Project Name : House Prices :Advanced Regression Techniques

The main aim of this project is to predict the house price based on various features which we will discuss as we go ahead

All the lifecycle In A data science projects :

1- Data Analysis 2- Feature Engineering 3- Feature Selection 4- Model Buliding 5- Model Deployment

In [19]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns',None)
```

In [20]:
```
datafeame=pd.read_csv('train.csv')
```

In [21]:
```
datafeame.head()
```

Out[21]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilitie
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPu
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPu
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPu
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPu
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPu

In [22]:
```
datafeame.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     588 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
```

In Data Analysis We will Analyze To Find out the below stuff

1. Missing Values
2. All The Numerical Variables
3. Distribution of the Numerical Variables
4. Categorical Variables
5. Cardinality of Categorical Variables
6. Outliers
7. Relationship between independent and dependent feature(SalePrice)

Missing Values

In [23]:
```
features_with_na=[item for item in datafeame.columns if datafeame[item].isnull().
```

In [24]:
```
datafeame['LotFrontage'].isnull().sum()
```

Out[24]: 259

In [25]:
```
features_with_na
```

Out[25]:
```
['LotFrontage',
 'Alley',
 'MasVnrType',
 'MasVnrArea',
 'BsmtQual',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'Electrical',
 'FireplaceQu',
 'GarageType',
 'GarageYrBlt',
 'GarageFinish',
 'GarageQual',
 'GarageCond',
 'PoolQC',
 'Fence',
 'MiscFeature']
```

In [26]:
```
datafeame['GarageFinish'].isnull().sum()
```

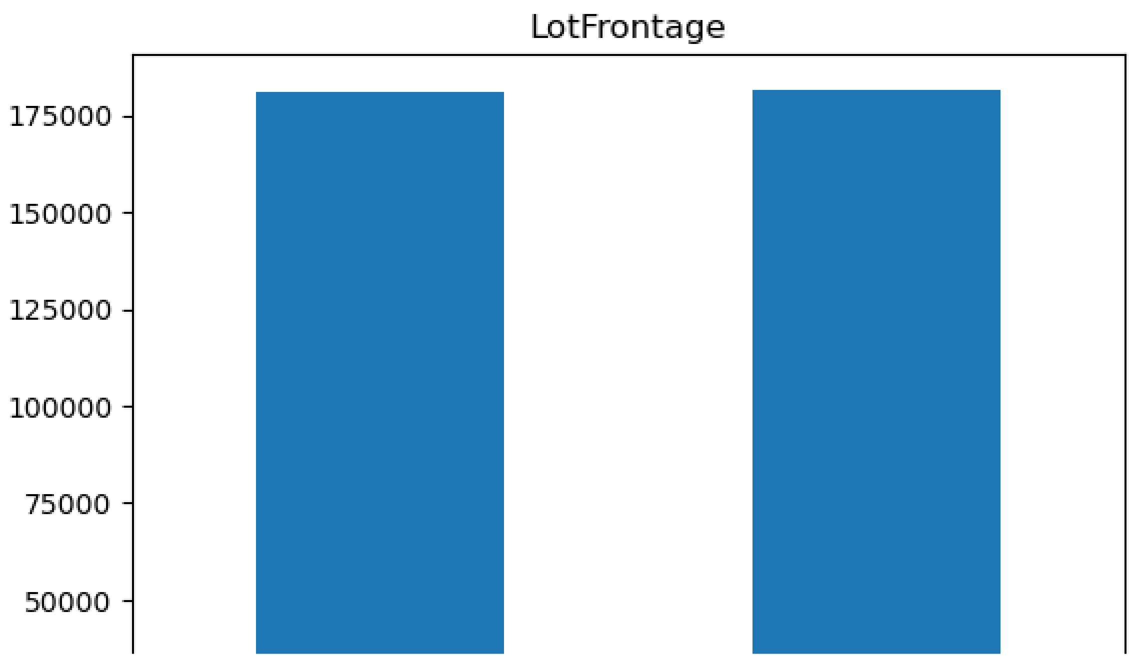
Out[26]:
```
81
```

In [27]:
```
for i in features_with_na:
    print(i,np.round(datafeame[i].isnull().mean(),2) ,'% missing values')
```

```
LotFrontage 0.18 % missing values
Alley 0.94 % missing values
MasVnrType 0.6 % missing values
MasVnrArea 0.01 % missing values
BsmtQual 0.03 % missing values
BsmtCond 0.03 % missing values
BsmtExposure 0.03 % missing values
BsmtFinType1 0.03 % missing values
BsmtFinType2 0.03 % missing values
Electrical 0.0 % missing values
FireplaceQu 0.47 % missing values
GarageType 0.06 % missing values
GarageYrBlt 0.06 % missing values
GarageFinish 0.06 % missing values
GarageQual 0.06 % missing values
GarageCond 0.06 % missing values
PoolQC 1.0 % missing values
Fence 0.81 % missing values
MiscFeature 0.96 % missing values
```

The relationsip between missing values and sales price

In [35]:
```
for feature in features_with_na :
    data=datafeame.copy()
#let us make a variable that indicates 1 if the observation was missing or xero c
    data[feature]=np.where(data[feature].isnull(),1,0)

# let us calculate the mean sale price where the information is missing or preser
    data.groupby(feature)['SalePrice'].mean().plot.bar()
    plt.title(feature)
    plt.show()
```



Numerical Values

In [37]:
```
numerical_features=[item for item in datafeame.columns if datafeame.dtypes[item]!
```

In [41]:
```
datafeame[numerical_features]
```

Out[41]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAd
0	1	60	65.0	8450	7	5	2003	200
1	2	20	80.0	9600	6	8	1976	197
2	3	60	68.0	11250	7	5	2001	200
3	4	70	60.0	9550	7	5	1915	197
4	5	60	84.0	14260	8	5	2000	200
...	...	...	...	...	...	...	...	.
1455	1456	60	62.0	7917	6	5	1999	200
1456	1457	20	85.0	13175	6	6	1978	198
1457	1458	70	66.0	9042	7	9	1941	200
1458	1459	20	68.0	9717	5	6	1950	199
1459	1460	20	75.0	9937	5	6	1965	196

1460 rows × 38 columns

Datetime_Variables

In [39]:
```
year_feature=[item for item in numerical_features if 'Yr' in item or 'Year' in it
```

In [42]:
```
datafeame[year_feature]
```

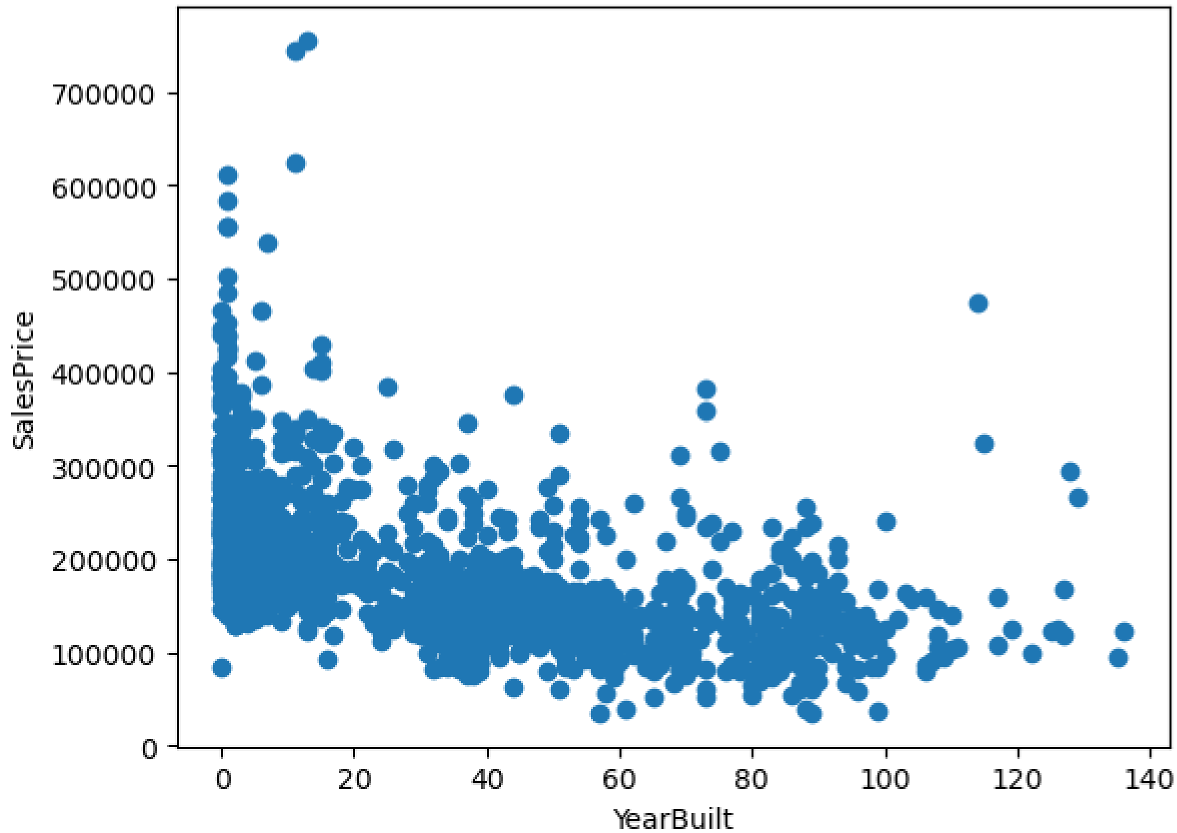
Out[42]:

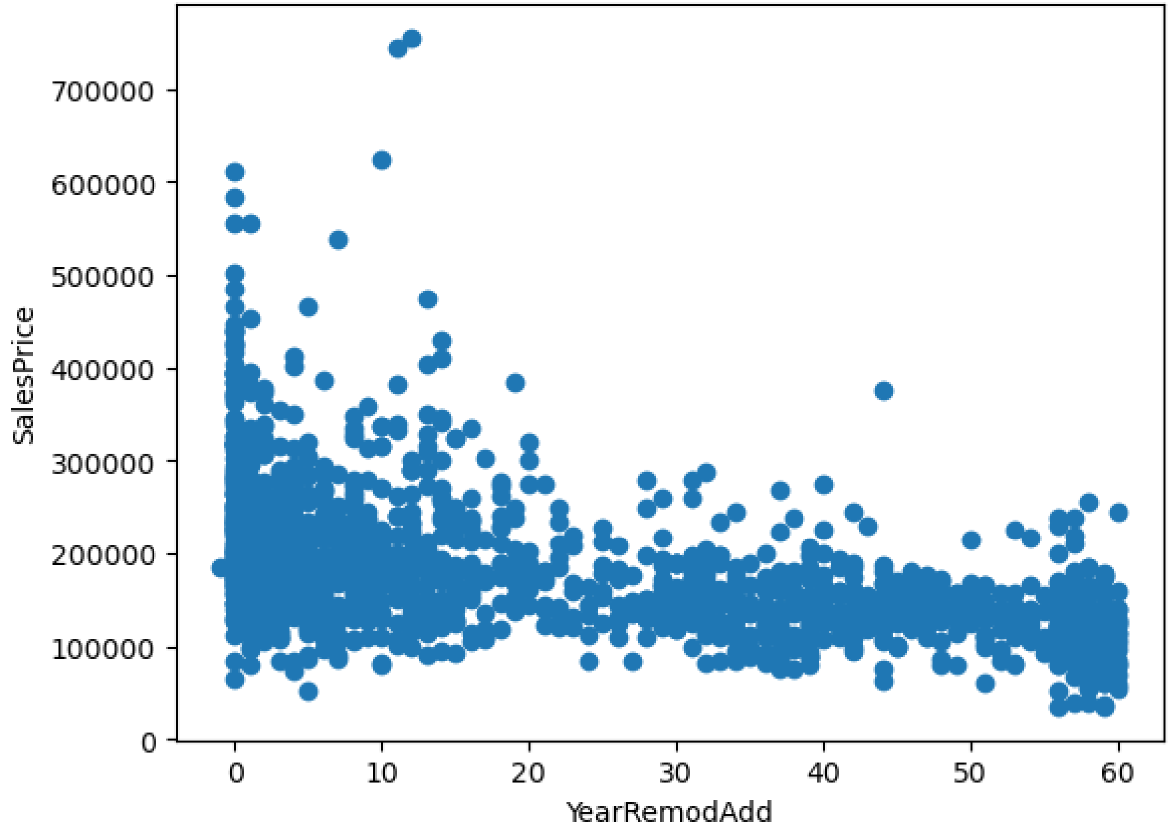
	YearBuilt	YearRemodAdd	GarageYrBlt	YrSold
0	2003	2003	2003.0	2008
1	1976	1976	1976.0	2007
2	2001	2002	2001.0	2008
3	1915	1970	1998.0	2006
4	2000	2000	2000.0	2008
...	...	...	...	...
1455	1999	2000	1999.0	2007
1456	1978	1988	1978.0	2010
1457	1941	2006	1941.0	2010
1458	1950	1996	1950.0	2010
1459	1965	1965	1965.0	2008

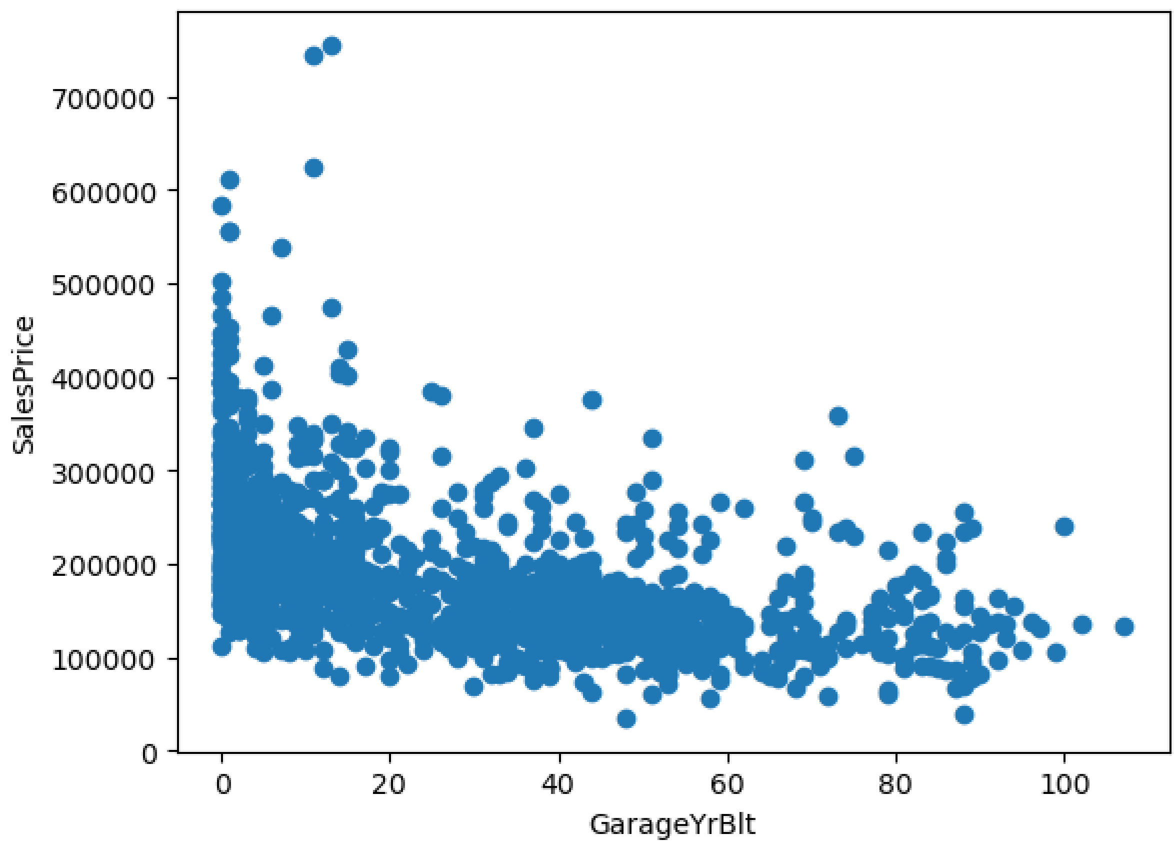
1460 rows × 4 columns

The relationship between year_feature and sale price

In [48]:
```
for feature in year_feature:
    if feature!='YrSold' :
        data=datafeame.copy()
        ##We will capture the difference between year variable and year the house
        data[feature]=data['YrSold']-data[feature]

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







In [49]:
```
##Numerical variables are usually of 2 types
## continous variable and discrete variables
```

In [50]:
```
numerical_features
```

Out[50]:
```
['Id',
 'MSSubClass',
 'LotFrontage',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'YearRemodAdd',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtFinSF2',
 'BsmtUnfSF',
 'TotalBsmtSF',
 '1stFlrSF',
 '2ndFlrSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'BsmtHalfBath',
 'FullBath',
 'HalfBath',
 'BedroomAbvGr',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageYrBlt',
 'GarageCars',
 'GarageArea',
 'WoodDeckSF',
 'OpenPorchSF',
 'EnclosedPorch',
 '3SsnPorch',
 'ScreenPorch',
 'PoolArea',
 'MiscVal',
 'MoSold',
 'YrSold',
 'SalePrice']
```

In [51]:
```
discrete_feature=[feature for feature in numerical_features if len(datafeame [fea
print('Discrete Variable Count :{}'.format (len(discrete_feature)))
```

```
Discrete Variable Count :17
```

In [52]:
```
datafeame[discrete_feature]
```

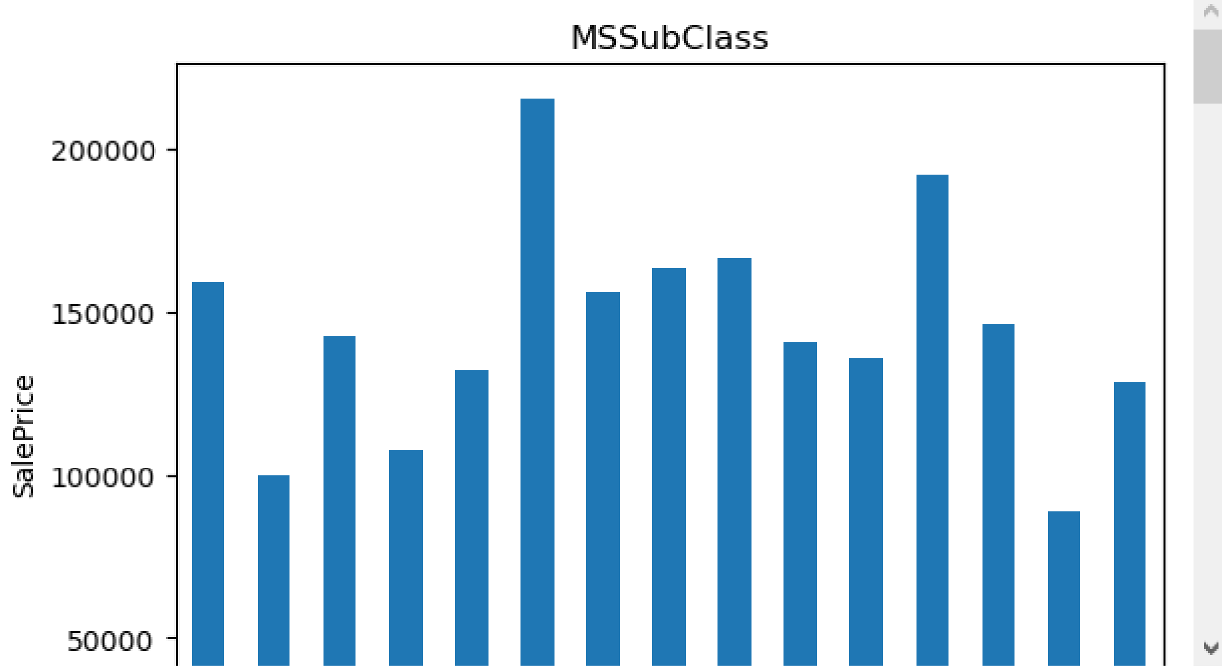
Out[52]:

	MSSubClass	OverallQual	OverallCond	LowQualFinSF	BsmtFullBath	BsmtHalfBath	FullBath
0	60	7	5	0	1	0	2
1	20	6	8	0	0	1	2
2	60	7	5	0	1	0	2
3	70	7	5	0	1	0	1
4	60	8	5	0	1	0	2
...	...	...	...	...	...	...	...
1455	60	6	5	0	0	0	2
1456	20	6	6	0	1	0	2
1457	70	7	9	0	0	0	2
1458	20	5	6	0	1	0	1
1459	20	5	6	0	1	0	1

1460 rows × 17 columns

The relationship between discrete variables and sale price

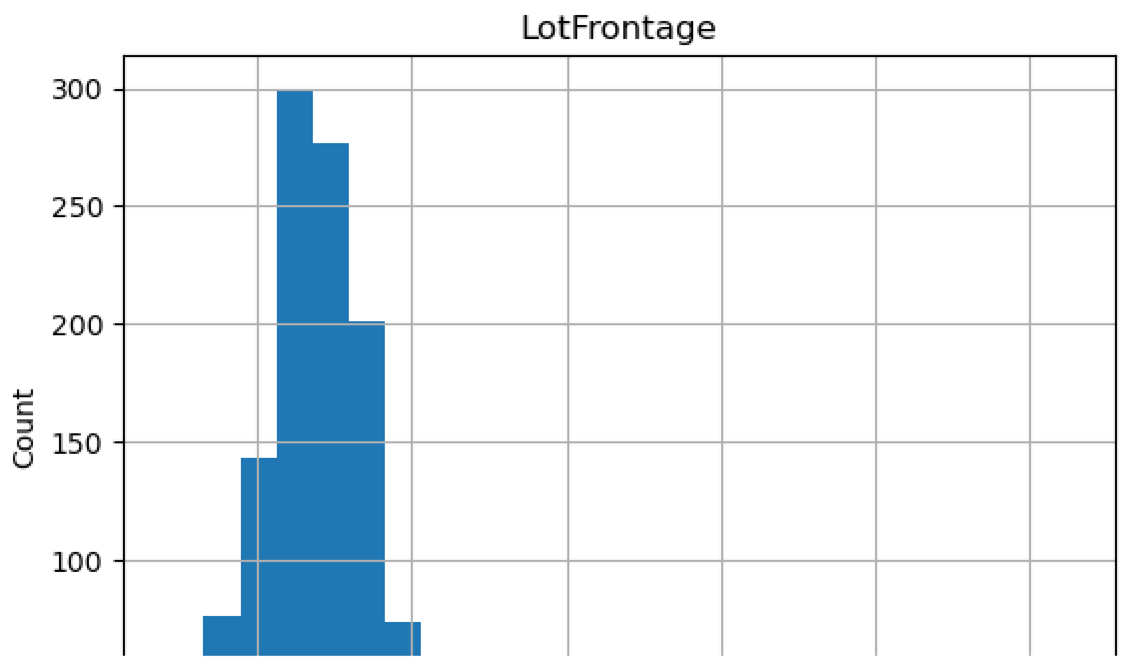
In [56]:
```
#let us find the relationship between discrete variables and sale price

for feature in discrete_feature:
    data=datafeame.copy()
    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.title(feature)
    plt.show()
```



In [57]:
```
continuous_feature=[feature for feature in numerical_features if feature not in d
print('Cintinuous feature Count :{}'.format (len(continuous_feature)))
```

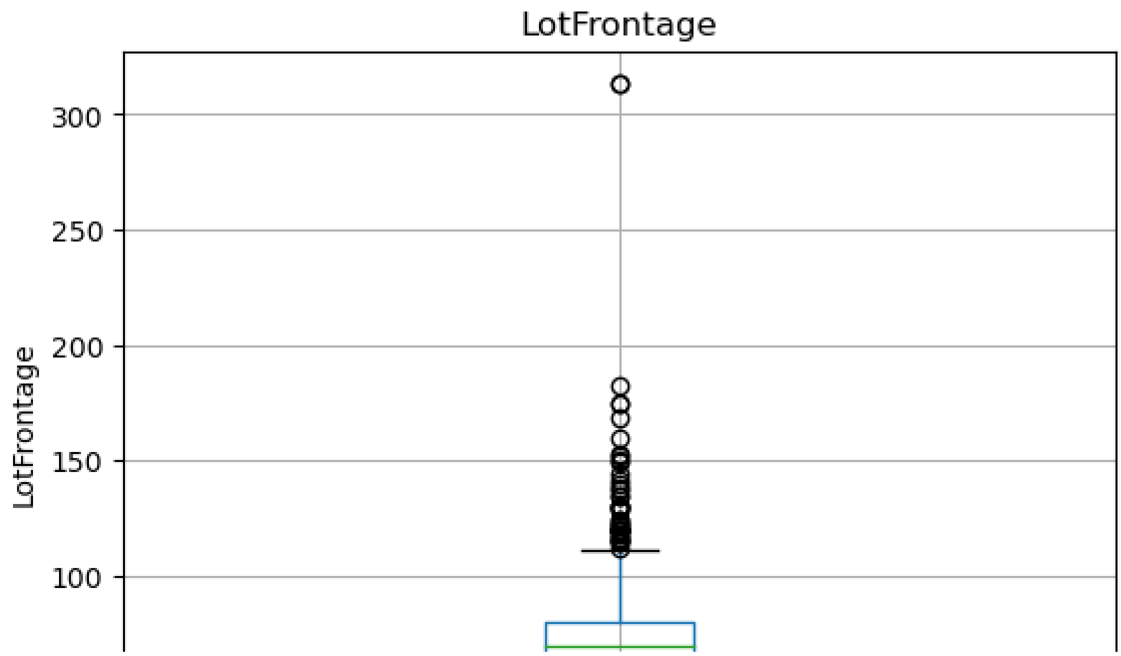
```
Cintinuous feature Count :16
```

In [61]:
```
#let us analyse the continuous values by creating histograms to understand the di
for feature in continuous_feature:
    data=datafeame.copy()
    data[feature].hist(bins=25)
    plt.xlabel(feature)
    plt.ylabel('Count')
    plt.title(feature)
    plt.show()
```



Outliers

In [64]:
```
for feature in continuous_feature:
    data=datafeame.copy()
    data.boxplot(column=feature)
    plt.ylabel(feature)
    plt.title(feature)
    plt.show()
```



In [67]:
```
for feature in numerical_features:
    data=datafeame.copy()
    data.boxplot(column=feature)
    plt.ylabel(feature)
    plt.title(feature)
    plt.show()
```

Categorical Variables

In [71]:
```
categorical_features=[feature for feature in datafeame .columns if data[feature].
```

In [72]:
```
datafeame[categorical_features]
```

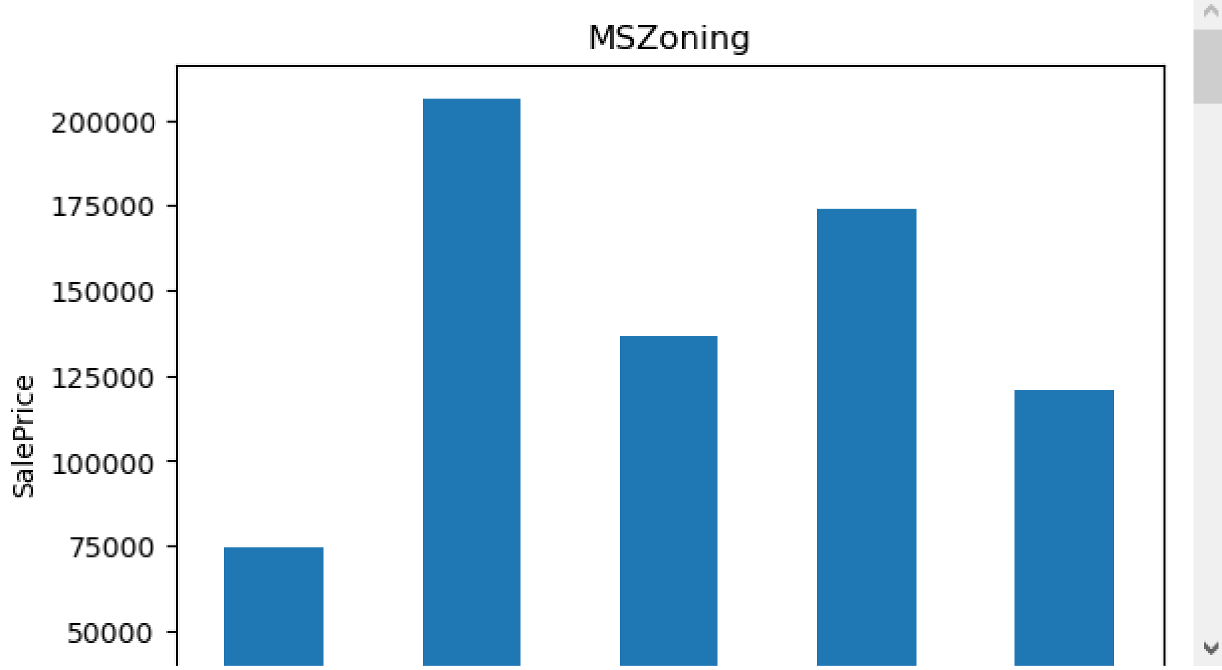
Out[72]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborh
0	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Coll
1	RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veer
2	RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Coll
3	RL	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crav
4	RL	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRi
...	...	...	...	...	...	...	...	...	
1455	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Gill
1456	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NWAr
1457	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Crav
1458	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAr
1459	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Edwa

1460 rows × 43 columns

The relationship between categorical variables and sale price

In [74]:
```
for feature in categorical_features:
    data=datafeame.copy()
    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.title(feature)
    plt.show()
```



How to deal with missing values

In [75]:
```
features_nan=[feature for feature in datafeame.columns if datafeame[feature].isnu
```

In [76]:
```
datafeame[features_nan]
```

Out[76]:

	Alley	MasVnrType	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	F
0	NaN	BrkFace	Gd	TA	No	GLQ	Unf	
1	NaN	NaN	Gd	TA	Gd	ALQ	Unf	
2	NaN	BrkFace	Gd	TA	Mn	GLQ	Unf	
3	NaN	NaN	TA	Gd	No	ALQ	Unf	
4	NaN	BrkFace	Gd	TA	Av	GLQ	Unf	
...	...	...	...	...	...	...	...	
1455	NaN	NaN	Gd	TA	No	Unf	Unf	
1456	NaN	Stone	Gd	TA	No	ALQ	Rec	
1457	NaN	NaN	TA	Gd	No	GLQ	Unf	
1458	NaN	NaN	TA	TA	Mn	GLQ	Rec	
1459	NaN	NaN	TA	TA	No	BLQ	LwQ	

1460 rows × 15 columns

In [77]:
```
for feature in features_nan:
    print('{}:{} % missing values'.format(feature,np.round(datafeame[feature].isr
```

```
Alley:0.94 % missing values
MasVnrType:0.6 % missing values
BsmtQual:0.03 % missing values
BsmtCond:0.03 % missing values
BsmtExposure:0.03 % missing values
BsmtFinType1:0.03 % missing values
BsmtFinType2:0.03 % missing values
FireplaceQu:0.47 % missing values
GarageType:0.06 % missing values
GarageFinish:0.06 % missing values
GarageQual:0.06 % missing values
GarageCond:0.06 % missing values
PoolQC:1.0 % missing values
Fence:0.81 % missing values
MiscFeature:0.96 % missing values
```

In [80]:
```
def rep_nan_values(datafeame,features_nan):
    data=datafeame.copy()
    data[features_nan]=data[features_nan].fillna('Missing_val')
    return data
```

In [81]:
```
datafeame=rep_nan_values(datafeame,features_nan)
```

In [82]:
```
datafeame
```

Out[82]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandCon
0	1	60	RL	65.0	8450	Pave	Missing_val	Reg	
1	2	20	RL	80.0	9600	Pave	Missing_val	Reg	
2	3	60	RL	68.0	11250	Pave	Missing_val	IR1	
3	4	70	RL	60.0	9550	Pave	Missing_val	IR1	
4	5	60	RL	84.0	14260	Pave	Missing_val	IR1	
...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	Missing_val	Reg	
1456	1457	20	RL	85.0	13175	Pave	Missing_val	Reg	
1457	1458	70	RL	66.0	9042	Pave	Missing_val	Reg	
1458	1459	20	RL	68.0	9717	Pave	Missing_val	Reg	
1459	1460	20	RL	75.0	9937	Pave	Missing_val	Reg	

1460 rows × 81 columns

In [84]:
```
numerical_with_nan=[feature for feature in datafeame.columns if datafeame[feature
```

In [85]:
```
datafeame[numerical_with_nan]
```

Out[85]:

	LotFrontage	MasVnrArea	GarageYrBlt
0	65.0	196.0	2003.0
1	80.0	0.0	1976.0
2	68.0	162.0	2001.0
3	60.0	0.0	1998.0
4	84.0	350.0	2000.0
...	...	...	...
1455	62.0	0.0	1999.0
1456	85.0	119.0	1978.0
1457	66.0	0.0	1941.0
1458	68.0	0.0	1950.0
1459	75.0	0.0	1965.0

1460 rows × 3 columns

In [86]:
```
for feature in numerical_with_nan :
    print('{}:{} % missing values'.format(feature,np.round(datafeame[feature].isr
```

```
LotFrontage:0.1774 % missing values
MasVnrArea:0.0055 % missing values
GarageYrBlt:0.0555 % missing values
```

To replace nan values

In [87]:
```
for feature in numerical_with_nan:
    #we will replace by using median since there are outliers
    median_value=datafeame[feature].median()
    #create a new feature to capture nan values
    datafeame[feature +'nan']=np.where(datafeame[feature].isnull(),1,0)
    datafeame[feature].fillna(median_value,inplace=True)
```

In [88]:
```
datafeame[numerical_with_nan].isnull().sum()
```

Out[88]:
```
LotFrontage    0
MasVnrArea     0
GarageYrBlt    0
dtype: int64
```

Replacing string values to int values

In [89]:
```
datafeame=pd.get_dummies(datafeame)
```

In [90]:
```
datafeame
```

Out[90]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAd
0	1	60	65.0	8450	7	5	2003	200
1	2	20	80.0	9600	6	8	1976	197
2	3	60	68.0	11250	7	5	2001	200
3	4	70	60.0	9550	7	5	1915	197
4	5	60	84.0	14260	8	5	2000	200
...	...	...	...	...	...	...	...	.
1455	1456	60	62.0	7917	6	5	1999	200
1456	1457	20	85.0	13175	6	6	1978	198
1457	1458	70	66.0	9042	7	9	1941	200
1458	1459	20	68.0	9717	5	6	1950	199
1459	1460	20	75.0	9937	5	6	1965	196

1460 rows × 307 columns

In [ ]: