Feature engineering house rant prediction

September 13, 2023

[1]:
```python
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

[2]:
```python
# import csv file
dataset = pd.read_csv("C:/Users/Vikas/Downloads/train.csv")
```

[4]:
```python
dataset.head(10)
```

[4]:
```
   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
5   6          50       RL         85.0    14115   Pave   NaN      IR1
6   7          20       RL         75.0    10084   Pave   NaN      Reg
7   8          60       RL          NaN    10382   Pave   NaN      IR1
8   9          50       RM         51.0     6120   Pave   NaN      Reg
9  10         190       RL         50.0     7420   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
5         Lvl    AllPub  …        0    NaN  MnPrv        Shed     700
6         Lvl    AllPub  …        0    NaN    NaN         NaN       0
7         Lvl    AllPub  …        0    NaN    NaN        Shed     350
8         Lvl    AllPub  …        0    NaN    NaN         NaN       0
9         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
5      10   2009          WD          Normal      143000
6       8   2007          WD          Normal      307000
7      11   2009          WD          Normal      200000
8       4   2008          WD         Abnorml      129900
9       1   2008          WD          Normal      118000

[10 rows x 81 columns]
```

1 in data analysis we will analyze to find out the below stuff

1.missimg values 2.all the numerical variables 3. distribution of the numerical variables 4. categorical variables 5.cardibality of categorical variables 6.outliers 7.relationship between the independent and dependent feature

2 missing values

[18]:

```
## here we will check the percentage of non null values present tin each feature
##.1 step make the list of feature which has missing values
features_with_na = [features for features in dataset.columns if␣
 ↪dataset[features].isnull().sum()>1]

##2. step print the features name and percentage of missing values
for features in features_with_na:
                                print(features, np.round(dataset[features].
 ↪isnull().mean(),4),  ' %missing values')
```

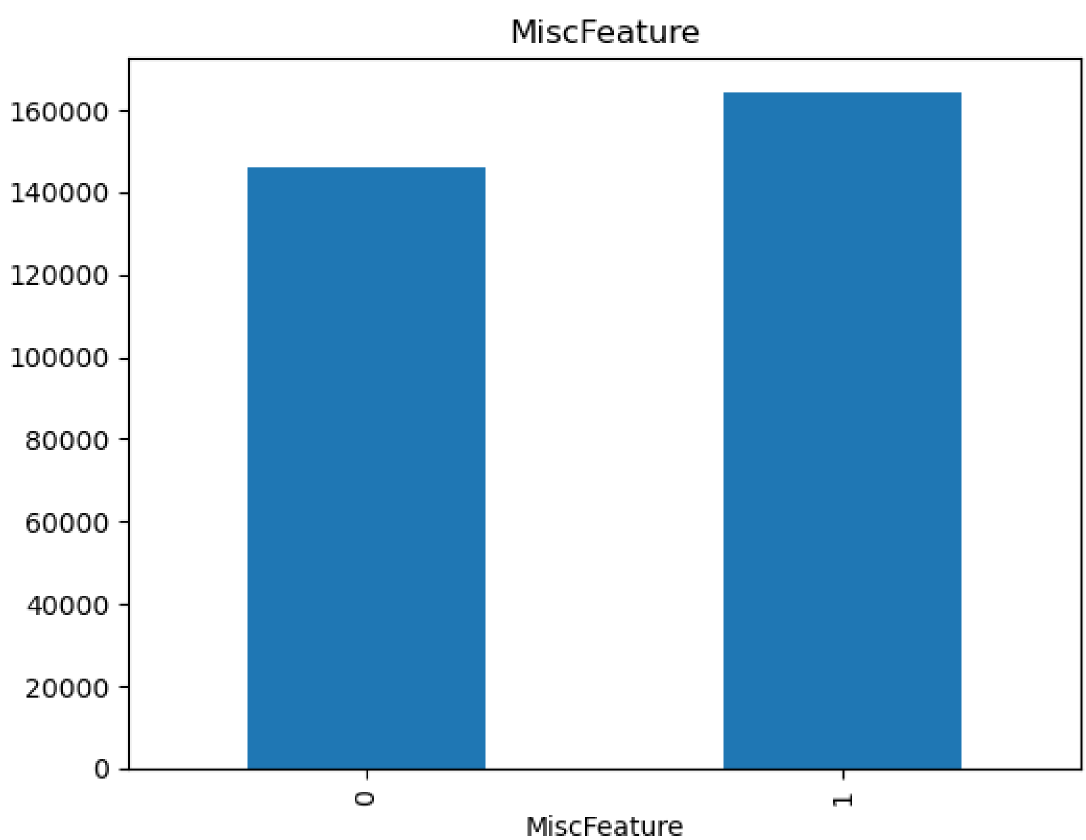
```
LotFrontage 0.1774  %missing values
Alley 0.9377  %missing values
MasVnrType 0.0055  %missing values
MasVnrArea 0.0055  %missing values
BsmtQual 0.0253  %missing values
BsmtCond 0.0253  %missing values
BsmtExposure 0.026  %missing values
BsmtFinType1 0.0253  %missing values
BsmtFinType2 0.026  %missing values
FireplaceQu 0.4726  %missing values
GarageType 0.0555  %missing values
GarageYrBlt 0.0555  %missing values
GarageFinish 0.0555  %missing values
```

```
GarageQual 0.0555  %missing values
GarageCond 0.0555  %missing values
PoolQC 0.9952  %missing values
Fence 0.8075  %missing values
MiscFeature 0.963  %missing values
```

3 since there are many missing values we need to find the relationship between missing values and sales price

lets plot diagram for this relationship

[20]:
```
for feature in features_with_na:
    data = dataset.copy()
    ## lets make a variable that indecates 1  if the observation was missig␣
 ↪null values is converted into 1 otherwise 0
data[feature] = np.where(data[feature].isnull(),1,0)
## lets caluclate the meansales price where the infomation is missing
data.groupby(feature)['SalePrice'].median().plot.bar()
plt.title(feature)
plt.show()
```



[25]:
```python
## how many features actually numerical variables
numerical_feature = [feature for feature in dataset.columns if dataset[feature].
  ↪dtypes !='O']
print('Number of numerical variables:', len(numerical_feature))

## visualise the numerical variables
dataset[numerical_features].head()
```

```
Number of numerical variables: 38
```

[25]:
```
   Id  MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond  YearBuilt  \
0   1          60         65.0     8450            7            5       2003
1   2          20         80.0     9600            6            8       1976
2   3          60         68.0    11250            7            5       2001
3   4          70         60.0     9550            7            5       1915
4   5          60         84.0    14260            8            5       2000

   YearRemodAdd  MasVnrArea  BsmtFinSF1  …  WoodDeckSF  OpenPorchSF  \
0          2003       196.0         706  …           0           61
1          1976         0.0         978  …         298            0
2          2002       162.0         486  …           0           42
3          1970         0.0         216  …           0           35
4          2000       350.0         655  …         192           84

   EnclosedPorch  3SsnPorch  ScreenPorch  PoolArea  MiscVal  MoSold  YrSold  \
0              0          0            0         0        0       2    2008
1              0          0            0         0        0       5    2007
2              0          0            0         0        0       9    2008
3            272          0            0         0        0       2    2006
4              0          0            0         0        0      12    2008

   SalePrice
0     208500
1     181500
2     223500
3     140000
4     250000

[5 rows x 38 columns]
```

[26]:
```python
year_feature = [feature for feature in numerical_features if 'Yr' in feature ␣
  ↪or 'Year' in feature]
year_feature
```

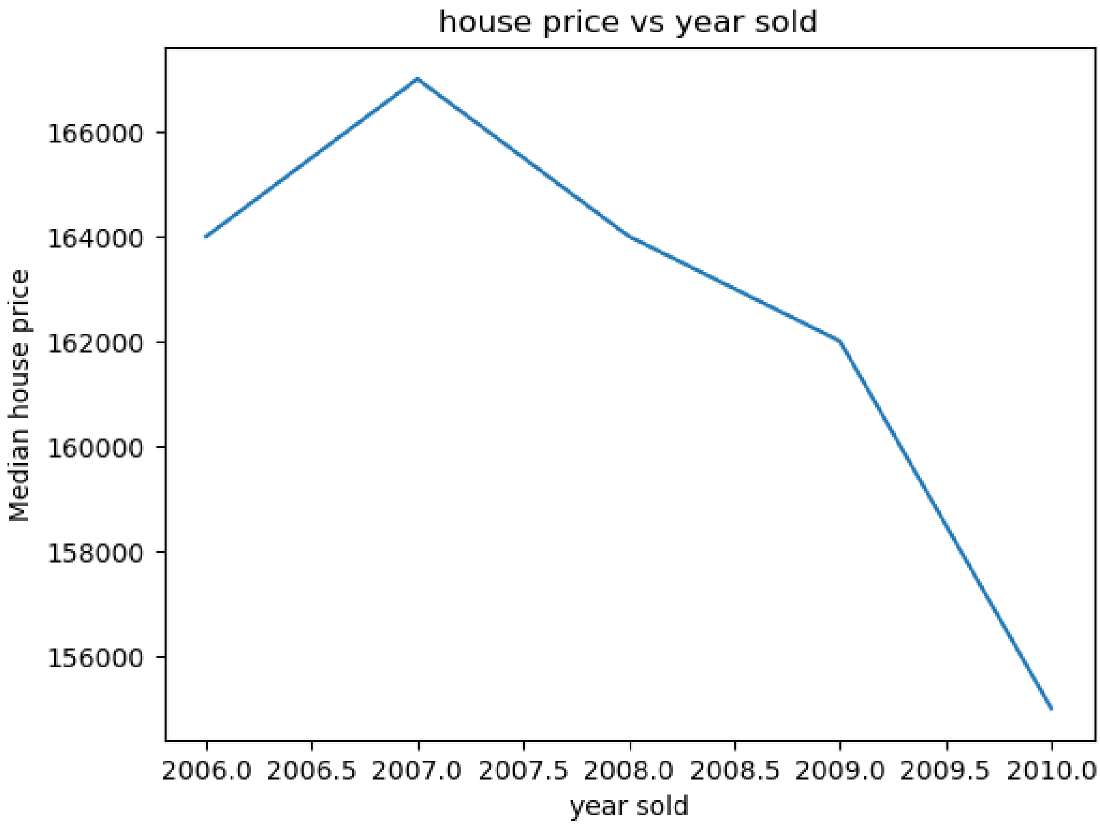
[26]:
```
['YearBuilt', 'YearRemodAdd', 'GarageYrBlt', 'YrSold']
```

[27]:
```
#3 lets explore the containt of these year varieables
for feature in year_feature:
    print(feature,dataset[feature].unique())
```

```
YearBuilt [2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 1965 2005 1962 2006
 1960 1929 1970 1967 1958 1930 2002 1968 2007 1951 1957 1927 1920 1966
 1959 1994 1954 1953 1955 1983 1975 1997 1934 1963 1981 1964 1999 1972
 1921 1945 1982 1998 1956 1948 1910 1995 1991 2009 1950 1961 1977 1985
 1979 1885 1919 1990 1969 1935 1988 1971 1952 1936 1923 1924 1984 1926
 1940 1941 1987 1986 2008 1908 1892 1916 1932 1918 1912 1947 1925 1900
 1980 1989 1992 1949 1880 1928 1978 1922 1996 2010 1946 1913 1937 1942
 1938 1974 1893 1914 1906 1890 1898 1904 1882 1875 1911 1917 1872 1905]
YearRemodAdd [2003 1976 2002 1970 2000 1995 2005 1973 1950 1965 2006 1962 2007
1960
 2001 1967 2004 2008 1997 1959 1990 1955 1983 1980 1966 1963 1987 1964
 1972 1996 1998 1989 1953 1956 1968 1981 1992 2009 1982 1961 1993 1999
 1985 1979 1977 1969 1958 1991 1971 1952 1975 2010 1984 1986 1994 1988
 1954 1957 1951 1978 1974]
GarageYrBlt [2003. 1976. 2001. 1998. 2000. 1993. 2004. 1973. 1931. 1939. 1965.
2005.
 1962. 2006. 1960. 1991. 1970. 1967. 1958. 1930. 2002. 1968. 2007. 2008.
 1957. 1920. 1966. 1959. 1995. 1954. 1953.   nan 1983. 1977. 1997. 1985.
 1963. 1981. 1964. 1999. 1935. 1990. 1945. 1987. 1989. 1915. 1956. 1948.
 1974. 2009. 1950. 1961. 1921. 1900. 1979. 1951. 1969. 1936. 1975. 1971.
 1923. 1984. 1926. 1955. 1986. 1988. 1916. 1932. 1972. 1918. 1980. 1924.
 1996. 1940. 1949. 1994. 1910. 1978. 1982. 1992. 1925. 1941. 2010. 1927.
 1947. 1937. 1942. 1938. 1952. 1928. 1922. 1934. 1906. 1914. 1946. 1908.
 1929. 1933.]
YrSold [2008 2007 2006 2009 2010]
```

[28]:
```
## we will check whether there is a relation between year the house is sold
dataset.groupby('YrSold')['SalePrice'].median().plot()
plt.xlabel('year sold')
plt.ylabel('Median house price')
plt.title('house price vs year sold')
```

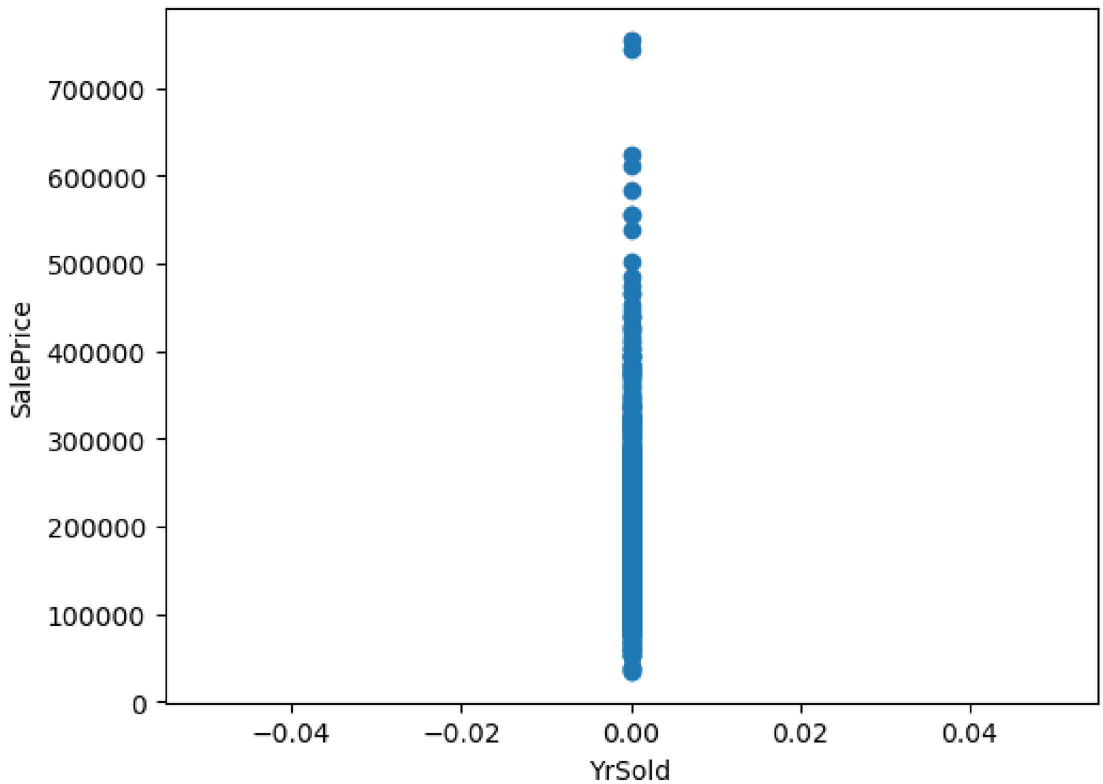
[28]:
```
Text(0.5, 1.0, 'house price vs year sold')
```



it tis pretty much amazing here year increases the price is decreses

[36]:
```
##here we will compare between the  all year feature  with sales price
for feature in year_feature:
    if feature !='YrSold':
        data=dataset.copy()
## we will copy the difference between year variable and year the house was␣
 ↪house sold for
data[feature] = data['YrSold']-data[feature]

plt.scatter(data[feature],data['SalePrice'])
plt.xlabel(feature)
plt.ylabel('SalePrice')
plt.show()
```



[14]:
```
## numerical variables are two types 1. continuos variables 2.descrete variable
discreate_feature=[feature for feature in numerical_features if␣
↪len(dataset[feature].unique())<25] and feature not in year_feature+['Id']
print("discrete variable count: ()".format(len)(discreate_feature))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[14], line 2
      1 ## numerical variables are two types 1. continuos variables 2.descrete␣
↪variable
----> 2 discreate_feature=[feature for feature in Numerical_features if␣
↪len(dataset[feature].unique())<25] and feature not in year_feature+['Id']
      3 print("discrete variable count: ()".format(len)(discreate_feature))

NameError: name 'Numerical_features' is not defined
```

[ ]: