



HOUSING PRICE PREDICTION



Submitted by:

ARUN KUMAR M

ACKNOWLEDGMENT

This project includes the professional reference of much external research analysis done by various organizations and individuals.

Such references are mentioned below:

1. <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154>.
2. Housing Sale Price Prediction Using Machine Learning Algorithms.
Author(s): Zhou, Yichen Advisor(s): Wu, Yingnian.
<https://escholarship.org/uc/item/3ft2m7z5>.
3. <https://ieeexplore.ieee.org/document/8473231/references#references>. House Price Prediction Using Machine Learning and Neural Networks.
4. Predicting House Price Using Regression Algorithm for Machine Learning.
<https://yalantis.com/blog/predictive-algorithm-for-house-price/>.

INTRODUCTION

1. Business Problem Framing :

We are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high

returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

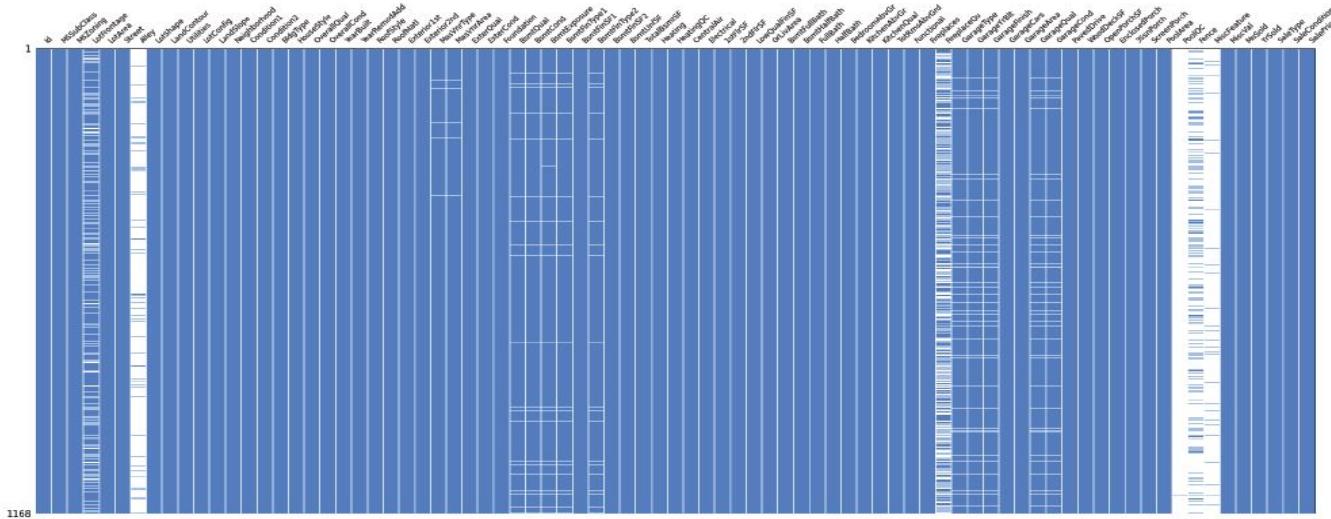
- **Conceptual Background of the Domain Problem :**

1. Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.
2. A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file.
3. The company is looking at prospective properties to buy houses to enter the market. We are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not.

Analytical Problem Framing

2. Mathematical/ Analytical Modelling of the Problem :

1. This is a **Linear Regression** Problem.
2. Mean, Median and Mode of each continuous column was calculated to get an overview of the data.
3. Data's which didn't provide any feature importance are removed from the data-set.
4. Skewed columns are treated with **Log Transform**.



5. General statistical data is obtained to perform imputation and visualization.
 6. The data is divided into categorical and numerical variables during the analysis.

3. Data Sources and their formats :

1. The Data-set seems to be divided into **2 sets**, one for training and one for testing.
 2. The **train data-set** seems to have **1168 rows and 81 columns**.
 3. The **test data-set** seems to have **292 rows and 80 columns**.
 4. Both the files were given as **csv files**.
 5. The data-set contains 3- float data types, 35-int data types and 43-object data types.

4. Data Preprocessing Done :

1. The columns which didn't give any feature importance have been dropped.
 2. The Null values are found and visualized using missing package as follows.
 3. We could see that the columns:
 - 1.'LotFrontage'
 - 2.'Alley'
 - 3.'MasVnrType'
 - 4.'MasVnrArea'
 - 5.'BsmtQual'
 - 6.'ExterCond'

7.'BsmtExposure'
8.'BsmtFinType1'
9.'BsmtFinType2'
10.'FireplaceQu'
11.'GarageType'
12.'GarageYrBlt'
13.'GarageFinish'
14.'GarageQual'
15.'GarageCond'
16.'PoolQC'
17.'Fence'
18.'MiscFeature'
are having Nan Values present in them.

4. so we will see the percentage of the Nan values in each columns.

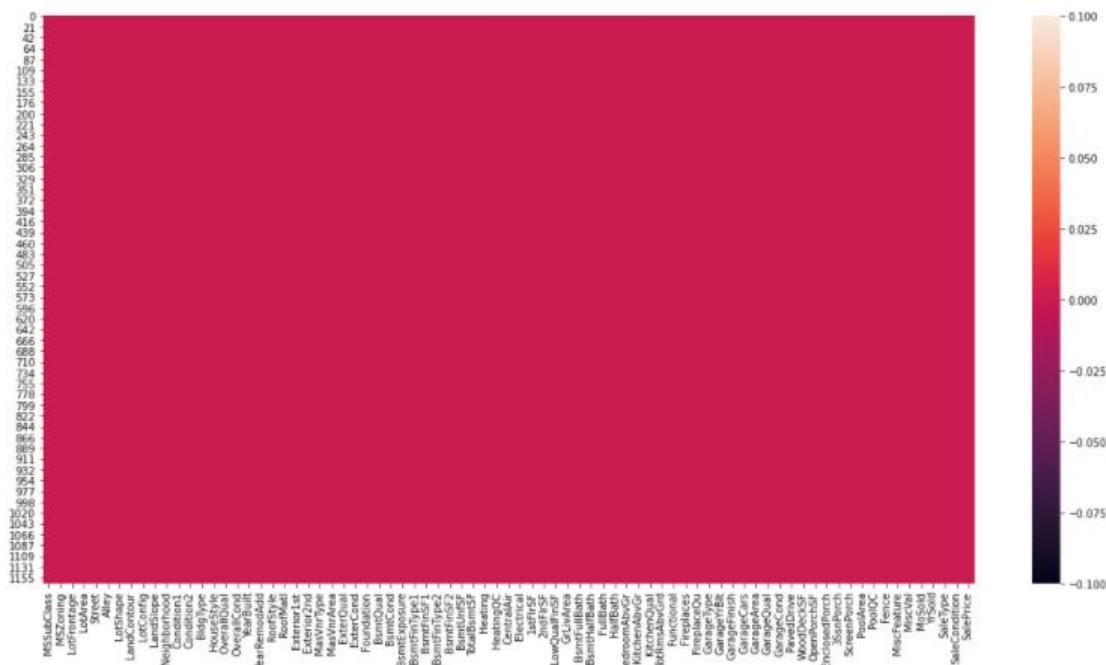
LotFrontage 18.32
Alley 93.41
MasVnrType 0.60
MasVnrArea 0.60
BsmtQual 2.57
BsmtCond 2.57
BsmtExposure 2.65
BsmtFinType1 2.57
BsmtFinType2 2.65
FireplaceQu 47.17
GarageType 5.48
GarageYrBlt 5.48
GarageFinish 5.48
GarageQual 5.48
GarageCond 5.48
PoolQC 99.40
Fence 79.71
MiscFeature 96.23

5. Now we will try to impute the Nan values individually.

6. since the data-set contains both categorical and continuous data,
we will look into each column containing Nan values individually and
impute them accordingly. Also, some of the Nan values are not actually
null. They are missing some data entry, which is given in the Data-set
description.

5. Data Inputs- Logic- Output Relationships :

1. Sales price is the target variable, which varies linearly with most of the continuous variables.
2. KitchenAboveGr and Enclosed Porch varies negatively.
3. Target correlates most with OverallQual, YearBuilt, YearRemodAdd, TotalBsmtSF, 1stFloorSF, GrLlvArea, FullBath, TotRmsAbvGrd, GarageCars, GarageArea. Apart from MSSubClass, overallQual, BsmtFinSf2, LowQualFinSF, BsmtHalfBath, KitchenAbvGr, EnclosedPorchAnd YRsold all the other variables are positively corelated with the target variable. whereas the previously mentioned are negatively correlated.

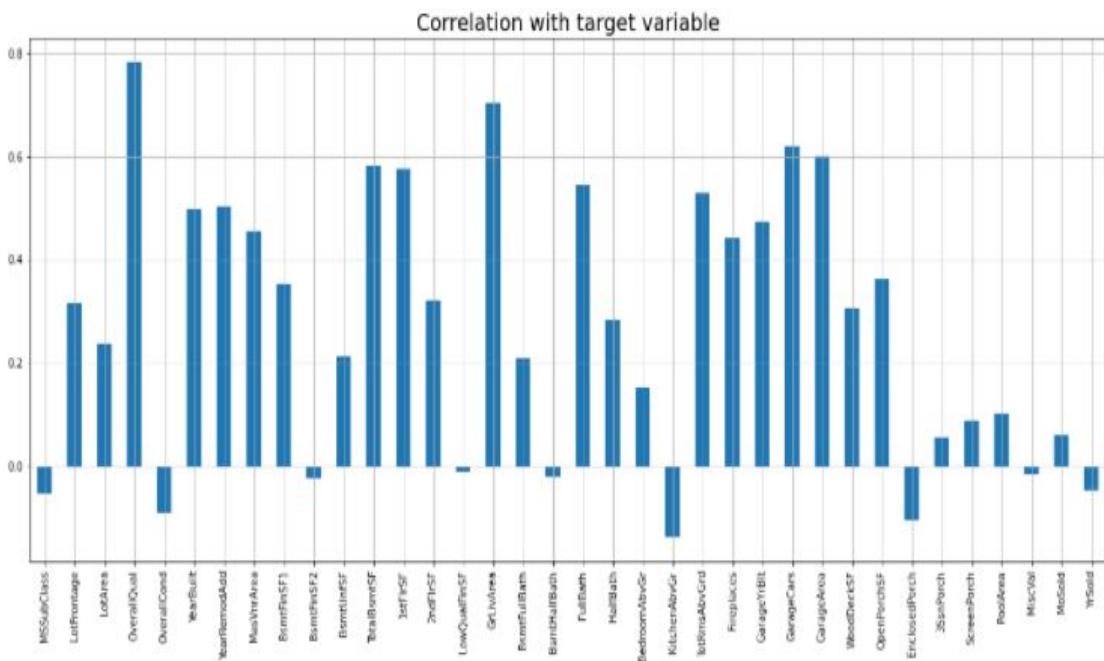


6. Hardware and Software Requirements and Tools Used :

1. **Software:** Jupyter Notebook.
2. **Hardware:** i5;16 gb ram
3. Dual monitor setup if possible (for easy manuvering through solutions).

Model/s Development and Evaluation

1. Identification of possible problem-solving approaches



(methods) :

1. Now for further Data exploration Process we will divide the dataset into Catagorical and continuous variables.

```

cat_List = [x for x in df_train.columns if df_train[x].dtype==object]
list (cat_List)

['MSZoning',
 'Street',
 'Alley',
 'Lotshape',
 'LandContour',
 'LotConfig',
 'LandSlope',
 'Neighborhood',
 'Condition1',
 'Condition2',
 'BldgType',
 'HouseStyle',
 'RoofStyle',
 'RoofMatl',
 'Exterior1st',
 'Exterior2nd',
 'MasVnrType',
 'ExterQual',
 'ExterCond',
 'Foundation',
 'BsmtQual',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'Heating',
 'HeatingQC',
 'CentralAir',
 'Electrical',
 'KitchenQual',
 'Functional',
 'FireplaceQu',
 'GarageType',
 'GarageFinish',
 'GarageQual',
 'GarageCond',
 'PavedDrive',
 'PoolQC',
 'Fence',
 'MiscFeature',
 'SaleType',
 'SaleCondition']

```

Continous Variables :

```

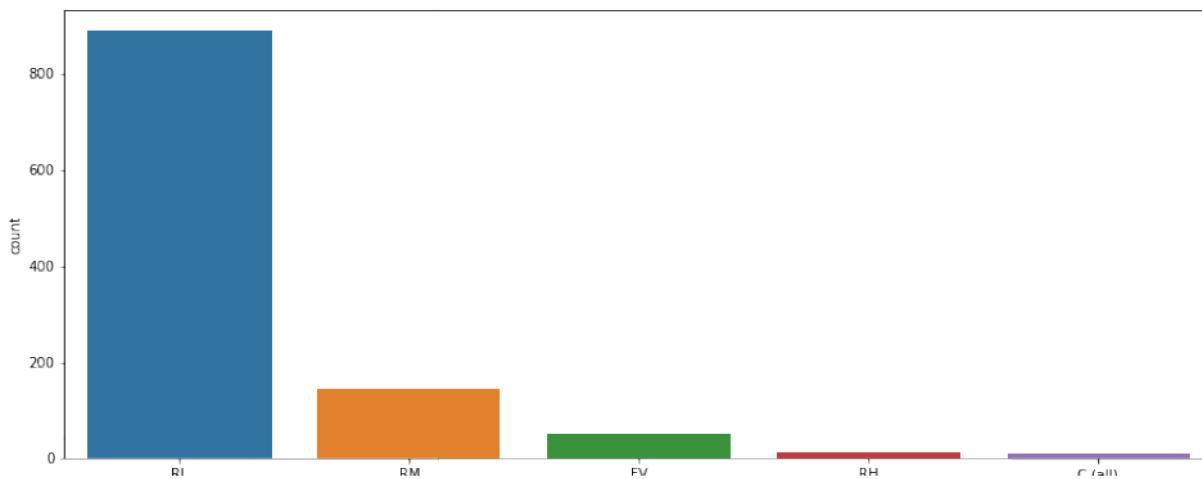
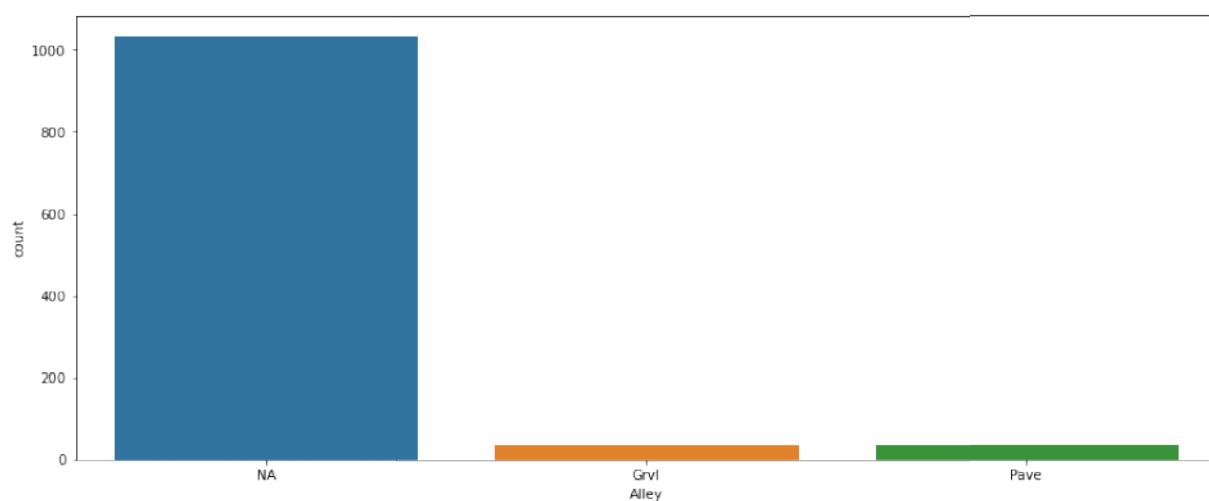
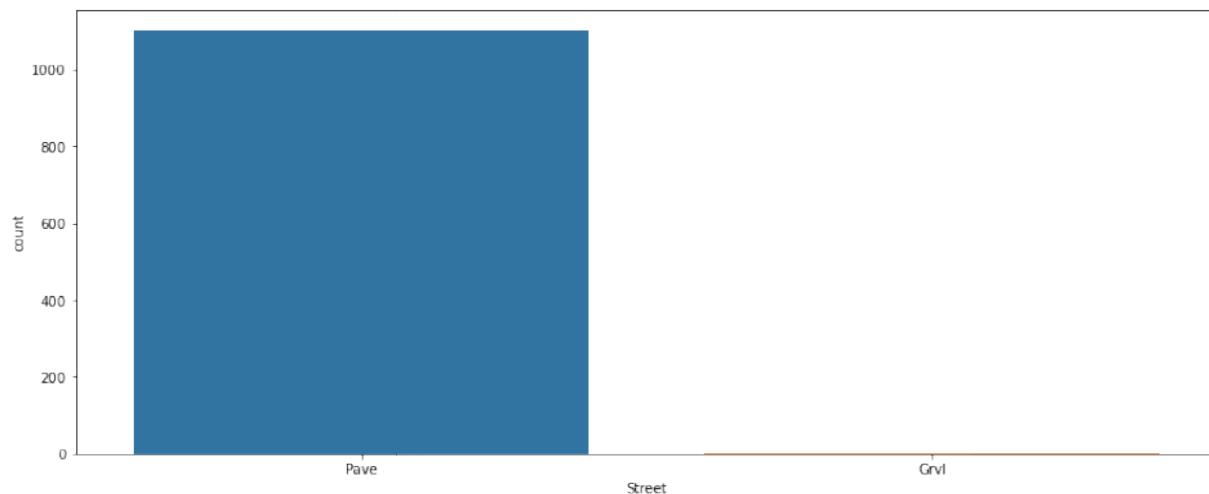
num_List = [x for x in df_train.columns if x not in cat_List]
list (num_List)

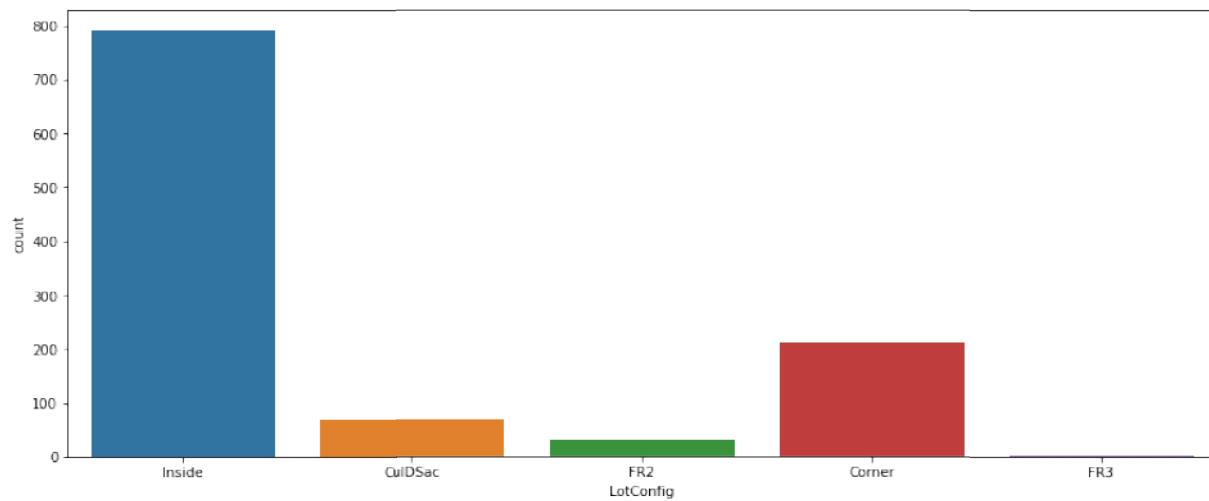
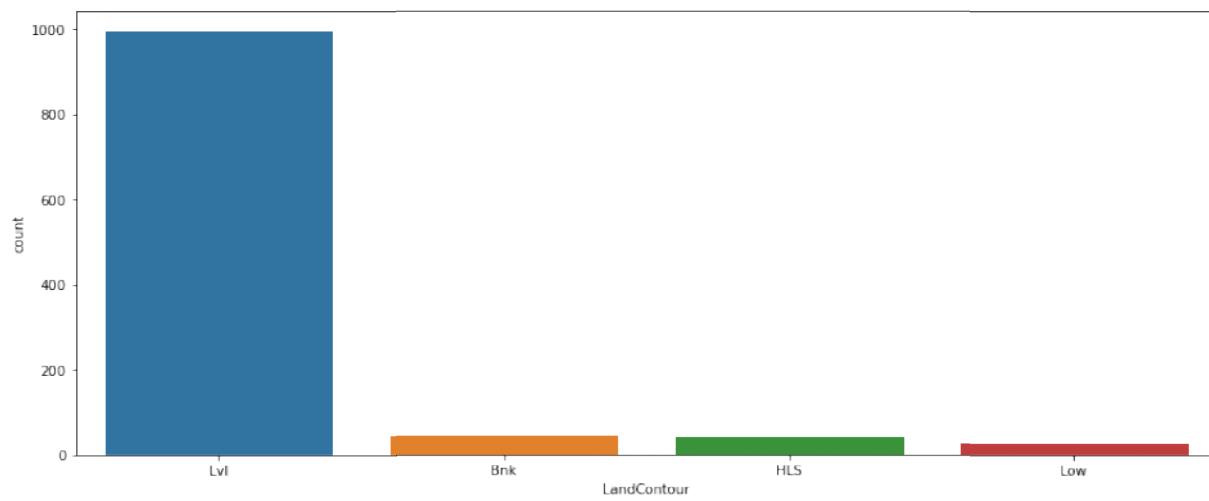
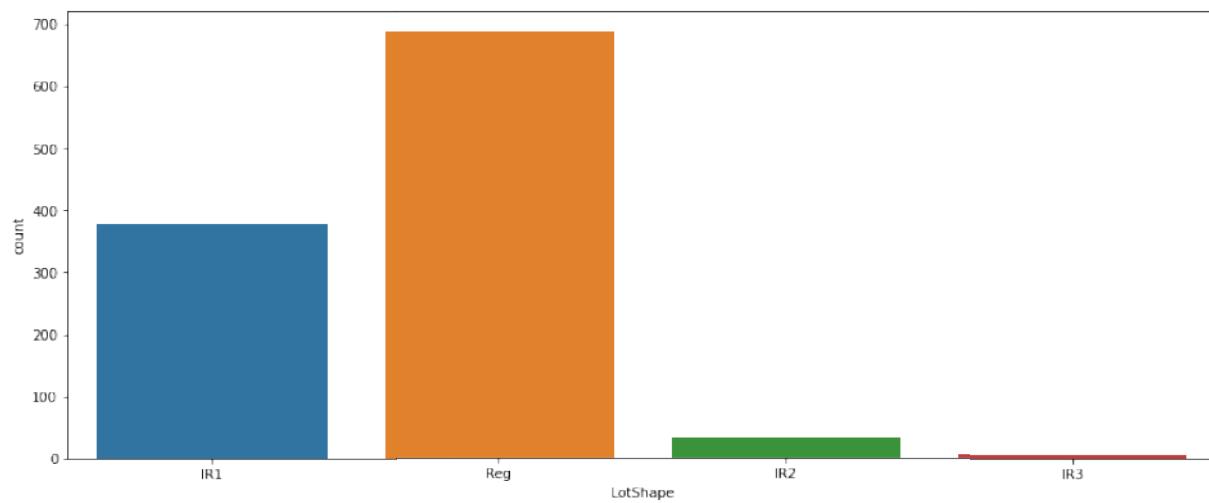
['MSSubClass',
 'LotFrontage',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'YearRemodAdd',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtFinSF2',
 'BsmtUnfSF',
 'TotalBsmtSF',
 '1stFlrSF',
 '2ndFlrSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'BsmtHalfBath',
 'FullBath',
 'HalfBath',
 'BedroomAbvGr',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

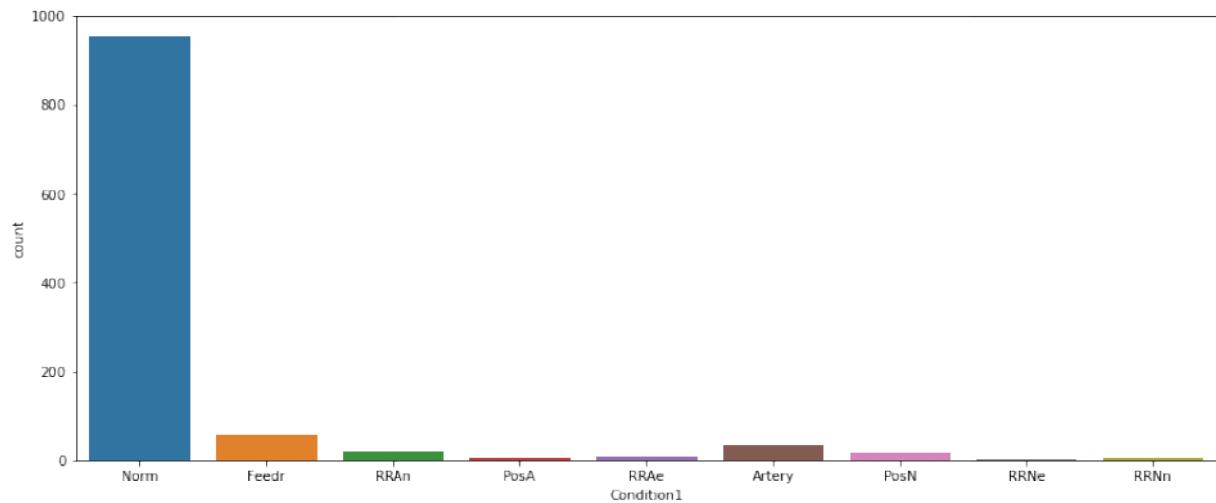
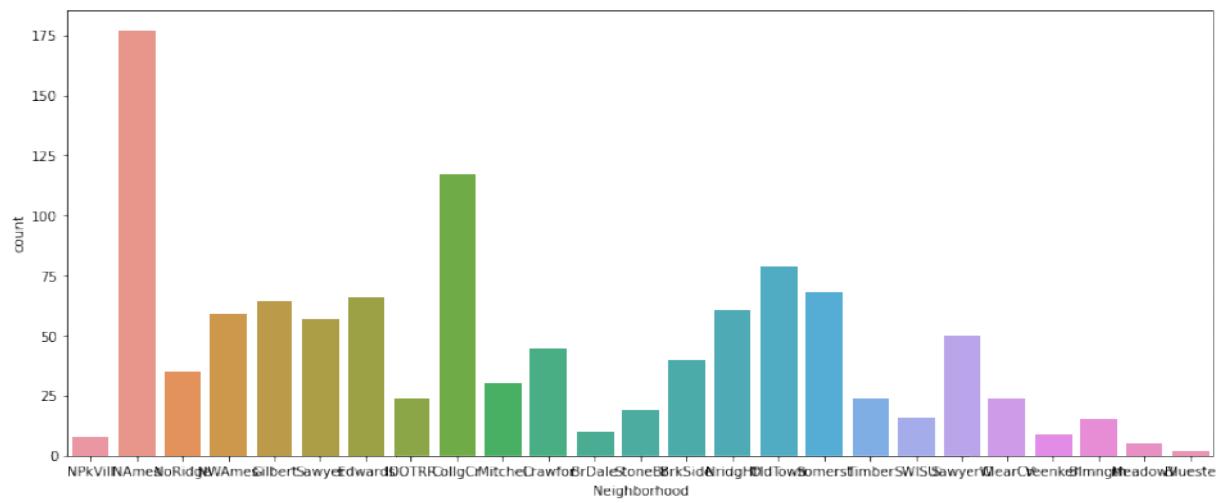
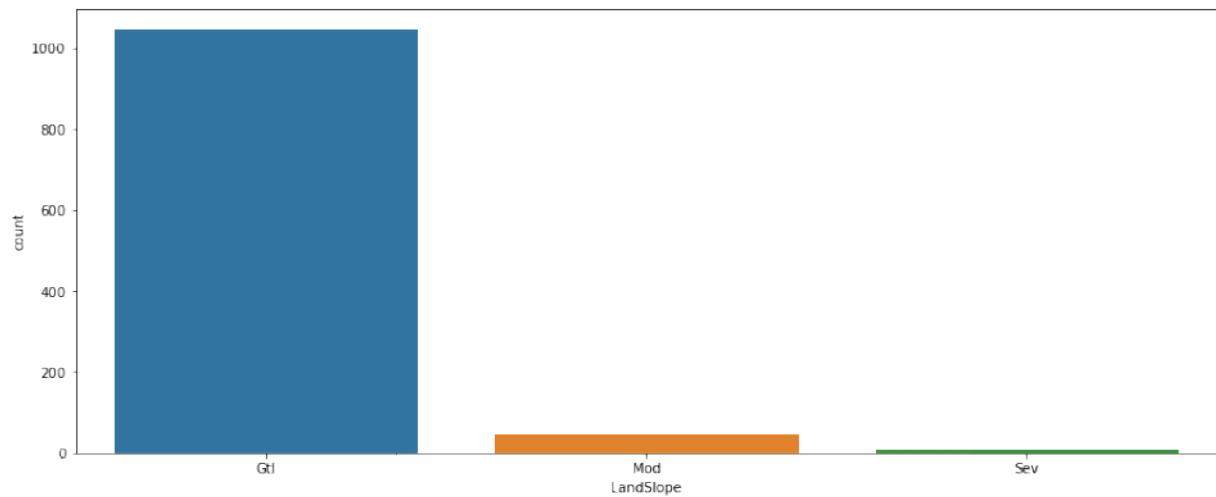
```

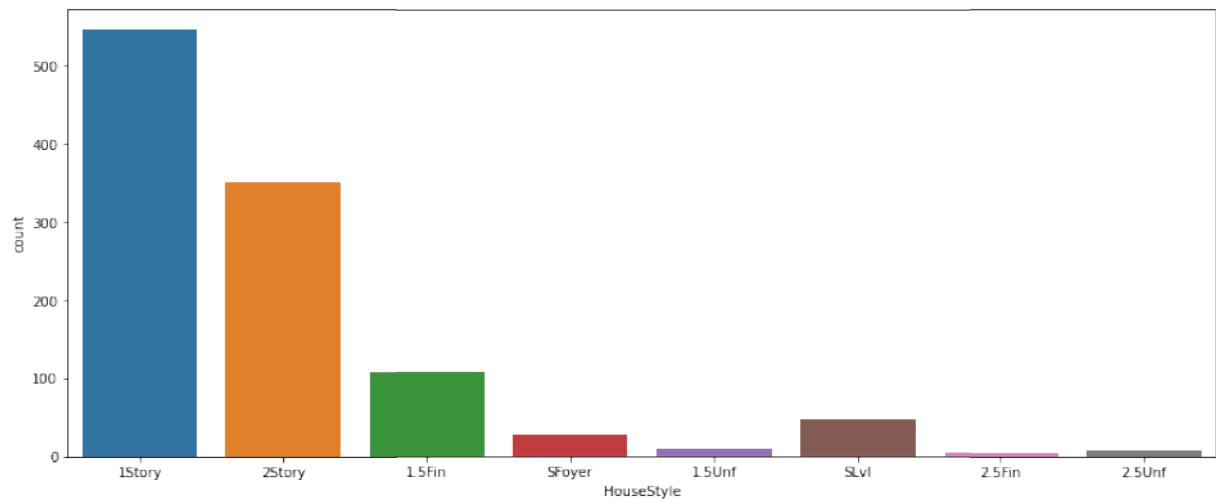
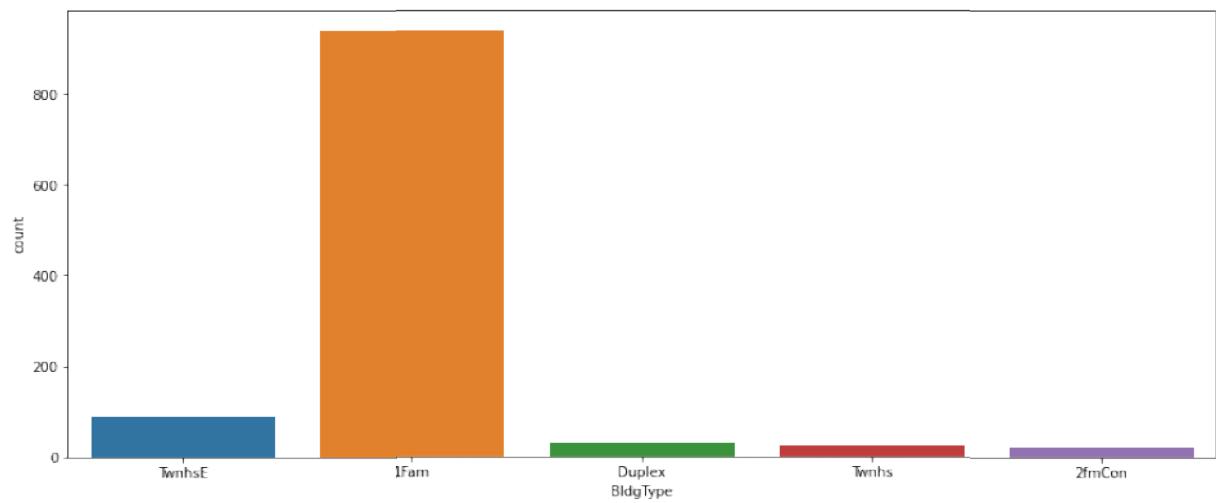
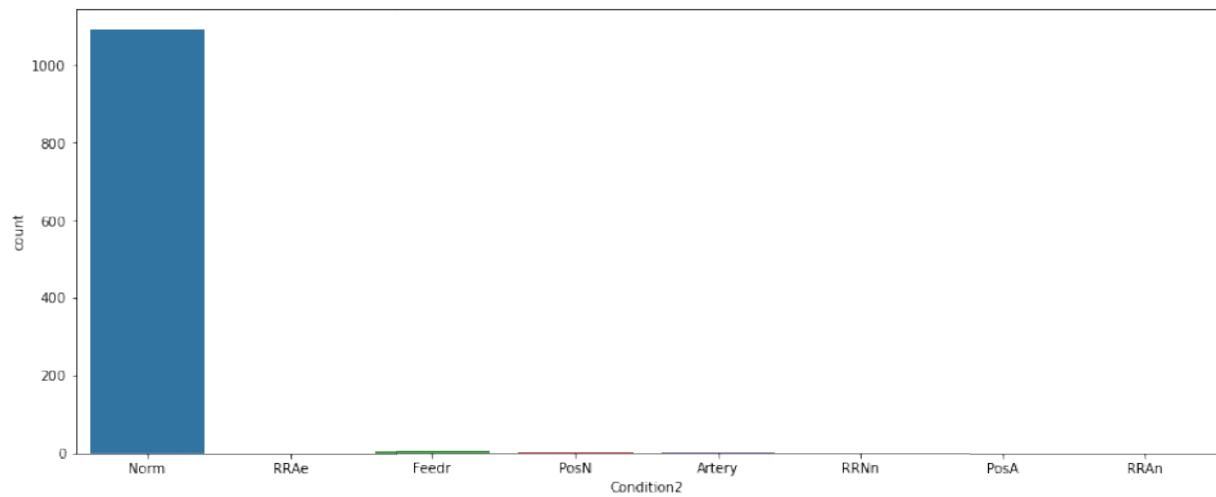
2. DATA VISUALIZATION :

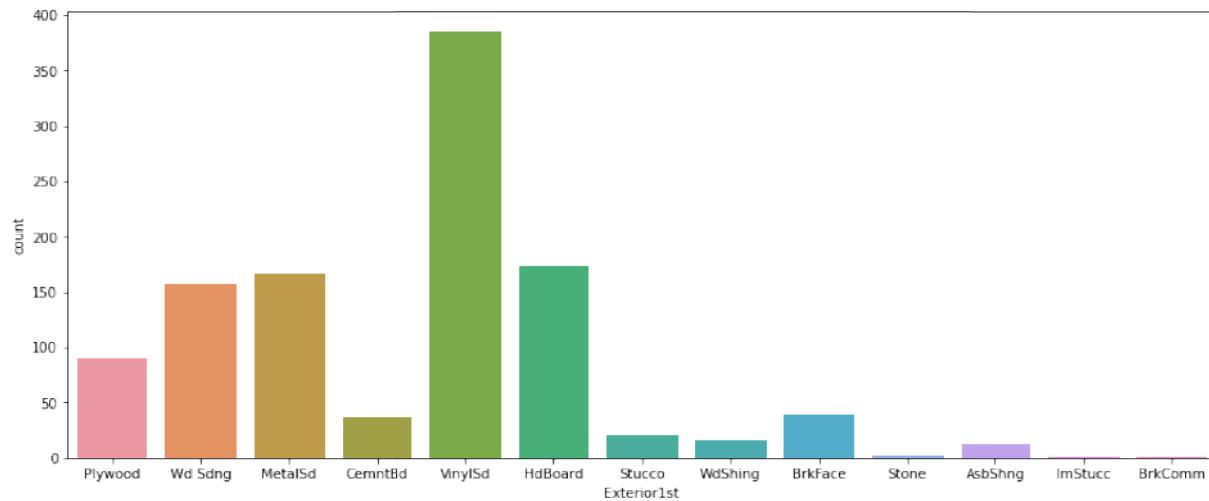
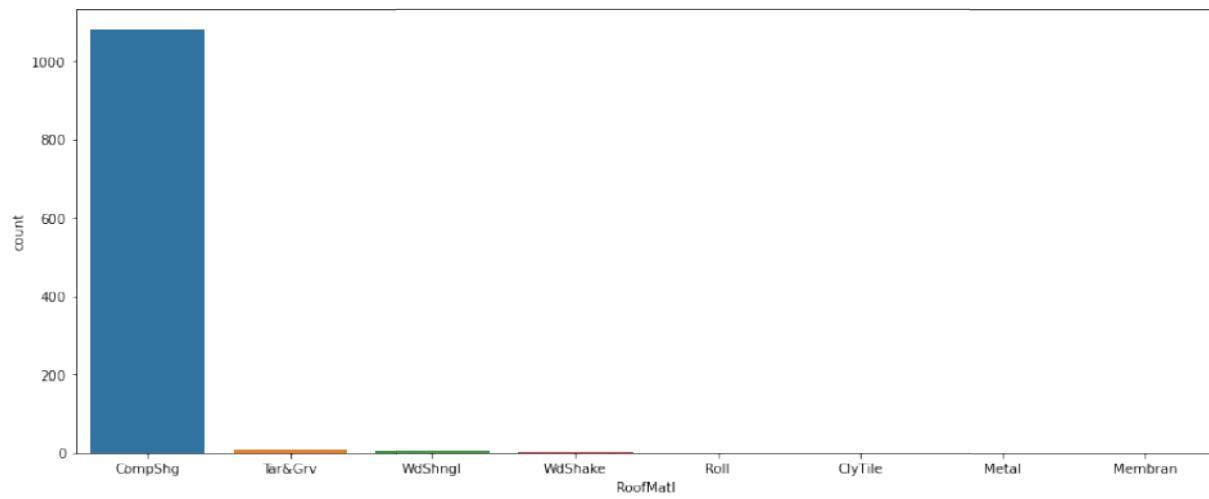
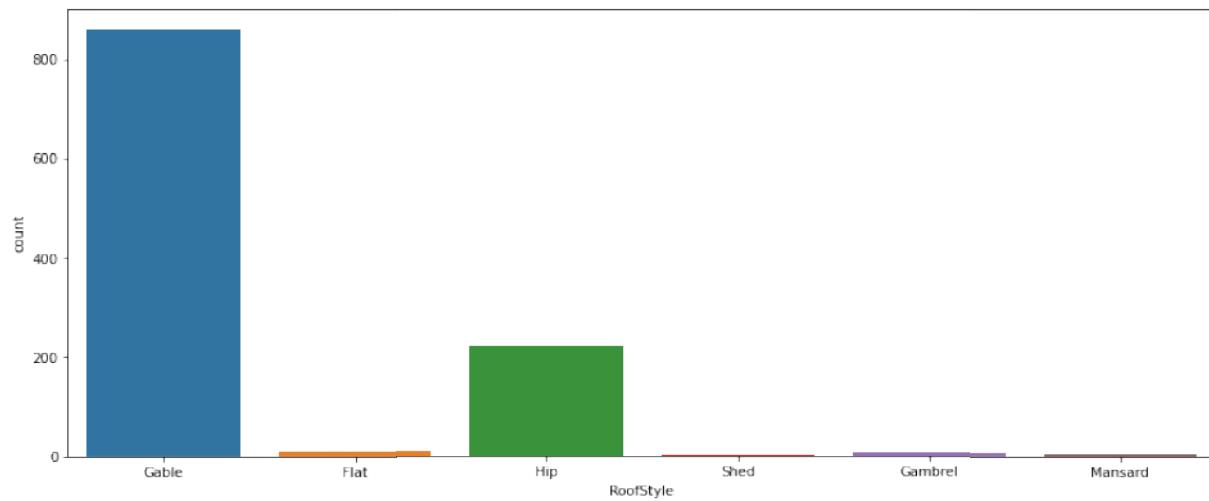
UNI-VARIATE ANALYSIS

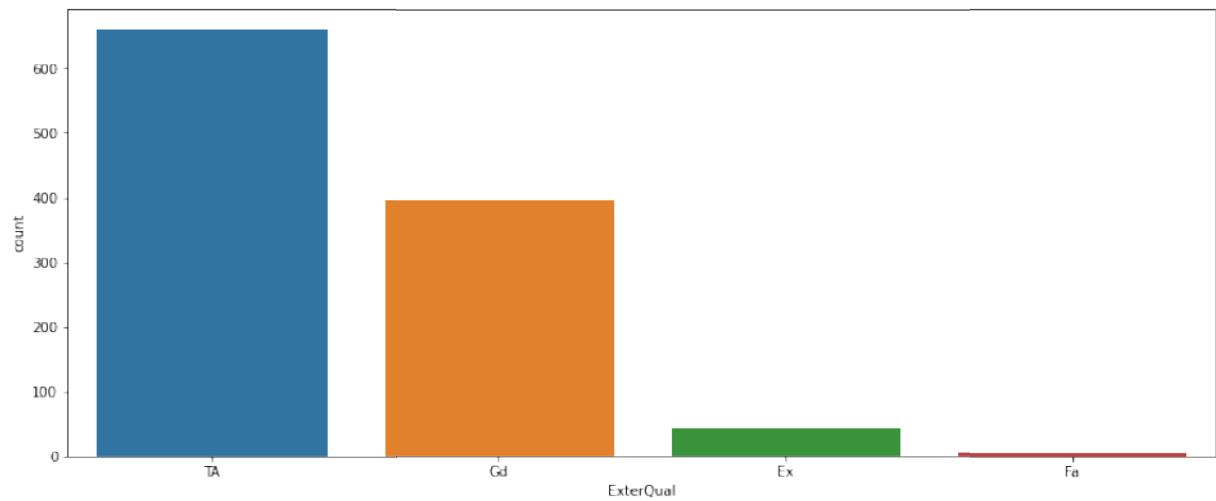
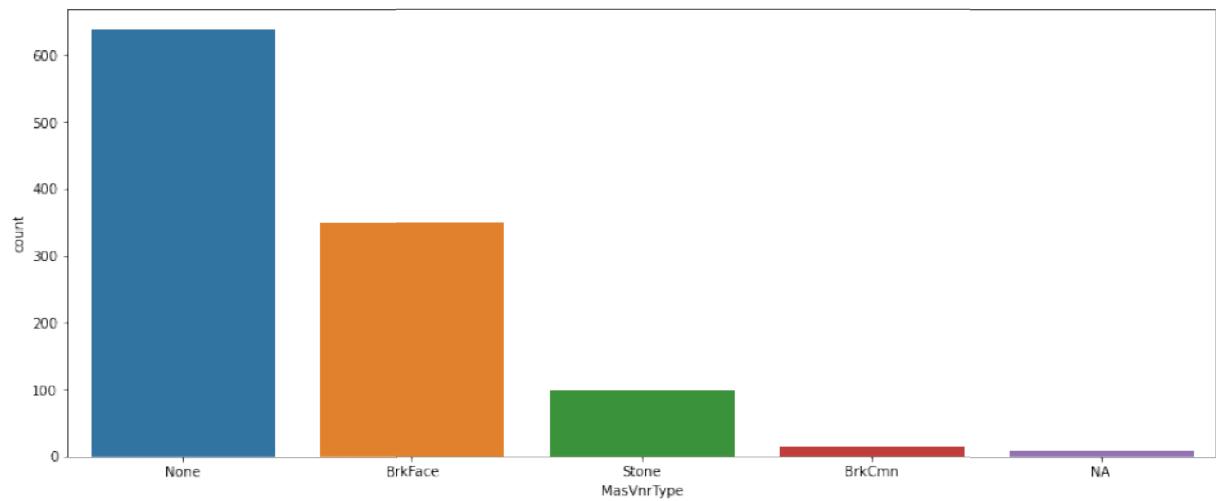
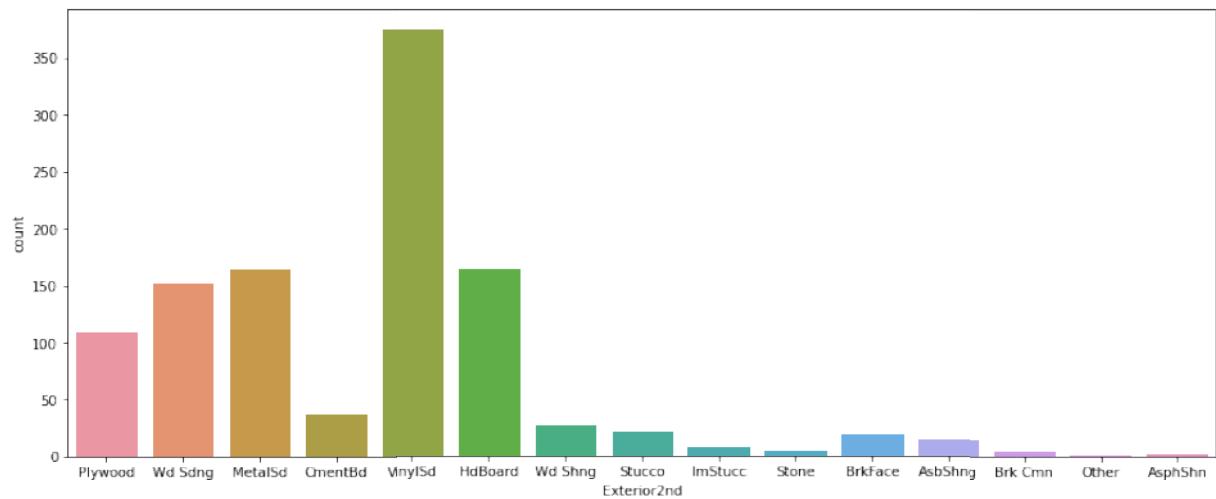


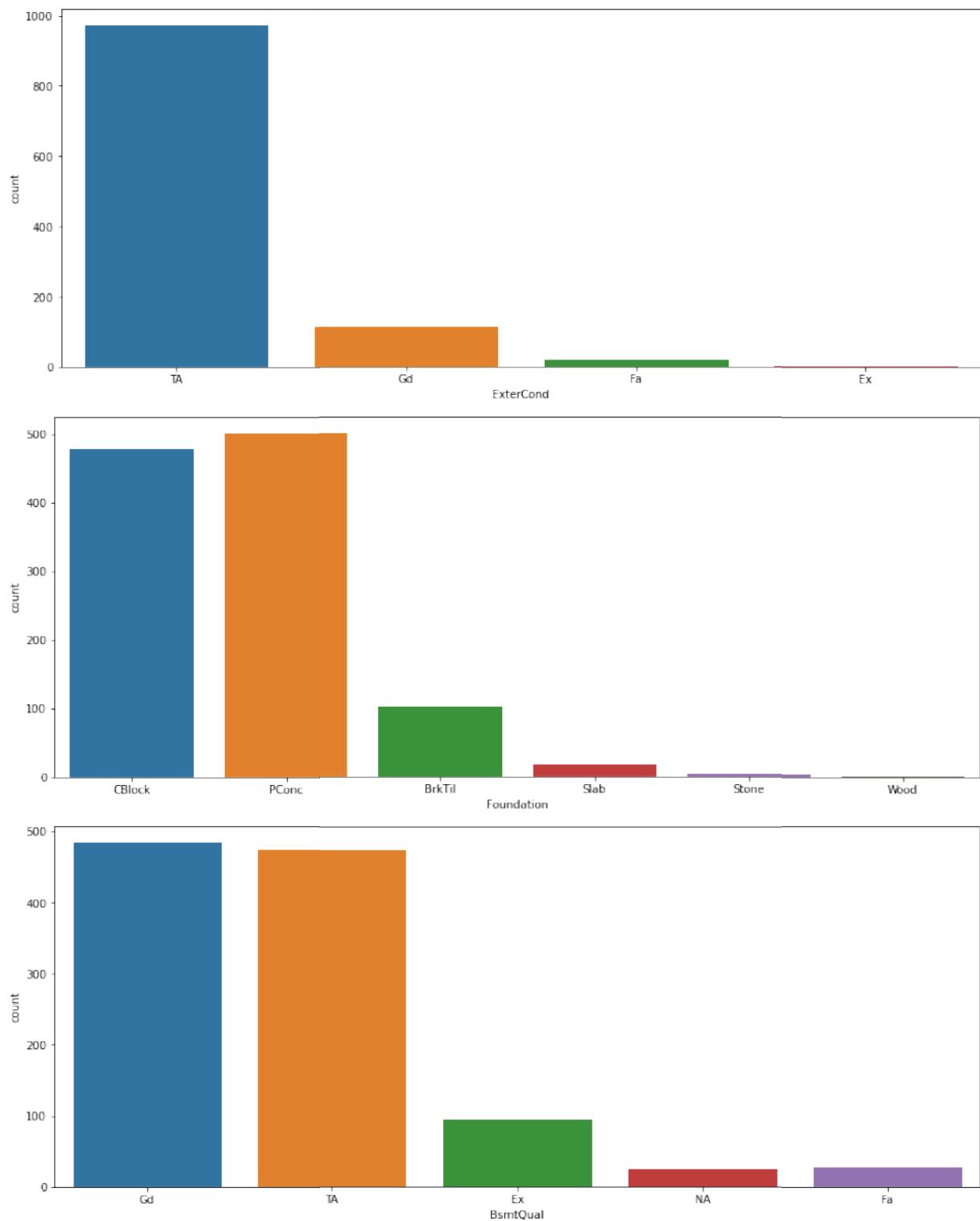


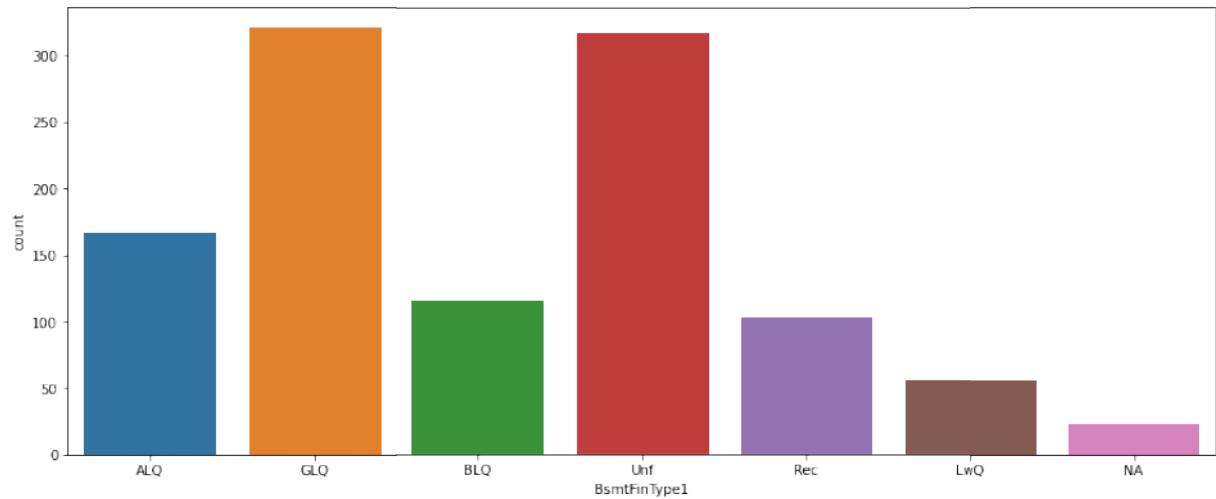
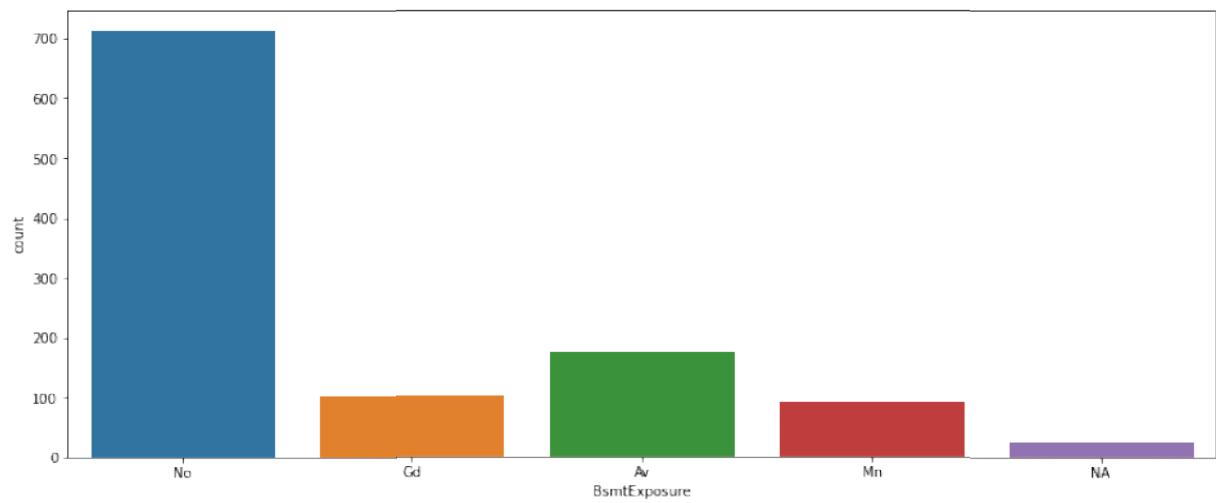
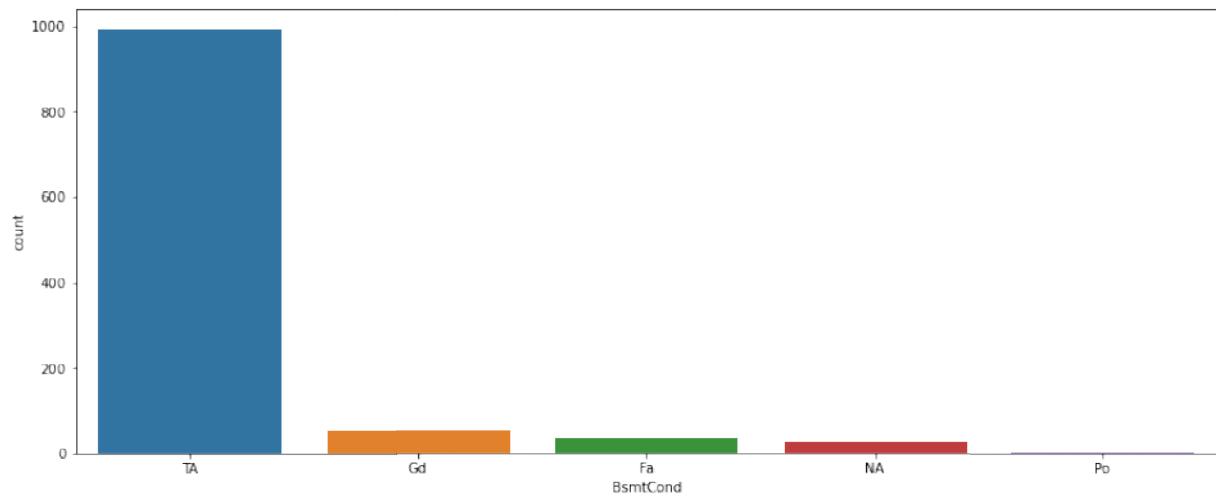


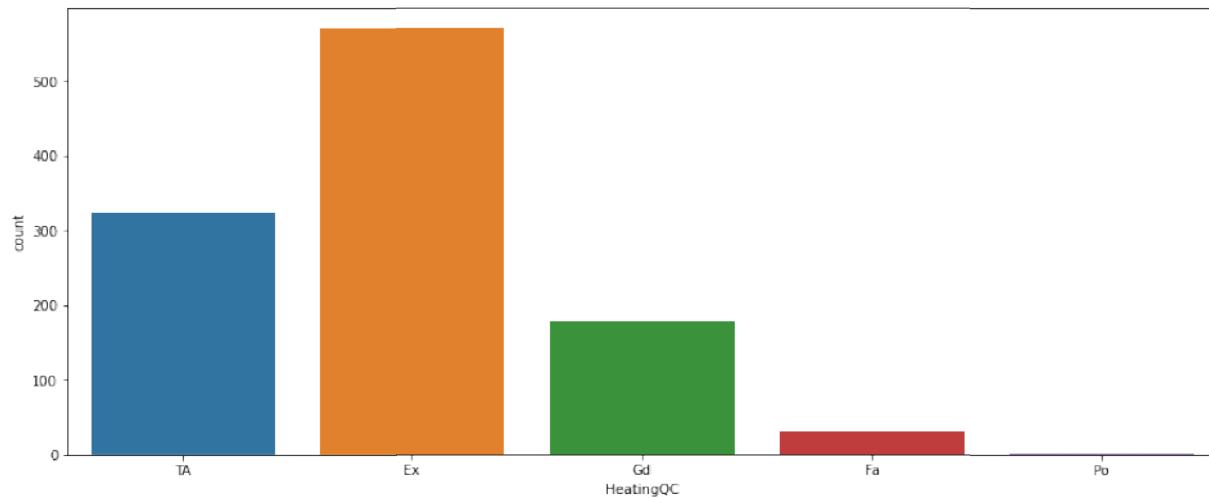
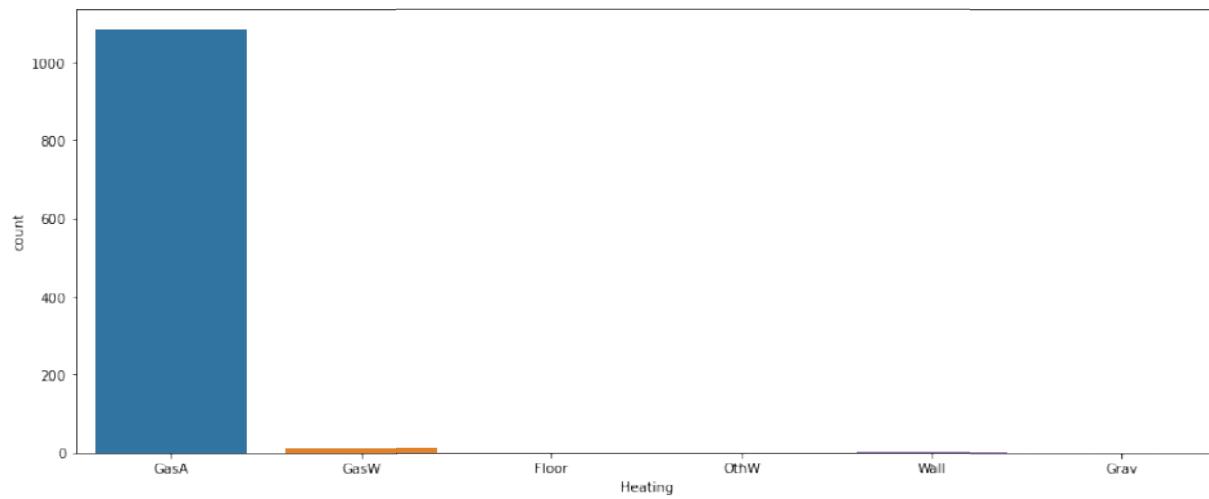
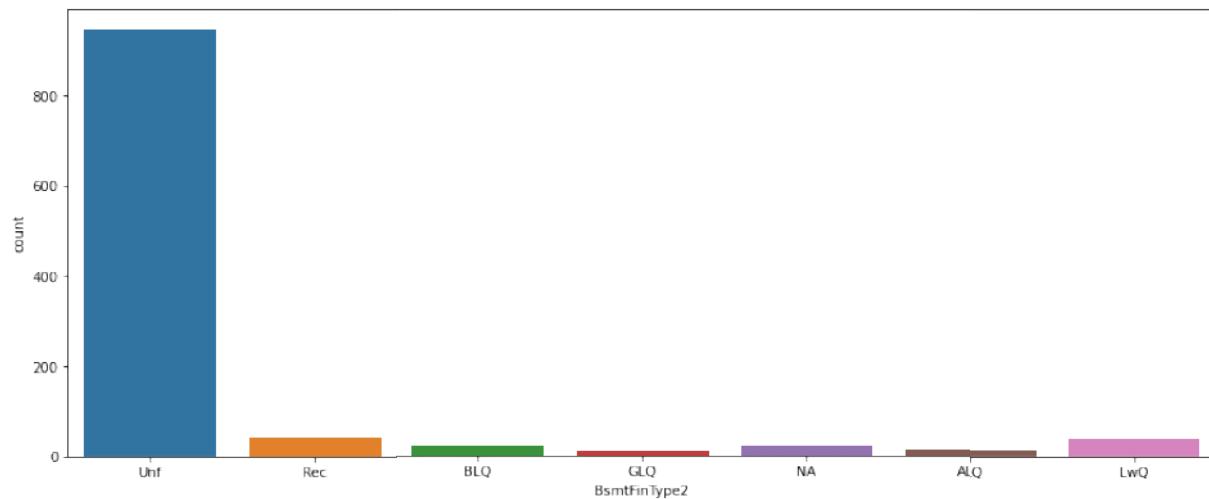


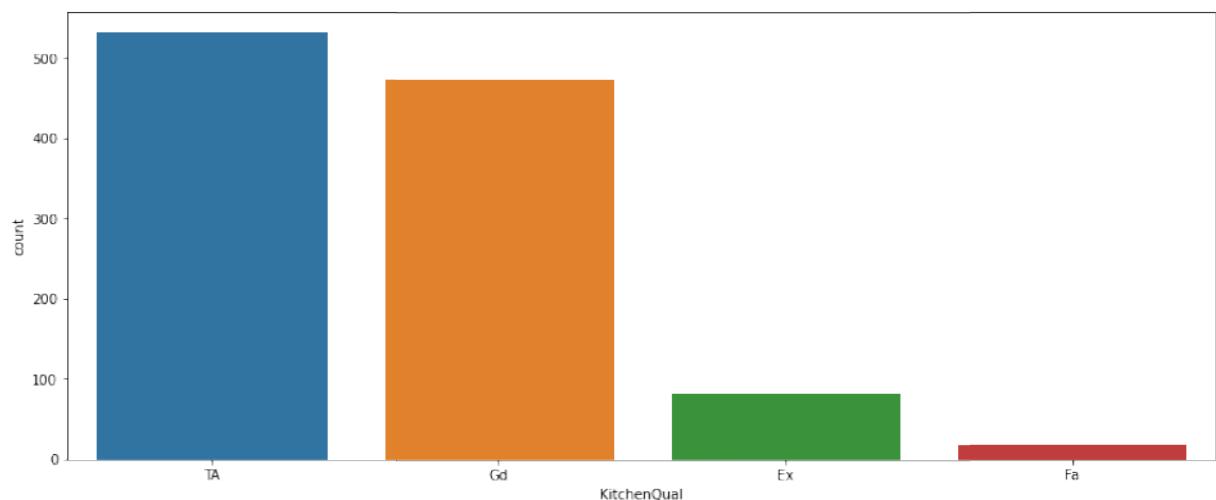
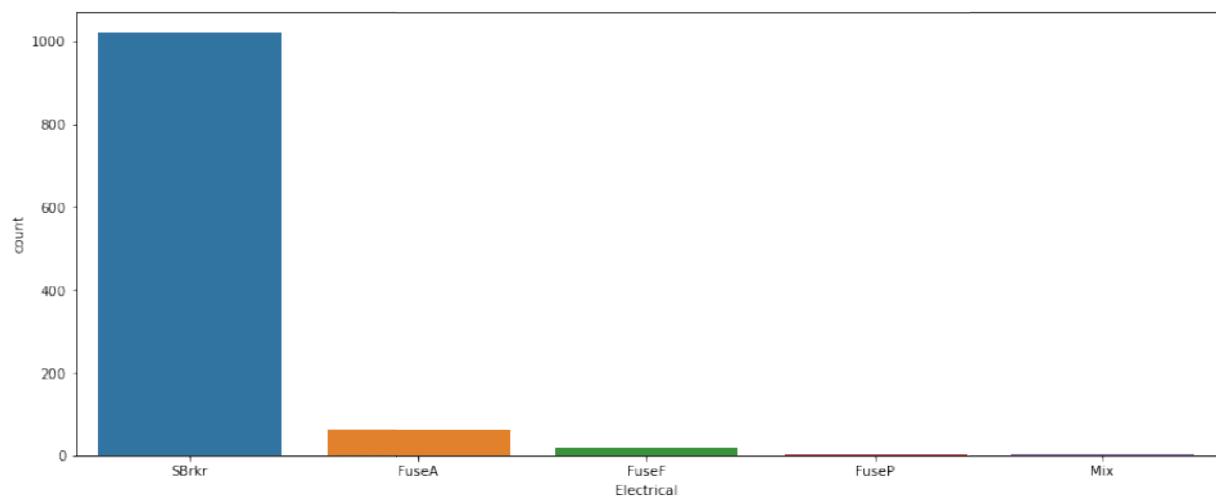
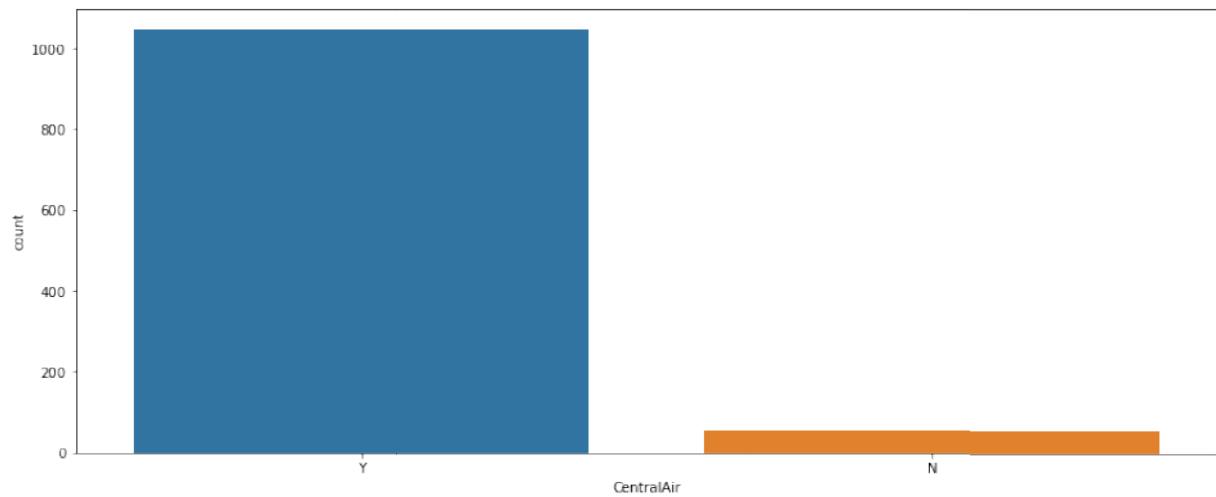


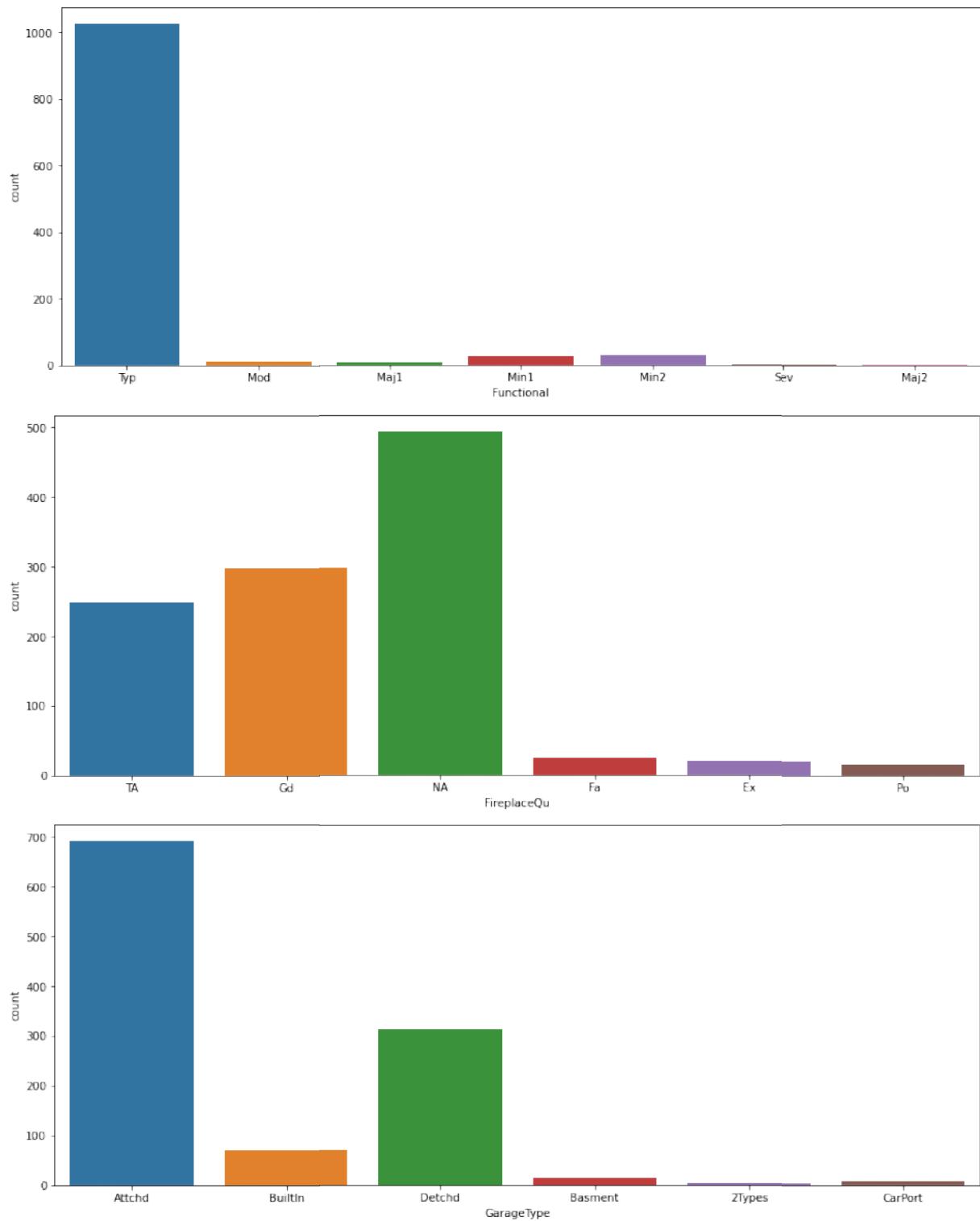


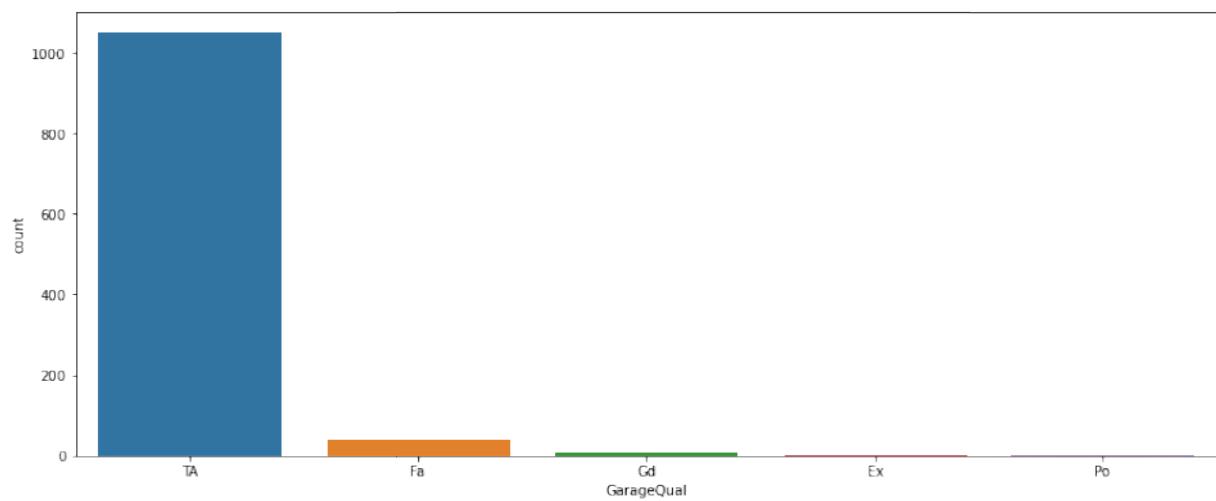
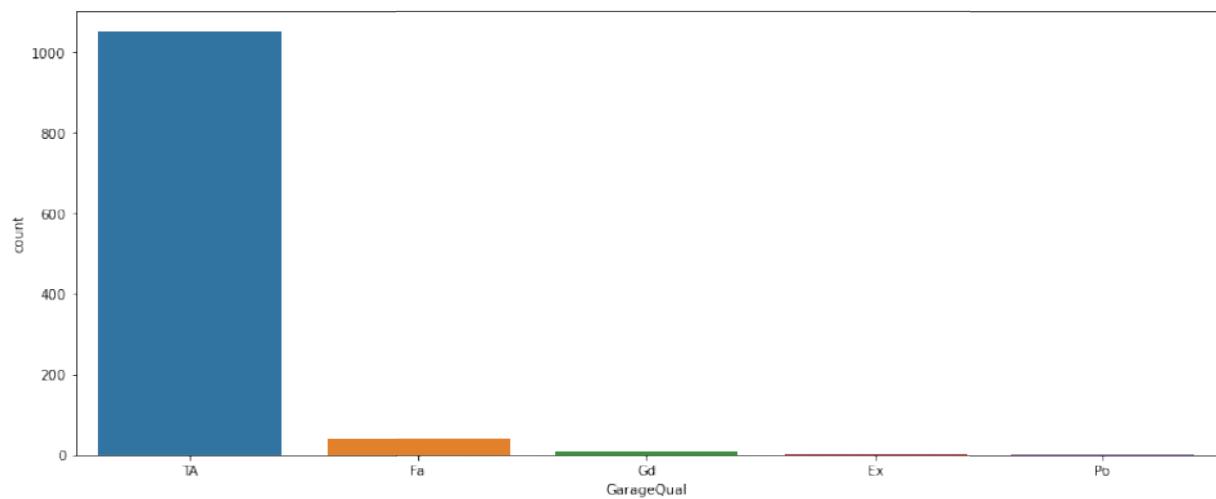
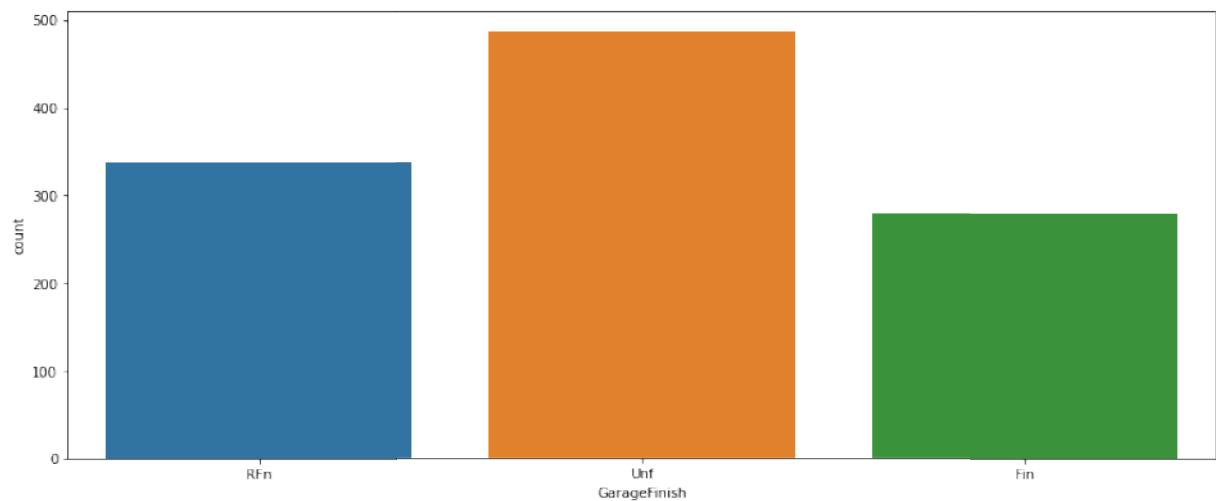


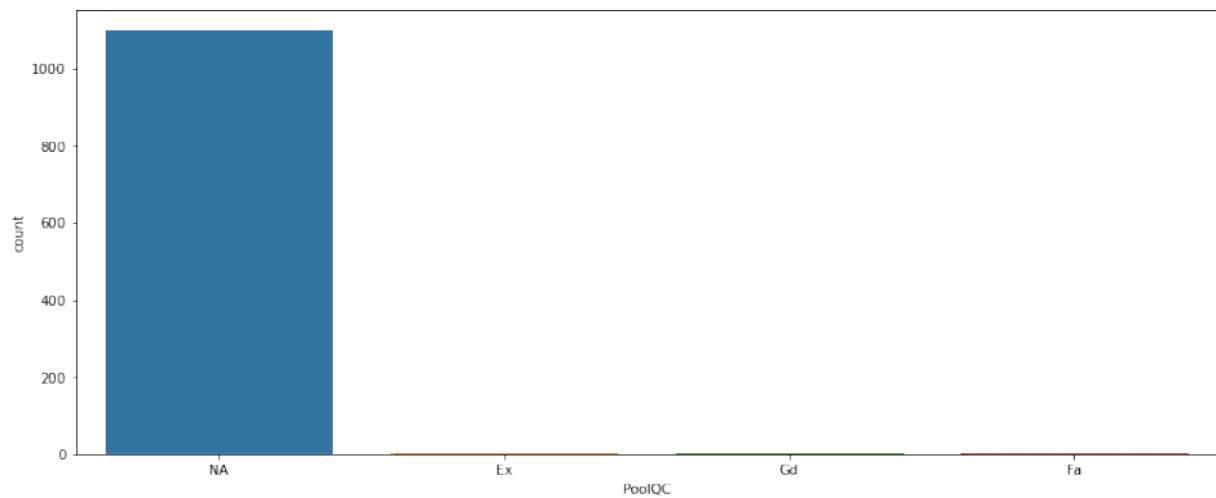
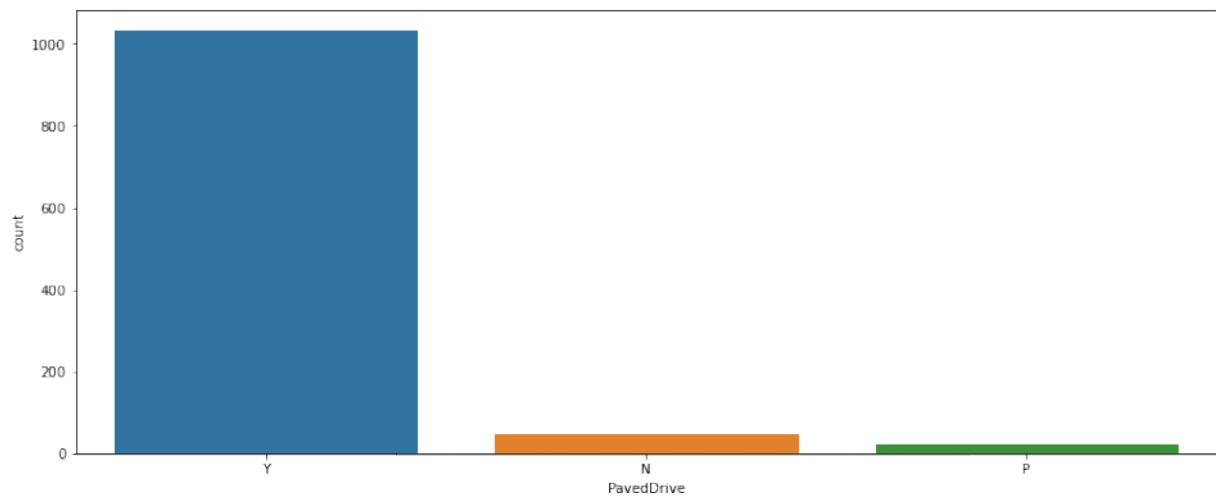
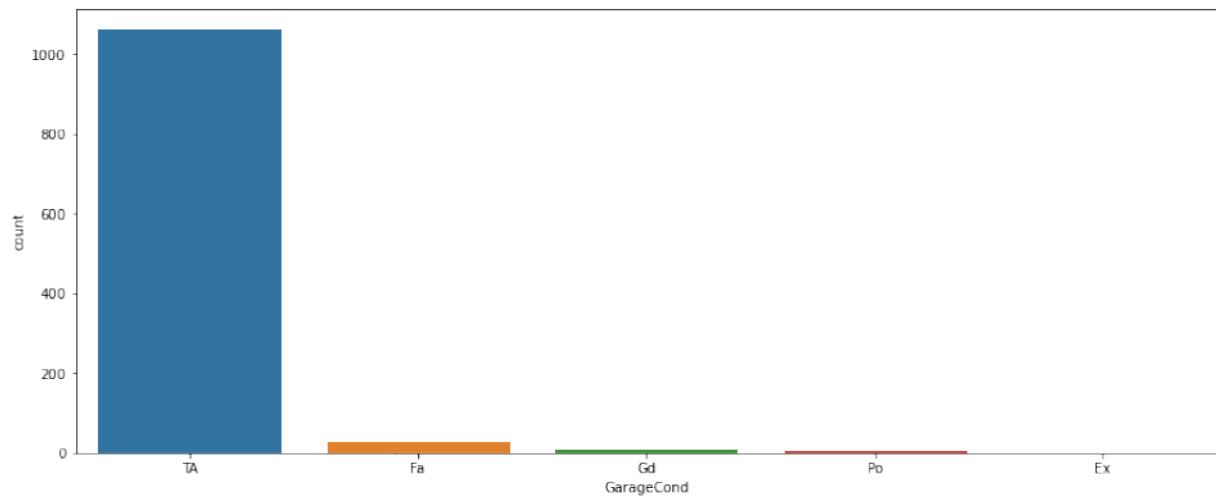


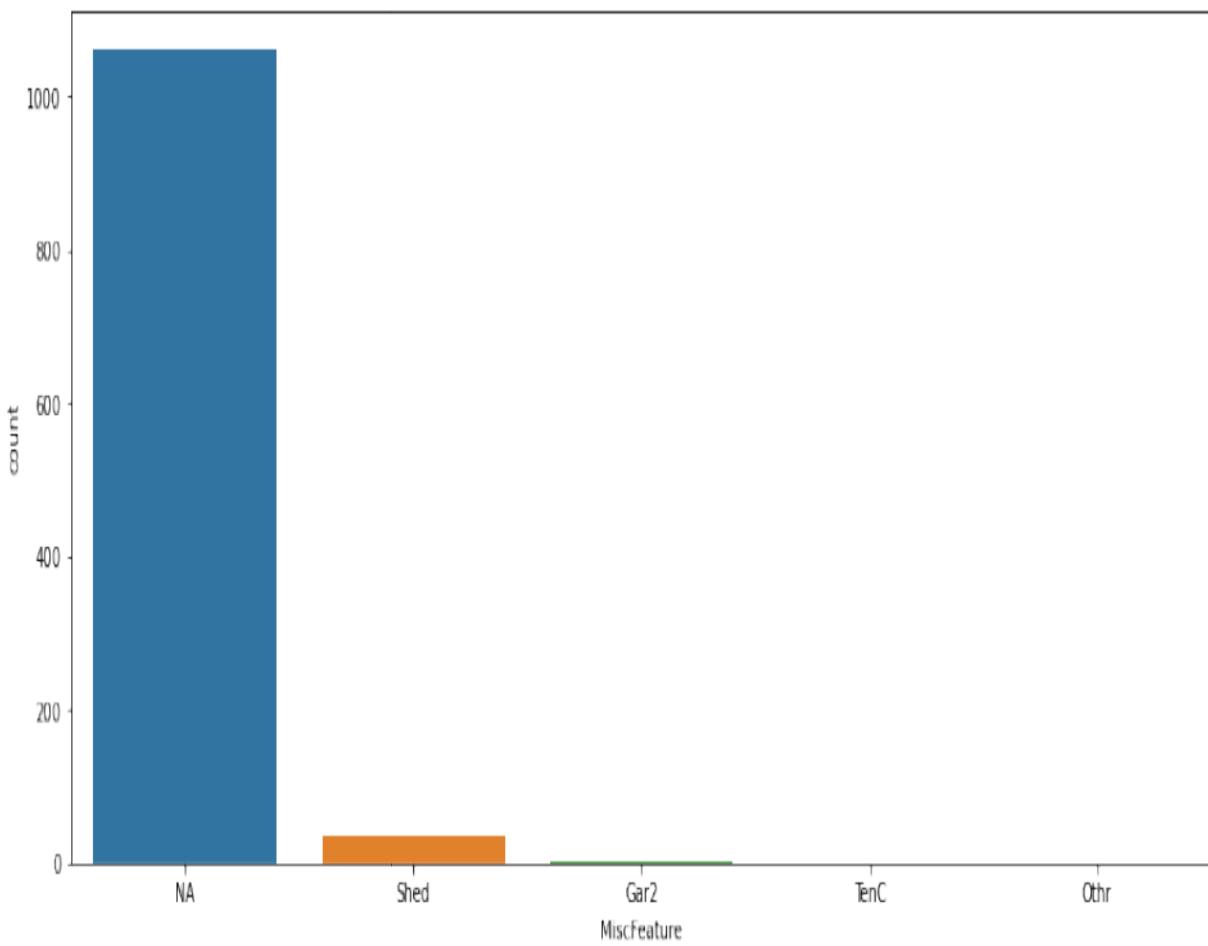
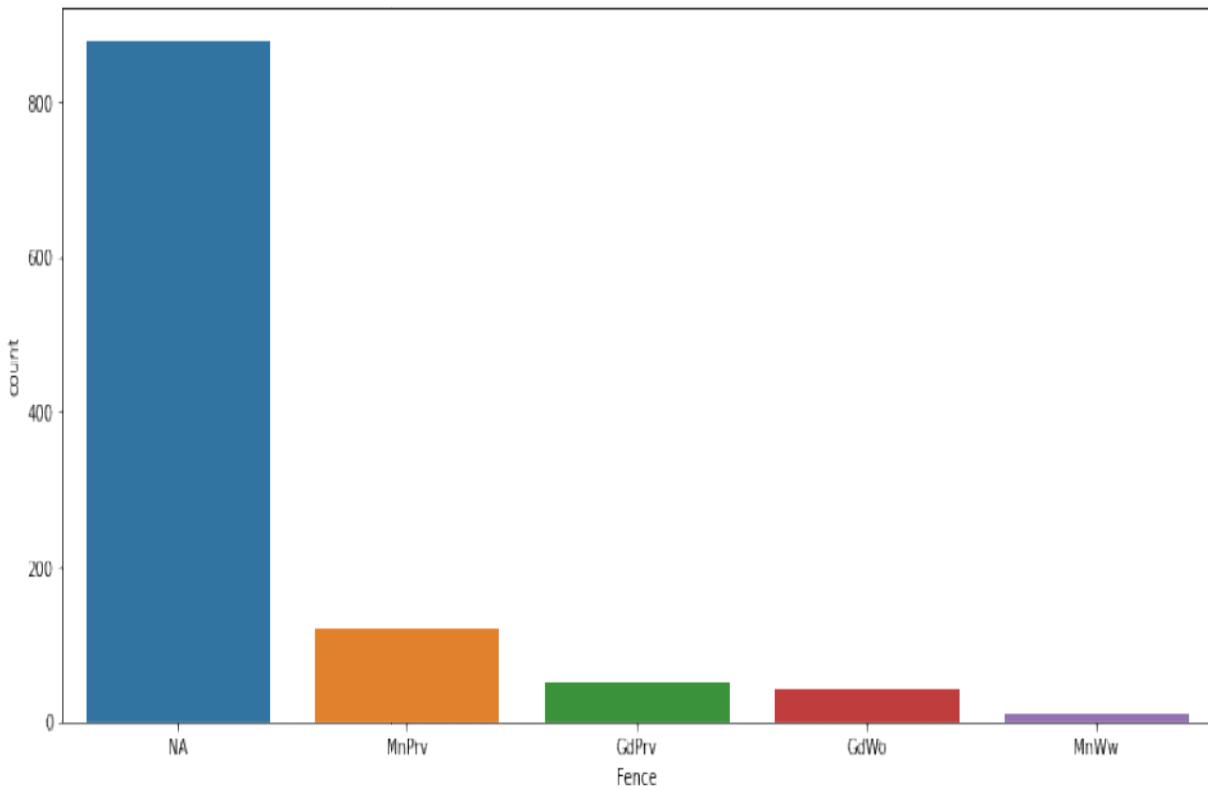


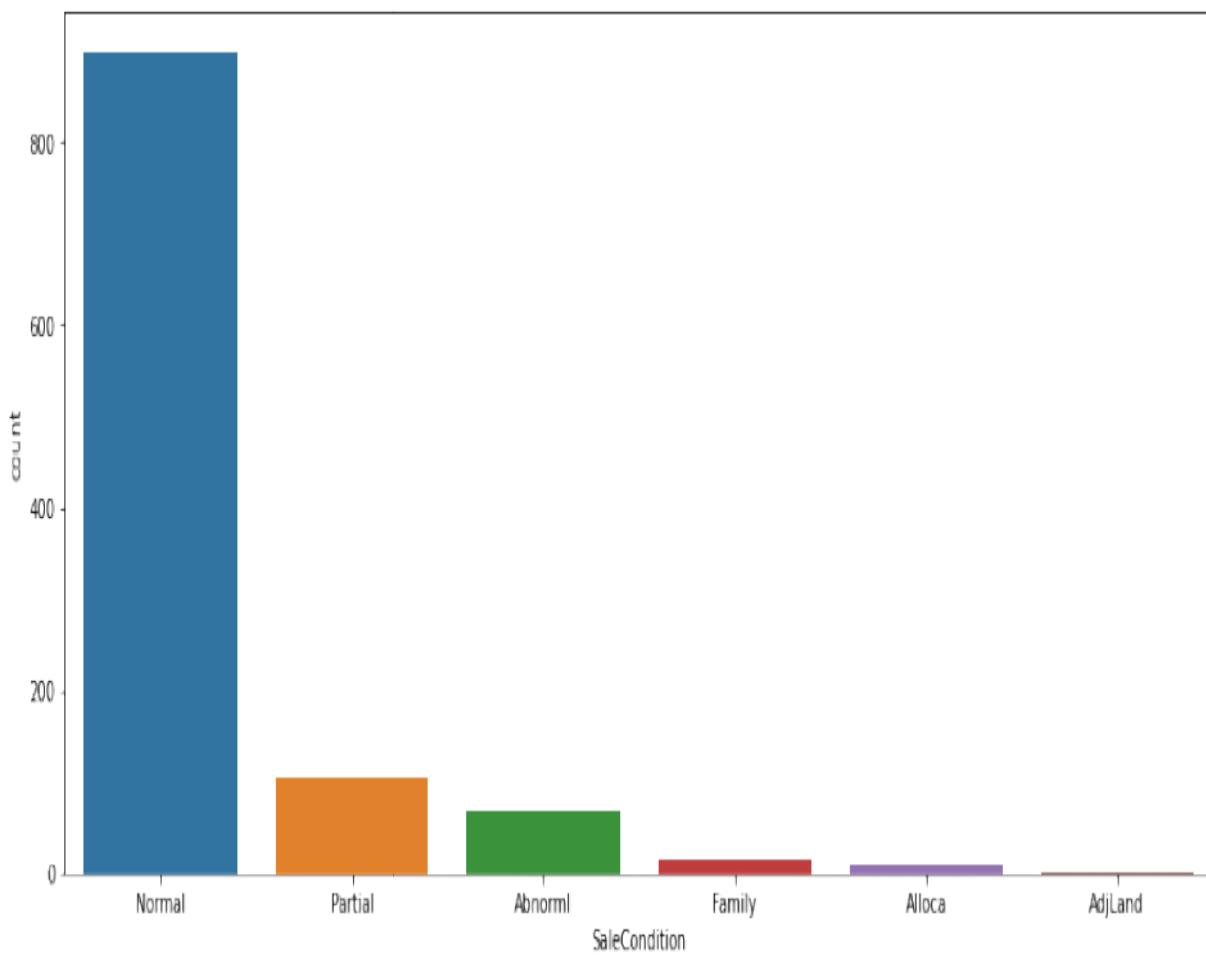
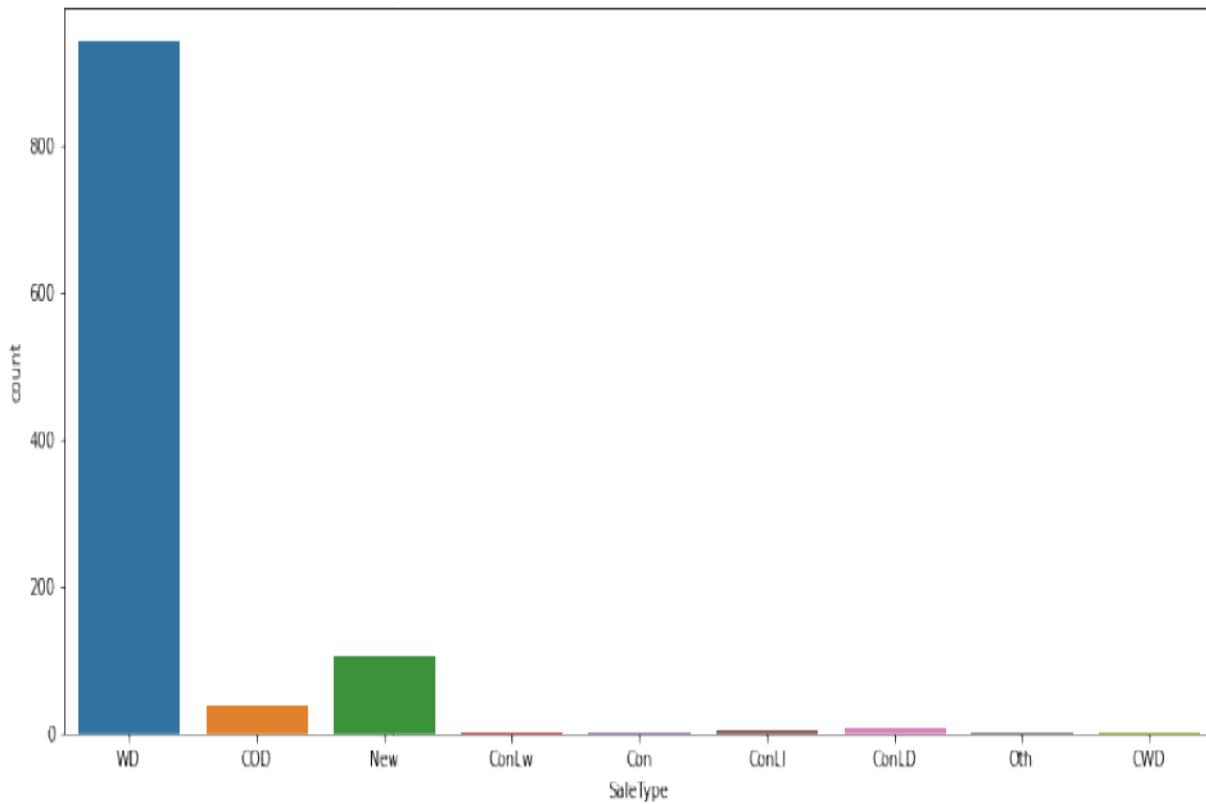












Observation on UNI-VARIATE ANALYSIS :

- 1.On considering the Zone of the house located we could infer that the Residential Low Density zones have higher sales.
- 2.90% of the houses have paved road access.
- 3.Many houses doesn't have an Alley access.
- 4.Most of the plot shapes of the houses are Regular and Slightly irregular. Only a handful of the property have a irregular lot shape.
- 5.Around 90% of the lot has a Flat Landcontour.
- 6.Also most of the Lot configuration are Inside Lot.
- 7.The slope of most property are Gentle.
- 8.The NWAmes and college creek are the most common neighborhood.
- 9.Most houses have the proximity to various conditions as normal.
- 10.Single family detached is the most common form of house found in the data-set.
- 11.One story, Two story are known to be the common house style followed by One and one-half story: 2nd level finished.
- 12.Gable is found to be the roof style for most of the houses followed by Hip.
- 13.Standard (Composite) Shingle is the most common material used for roofing.
- 14.Exterior condition of the most houses are Average / Typical.
- 15.Cinder Block and Poured Contrete are the common type of foundation used followed by Brick & Tile.
- 16.The height of the basement for most houses are found to be Good and Typical.
- 17.Also most basements seems to have slight dampness.
- 18.Most of the basement have no exposure.
- 19.More than 95% of the houses have Gas forced warm air furnace Type of heating.
- 20.Most heating equipments in the houses are found to be excellent to average.
- 21.90% of houses have Central air conditioning.

22.Standard Circuit Breakers & Romex is found to be the most common Electrical system.

23.Most houses have excellent kitchen condition.

24.Most houses have Typical Home functionality.

25.Most of the houses dont have a fire place.

26.Many houses have a attached garage.

27.Also many garages seems to be unfinished.

28.Garage quality of most houses seems to be typical.

29.Paved driveway is the most commn drive way.

30.Almost 95% of the houses dont have a Pool.

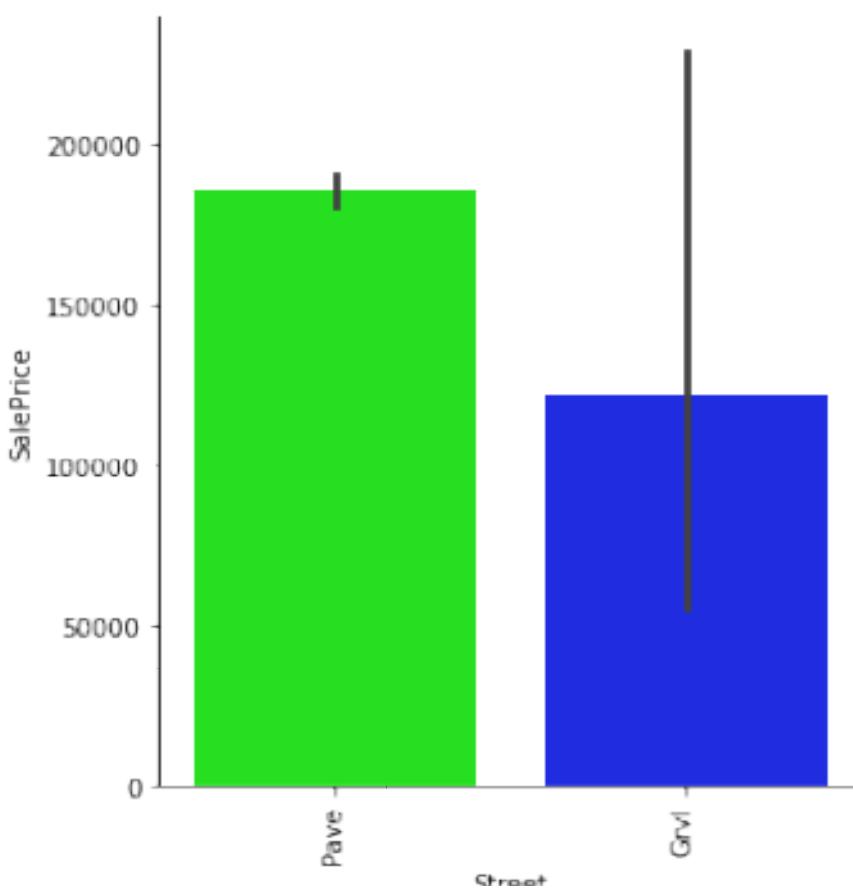
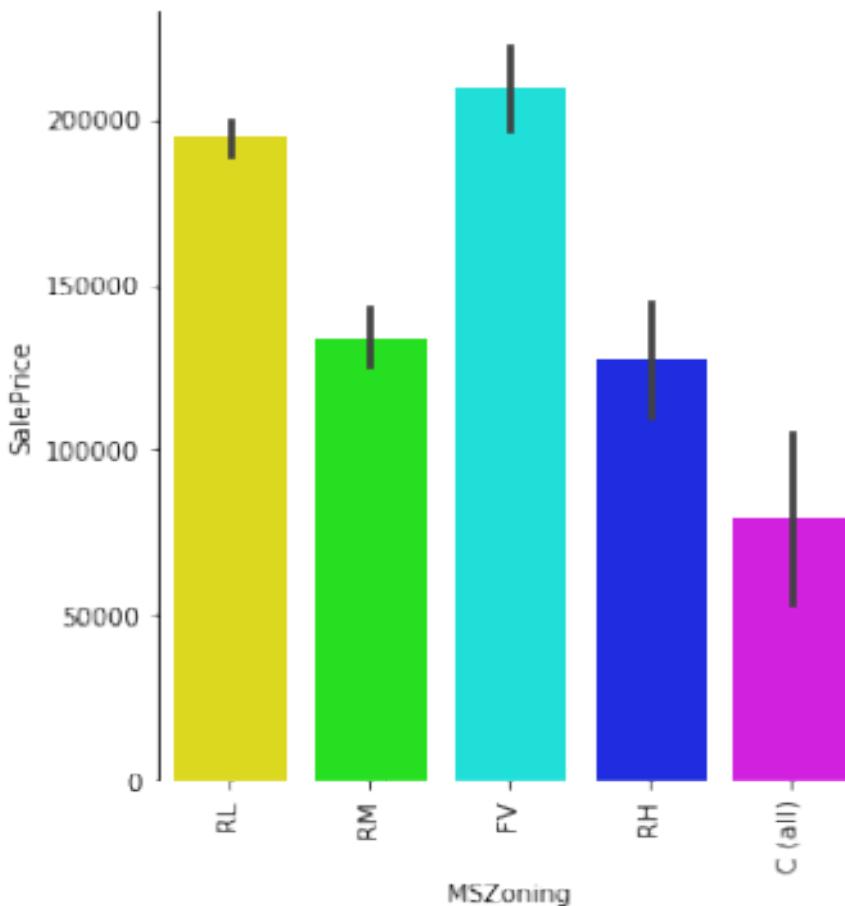
31.Many houses dont have a Fence.

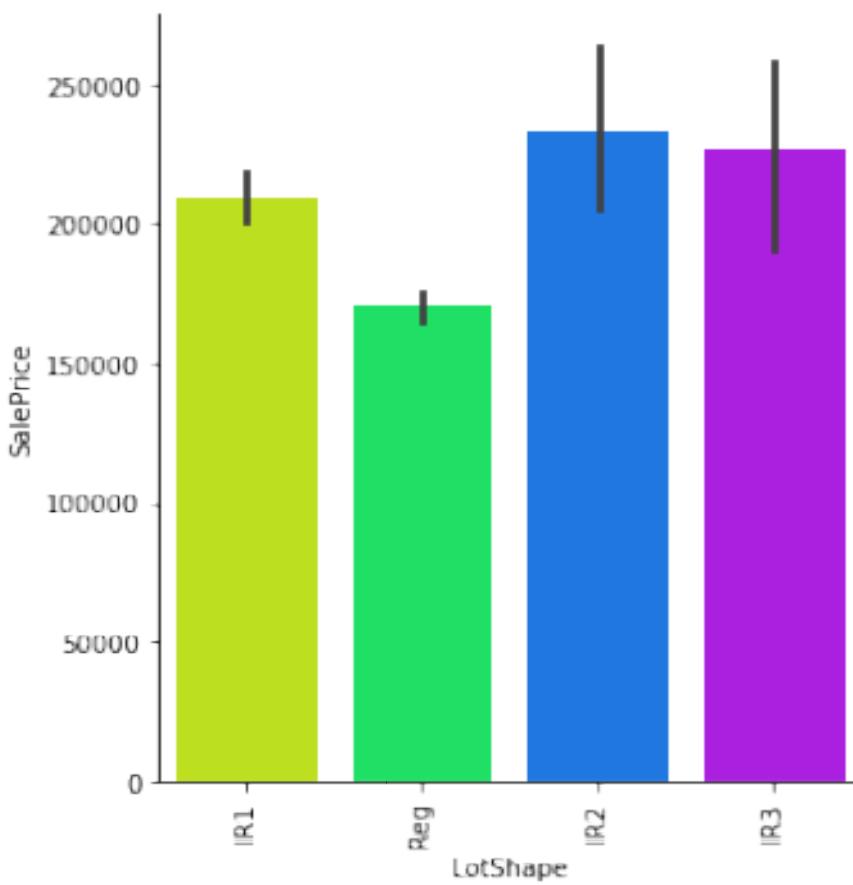
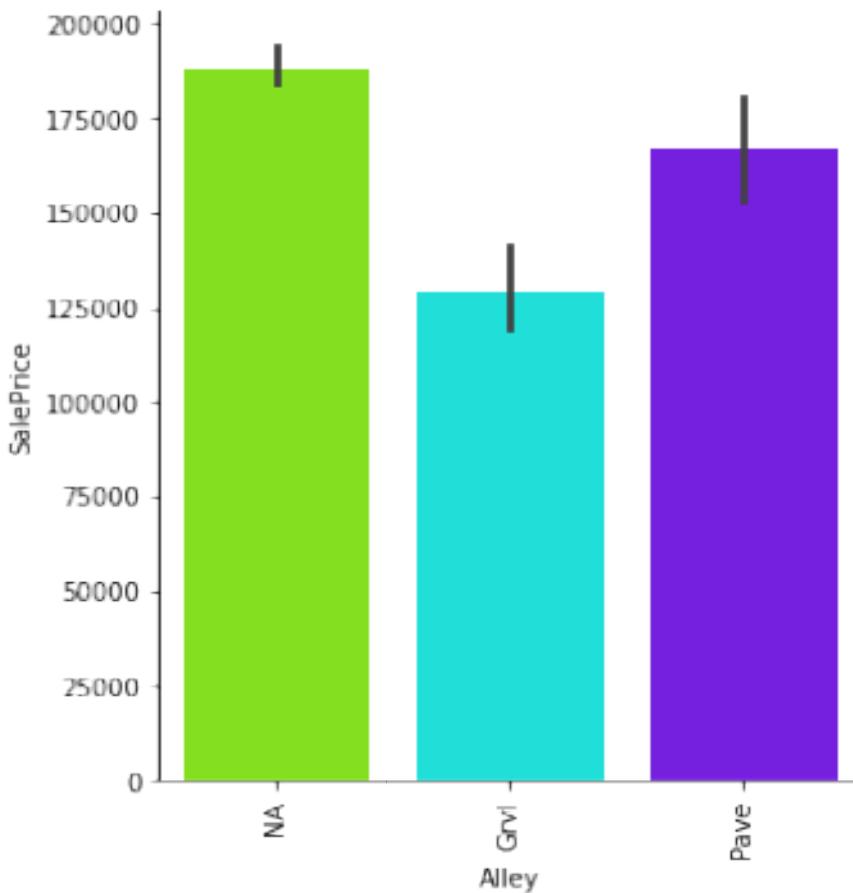
32.Many dont have any of the following features Elevator,2nd Garage,Other,Shed,Tennis Court.

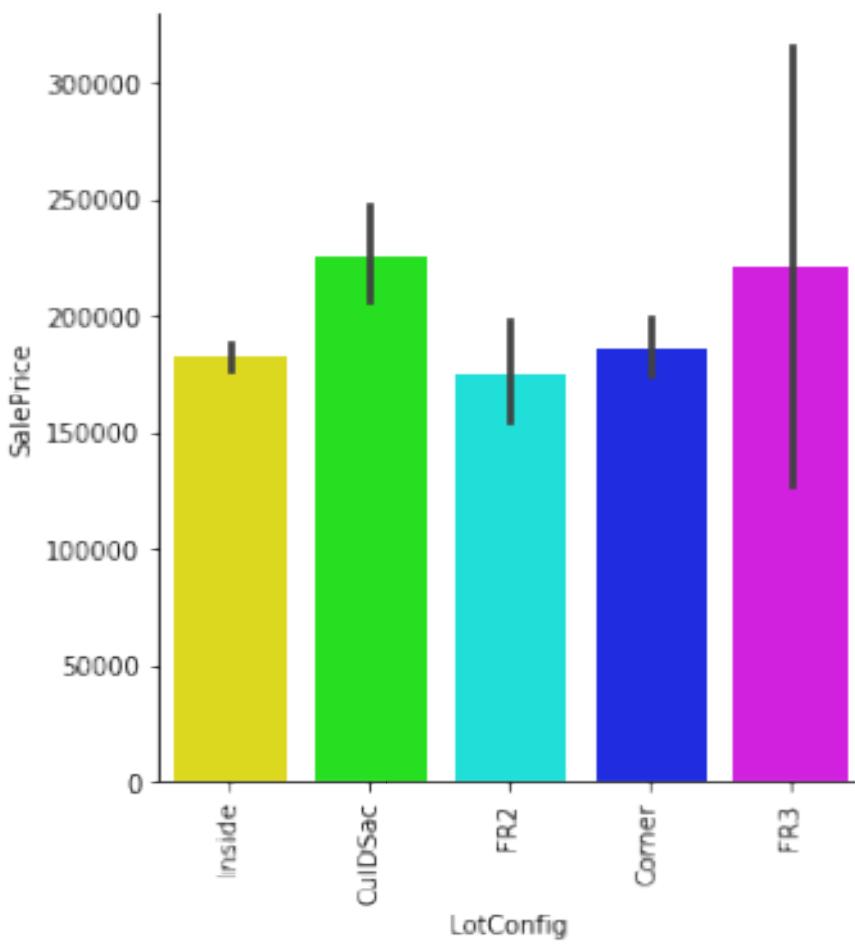
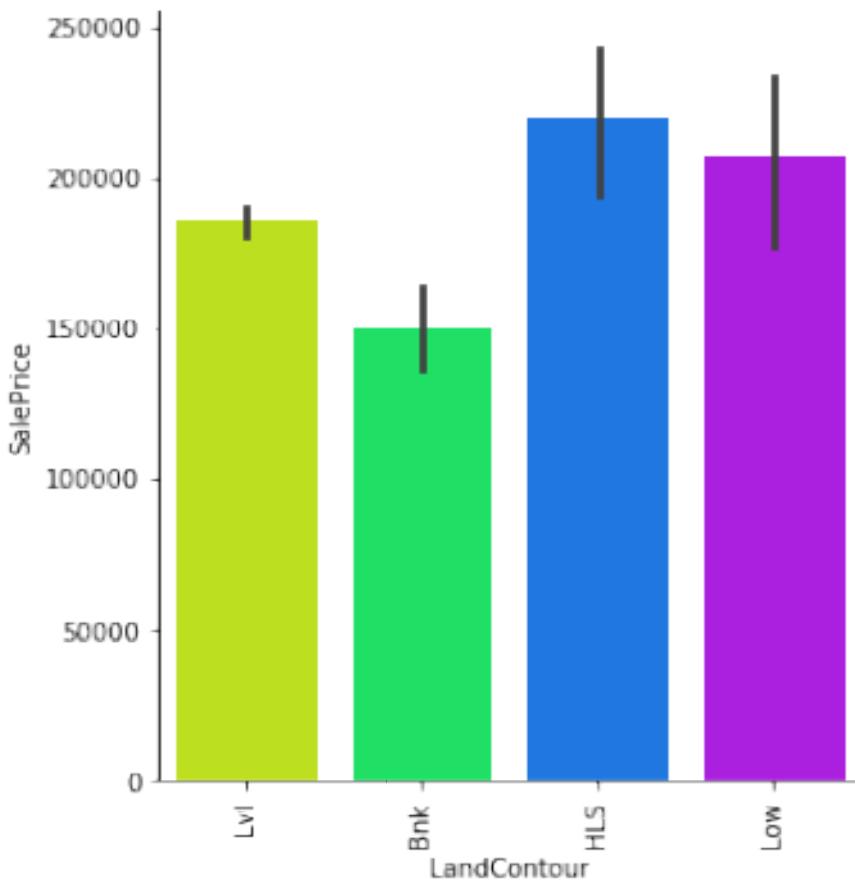
33.Most of the houses are sold by Warranty Deed - Conventional, followed by new homes and Court Officer Deed.

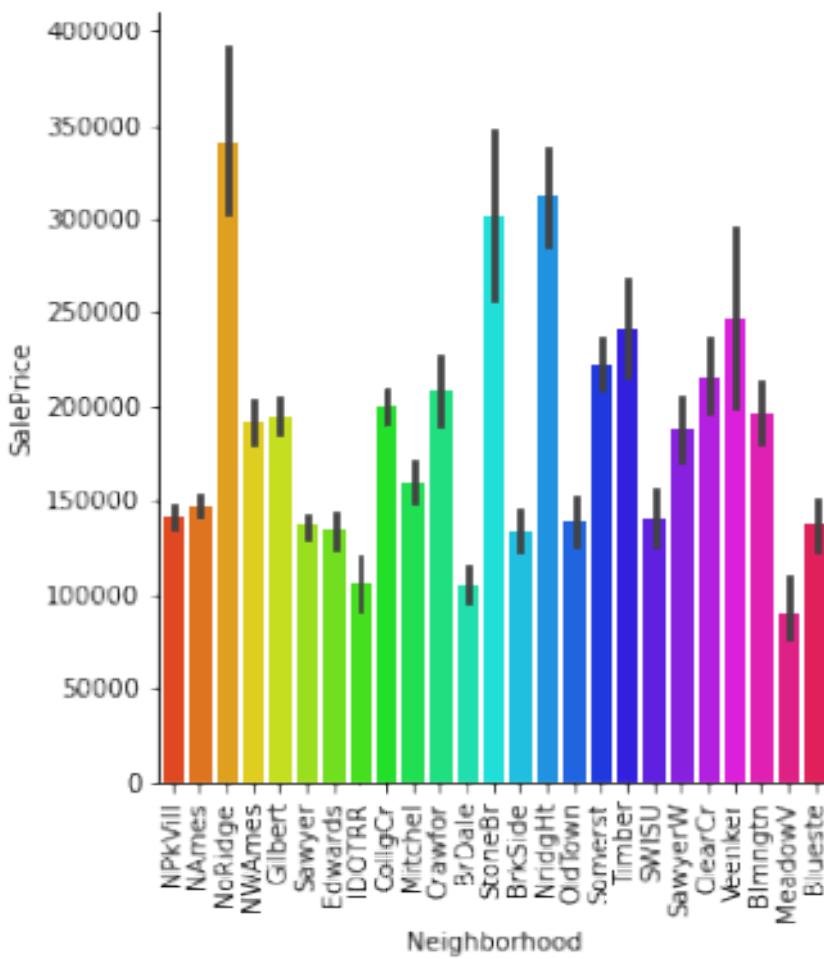
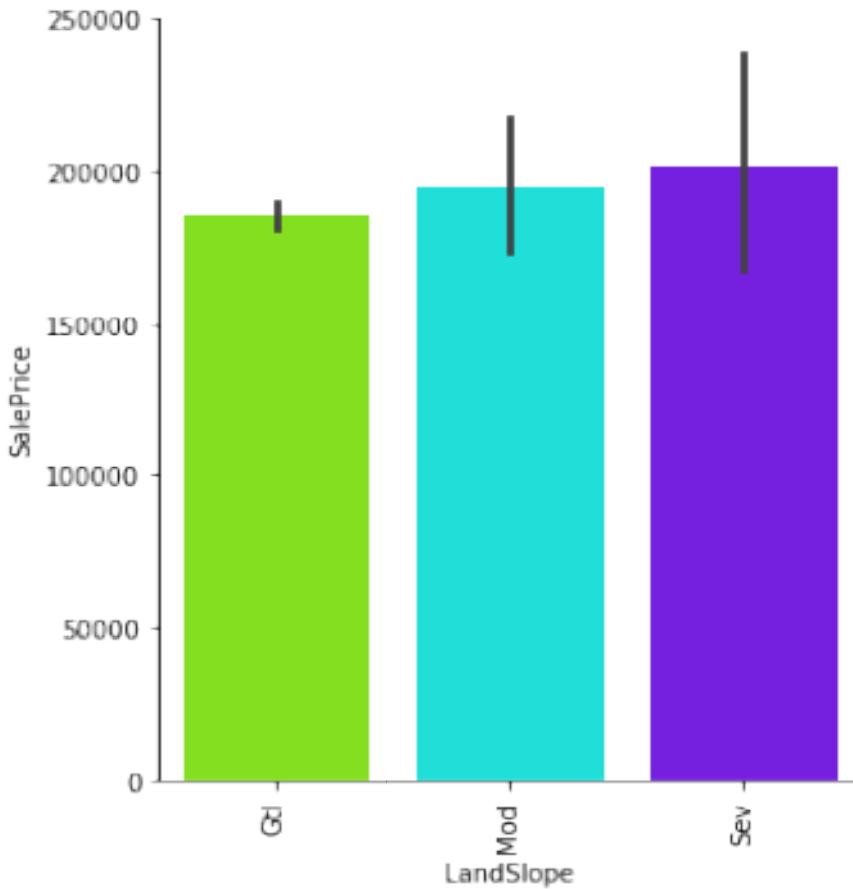
34.Most of the houses are sold under Normal condition. but some of them are sold in partial construction and abnormal conditions such as trade, foreclosure, short sale.

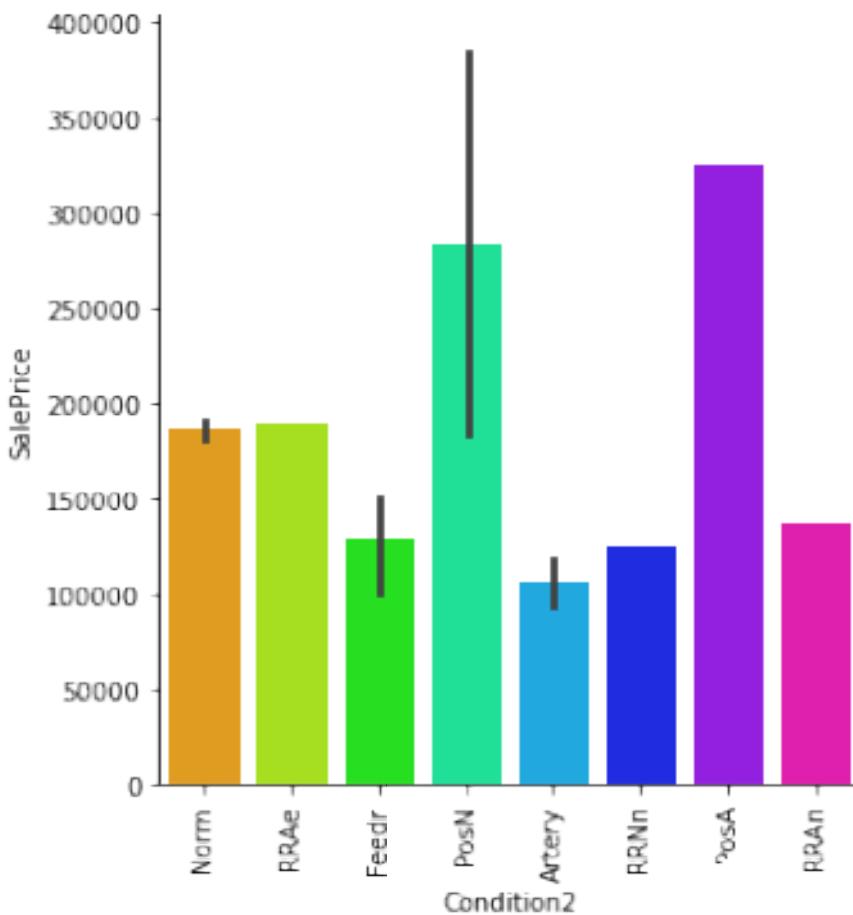
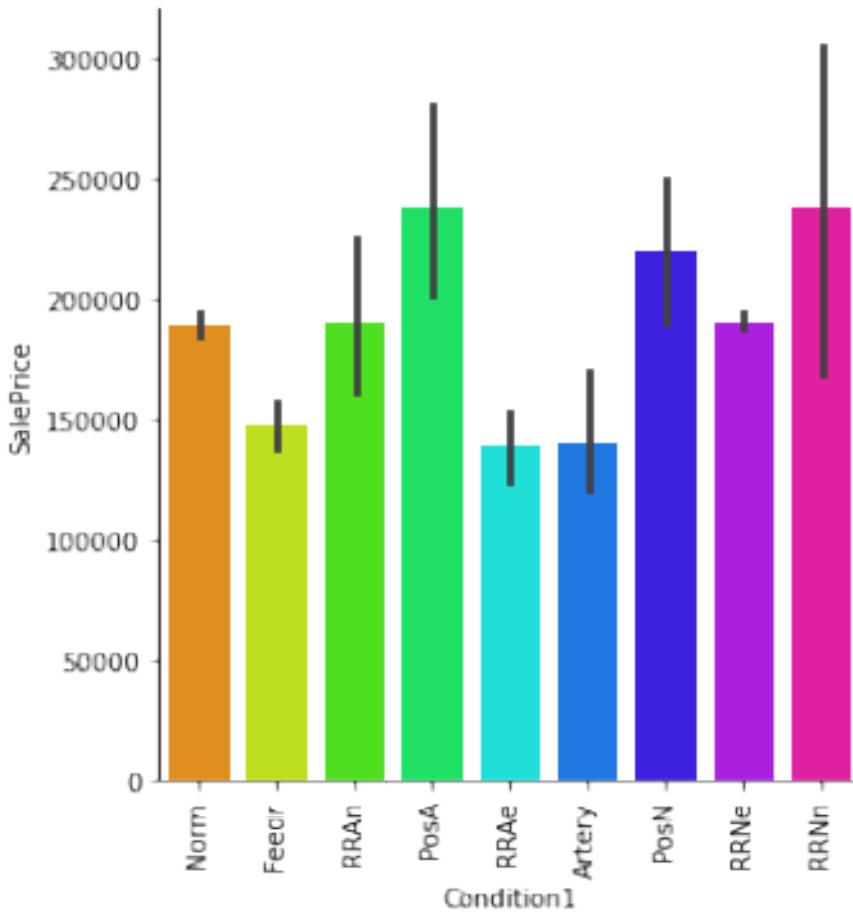
BI-VARIATE ANALYSIS :

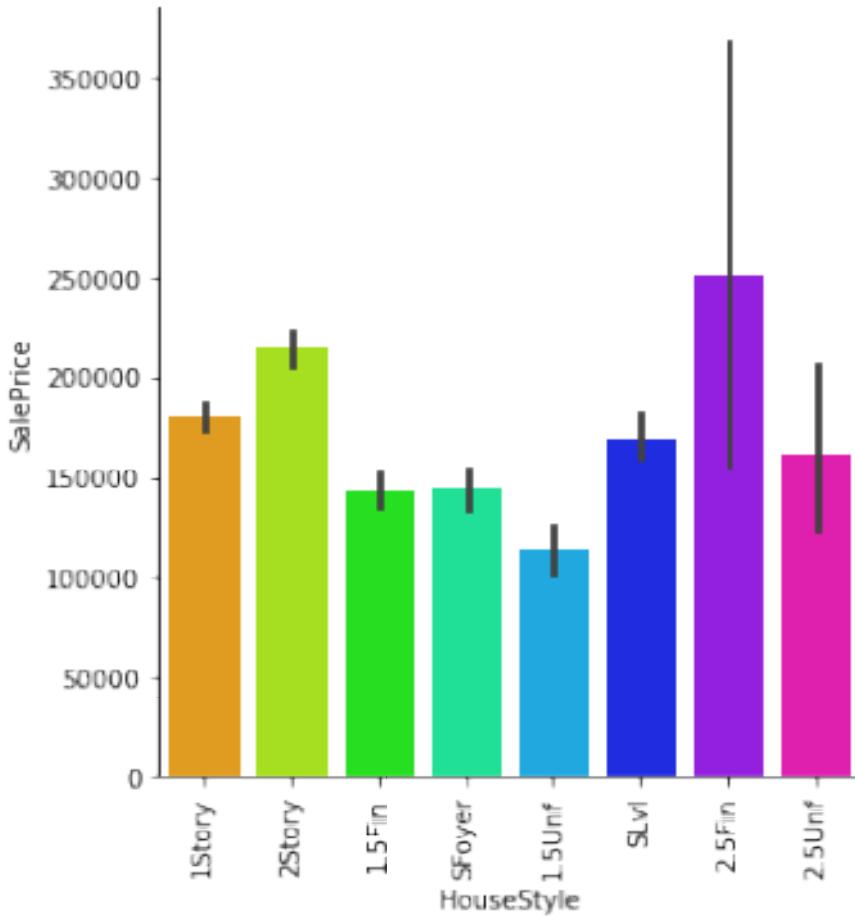
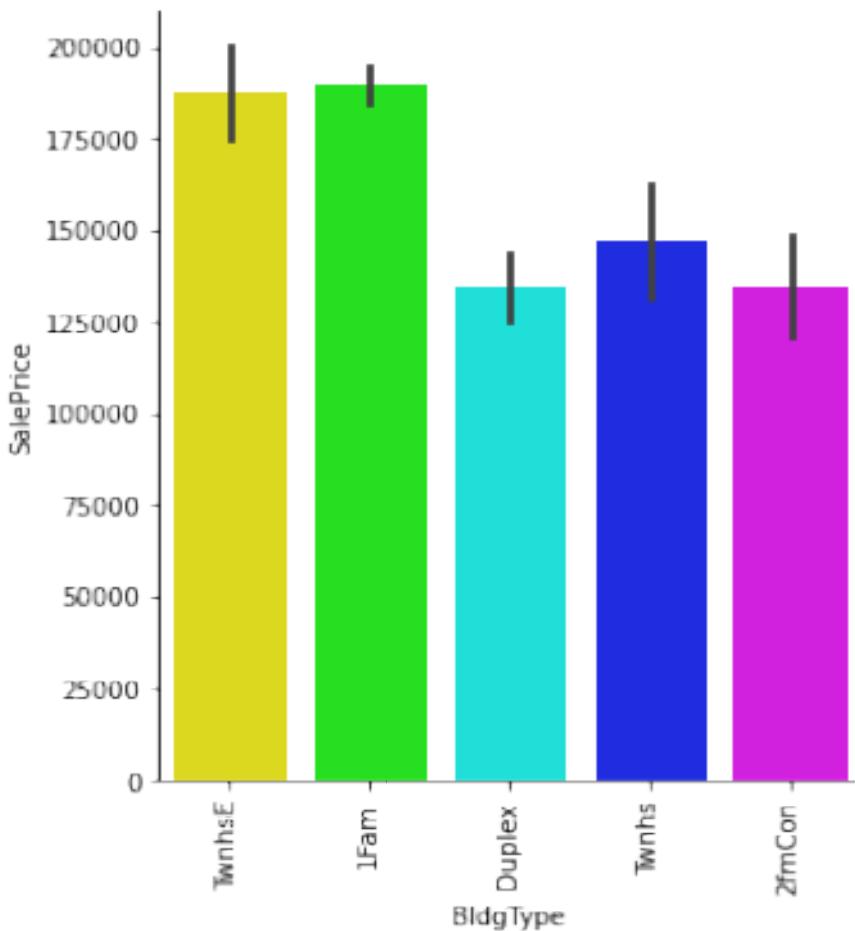


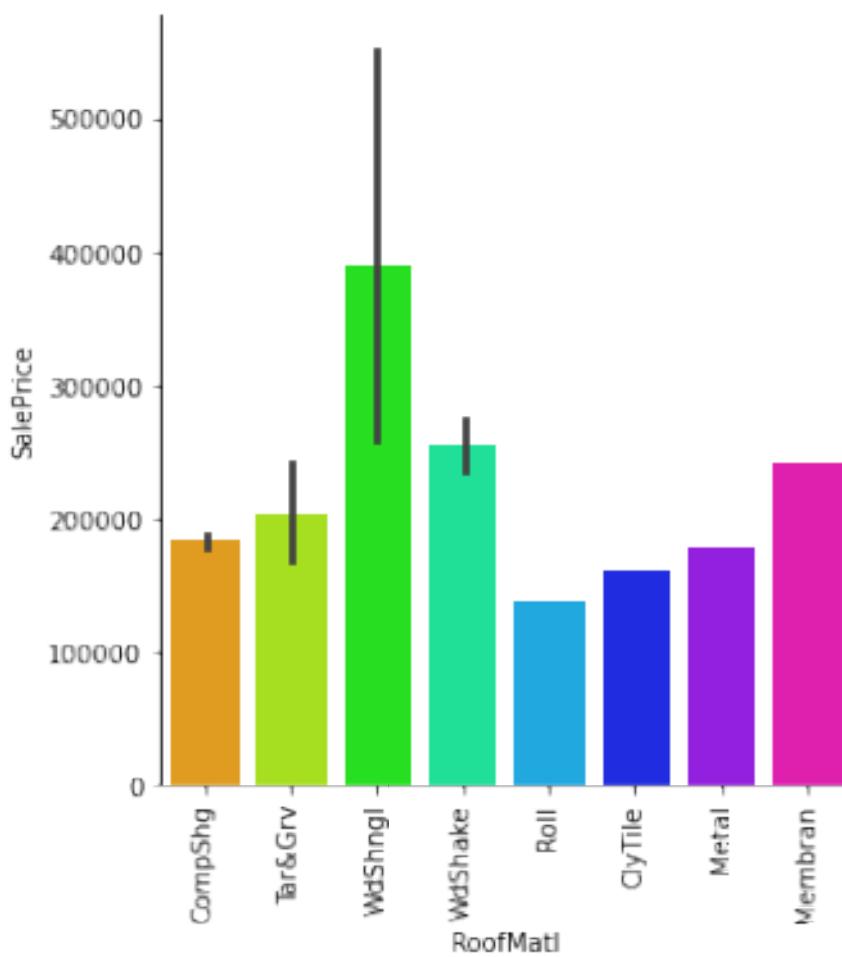
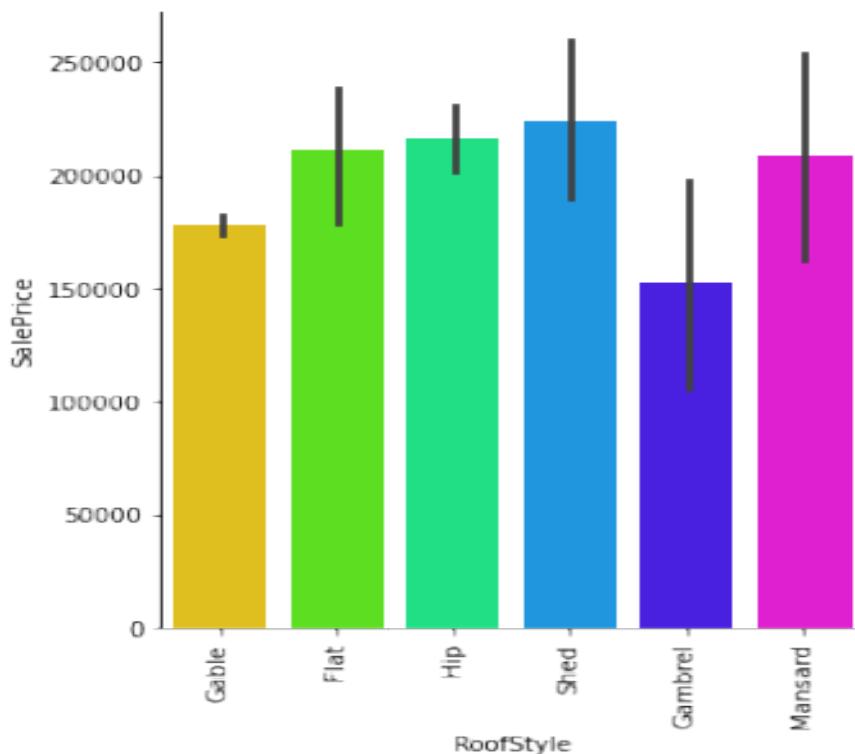


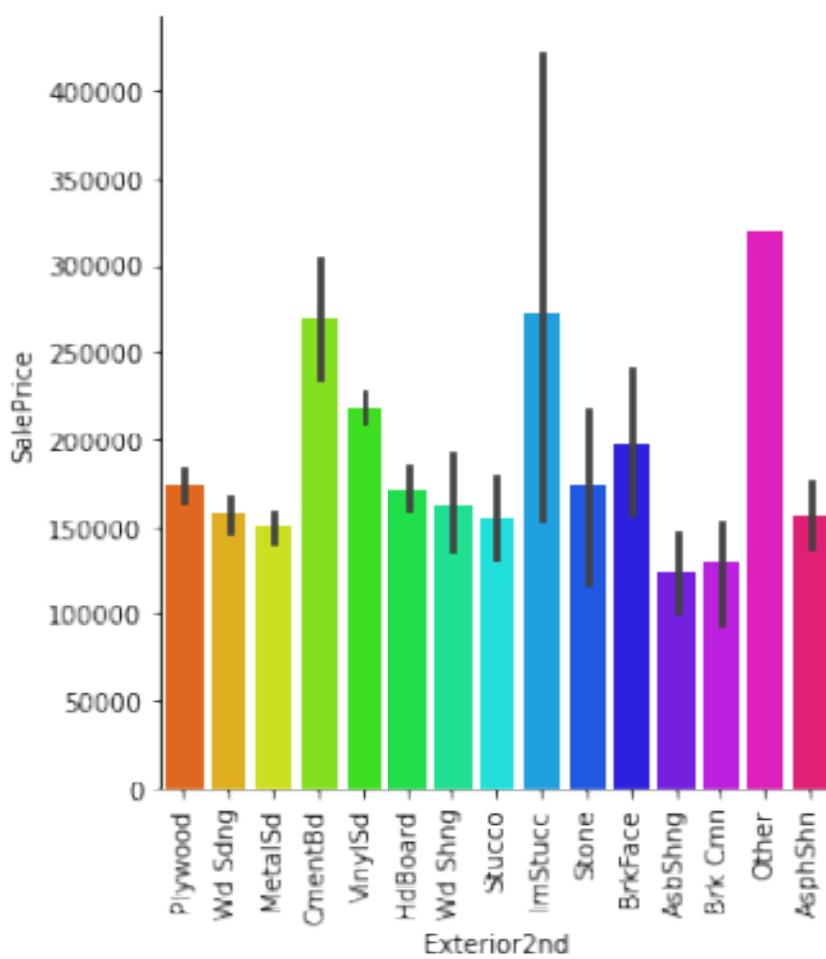
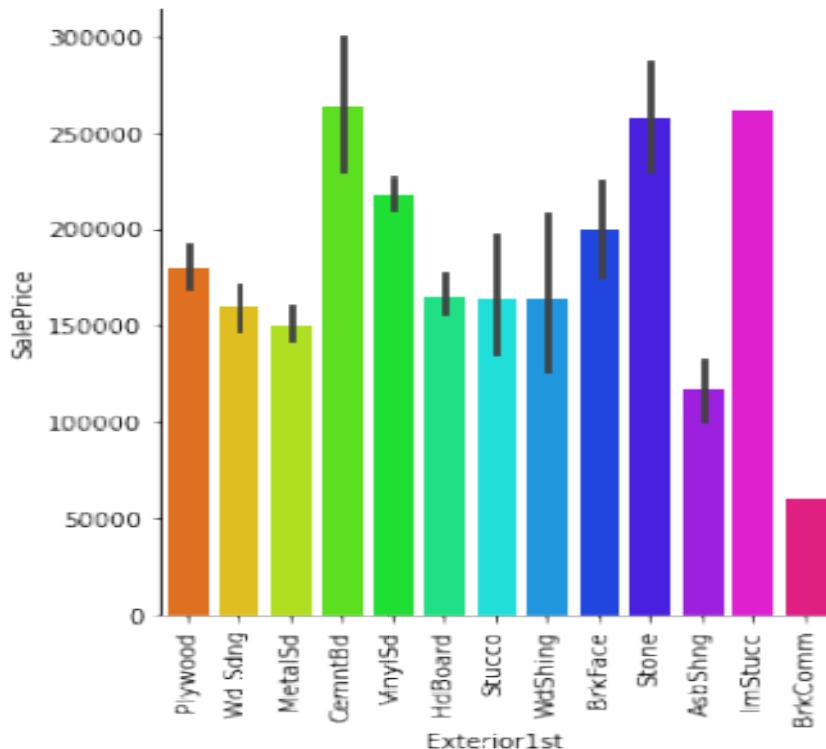


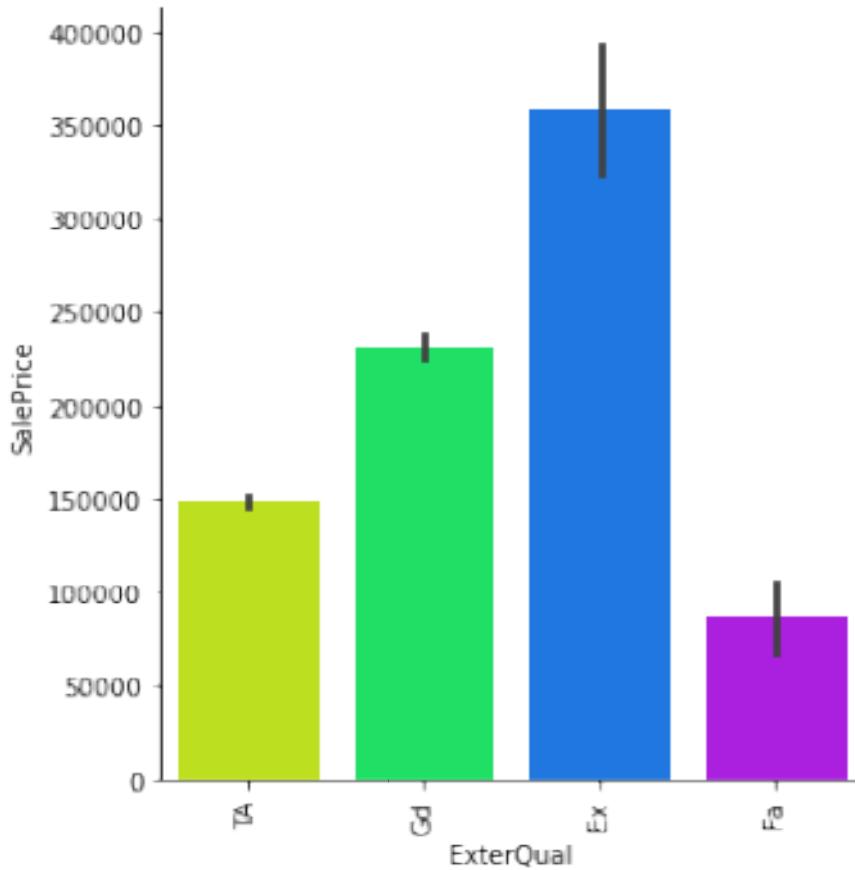
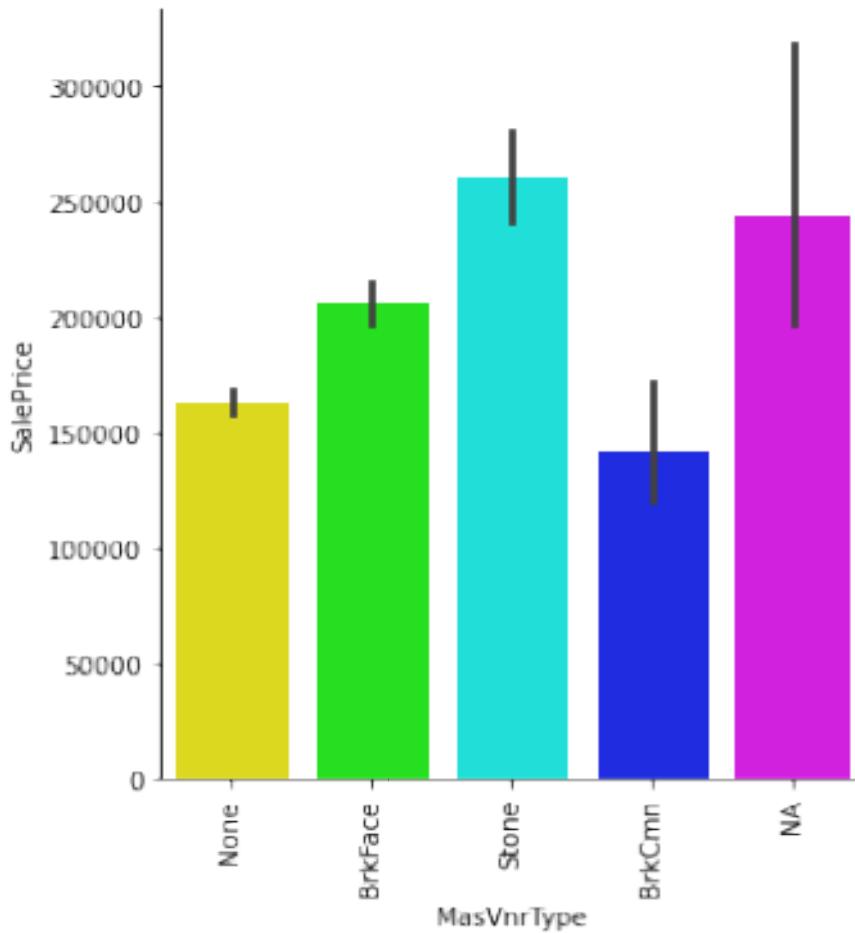


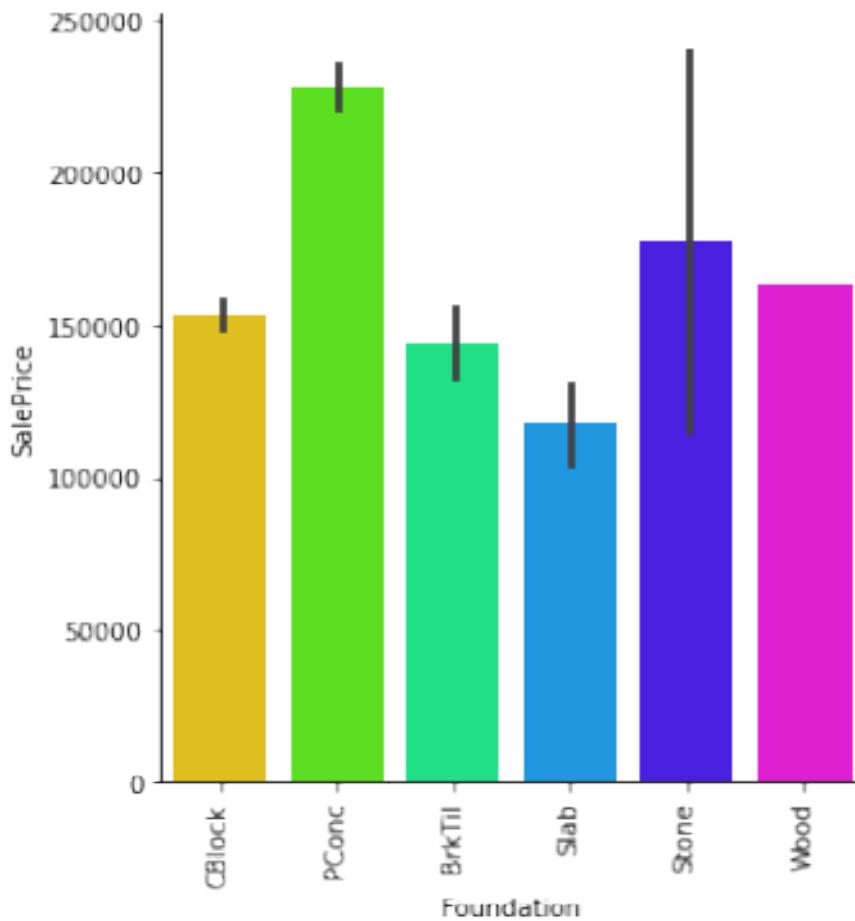
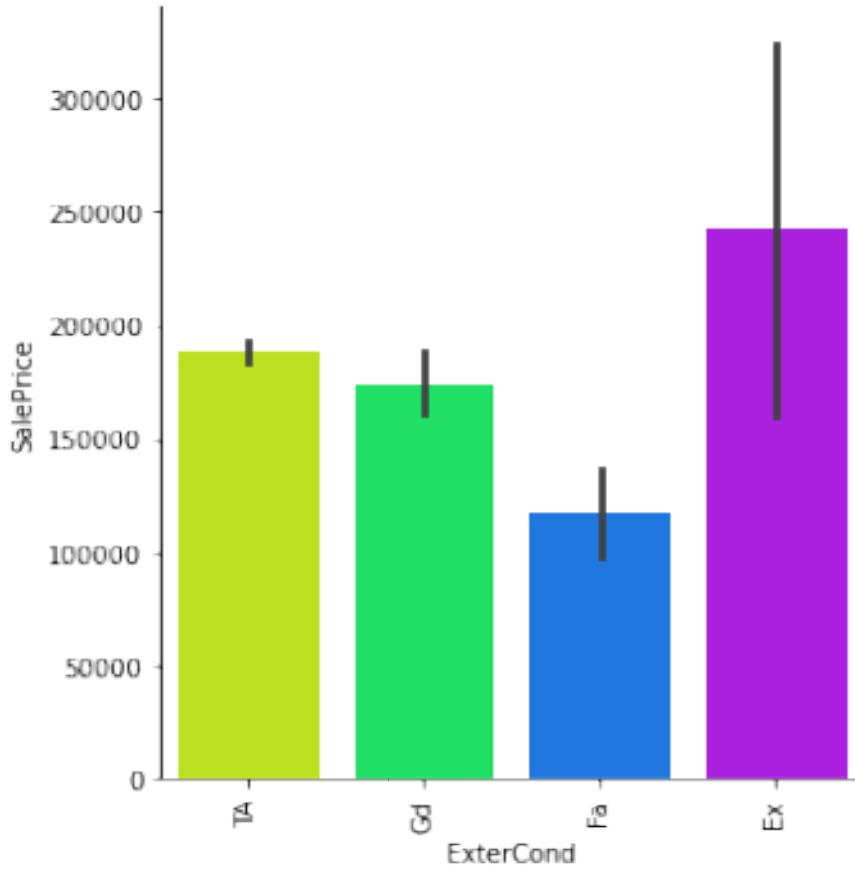


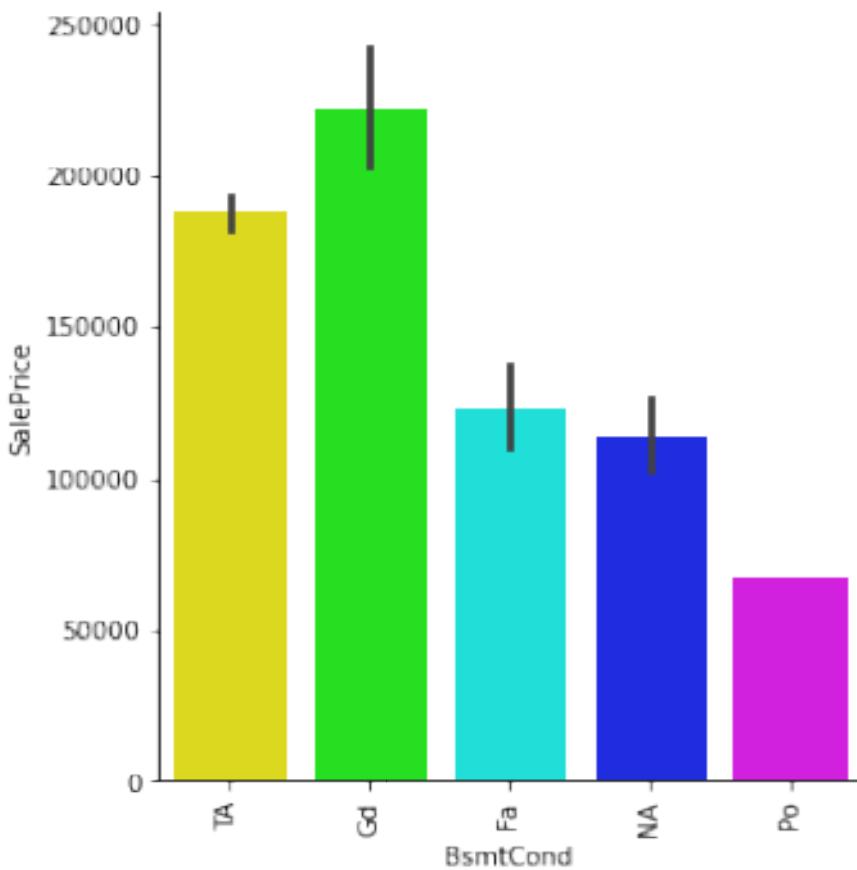
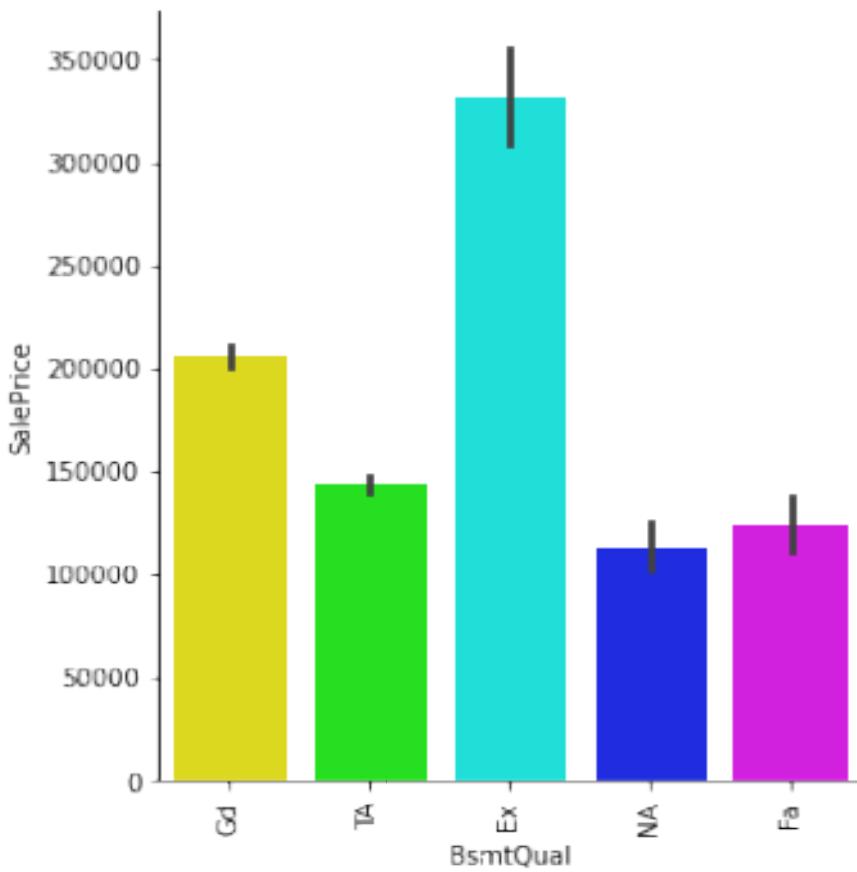


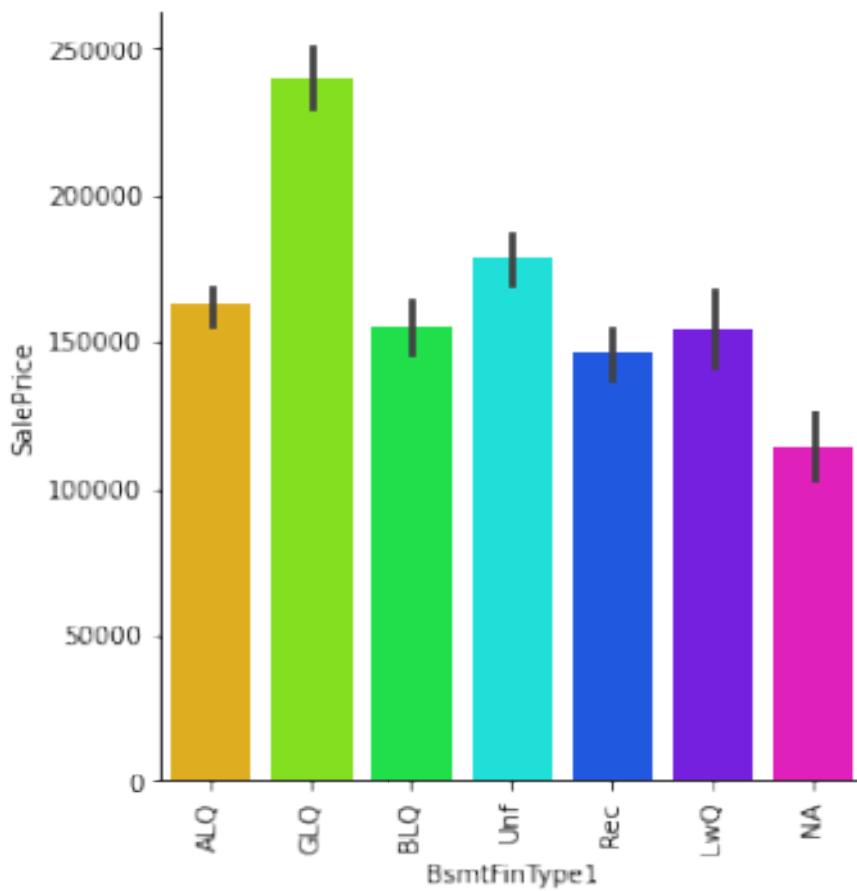
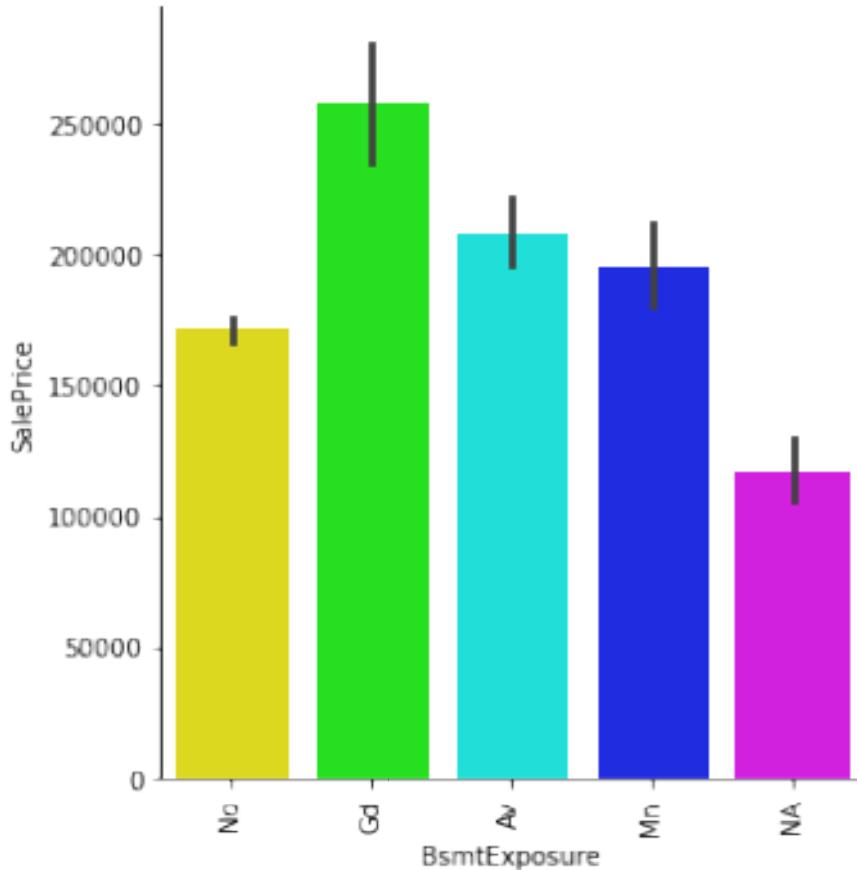


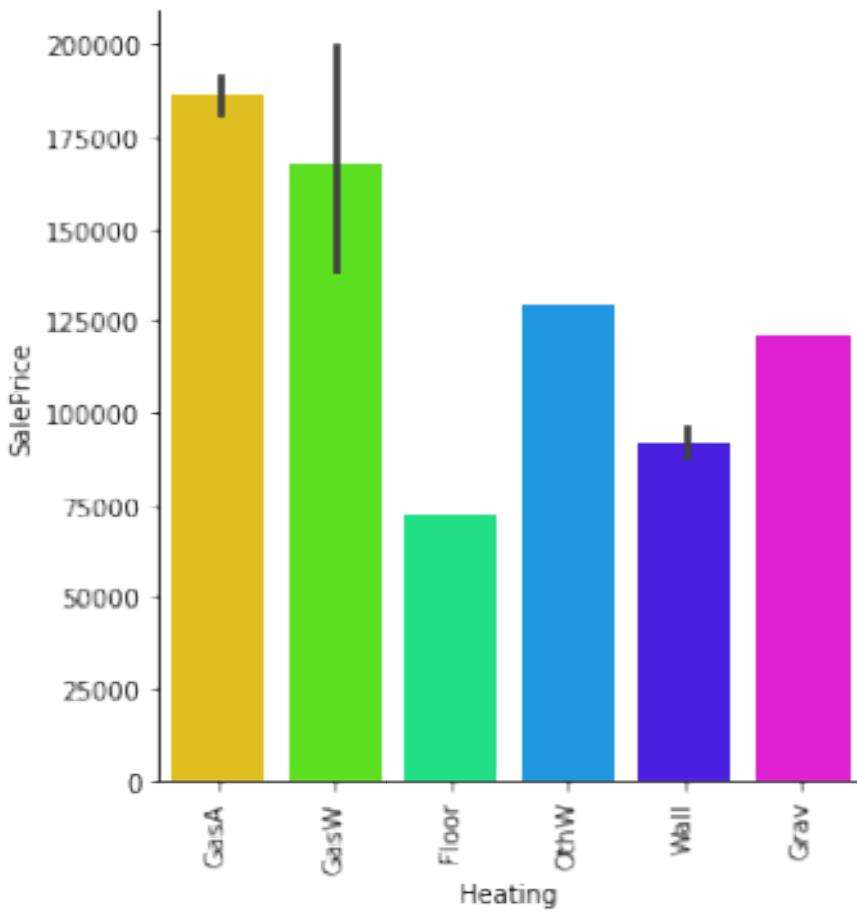
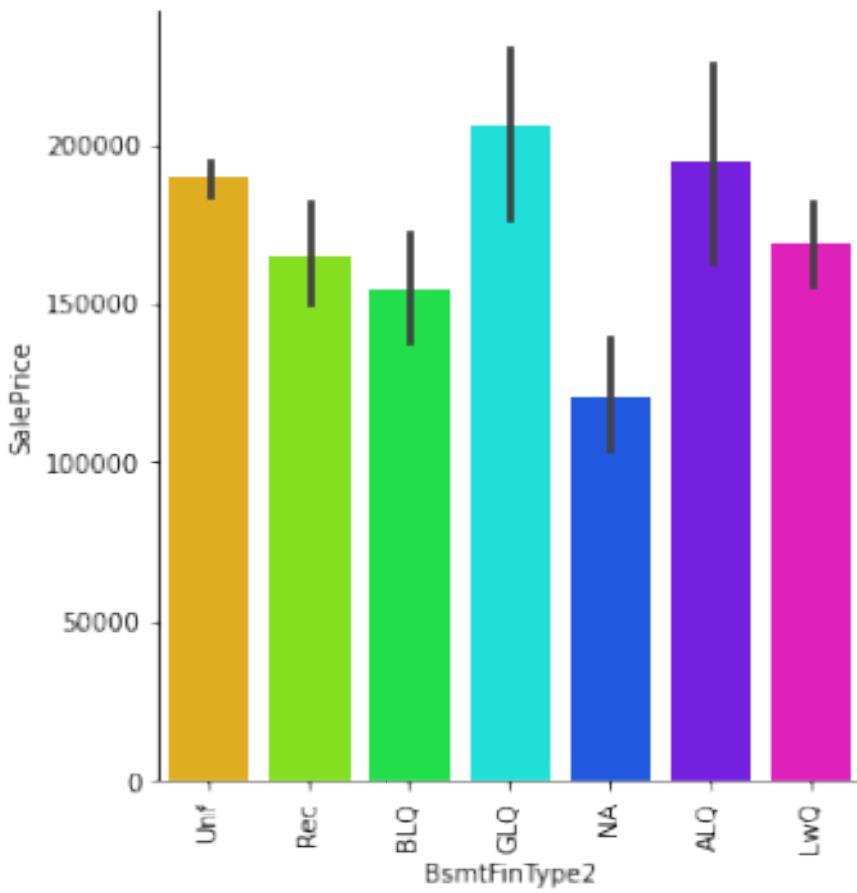


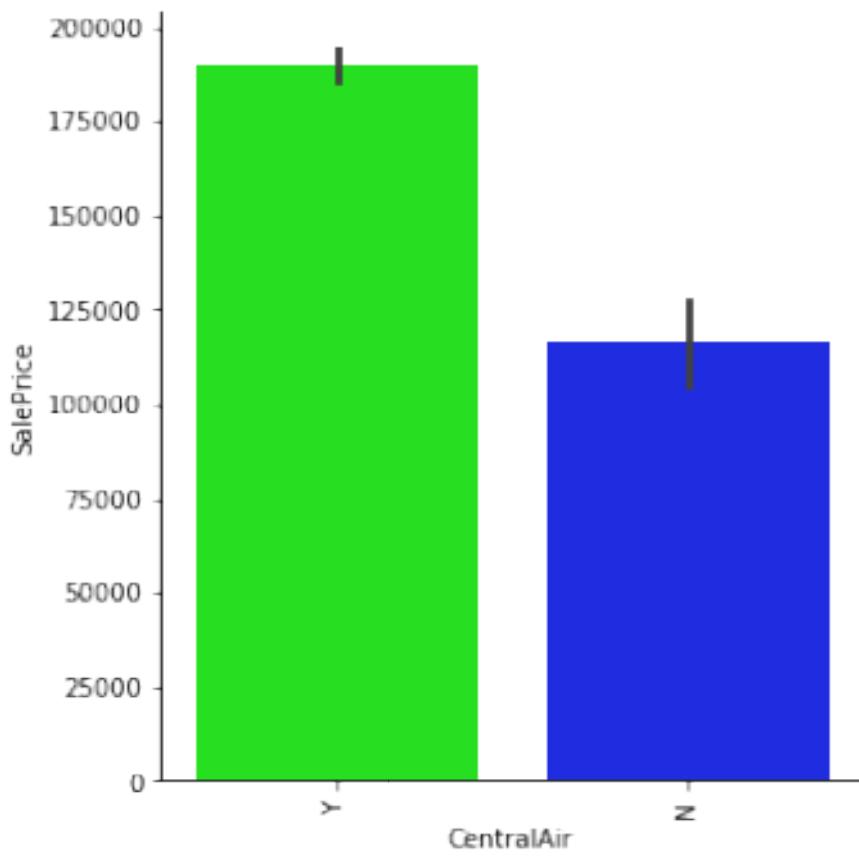
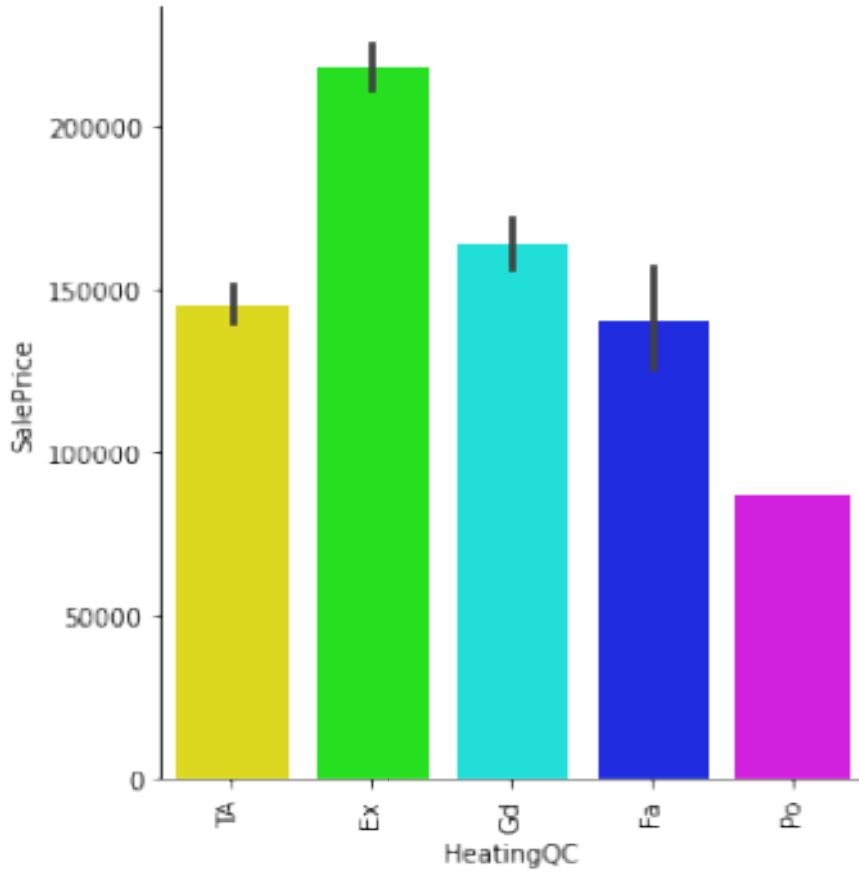


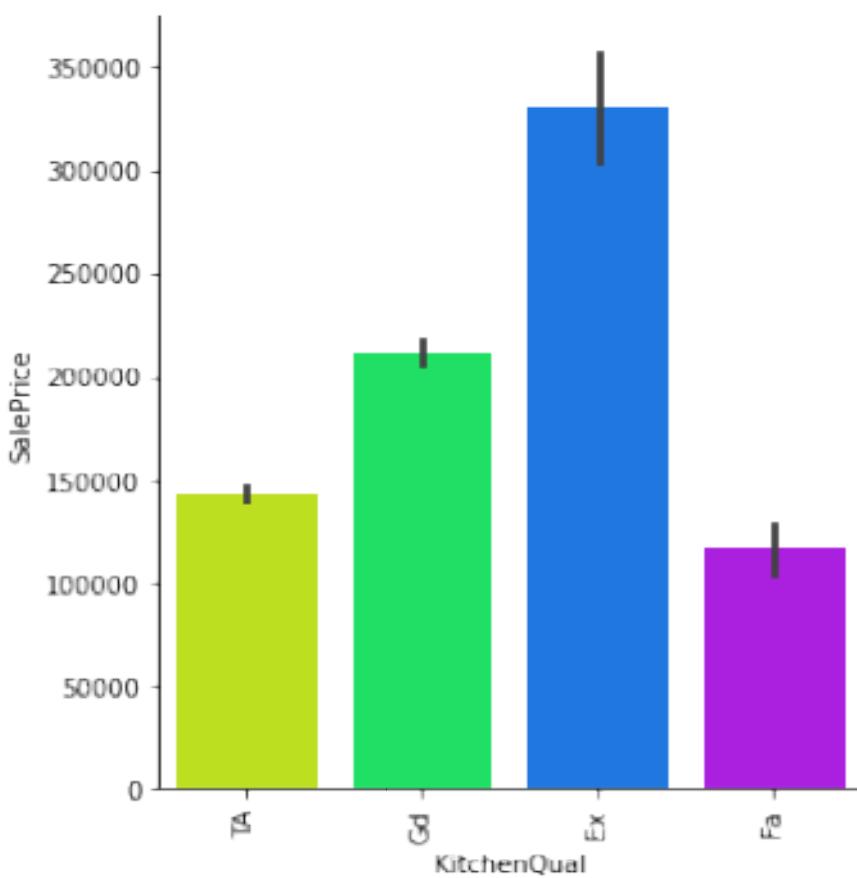
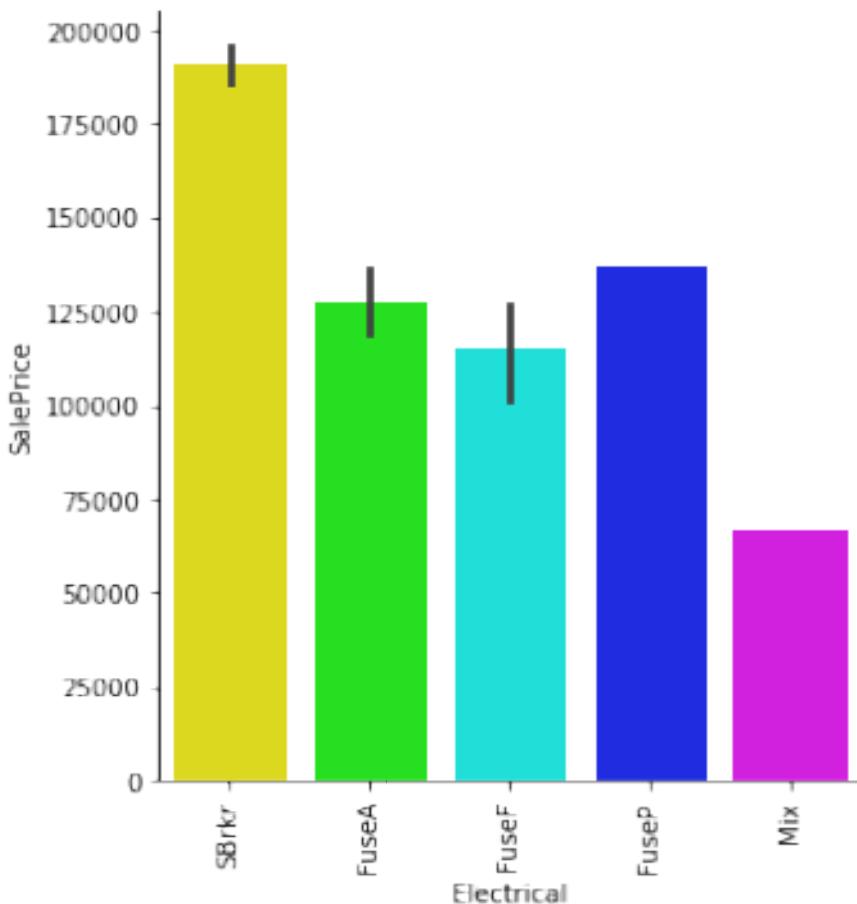


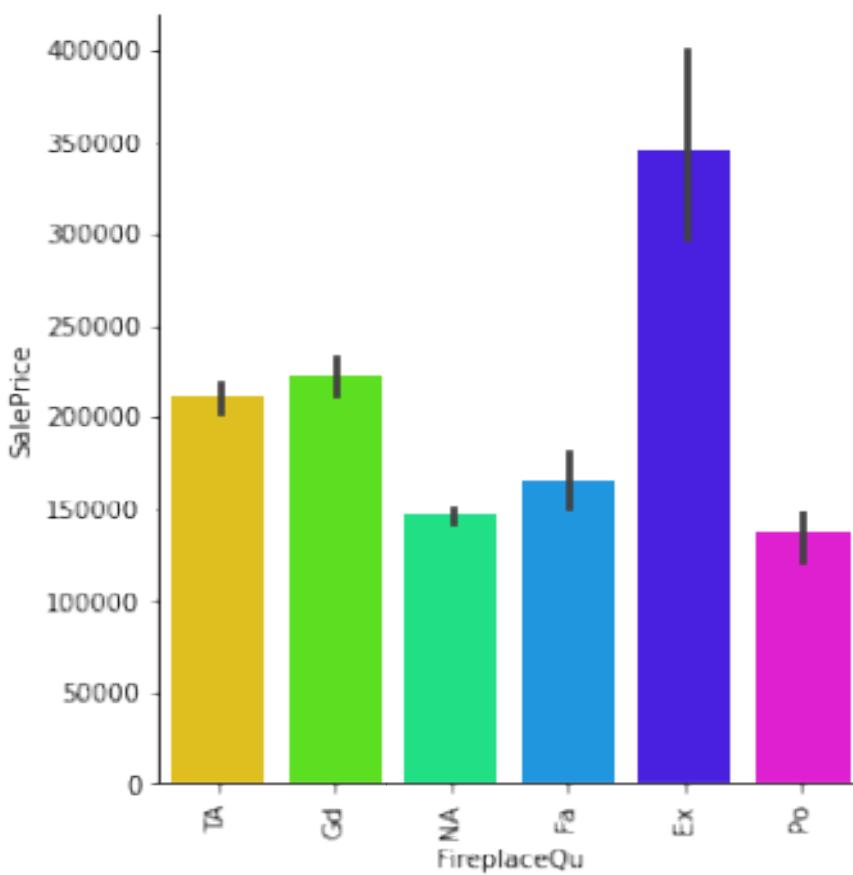
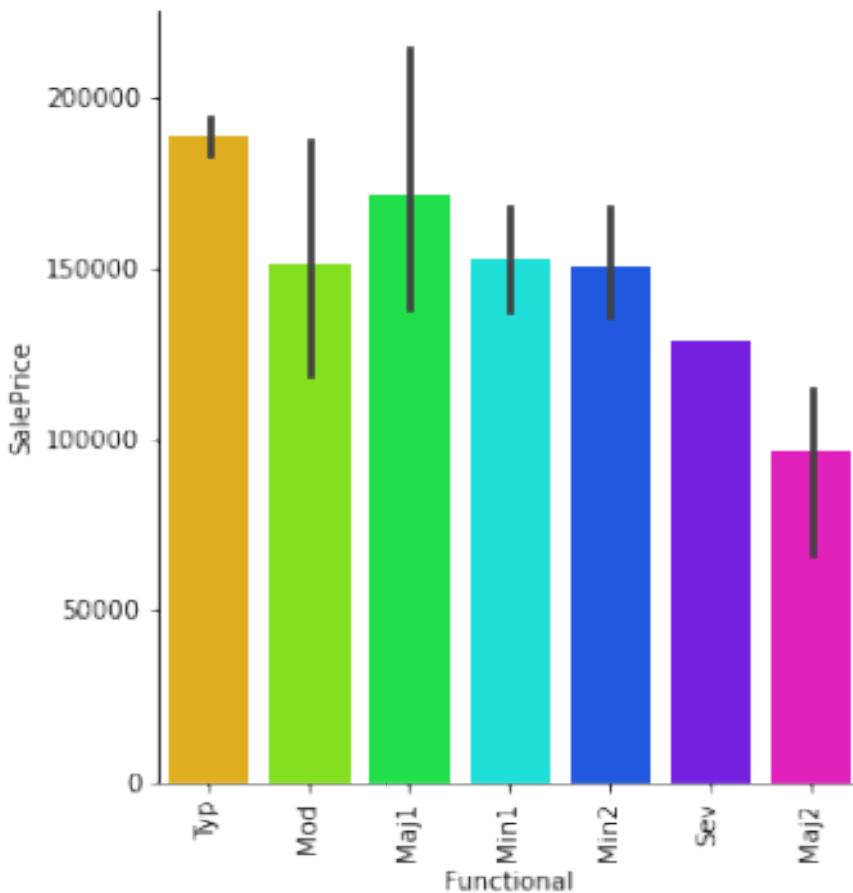


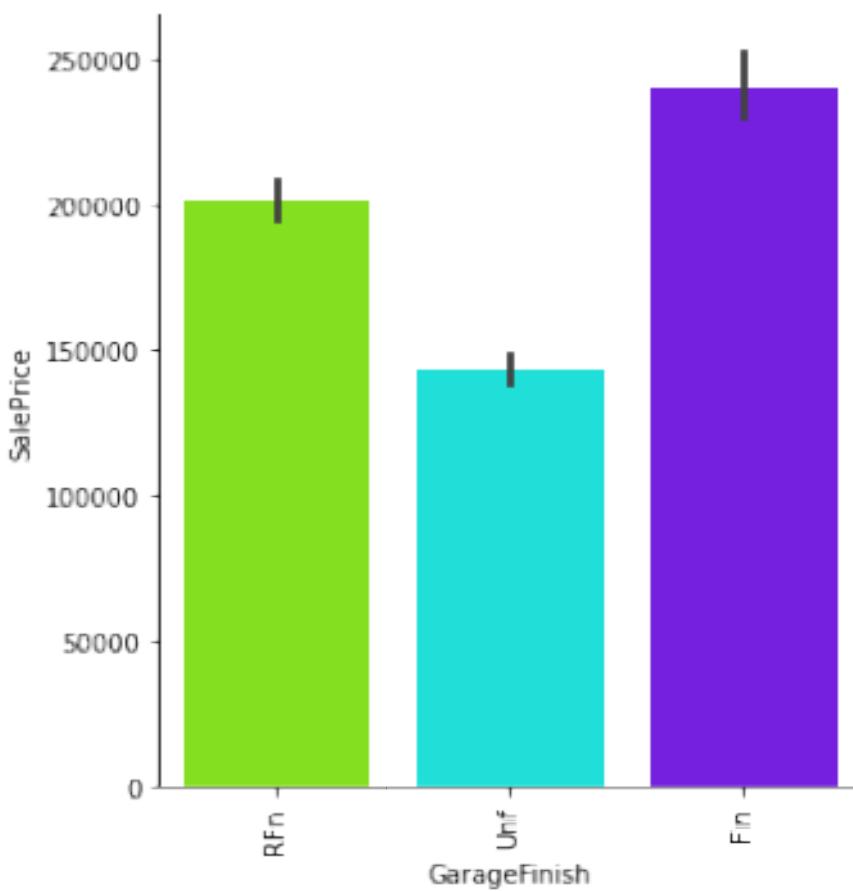
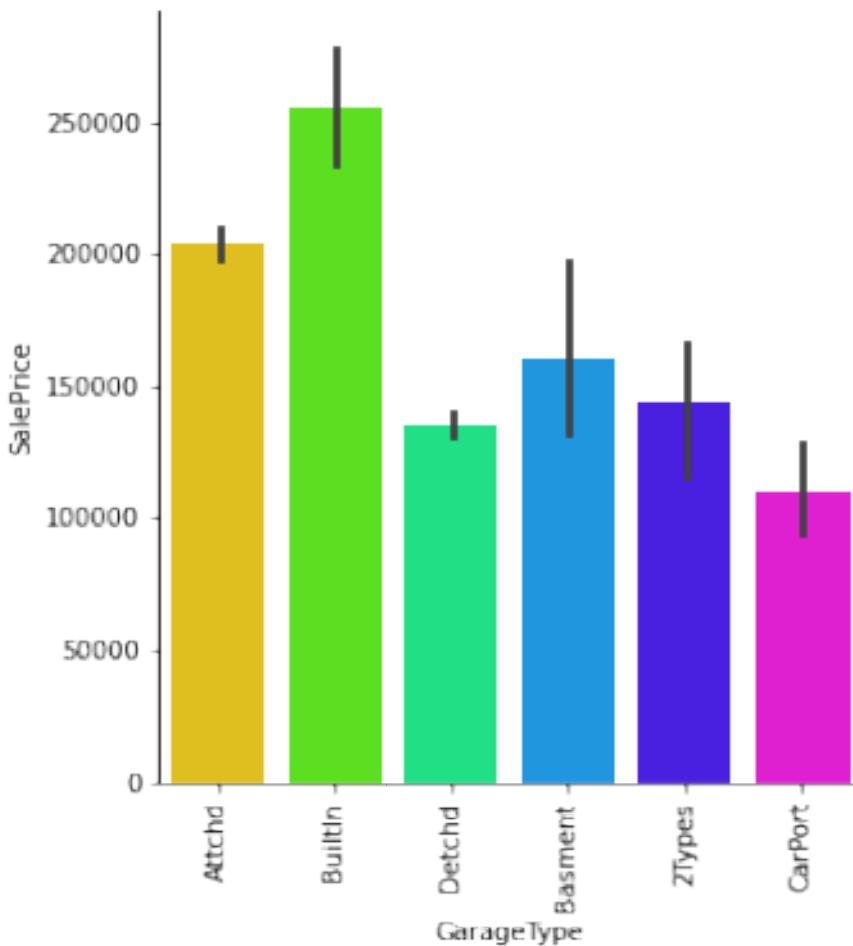


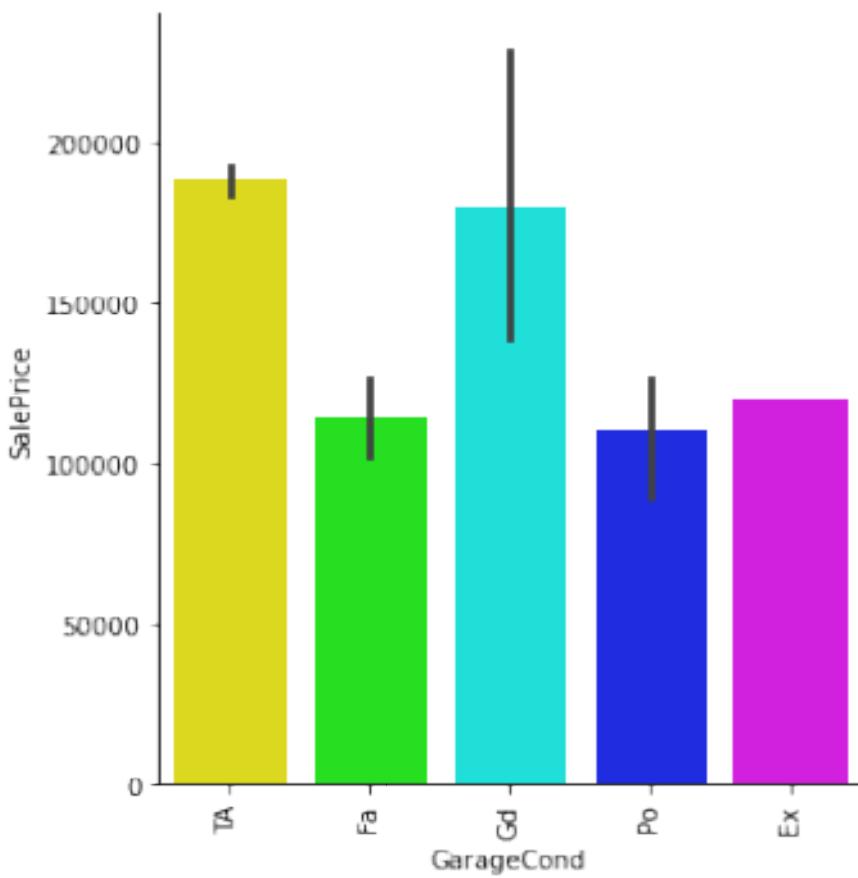
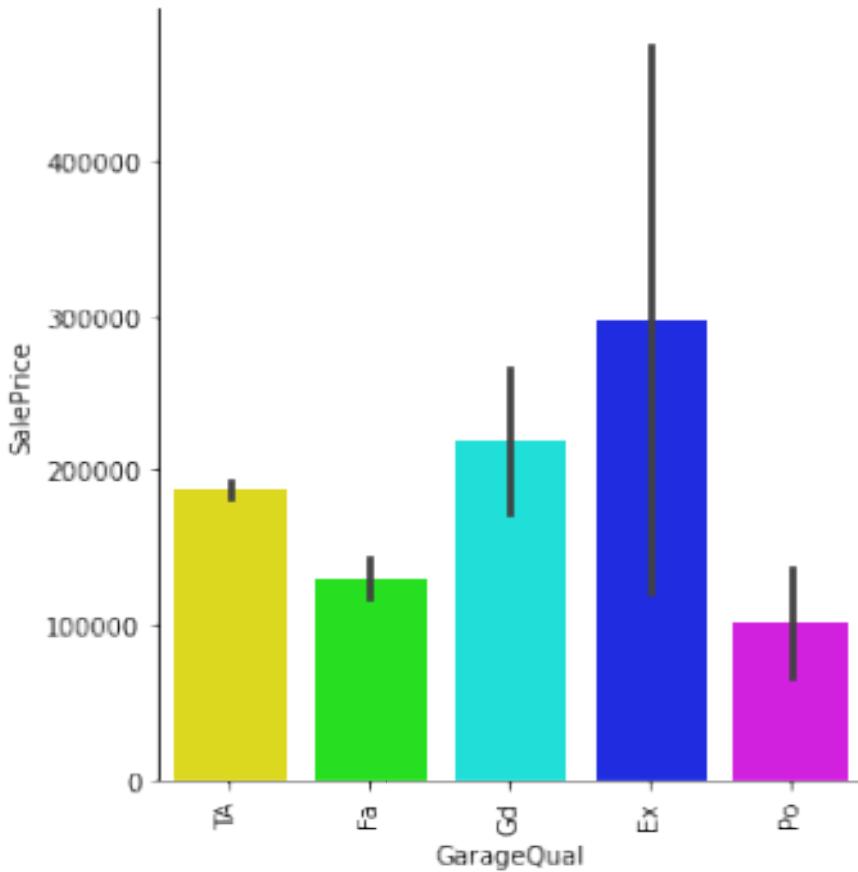


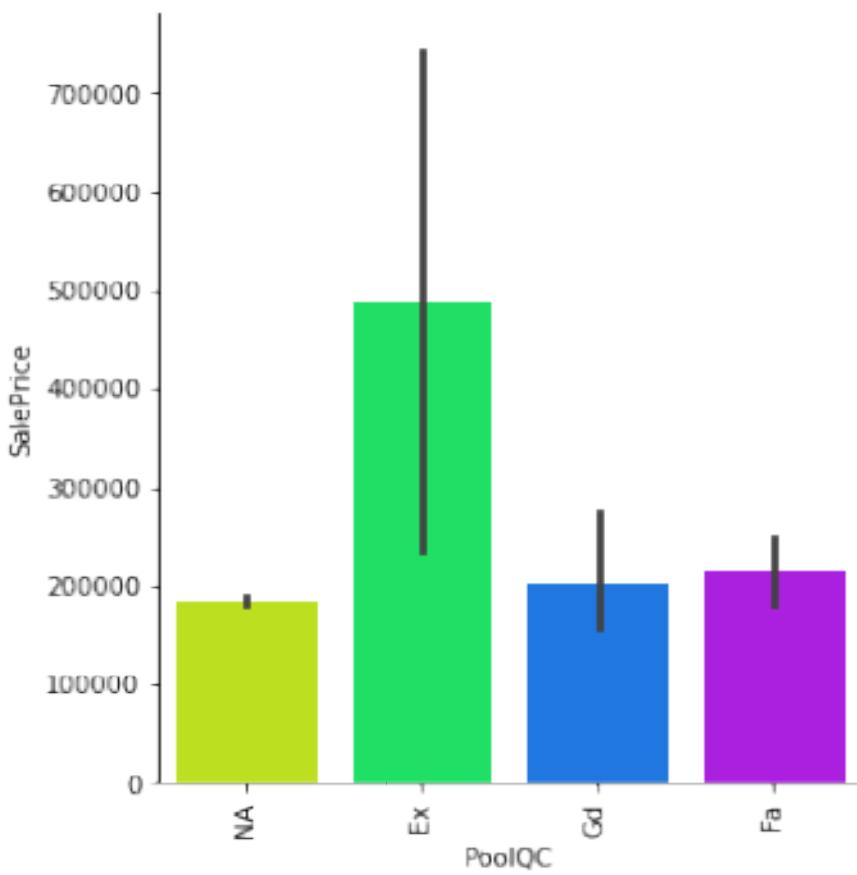
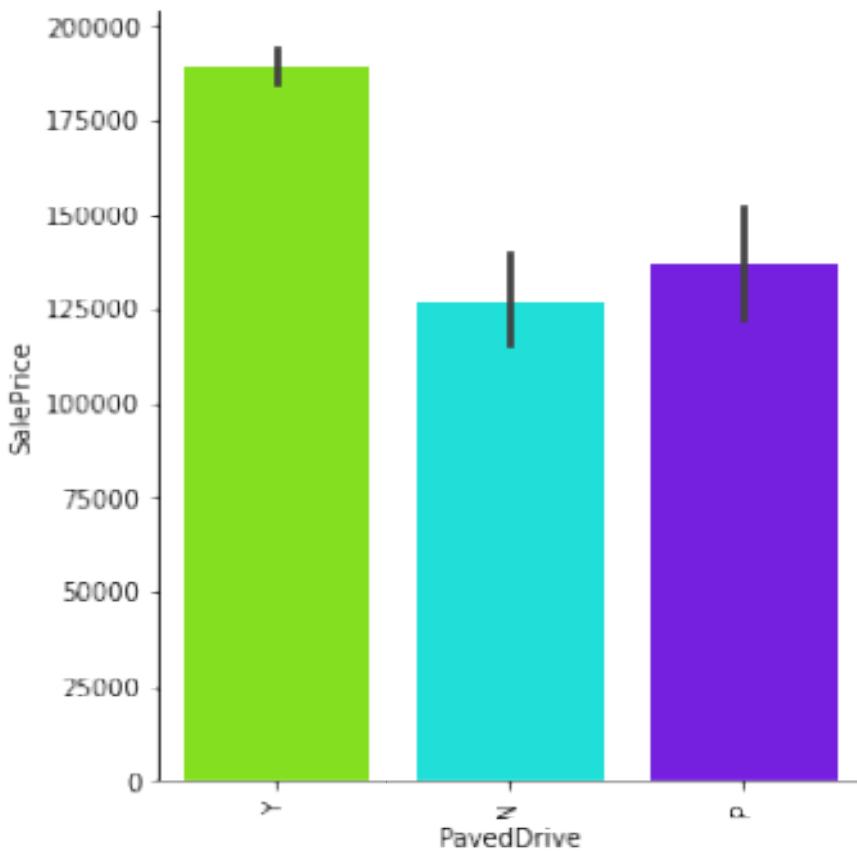


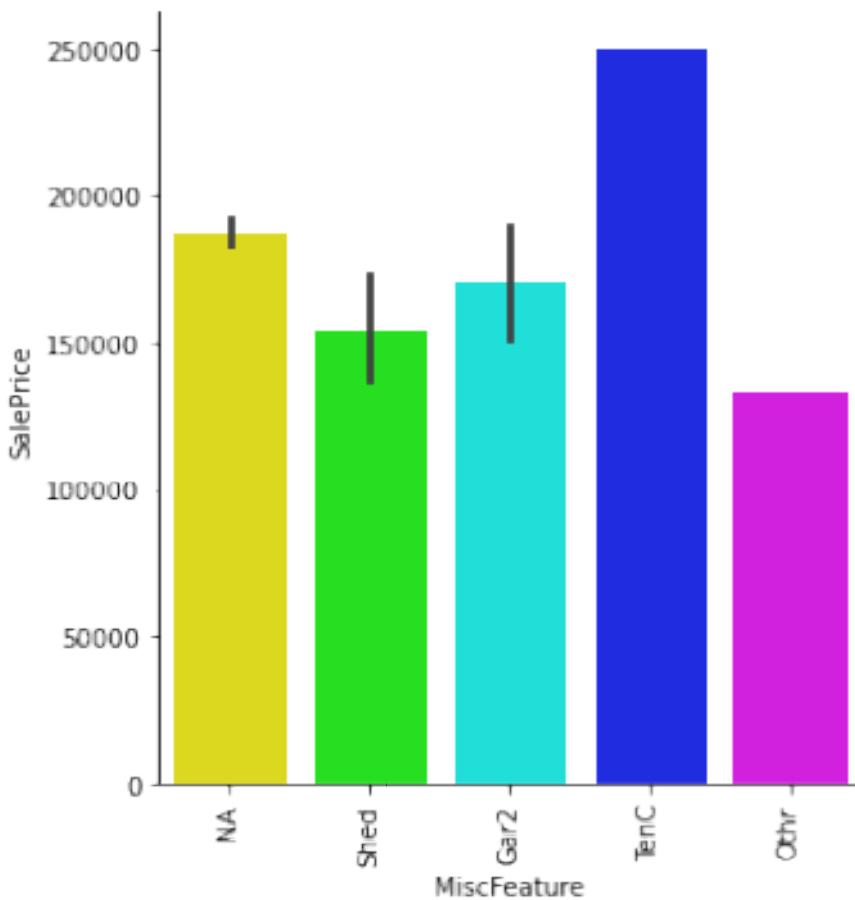
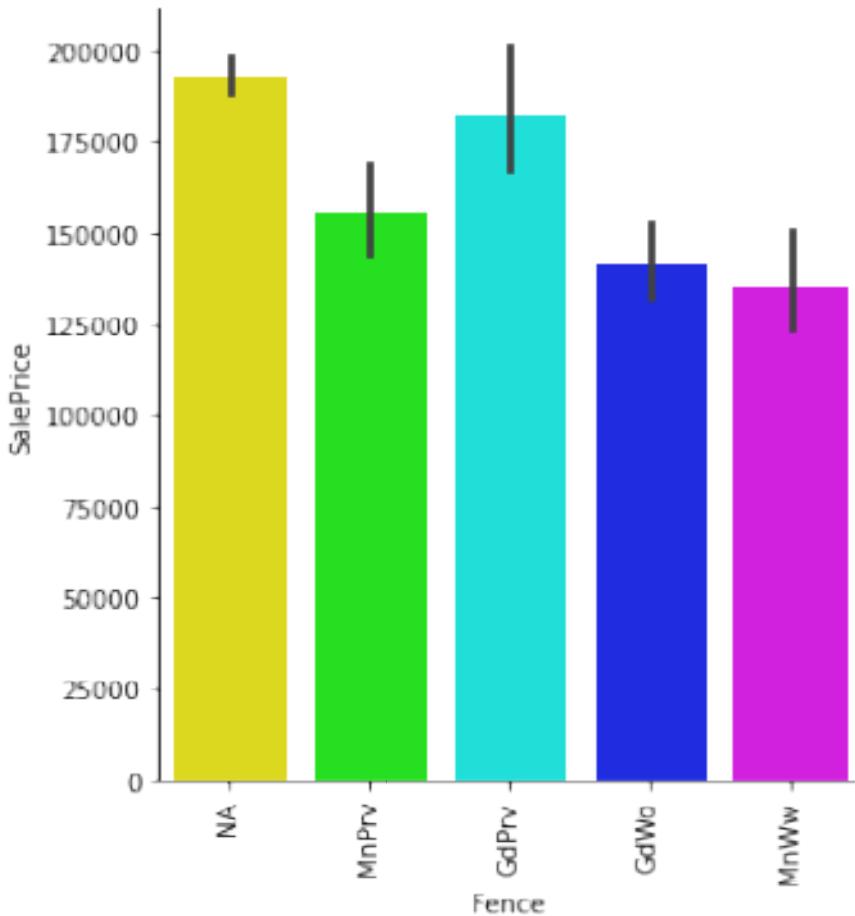


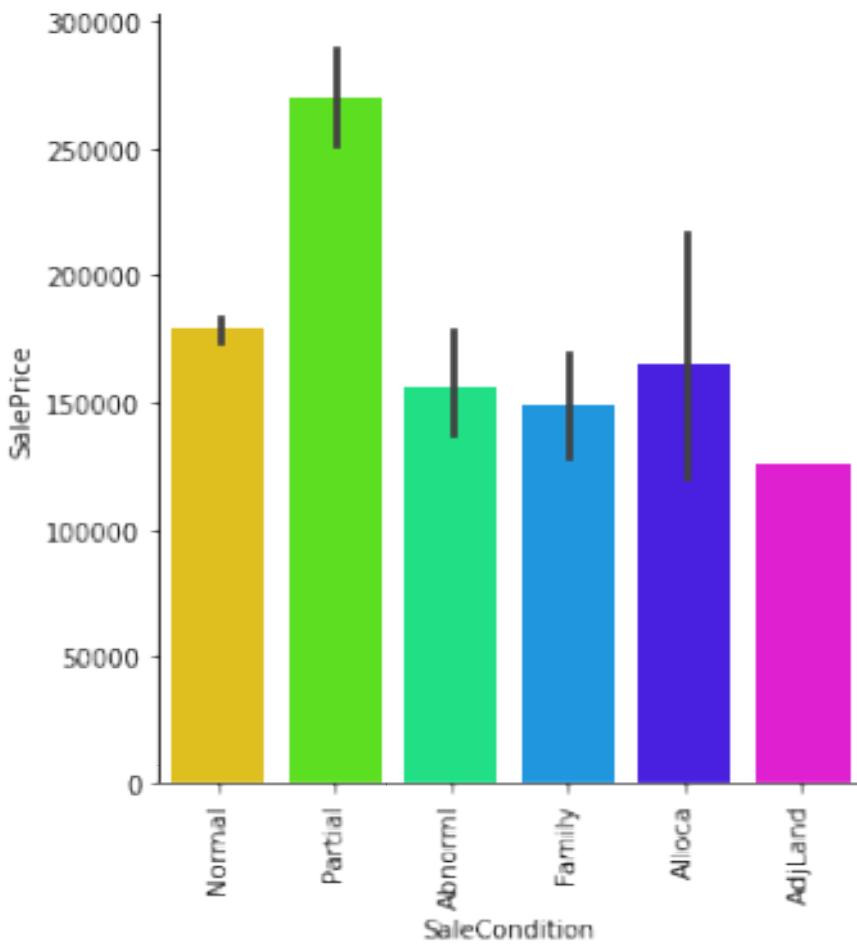
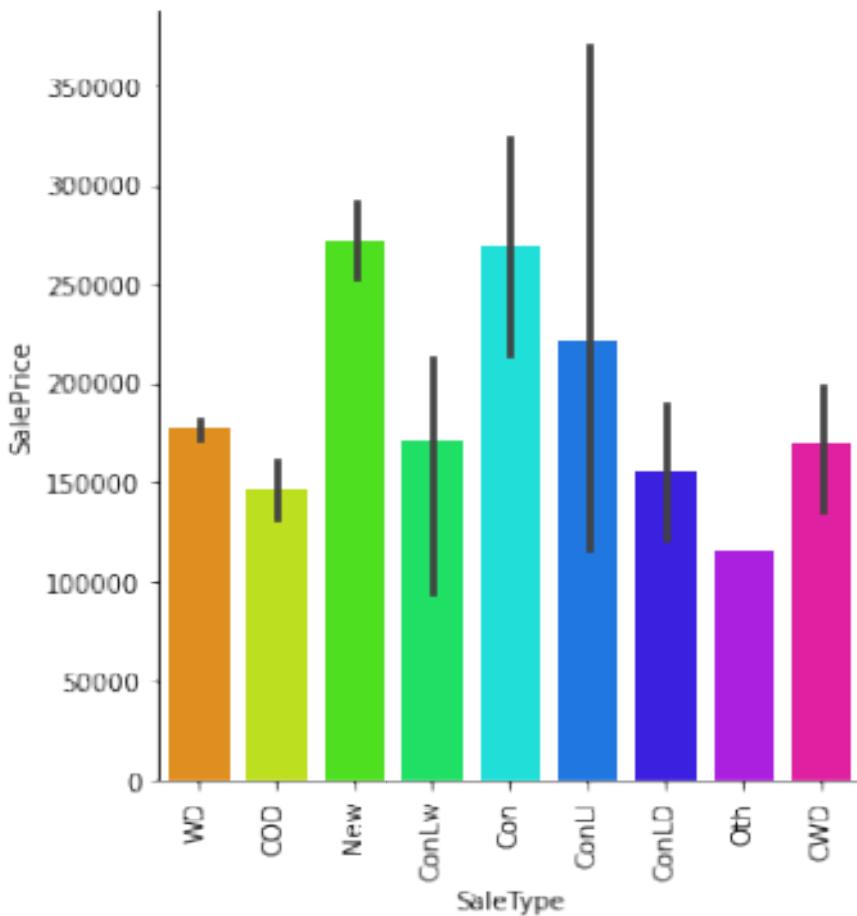


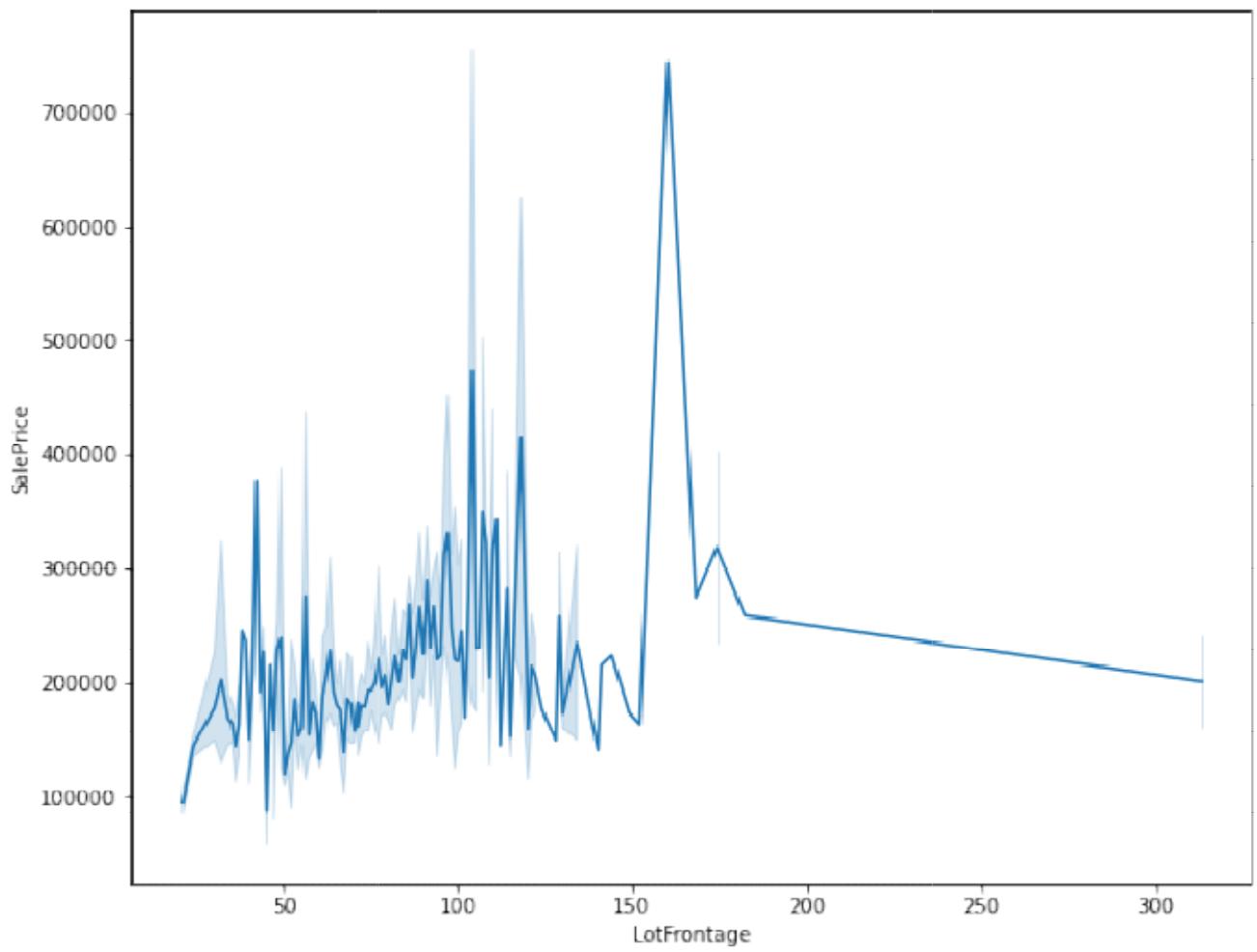
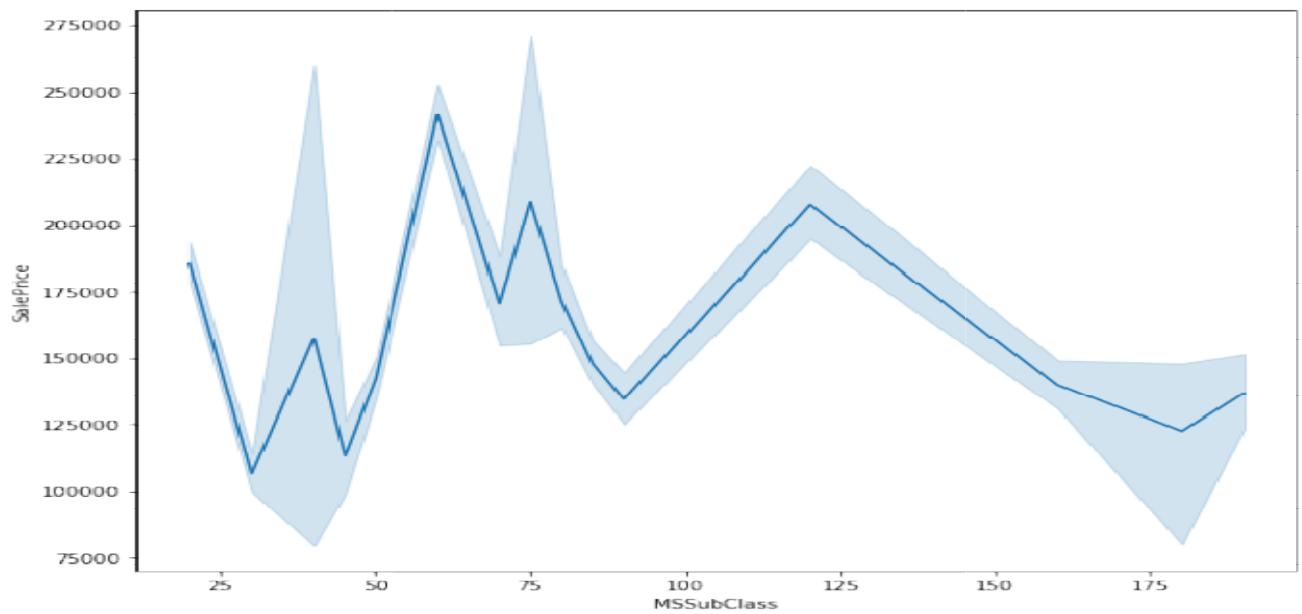


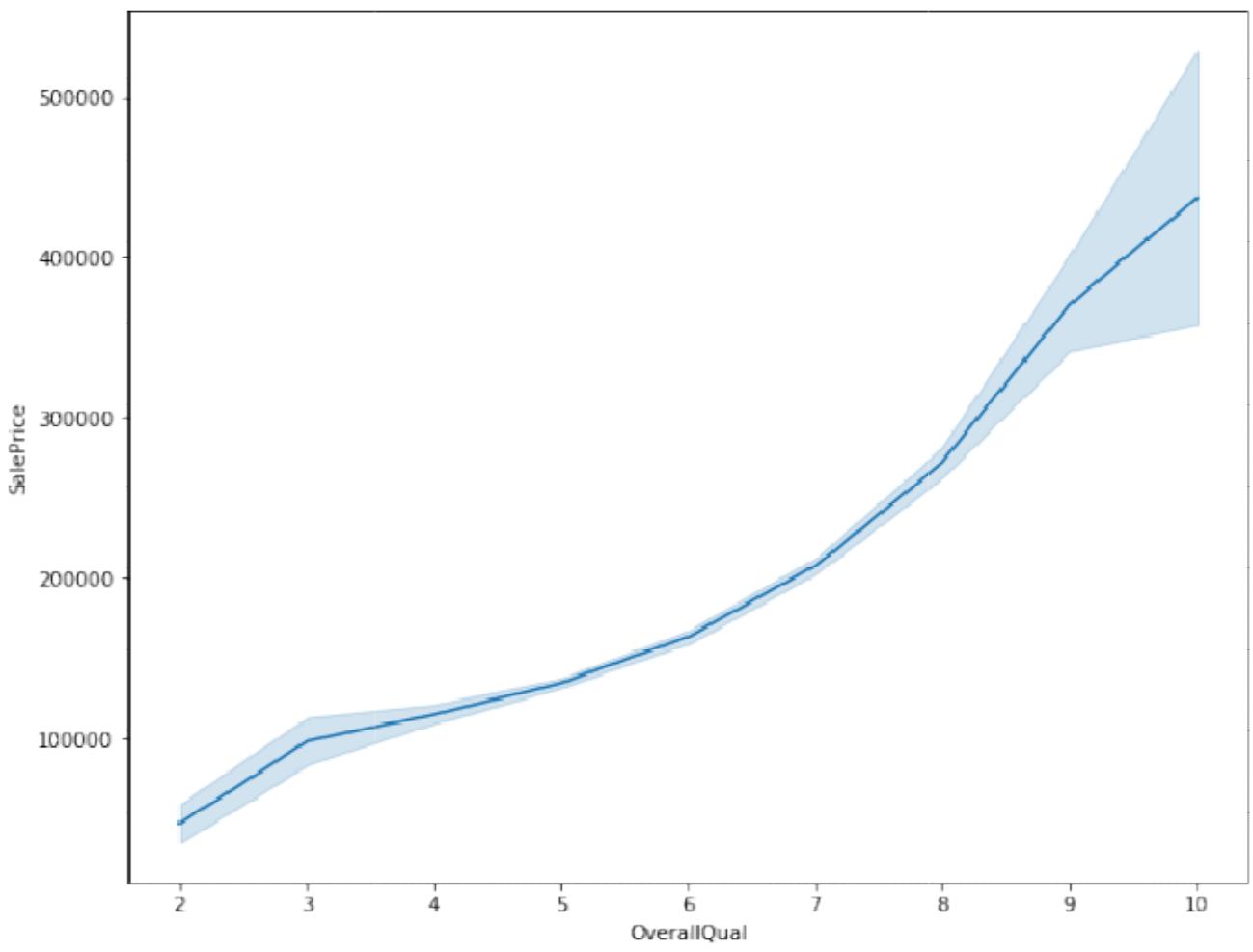
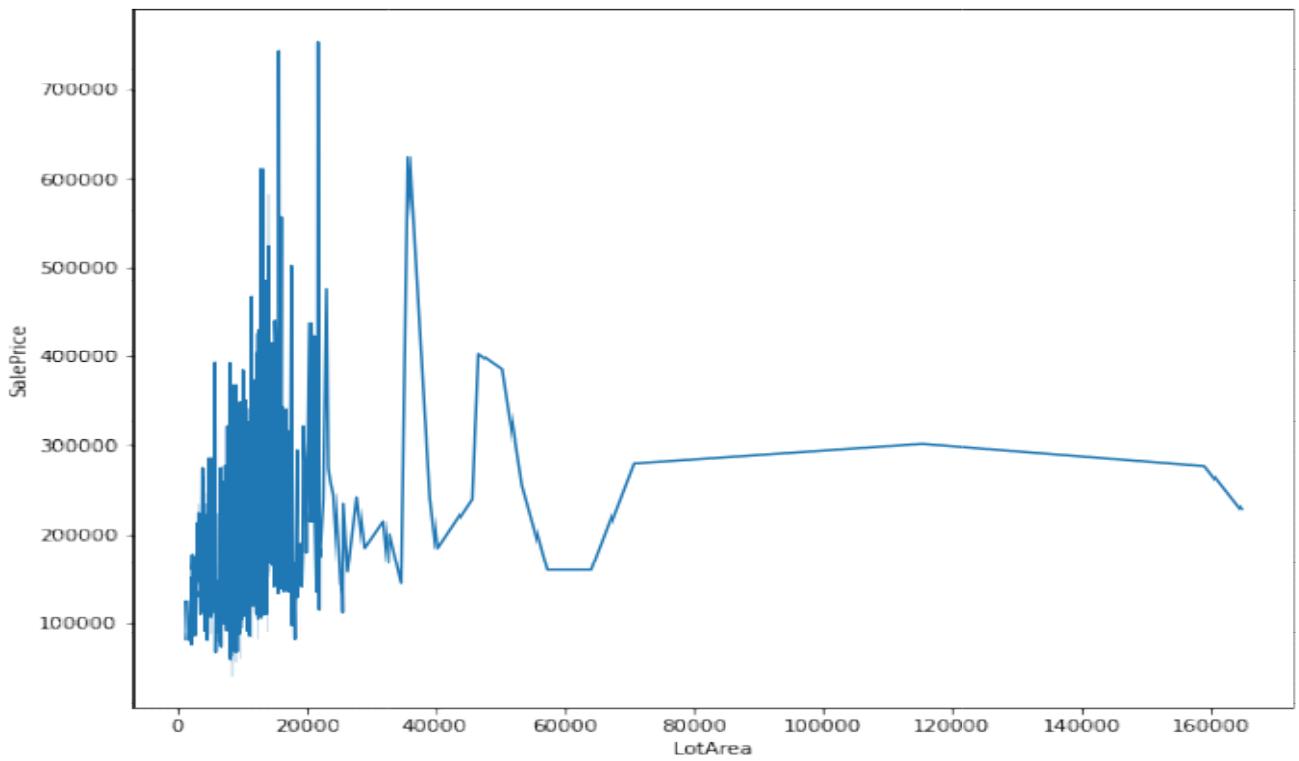


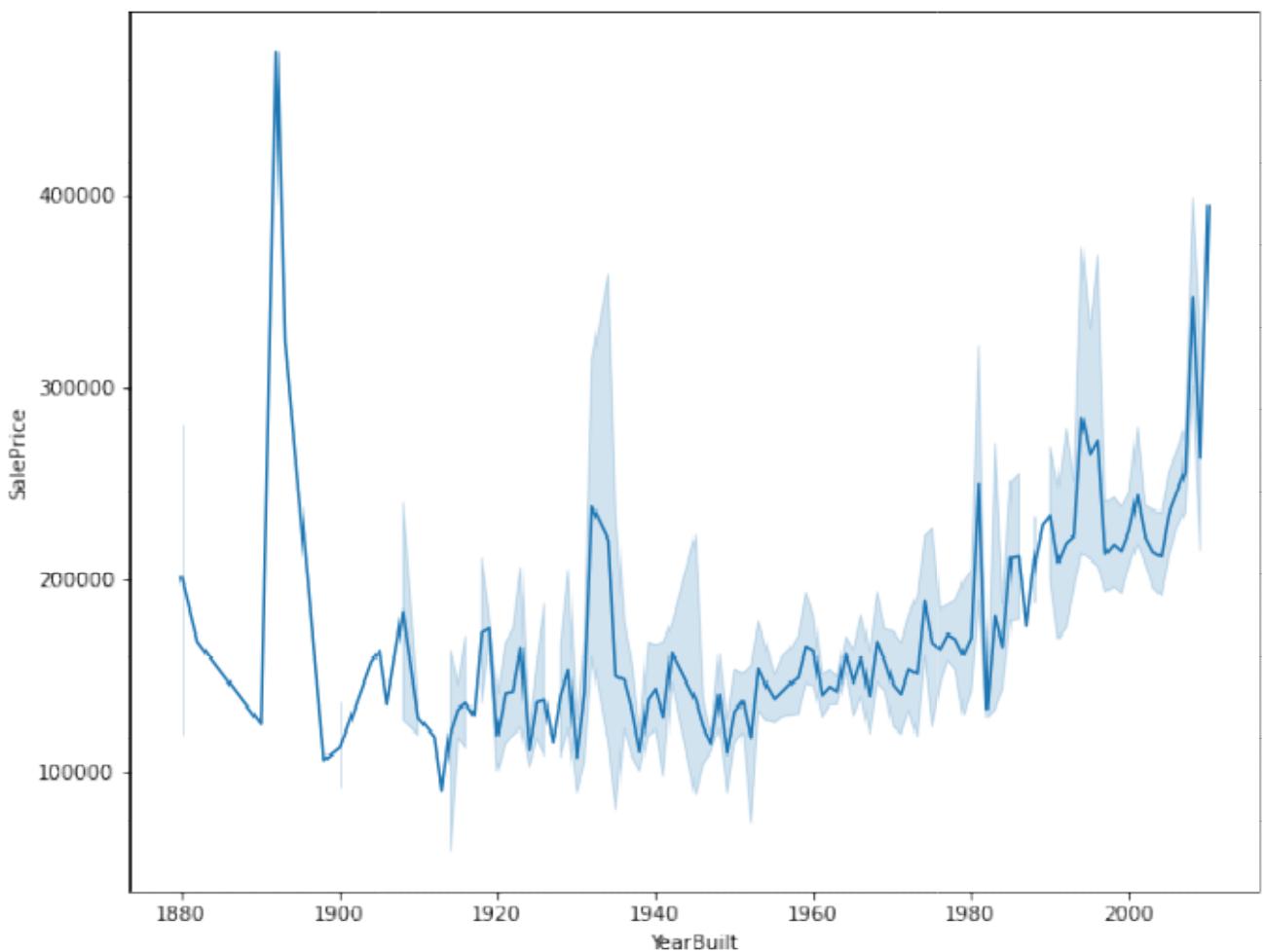
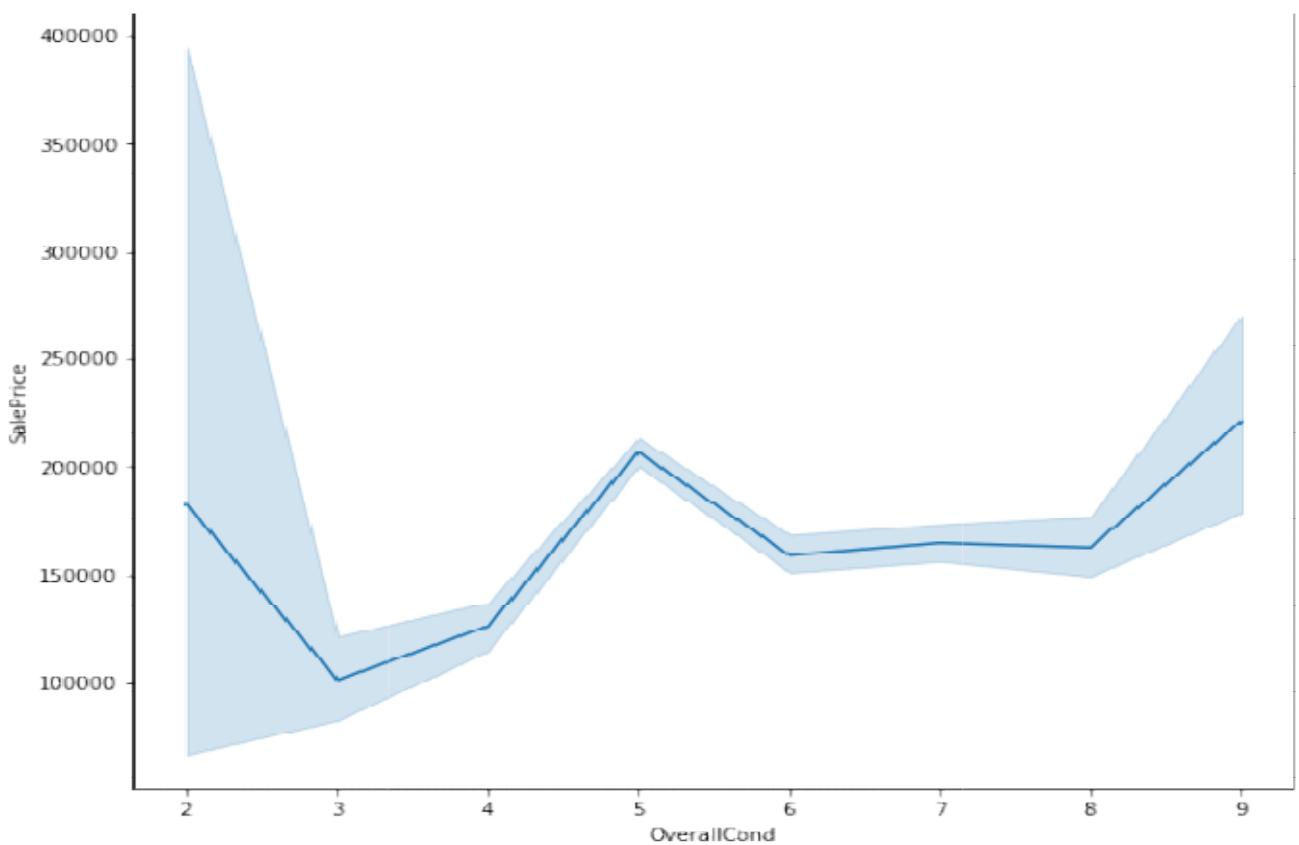


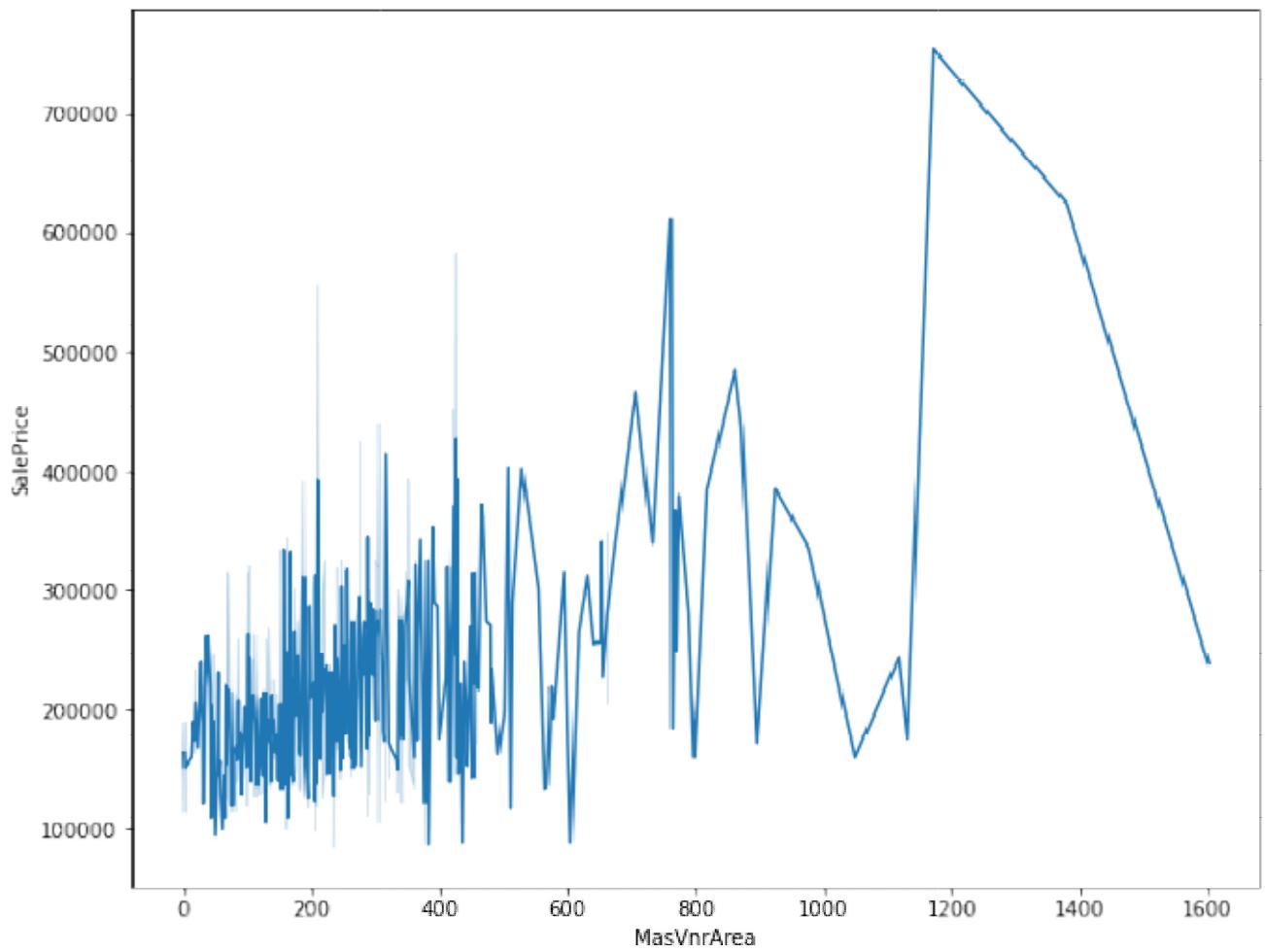
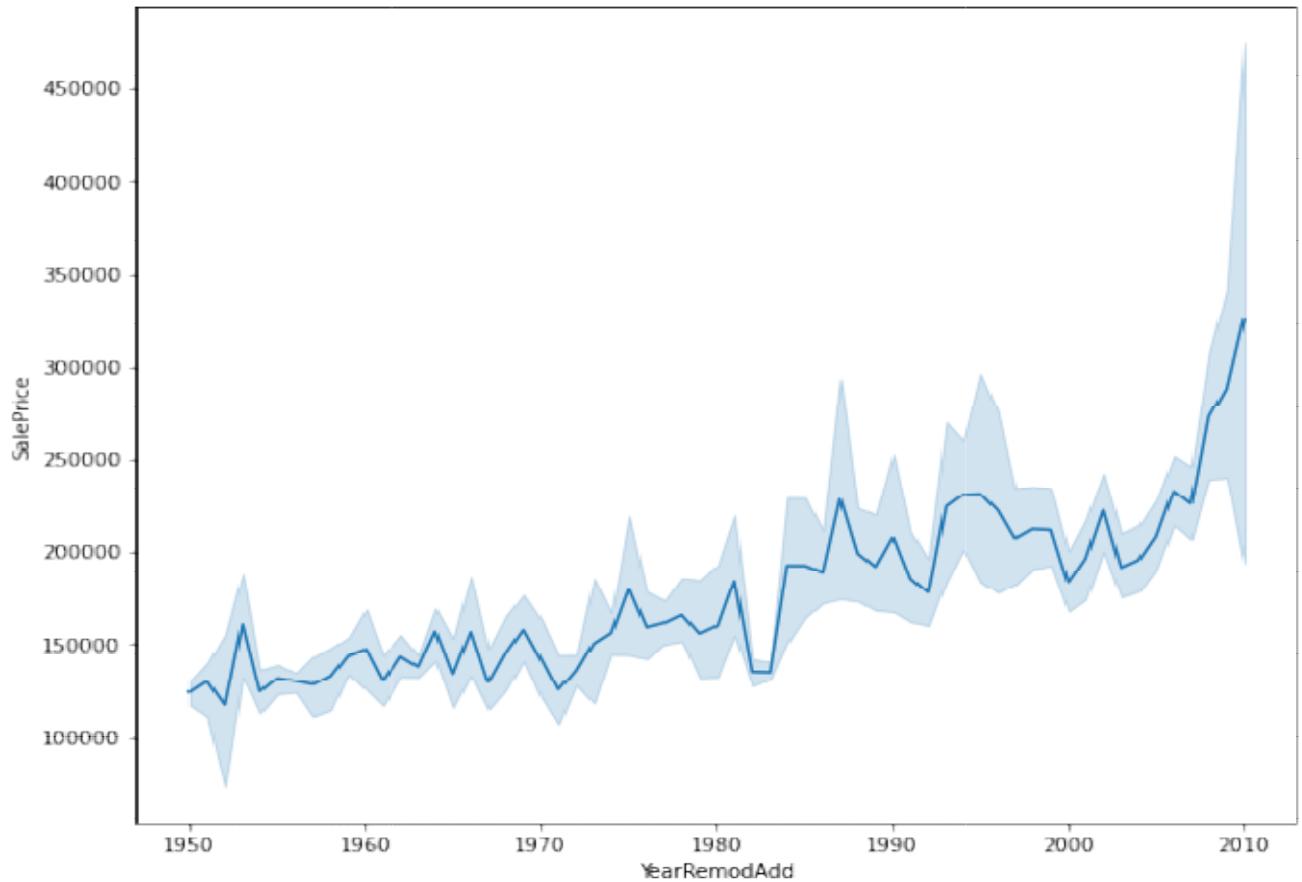


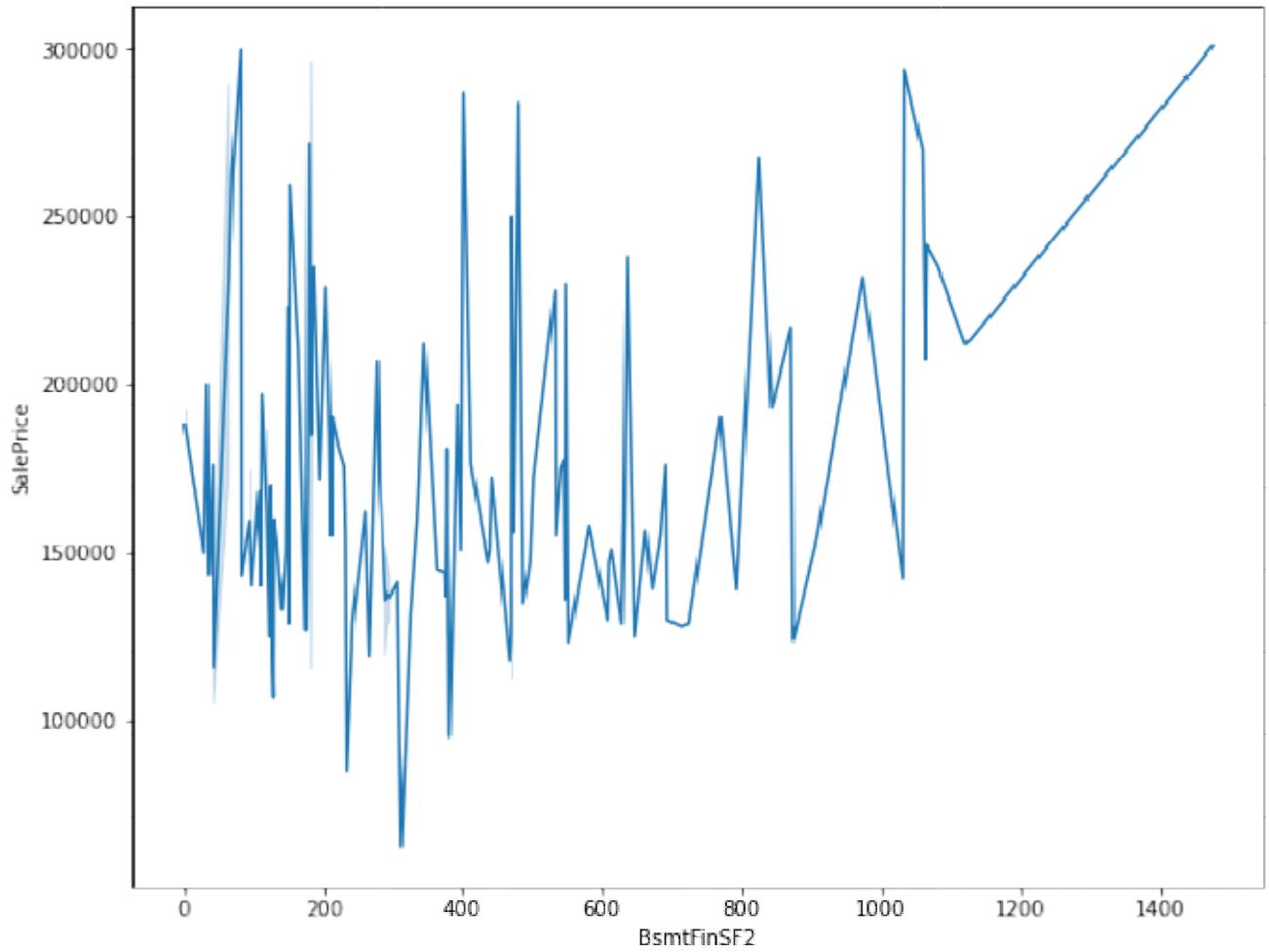
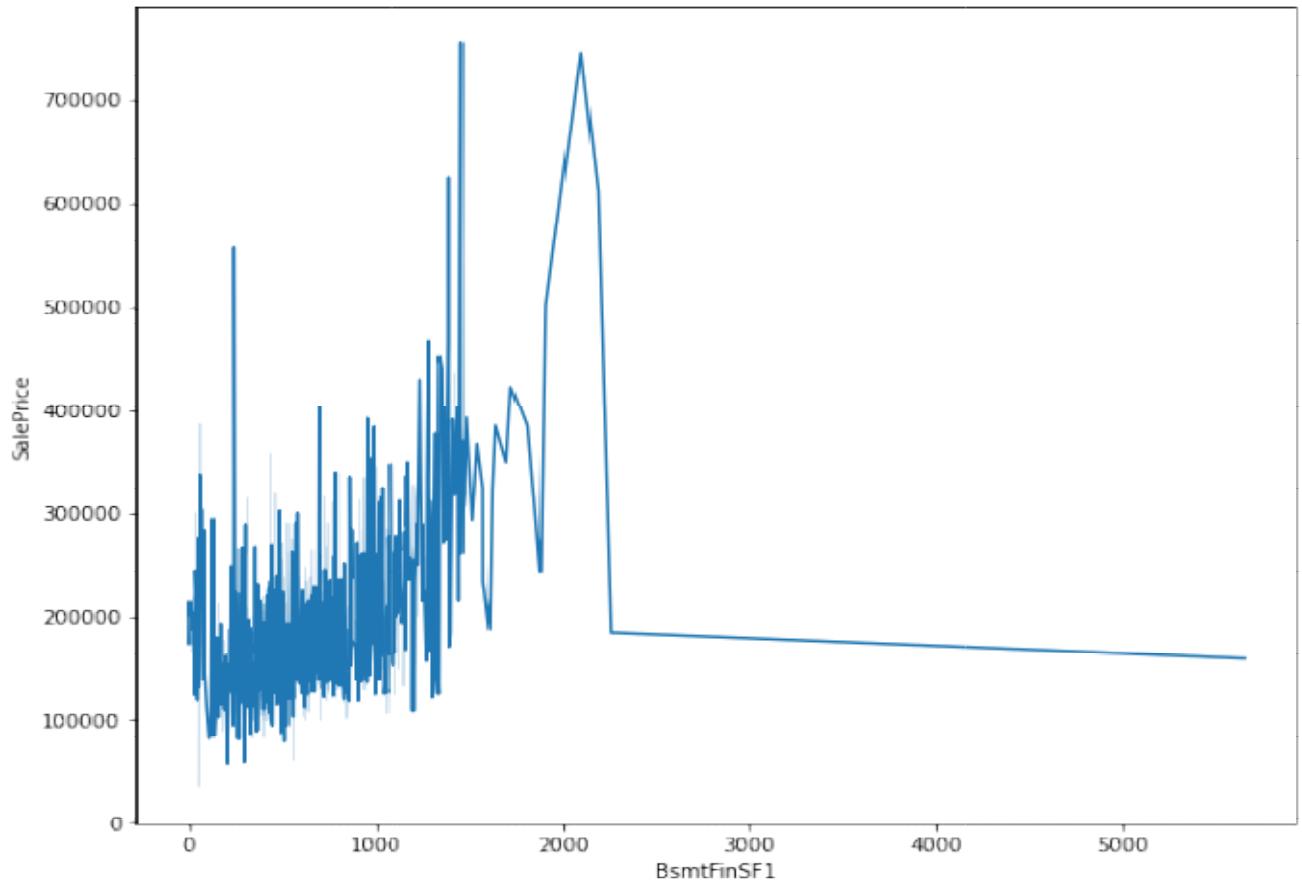


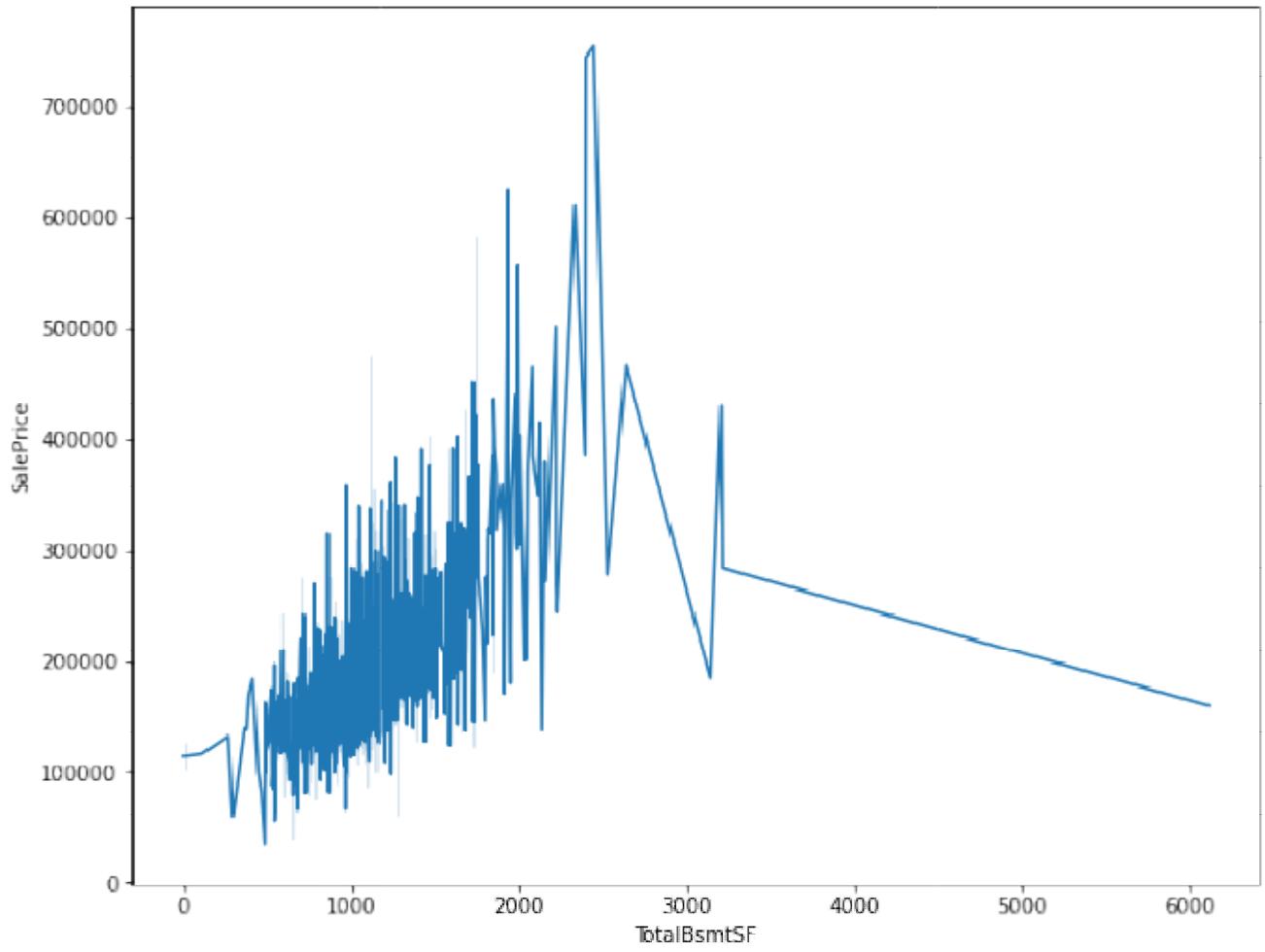
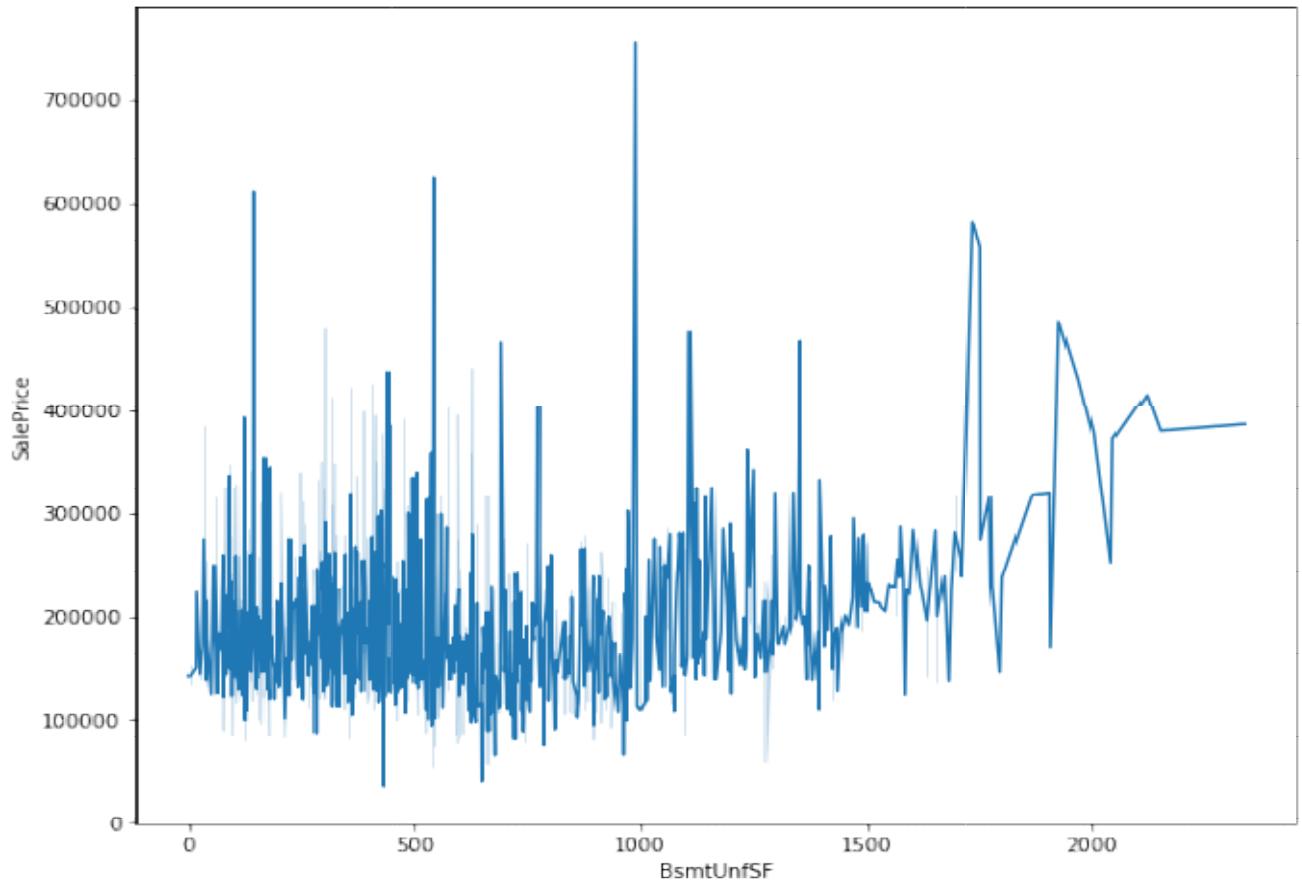


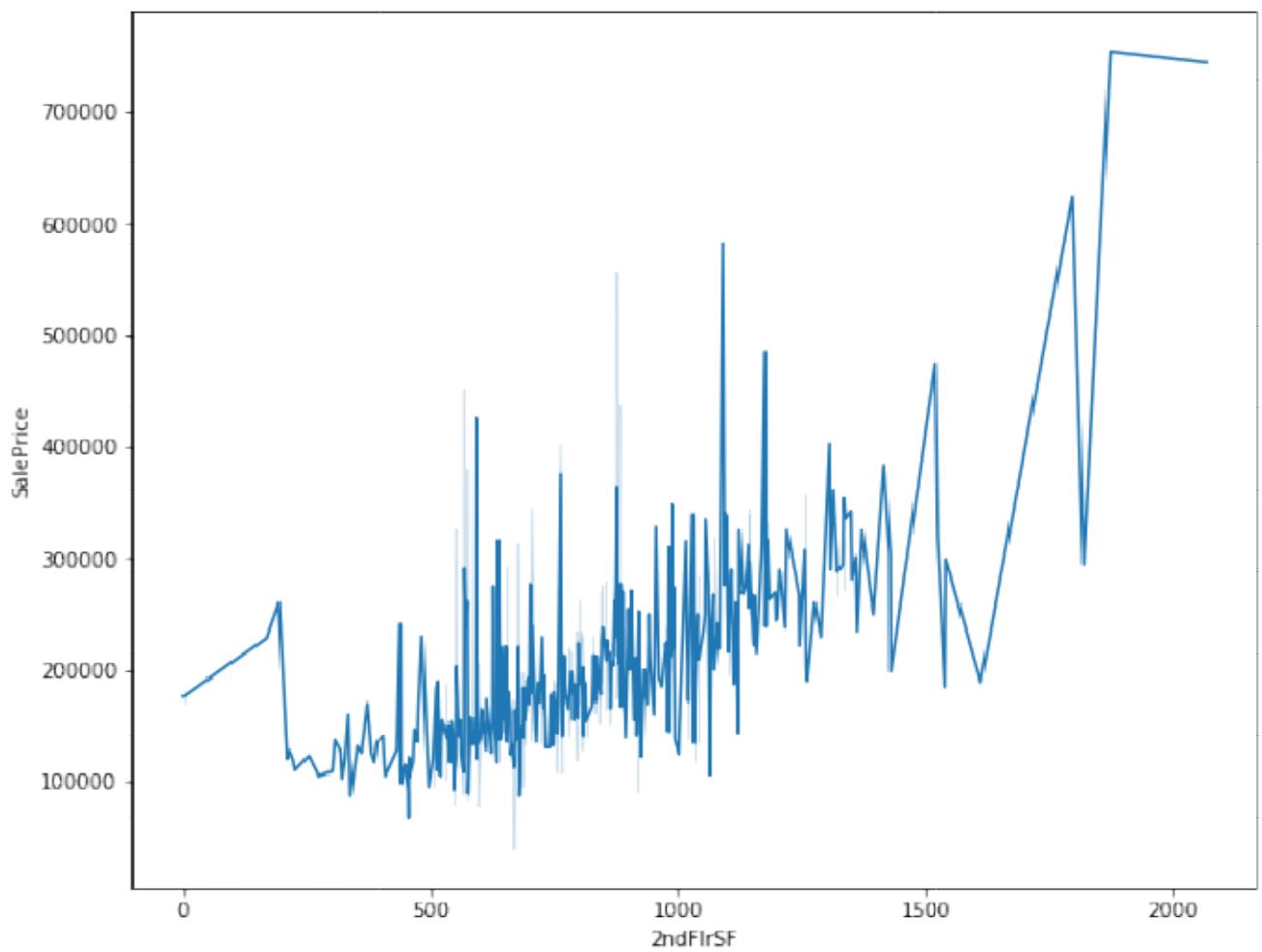
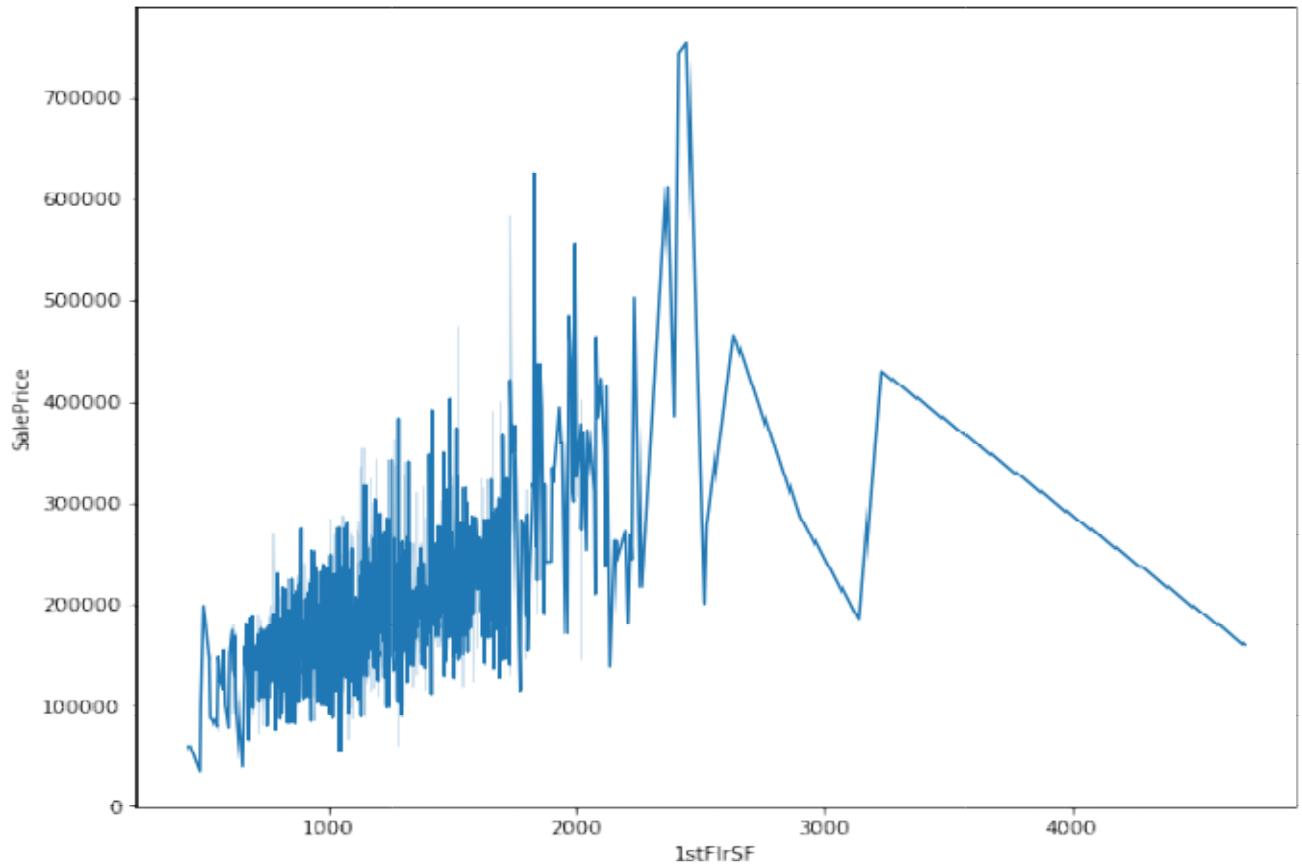


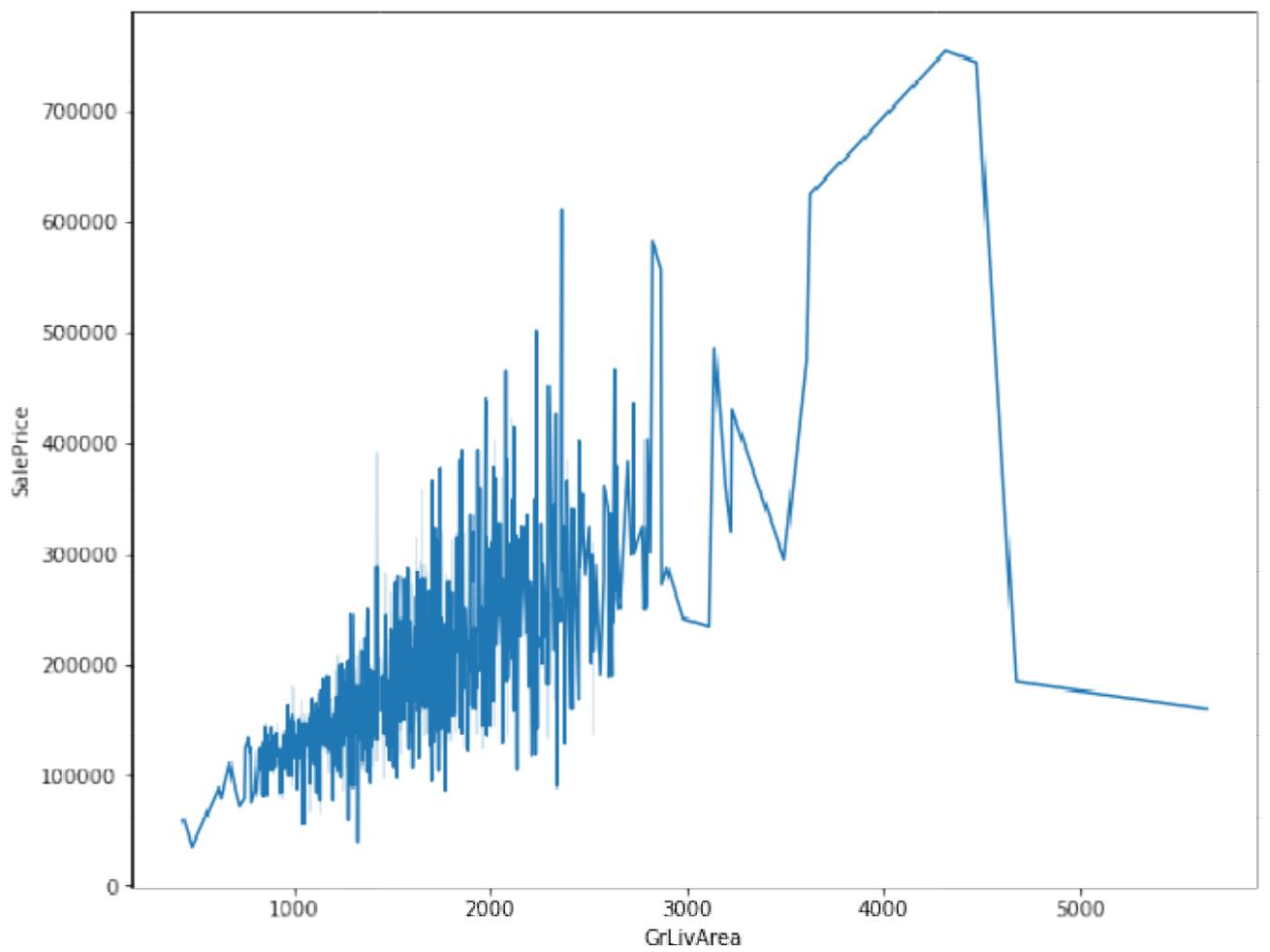
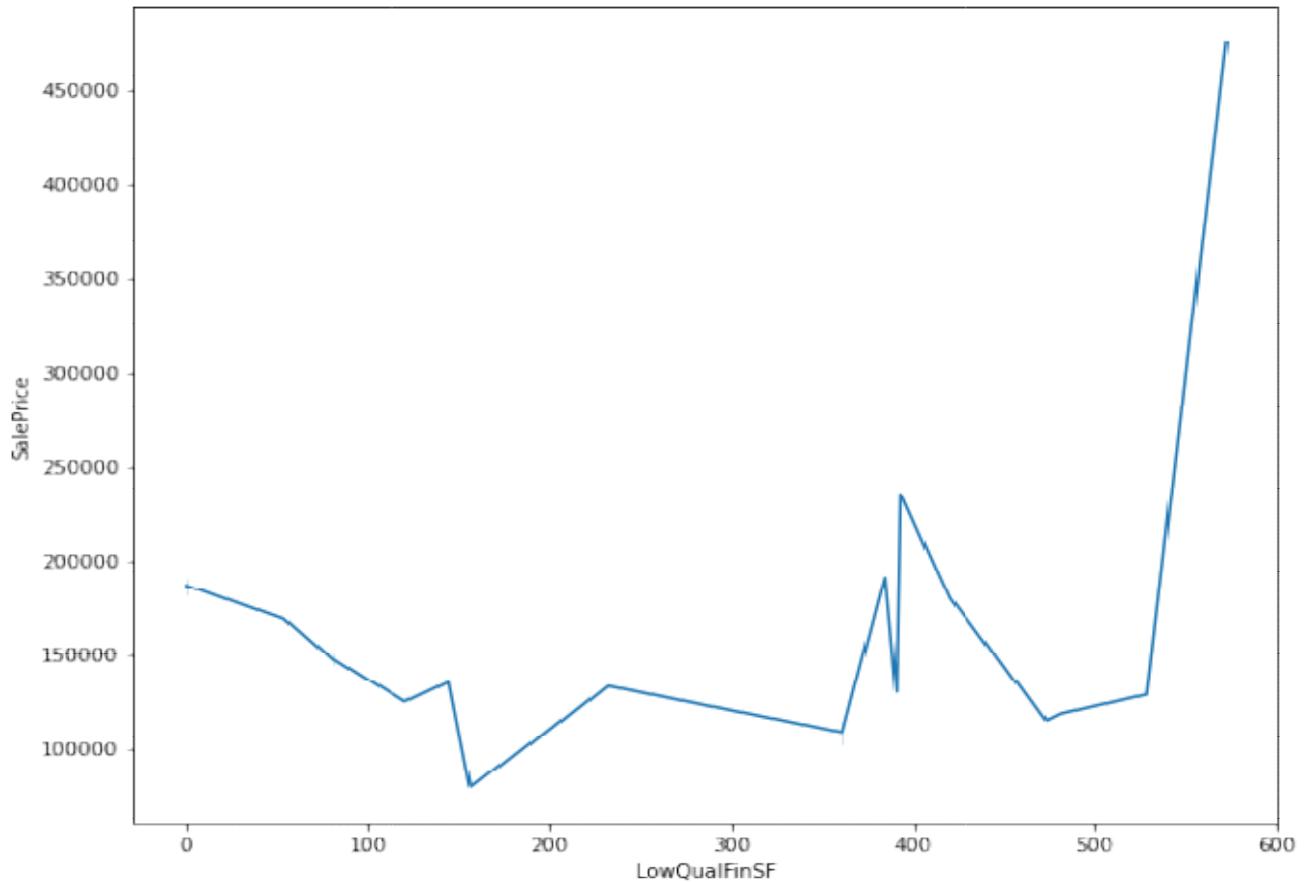


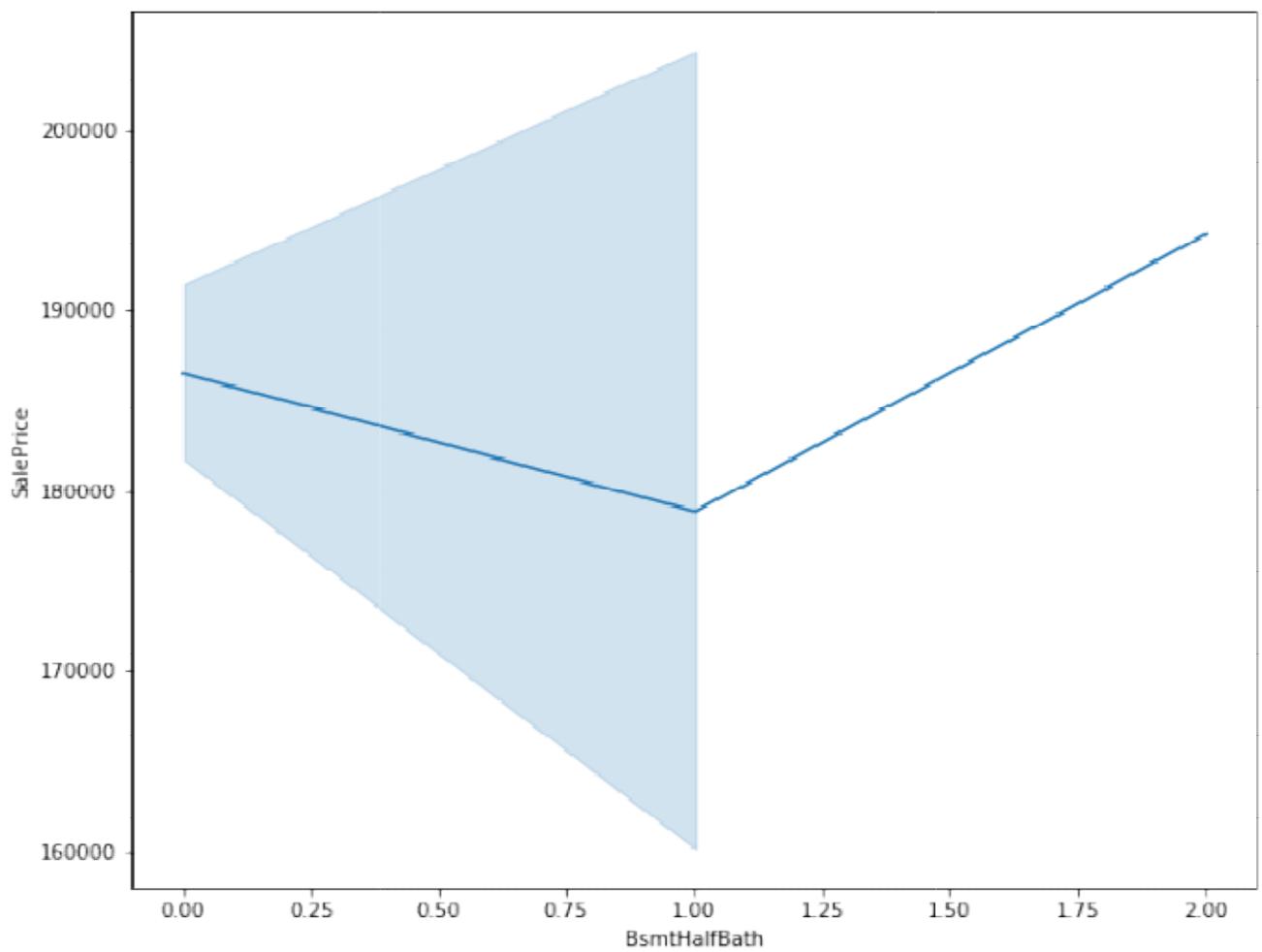
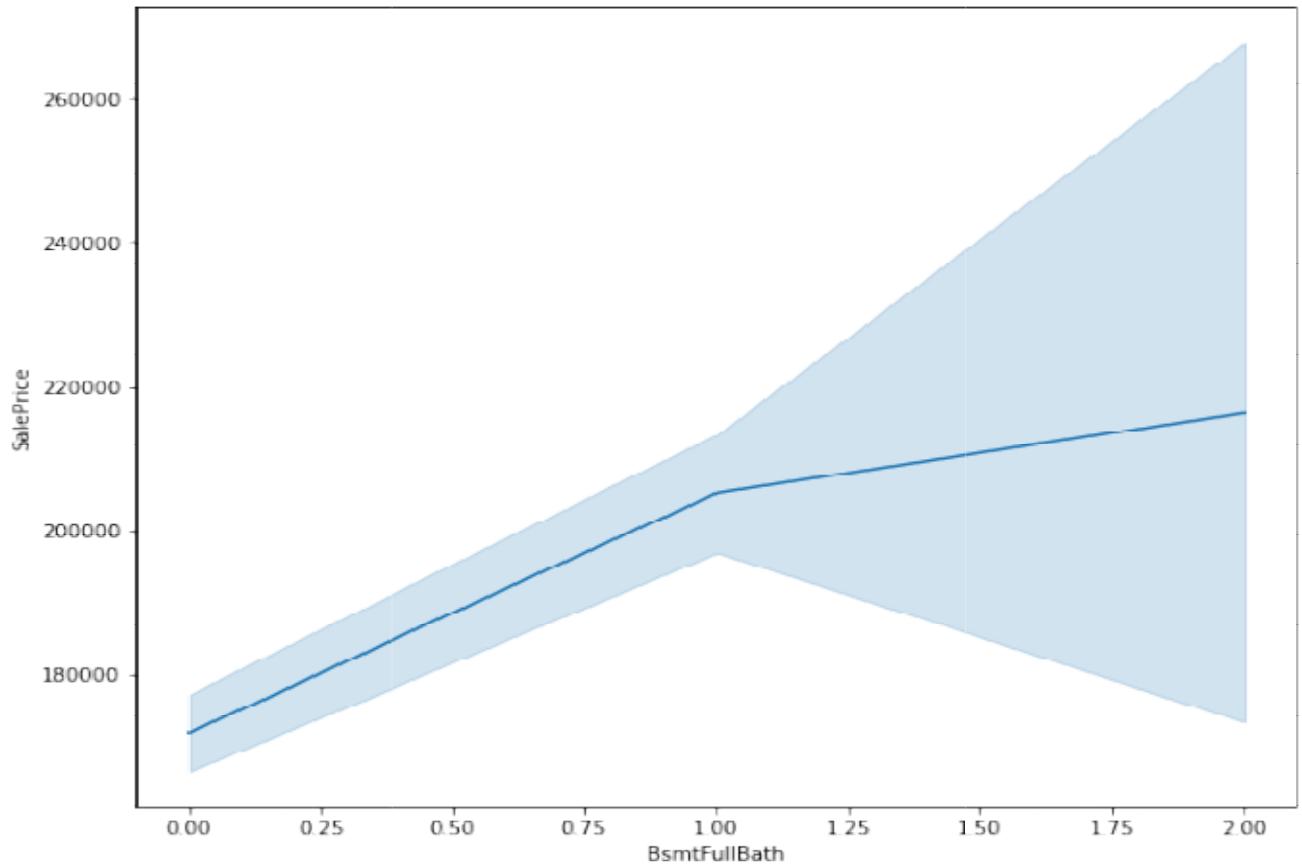


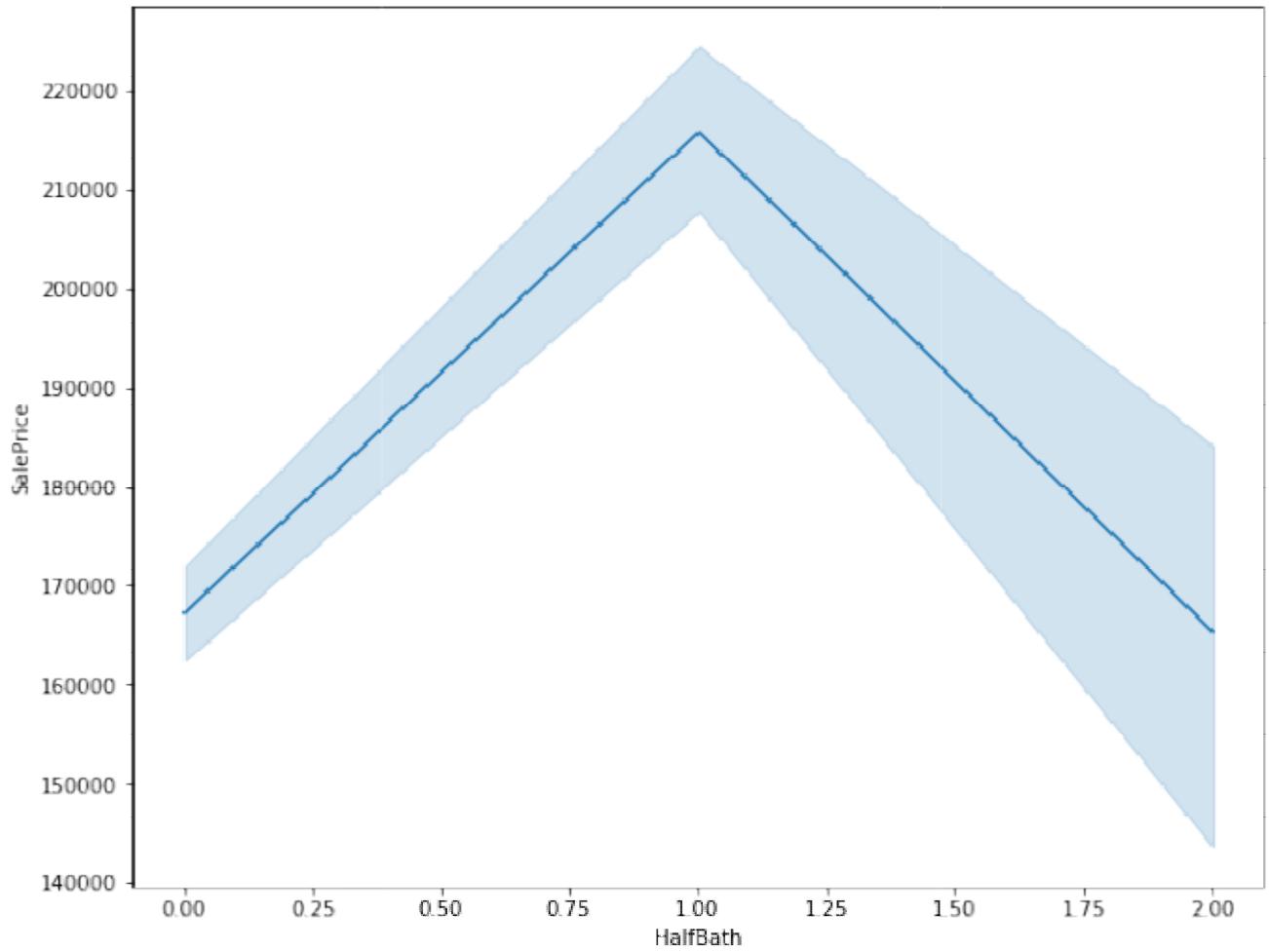
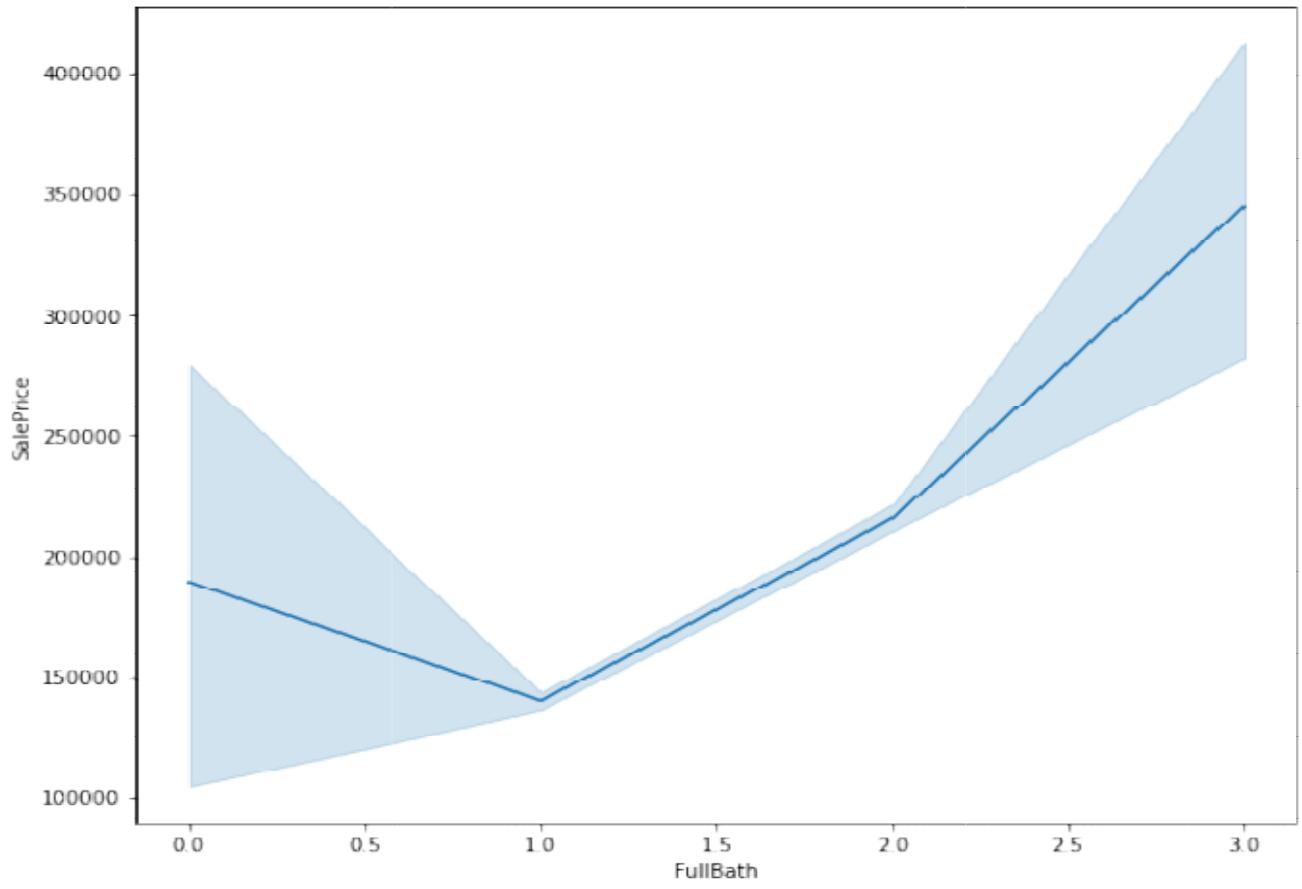


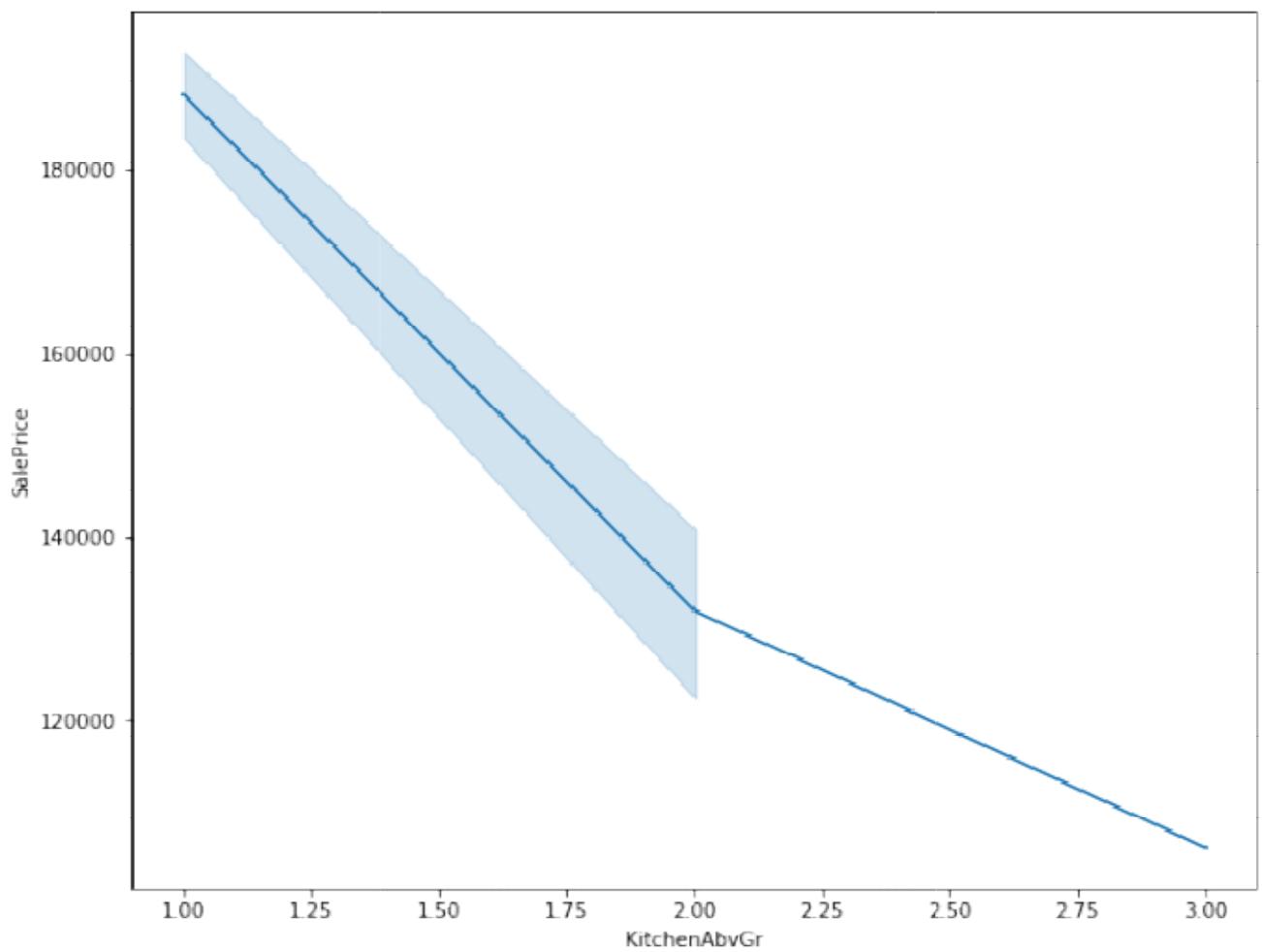
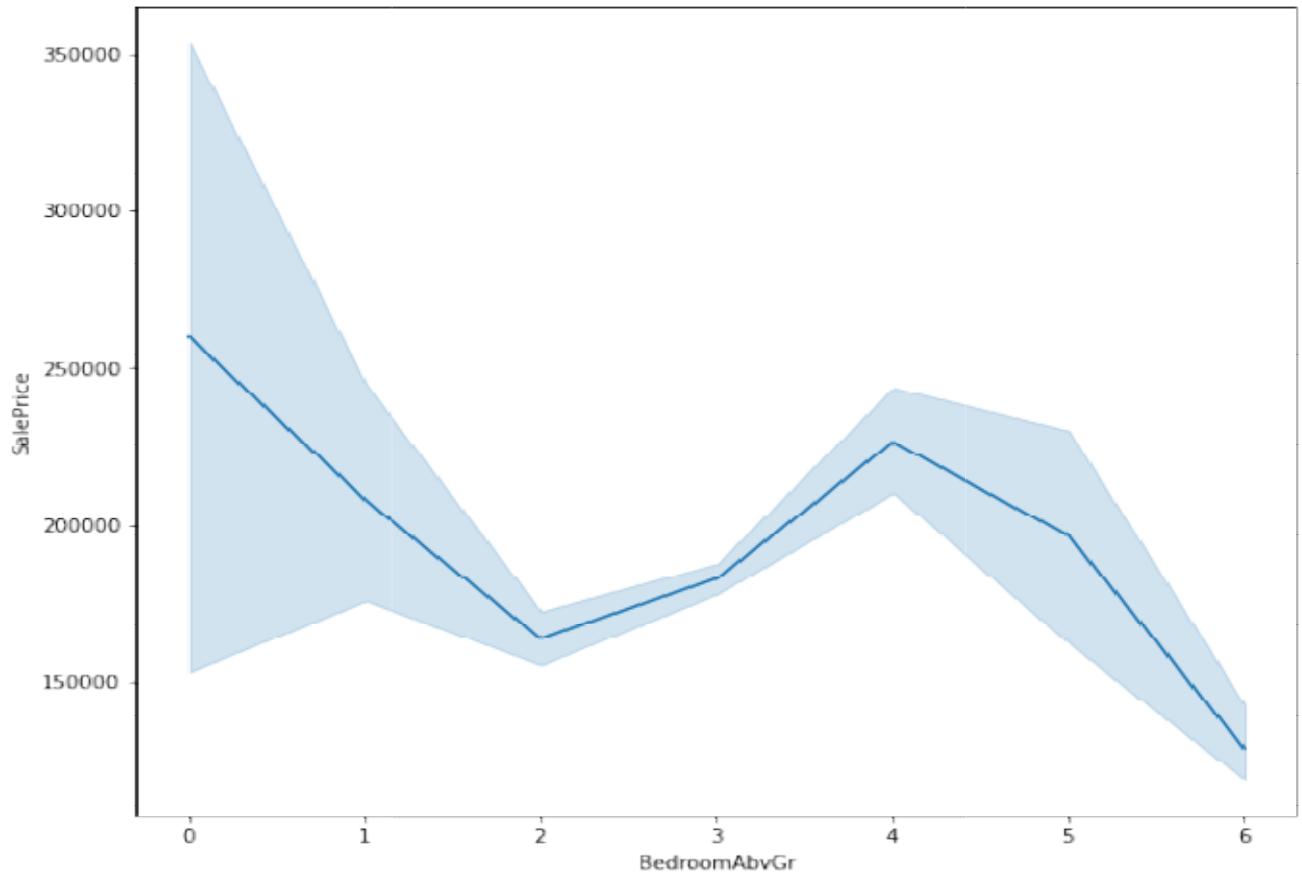


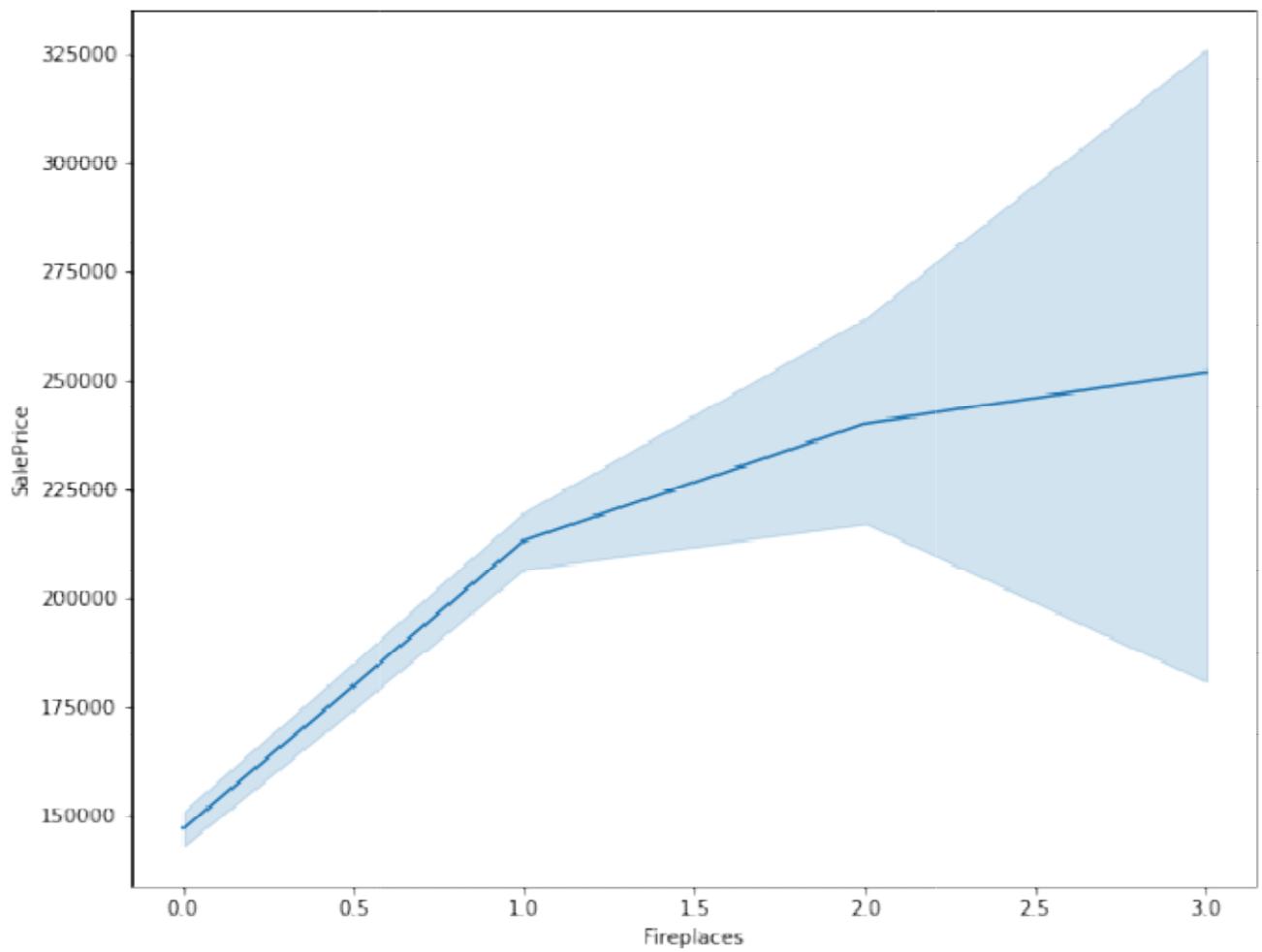
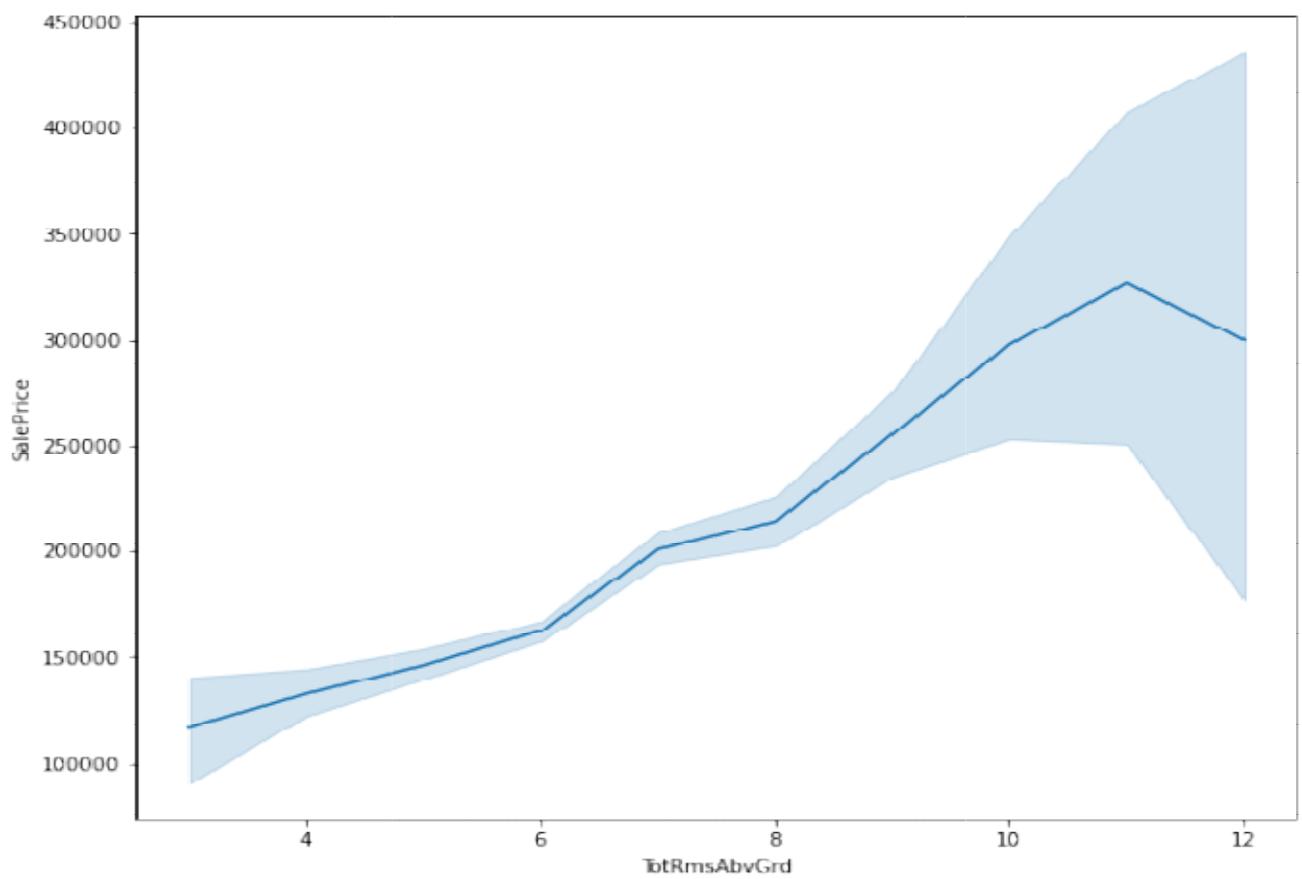


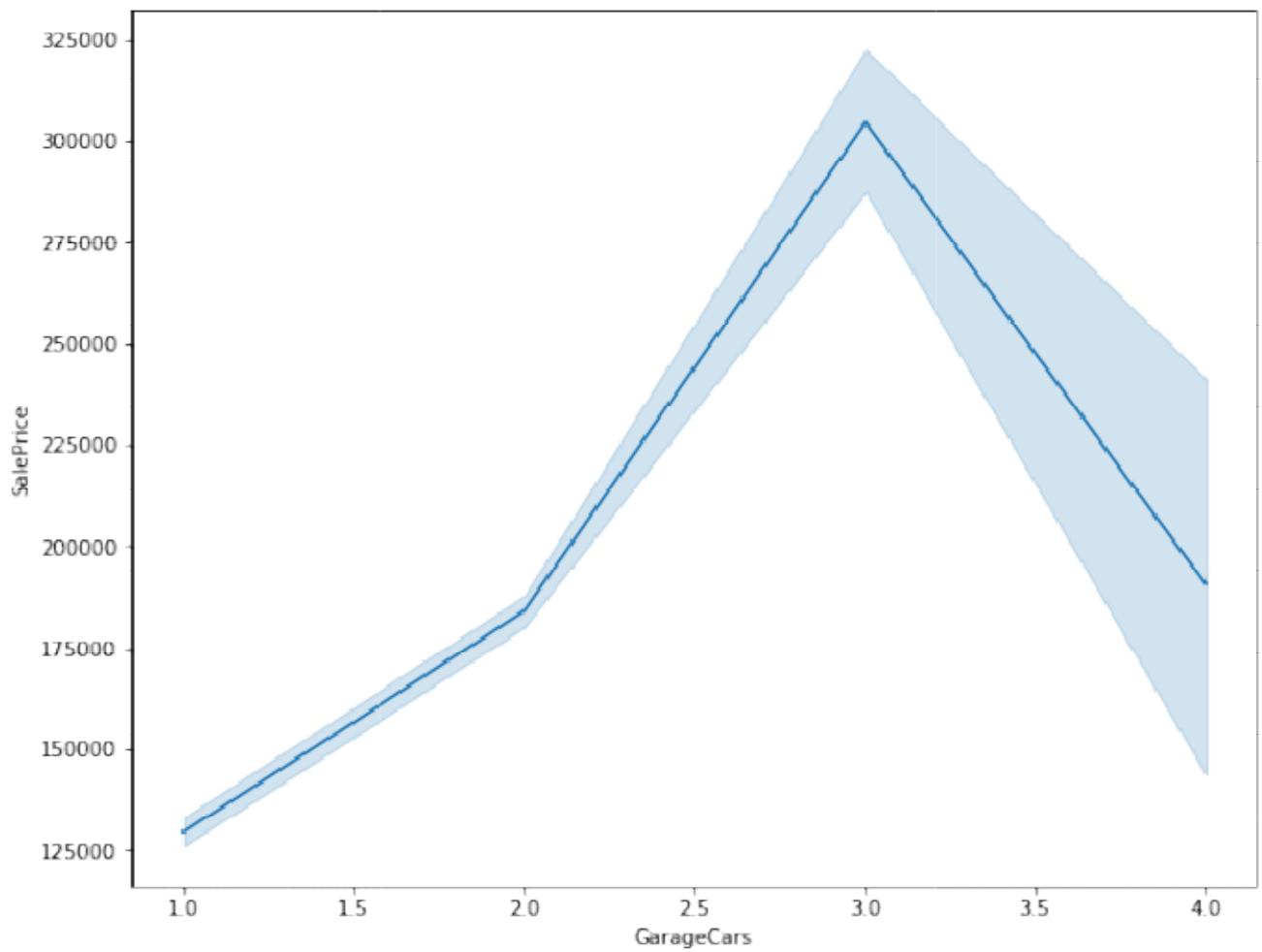
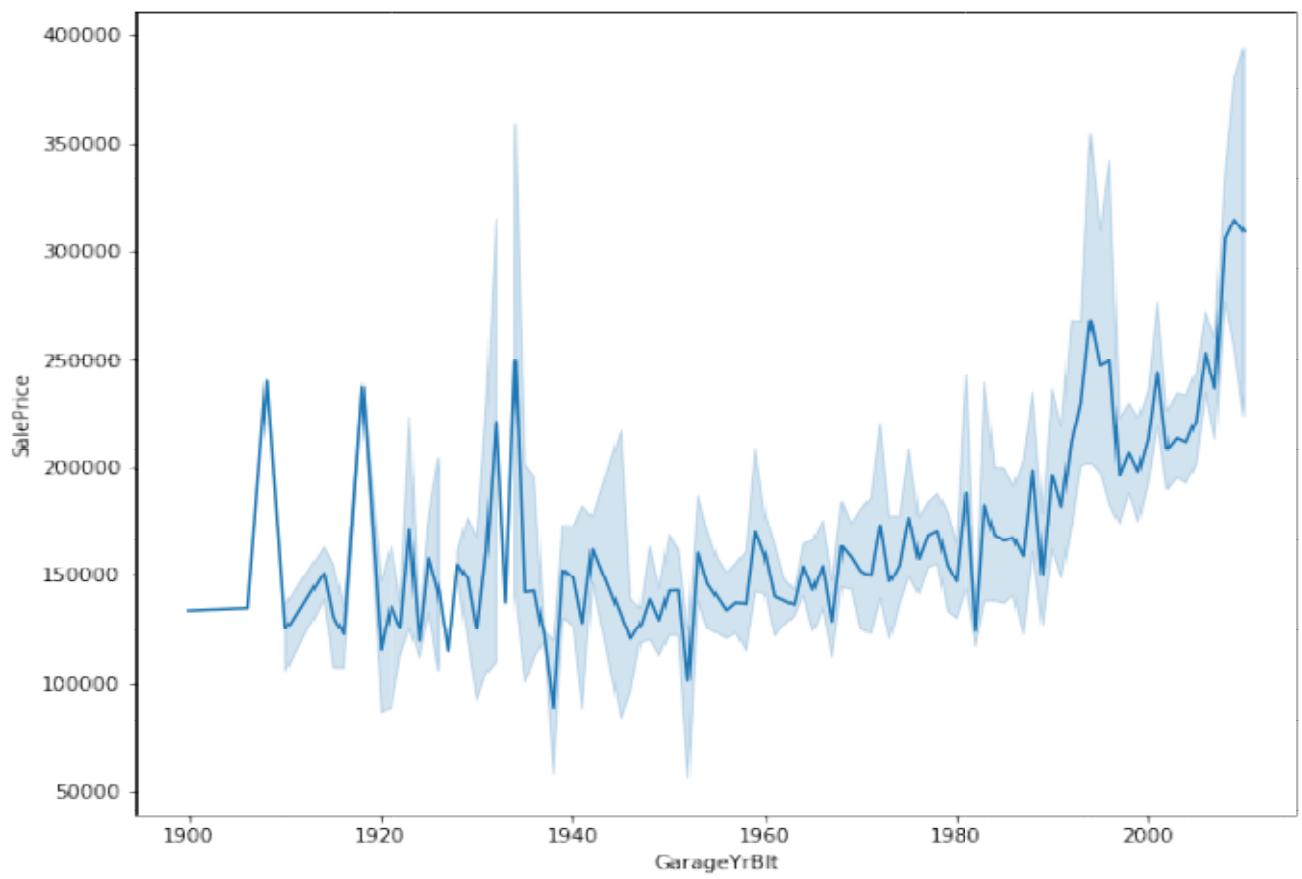


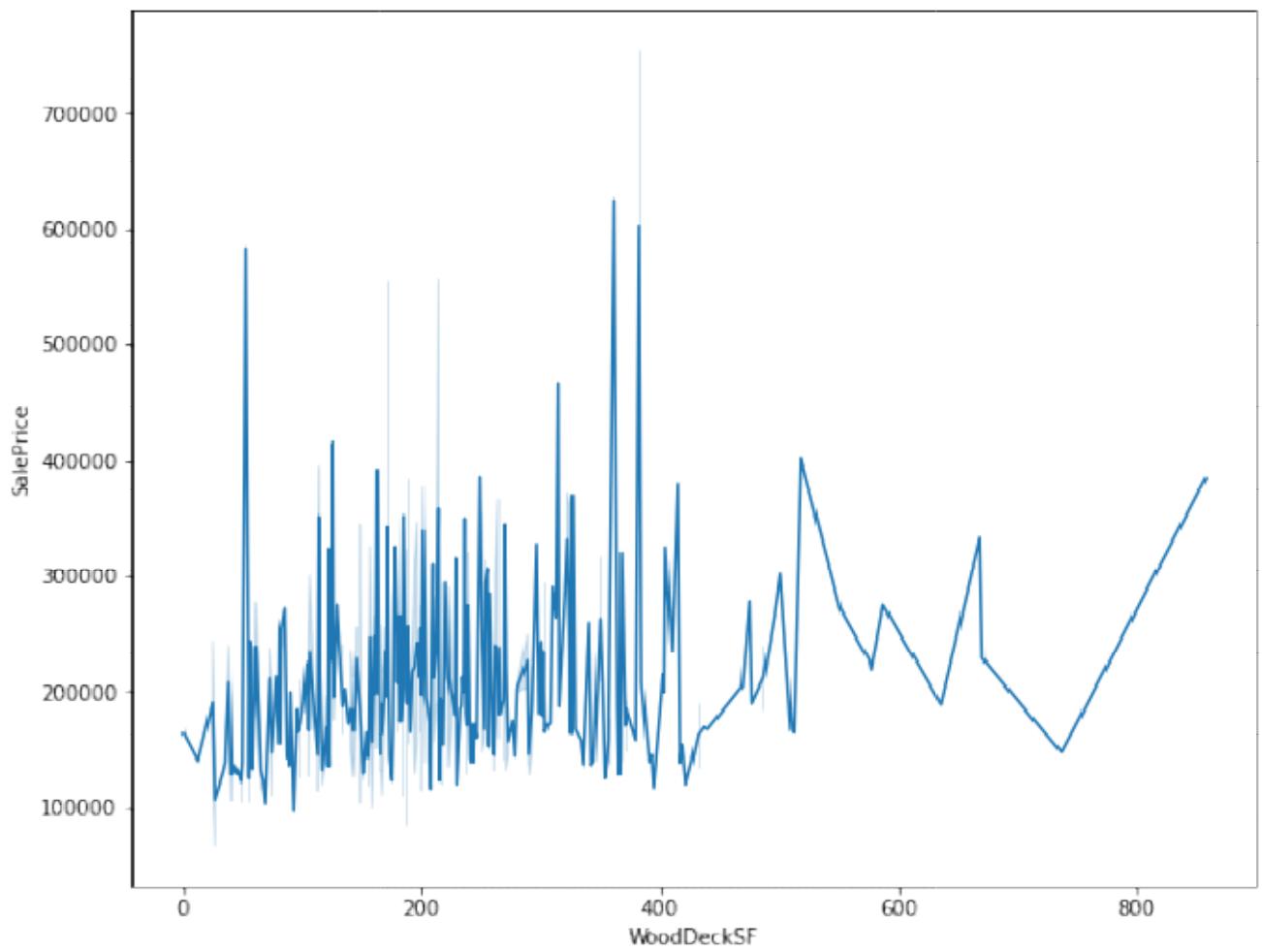
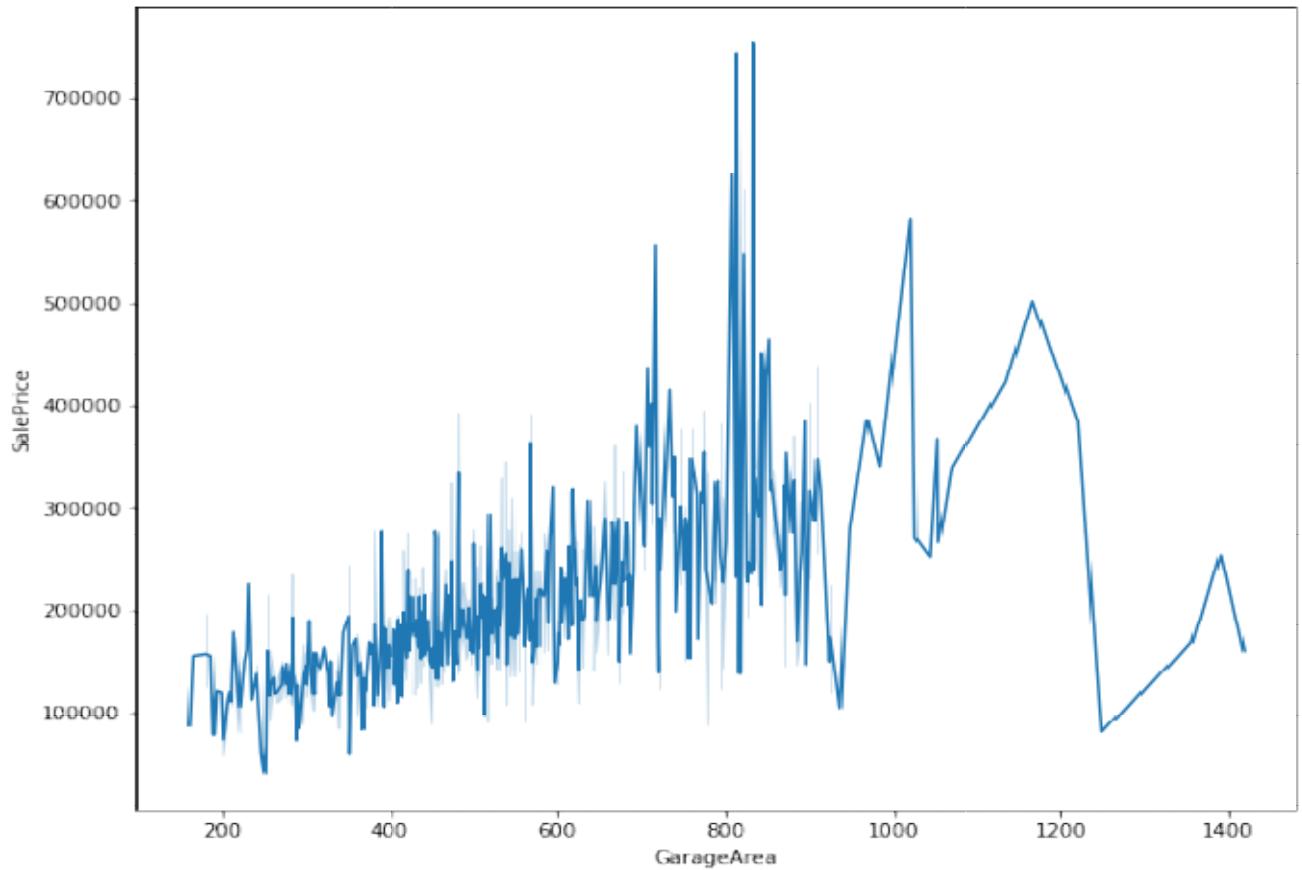


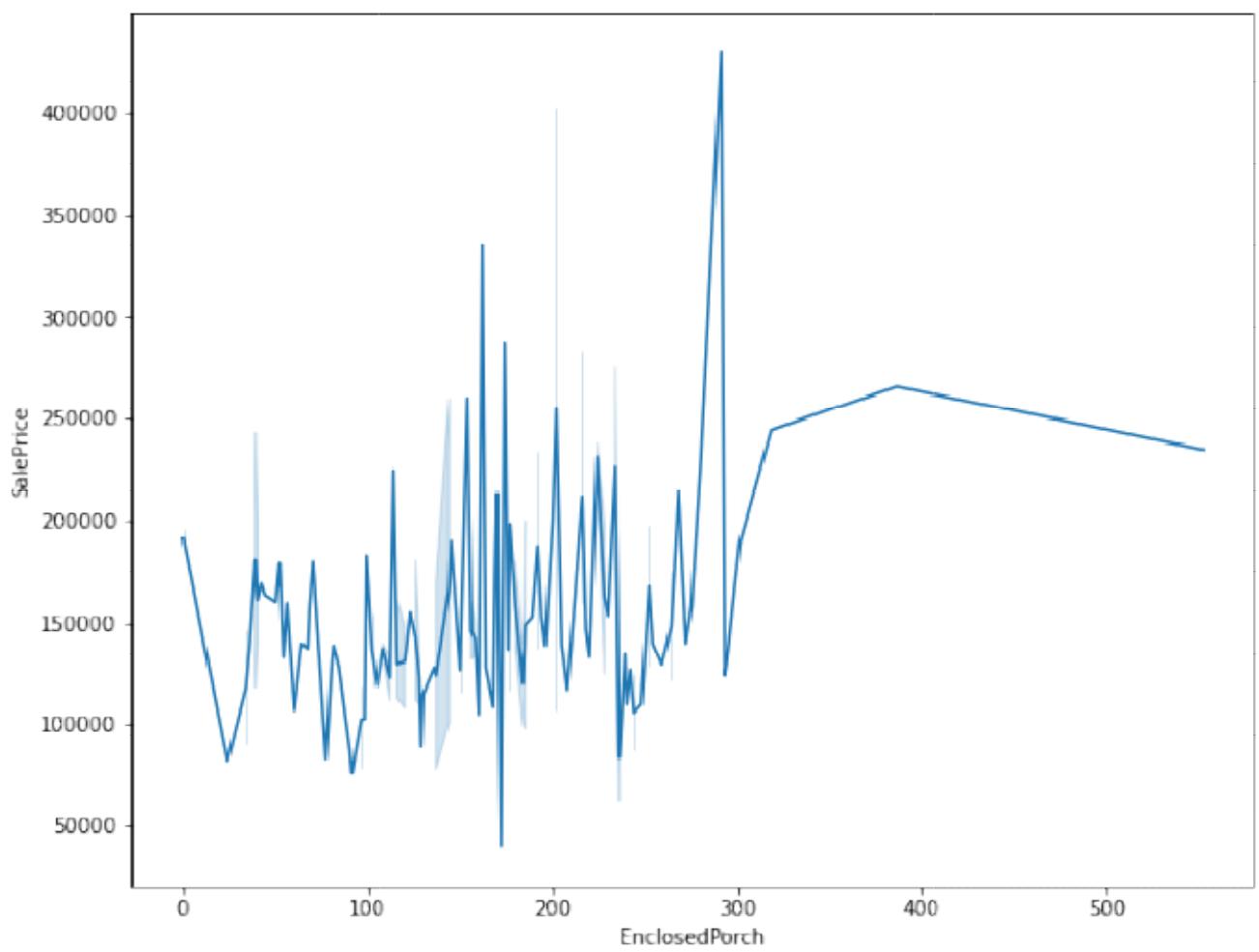
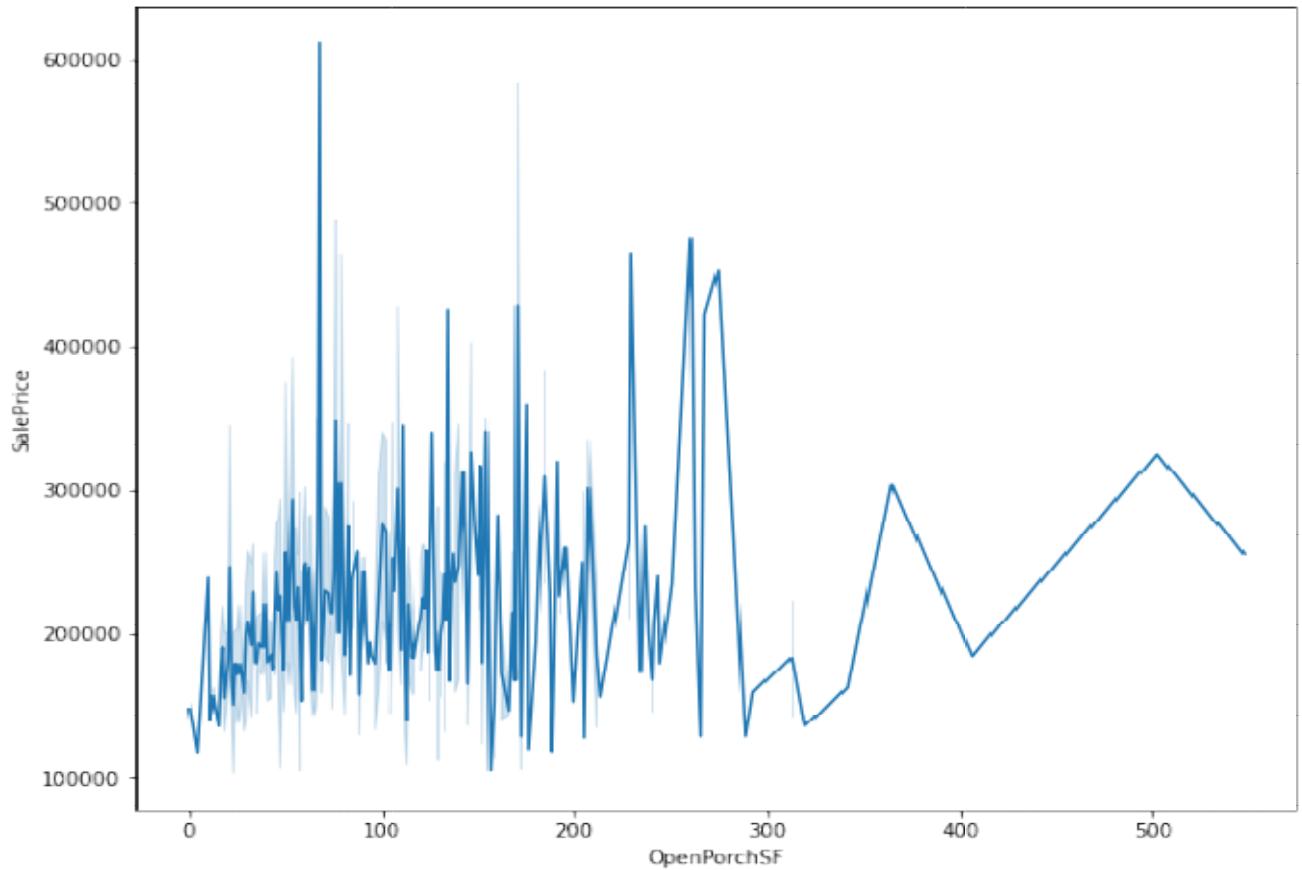


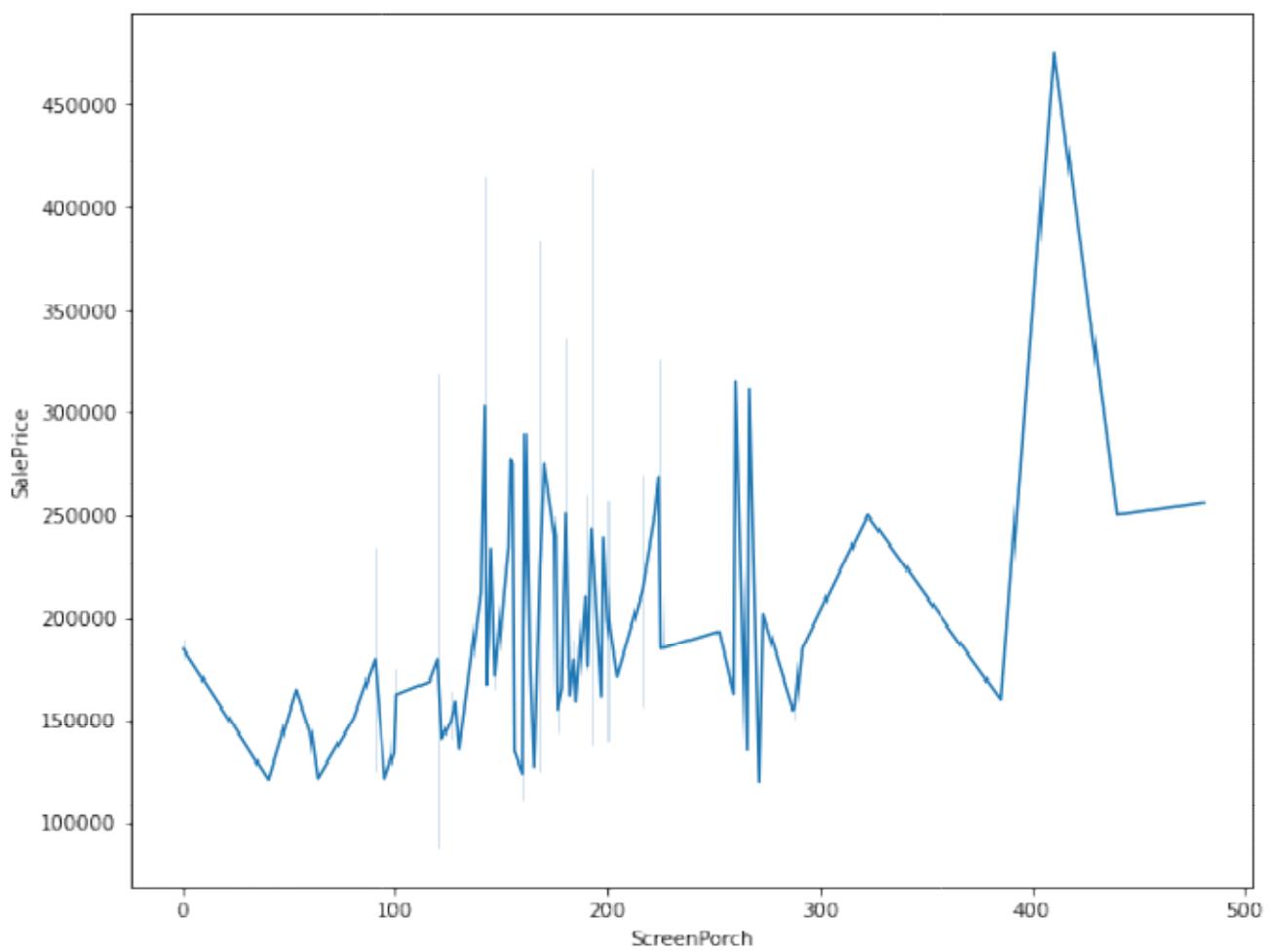
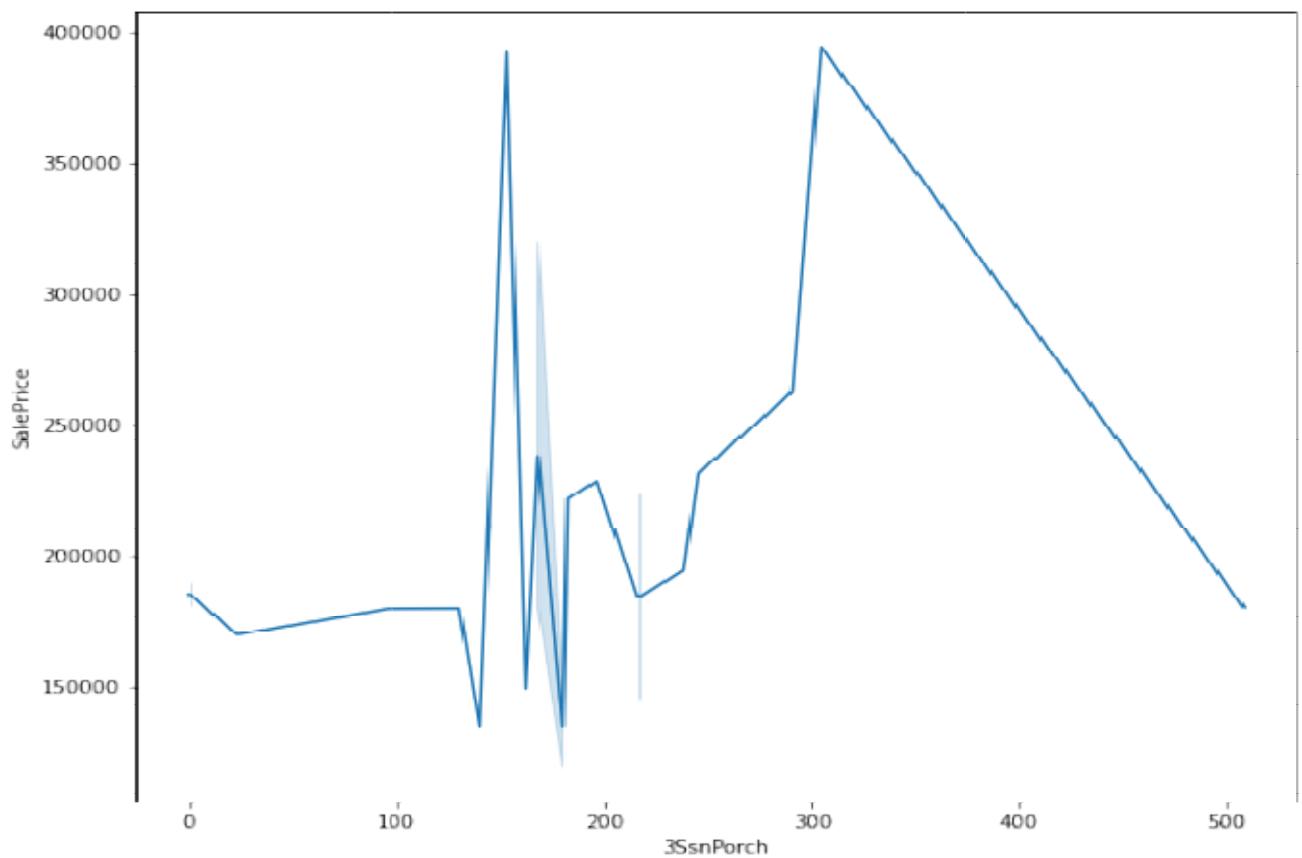


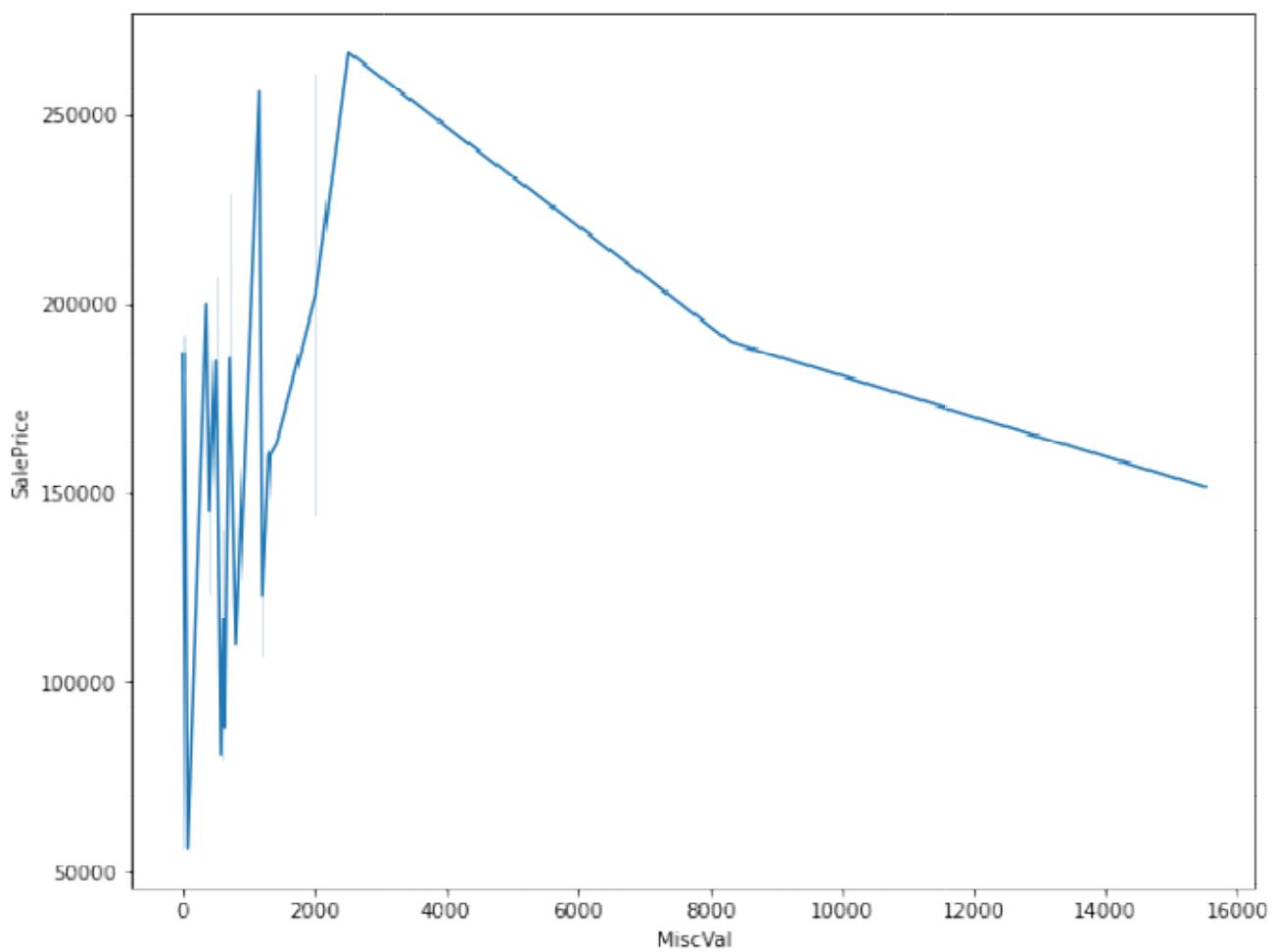
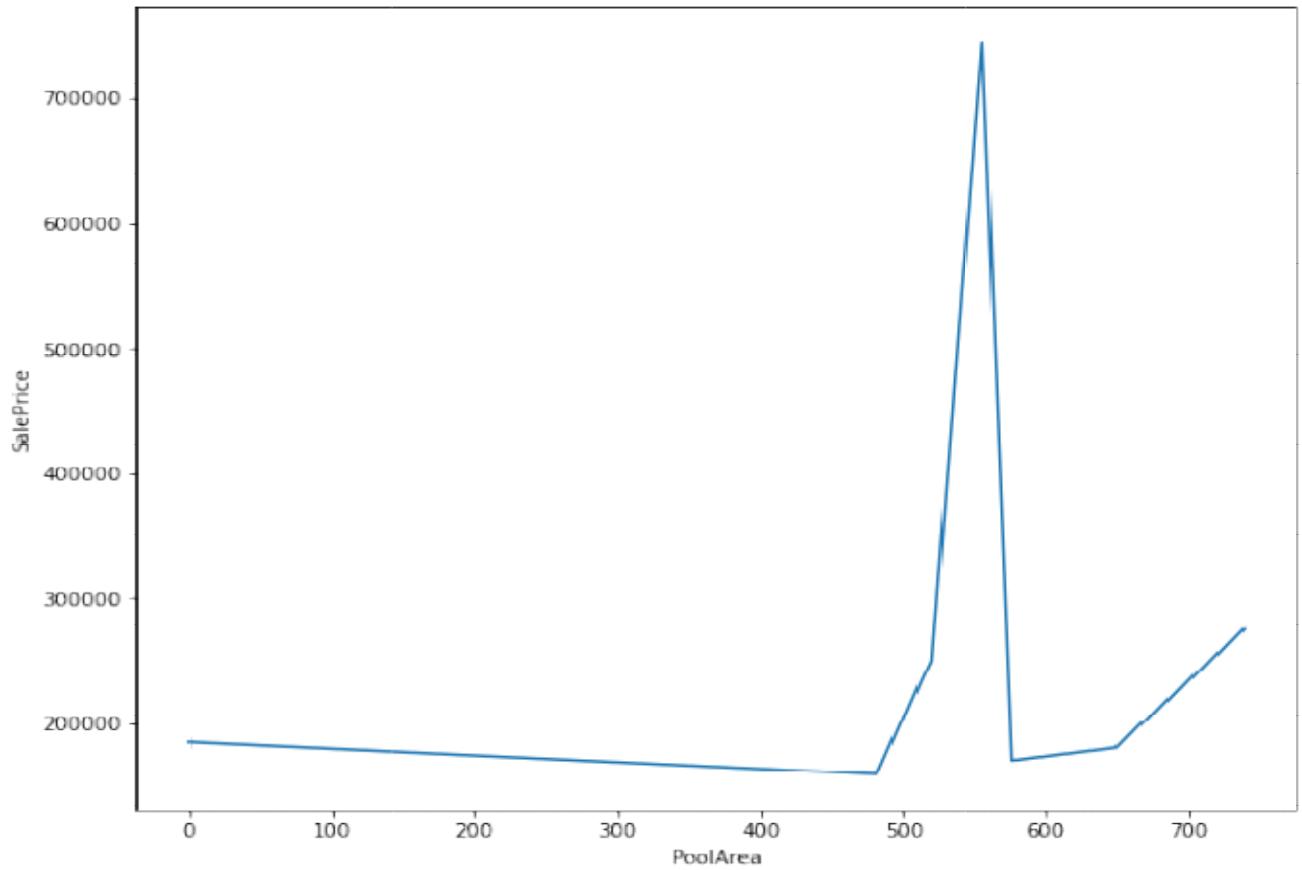


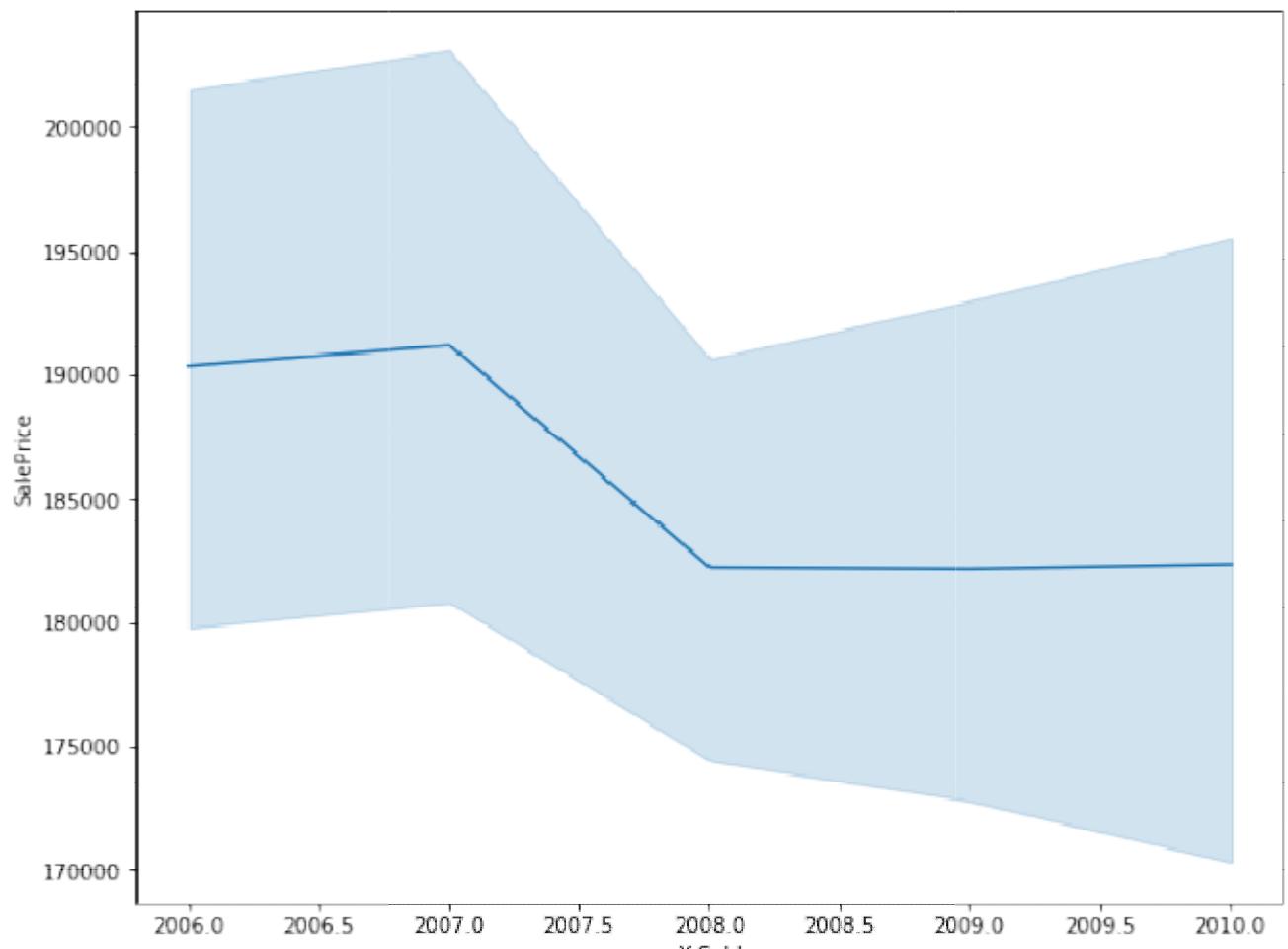
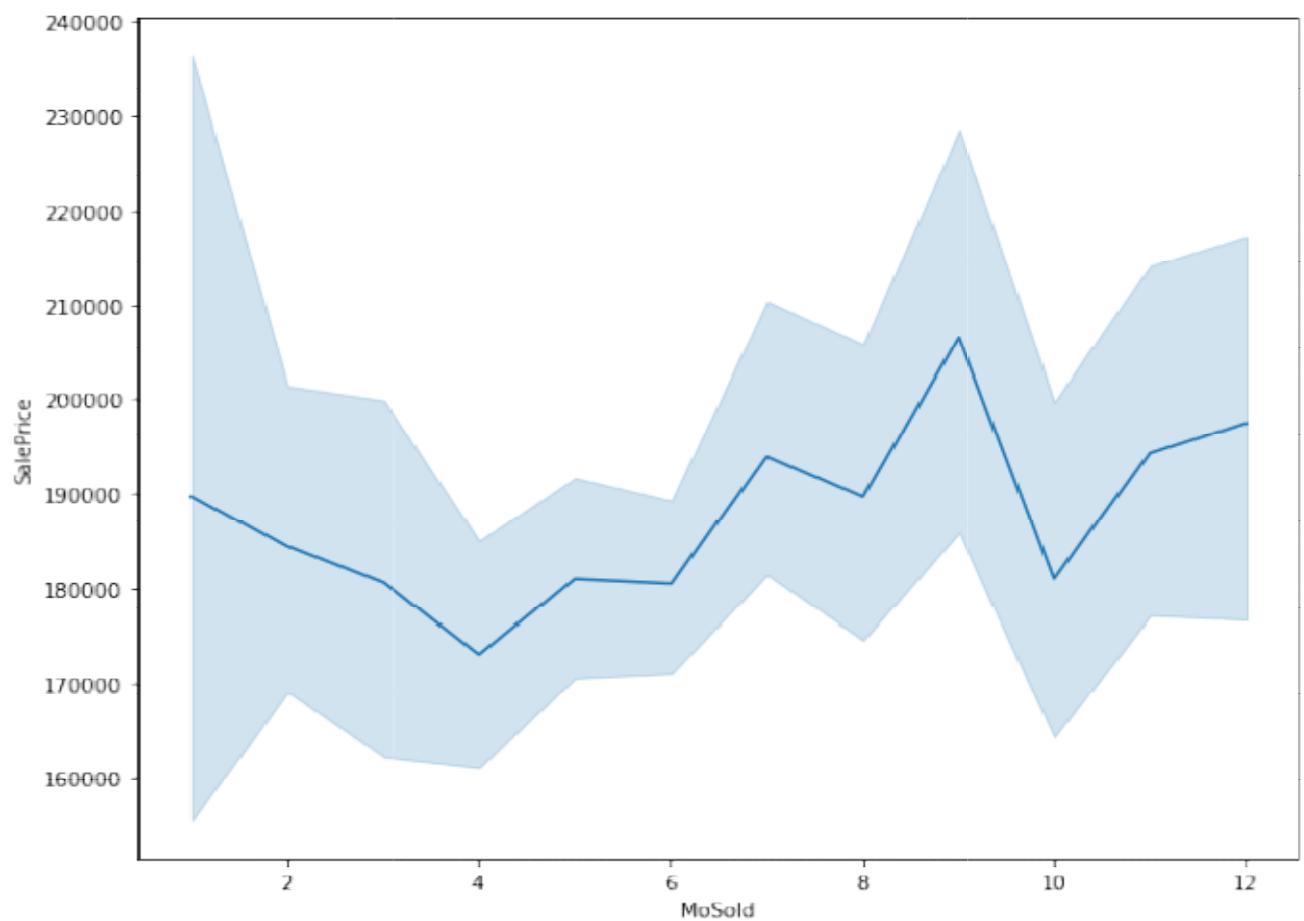












Observation on BI-VARIATE ANALYSIS :

- 1.2-STORY 1946 & NEWER, 2-STORY 1945 & OLDER is known to be the popular houses selling at higher price followed by 1-STORY PUD (Planned Unit Development) - 1946 & NEWER.
- 2.Lots between 150-200 linear feet seems to have higher sales price.
- 3.Price of the houses increases with the overall material and finish of the house.
- 4.Houses built between 1880-1900 and 1980-2006 seems to have higher sales price.
- 5.Houses with garage of 3 car capacity seems to have higher sales price and are selling more in numbers.
- 6.Most expected garage area is 600-900 sqft.
- 7.Houses with openporch sells quickly.
- 8.Also enclosed Porch seems to have higher sales price.

Feature Engineering:

1. During feature engineering the categorical datas are turned into numerical values using Label Encoding.
2. While implementing the z-score we found that the data loss is more than 60%. So we continued the data processing without implementing z-score.

Machine Learning Models:

For the above data-set we will use

- 1.LINEAR REGRESSION
- 2.RANDOM FOREST REGRESSOR
- 3.SGD REGRESSOR
- 4.LASSO
- 5.RIDGE

6.DecisionTreeRegressor

7.KNeighborsRegressor

8.AdaBoostRegressor

9.GradientBoostingRegressor

```
#Linear Regression
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
pred=lr.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(lr,x,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=92.06181212557935,
cross_value_score=1.6779585864399245e+18
Mean_Squared_Error = 278691916.7122572

```
#RANDOM FOREST REGRESSOR
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
pred=rf.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(rf,x,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=92.2257254871677,
cross_value_score=87.17095890037335
Mean_Squared_Error = 272937287.3637885

```
#SGD REGRESSOR
from sklearn.linear_model import SGDRegressor
SGD=SGDRegressor()
SGD.fit(x_train,y_train)
pred=SGD.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(SGD,x,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=91.93737670042663,
cross_value_score=85.54106101830634
Mean_Squared_Error = 283060564.53104573

```
#LASSO
from sklearn.linear_model import Lasso
la=Lasso()
la.fit(x_train,y_train)
pred=la.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(la,x,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=92.00380150355615,
cross_value_score=88.51205615469793
Mean_Squared_Error = 277919919.1238142

```
#RIDGE
from sklearn.linear_model import Ridge
Ri=Ridge()
Ri.fit(x_train,y_train)
pred=Ri.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(Ri,x,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=92.12978875507656,
cross_value_score=88.56042590959768
Mean_Squared_Error = 276305410.18634605

```
#DecisionTreeRegressor
from sklearn.tree import DecisionTreeRegressor
DTR=DecisionTreeRegressor(random_state=3)
DTR.fit(x_train,y_train)
pred=DTR.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(DTR,X,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)

Accuracy=80.54369299245303,
cross_value_score=69.39586767481046
Mean_Squared_Error = 683067165.6875
```

```
#KNeighborsRegressor
from sklearn.neighbors import KNeighborsRegressor
KNN=KNeighborsRegressor()
KNN.fit(x_train,y_train)
pred=KNN.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(KNN,X,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=82.52188948209538,
cross_value_score=80.33959967354805
Mean_Squared_Error = 613617137.5383333

```
#AdaBoostRegressor
from sklearn.ensemble import AdaBoostRegressor
AB=AdaBoostRegressor(random_state=3)
AB.fit(x_train,y_train)
pred=AB.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(AB,X,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=88.91221295913786,
cross_value_score=83.11447209192357
Mean_Squared_Error = 389267257.3891082

```
#GradientBoostingRegressor
from sklearn.ensemble import GradientBoostingRegressor
GB=GradientBoostingRegressor(random_state=3)
GB.fit(x_train,y_train)
pred=GB.predict(x_test)
r2score=r2_score(y_test,pred)
cvscore=abs(cross_val_score(GB,X,y, cv=10,scoring='r2')).mean()
print( f"Accuracy={r2score*100},\n cross_value_score={cvscore*100}")
mse=mean_squared_error(y_test,pred)
print("Mean_Squared_Error =",mse)
```

Accuracy=91.44699020812654,
cross_value_score=88.24971351814632
Mean_Squared_Error = 300276930.9903603

From the above results we could clearly see that the RIDGE REGRESSOR is having a good r2 Score and Cross-validation score.

So, We will Choose Ridge Regressor as our final Model.

HYPER PARAMETER TUNING - Ridge

```
#using GridSearch CV
from sklearn.model_selection import GridSearchCV,KFold
folds = KFold(n_splits=10,shuffle=True,random_state=42)
params = {'alpha':[0.001,0.01,0.1,0.2,0.5,0.9,1.0, 5.0, 10.0,20.0]}
Model_cv = GridSearchCV(ridge,param_grid=params,scoring='r2',cv=folds,verbose=1,return_train_score=True)
Model_cv.fit(x_train,y_train)
```

Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed: 0.1s finished

```
GridSearchCV(cv=KFold(n_splits=10, random_state=42, shuffle=True),
            estimator=Ridge(),
            param_grid={'alpha': [0.001, 0.01, 0.1, 0.2, 0.5, 0.9, 1.0, 5.0,
                                 10.0, 20.0]},
            return_train_score=True, scoring='r2', verbose=1)
```

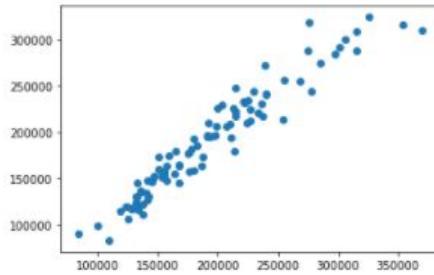
```
Model_cv.best_params_
{'alpha': 20.0}
```

```
ridge = Ridge(alpha=20)
ridge.fit(x_train,y_train)

y_train_pred = ridge.predict(x_train)
print("Train r2 score=",r2_score(y_train,y_train_pred))
y_test_pred = ridge.predict(x_test)
print("Test r2 score =",r2_score(y_test,y_test_pred))
```

```
Train r2 score= 0.9079598067875739
Test r2 score = 0.9233240226721764
```

```
plt.scatter(y_test,y_test_pred)
<matplotlib.collections.PathCollection at 0x2befb9092b0>
```



We could see that the scatter is a straight line which implies that the model is working properly.

SAVING THE FINAL MODEL :

```
import pickle
filename="Housing_Price_Prediction.pkl"
pickle.dump(ridge,open(filename,"wb"))
```

PREDICTION ON THE TEST DATA-SET

we will now do all the Data pre-processing done for the Training dataset, on the Test Data-set.

```
fitted_model=pickle.load(open("Housing_Price_Prediction.pkl","rb"))
```

```
fitted_model
```

```
Ridge(alpha=20)
```

```
predictions=fitted_model.predict(dftc_new)
```

```
In [142]: predictions
217124.16808233, 151807.29001352, 178289.12404147, 285548.18570866,
165515.93222087, 288615.94592682, 280286.58584415, 158175.18476662,
238463.01561598, 293672.46425804, 184799.2624431, 256841.72229453,
146065.28034167, 281092.1288671, 150259.18441568, 188097.86155351,
184173.99922088, 190029.93799381, 237600.98530025, 105803.38286014,
330005.01843919, 182665.89541844, 178128.00865813, 246251.36478584,
137425.98216462, 161198.46983041, 125820.797080559, 206542.79779216,
172261.09832485, 238232.98976508, 173638.91318432, 341941.79172481,
128941.2191622, 240754.07897117, 95335.76435771, 123175.11018979,
281186.25857274, 171794.41382737, 261175.16548898, 184399.28806667,
193268.04067689, 217087.57152145, 225531.6886756, 184149.06068472,
232578.09019144, 258884.51499826, 150217.51812667, 186394.0695125,
146051.29815784, 286691.89050587, 151306.99798776, 99654.42613136,
117434.69978842, 192321.00312903, 273669.56851395, 152646.11991286,
197890.68325888, 133442.64040031, 187350.93877995, 79051.47603982,
170229.43042477, 238236.85716875, 174828.31635233, 164561.91941869,
200639.16040676, 277197.13598848, 211698.1741203, 138833.60857886,
267533.11003886, 133429.11789057, 131886.84678713, 400063.46211989,
81845.00874993, 348488.4273413, 281865.71211852, 238834.6890847,
156693.04286153, 140913.4640047, 182120.7873465, 203773.0168293,
```

```
In [143]: df_pred=pd.DataFrame({"Sales Price" : predictions })
df_pred
```

```
Out[143]:
Sales Price
0    302016.483261
1    206584.357376
2    244227.463463
3    190689.427366
4    224246.056919
5    141388.731672
6    282741.839999
7    250552.524970
8    179418.577759
9    85453.020010
10   147133.246392
```

CONCLUSION REMARKS:

1. we could see that using "RIDGE regression" gives a most promising predictions when compared with all other Models.
2. Also many Factors influence the sale of a House, but year built, plot area, garage type, road access, availability of some basic features seems to have a higher hand in deciding the sales price for a house.

END