e+00,  8.24041023e-01,
 -3.13248901e-01,  1.36470133e-01,  8.70172658e-01,
 -8.59726954e-01],
[ 5.32501320e-01, -8.93126115e-02,  4.29290832e-01,
  4.04889621e-01,  9.16299467e-01,  1.98656780e-01,
  1.16316000e+00],
[ 1.61904490e-01,  4.47329689e-01, -1.05102238e+00,
 -5.56557354e-02,  9.16299467e-01, -2.51818454e-02,
 -8.59726954e-01],
[ 1.69061641e+00,  1.52061429e+00,  2.20566669e+00,
 -1.10296710e-01, -1.42318853e+00, -2.49020471e-01,
 -8.59726954e-01],
```

```
       [-1.62367736e-01, -1.08593403e+00, -8.53647287e-01,
         1.23878895e-01,  9.16299467e-01, -1.03245566e+00,
         1.16316000e+00],
       [-1.22783362e+00, -1.26494257e-02,  3.45406417e-02,
        -2.82025487e-01,  9.16299467e-01, -8.08617036e-01,
         1.16316000e+00],
       [ 4.86176716e-01,  2.94003318e-01,  3.45406417e-02,
         1.62908162e-01,  1.36470133e-01,  7.58253345e-01,
         1.16316000e+00]])
```

```
X_test[:,1] #1 index col values
```

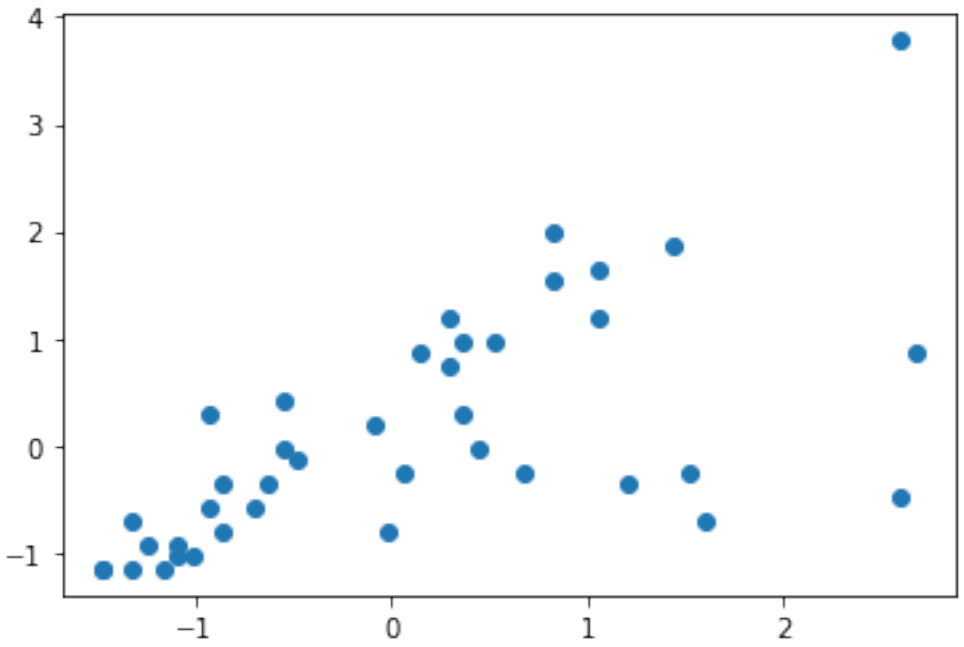
```
array([-1.00927084,  0.3706665 ,  1.4439511 ,  2.59389889, -
1.31592358,
        0.83064562, -1.46924996,  2.59389889,  1.59727748,
1.06063518,
        0.3706665 , -0.85594447, -0.7026181 ,  0.52399288,
0.29400332,
       -1.16259721,  0.67731925,  0.83064562, -1.08593403, -
1.31592358,
       -0.47262854,  1.06063518, -0.54929173, -0.93260766, -
1.46924996,
       -1.2392604 , -0.93260766, -0.85594447, -0.54929173, -
0.62595491,
        1.21396155,  0.14067695,  0.06401376,  2.67056208, -
0.08931261,
        0.44732969,  1.52061429, -1.08593403, -0.01264943,
0.29400332])
```

```
X_test[:,-2] #2nd last colunm values
```

```
array([-1.03245566,  0.31057609,  1.87744647, -0.4728591 , -
0.69669772,
        1.98936579, -1.14437497,  3.78007479, -0.69669772,
1.2059306 ,
        0.98209197, -0.80861704, -0.58477841,  0.98209197,
1.2059306 ,
       -1.14437497, -0.24902047,  1.54168853, -0.92053635, -
1.14437497,
       -0.13710116,  1.65360785,  0.42249541, -0.58477841, -
1.14437497,
       -0.92053635,  0.31057609, -0.36093978, -0.02518185, -
0.36093978,
       -0.36093978,  0.87017266, -0.24902047,  0.87017266,
0.19865678,
       -0.02518185, -0.24902047, -1.03245566, -0.80861704,
0.75825334])
```

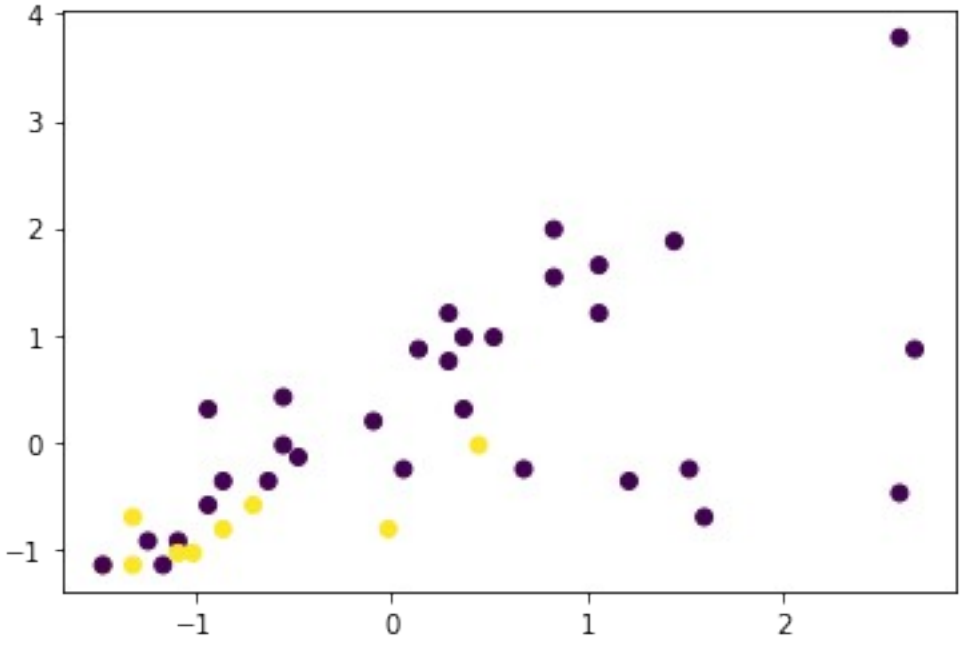
```
plt.scatter(X_test[:,1],X_test[:,-2])
```

```
<matplotlib.collections.PathCollection at 0x1c2ab094e80>
```



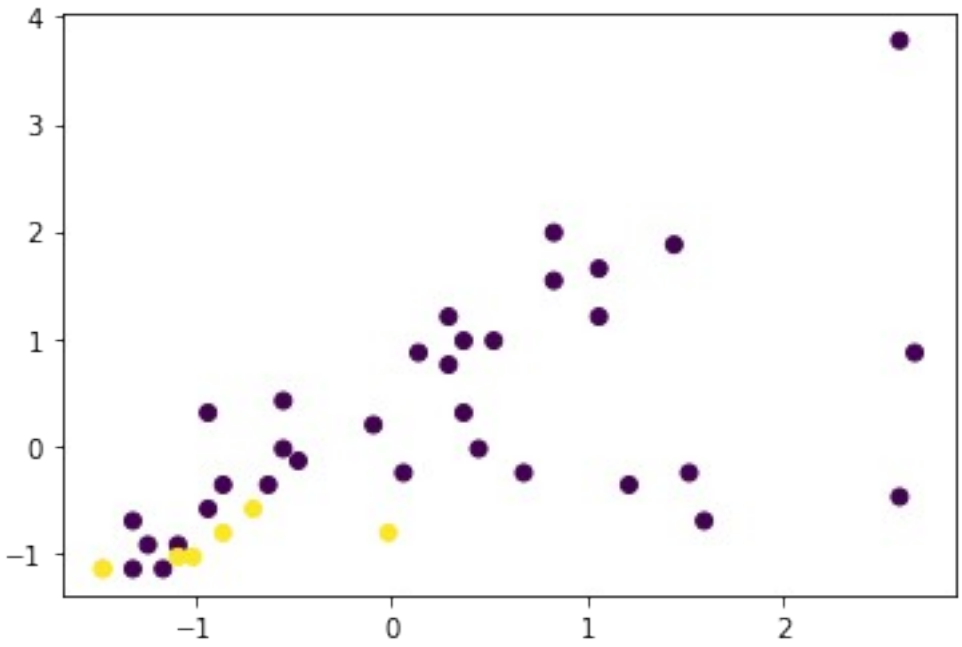

```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c = y_test) #coloring based on
actual values
```

```
<matplotlib.collections.PathCollection at 0x1c2ab0fc1f0>
```




```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c = yhat)
```

```
<matplotlib.collections.PathCollection at 0x1c2ab1556d0>
```




```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(solver='saga')
LR.fit(X_train,y_train) # Training
LR
```

```
LogisticRegression(solver='saga')
```

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat) #actualvale,predvalue
```

```
0.75
```

```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(solver='lbfgs')
LR.fit(X_train,y_train) # Training
LR
```

```
LogisticRegression()
```

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat) #actualvale,predvalue
```

```
0.75
```

```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(solver='liblinear')
LR.fit(X_train,y_train) # Training
LR
```

```
LogisticRegression(solver='liblinear')
```

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat) #actualvale,predvalue
```

```
0.75
```

Ans : The Best possible accuracy will be 75%

Que 5 : We are providing you the cell dataset and we expect you to use all the independent variables for creating the SVM machine learning model and change the hyperparameters so that you can get the best accuracy

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

```
cell_df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/cell_samples.csv")
cell_df.head()
```

```
        ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize
BareNuc  \
0  1000025      5         1          1        1            2       1

1  1002945      5         4          4        5            7      10

2  1015425      3         1          1        1            2       2

3  1016277      6         8          8        1            3       4

4  1017023      4         1          1        3            2       1


   BlandChrom  NormNucl  Mit  Class
0           3         1    1      2
1           3         2    1      2
2           3         1    1      2
3           3         7    1      2
4           3         1    1      2
```

```
cell_df.tail()
```

```
         ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize BareNuc
\
694  776715      3         1          1        1            3       2
```

```
695  841769      2          1           1         1             2       1

696  888820      5         10          10         3             7       3

697  897471      4          8           6         4             3       4

698  897471      4          8           8         5             4       5


     BlandChrom  NormNucl  Mit  Class
694           1         1    1      2
695           1         1    1      2
696           8        10    2      4
697          10         6    1      4
698          10         4    1      4

cell_df

          ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize BareNuc
\
0    1000025      5         1          1         1             2       1

1    1002945      5         4          4         5             7      10

2    1015425      3         1          1         1             2       2

3    1016277      6         8          8         1             3       4

4    1017023      4         1          1         3             2       1

..       ...    ...       ...        ...       ...           ...     ...

694   776715      3         1          1         1             3       2

695   841769      2         1          1         1             2       1

696   888820      5        10         10         3             7       3

697   897471      4         8          6         4             3       4

698   897471      4         8          8         5             4       5


     BlandChrom  NormNucl  Mit  Class
0             3         1    1      2
1             3         2    1      2
2             3         1    1      2
3             3         7    1      2
```

```
4             3         1    1      2
..          ...       ...  ...    ...
694           1         1    1      2
695           1         1    1      2
696           8        10    2      4
697          10         6    1      4
698          10         4    1      4

[699 rows x 11 columns]
```

Data pre-processing and selection

```
cell_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699 entries, 0 to 698
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   ID           699 non-null    int64
 1   Clump        699 non-null    int64
 2   UnifSize     699 non-null    int64
 3   UnifShape    699 non-null    int64
 4   MargAdh      699 non-null    int64
 5   SingEpiSize  699 non-null    int64
 6   BareNuc      699 non-null    object
 7   BlandChrom   699 non-null    int64
 8   NormNucl     699 non-null    int64
 9   Mit          699 non-null    int64
 10  Class        699 non-null    int64
dtypes: int64(10), object(1)
memory usage: 60.2+ KB
```

```
cell_df.describe()
```

```
                ID       Clump    UnifSize   UnifShape     MargAdh  \
count  6.990000e+02  699.000000  699.000000  699.000000  699.000000
mean   1.071704e+06    4.417740    3.134478    3.207439    2.806867
std    6.170957e+05    2.815741    3.051459    2.971913    2.855379
min    6.163400e+04    1.000000    1.000000    1.000000    1.000000
25%    8.706885e+05    2.000000    1.000000    1.000000    1.000000
50%    1.171710e+06    4.000000    1.000000    1.000000    1.000000
75%    1.238298e+06    6.000000    5.000000    5.000000    4.000000
max    1.345435e+07   10.000000   10.000000   10.000000   10.000000

       SingEpiSize  BlandChrom    NormNucl         Mit       Class
count   699.000000  699.000000  699.000000  699.000000  699.000000
mean      3.216023    3.437768    2.866953    1.589413    2.689557
std       2.214300    2.438364    3.053634    1.715078    0.951273
min       1.000000    1.000000    1.000000    1.000000    2.000000
```

```
25%       2.000000    2.000000    1.000000    1.000000    2.000000
50%       2.000000    3.000000    1.000000    1.000000    2.000000
75%       4.000000    5.000000    4.000000    1.000000    4.000000
max      10.000000   10.000000   10.000000   10.000000    4.000000
```

```
cell_df.drop('BareNuc',axis = 1,inplace = True)
```

```
cell_df
```

```
          ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize
BlandChrom  \
0    1000025      5         1          1        1            2
3
1    1002945      5         4          4        5            7
3
2    1015425      3         1          1        1            2
3
3    1016277      6         8          8        1            3
3
4    1017023      4         1          1        3            2
3
..       ...    ...       ...        ...      ...          ...
...
694   776715      3         1          1        1            3
1
695   841769      2         1          1        1            2
1
696   888820      5        10         10        3            7
8
697   897471      4         8          6        4            3
10
698   897471      4         8          8        5            4
10

     NormNucl  Mit  Class
0           1    1      2
1           2    1      2
2           1    1      2
3           7    1      2
4           1    1      2
..        ...  ...    ...
694         1    1      2
695         1    1      2
696        10    2      4
697         6    1      4
698         4    1      4

[699 rows x 10 columns]
```

```
feature_df = cell_df[['Clump', 'UnifSize', 'UnifShape', 'MargAdh',
'SingEpiSize', 'BlandChrom', 'NormNucl', 'Mit']]
feature_df # Independent Variable
```

```
     Clump  UnifSize  UnifShape  MargAdh  SingEpiSize  BlandChrom
NormNucl  \
0        5         1          1        1            2           3
1
1        5         4          4        5            7           3
2
2        3         1          1        1            2           3
1
3        6         8          8        1            3           3
7
4        4         1          1        3            2           3
1
..     ...       ...        ...      ...          ...         ...
...
694      3         1          1        1            3           1
1
695      2         1          1        1            2           1
1
696      5        10         10        3            7           8
10
697      4         8          6        4            3          10
6
698      4         8          8        5            4          10
4

     Mit
0      1
1      1
2      1
3      1
4      1
..   ...
694    1
695    1
696    2
697    1
698    1

[699 rows x 8 columns]
```

```
feature_df.dtypes
```

```
Clump          int64
UnifSize       int64
UnifShape      int64
MargAdh        int64
SingEpiSize    int64
```

```
BlandChrom     int64
NormNucl       int64
Mit            int64
dtype: object
```

```
X = np.asarray(feature_df) # Independet variable independent variable
array got created
X[0:5] # show me elements from zeroth row to 5th row
```

```
array([[5, 1, 1, 1, 2, 3, 1, 1],
       [5, 4, 4, 5, 7, 3, 2, 1],
       [3, 1, 1, 1, 2, 3, 1, 1],
       [6, 8, 8, 1, 3, 3, 7, 1],
       [4, 1, 1, 3, 2, 3, 1, 1]], dtype=int64)
```

```
cell_df['Class'] = cell_df['Class'].astype('int')
y = np.asarray(cell_df['Class']) # Dependent variable
y [0:5]
```

```
array([2, 2, 2, 2, 2])
```

Train/Test dataset

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=50)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (559, 8) (559,)
Test set: (140, 8) (140,)
```

Modeling (SVM with Scikit-learn)

The SVM algorithm offers a choice of kernel functions for performing its processing. Basically, mapping data into a higher dimensional space is called kernelling. The mathematical function used for the transformation is known as the kernel function, and can be of different types, such as:

1.Linear

2.Polynomial

3.Radial basis function (RBF)

4.Sigmoid

Each of these functions has its characteristics, its pros and cons, and its equation, but as there's no easy way of knowing which function performs best with any given dataset, we usually choose different functions in turn and compare the results. Let's just use the default, RBF (Radial Basis Function) for this lab.

```
from sklearn import svm
clf = svm.SVC(kernel='poly')
clf.fit(X_train, y_train) # Question and Answers
```

```
SVC(kernel='poly')
```

```
yhat = clf.predict(X_test) # Question
yhat [0:5]
```

```
array([2, 2, 2, 2, 2])
```

Evaluation

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat, average='weighted')
```

```
0.9425054112554112
```

```
# write your code here

clf2 = svm.SVC(kernel='rbf')
clf2.fit(X_train, y_train)
yhat2 = clf2.predict(X_test)
print("Avg F1-score: %.4f" % f1_score(y_test, yhat2,
average="weighted"))
```

```
Avg F1-score: 0.9714
```

```
# write your code here

clf2 = svm.SVC(kernel='linear')
clf2.fit(X_train, y_train)
yhat2 = clf2.predict(X_test)
print("Avg F1-score: %.4f" % f1_score(y_test, yhat2,
average="weighted"))
```

```
Avg F1-score: 0.9644
```

```
 cell_df
```

```
          ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize
BlandChrom  \
0    1000025      5         1          1        1            2
3
1    1002945      5         4          4        5            7
3
2    1015425      3         1          1        1            2
3
3    1016277      6         8          8        1            3
3
4    1017023      4         1          1        3            2
3
..       ...    ...       ...        ...      ...          ...
...
694   776715      3         1          1        1            3
1
695   841769      2         1          1        1            2
1
696   888820      5        10         10        3            7
8
697   897471      4         8          6        4            3
10
698   897471      4         8          8        5            4
10

     NormNucl  Mit  Class
0           1    1      2
```

```
1           2    1      2
2           1    1      2
3           7    1      2
4           1    1      2
..        ...  ...    ...
694         1    1      2
695         1    1      2
696        10    2      4
697         6    1      4
698         4    1      4

[699 rows x 10 columns]
```

```
X_test
```

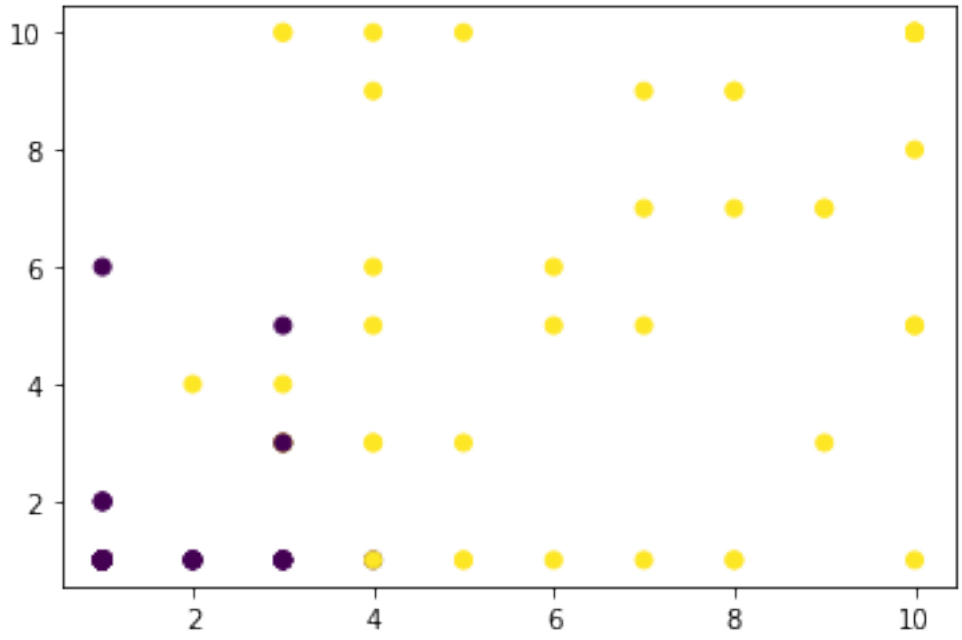
```
array([[4, 1, 1, ..., 2, 1, 1],
       [2, 3, 1, ..., 1, 1, 1],
       [5, 3, 1, ..., 2, 1, 1],
       ...,
       [5, 8, 7, ..., 5, 7, 1],
       [2, 1, 1, ..., 2, 1, 1],
       [1, 2, 3, ..., 1, 1, 1]], dtype=int64)
```

```
X_test.shape
```

```
(140, 8)
```

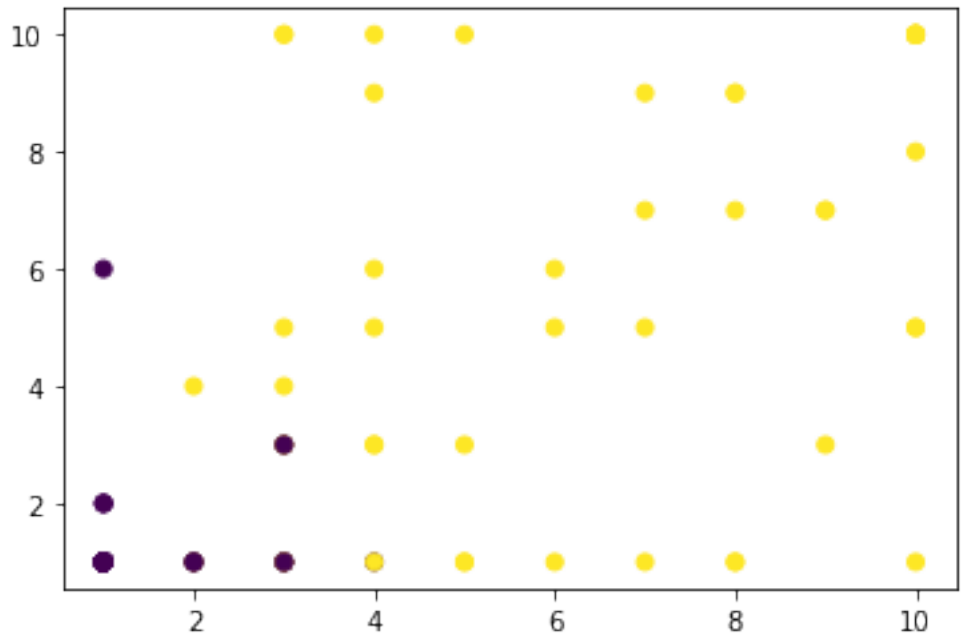
```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c=y_test)
```

```
<matplotlib.collections.PathCollection at 0x1c2ab1e0190>
```



```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c=yhat2)
```

```
<matplotlib.collections.PathCollection at 0x1c2ab222ee0>
```




```
 cell_df
```

```
          ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize
BlandChrom  \
0    1000025      5         1          1        1            2
3
1    1002945      5         4          4        5            7
3
2    1015425      3         1          1        1            2
3
3    1016277      6         8          8        1            3
3
4    1017023      4         1          1        3            2
3
..       ...    ...       ...        ...      ...          ...
...
694   776715      3         1          1        1            3
1
695   841769      2         1          1        1            2
1
696   888820      5        10         10        3            7
8
697   897471      4         8          6        4            3
10
698   897471      4         8          8        5            4
10
```

```
     NormNucl  Mit  Class
0           1    1      2
1           2    1      2
2           1    1      2
3           7    1      2
4           1    1      2
..        ...  ...    ...
694         1    1      2
695         1    1      2
696        10    2      4
697         6    1      4
698         4    1      4

[699 rows x 10 columns]
```

```
clf2.predict([[6,13,13,5,7,3,2,4]])
```

```
array([4])
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=50)
from sklearn import svm
clf = svm.SVC(kernel='poly')
clf.fit(X_train, y_train) #Training Model
training_pred = clf.predict(X_train)
print(f"training accuracy is {f1_score(y_train, training_pred,
average='weighted') }")
```

```
training accuracy is 0.9676813490496357
```

```
yhat = clf.predict(X_test) #final
from sklearn.metrics import f1_score

print(f"testing accuracy is {f1_score(y_test, yhat,
average='weighted')  }")
```

```
testing accuracy is 0.9425054112554112
```

Best accuracy can be give the hyperparameter is rbf (Radial Basis Function) = 97.14 %

Que 6 : Take the same cell Dataset and instead of SVM apply logistic regression in it.

```
import pandas as pd
import numpy as np
```

```
from sklearn import preprocessing
import matplotlib.pyplot as plt

cell_df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/cell_samples.csv")
cell_df.head()
```

```
        ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize
BareNuc  \
0  1000025      5         1          1        1            2       1

1  1002945      5         4          4        5            7      10

2  1015425      3         1          1        1            2       2

3  1016277      6         8          8        1            3       4

4  1017023      4         1          1        3            2       1


   BlandChrom  NormNucl  Mit  Class
0           3         1    1      2
1           3         2    1      2
2           3         1    1      2
3           3         7    1      2
4           3         1    1      2
```

```
cell_df.tail()
```

```
         ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize BareNuc
\
694  776715      3         1          1        1            3       2

695  841769      2         1          1        1            2       1

696  888820      5        10         10        3            7       3

697  897471      4         8          6        4            3       4

698  897471      4         8          8        5            4       5


     BlandChrom  NormNucl  Mit  Class
694           1         1    1      2
695           1         1    1      2
696           8        10    2      4
697          10         6    1      4
698          10         4    1      4
```

```
set(cell_df['Class'])
```

```
{2, 4}
```

```
cell_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699 entries, 0 to 698
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   ID           699 non-null    int64 
 1   Clump        699 non-null    int64 
 2   UnifSize     699 non-null    int64 
 3   UnifShape    699 non-null    int64 
 4   MargAdh      699 non-null    int64 
 5   SingEpiSize  699 non-null    int64 
 6   BareNuc      699 non-null    object
 7   BlandChrom   699 non-null    int64 
 8   NormNucl     699 non-null    int64 
 9   Mit          699 non-null    int64 
 10  Class        699 non-null    int64 
dtypes: int64(10), object(1)
memory usage: 60.2+ KB
```

```
cell_df.isnull().sum()
```

```
ID             0
Clump          0
UnifSize       0
UnifShape      0
MargAdh        0
SingEpiSize    0
BareNuc        0
BlandChrom     0
NormNucl       0
Mit            0
Class          0
dtype: int64
```

```
cell_df.drop('BareNuc',axis = 1,inplace = True)
```

```
cell_df
```

```
          ID  Clump  UnifSize  UnifShape  MargAdh  SingEpiSize
BlandChrom  \
0    1000025      5         1          1        1            2
3
1    1002945      5         4          4        5            7
3
2    1015425      3         1          1        1            2
3
3    1016277      6         8          8        1            3
3
```

```
4    1017023      4         1          1        3            2
3
..       ...    ...       ...        ...      ...          ...
...
694   776715      3         1          1        1            3
1
695   841769      2         1          1        1            2
1
696   888820      5        10         10        3            7
8
697   897471      4         8          6        4            3
10
698   897471      4         8          8        5            4
10

     NormNucl  Mit  Class
0           1    1      2
1           2    1      2
2           1    1      2
3           7    1      2
4           1    1      2
..        ...  ...    ...
694         1    1      2
695         1    1      2
696        10    2      4
697         6    1      4
698         4    1      4

[699 rows x 10 columns]
```

```
cell_df =
cell_df[["Clump","UnifSize","UnifShape","MargAdh","SingEpiSize","Bland
Chrom","NormNucl","Mit","Class"]]
cell_df.head()
```

```
   Clump  UnifSize  UnifShape  MargAdh  SingEpiSize  BlandChrom
NormNucl  \
0      5         1          1        1            2           3
1
1      5         4          4        5            7           3
2
2      3         1          1        1            2           3
1
3      6         8          8        1            3           3
7
4      4         1          1        3            2           3
1

   Mit  Class
0    1      2
```

```
1    1      2
2    1      2
3    1      2
4    1      2
```

```
cell_df.tail()
```

```
     Clump  UnifSize  UnifShape  MargAdh  SingEpiSize  BlandChrom
NormNucl  \
694      3         1          1        1            3           1
1
695      2         1          1        1            2           1
1
696      5        10         10        3            7           8
10
697      4         8          6        4            3          10
6
698      4         8          8        5            4          10
4

     Mit  Class
694    1      2
695    1      2
696    2      4
697    1      4
698    1      4
```

```
cell_df.shape
```

```
(699, 9)
```

```
cell_df.columns
```

```
Index(['Clump', 'UnifSize', 'UnifShape', 'MargAdh', 'SingEpiSize',
       'BlandChrom', 'NormNucl', 'Mit', 'Class'],
      dtype='object')
```

```
# Independent variable
# asarray is used to concert columns to same data type to the array
X =
np.asarray(cell_df[["Clump","UnifSize","UnifShape","MargAdh","SingEpiS
ize","BlandChrom","NormNucl","Mit"]]) # independent variable
X[0:1]#0,1,2,3,4
```

```
array([[5, 1, 1, 1, 2, 3, 1, 1]], dtype=int64)
```

```
# Dependent variable
y = np.asarray(cell_df['Class']) #dependent variable
y [0:9]
```

```
array([2, 2, 2, 2, 2, 4, 2, 2, 2], dtype=int64)
```

```
X[0:5]
```

```
array([[5, 1, 1, 1, 2, 3, 1, 1],
       [5, 4, 4, 5, 7, 3, 2, 1],
       [3, 1, 1, 1, 2, 3, 1, 1],
       [6, 8, 8, 1, 3, 3, 7, 1],
       [4, 1, 1, 3, 2, 3, 1, 1]], dtype=int64)
```

Train/Test dataset

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y,
test_size=0.2, random_state=200) # trainingsize = 80%  test = 20%
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (559, 8) (559,)
Test set: (140, 8) (140,)
```

Modeling (Logistic Regression with Scikit-learn)

Lets build our model using LogisticRegression from Scikit-learn package. This function implements logistic regression and can use different numerical optimizers to find parameters, including 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga' solvers. You can find extensive information about the pros and cons of these optimizers if you search it in internet.

```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(solver='saga')
LR.fit(X_train,y_train) # training
LR
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\linear_model\
_sag.py:352: ConvergenceWarning: The max_iter was reached which means
the coef_ did not converge
  warnings.warn(
```

```
LogisticRegression(solver='saga')
```

```
yhat = LR.predict(X_test)# only questions passed and answers are saved
for evaluation
yhat[:5]
```

```
array([4, 2, 4, 4, 2], dtype=int64)
```

```
yhat_prob = LR.predict_proba(X_test)
yhat_prob[:5]
```

```
array([[0.20679859, 0.79320141],
       [0.68021412, 0.31978588],
       [0.21399263, 0.78600737],
       [0.00104075, 0.99895925],
       [0.52621739, 0.47378261]])
```

Evaluation

```
from sklearn.metrics import f1_score
f1_score(y_test, yhat, average='weighted') # actualvalue,predvalue
```

```
0.9498499911759516
```

```
cell_df
```

```
     Clump  UnifSize  UnifShape  MargAdh  SingEpiSize  BlandChrom
NormNucl  \
0        5         1          1        1            2           3
1
1        5         4          4        5            7           3
2
2        3         1          1        1            2           3
1
3        6         8          8        1            3           3
7
4        4         1          1        3            2           3
1
..     ...       ...        ...      ...          ...         ...
...
694      3         1          1        1            3           1
1
695      2         1          1        1            2           1
1
696      5        10         10        3            7           8
10
697      4         8          6        4            3          10
6
698      4         8          8        5            4          10
4

     Mit  Class
0      1      2
1      1      2
2      1      2
3      1      2
4      1      2
..   ...    ...
694    1      2
```

```
695    1      2
696    2      4
697    1      4
698    1      4

[699 rows x 9 columns]
```

```
X_test
```

```
array([[10,  4,  3, ...,  5,  3,  2],
       [ 5,  1,  2, ...,  2,  1,  1],
       [10,  4,  2, ...,  4,  3, 10],
       ...,
       [ 4,  3,  3, ...,  3,  3,  1],
       [ 5,  5,  5, ...,  4,  3,  1],
       [10,  5,  7, ...,  8,  9,  1]], dtype=int64)
```

```
X_test[:,1] #1 index col values .
```

```
array([ 4,  1,  4,  7,  3,  1,  1,  8,  1, 10,  1,  5,  1,  1,  1,  3,
1,
        3,  8,  1,  1,  1,  1,  1,  1,  1,  7,  9,  1,  1,  1,  1,  2,
1,
        1,  9, 10,  1,  1,  1,  5,  1, 10,  6,  1, 10, 10,  3, 10,  1,
1,
        1,  1,  1,  3,  1,  5,  5,  1,  8,  1,  1,  7,  5,  1,  1,  2,
1,
        4,  1,  7,  4,  1,  1,  3,  1,  4,  1,  1,  1,  1,  4,  1,  5,
1,
        1,  2,  1,  1,  6,  1,  5,  1,  1,  1,  1,  1,  1,  1,  1,  1,
2,
        1,  1,  1,  1,  2,  1, 10,  1,  4,  1,  7,  3,  4,  1,  1,  1,
6,
        3,  6,  1,  1,  1,  1, 10, 10,  1,  1,  6,  8,  1,  1,  1,  1,
10,
        2,  3,  5,  5], dtype=int64)
```

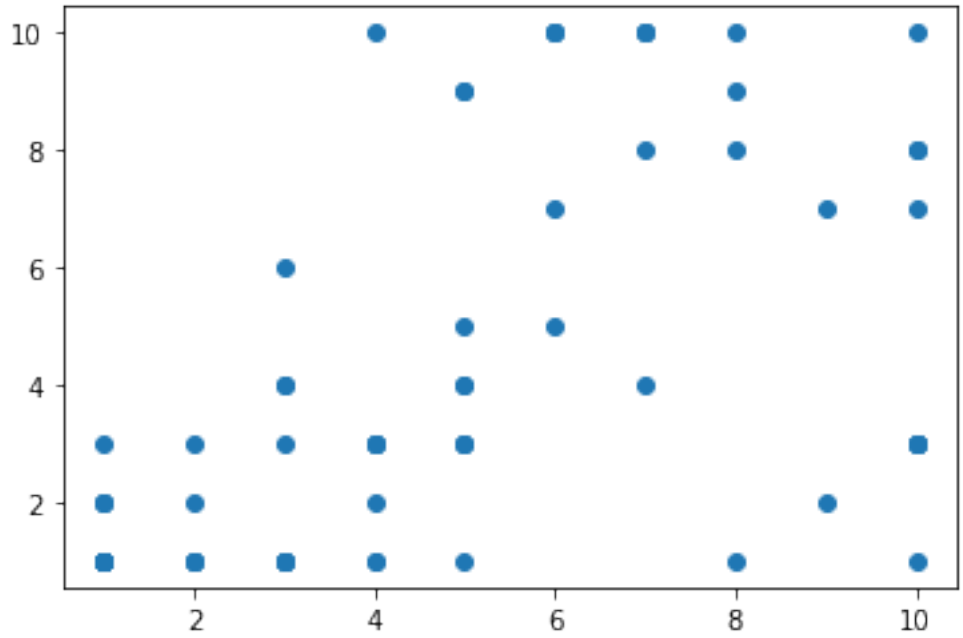
```
X_test[:,-2] #2nd last colunm values
```

```
array([ 3,  1,  3, 10,  1,  1,  1,  1,  1,  3,  1,  4,  1,  1,  1,  6,
1,
        1, 10,  1,  1,  1,  1,  1,  1,  1,  4,  2,  1,  1,  1,  1,  3,
2,
        1,  7,  3,  1,  1,  1,  1,  1,  3,  7,  1,  7, 10,  4,  1,  1,
1,
        1,  1,  1,  1,  2,  4,  5,  1,  8,  1,  1,  8,  3,  1,  1,  2,
1,
        3,  1, 10,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  3,  1,  9,
1,
        1,  1,  1,  1, 10,  1,  3,  1,  2,  1,  1,  1,  1,  1,  1,  1,
1,
```

```
        1,  1,  3,  1,  1,  1,  8,  1, 10,  1, 10,  4,  2,  1,  1,  1,
5,
        1, 10,  1,  1,  1,  1,  8,  3,  1,  1, 10,  9,  1,  1,  1,  1,
3,
        1,  3,  3,  9], dtype=int64)
```

```
plt.scatter(X_test[:,1],X_test[:,-2])
```

```
<matplotlib.collections.PathCollection at 0x1c2ac2742e0>
```




```
cell_df
```

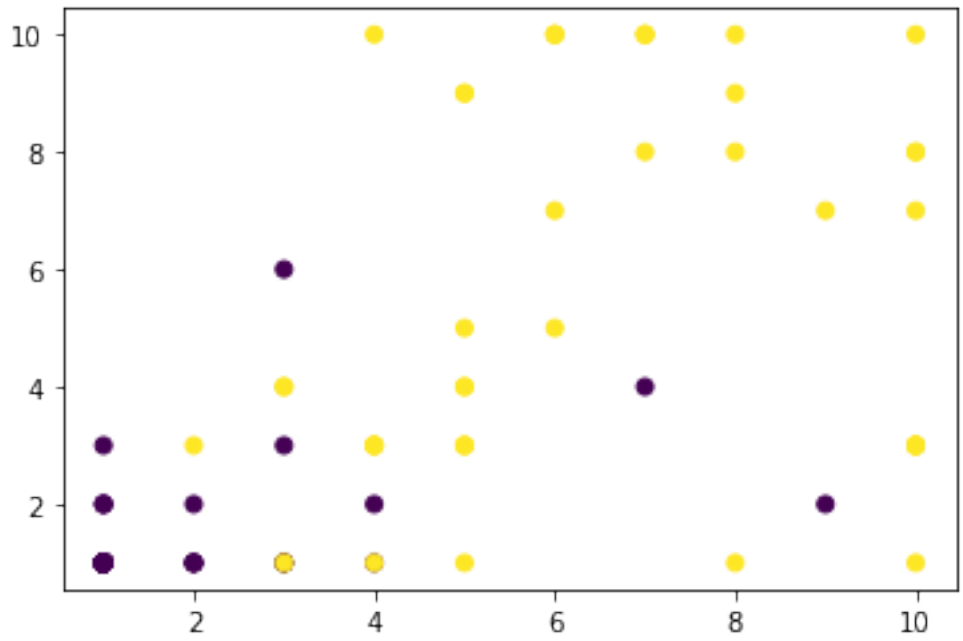
```
     Clump  UnifSize  UnifShape  MargAdh  SingEpiSize  BlandChrom
NormNucl  \
0        5         1          1        1            2           3
1
1        5         4          4        5            7           3
2
2        3         1          1        1            2           3
1
3        6         8          8        1            3           3
7
4        4         1          1        3            2           3
1
..     ...       ...        ...      ...          ...         ...
...
694      3         1          1        1            3           1
1
695      2         1          1        1            2           1
1
696      5        10         10        3            7           8
```

```
10
697      4          8          6         4              3           10
6
698      4          8          8         5              4           10
4
```

```
     Mit  Class
0      1      2
1      1      2
2      1      2
3      1      2
4      1      2
..   ...    ...
694    1      2
695    1      2
696    2      4
697    1      4
698    1      4

[699 rows x 9 columns]
```

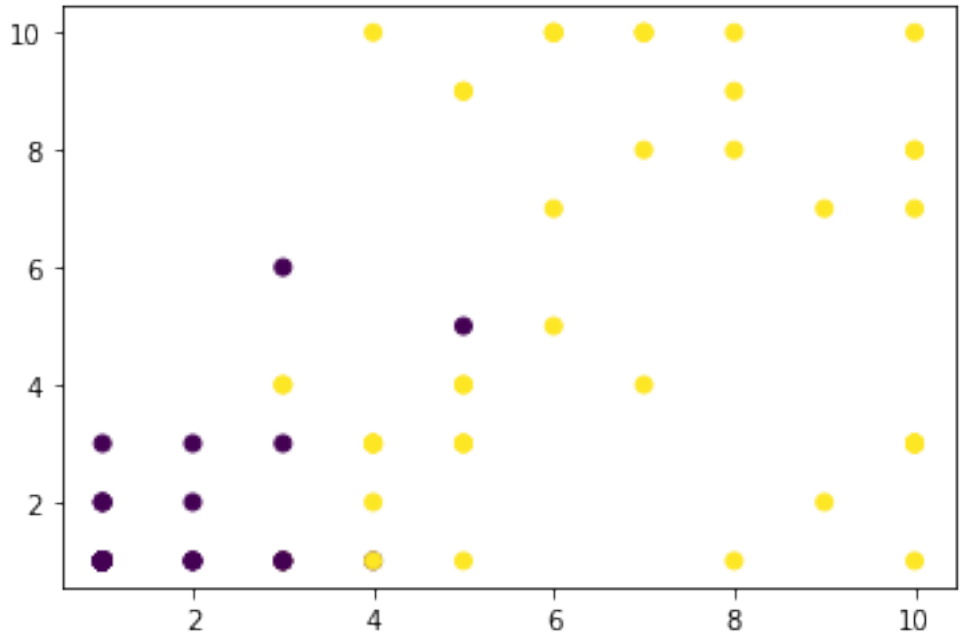
```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c = y_test) # Coloring based on
actual values
```

```
<matplotlib.collections.PathCollection at 0x1c2ac2d7970>
```




```
import matplotlib.pyplot as plt
plt.scatter(X_test[:,1],X_test[:,-2],c = yhat)
```

```
<matplotlib.collections.PathCollection at 0x1c2ac331f10>
```



Q7 : We are providing you a dataset apart from churn and cell dataset which is titanic dataset remove unnecessary column which are not usefull with aspect of machine learning and apply label encoding where ever its necessary and store processed data into your memory

Dependent Column : Survived

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

titanic_df = pd.read_csv("C:/Users/Lenovo/Documents/Data
Set/titanic.csv")

titanic_df
```

```
     Unnamed: 0  PassengerId  Survived  Pclass  \
0             0            1         0       3
1             1            2         1       1
2             2            3         1       3
3             3            4         1       1
4             4            5         0       3
..          ...          ...       ...     ...
707         885          886         0       3
708         886          887         0       2
709         887          888         1       1
710         889          890         1       1
```

```
711         890          891         0       3

                                                  Name     Sex   Age  
SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0  
1   
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0  
1   
2                               Heikkinen, Miss. Laina  female  26.0  
0   
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0  
1   
4                             Allen, Mr. William Henry    male  35.0  
0   
..                                                 ...     ...   ...  
...   
707               Rice, Mrs. William (Margaret Norton)  female  39.0  
0   
708                              Montvila, Rev. Juozas    male  27.0  
0   
709                       Graham, Miss. Margaret Edith  female  19.0  
0   
710                                Behr, Mr. Karl Howell    male  26.0  
0   
711                                  Dooley, Mr. Patrick    male  32.0  
0   

     Parch            Ticket     Fare Embarked  
0        0         A/5 21171   7.2500        S  
1        0          PC 17599  71.2833        C  
2        0  STON/O2. 3101282   7.9250        S  
3        0            113803  53.1000        S  
4        0            373450   8.0500        S  
..     ...               ...      ...      ...  
707      5            382652  29.1250        Q  
708      0            211536  13.0000        S  
709      0            112053  30.0000        S  
710      0            111369  30.0000        C  
711      0            370376   7.7500        Q  

[712 rows x 12 columns]
```

```
titanic_df.head()
```

```
   Unnamed: 0  PassengerId  Survived  Pclass  \
0           0            1         0       3   
1           1            2         1       1   
2           2            3         1       3   
3           3            4         1       1   
4           4            5         0       3   
```

```
                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Embarked
0      0         A/5 21171   7.2500        S
1      0          PC 17599  71.2833        C
2      0  STON/O2. 3101282   7.9250        S
3      0            113803  53.1000        S
4      0            373450   8.0500        S

titanic_df.tail()

     Unnamed: 0  PassengerId  Survived  Pclass  \
707         885          886         0       3
708         886          887         0       2
709         887          888         1       1
710         889          890         1       1
711         890          891         0       3

                                     Name     Sex   Age  SibSp  Parch
Ticket  \
707  Rice, Mrs. William (Margaret Norton)  female  39.0      0      5
382652
708                 Montvila, Rev. Juozas    male  27.0      0      0
211536
709          Graham, Miss. Margaret Edith  female  19.0      0      0
112053
710                 Behr, Mr. Karl Howell    male  26.0      0      0
111369
711                   Dooley, Mr. Patrick    male  32.0      0      0
370376

       Fare Embarked
707  29.125        Q
708  13.000        S
709  30.000        S
710  30.000        C
711   7.750        Q
```

```
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 712 entries, 0 to 711
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   712 non-null    int64  
 1   PassengerId  712 non-null    int64  
 2   Survived     712 non-null    int64  
 3   Pclass       712 non-null    int64  
 4   Name         712 non-null    object 
 5   Sex          712 non-null    object 
 6   Age          712 non-null    float64
 7   SibSp        712 non-null    int64  
 8   Parch        712 non-null    int64  
 9   Ticket       712 non-null    object 
 10  Fare         712 non-null    float64
 11  Embarked     712 non-null    object 
dtypes: float64(2), int64(6), object(4)
memory usage: 66.9+ KB
```

```
titanic_df.isnull().sum()
```

```
Unnamed: 0     0
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

```
titanic_df.shape
```

```
(712, 12)
```

```
titanic_df = titanic_df.drop(["Name","Ticket","Unnamed:
0","PassengerId","Age"],axis = 1)
```

```
titanic_df
```

```
     Survived  Pclass     Sex  SibSp  Parch     Fare Embarked
0           0       3    male      1      0   7.2500        S
1           1       1  female      1      0  71.2833        C
2           1       3  female      0      0   7.9250        S
3           1       1  female      1      0  53.1000        S
```

```
4           0       3    male      0      0   8.0500        S
..        ...     ...     ...    ...    ...      ...      ...
707         0       3  female      0      5  29.1250        Q
708         0       2    male      0      0  13.0000        S
709         1       1  female      0      0  30.0000        S
710         1       1    male      0      0  30.0000        C
711         0       3    male      0      0   7.7500        Q

[712 rows x 7 columns]
```

```
X =
titanic_df[["Pclass","Sex","SibSp","Parch","Fare","Embarked"]].values
```

```
X
```

```
array([[3, 'male', 1, 0, 7.25, 'S'],
       [1, 'female', 1, 0, 71.2833, 'C'],
       [3, 'female', 0, 0, 7.925, 'S'],
       ...,
       [1, 'female', 0, 0, 30.0, 'S'],
       [1, 'male', 0, 0, 30.0, 'C'],
       [3, 'male', 0, 0, 7.75, 'Q']], dtype=object)
```

Label Encoding

```
from sklearn import preprocessing
le_Sex = preprocessing.LabelEncoder()
le_Sex.fit(['female','male'])
X[:,1] = le_Sex.transform(X[:,1])  # I AM UPDATING MY FIRST COLUMN
FROM F,M TO 0,1


le_Embarked = preprocessing.LabelEncoder()
le_Embarked.fit([ 'S', 'C', 'Q'])
X[:,5] = le_Embarked.transform(X[:,5]) # I AM UPDATING MY SECOND
COLUMN FROM S,C,Q TO 0,1,2

X[0:6]
```

```
array([[3, 1, 1, 0, 7.25, 2],
       [1, 0, 1, 0, 71.2833, 0],
       [3, 0, 0, 0, 7.925, 2],
       [1, 0, 1, 0, 53.1, 2],
       [3, 1, 0, 0, 8.05, 2],
       [1, 1, 0, 0, 51.8625, 2]], dtype=object)
```

```
X
```

```
array([[3, 1, 1, 0, 7.25, 2],
       [1, 0, 1, 0, 71.2833, 0],
       [3, 0, 0, 0, 7.925, 2],
```

```
       ...,
       [1, 0, 0, 0, 30.0, 2],
       [1, 1, 0, 0, 30.0, 0],
       [3, 1, 0, 0, 7.75, 1]], dtype=object)
```

```
Y = titanic_df[["Survived"]]
```

```
Y
```

```
     Survived
0           0
1           1
2           1
3           1
4           0
..        ...
707         0
708         0
709         1
710         1
711         0

[712 rows x 1 columns]
```

```
set(titanic_df["Survived"])
```

```
{0, 1}
```

```
from sklearn import preprocessing
X = preprocessing.StandardScaler().fit(X).transform(X)
```

```
X
```

```
array([[ 0.90859974,  0.75613751,  0.52251079, -0.50678737, -
0.51637992,
         0.51958818],
       [-1.48298257, -1.32251077,  0.52251079, -0.50678737,
0.69404605,
        -2.04948671],
       [ 0.90859974, -1.32251077, -0.55271372, -0.50678737, -
0.50362035,
         0.51958818],
       ...,
       [-1.48298257, -1.32251077, -0.55271372, -0.50678737, -
0.08633507,
         0.51958818],
       [-1.48298257,  0.75613751, -0.55271372, -0.50678737, -
0.08633507,
        -2.04948671],
       [ 0.90859974,  0.75613751, -0.55271372, -0.50678737, -
0.50692839,
        -0.76494927]])
```

```
from sklearn import preprocessing
Y = preprocessing.StandardScaler().fit(Y).transform(Y)

Y

array([[-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
```

```
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
```

```
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
```

```
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
```

```
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
```

```
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
```

```
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
```

```
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
```

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
```

```
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338]])
```

Que.8 Use that processed titanic dataset and apply svm in it

Train and Test Dataset

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=100)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

```
Train set: (569, 6) (569, 1)
Test set: (143, 6) (143, 1)
```

```
X_train
```

```
array([[ 0.90859974,  0.75613751,  4.82340884,  1.83628152,
0.23312681,
         0.51958818],
       [ 0.90859974,  0.75613751,  3.74818432,  0.66474707, -
0.10287526,
        -0.76494927],
       [ 0.90859974,  0.75613751,  0.52251079,  0.66474707, -
0.19691804,
         0.51958818],
       ...,
       [-1.48298257,  0.75613751,  0.52251079,  4.17935042,  4.3180803
,
         0.51958818],
       [ 0.90859974,  0.75613751, -0.55271372, -0.50678737, -
0.48983623,
         0.51958818],
```

```
       [-1.48298257,  0.75613751,  1.5977353 , -0.50678737,
1.87296816,
         0.51958818]])

X_test

array([[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
         6.64747074e-01, -2.96258635e-02,  5.19588181e-01],
       [-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
         6.64747074e-01,  9.18520007e-01, -2.04948671e+00],
       [ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
         4.17935042e+00, -1.26031522e-01,  5.19588181e-01],
       [ 9.08599738e-01, -1.32251077e+00,  5.22510788e-01,
        -5.06787373e-01, -5.04958689e-01,  5.19588181e-01],
       [ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -4.73848015e-01,  5.19588181e-01],
       [-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -7.68835392e-02,  5.19588181e-01],
       [-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
        -5.06787373e-01, -2.56462707e-01,  5.19588181e-01],
       [-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01,  1.76318123e-02, -2.04948671e+00],
       [-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -4.07687269e-01,  5.19588181e-01],
       [-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -6.53427183e-01,  5.19588181e-01],
       [-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
       [-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
       [-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
         1.83628152e+00,  6.88690808e-01,  5.19588181e-01],
       [ 9.08599738e-01, -1.32251077e+00,  5.22510788e-01,
        -5.06787373e-01, -3.79332664e-01,  5.19588181e-01],
       [-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -3.69015368e-01, -2.04948671e+00],
       [ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -3.49087751e-01,  5.19588181e-01],
       [-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
        -5.06787373e-01,  9.51344005e-02, -2.04948671e+00],
       [-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
        -5.06787373e-01, -1.61947356e-01,  5.19588181e-01],
       [-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
        -5.06787373e-01, -1.61947356e-01,  5.19588181e-01],
       [-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -4.24648288e-02,  5.19588181e-01],
       [-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -8.63350744e-02,  5.19588181e-01],
       [ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -5.01650651e-01, -7.64949267e-01],
       [ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
```

```
 -5.06787373e-01, -5.04958689e-01,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01,  1.64359541e-03, -2.04948671e+00],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -1.51550667e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -4.26590340e-01, -2.04948671e+00],
[-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
  3.00781597e+00, -2.18656566e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.36041875e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01,  1.91045862e+00, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.89679337e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
  6.64747074e-01, -3.99810360e-01, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04172321e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
  6.64747074e-01,  3.34150044e+00,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
  6.64747074e-01,  5.18169991e-01, -2.04948671e+00],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -5.16379924e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.16379924e-01,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -1.29418952e-01, -2.04948671e+00],
[-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
  6.64747074e-01, -1.61947356e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -4.36041875e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
 -5.06787373e-01,  4.03254446e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.07687269e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
  6.64747074e-01, -2.18656566e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
 -5.06787373e-01,  1.48781810e+00, -2.04948671e+00],
[-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
  6.64747074e-01,  1.44260763e+00, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -4.40924538e-01, -2.04948671e+00],
[-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
  1.83628152e+00,  5.75272386e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
```

```
  6.64747074e-01, -1.61947356e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
  1.83628152e+00,  2.46305964e+00,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
  5.35088486e+00,  9.67884191e-02,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
  6.64747074e-01, -2.65441665e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.06062628e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
  1.83628152e+00, -1.61947356e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
  1.83628152e+00, -1.57221588e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
  6.64747074e-01, -4.92435404e-01, -2.04948671e+00],
[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
  1.83628152e+00, -3.79332664e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -1.29418952e-01, -2.04948671e+00],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -1.51550667e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.73848015e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -1.61947356e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04172321e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.16379924e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
 -5.06787373e-01,  1.04784914e+00, -7.64949267e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
  6.64747074e-01, -4.17611381e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
  6.64747074e-01, -8.63350744e-02,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00,  5.22510788e-01,
  4.17935042e+00, -1.26031522e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.11733549e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
  6.64747074e-01, -3.55860722e-01, -2.04948671e+00],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -7.68835392e-02,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -4.72193997e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
```

```
  6.64747074e-01, -1.61947356e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.07687269e-01,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
  6.64747074e-01,  8.07543862e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.18742807e-01,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
  1.83628152e+00,  1.48781810e+00, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.16773107e-01, -2.04948671e+00],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01,  2.82354189e-01, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04172321e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.06062628e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.03620351e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.06455812e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
  6.64747074e-01, -1.61947356e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
  5.35088486e+00, -6.22336597e-02,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -4.07687269e-01,  5.19588181e-01],
[-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
  6.64747074e-01, -1.05238145e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.30636619e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
  6.64747074e-01, -5.16773107e-01, -2.04948671e+00],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -5.10709002e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -5.11733549e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -3.88784199e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.79203255e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01,  7.35948483e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04958689e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04172321e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
```

```
  4.17935042e+00, -2.55044977e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -5.07244070e-01, -7.64949267e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.16852500e-01, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01,  4.14516896e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00,  1.59773530e+00,
  1.83628152e+00, -3.63414181e-03,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -4.67388836e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04958689e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01,  2.21133312e+00,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -1.61947356e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -1.43044285e-01,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01,  3.50325849e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
  6.64747074e-01, -4.94562000e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -3.53813519e-01,  5.19588181e-01],
[-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -4.07687269e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.06062628e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
  1.83628152e+00, -3.65234754e-01, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
  6.64747074e-01, -4.42973631e-01,  5.19588181e-01],
[-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
 -5.06787373e-01,  9.24506610e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
  6.64747074e-01, -3.81222971e-01,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -8.63350744e-02,  5.19588181e-01],
[-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01,  8.43695984e-01, -2.04948671e+00],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.06062628e-01,  5.19588181e-01],
[ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
 -5.06787373e-01, -5.06455812e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
 -5.06787373e-01, -5.04958689e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
 -5.06787373e-01, -5.20239931e-01,  5.19588181e-01],
[ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
```

```
        -5.06787373e-01, -5.04172321e-01,  5.19588181e-01],
       [ 9.08599738e-01, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -5.16773107e-01, -2.04948671e+00],
       [ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -5.04958689e-01,  5.19588181e-01],
       [-1.48298257e+00, -1.32251077e+00, -5.52713724e-01,
         6.64747074e-01,  4.42557710e-01, -2.04948671e+00],
       [ 9.08599738e-01,  7.56137507e-01,  5.22510788e-01,
         6.64747074e-01, -2.71348875e-01,  5.19588181e-01],
       [ 9.08599738e-01, -1.32251077e+00,  5.22510788e-01,
         5.35088486e+00, -6.01070643e-02,  5.19588181e-01],
       [-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
         6.64747074e-01,  5.06828149e-01, -2.04948671e+00],
       [-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
       [ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -5.06928388e-01, -7.64949267e-01],
       [ 9.08599738e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -5.16773107e-01, -2.04948671e+00],
       [-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -4.54944945e-01,  5.19588181e-01],
       [ 9.08599738e-01, -1.32251077e+00,  1.59773530e+00,
         6.64747074e-01, -2.89386185e-01, -2.04948671e+00],
       [-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
         1.83628152e+00,  2.21133312e+00,  5.19588181e-01],
       [-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
        -5.06787373e-01, -1.61947356e-01,  5.19588181e-01],
       [-2.87191414e-01, -1.32251077e+00, -5.52713724e-01,
        -5.06787373e-01, -4.14303344e-01,  5.19588181e-01],
       [-2.87191414e-01, -1.32251077e+00,  5.22510788e-01,
         1.83628152e+00,  5.75272386e-01,  5.19588181e-01],
       [-1.48298257e+00, -1.32251077e+00,  5.22510788e-01,
        -5.06787373e-01,  3.50325849e-01,  5.19588181e-01],
       [ 9.08599738e-01,  7.56137507e-01,  3.74818432e+00,
         6.64747074e-01,  9.67884191e-02,  5.19588181e-01],
       [-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -1.61947356e-01,  5.19588181e-01],
       [-1.48298257e+00,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -1.51550667e-01,  5.19588181e-01],
       [-2.87191414e-01,  7.56137507e-01, -5.52713724e-01,
        -5.06787373e-01, -4.07687269e-01,  5.19588181e-01]])

Y_train

array([[-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
```

```
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
```

[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],
[1.21335165],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[1.21335165],
[-0.82416338],

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
```

```
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
```

```
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
```

```
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
```

[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[1.21335165],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[1.21335165],
[1.21335165],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],
[-0.82416338],
[-0.82416338],
[1.21335165],
[-0.82416338],
[1.21335165],

```
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
```

```
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
```

```
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
```

```
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165]])
```

```
Y_test
```

```
array([[ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
       [ 1.21335165],
       [-0.82416338],
```

```
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
```

```
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[-0.82416338],
[ 1.21335165],
[-0.82416338],
[ 1.21335165],
```

```
       [ 1.21335165],
       [ 1.21335165],
       [ 1.21335165],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338],
       [-0.82416338]])
```

```
from sklearn import preprocessing
from sklearn import utils

#convert y values to categorical values
value = preprocessing.LabelEncoder()
Y_train_transformed = value.fit_transform(Y_train)

#view transformed values
print(Y_train_transformed)
```

```
[0 0 0 0 1 1 0 1 1 1 1 0 1 0 1 0 1 0 1 0 0 1 0 0 1 1 0 0 0 1 1 1 0 0 1
1 0
 0 1 1 1 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0
1 1
 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 1 1 0 1 0
0 0
 0 1 0 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0
1 1
 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 1 1 0 0 1 0 0
1 0
 0 1 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1
1 0
 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 1
1 0
 1 1 0 1 0 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 1 0 1 0 1 0 0 0 0 1 0
0 0
 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1
0 1
 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0
0 0
 0 0 0 1 1 1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1
0 1
 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 1 0 1 1 0 0
1 1
 0 0 1 1 0 1 1 0 0 0 1 0 1 0 1 0 1 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1
0 0
 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0
0 1
 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1
1 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 1]
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\preprocessing\
_label.py:115: DataConversionWarning: A column-vector y was passed
when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
from sklearn import preprocessing
from sklearn import utils

#convert y values to categorical values
value = preprocessing.LabelEncoder()
Y_test_transformed = value.fit_transform(Y_test)

#view transformed values
print(Y_test_transformed)
```

```
[1 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1
0 0
 0 1 0 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0
1 0
 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0
0 0
 1 1 1 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0]
```

Modelling (Sciket-Learn)

```
from sklearn import svm
clf = svm.SVC(kernel='rbf')
clf.fit(X_train, Y_train_transformed) # question and answers
```

```
SVC()
```

```
yhat = clf.predict(X_test) #question
yhat [0:5]
```

```
array([1, 1, 0, 0, 0], dtype=int64)
```

Evaluation

```
from sklearn.metrics import f1_score
```

```
f1_score(Y_test_transformed, yhat) #actualvale,predvalue
```

```
0.6923076923076923
```

```
from sklearn import svm
clf2 = svm.SVC(kernel='linear')
clf2.fit(X_train, Y_train_transformed)
yhat2 = clf2.predict(X_test)
yhat2
```

```
array([1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1,
1,
       0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1,
1,
       1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,
1,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
0,
       0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0], dtype=int64)
```

```
from sklearn.metrics import f1_score
```

```
f1_score(Y_test_transformed, yhat2) #actualvale,predvalue
```

```
0.7627118644067797
```

```
from sklearn import svm
clf = svm.SVC(kernel='poly')
clf.fit(X_train, Y_train_transformed) # question and answers
```

```
SVC(kernel='poly')
```

```
yhat3 = clf.predict(X_test) #question
yhat3 [0:5]
```

```
array([1, 1, 0, 0, 0], dtype=int64)
```

Evaluation

```
from sklearn.metrics import f1_score
```

```
f1_score(Y_test_transformed, yhat)
```

```
0.6923076923076923
```

Ans: The possible accuracy gives the kernel as linear that is 76.27%