FLIP ROBO

Housing Project

Submitted by:

Kshitij Chawla

ACKNOWLEDGMENT

In this project different libraries and methods are used that are available in python which helped in completion of the project:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

http://scikit-learn.org/stable/modules/model_evaluation.html

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

https://seaborn.pydata.org/generated/seaborn.countplot.html

https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

https://www.analyticsvidhya.com/blog/2020/10/how-to-choose-evaluation-metrics-for-classification-model/

https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

INTRODUCTION

- Business Problem Framing

Houses are one of the necessary need 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.

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

below.

The company is looking at prospective properties to buy houses to enter the market. You 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. For this company wants to know:

•Which variables are important to predict the price of variable?

•How do these variables describe the price of the house?

- ## Conceptual Background of the Domain Problem

The project will require knowledge and practice in building Graphs /plots and analysing them to get the relationship between dataset, Knowledge of Different Learning Models to build and predict the required output. Basic Data science concepts to increase the quality of the dataset and Python Knowledge (Coding Language) which will be used to solve the complete Micro Credit Defaulter project. Understanding of calculating F2 score, accuracy, skewness and basic mathematics/statistical approaches will help to build an accurate model for this project.

- ## Review of Literature

Market price is what a willing, ready and bank-qualified buyer will pay for a property and what the seller will accept for it. The transaction that takes place determines the market price, which will then influence the market value of future sales. Price is determined by local supply and demand, the property’s condition and what other similar properties have sold for without adding in the value component.

Market value is an opinion of what a property would sell for in a competitive market based on the features and benefits of that property (the value), the overall real estate market, supply and demand, and what other similar properties have sold for in the same condition.

The major difference between market value and market price is that the market value, in the eyes of the seller, might be much more than what a buyer will pay for the property or it’s true market price. Value can create demand, which can influence price. But, without the demand function, value alone cannot influence price. As supply increases and demand decreases, price goes down, and value is not influential. As supply decreases and demand increases, the price will rise, and value will influence price. Market value and market price can be equal in a balanced market.

However, buyers and sellers can view value differently. A seller might feel that their in-ground pool is a benefit, but the buyer could see it as a negative and place less value on the property. Or the seller could feel the new roof they put on the house has great value; however, the buyer places no value on this because they expect the property to have a roof in good condition. Or a builder might feel he has superior quality and demand a higher price, but the buyer places less value on quality and more value on the lot, neighborhood and floor plan of the property.

- Motivation for the Problem Undertaken

  I wanted to solve the real-life problem using the Technical skills gathered during the course of being a Data Analyst and improving the skill set.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem----

  Regression Models->

  Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.

  Decision Tree –

  It is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility.

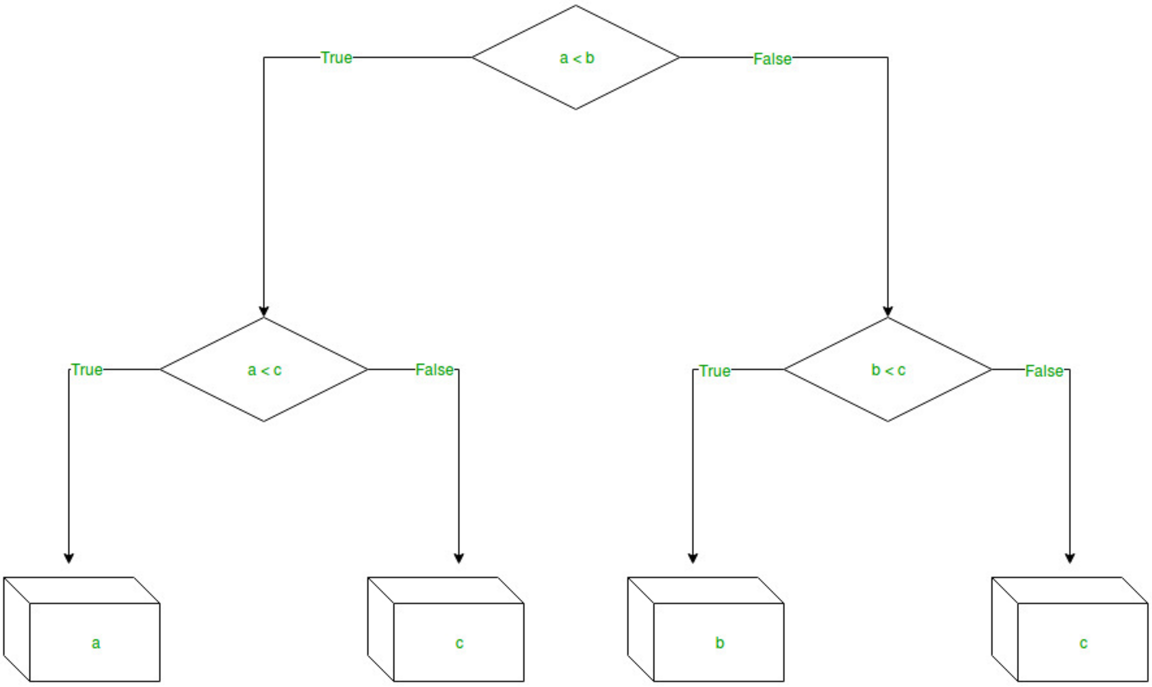
  Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

  The branches/edges represent the result of the node and the nodes have either:

  Conditions [Decision Nodes]

  Result [End Nodes]

  The branches/edges represent the truth/falsity of the statement and takes makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:



Random Forest –

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Naive Bayes –

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Linear Regression –

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression.

SVM –

Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line.

We used different Plots/ graphs to perform EDA on the dataset->

1) Box Plot: It is a type of chart that depicts a group of numerical data through their quartiles. It is a simple way to visualize the shape of our data. It makes comparing characteristics of data between categories very easy.
2) Count Plot: IT is kind of like a histogram or a bar graph for some categorical area. It simply shows the number of occurrences of an item based on a certain type of category
3) Heat Map: It contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.

4) Scatter Plot: A scatter plot is a diagram where each value in the data set is represented by a dot. The Matplotlib module has a method for drawing scatter plots

● Data Sources and their formats

Below are the fields present in our dataset with the information what these fields describe

MSSubClass: Identifies the type of dwelling involved in the sale.

20	1-STORY 1946 & NEWER ALL STYLES
30	1-STORY 1945 & OLDER
40	1-STORY W/FINISHED ATTIC ALL AGES
45	1-1/2 STORY - UNFINISHED ALL AGES
50	1-1/2 STORY FINISHED ALL AGES
60	2-STORY 1946 & NEWER
70	2-STORY 1945 & OLDER
75	2-1/2 STORY ALL AGES
80	SPLIT OR MULTI-LEVEL
85	SPLIT FOYER
90	DUPLEX - ALL STYLES AND AGES
120	1-STORY PUD (Planned Unit Development) - 1946 & NEWER
150	1-1/2 STORY PUD - ALL AGES
160	2-STORY PUD - 1946 & NEWER
180	PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
190	2 FAMILY CONVERSION - ALL STYLES AND AGES

MSZoning: Identifies the general zoning classification of the sale.

A	Agriculture
C	Commercial
FV	Floating Village Residential
I	Industrial
RH	Residential High Density
RL	Residential Low Density
RP	Residential Low Density Park
RM	Residential Medium Density

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

Grvl	Gravel
Pave	Paved

Alley: Type of alley access to property

Grvl	Gravel
Pave	Paved
NA	No alley access

LotShape: General shape of property

Reg	Regular
IR1	Slightly irregular
IR2	Moderately Irregular
IR3	Irregular

LandContour: Flatness of the property

Lvl	Near Flat/Level
Bnk	Banked - Quick and significant rise from street grade to building

HLS	Hillside - Significant slope from side to side
Low	Depression

Utilities: Type of utilities available

AllPub	All public Utilities (E,G,W,& S)
NoSewr	Electricity, Gas, and Water (Septic Tank)
NoSeWa	Electricity and Gas Only
ELO	Electricity only

LotConfig: Lot configuration

Inside	Inside lot
Corner	Corner lot
CulDSac	Cul-de-sac
FR2	Frontage on 2 sides of property
FR3	Frontage on 3 sides of property

LandSlope: Slope of property

Gtl	Gentle slope
Mod	Moderate Slope
Sev	Severe Slope

Neighborhood: Physical locations within Ames city limits

Blmngtn	Bloomington Heights
Blueste	Bluestem
BrDale	Briardale
BrkSide	Brookside
ClearCr	Clear Creek
CollgCr	College Creek
Crawfor	Crawford

Edwards	Edwards
Gilbert	Gilbert
IDOTRR	Iowa DOT and Rail Road
MeadowV	Meadow Village
Mitchel	Mitchell
Names	North Ames
NoRidge	Northridge
NPkVill	Northpark Villa
NridgHt	Northridge Heights
NWAmes	Northwest Ames
OldTown	Old Town
SWISU	South & West of Iowa State University
Sawyer	Sawyer
SawyerW	Sawyer West
Somerst	Somerset
StoneBr	Stone Brook
Timber	Timberland
Veenker	Veenker

Condition1: Proximity to various conditions

Artery	Adjacent to arterial street
Feedr	Adjacent to feeder street
Norm	Normal
RRNn	Within 200' of North-South Railroad
RRAn	Adjacent to North-South Railroad
PosN	Near positive off-site feature--park, greenbelt, etc.

PosA Adjacent to postive off-site feature

RRNe Within 200' of East-West Railroad

RRAe Adjacent to East-West Railroad

Condition2: Proximity to various conditions (if more than one is present)

Artery Adjacent to arterial street

Feedr Adjacent to feeder street

Norm Normal

RRNn Within 200' of North-South Railroad

RRAn Adjacent to North-South Railroad

PosN Near positive off-site feature--park, greenbelt, etc.

PosA Adjacent to postive off-site feature

RRNe Within 200' of East-West Railroad

RRAe Adjacent to East-West Railroad

BldgType: Type of dwelling

1Fam Single-family Detached

2FmCon Two-family Conversion; originally built as one-family dwelling

Duplx Duplex

TwnhsE Townhouse End Unit

TwnhsITownhouse Inside Unit

HouseStyle: Style of dwelling

1Story One story

1.5Fin One and one-half story: 2nd level finished

1.5Unf One and one-half story: 2nd level unfinished

2Story Two story

2.5Fin	Two and one-half story: 2nd level finished
2.5Unf	Two and one-half story: 2nd level unfinished
SFoyer	Split Foyer
SLvl	Split Level

OverallQual: Rates the overall material and finish of the house

10	Very Excellent
9	Excellent
8	Very Good
7	Good
6	Above Average
5	Average
4	Below Average
3	Fair
2	Poor
1	Very Poor

OverallCond: Rates the overall condition of the house

10	Very Excellent
9	Excellent
8	Very Good
7	Good
6	Above Average
5	Average
4	Below Average
3	Fair
2	Poor

1	Very Poor

YearBuilt: Original construction date

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)

RoofStyle: Type of roof

Flat	Flat
Gable	Gable
Gambrel	Gabrel (Barn)
Hip	Hip
Mansard	Mansard
Shed	Shed

RoofMatl: Roof material

ClyTile	Clay or Tile
CompShg	Standard (Composite) Shingle
Membran	Membrane
Metal	Metal
Roll	Roll
Tar&Grv	Gravel & Tar
WdShake	Wood Shakes
WdShngl	Wood Shingles

Exterior1st: Exterior covering on house

AsbShng	Asbestos Shingles
AsphShn	Asphalt Shingles
BrkComm	Brick Common
BrkFace	Brick Face
CBlock	Cinder Block

CemntBd	Cement Board
HdBoard	Hard Board
ImStucc	Imitation Stucco
MetalSd	Metal Siding
Other	Other
Plywood	Plywood
PreCast	PreCast
Stone	Stone
Stucco	Stucco
VinylSd	Vinyl Siding
Wd Sdng	Wood Siding
WdShing	Wood Shingles

Exterior2nd: Exterior covering on house (if more than one material)

AsbShng	Asbestos Shingles
AsphShn	Asphalt Shingles
BrkComm	Brick Common
BrkFace	Brick Face
CBlock	Cinder Block
CemntBd	Cement Board
HdBoard	Hard Board
ImStucc	Imitation Stucco
MetalSd	Metal Siding
Other	Other
Plywood	Plywood
PreCast	PreCast

Stone	Stone
Stucco	Stucco
VinylSd	Vinyl Siding
Wd Sdng	Wood Siding
WdShing	Wood Shingles

MasVnrType: Masonry veneer type

BrkCmn	Brick Common
BrkFace	Brick Face
CBlock	Cinder Block
None	None
Stone	Stone

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

ExterCond: Evaluates the present condition of the material on the exterior

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

Foundation: Type of foundation

BrkTil	Brick & Tile
CBlock	Cinder Block
PConc	Poured Contrete
Slab	Slab
Stone	Stone
Wood	Wood

BsmtQual: Evaluates the height of the basement

Ex	Excellent (100+ inches)
Gd	Good (90-99 inches)
TA	Typical (80-89 inches)
Fa	Fair (70-79 inches)
Po	Poor (<70 inches
NA	No Basement

BsmtCond: Evaluates the general condition of the basement

Ex	Excellent
Gd	Good
TA	Typical - slight dampness allowed
Fa	Fair - dampness or some cracking or settling
Po	Poor - Severe cracking, settling, or wetness
NA	No Basement

BsmtExposure: Refers to walkout or garden level walls

Gd	Good Exposure
Av	Average Exposure (split levels or foyers typically score average or above)
Mn	Mimimum Exposure

No	No Exposure
NA	No Basement

BsmtFinType1: Rating of basement finished area

GLQ	Good Living Quarters
ALQ	Average Living Quarters
BLQ	Below Average Living Quarters
Rec	Average Rec Room
LwQ	Low Quality
Unf	Unfinshed
NA	No Basement

BsmtFinSF1: Type 1 finished square feet

BsmtFinType2: Rating of basement finished area (if multiple types)

GLQ	Good Living Quarters
ALQ	Average Living Quarters
BLQ	Below Average Living Quarters
Rec	Average Rec Room
LwQ	Low Quality
Unf	Unfinshed
NA	No Basement

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

Floor	Floor Furnace
GasA	Gas forced warm air furnace
GasW	Gas hot water or steam heat
Grav	Gravity furnace
OthW	Hot water or steam heat other than gas
Wall	Wall furnace

HeatingQC: Heating quality and condition

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

CentralAir: Central air conditioning

N	No
Y	Yes

Electrical: Electrical system

SBrkr	Standard Circuit Breakers & Romex
FuseA	Fuse Box over 60 AMP and all Romex wiring (Average)
FuseF	60 AMP Fuse Box and mostly Romex wiring (Fair)
FuseP	60 AMP Fuse Box and mostly knob & tube wiring (poor)
Mix	Mixed

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

Ex	Excellent
Gd	Good
TA	Typical/Average
Fa	Fair
Po	Poor

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

Functional: Home functionality (Assume typical unless deductions are warranted)

Typ	Typical Functionality
Min1	Minor Deductions 1
Min2	Minor Deductions 2
Mod	Moderate Deductions
Maj1	Major Deductions 1

Maj2 Major Deductions 2

Sev Severely Damaged

Sal Salvage only

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

Ex Excellent - Exceptional Masonry Fireplace

Gd Good - Masonry Fireplace in main level

TA Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement

Fa Fair - Prefabricated Fireplace in basement

Po Poor - Ben Franklin Stove

NA No Fireplace

GarageType: Garage location

2Types More than one type of garage

Attchd Attached to home

Basment Basement Garage

BuiltIn Built-In (Garage part of house - typically has room above garage)

CarPort Car Port

Detchd Detached from home

NA No Garage

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

Fin	Finished
RFn	Rough Finished
Unf	Unfinished
NA	No Garage

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

Ex	Excellent
Gd	Good
TA	Typical/Average
Fa	Fair
Po	Poor
NA	No Garage

GarageCond: Garage condition

Ex	Excellent
Gd	Good
TA	Typical/Average
Fa	Fair
Po	Poor
NA	No Garage

PavedDrive: Paved driveway

Y	Paved
P	Partial Pavement
N	Dirt/Gravel

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
NA	No Pool

Fence: Fence quality

GdPrv	Good Privacy
MnPrv	Minimum Privacy
GdWo	Good Wood
MnWw	Minimum Wood/Wire
NA	No Fence

MiscFeature: Miscellaneous feature not covered in other categories

Elev	Elevator
Gar2	2nd Garage (if not described in garage section)
Othr	Other
Shed	Shed (over 100 SF)

TenC	Tennis Court
NA	None

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

SaleType: Type of sale

WD	Warranty Deed - Conventional
CWD	Warranty Deed - Cash
VWD	Warranty Deed - VA Loan
New	Home just constructed and sold
COD	Court Officer Deed/Estate
Con	Contract 15% Down payment regular terms
ConLw	Contract Low Down payment and low interest
ConLI	Contract Low Interest
ConLD	Contract Low Down
Oth	Other

SaleCondition: Condition of sale

Normal	Normal Sale
Abnorml	Abnormal Sale - trade, foreclosure, short sale
AdjLand	Adjoining Land Purchase
Alloca	Allocation - two linked properties with separate deeds, typically condo with a garage unit
Family	Sale between family members
Partial	Home was not completed when last assessed (associated with New Homes)

Data types of the fields:

Below is the information of all the attributes with their respective datatypes:

Column name	datatype
Id	int64
MSSubClass	int64
MSZoning	object
LotFrontage	float64
LotArea	int64
Street	object
Alley	object
LotShape	object
LandContour	object
Utilities	object
LotConfig	object
LandSlope	object
Neighborhood	object
Condition1	object
Condition2	object
BldgType	object
HouseStyle	object
OverallQual	int64
OverallCond	int64
YearBuilt	int64
YearRemodAdd	int64
RoofStyle	object

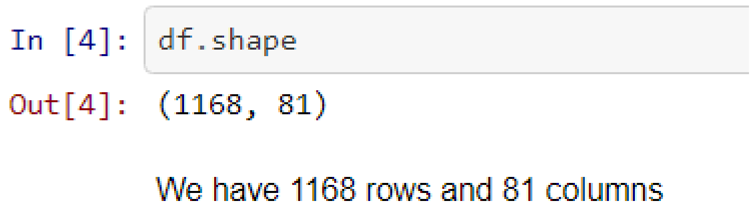
RoofMatl	object
Exterior1st	object
Exterior2nd	object
MasVnrType	object
MasVnrArea	float64
ExterQual	object
ExterCond	object
Foundation	object
BsmtQual	object
BsmtCond	object
BsmtExposure	object
BsmtFinType1	object
BsmtFinSF1	int64
BsmtFinType2	object
BsmtFinSF2	int64
BsmtUnfSF	int64
TotalBsmtSF	int64
Heating	object
HeatingQC	object
CentralAir	object
Electrical	object
1stFlrSF	int64
2ndFlrSF	int64
LowQualFinSF	int64
GrLivArea	int64

BsmtFullBath	int64
BsmtHalfBath	int64
FullBath	int64
HalfBath	int64
BedroomAbvGr	int64
KitchenAbvGr	int64
KitchenQual	object
TotRmsAbvGrd	int64
Functional	object
Fireplaces	int64
FireplaceQu	object
GarageType	object
GarageYrBlt	float64
GarageFinish	object
GarageCars	int64
GarageArea	int64
GarageQual	object
GarageCond	object
PavedDrive	object
WoodDeckSF	int64
OpenPorchSF	int64
EnclosedPorch	int64
3SsnPorch	int64
ScreenPorch	int64
PoolArea	int64

PoolQC	object
Fence	object
MiscFeature	object
MiscVal	int64
MoSold	int64
YrSold	int64
SaleType	object
SaleCondition	object
SalePrice	int64

- Data Pre-processing Done

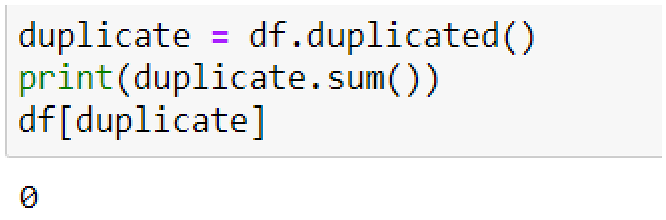
1) First we checked the data set dimensions



```
In [4]: df.shape

Out[4]: (1168, 81)
```

We have 1168 rows and 81 columns

2) Then we checked whether there is any repeating data available



```
duplicate = df.duplicated()
print(duplicate.sum())
df[duplicate]

0
```

3) We checked the outliers using the Box Plot and replaced the outliers with more appropriate values. Removal of outliers can also be done but taking the Data Loss percentage into consideration It is better to replace the outlier

- Hardware and Software Requirements and Tools Used

  1) Software: Jupyter Notebook - To code and build the project in python
  2) Libraries:
     a) numpy - To perform basic math operations
     b) pandas - To perform basic File operations
     c) Matplotlib - To plot Different Graphs/ Plots
     d) Seaborn - Advance library to enhance the quality of graphs/plots
     e) warnings - To ignore the unwanted warnings raised while interpreting the code
     f) sklearn - To build the Prediction models
     g) imblearn - To balance our dataset distribution

Model/s Development and Evaluation

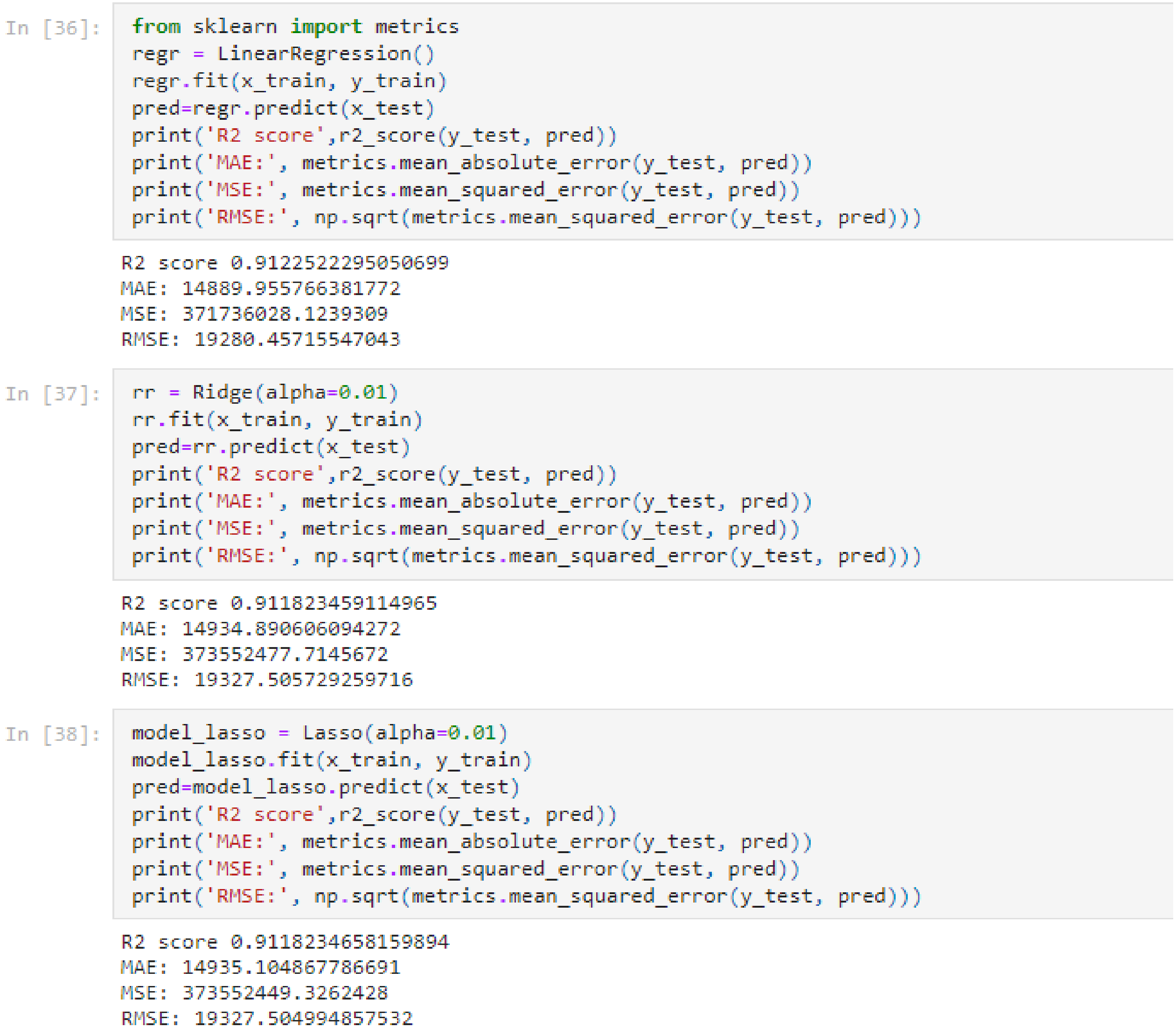
- Identification of possible problem-solving approaches (methods)

  We used different approaches from checking the dataset quality to building the model. We checked the null values and repeated rows in the dataset. For checking the Outliers, we used Box Plot and to remove the outliers we used IQR method. Then we moved to next step of checking data distribution and skewness. To scale the data, we used MinMax Scaler method and to remove the skewness we first checked the log and square root method but skewness of the dataset was not getting removed from it so we performed the Power transform to remove skewness.. We started building different models and checked their R2 score and selected the best suited model to perform Hyper tuning on. We got Random Forest Algo with the best result and after performing Hyper tuning we finalized the model.

- Testing of Identified Approaches (Algorithms)

  1) Linear Regression
  2) Decision Tree
  3) Elastic Net
  4) Lasso
  5) Random Forest
  6) Ridge

- Run and Evaluate selected models



In [36]:

```
from sklearn import metrics
regr = LinearRegression()
regr.fit(x_train, y_train)
pred=regr.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.9122522295050699
MAE: 14889.955766381772
MSE: 371736028.1239309
RMSE: 19280.45715547043
```

In [37]:

```
rr = Ridge(alpha=0.01)
rr.fit(x_train, y_train)
pred=rr.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.911823459114965
MAE: 14934.890606094272
MSE: 373552477.7145672
RMSE: 19327.505729259716
```

In [38]:

```
model_lasso = Lasso(alpha=0.01)
model_lasso.fit(x_train, y_train)
pred=model_lasso.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.9118234658159894
MAE: 14935.104867786691
MSE: 373552449.3262428
RMSE: 19327.504994857532
```

```
In [39]:  model_enet = ElasticNet(alpha = 0.01)
          model_enet.fit(x_train, y_train)
          pred=model_enet.predict(x_test)
          print('R2 score',r2_score(y_test, pred))
          print('MAE:', metrics.mean_absolute_error(y_test, pred))
          print('MSE:', metrics.mean_squared_error(y_test, pred))
          print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.9119966779517421
MAE: 14928.021981156287
MSE: 372818650.72367525
RMSE: 19308.512390230255
```

```
In [40]:  from sklearn.tree import DecisionTreeRegressor
          from sklearn import metrics
          dtr=DecisionTreeRegressor()
          dtr.fit(x_train,y_train)
          pred=dtr.predict(x_test)
          print('R2 score', r2_score(y_test, pred))
          print('MAE:', metrics.mean_absolute_error(y_test, pred))
          print('MSE:', metrics.mean_squared_error(y_test, pred))
          print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.77230316602535
MAE: 23381.713675213676
MSE: 964618430.767094
RMSE: 31058.30695268327
```

```
In [41]:  from sklearn.ensemble import RandomForestRegressor
          rdr = RandomForestRegressor()
          rdr.fit(x_train,y_train)
          pred1=rdr.predict(x_test)
          print('R2 score',r2_score(y_test, pred1))
          print('MAE:', metrics.mean_absolute_error(y_test, pred1))
          print('MSE:', metrics.mean_squared_error(y_test, pred1))
          print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,pred1)))
```

```
R2 score 0.8870777493514164
MAE: 15944.455769230768
MSE: 478385589.8121632
RMSE: 21872.027565183875
```

- Key Metrics for success in solving problem under consideration

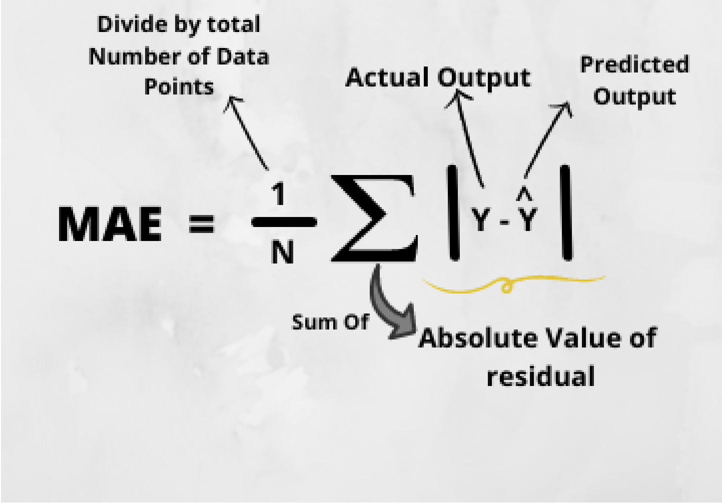
1) Mean Absolute Error(MAE)

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

To better understand, let's take an example you have input data and output data and use Linear Regression, which draws a best-fit line.

Now you have to find the MAE of your model which is basically a mistake made by the model known as an error. Now find the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset.

so, sum all the errors and divide them by a total number of observations And this is MAE. And we aim to get a minimum MAE because this is a loss.


Divide by total Number of Data Points
Actual Output
Predicted Output
MAE = 1/N ∑ |Y - Ŷ|
Sum Of
Absolute Value of residual

Advantages of MAE

The MAE you get is in the same unit as the output variable.

It is most Robust to outliers.

Disadvantages of MAE

The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.

```
from sklearn.metrics import mean_absolute_error
```

```
print("MAE",mean_absolute_error(y_test,y_pred))
```

Now to overcome the disadvantage of MAE next metric came as MSE.

2) Mean Squared Error(MSE)

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

So, above we are finding the absolute difference and here we are finding the squared difference.

What actually the MSE represents? It represents the squared distance between actual and predicted values. we perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}}$$

Advantages of MSE

The graph of MSE is differentiable, so you can easily use it as a loss function.

Disadvantages of MSE

The value you get after calculating MSE is a squared unit of output. for example, the output variable is in meter(m) then after calculating MSE the output we get is in meter squared.

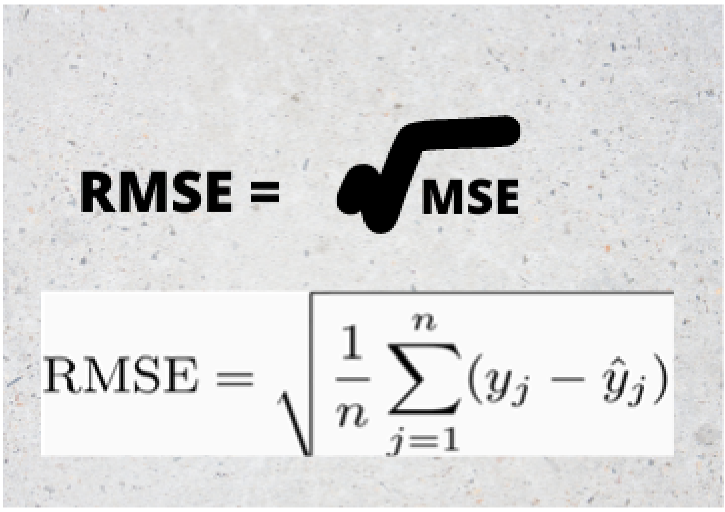
If you have outliers in the dataset then it penalizes the outliers most and the calculated MSE is bigger. So, in short, It is not Robust to outliers which were an advantage in MAE.

```
from sklearn.metrics import mean_squared_error
```

```
print("MSE",mean_squared_error(y_test,y_pred))
```

3) Root Mean Squared Error(RMSE)

As RMSE is clear by the name itself, that it is a simple square root of mean squared error.



$$\text{RMSE} = \sqrt{\text{MSE}}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)}$$

Advantages of RMSE

The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.

Disadvantages of RMSE

It is not that robust to outliers as compared to MAE.

for performing RMSE we have to NumPy NumPy square root function over MSE.

```
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
```

Most of the time people use RMSE as an evaluation metric and mostly when you are working with deep learning techniques the most preferred metric is RMSE.

4) Root Mean Squared Log Error(RMSLE)

Taking the log of the RMSE metric slows down the scale of error. The metric is very helpful when you are developing a model without calling the inputs. In that case, the output will vary on a large scale.

To control this situation of RMSE we take the log of calculated RMSE error and resultant we get as RMSLE.

To perform RMSLE we have to use the NumPy log function over RMSE.

```
print("RMSE",np.log(np.sqrt(mean_squared_error(y_test,y_pred)))
)
```

It is a very simple metric that is used by most of the datasets hosted for Machine Learning competitions.

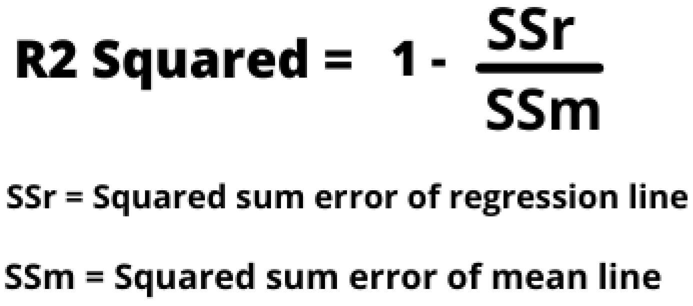
5) R Squared (R2)

R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.

In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.

So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically R2 squared calculates how must regression line is better than a mean line.

Hence, R2 squared is also known as Coefficient of Determination or sometimes also known as Goodness of fit.



$$\text{R2 Squared} = 1 - \frac{SSr}{SSm}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

R2 Squared

Now, how will you interpret the R2 score? suppose If the R2 score is zero then the above regression line by mean line is equal means 1 so 1-1 is zero. So, in this case, both lines are overlapping means model performance is worst, It is not capable to take advantage of the output column.

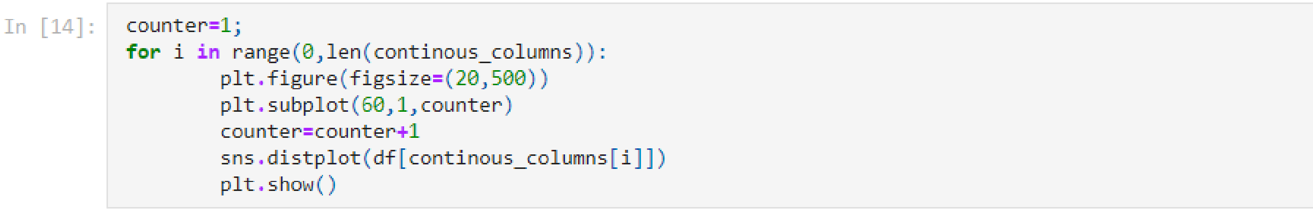
Now the second case is when the R2 score is 1, it means when the division term is zero and it will happen when the regression line does not make any mistake, it is perfect. In the real world, it is not possible.

So we can conclude that as our regression line moves towards perfection, R2 score move towards one. And the model performance improves.

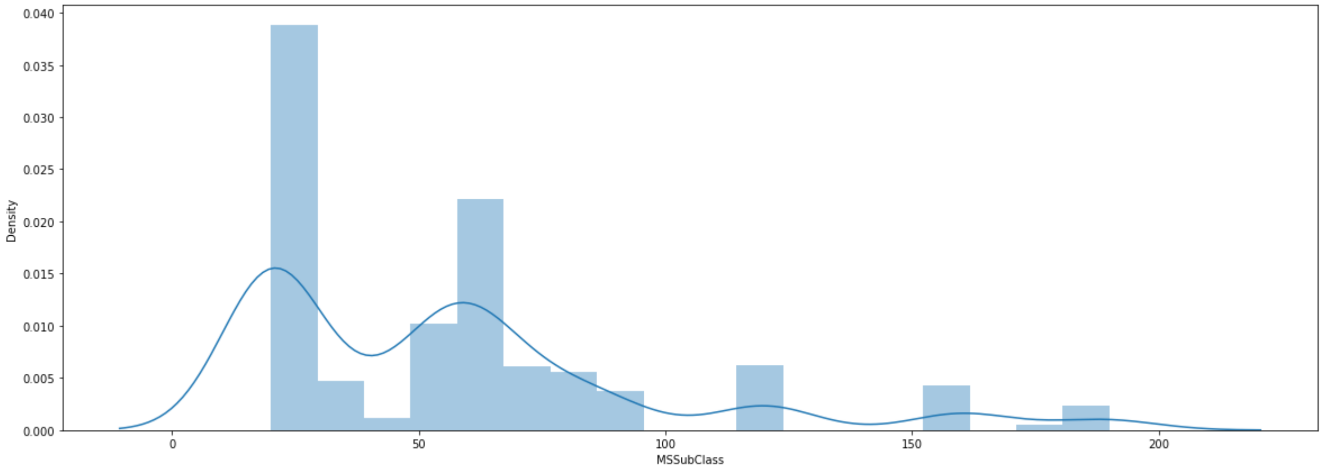
The normal case is when the R2 score is between zero and one like 0.8 which means your model is capable to explain 80 per cent of the variance of data.

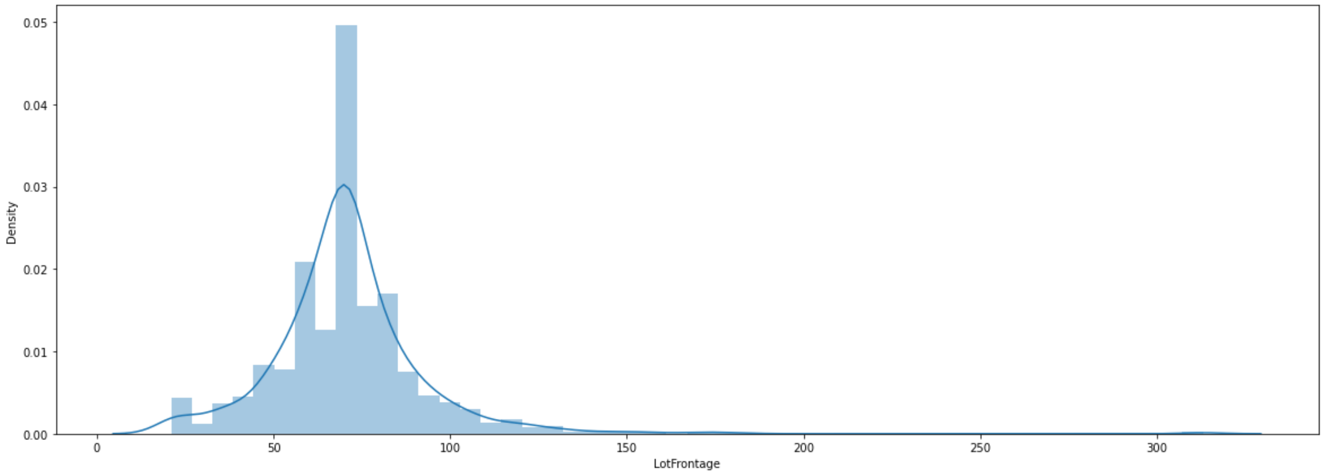
```
from sklearn.metrics import r2_score

r2 = r2_score(y_test,y_pred)

print(r2)
```

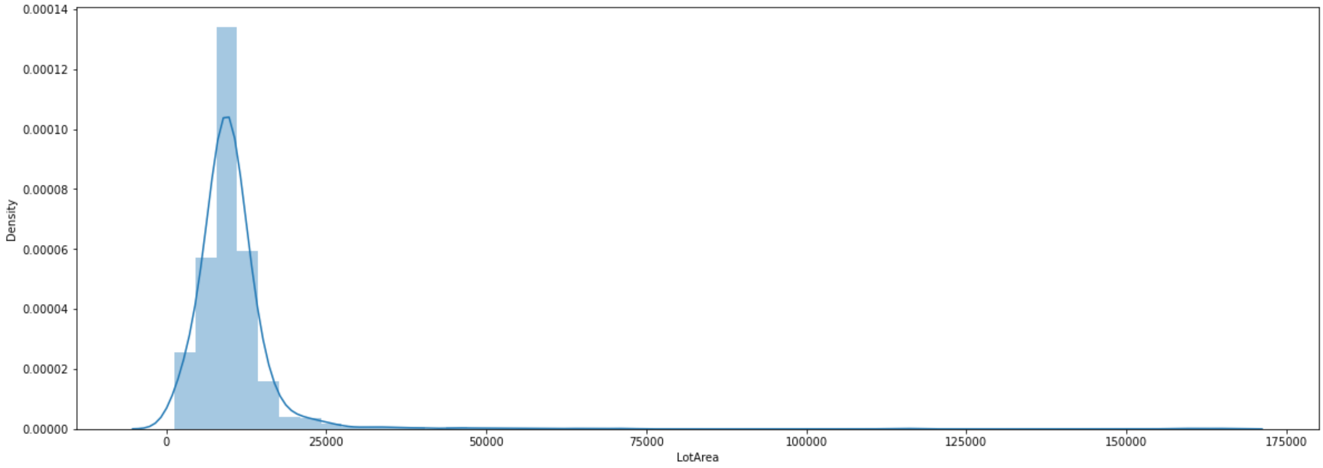
- Visualizations

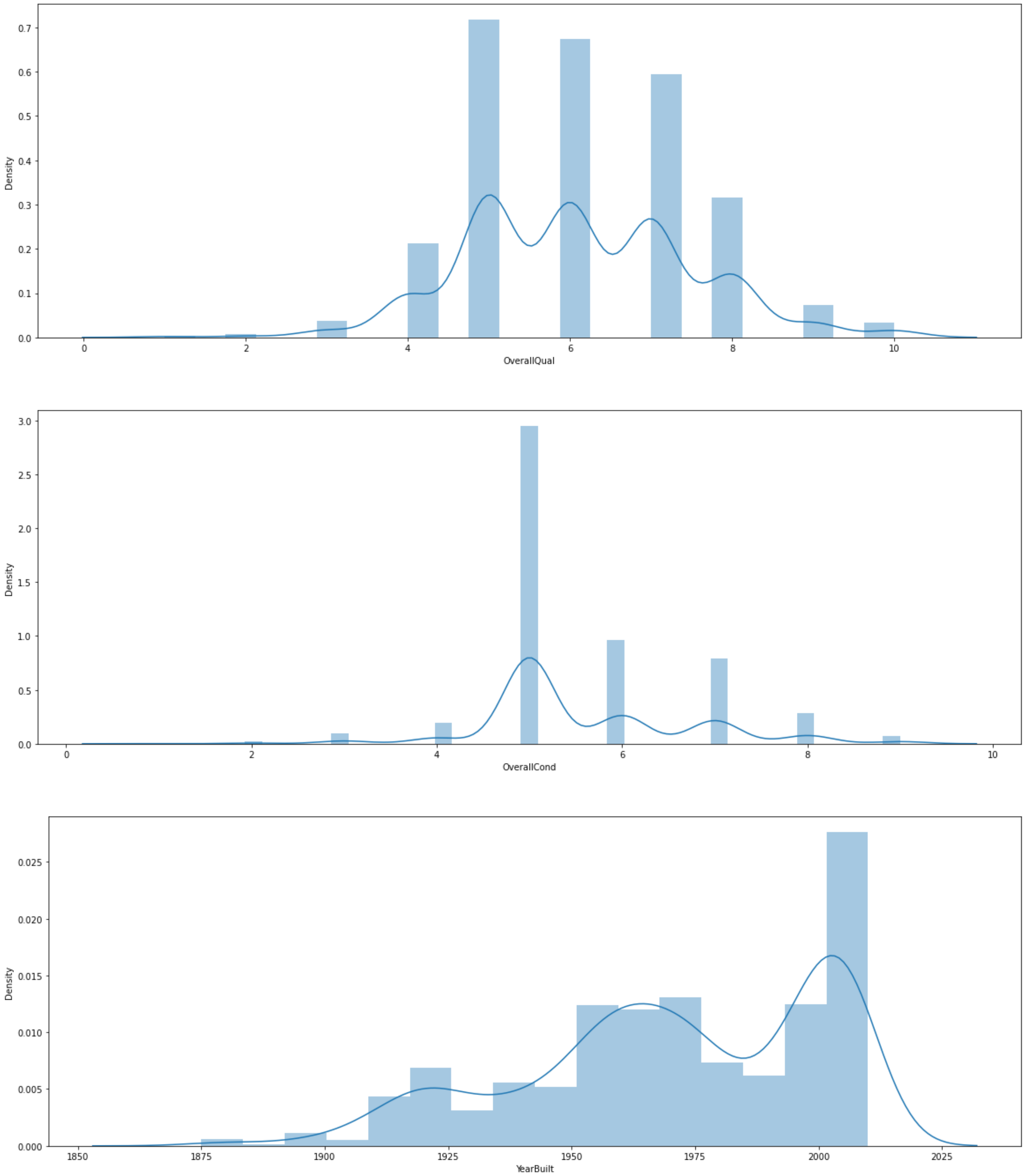


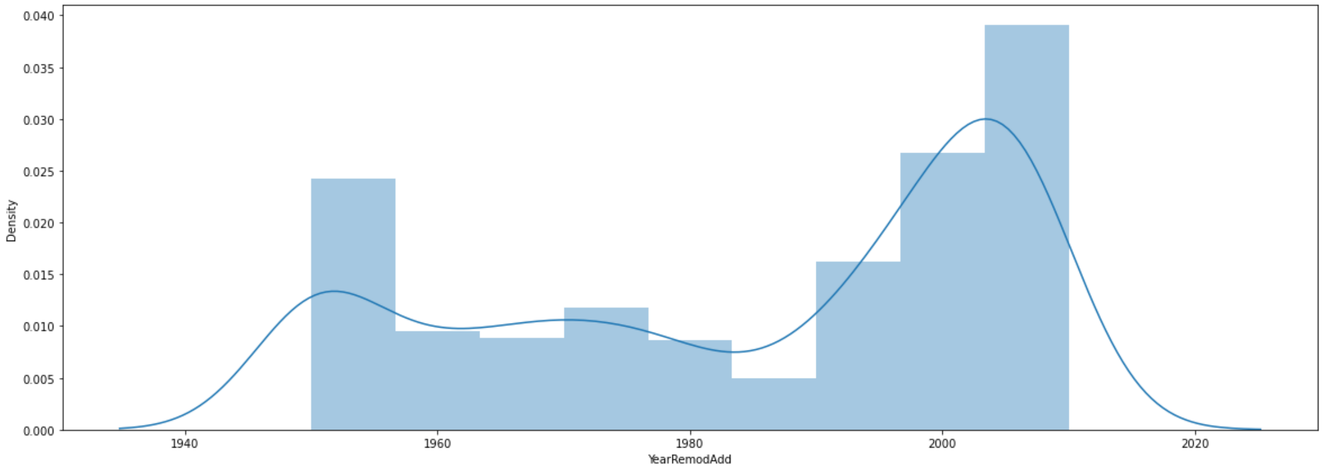
```
In [14]: counter=1;
         for i in range(0,len(continous_columns)):
                 plt.figure(figsize=(20,500))
                 plt.subplot(60,1,counter)
                 counter=counter+1
                 sns.distplot(df[continous_columns[i]])
                 plt.show()
```

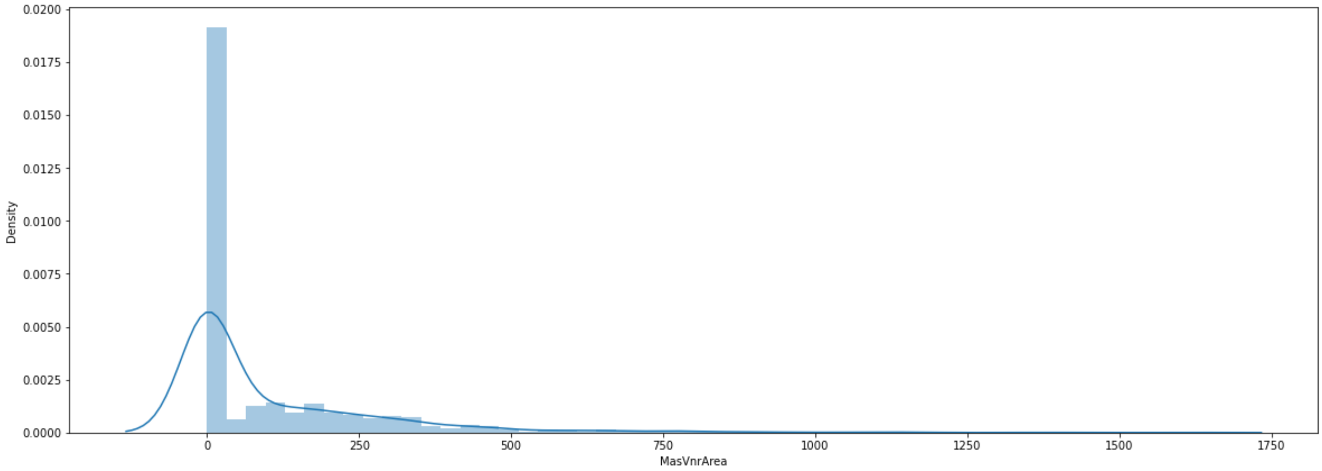


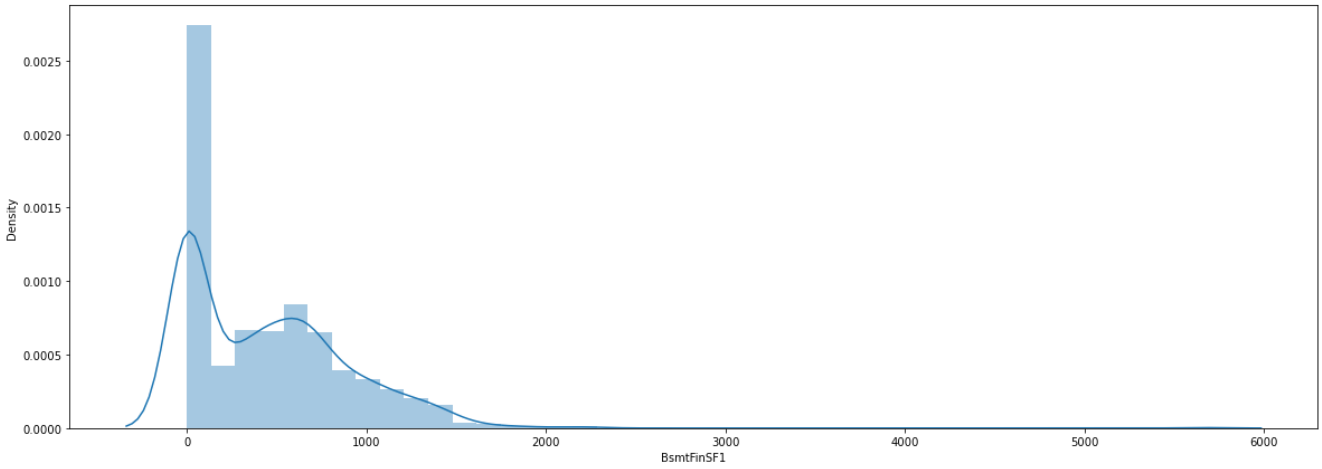


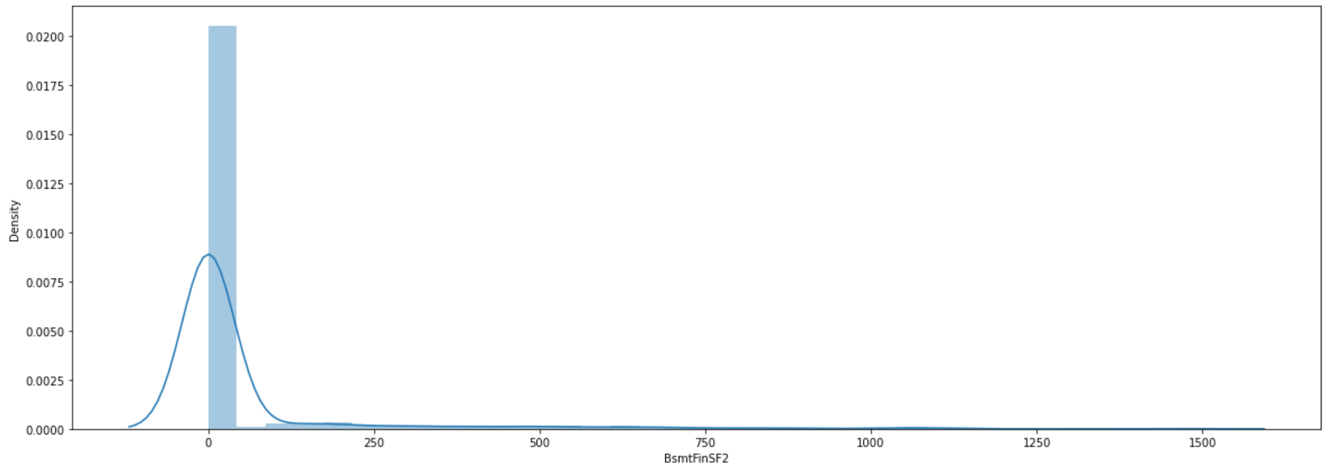


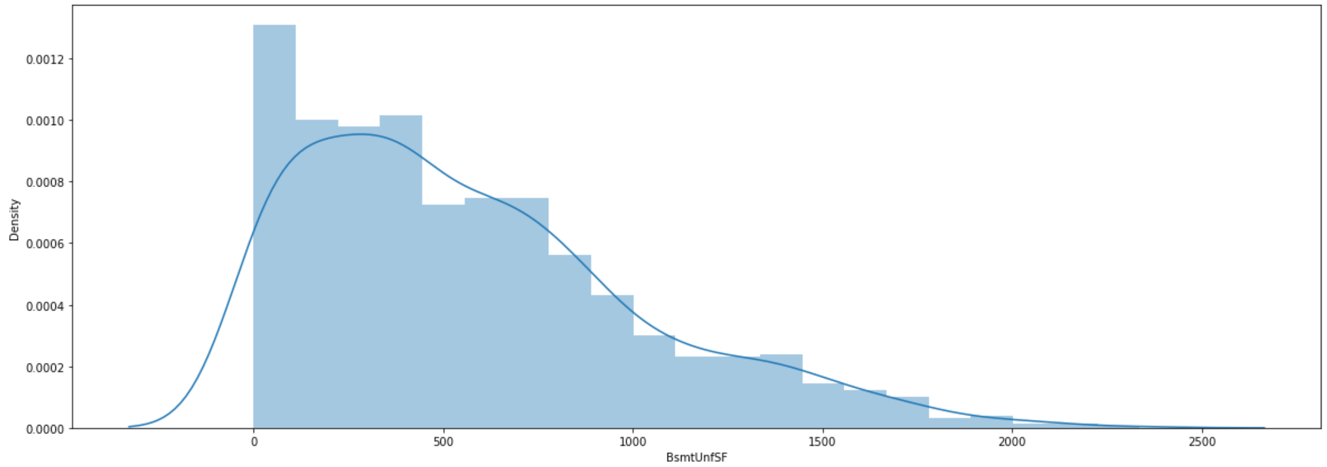

Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2
4
6
8
10
OverallQual
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
2
4
6
8
10
OverallCond
Density
0.000
0.005
0.010
0.015
0.020
0.025
1850
1875
1900
1925
1950
1975
2000
2025
YearBuilt

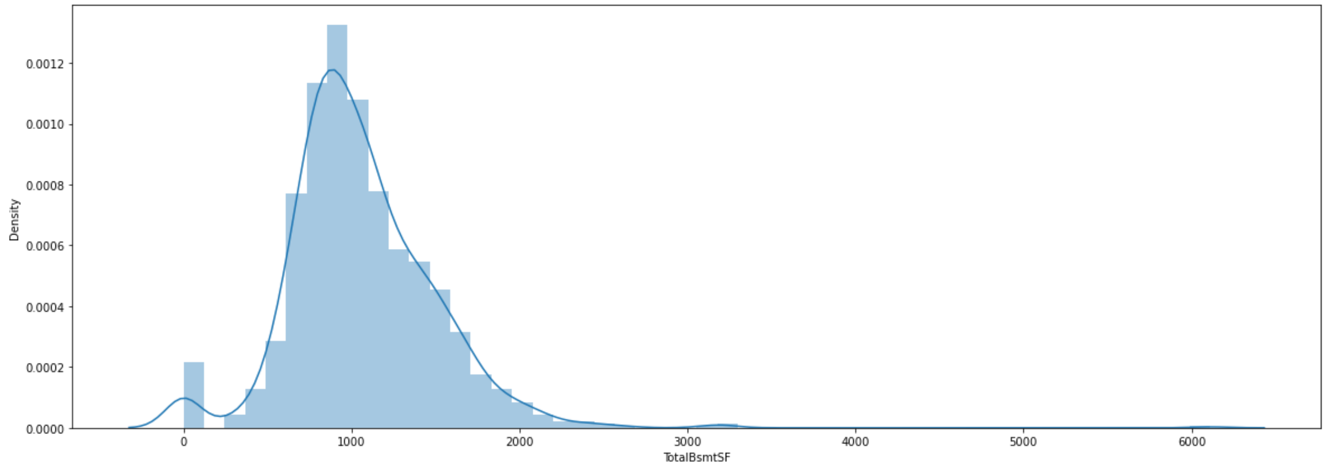

Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
1940
1960
1980
2000
2020
YearRemodAdd

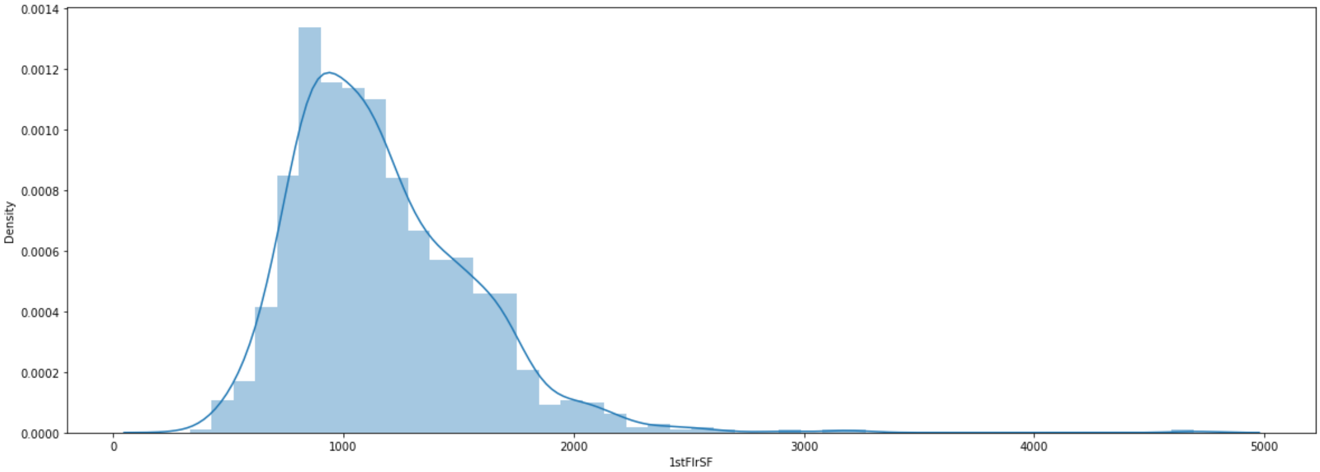

Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
250
500
750
1000
1250
1500
1750
MasVnrArea

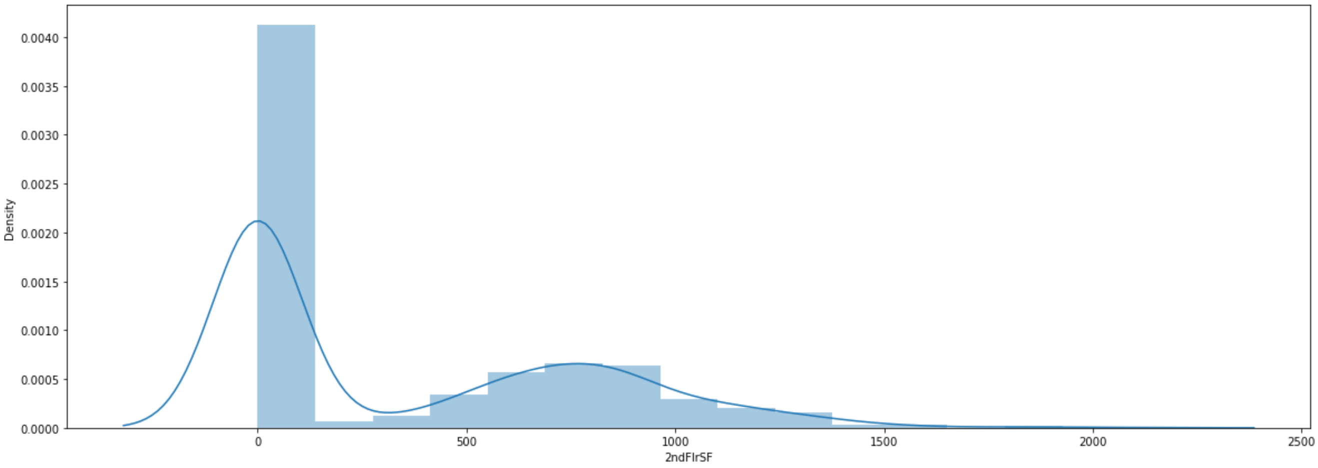

Density
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
1000
2000
3000
4000
5000
6000
BsmtFinSF1

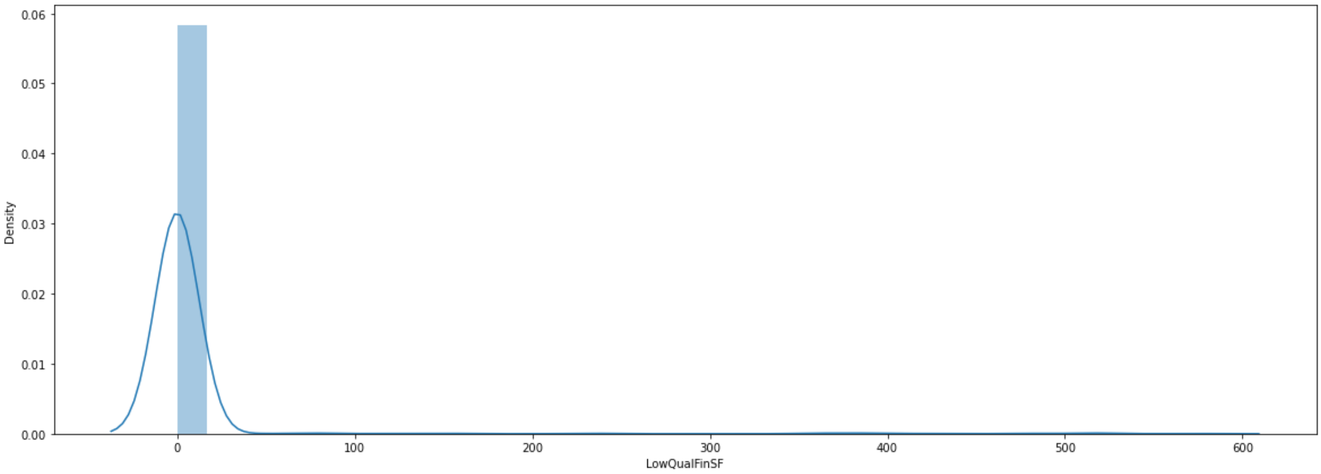

0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
0
250
500
750
1000
1250
1500
BsmtFinSF2

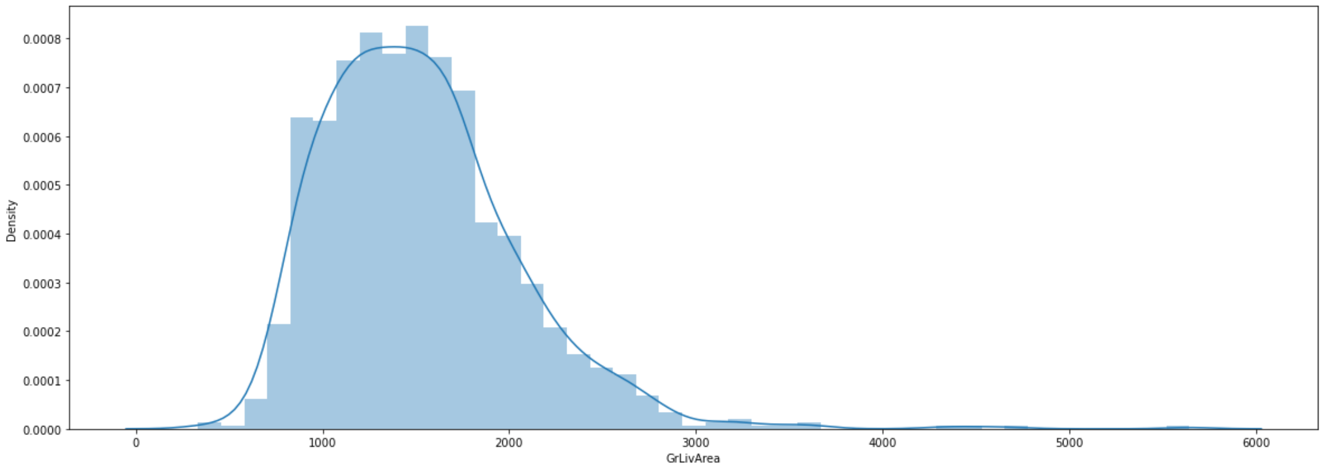

0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
500
1000
1500
2000
2500
BsmtUnfSF

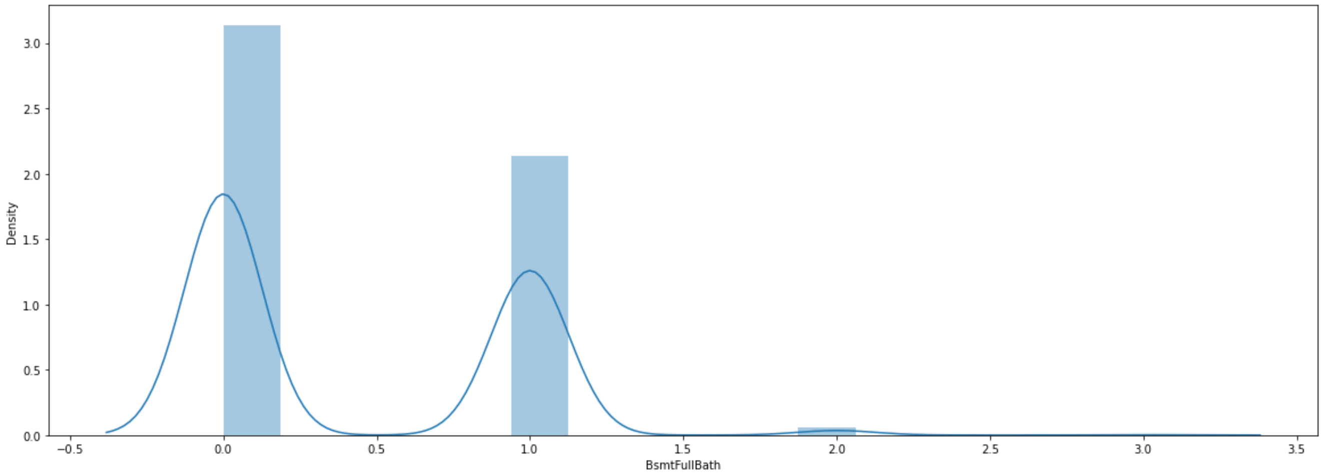

0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
1000
2000
3000
4000
5000
6000
TotalBsmtSF

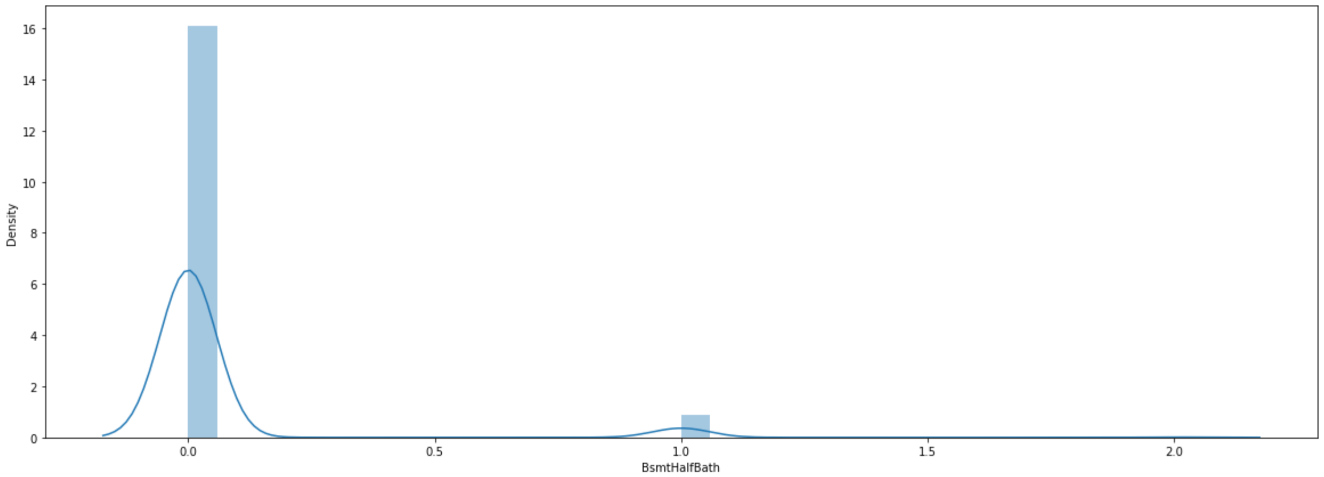

0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
1000
2000
3000
4000
5000
1stFlrSF

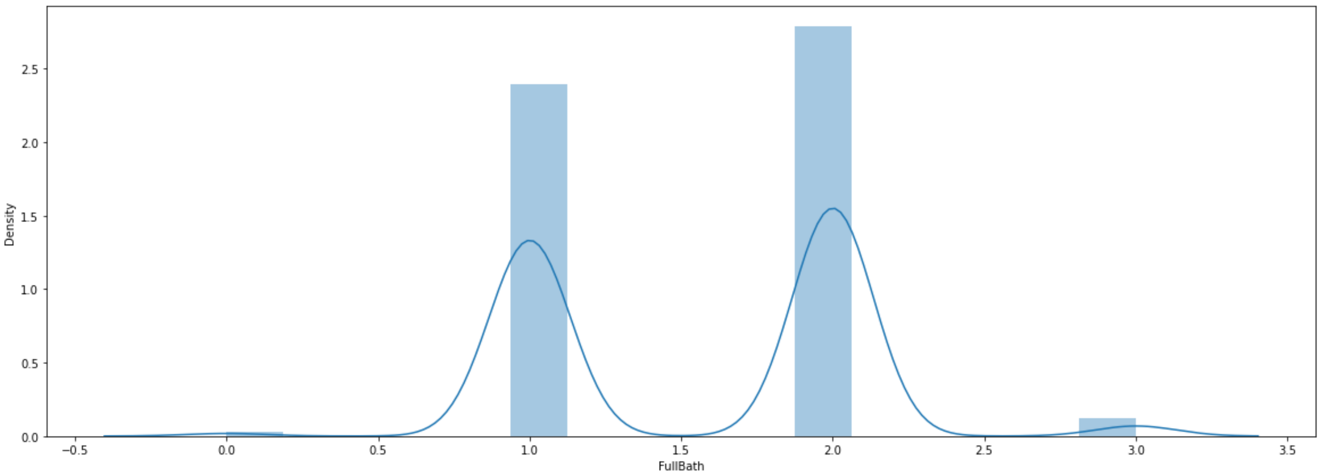

0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Density
0
500
1000
1500
2000
2500
2ndFlrSF


0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
0
100
200
300
400
500
600
LowQualFinSF

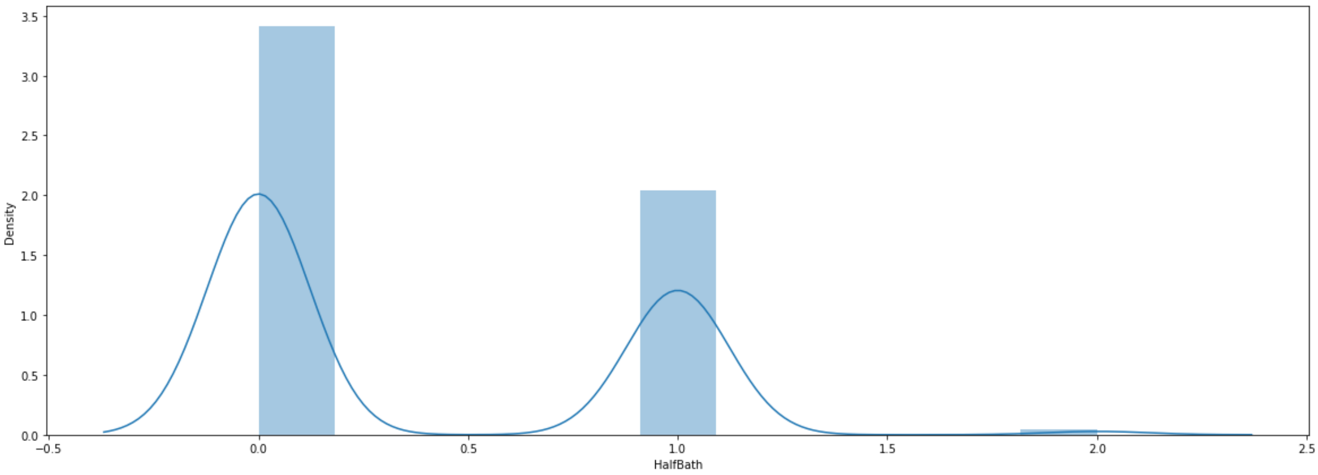

Density
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
1000
2000
3000
4000
5000
6000
GrLivArea


Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
BsmtFullBath

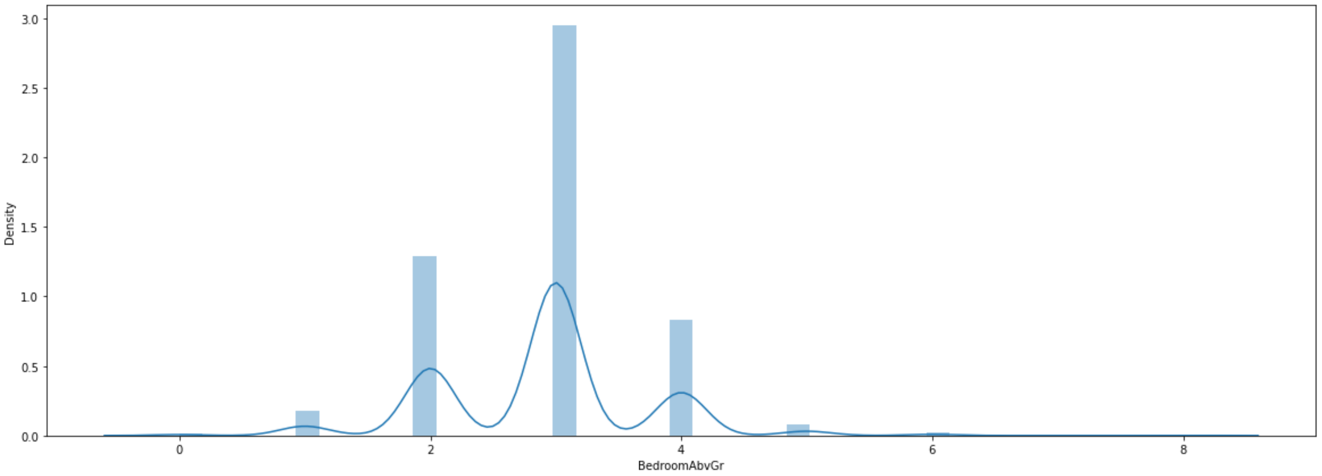

Density
16
14
12
10
8
6
4
2
0
0.0
0.5
1.0
1.5
2.0
BsmtHalfBath



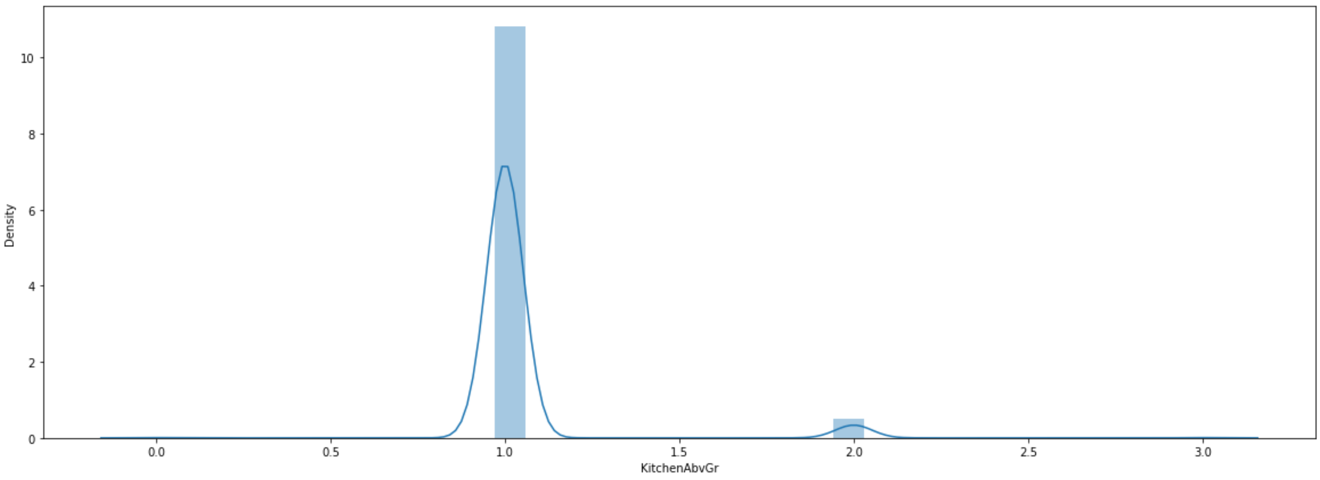
Density
0.0
0.5
1.0
1.5
2.0
2.5
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
FullBath

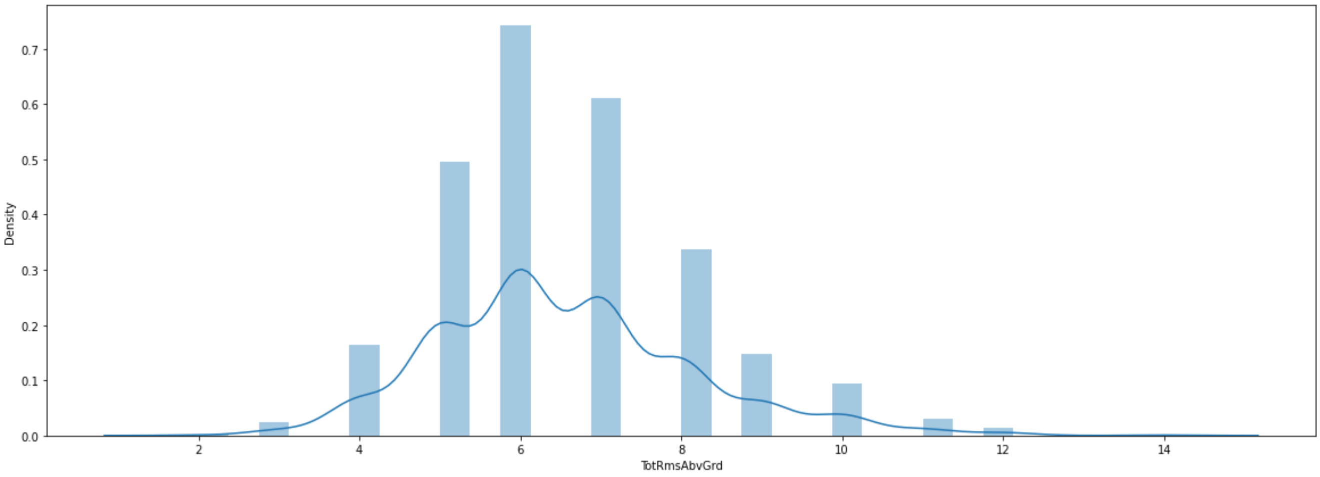


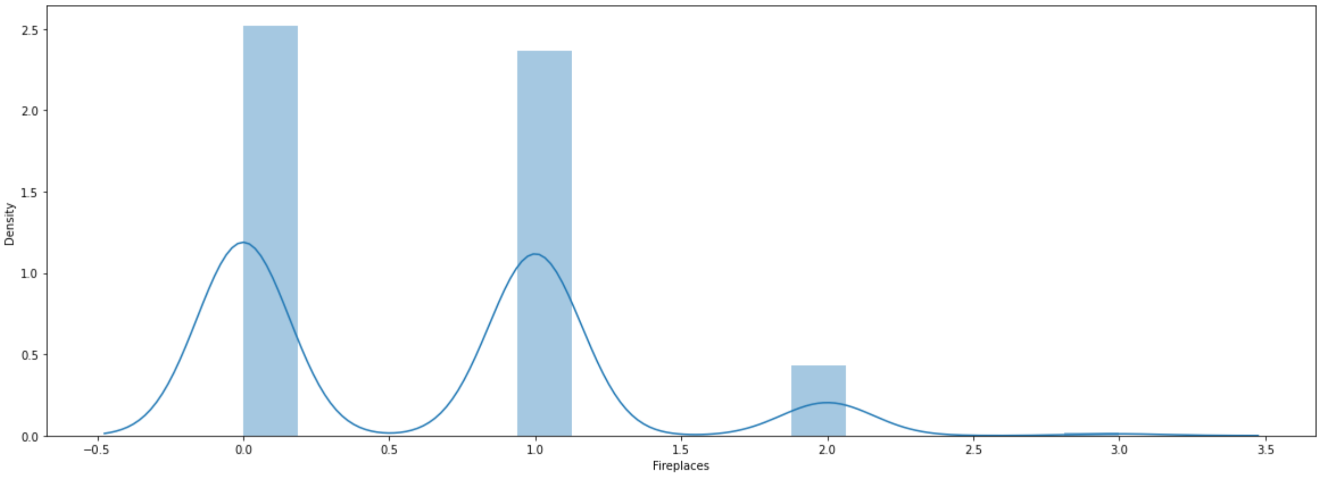
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
HalfBath

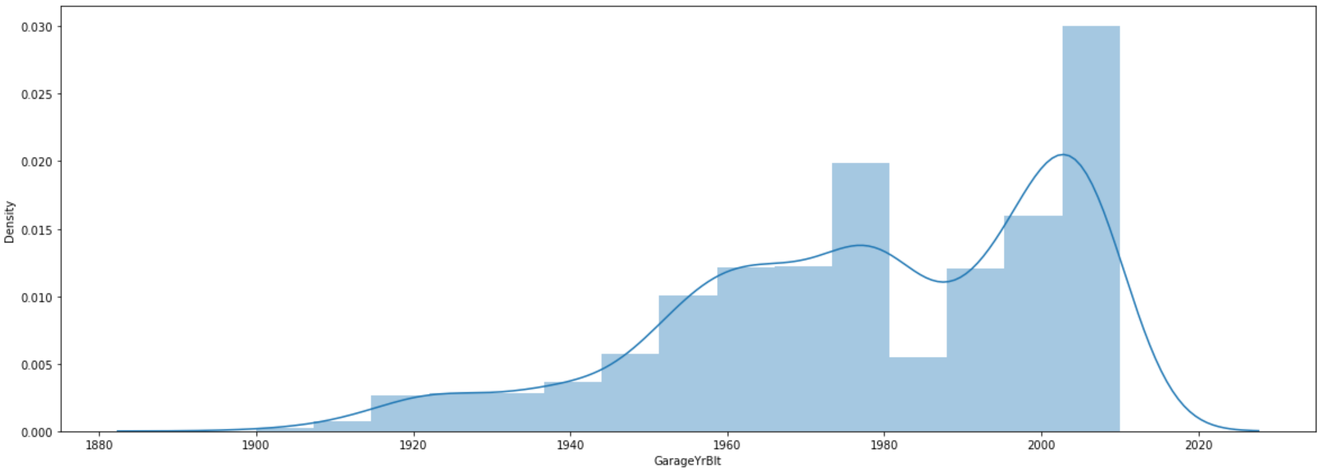


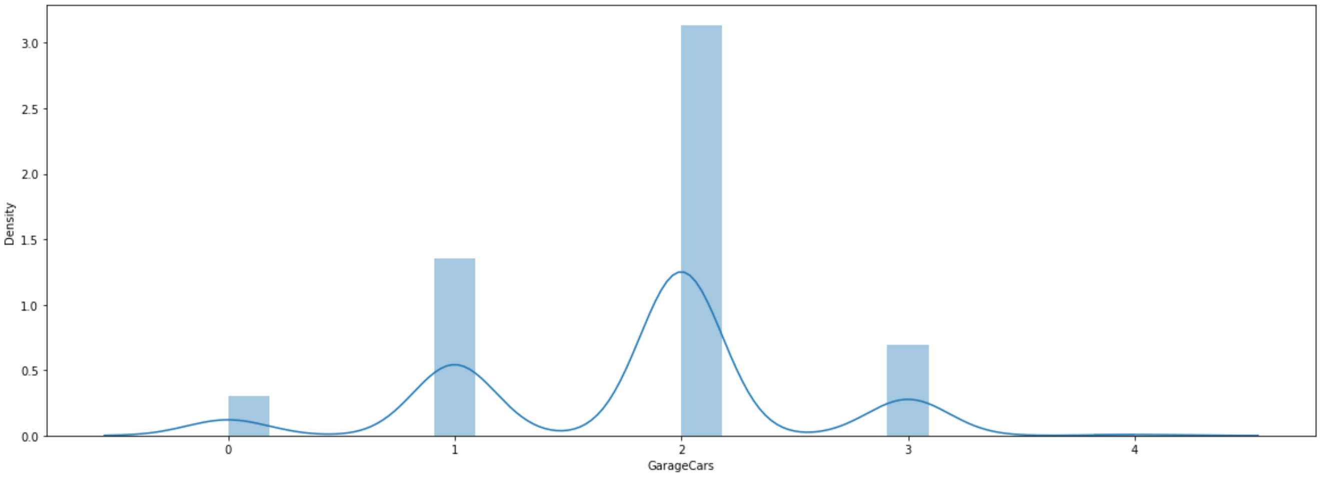
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
2
4
6
8
BedroomAbvGr

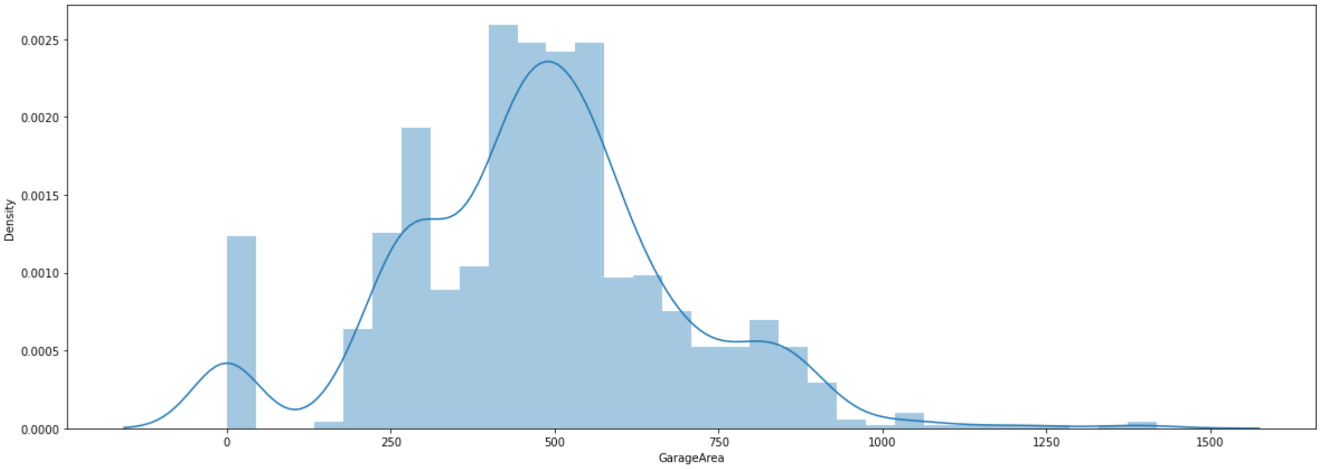

Density
0
2
4
6
8
10
0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr

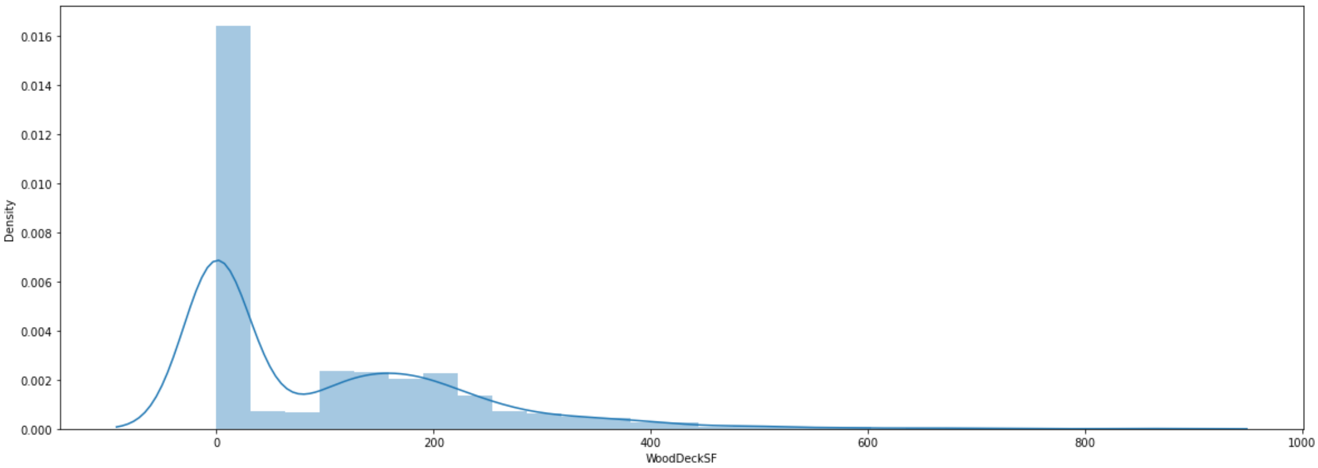

Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2
4
6
8
10
12
14
TotRmsAbvGrd


Density
0.0
0.5
1.0
1.5
2.0
2.5
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Fireplaces

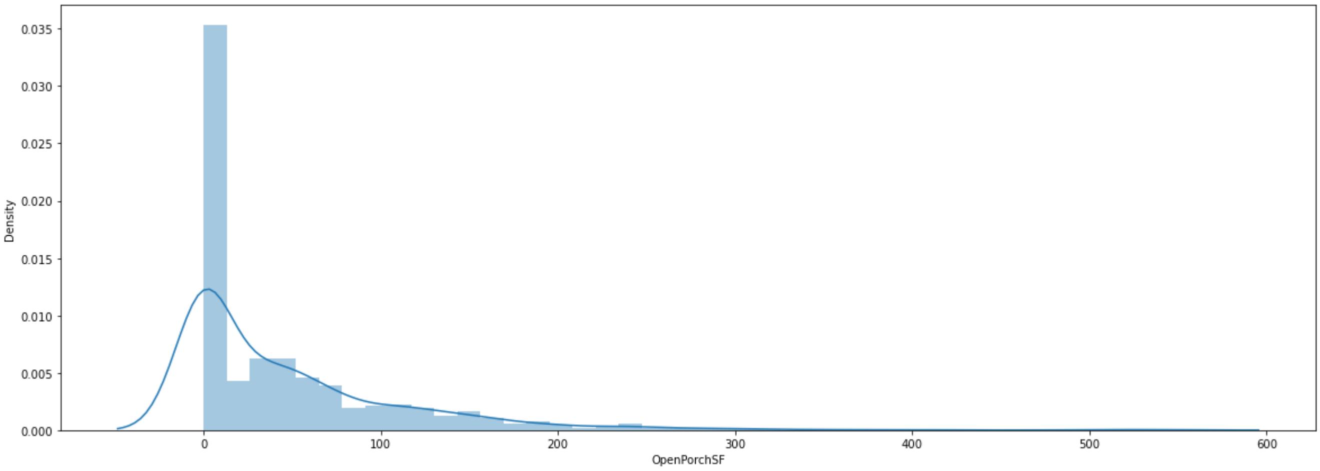

Density
0.030
0.025
0.020
0.015
0.010
0.005
0.000
1880
1900
1920
1940
1960
1980
2000
2020
GarageYrBlt


Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
GarageCars

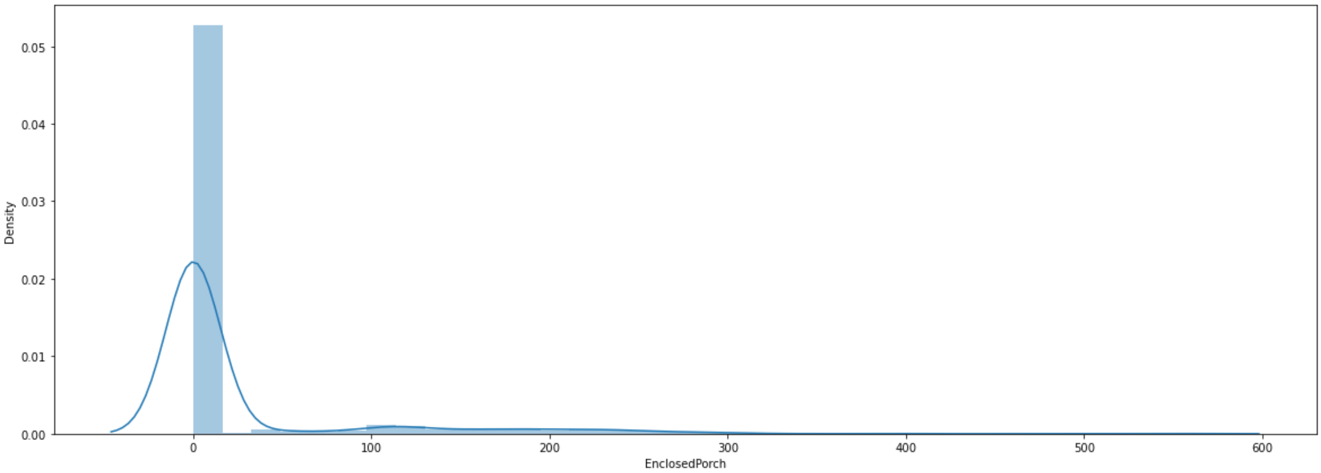

Density
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
250
500
750
1000
1250
1500
GarageArea



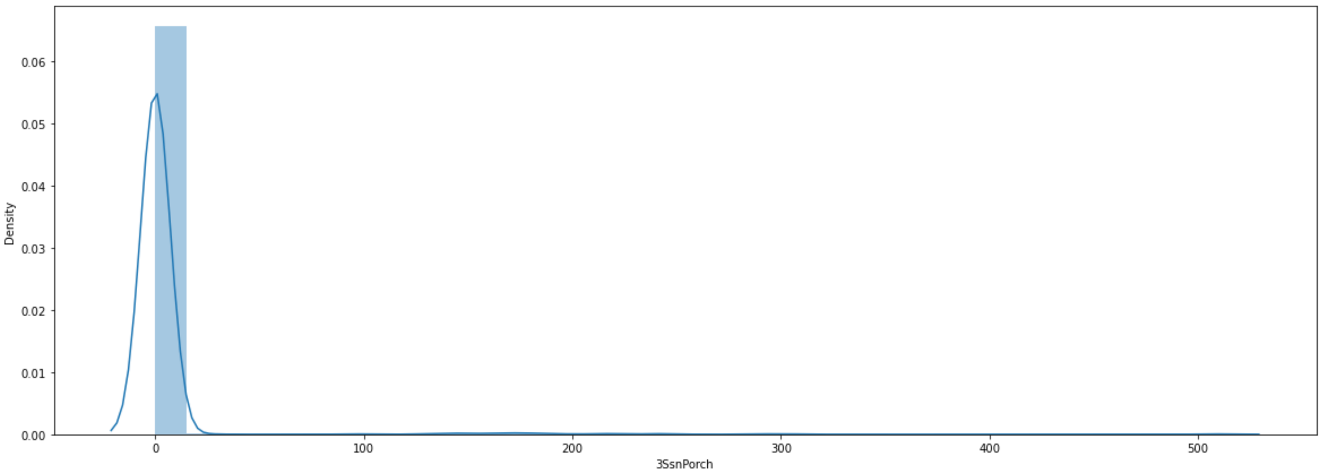
Density
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
200
400
600
800
1000
WoodDeckSF



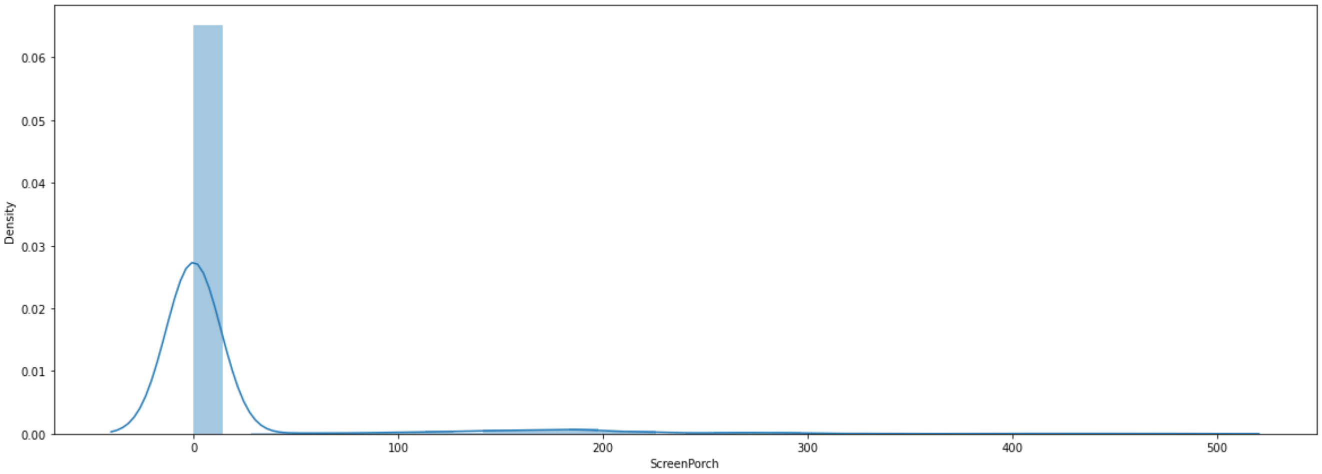
Density
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
100
200
300
400
500
600
OpenPorchSF



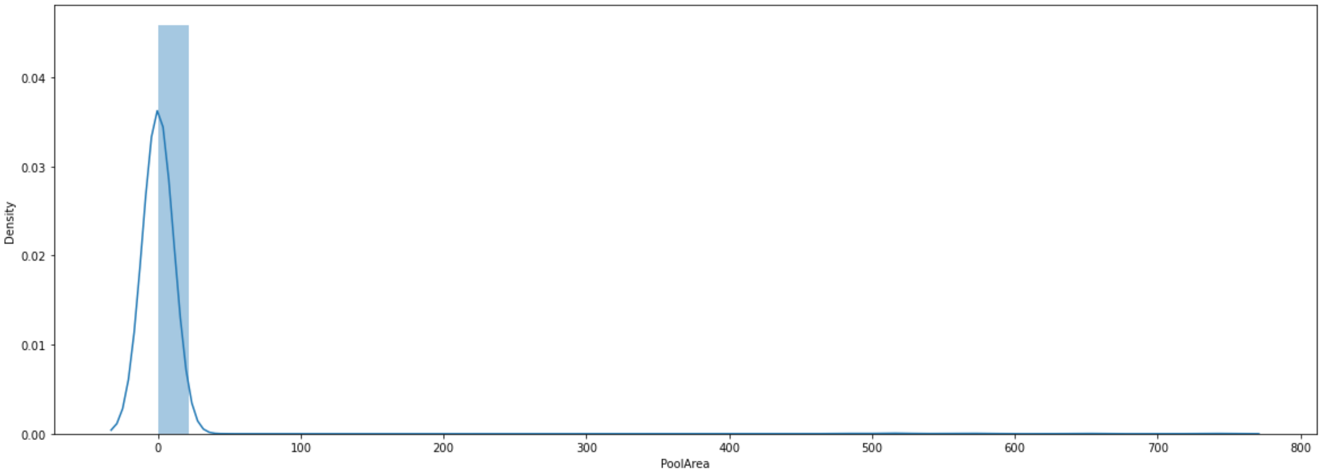
Density
0.05
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
500
600
EnclosedPorch



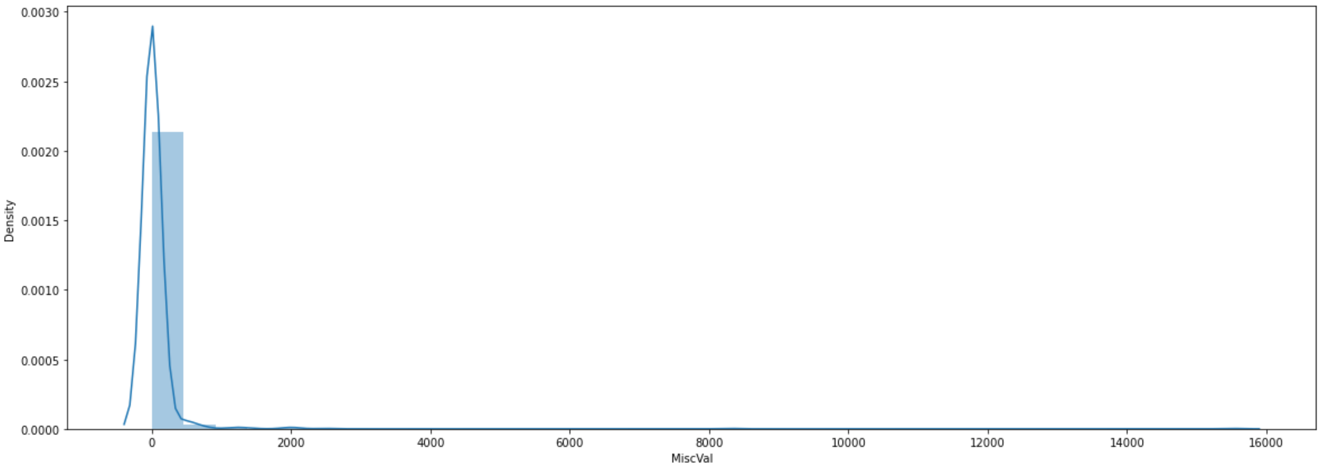
Density
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
500
3SsnPorch

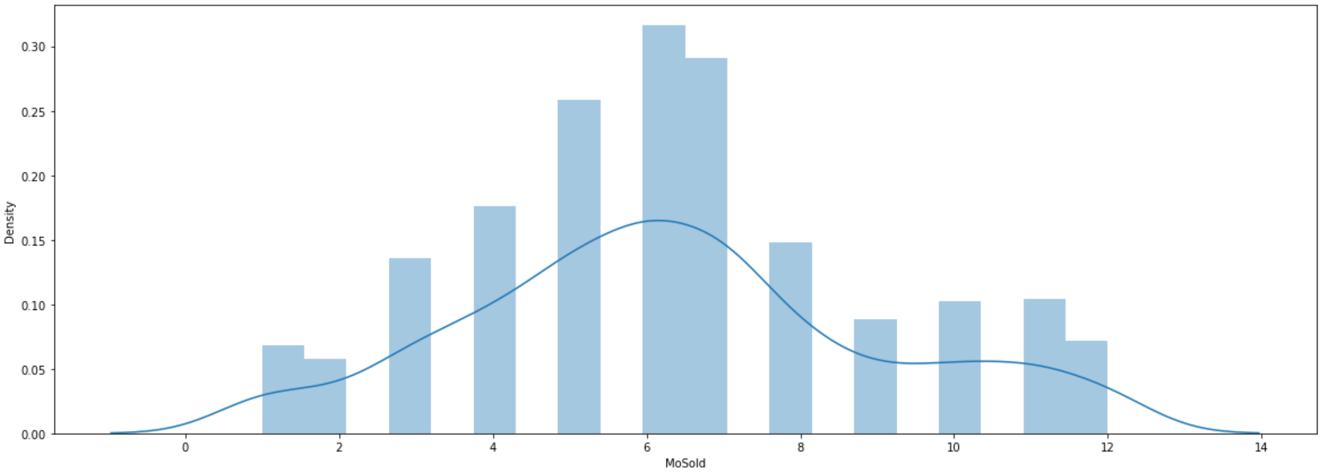


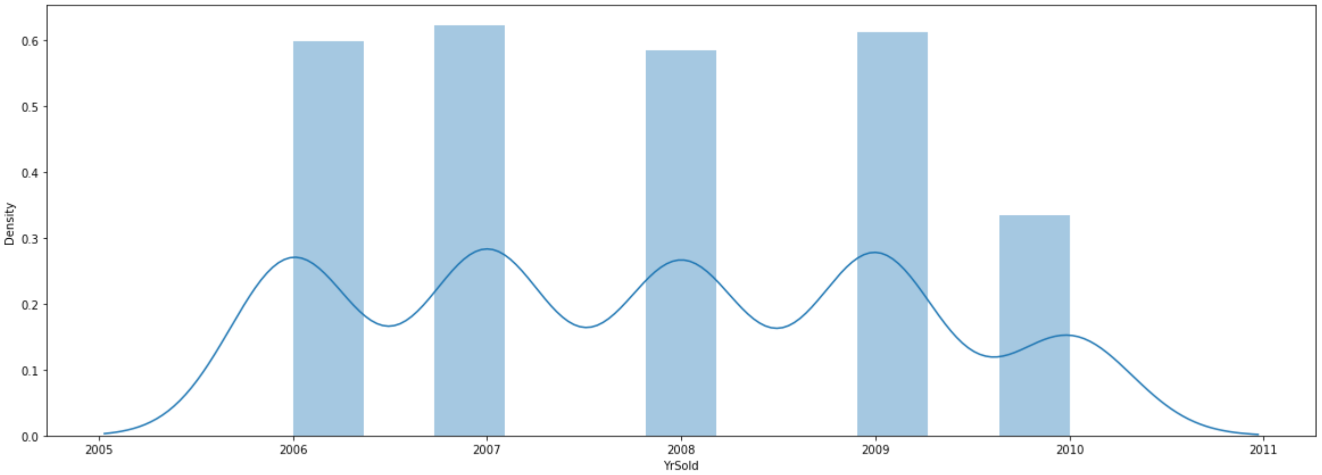
Density
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
500
ScreenPorch

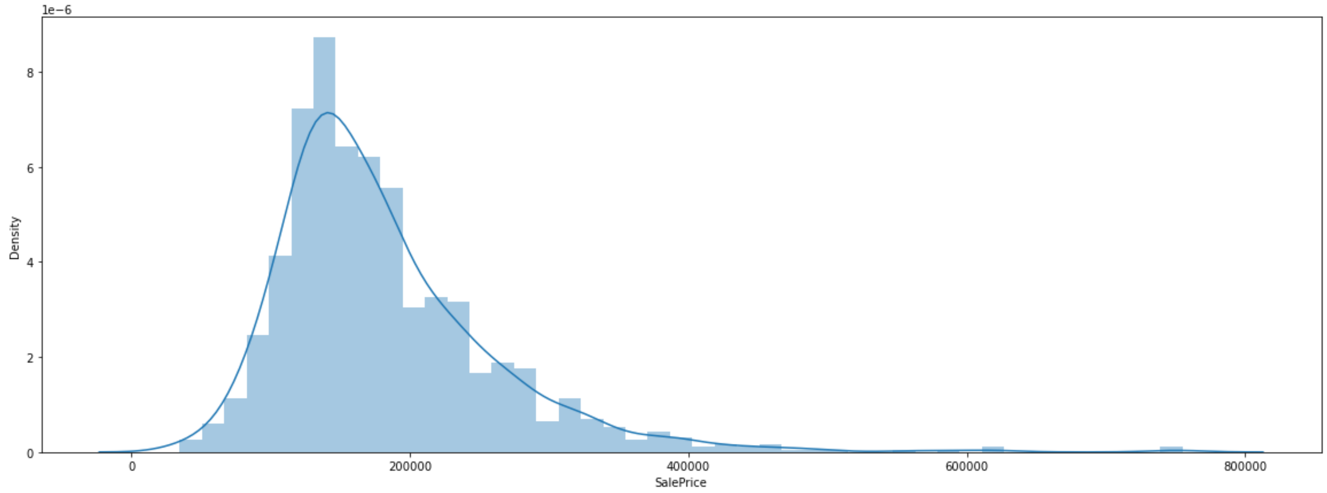


Density
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
500
600
700
800
PoolArea


0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Density
0
2000
4000
6000
8000
10000
12000
14000
16000
MiscVal


0.30
0.25
0.20
0.15
0.10
0.05
0.00
Density
0
2
4
6
8
10
12
14
MoSold


0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
2005
2006
2007
2008
2009
2010
2011
YrSold



Findings:

MSSubClass -> not normally distributed

LotFrontage -> normally distributed

LotArea -> Normally distributed

OverallQual-> Not normally distributed

Overall cond-> Not normally distributed

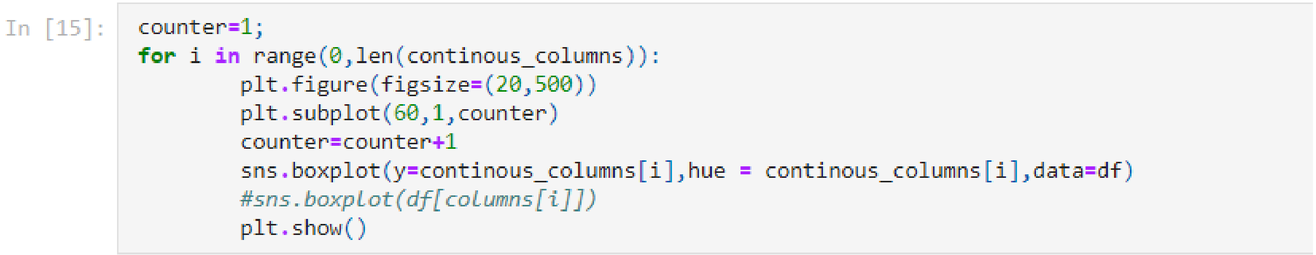
Year Built -> Not normally distributed

Year remod add->not normally distributed

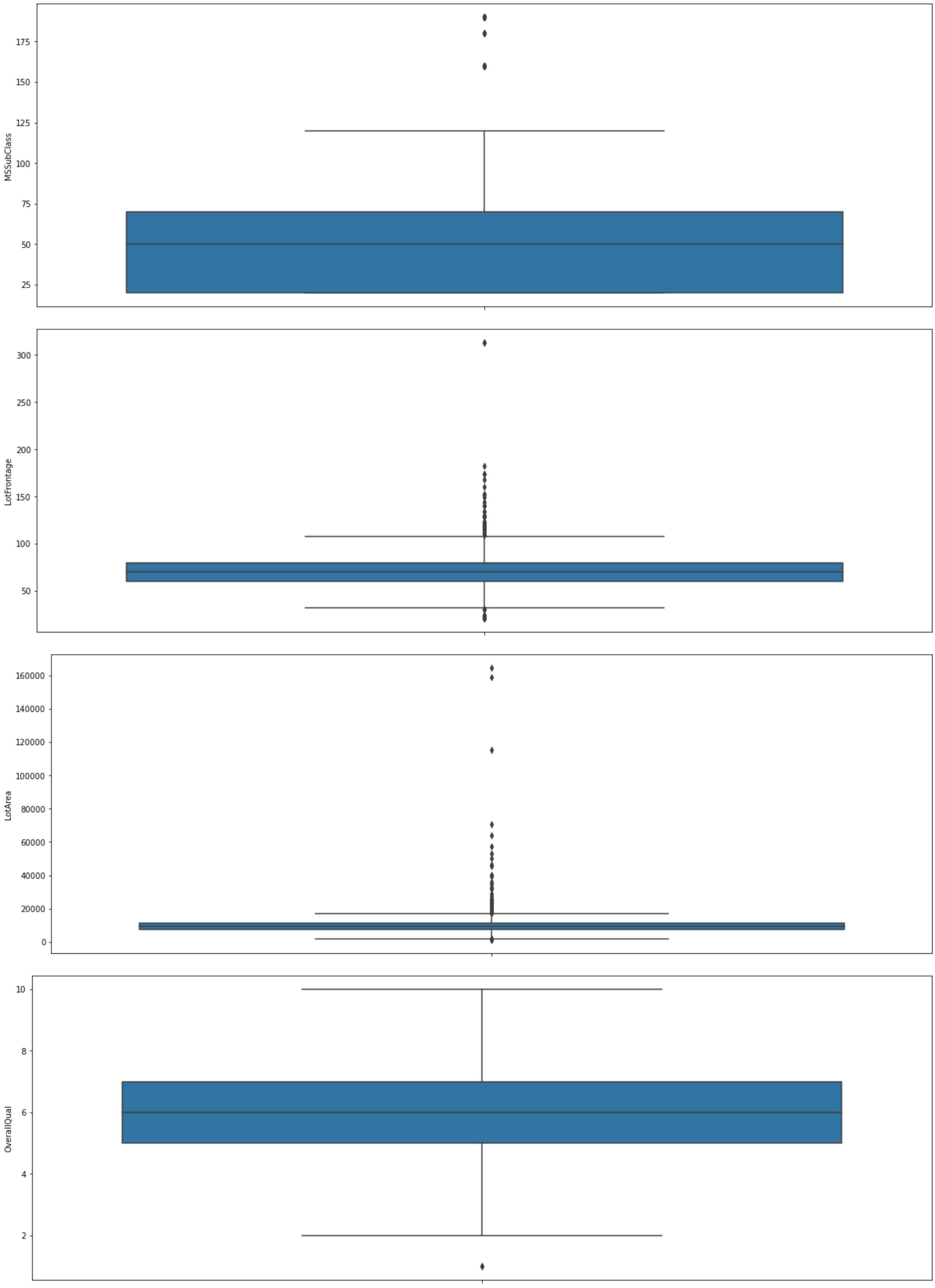
BsmtFinSF1 ->not normally distributed

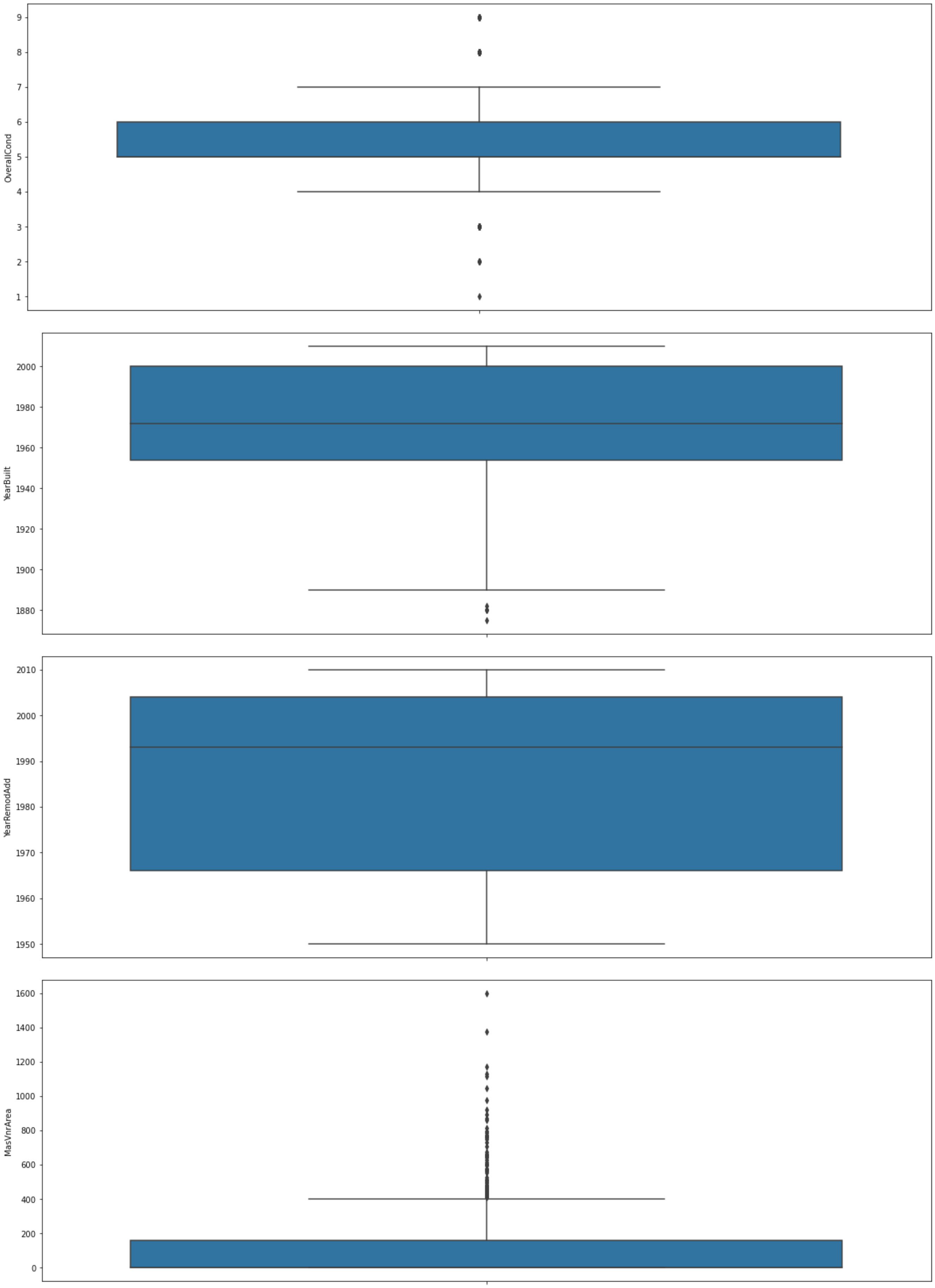
GaragerBlt ->not normally distributed

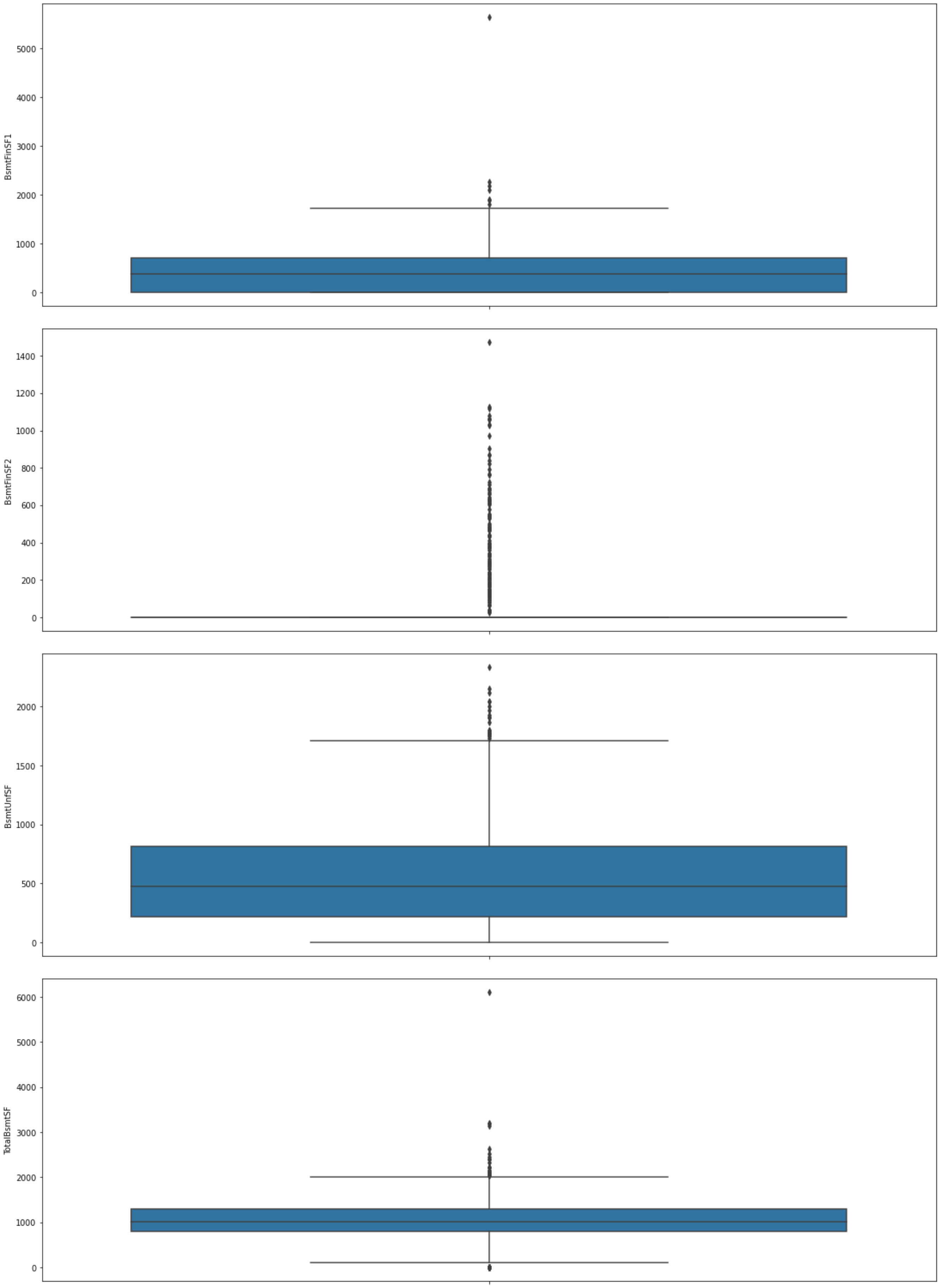
Garage Area -> not normally distributed

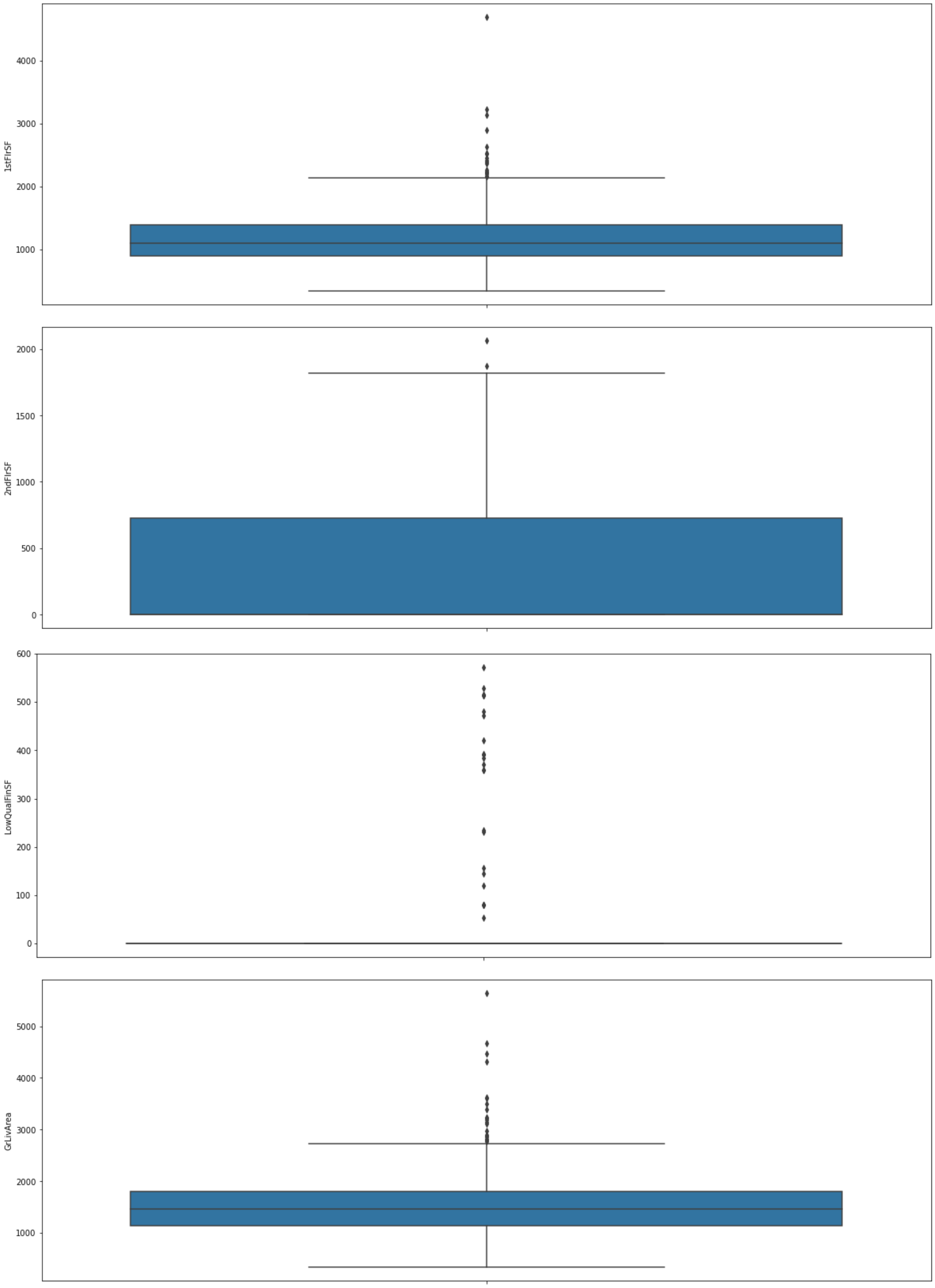


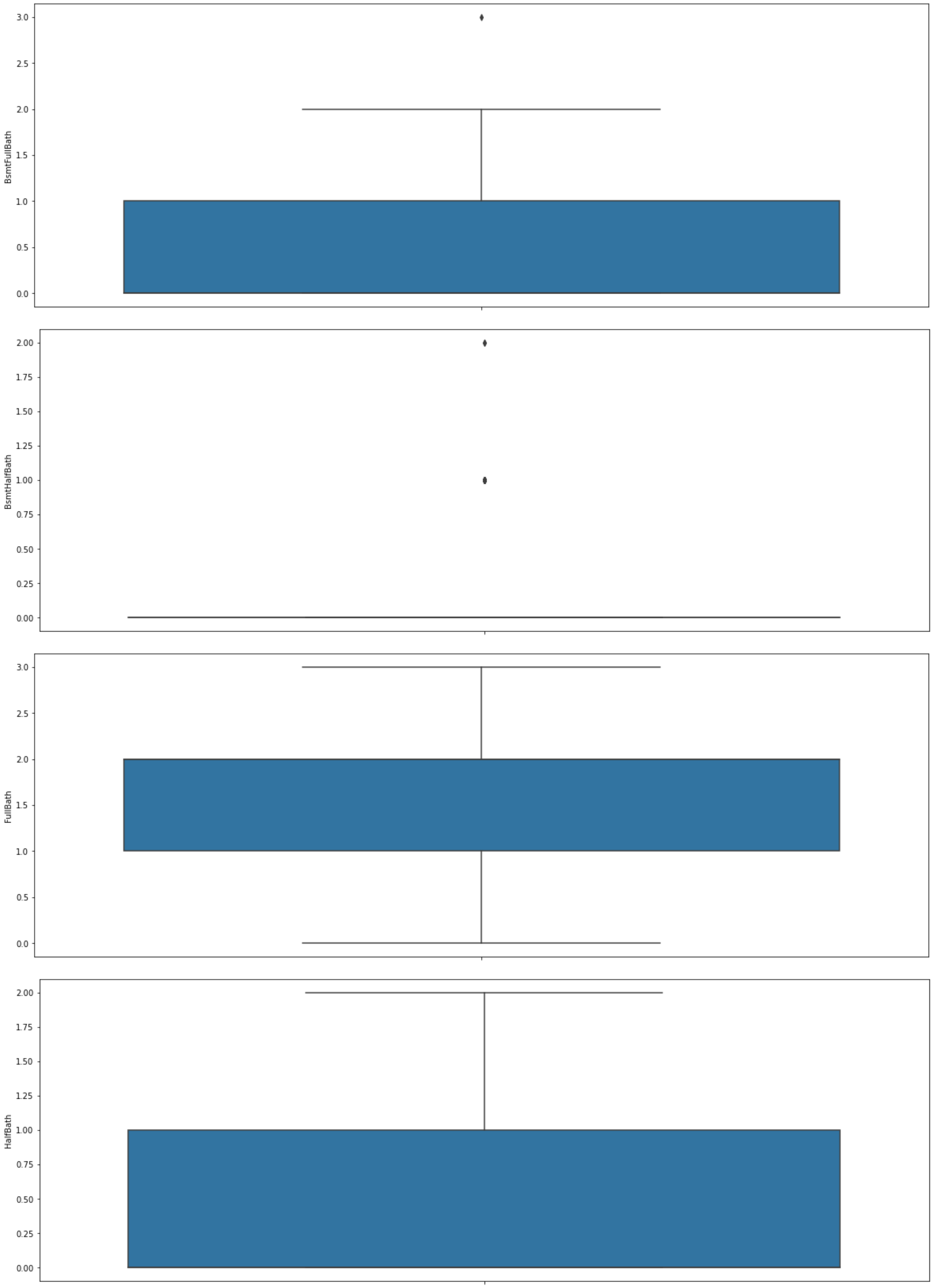
```
In [15]:  counter=1;
          for i in range(0,len(continous_columns)):
                  plt.figure(figsize=(20,500))
                  plt.subplot(60,1,counter)
                  counter=counter+1
                  sns.boxplot(y=continous_columns[i],hue = continous_columns[i],data=df)
                  #sns.boxplot(df[columns[i]])
                  plt.show()
```

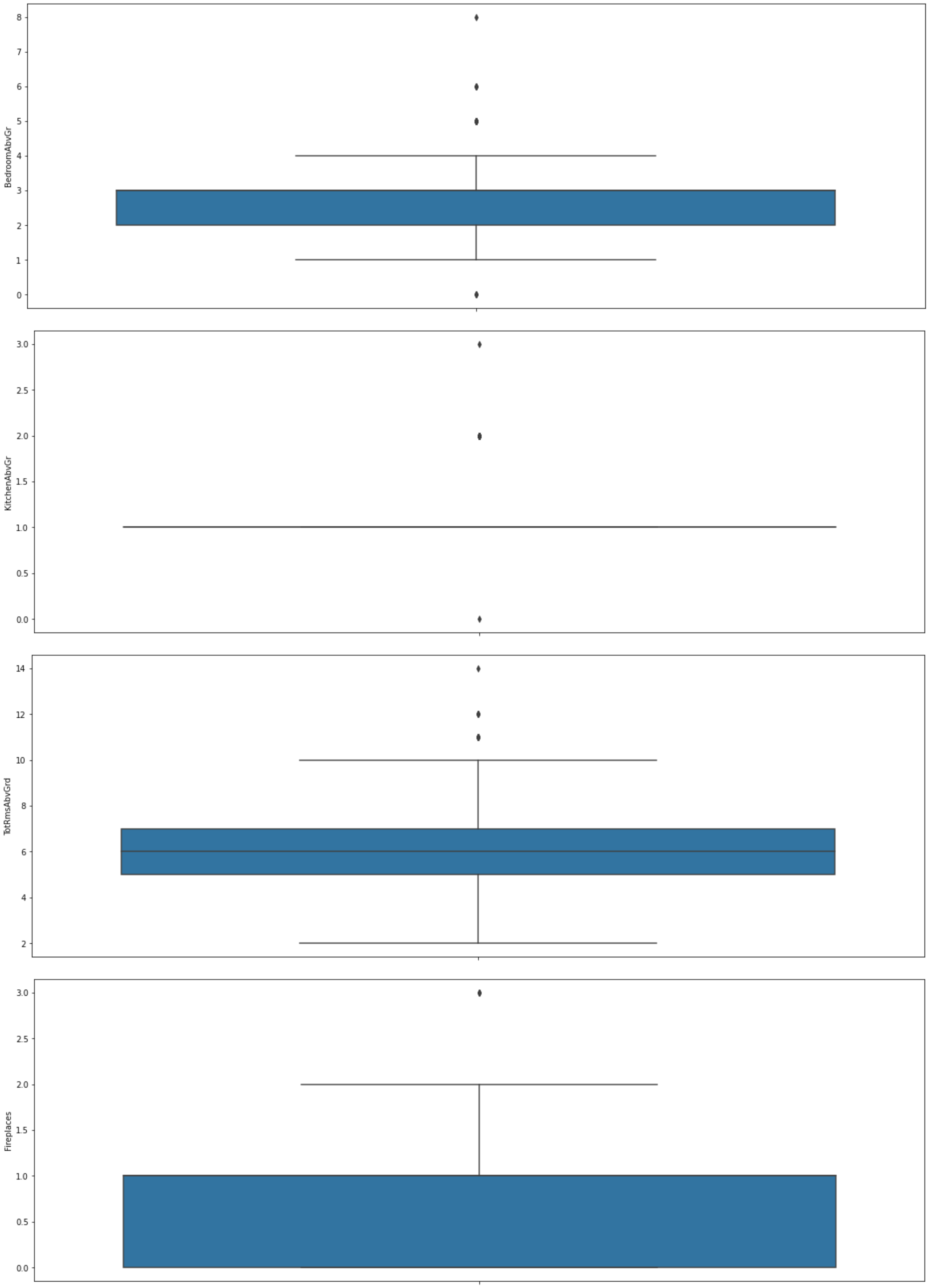

MSSubClass
25
50
75
100
125
150
175
LotFrontage
50
100
150
200
250
300
LotArea
0
20000
40000
60000
80000
100000
120000
140000
160000
OverallQual
2
4
6
8
10

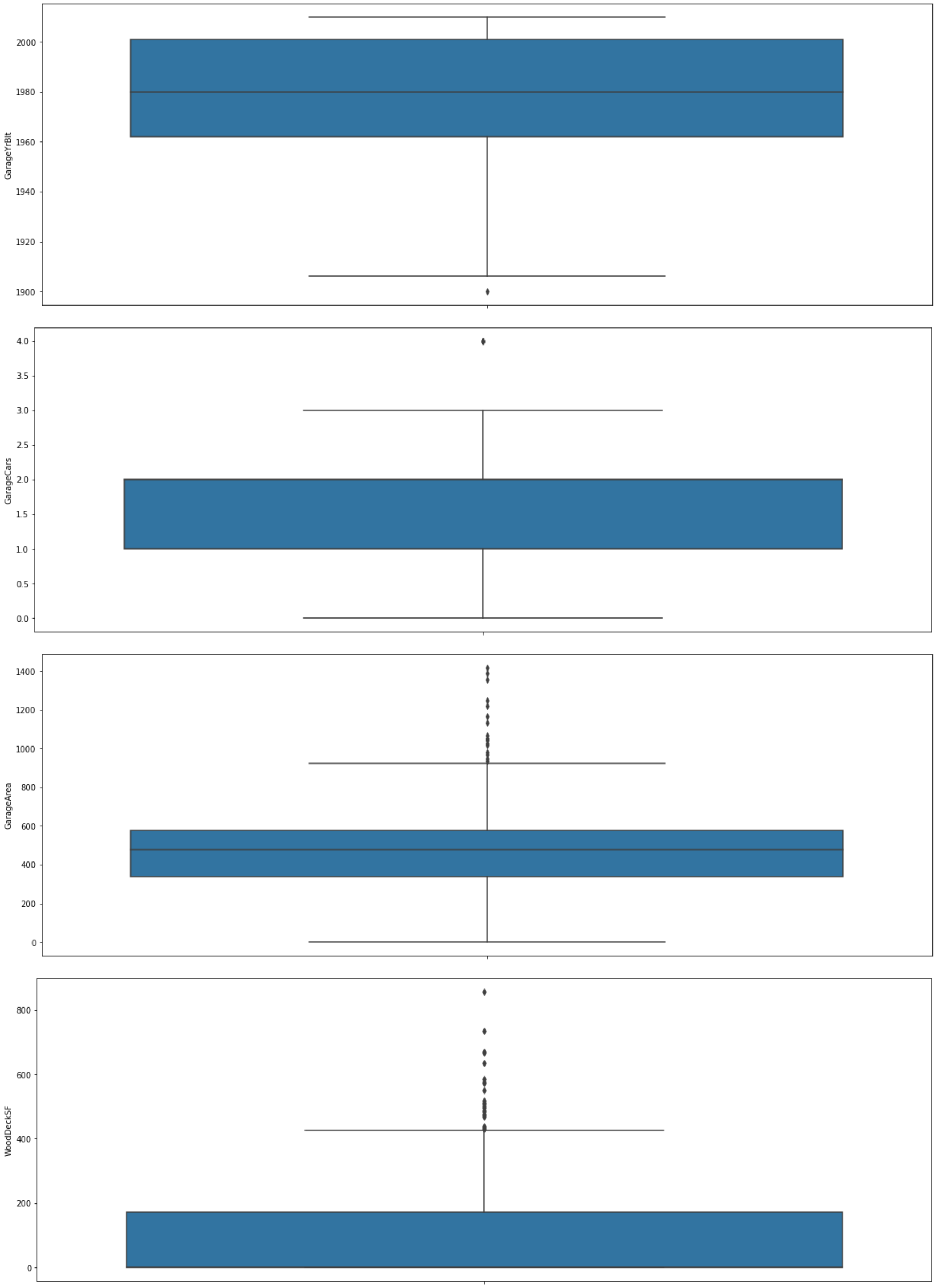

OverallCond
9
8
7
6
5
4
3
2
1
YearBuilt
2000
1980
1960
1940
1920
1900
1880
YearRemodAdd
2010
2000
1990
1980
1970
1960
1950
MasVnrArea
1600
1400
1200
1000
800
600
400
200
0

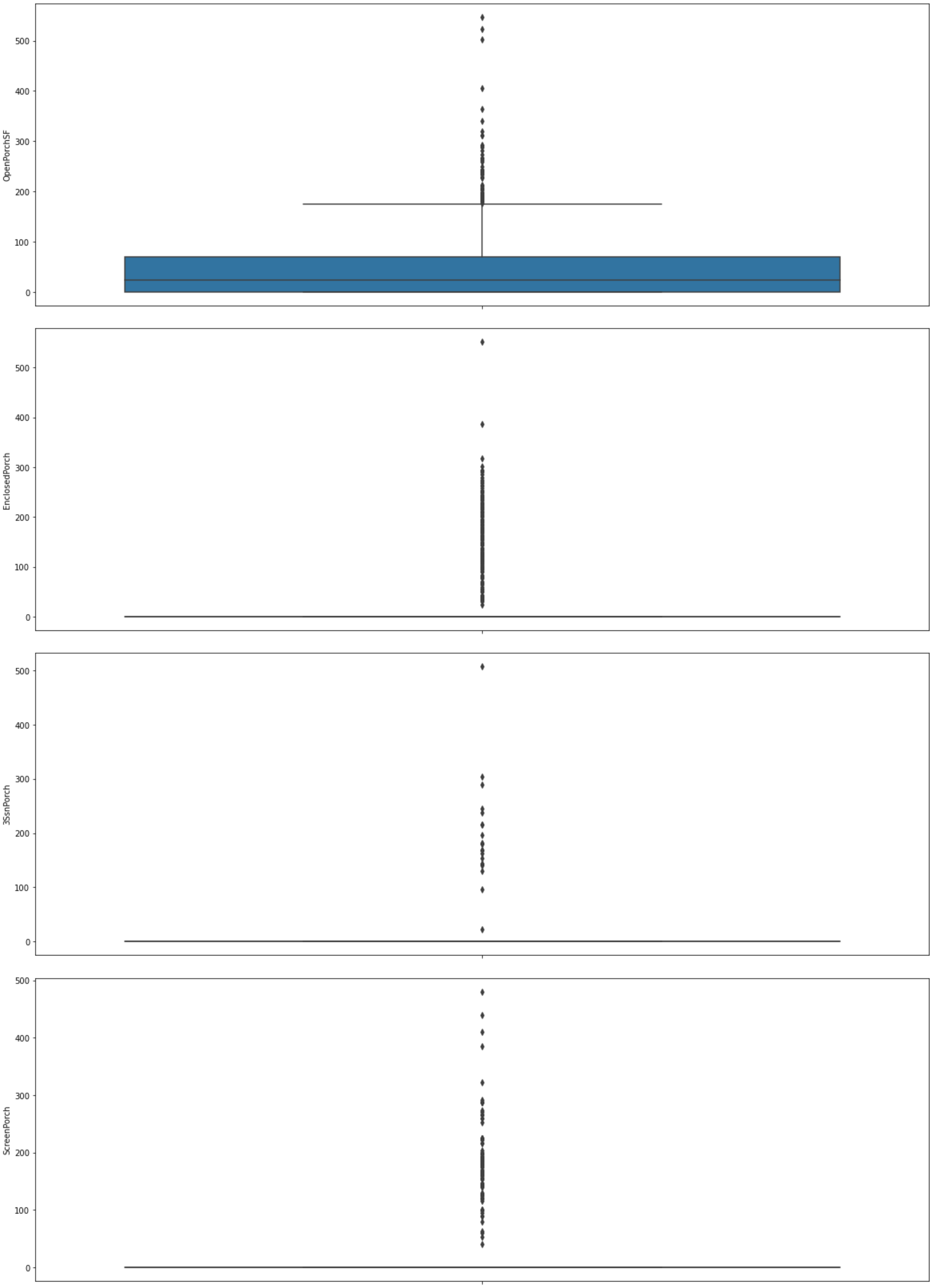

BsmtFinSF1
0
1000
2000
3000
4000
5000
BsmtFinSF2
0
200
400
600
800
1000
1200
1400
BsmtUnfSF
0
500
1000
1500
2000
TotalBsmtSF
0
1000
2000
3000
4000
5000
6000

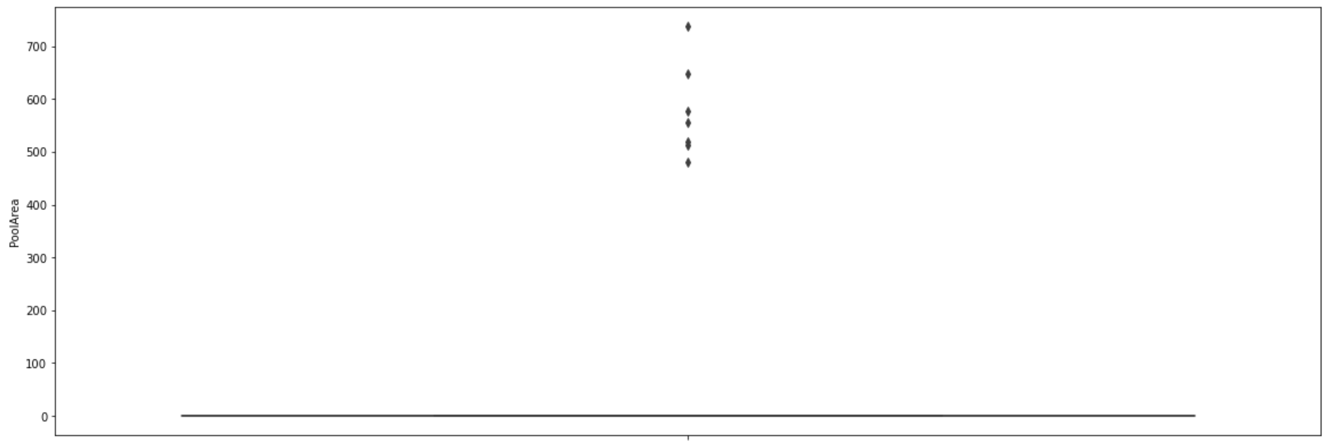

1stFlrSF
4000
3000
2000
1000
2ndFlrSF
2000
1500
1000
500
0
LowQualFinSF
600
500
400
300
200
100
0
GrLivArea
5000
4000
3000
2000
1000

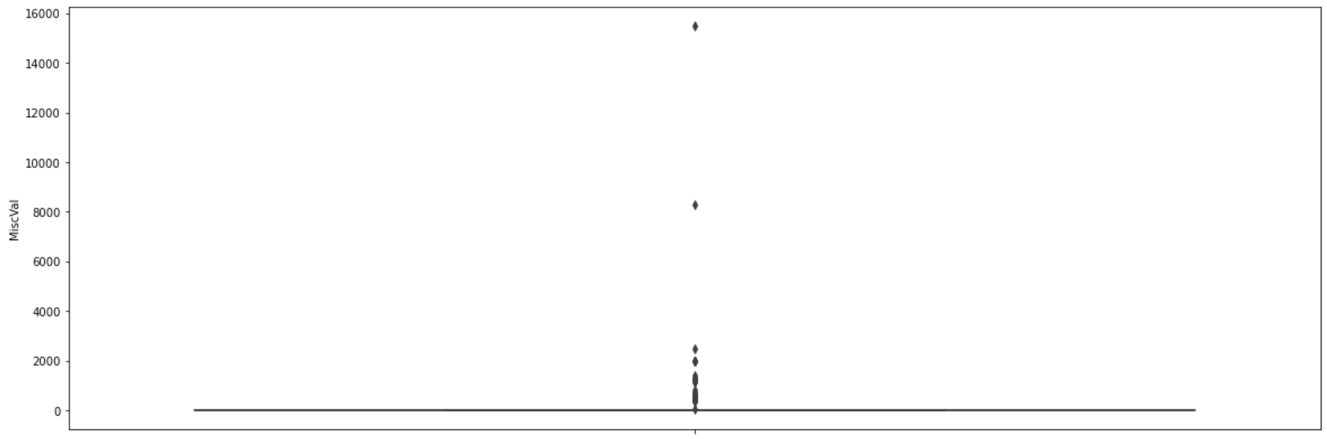

BsmtFullBath
3.0
2.5
2.0
1.5
1.0
0.5
0.0
BsmtHalfBath
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
FullBath
3.0
2.5
2.0
1.5
1.0
0.5
0.0
HalfBath
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

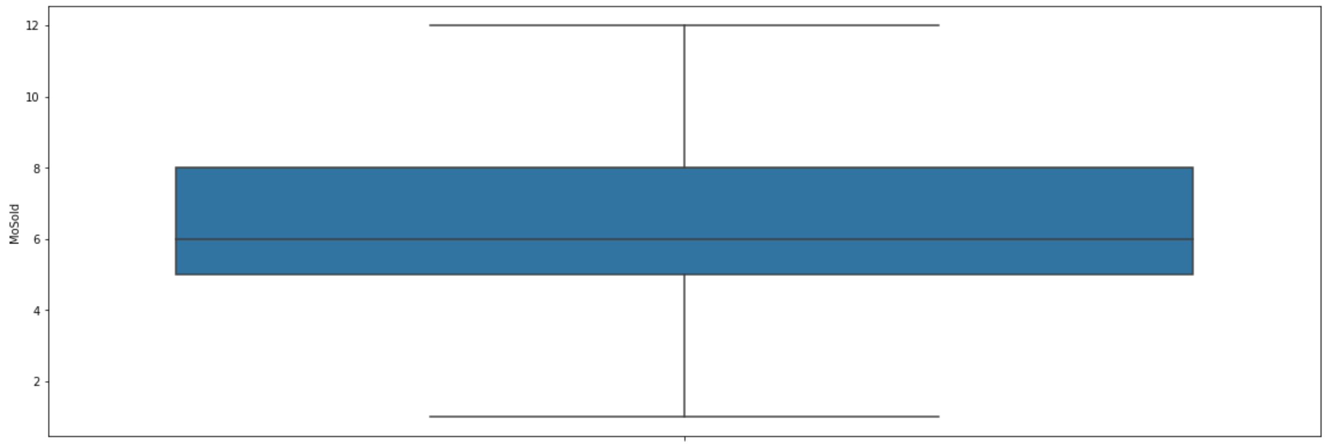

BedroomAbvGr
0
1
2
3
4
5
6
7
8
KitchenAbvGr
0.0
0.5
1.0
1.5
2.0
2.5
3.0
TotRmsAbvGrd
2
4
6
8
10
12
14
Fireplaces
0.0
0.5
1.0
1.5
2.0
2.5
3.0

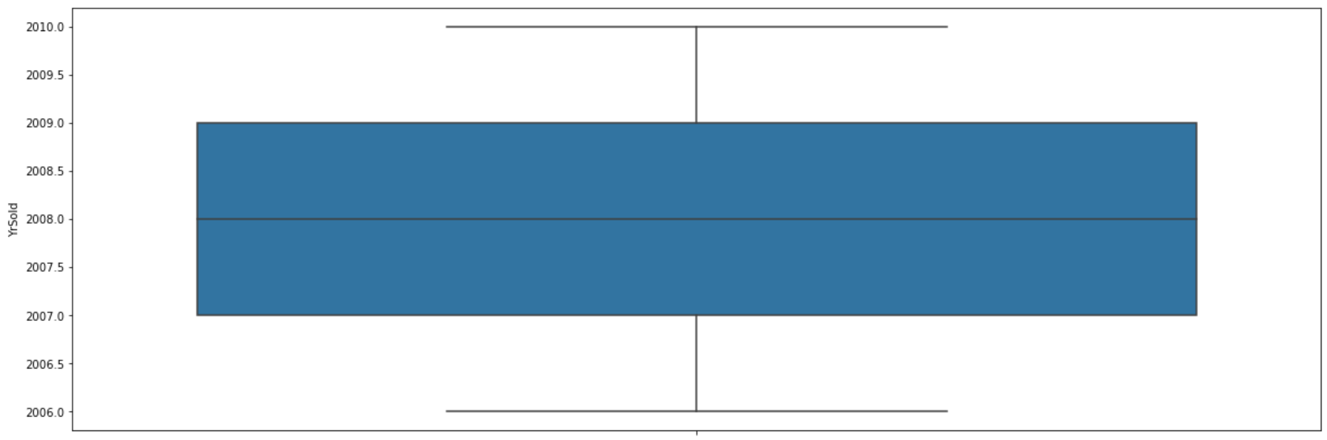

GarageYrBlt
2000
1980
1960
1940
1920
1900
GarageCars
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
GarageArea
1400
1200
1000
800
600
400
200
0
WoodDeckSF
800
600
400
200
0


OpenPorchSF
0
100
200
300
400
500
EnclosedPorch
0
100
200
300
400
500
3SsnPorch
0
100
200
300
400
500
ScreenPorch
0
100
200
300
400
500





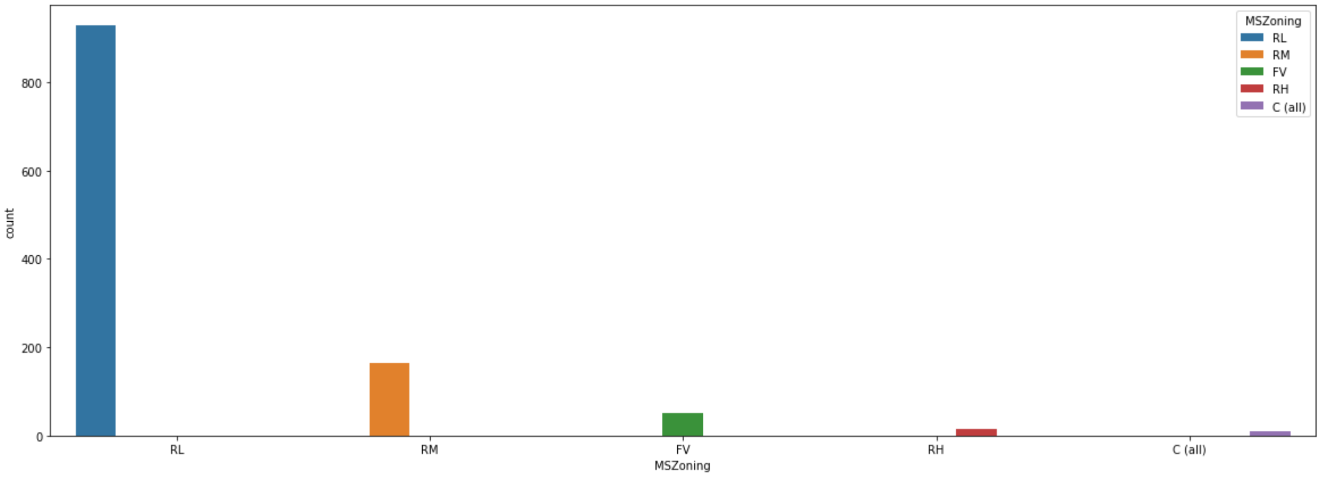


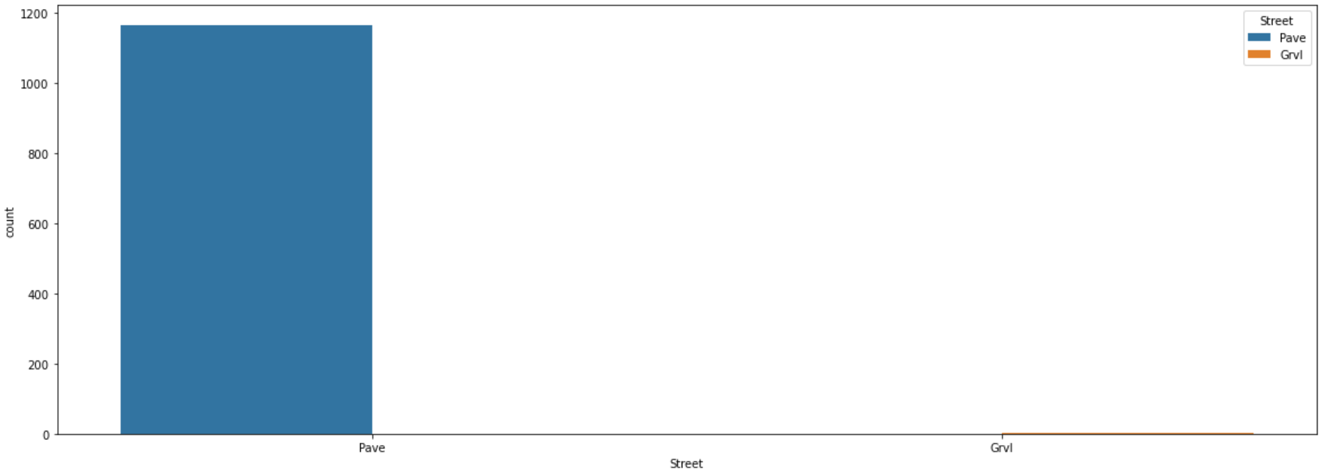


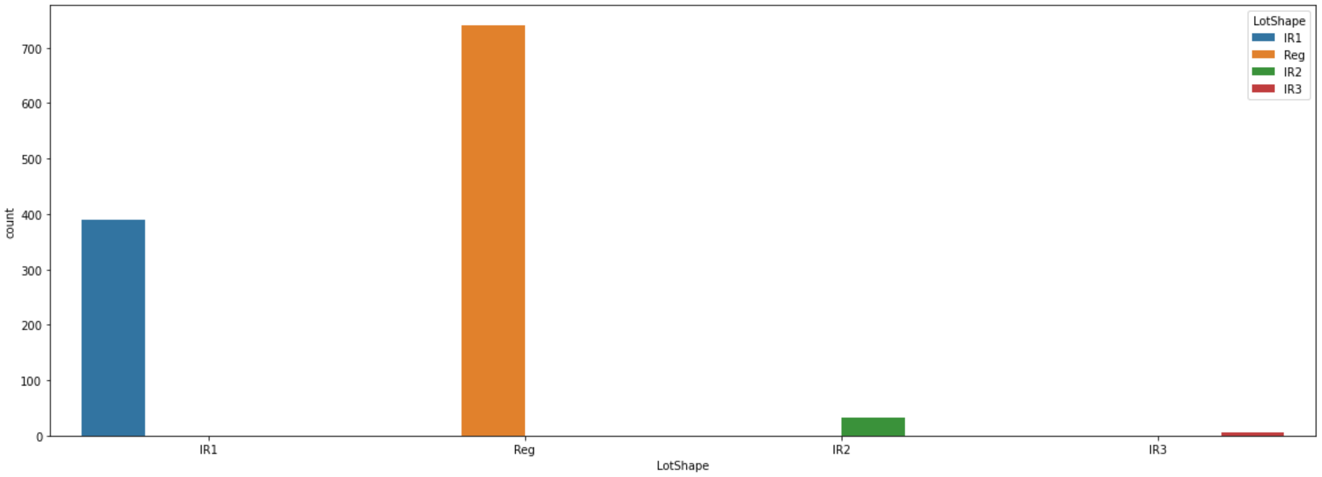
Findings:
columns { 'MSSubClass', 'LotFrontage', 'LotArea', 'Alley', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF',

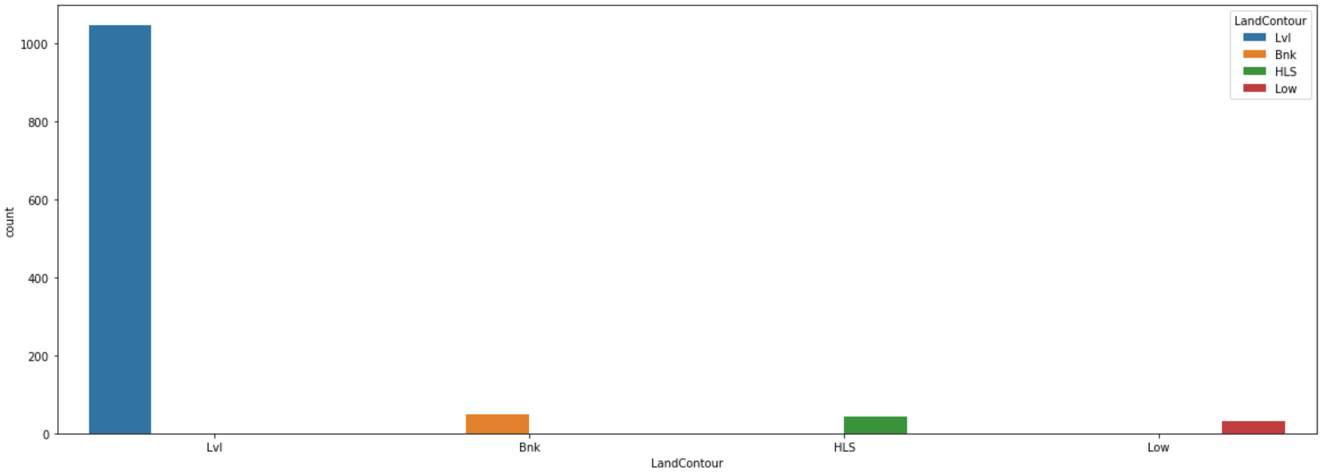
'2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'FireplaceQu', 'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal', 'SalePrice' } in the dataset have outliers present in them.

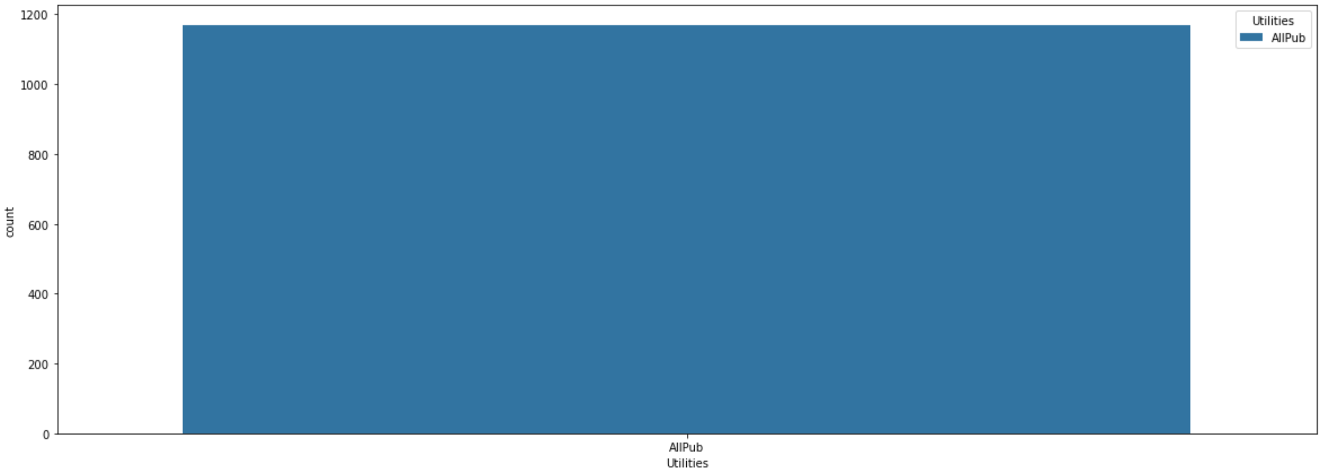
```
In [18]: counter=1;


         for column in categorical_columns:
                 plt.figure(figsize=(20,500))
                 plt.subplot(60,1,counter)
                 counter=counter+1
                 sns.countplot(x=column,hue=column,data=df)
                 plt.show()
```

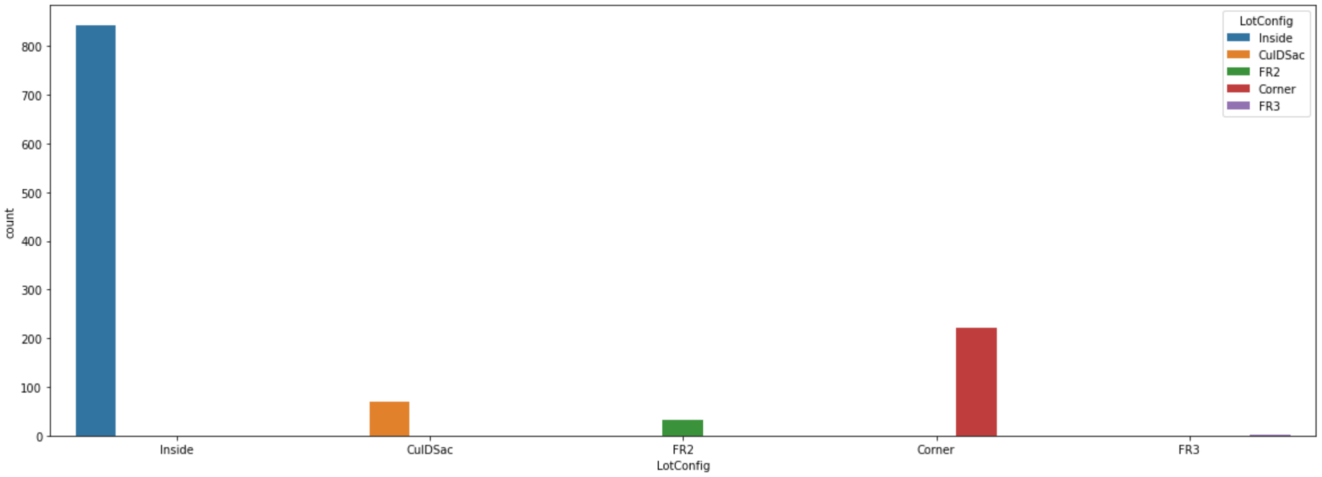


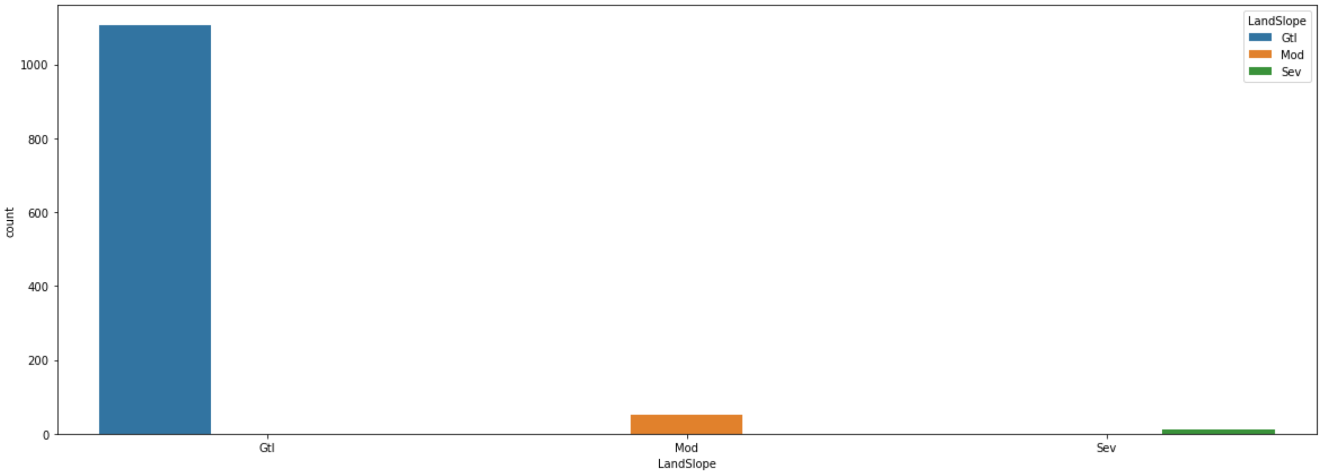


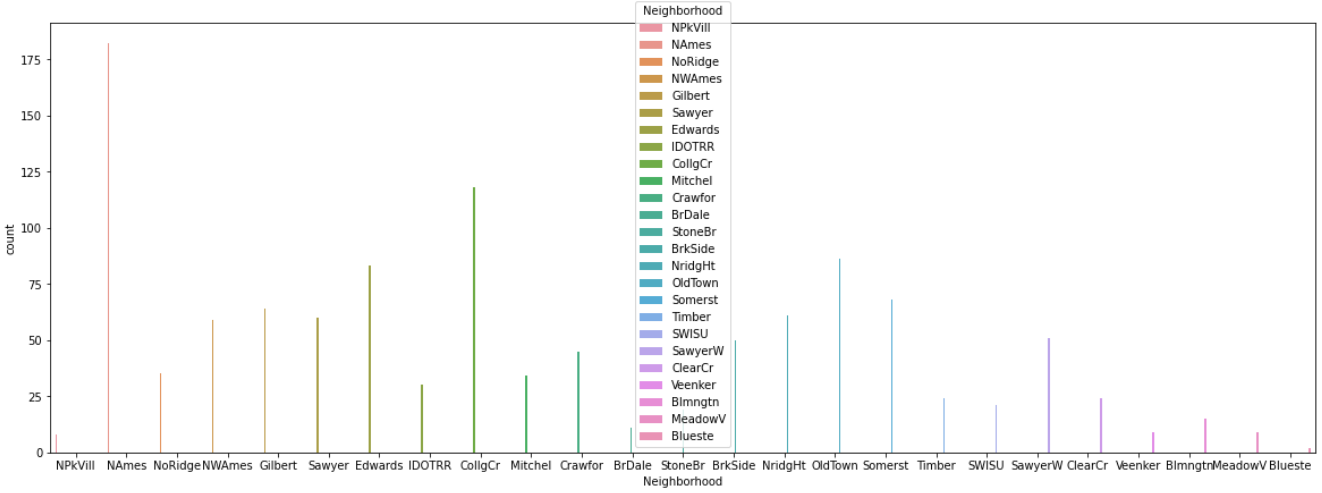


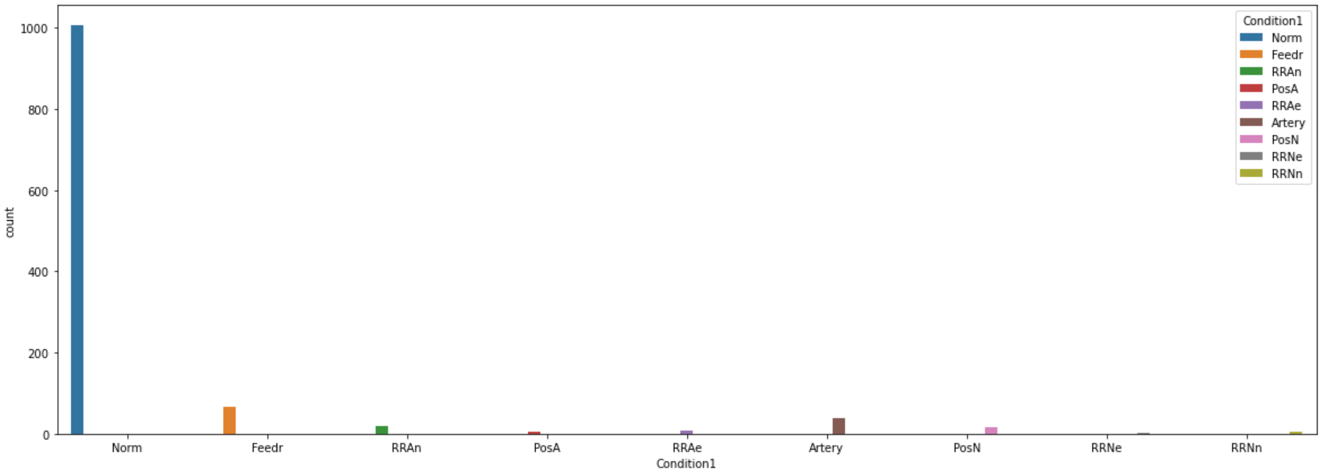

LandContour
Lvl
Bnk
HLS
Low
count
0
200
400
600
800
1000
Lvl
Bnk
HLS
Low
LandContour

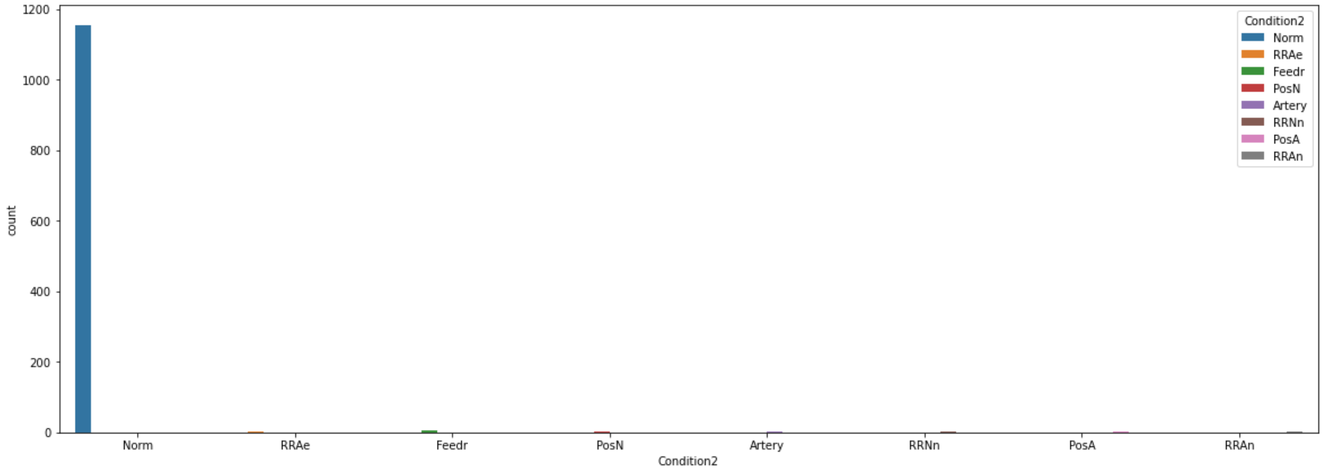

Utilities
AllPub
count
0
200
400
600
800
1000
1200
AllPub
Utilities

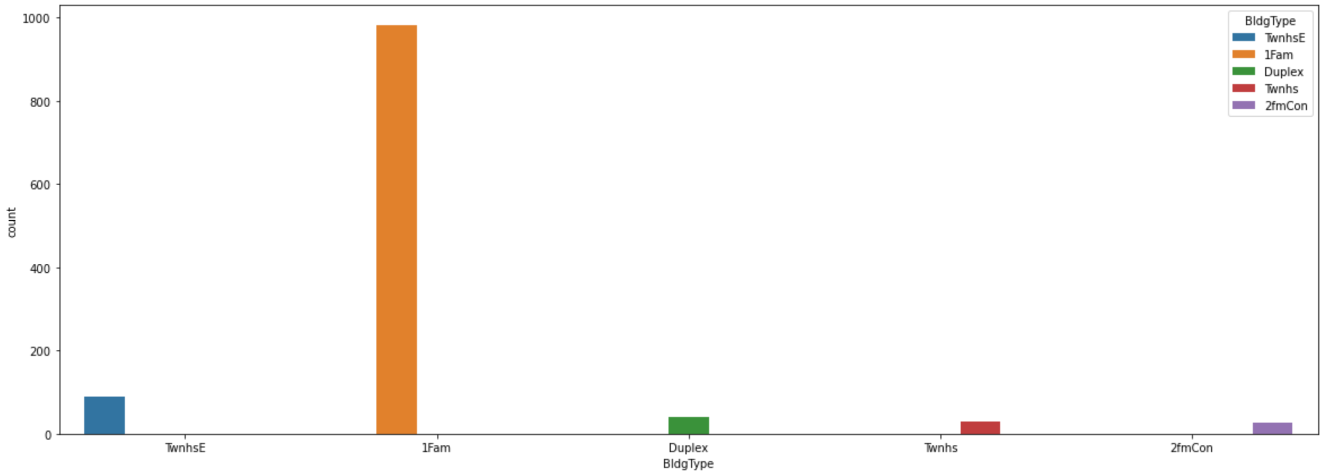

LotConfig
Inside
CulDSac
FR2
Corner
FR3
count
0
100
200
300
400
500
600
700
800
Inside
CulDSac
FR2
Corner
FR3
LotConfig

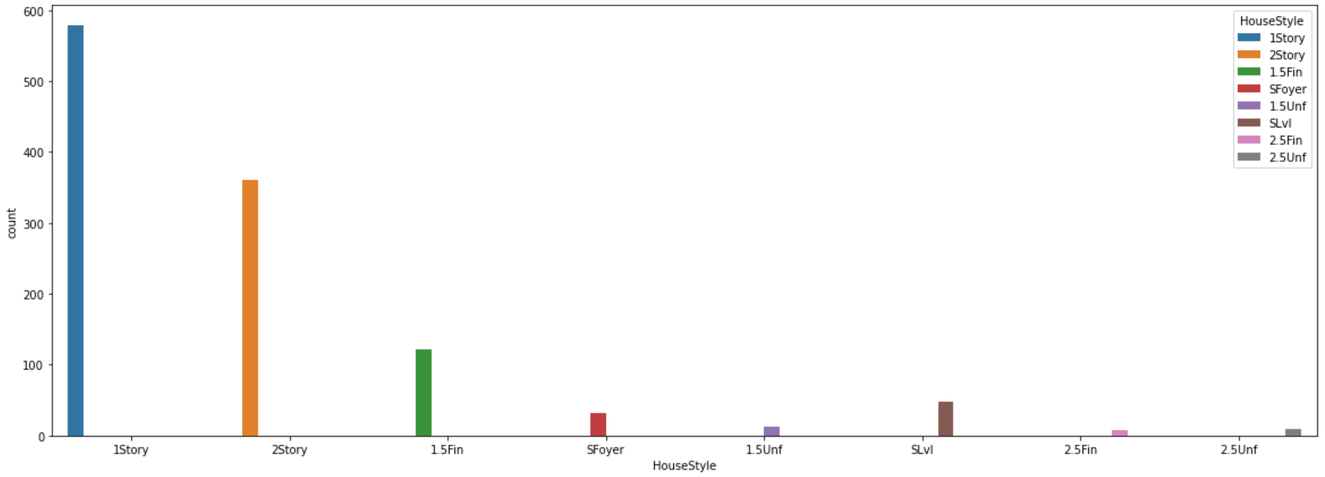

count
0
200
400
600
800
1000
Gtl
Mod
Sev
LandSlope
LandSlope
Gtl
Mod
Sev

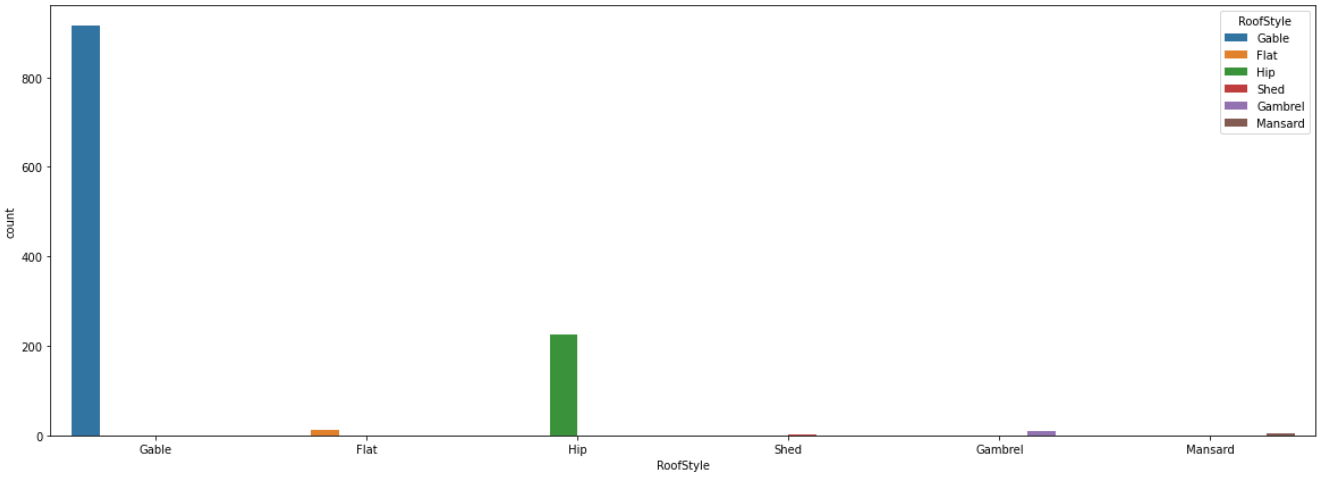

count
0
25
50
75
100
125
150
175
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
Sawyer
Edwards
IDOTRR
CollgCr
Mitchel
Crawfor
BrDale
StoneBr
BrkSide
NridgHt
OldTown
Somerst
Timber
SWISU
SawyerW
ClearCr
Veenker
Blmngtn
MeadowV
Blueste
Neighborhood
Neighborhood
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
Sawyer
Edwards
IDOTRR
CollgCr
Mitchel
Crawfor
BrDale
StoneBr
BrkSide
NridgHt
OldTown
Somerst
Timber
SWISU
SawyerW
ClearCr
Veenker
Blmngtn
MeadowV
Blueste

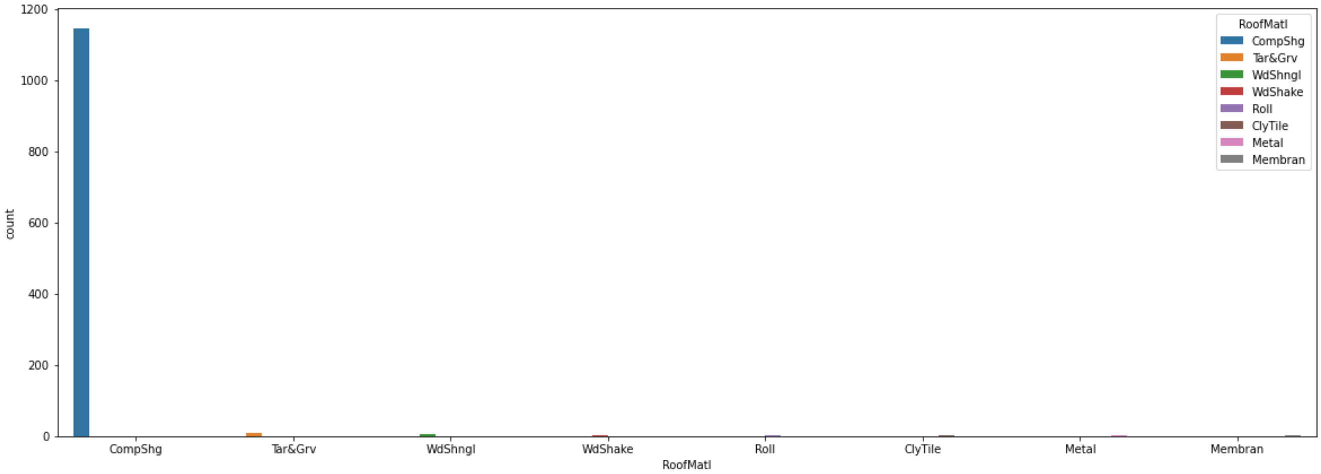

count
0
200
400
600
800
1000
Norm
Feedr
RRAn
PosA
RRAe
Artery
PosN
RRNe
RRNn
Condition1
Condition1
Norm
Feedr
RRAn
PosA
RRAe
Artery
PosN
RRNe
RRNn

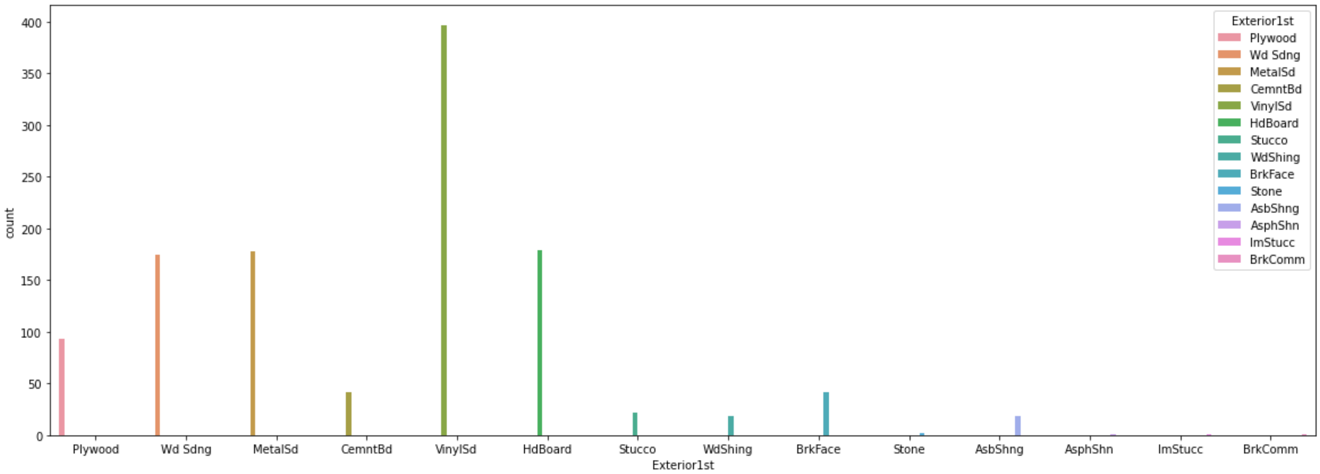

1200
1000
800
600
400
200
0
count
Norm
RRAe
Feedr
PosN
Artery
RRNn
PosA
RRAn
Condition2
Condition2
Norm
RRAe
Feedr
PosN
Artery
RRNn
PosA
RRAn

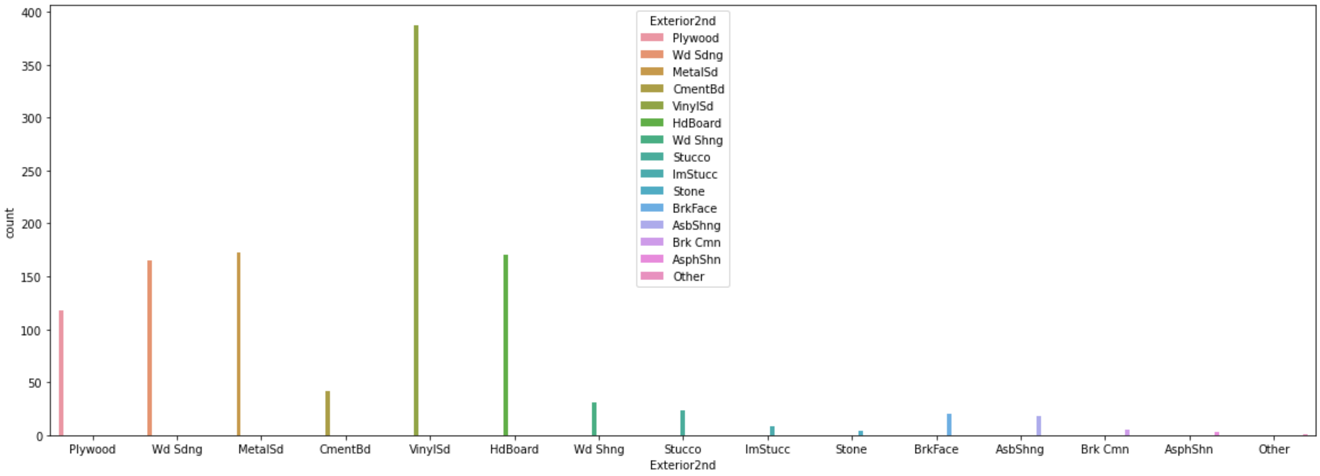

1000
800
600
400
200
0
count
TwnhsE
1Fam
Duplex
Twnhs
2fmCon
BldgType
BldgType
TwnhsE
1Fam
Duplex
Twnhs
2fmCon


600
500
400
300
200
100
0
count
1Story
2Story
1.5Fin
SFoyer
1.5Unf
SLvl
2.5Fin
2.5Unf
HouseStyle
HouseStyle
1Story
2Story
1.5Fin
SFoyer
1.5Unf
SLvl
2.5Fin
2.5Unf

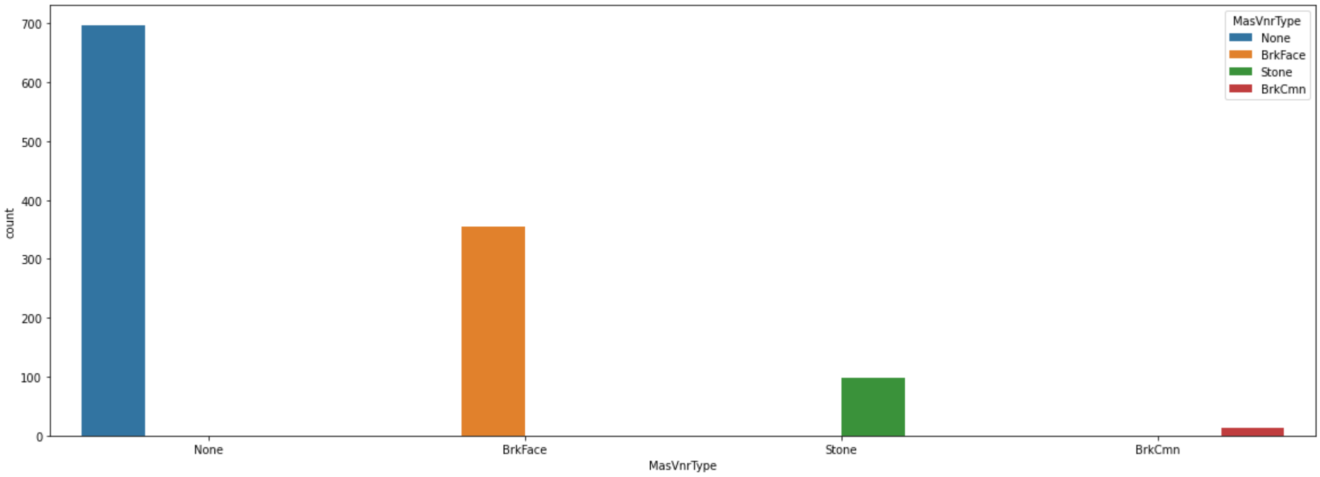

RoofStyle
Gable
Flat
Hip
Shed
Gambrel
Mansard
800
600
400
200
0
count
Gable
Flat
Hip
Shed
Gambrel
Mansard
RoofStyle


RoofMatl
CompShg
Tar&Grv
WdShngl
WdShake
Roll
ClyTile
Metal
Membran
1200
1000
800
600
400
200
0
count
CompShg
Tar&Grv
WdShngl
WdShake
Roll
ClyTile
Metal
Membran
RoofMatl

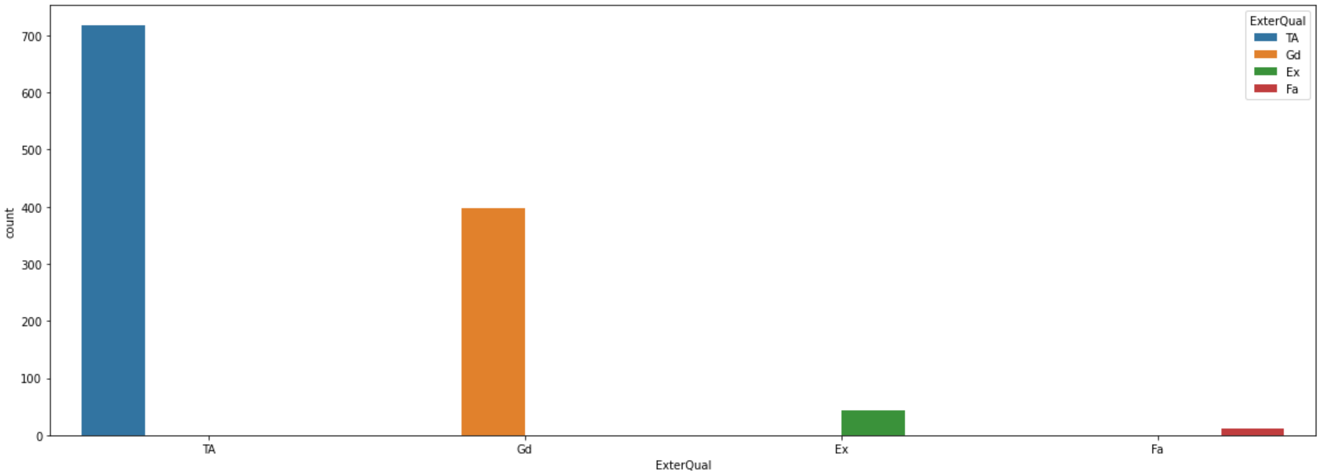

Exterior1st
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
400
350
300
250
200
150
100
50
0
count
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
Exterior1st



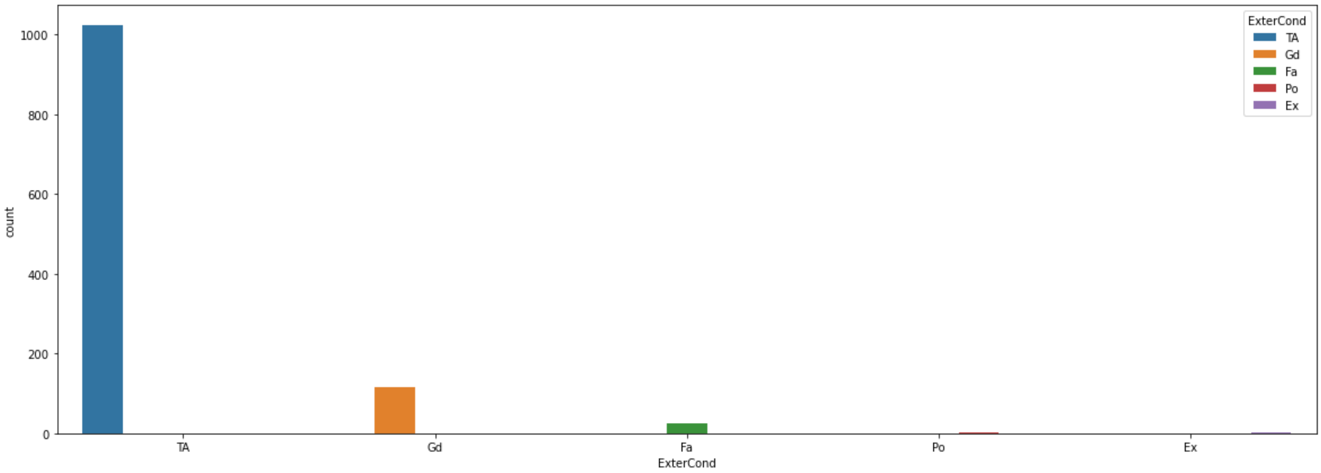
count
400
350
300
250
200
150
100
50
0
Exterior2nd
Plywood
Wd Sdng
MetalSd
CmentBd
VinylSd
HdBoard
Wd Shng
Stucco
ImStucc
Stone
BrkFace
AsbShng
Brk Cmn
AsphShn
Other
Exterior2nd

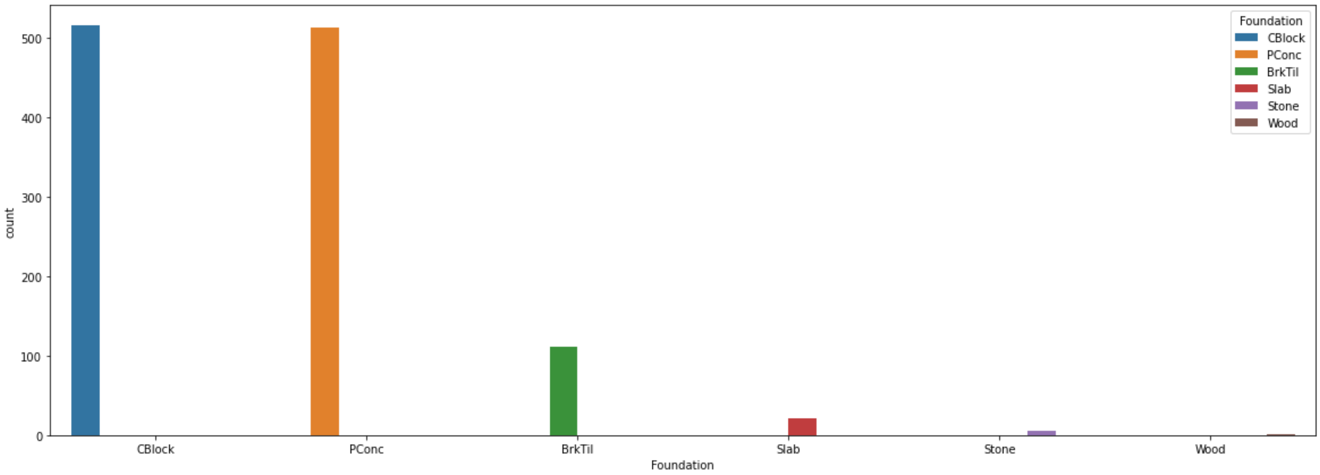


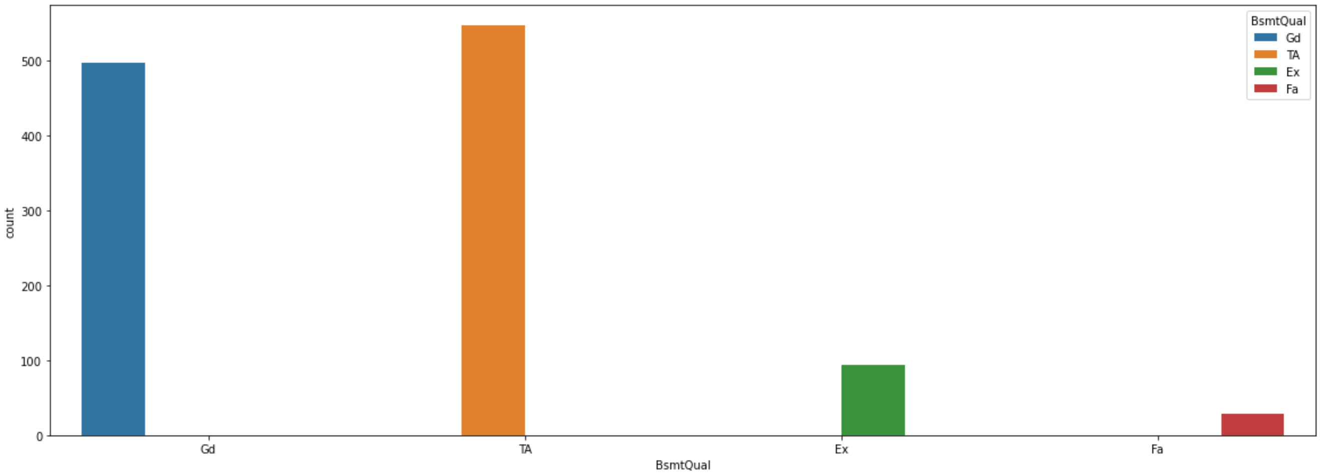
count
700
600
500
400
300
200
100
0
MasVnrType
None
BrkFace
Stone
BrkCmn
MasVnrType

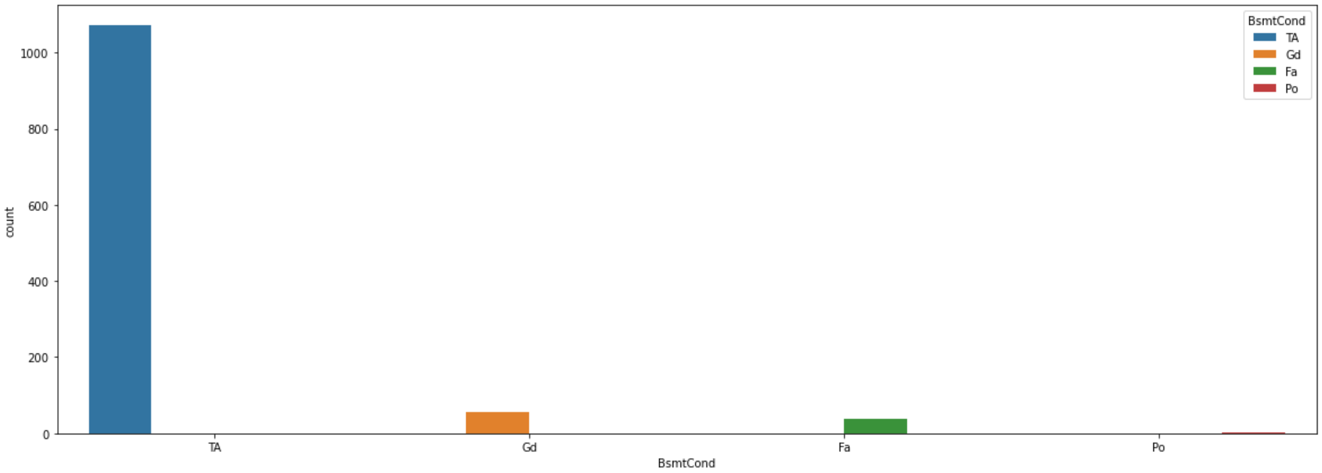


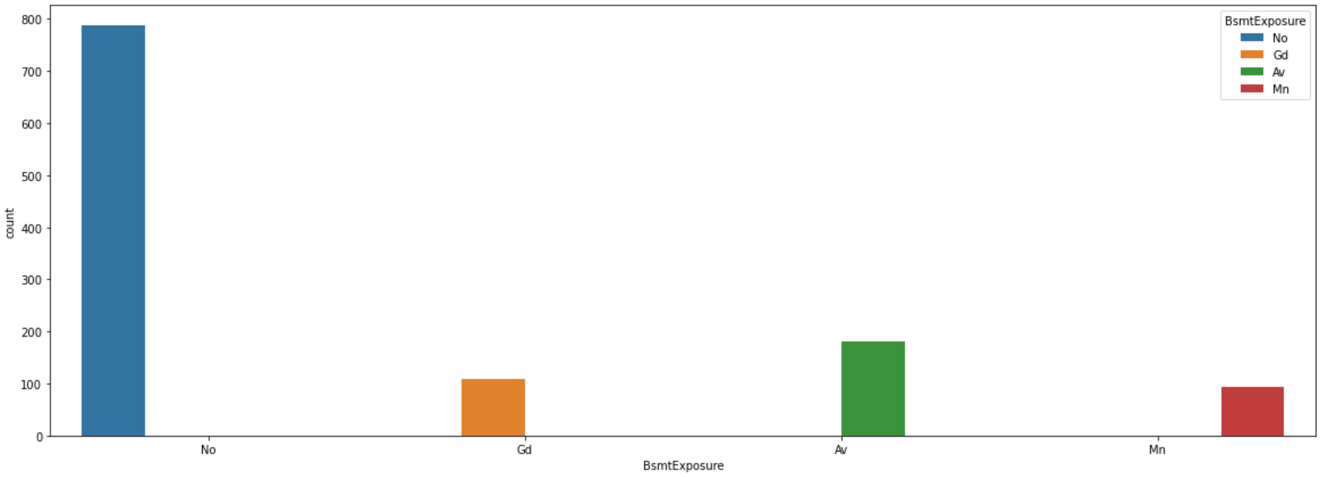
count
700
600
500
400
300
200
100
0
ExterQual
TA
Gd
Ex
Fa
ExterQual

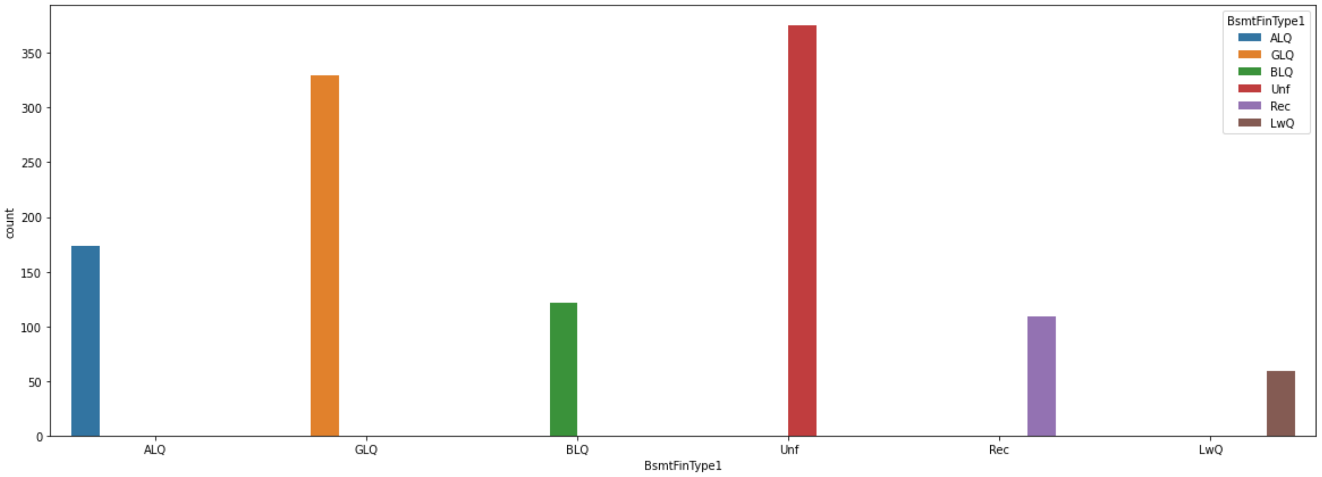

count
1000
800
600
400
200
0
TA
Gd
Fa
Po
Ex
ExterCond
ExterCond
TA
Gd
Fa
Po
Ex

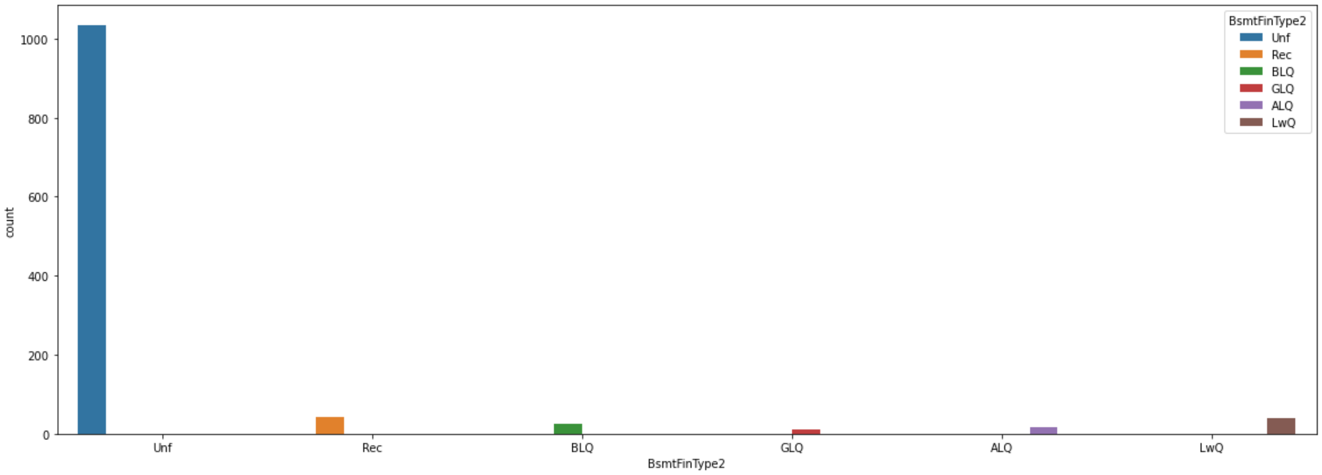

count
500
400
300
200
100
0
CBlock
PConc
BrkTil
Slab
Stone
Wood
Foundation
Foundation
CBlock
PConc
BrkTil
Slab
Stone
Wood

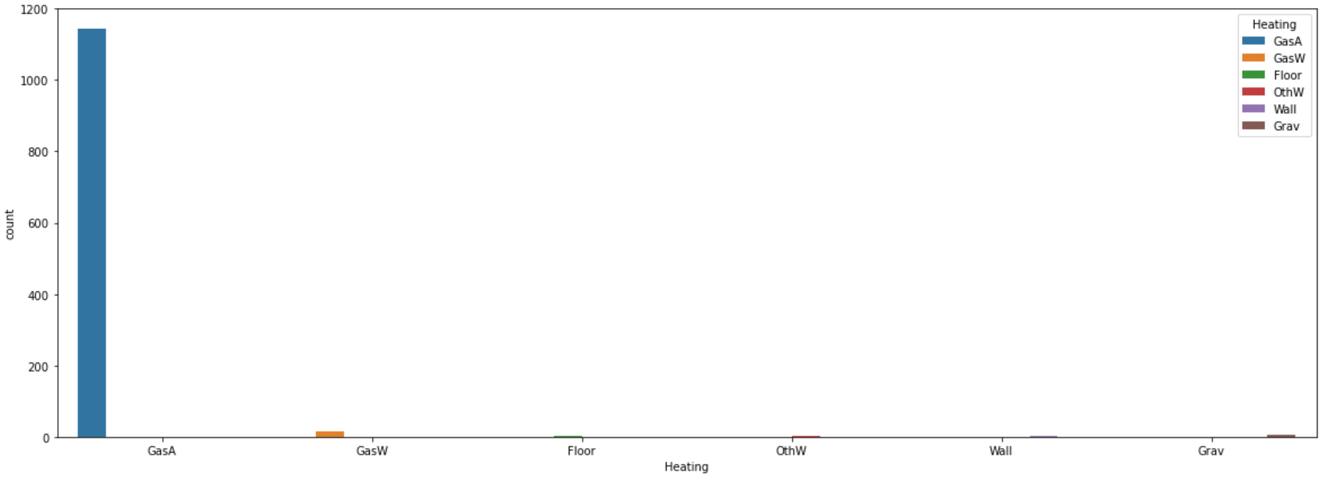

count
500
400
300
200
100
0
Gd
TA
Ex
Fa
BsmtQual
BsmtQual
Gd
TA
Ex
Fa

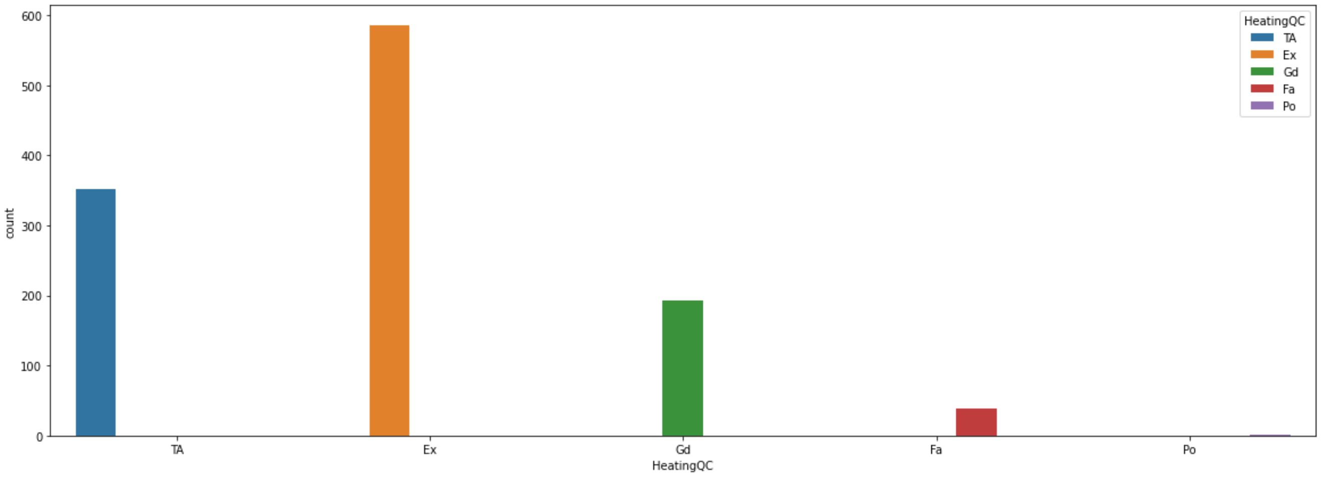

count
0
200
400
600
800
1000
TA
Gd
Fa
Po
BsmtCond
BsmtCond
TA
Gd
Fa
Po

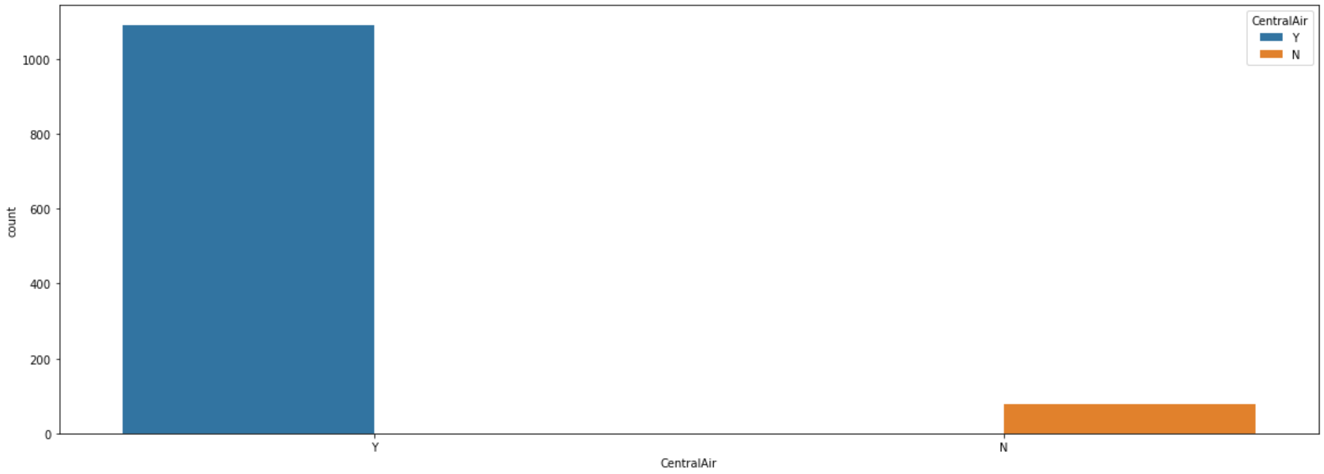

count
0
100
200
300
400
500
600
700
800
No
Gd
Av
Mn
BsmtExposure
BsmtExposure
No
Gd
Av
Mn

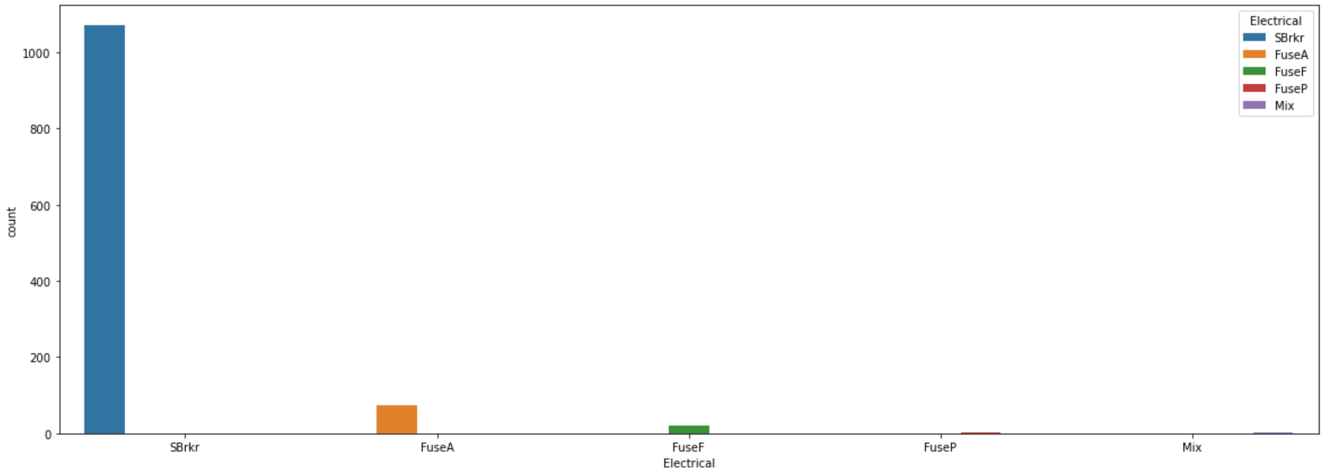

count
0
50
100
150
200
250
300
350
ALQ
GLQ
BLQ
Unf
Rec
LwQ
BsmtFinType1
BsmtFinType1
ALQ
GLQ
BLQ
Unf
Rec
LwQ

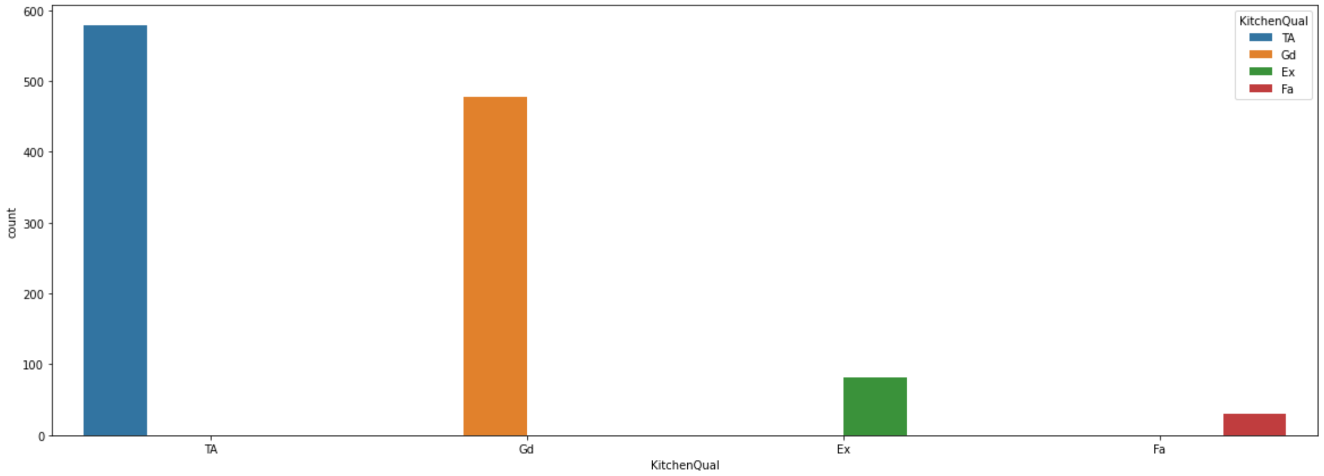

count
1000
800
600
400
200
0
Unf
Rec
BLQ
GLQ
ALQ
LwQ
BsmtFinType2
BsmtFinType2
Unf
Rec
BLQ
GLQ
ALQ
LwQ

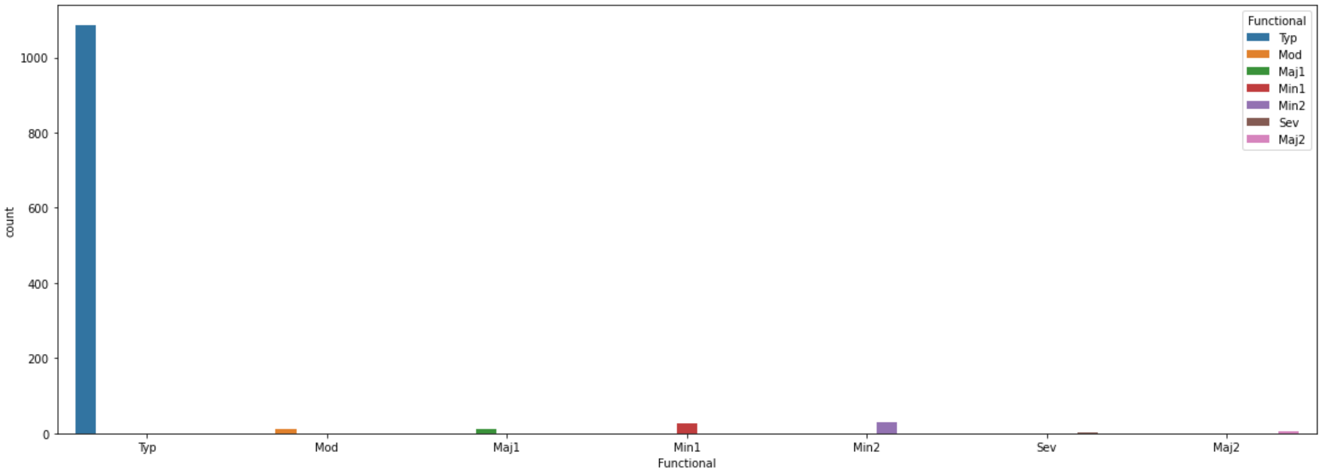

count
1200
1000
800
600
400
200
0
GasA
GasW
Floor
OthW
Wall
Grav
Heating
Heating
GasA
GasW
Floor
OthW
Wall
Grav

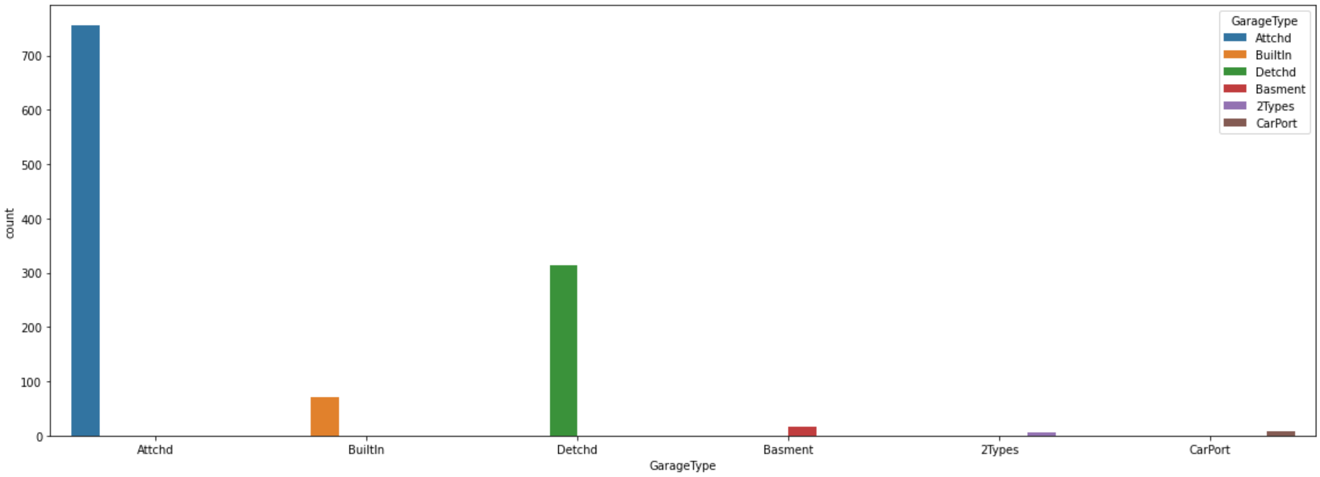

count
600
500
400
300
200
100
0
TA
Ex
Gd
Fa
Po
HeatingQC
HeatingQC
TA
Ex
Gd
Fa
Po

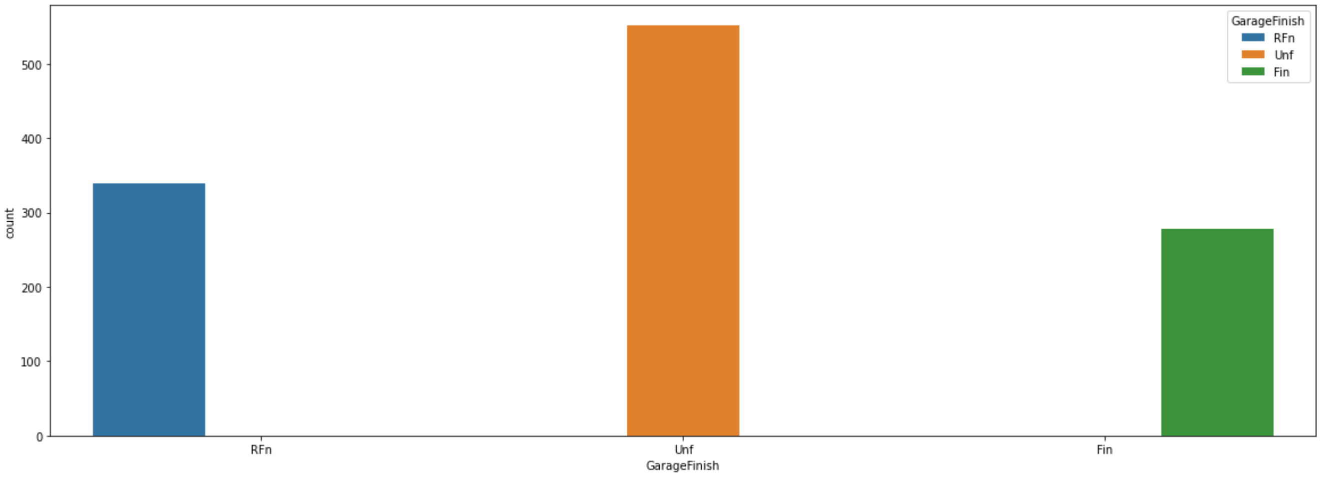

CentralAir
Y
N
count
0
200
400
600
800
1000
Y
N
CentralAir

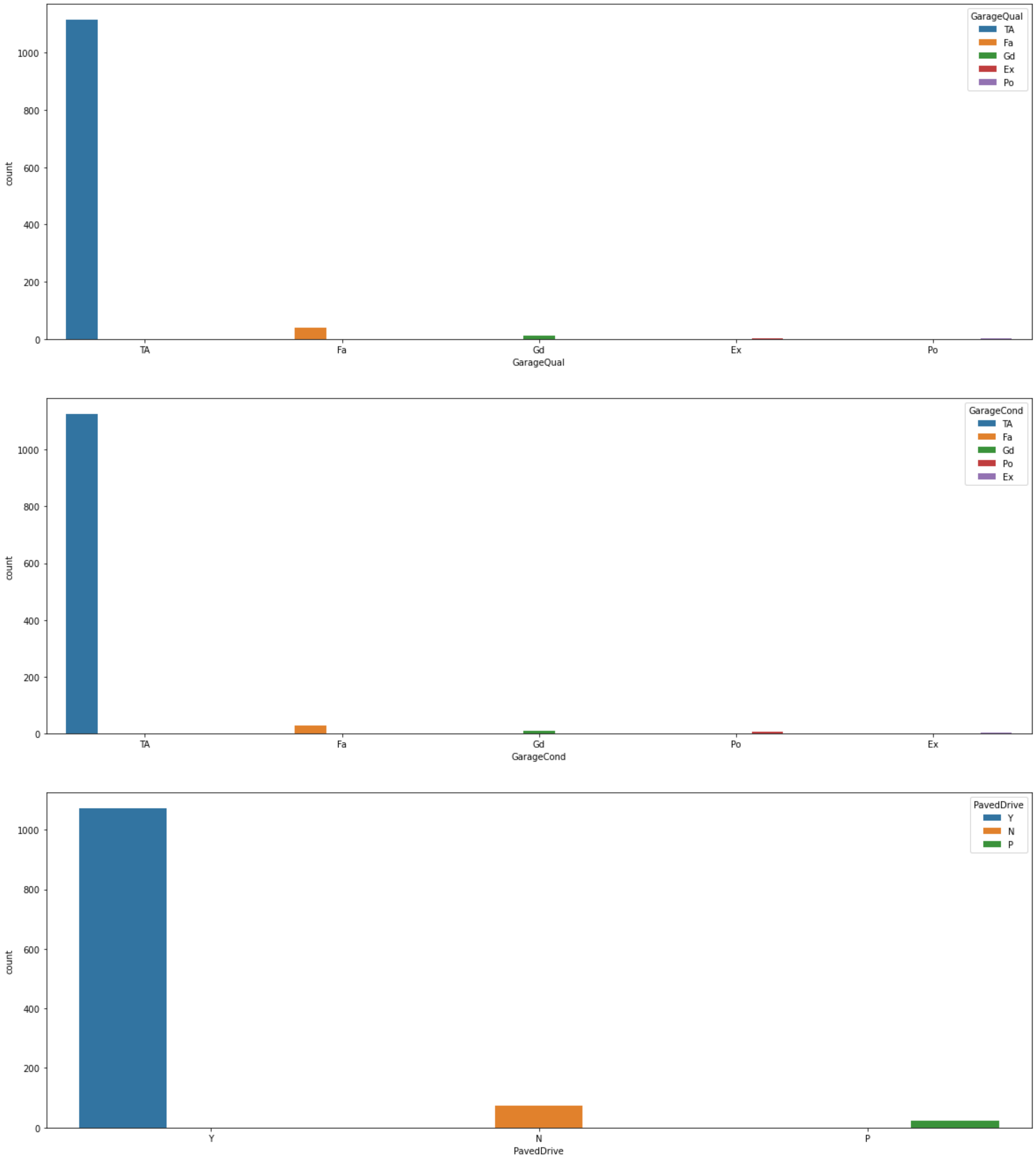

Electrical
SBrkr
FuseA
FuseF
FuseP
Mix
count
0
200
400
600
800
1000
SBrkr
FuseA
FuseF
FuseP
Mix
Electrical


KitchenQual
TA
Gd
Ex
Fa
count
0
100
200
300
400
500
600
TA
Gd
Ex
Fa
KitchenQual

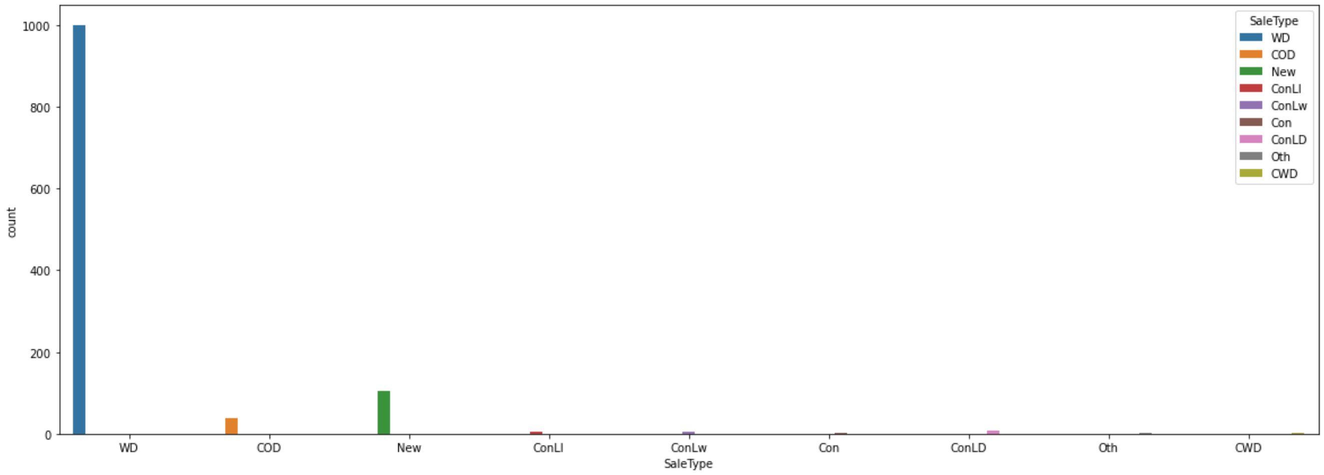

count
0
200
400
600
800
1000
Typ
Mod
Maj1
Min1
Min2
Sev
Maj2
Functional
Functional
Typ
Mod
Maj1
Min1
Min2
Sev
Maj2

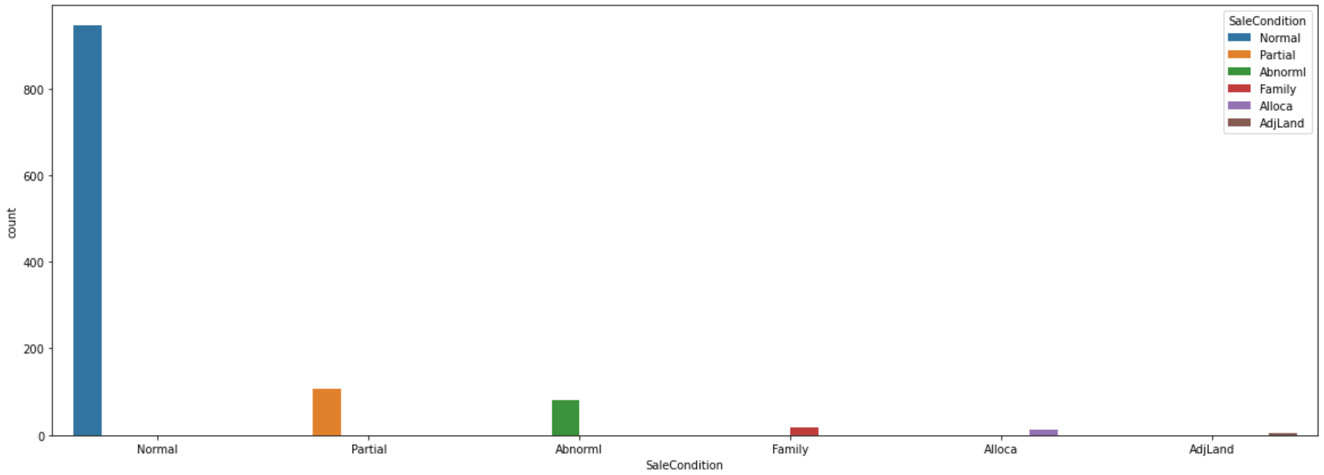

count
0
100
200
300
400
500
600
700
Attchd
BuiltIn
Detchd
Basment
2Types
CarPort
GarageType
GarageType
Attchd
BuiltIn
Detchd
Basment
2Types
CarPort


count
0
100
200
300
400
500
RFn
Unf
Fin
GarageFinish
GarageFinish
RFn
Unf
Fin



GarageQual
TA
Fa
Gd
Ex
Po
count
0
200
400
600
800
1000
TA
Fa
Gd
Ex
Po
GarageQual
GarageCond
TA
Fa
Gd
Po
Ex
count
0
200
400
600
800
1000
TA
Fa
Gd
Po
Ex
GarageCond
PavedDrive
Y
N
P
count
0
200
400
600
800
1000
Y
N
P
PavedDrive

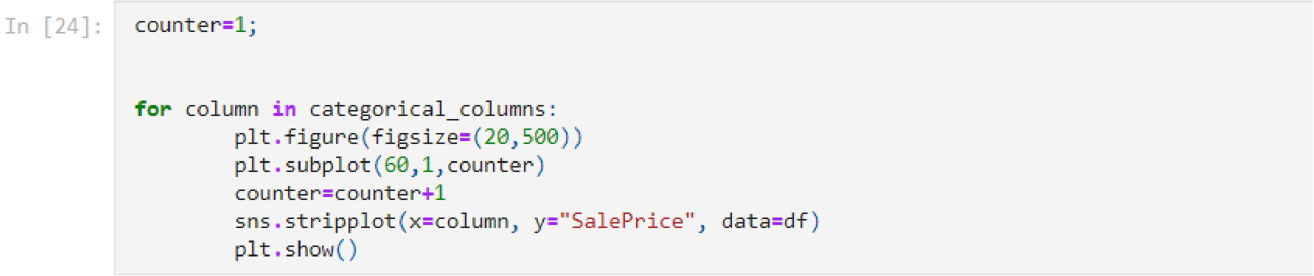




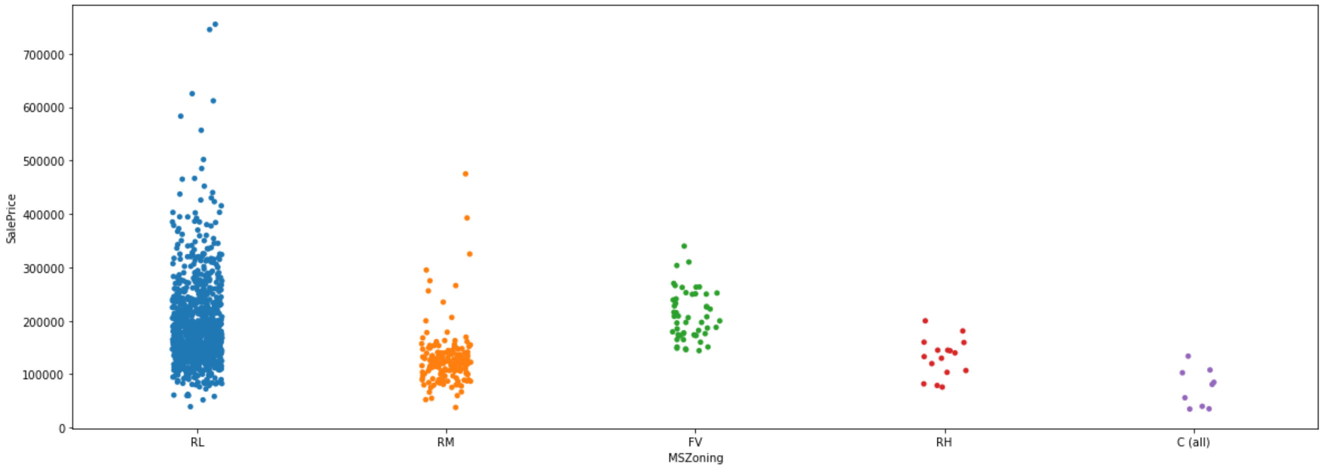
Findings:

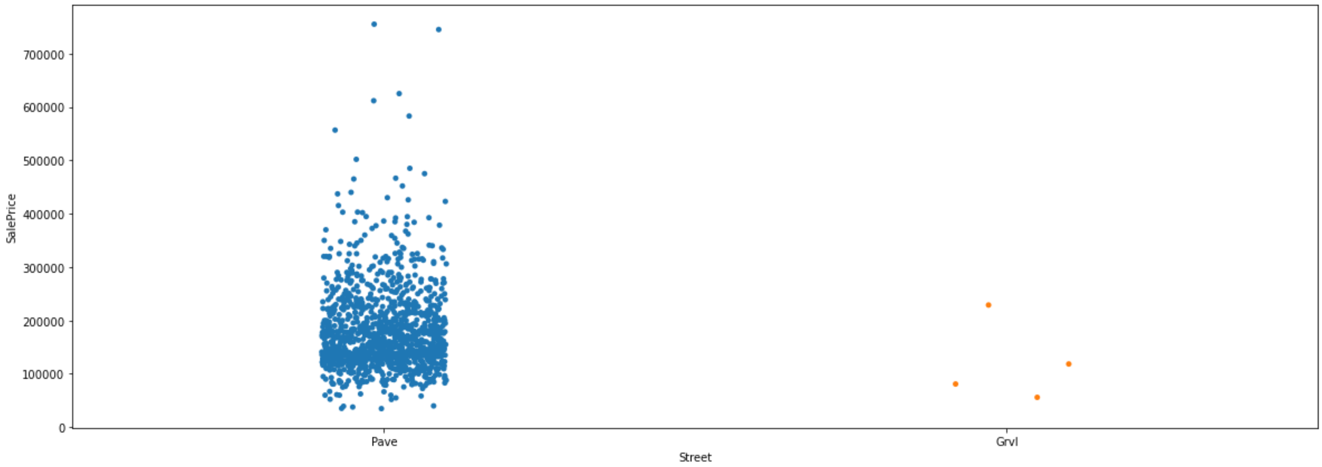
MSZoning -> majority are RL
Street-> majority streets are Pave style
LotShape-> majority are Reg shape
LandContour -> LVL have the highest count in the dataset
Utilities -> All values are AllPub
LotConfig -> We have very few FR3 and majority are Inside
LandSlope -> Landslope is Gentle for majority of houses
Neighborhood -> their are many differnt neighborhood present
Condition1 ->Majority of the houses are norm
Condition2 -> Majority of the houses are norm
BldgType -> Maajority of the houses are 1Fam
HouseStyle -> We have very few 2.5fin houses
RoofStyle -> We have very few houses with shed
RoofMatl -> Most of the houses have Compshg
Exterior1st -> Most of the houses have VinylSd
Exterior2nd -> Most of the houses have VinylSd
MasVnrType -> Most of the houses dont have this
Foundation -> Their are 0 houses with wood foundation
BsmtQual -> Very few houses have Fa quality
BsmtCond -> Most of the houses have TA
BsmtExposure -> Most of the houses dont have exposure
BsmtFinType1 -> Very few houses have LwQ
BsmtFinType2 -> Most of the houses have Unf
Heating -> Almost all the houses have GasA type
HeatingQC -> Most of the houses have Ex

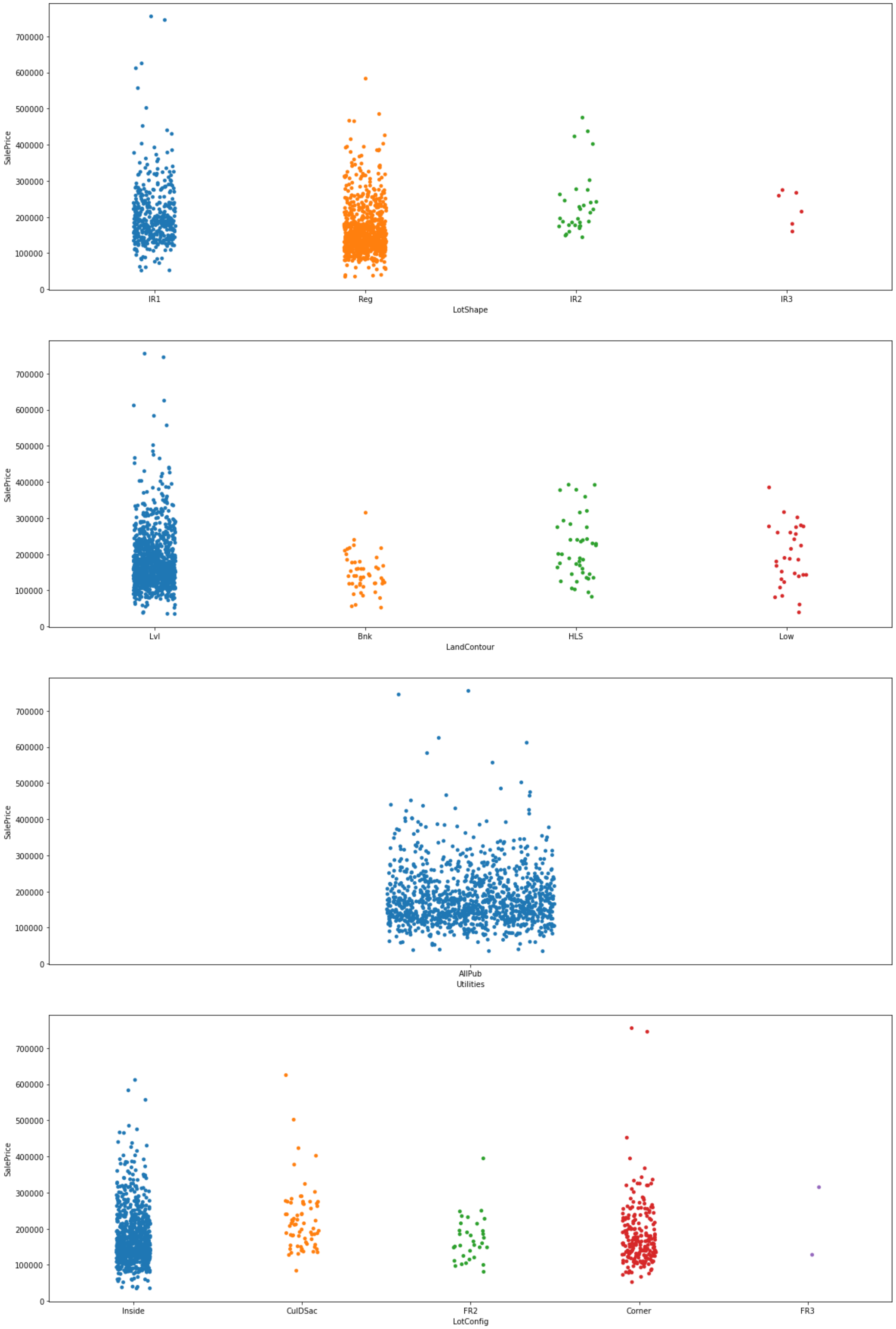
CentralAir -> Very houses doesnot have central air facility
Electrical -> Their are no houses with FuseP and Mix
KitchenQual -> Most of the houses have TA quality
GarageType -> Very few houses have Basement, 2types and CarPort
GarageFinish -> Majority of the houses have GarageFinish
PavedDrive -> Most of the houses have Paved drive
SaleType -> Almost all the houses have WD salestype
SaleCondition -> Most of the houses have normal sales condition

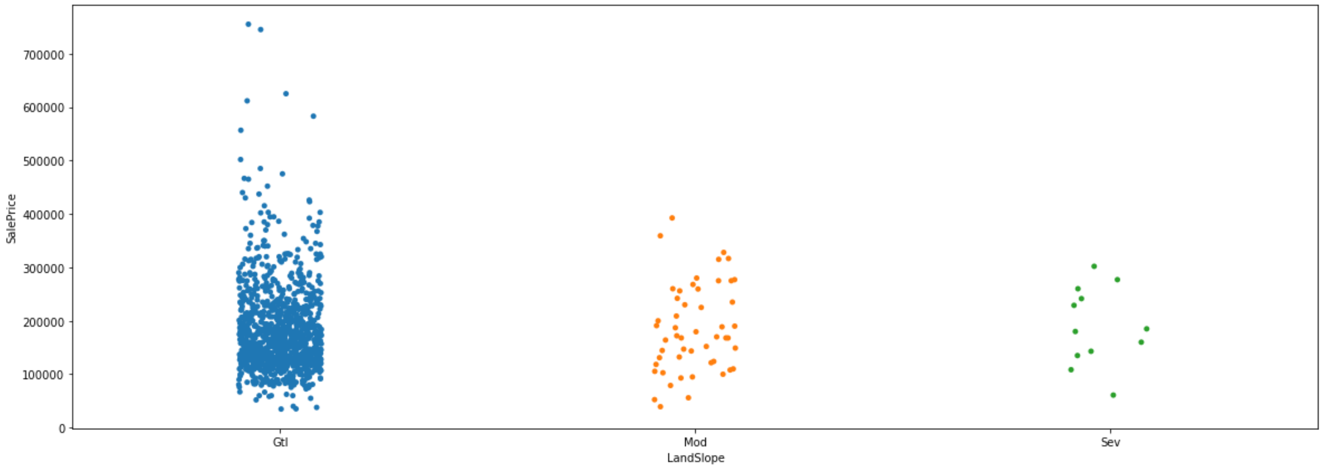


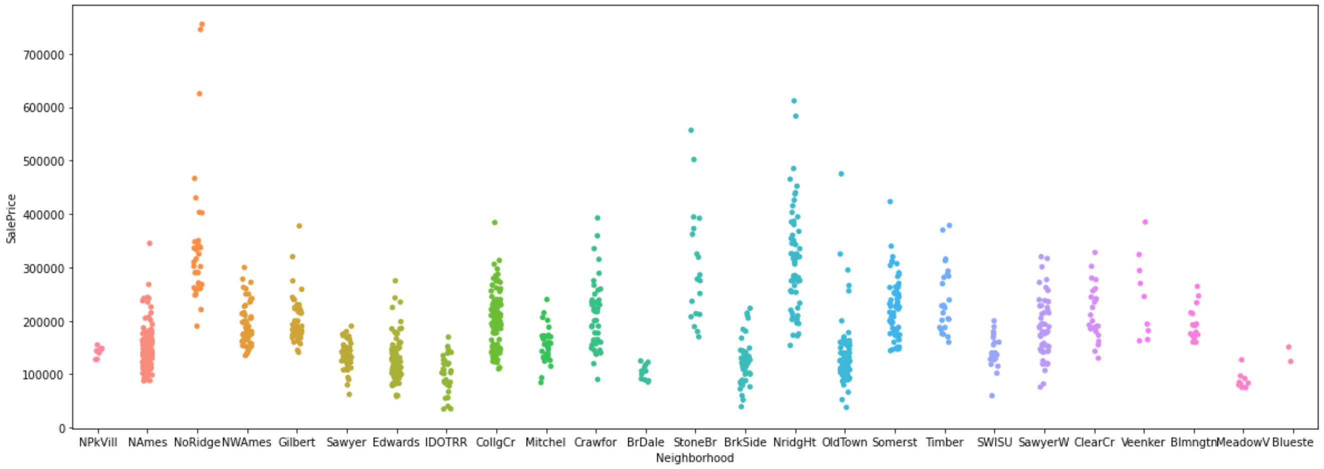
```
In [24]: counter=1;


         for column in categorical_columns:
                 plt.figure(figsize=(20,500))
                 plt.subplot(60,1,counter)
                 counter=counter+1
                 sns.stripplot(x=column, y="SalePrice", data=df)
                 plt.show()
```

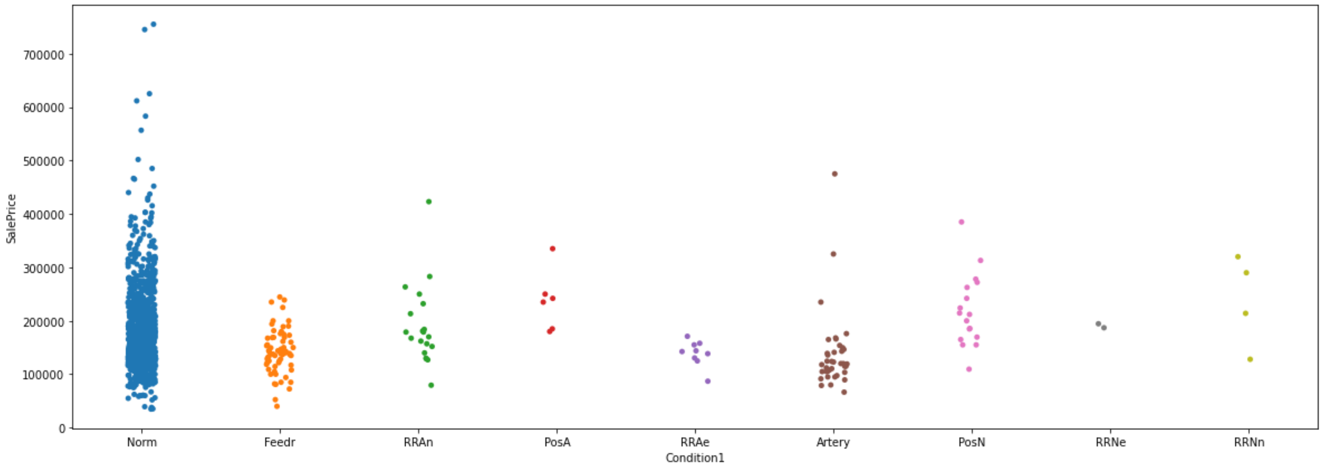


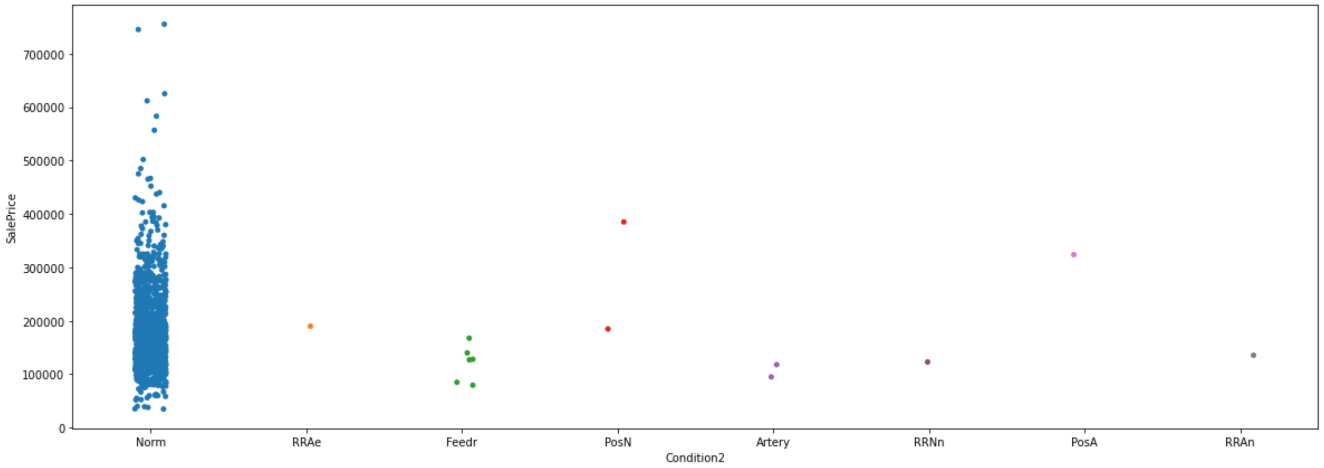


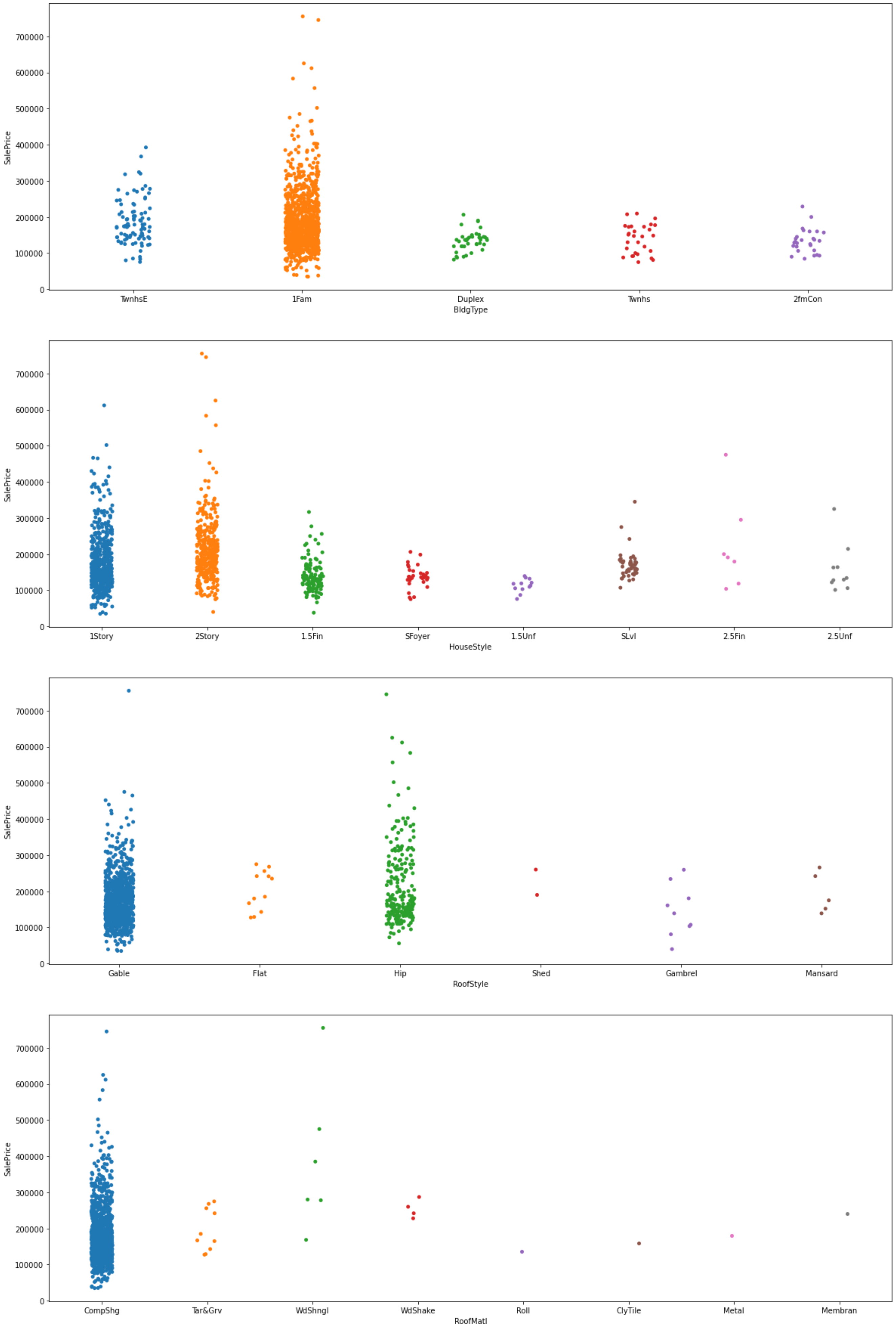

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
IR1
Reg
IR2
IR3
LotShape
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Lvl
Bnk
HLS
Low
LandContour
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
AllPub
Utilities
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Inside
CulDSac
FR2
Corner
FR3
LotConfig

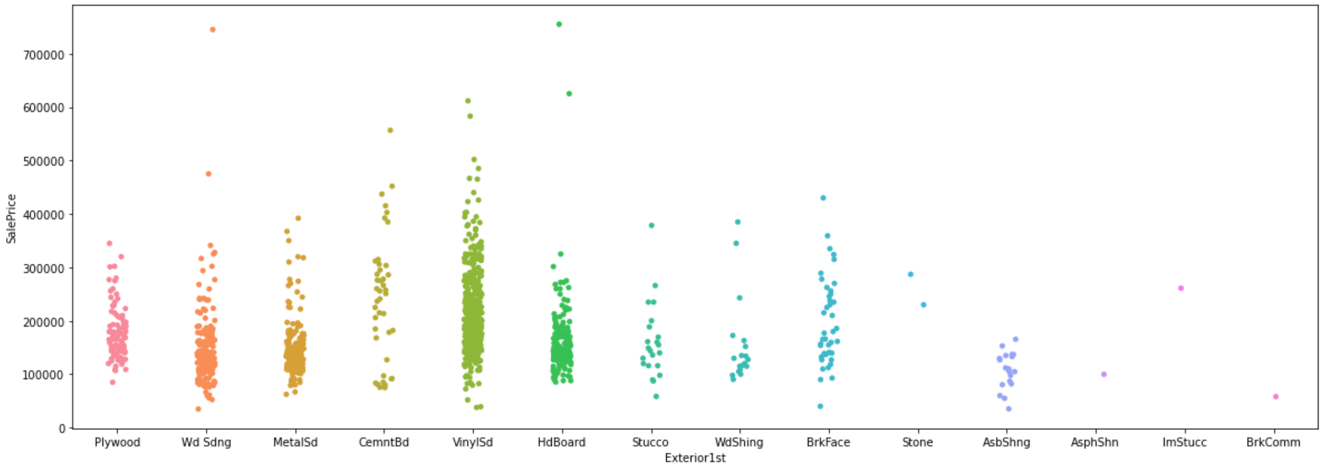

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Gtl
Mod
Sev
LandSlope

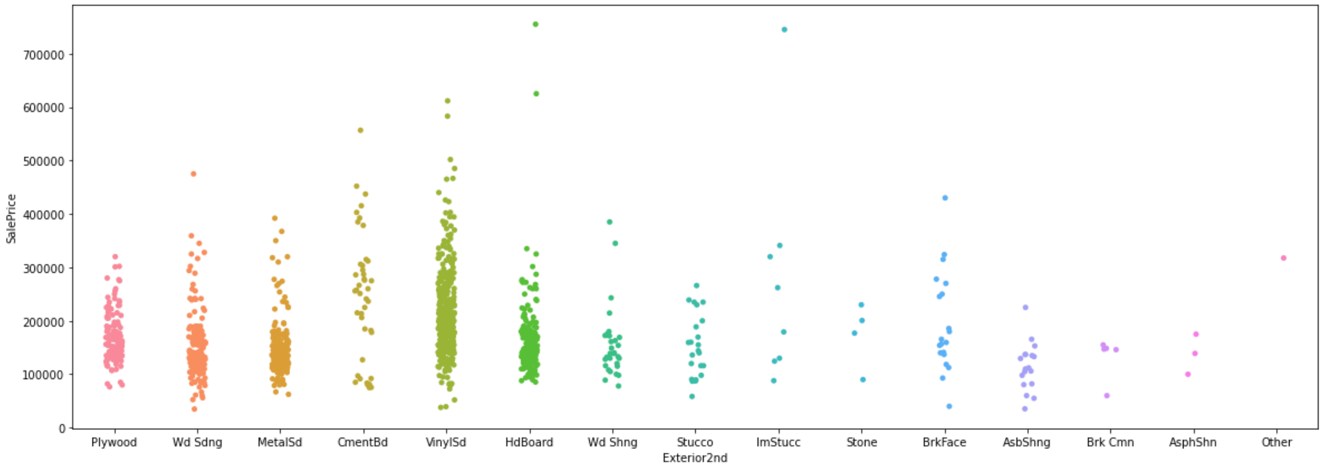

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
Sawyer
Edwards
IDOTRR
CollgCr
Mitchel
Crawfor
BrDale
StoneBr
BrkSide
NridgHt
OldTown
Somerst
Timber
SWISU
SawyerW
ClearCr
Veenker
Blmngtn
MeadowV
Blueste
Neighborhood

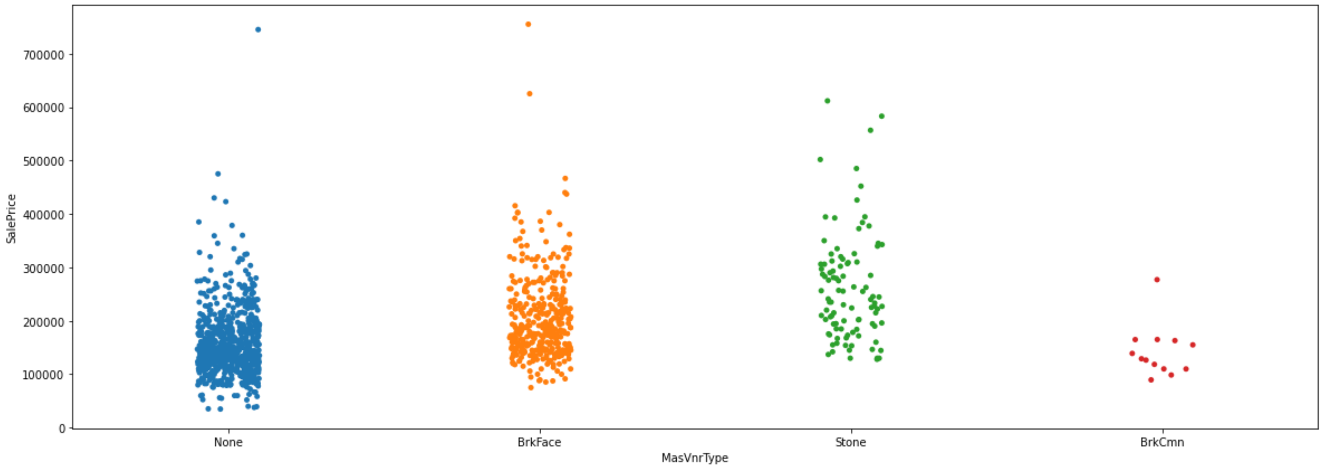

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Norm
Feedr
RRAn
PosA
RRAe
Artery
PosN
RRNe
RRNn
Condition1

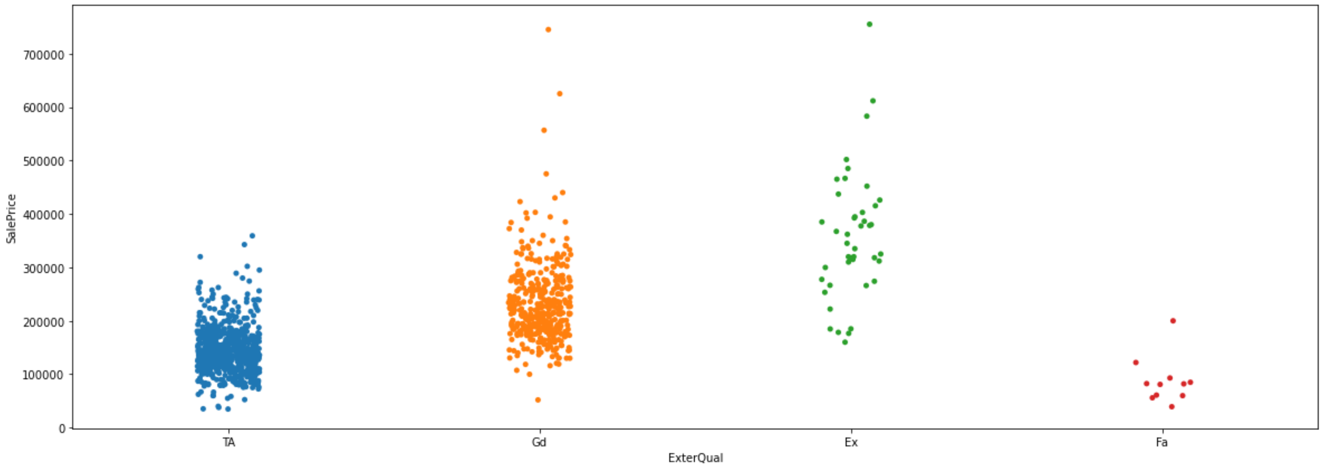

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Norm
RRAe
Feedr
PosN
Artery
RRNn
PosA
RRAn
Condition2

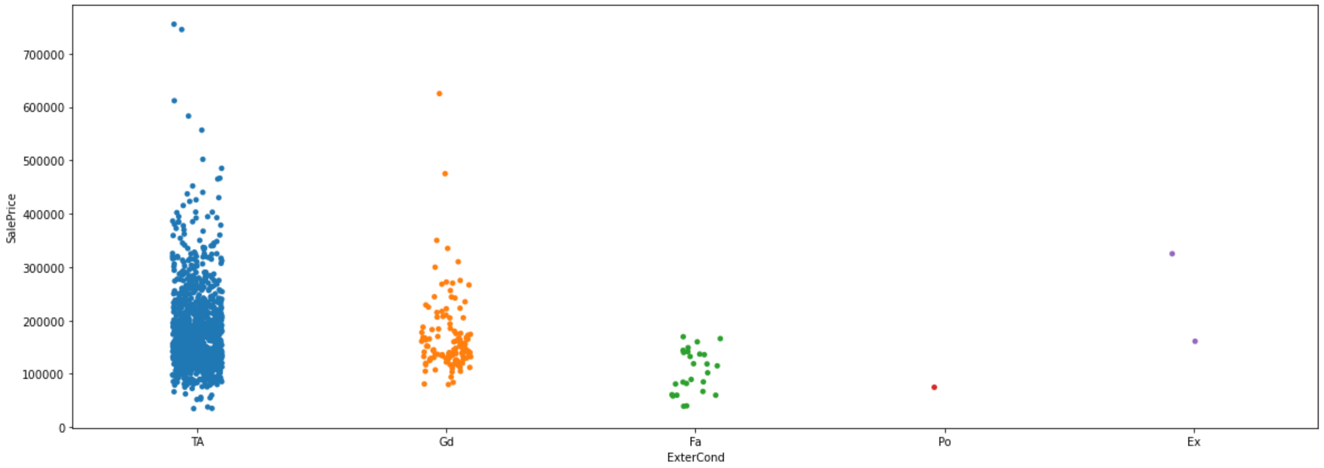

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TwnhsE
1Fam
Duplex
Twnhs
2fmCon
BldgType
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1Story
2Story
1.5Fin
SFoyer
1.5Unf
SLvl
2.5Fin
2.5Unf
HouseStyle
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Gable
Flat
Hip
Shed
Gambrel
Mansard
RoofStyle
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
CompShg
Tar&Grv
WdShngl
WdShake
Roll
ClyTile
Metal
Membran
RoofMatl

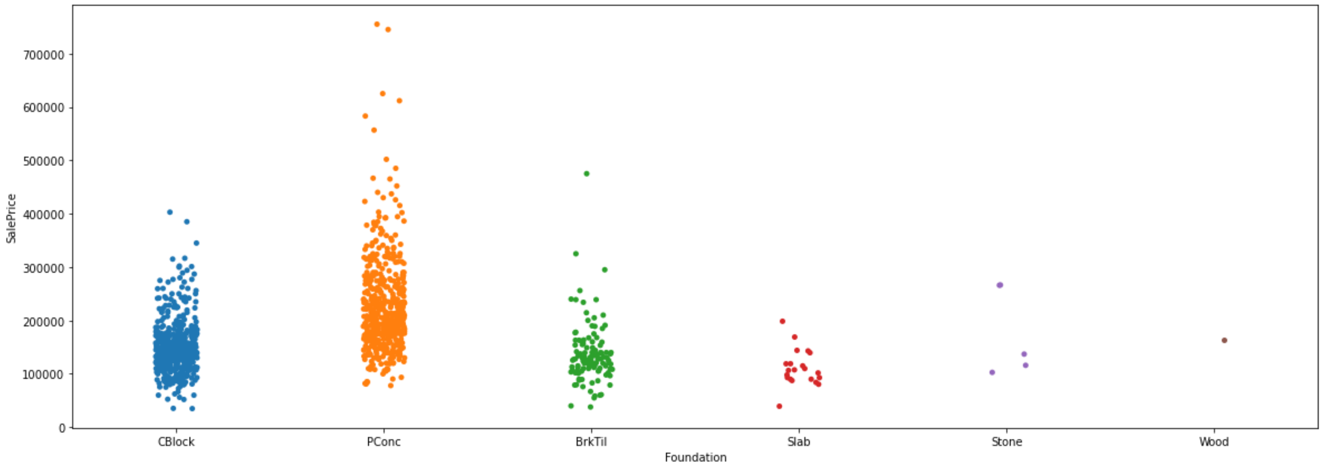

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
Exterior1st

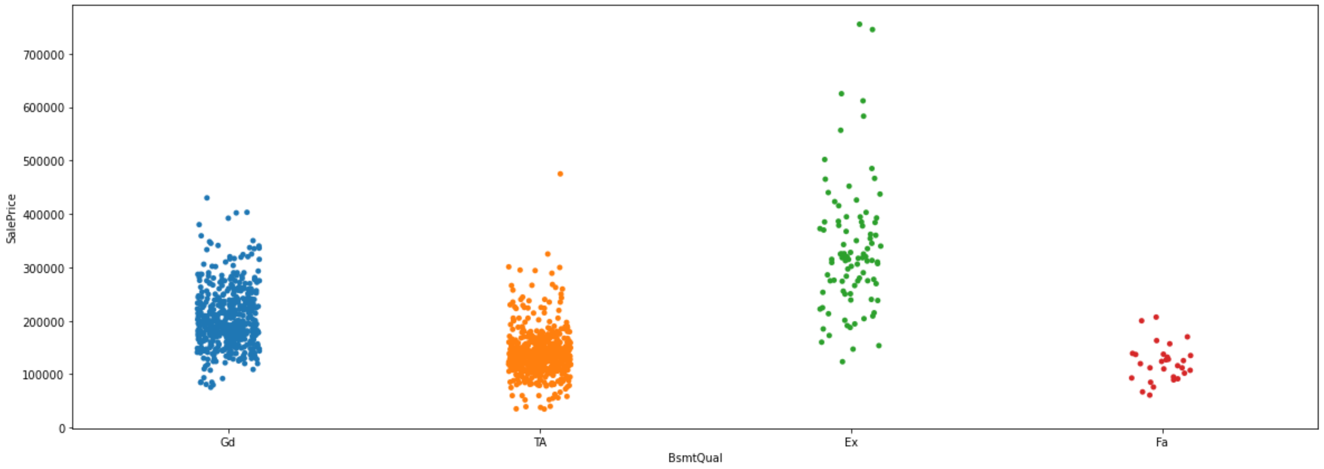

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Plywood
Wd Sdng
MetalSd
CmentBd
VinylSd
HdBoard
Wd Shng
Stucco
ImStucc
Stone
BrkFace
AsbShng
Brk Cmn
AsphShn
Other
Exterior2nd

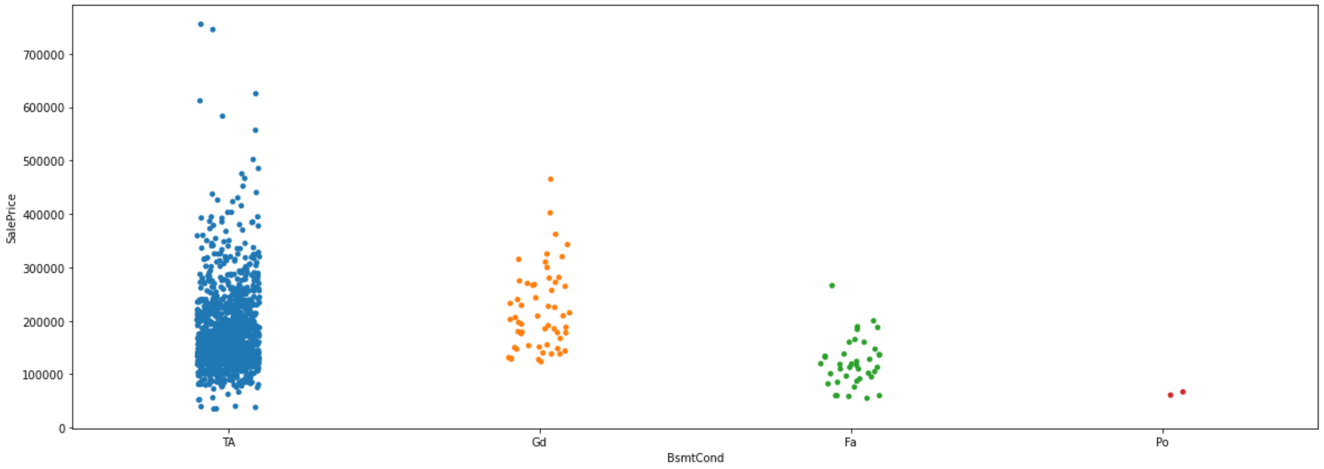

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
None
BrkFace
Stone
BrkCmn
MasVnrType

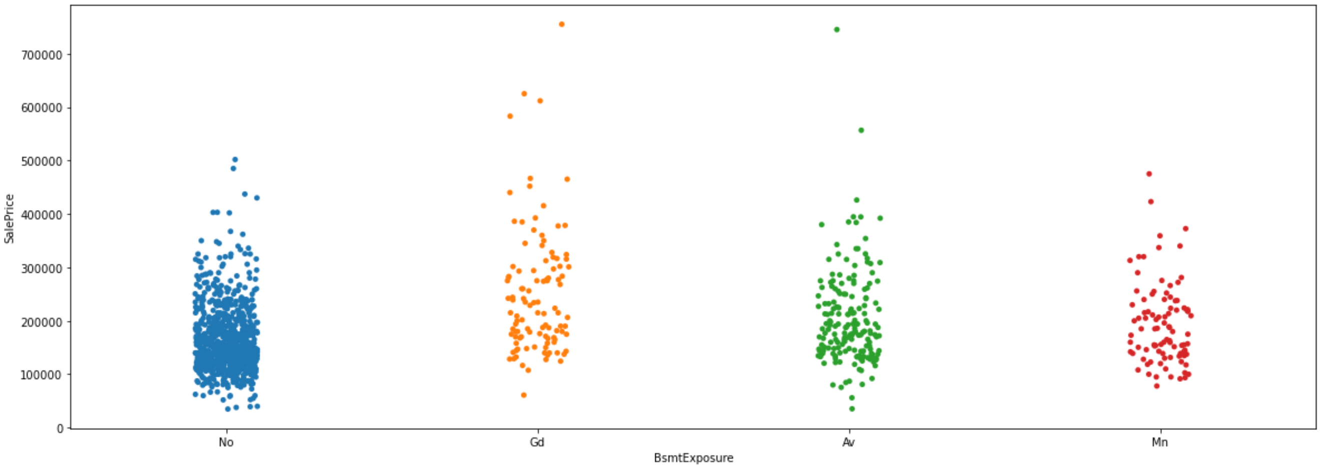

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Ex
Fa
ExterQual

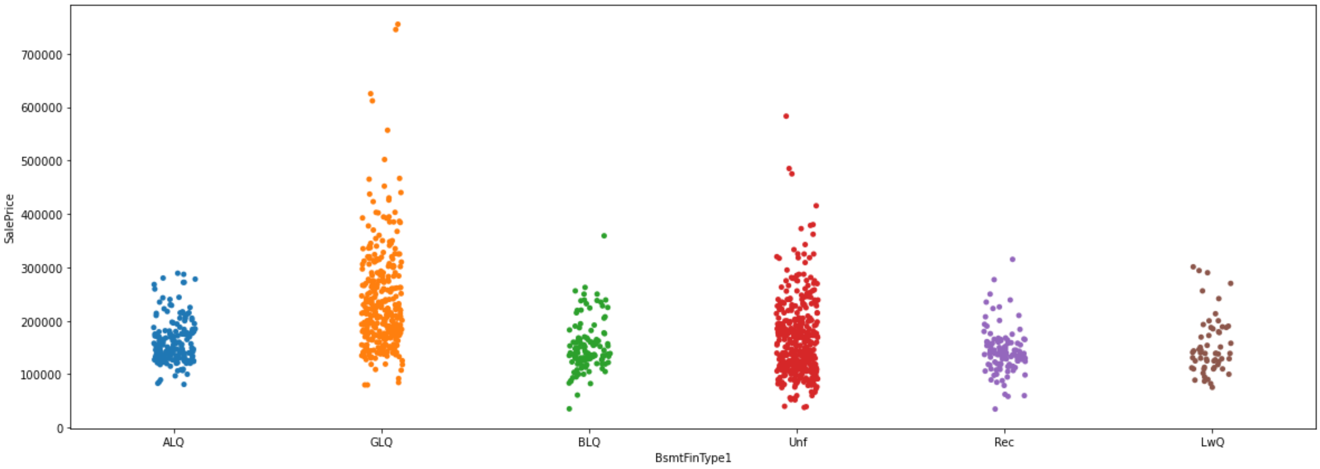

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Fa
Po
Ex
ExterCond

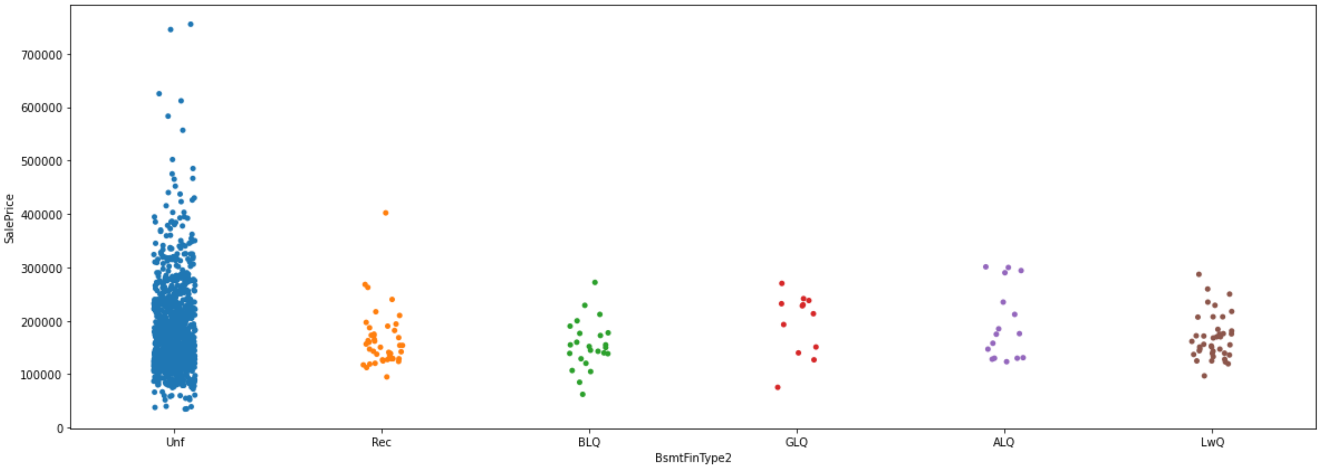

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
CBlock
PConc
BrkTil
Slab
Stone
Wood
Foundation

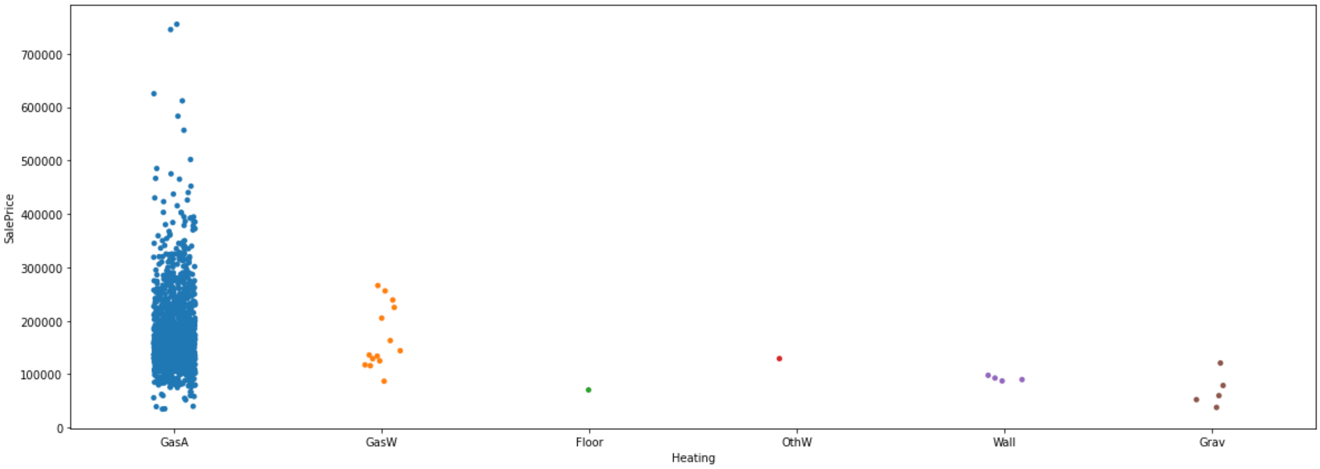

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Gd
TA
Ex
Fa
BsmtQual

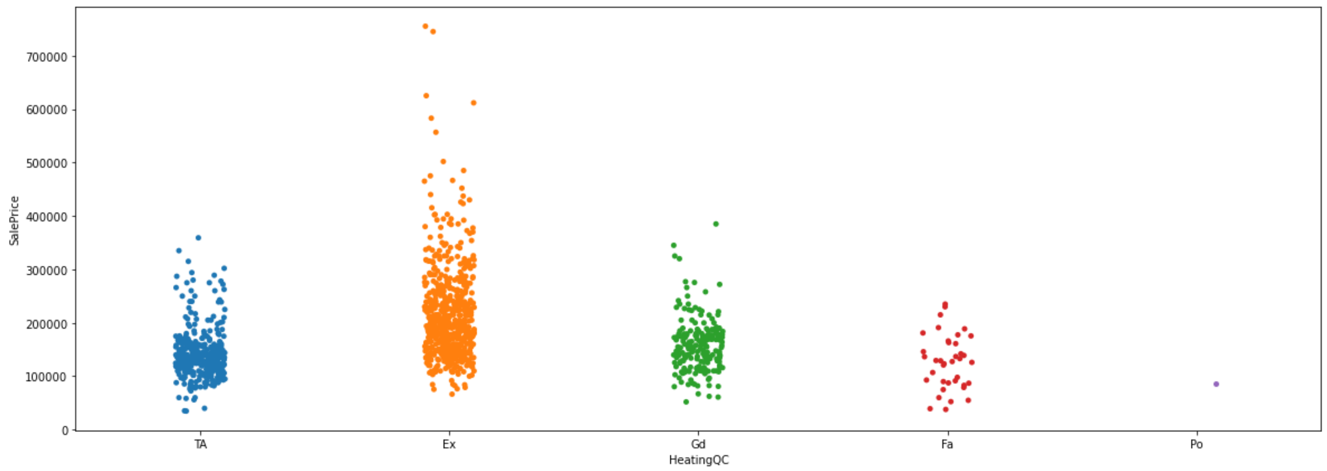

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Fa
Po
BsmtCond

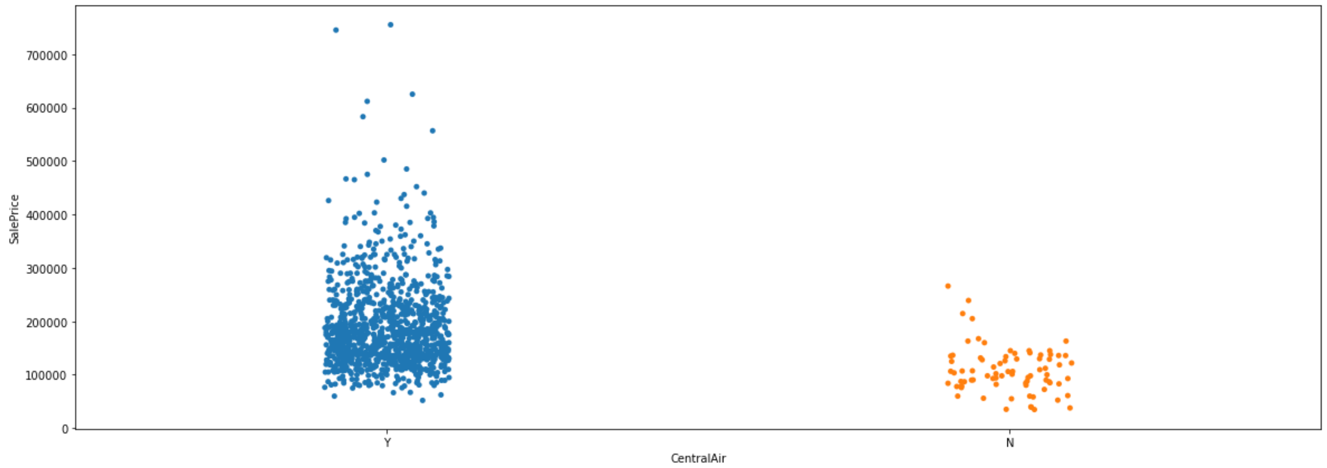

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
No
Gd
Av
Mn
BsmtExposure

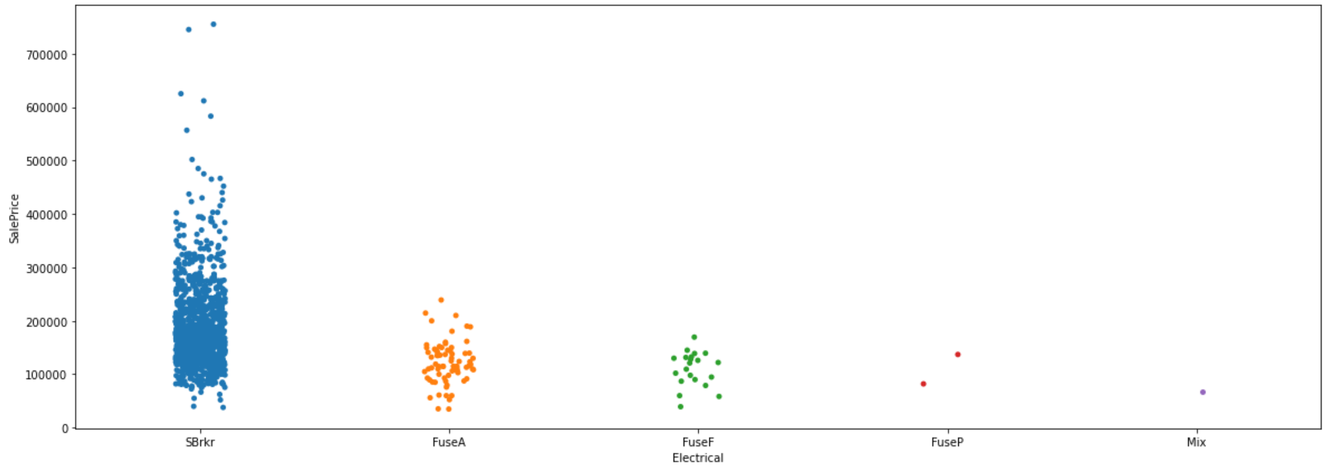

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
ALQ
GLQ
BLQ
Unf
Rec
LwQ
BsmtFinType1

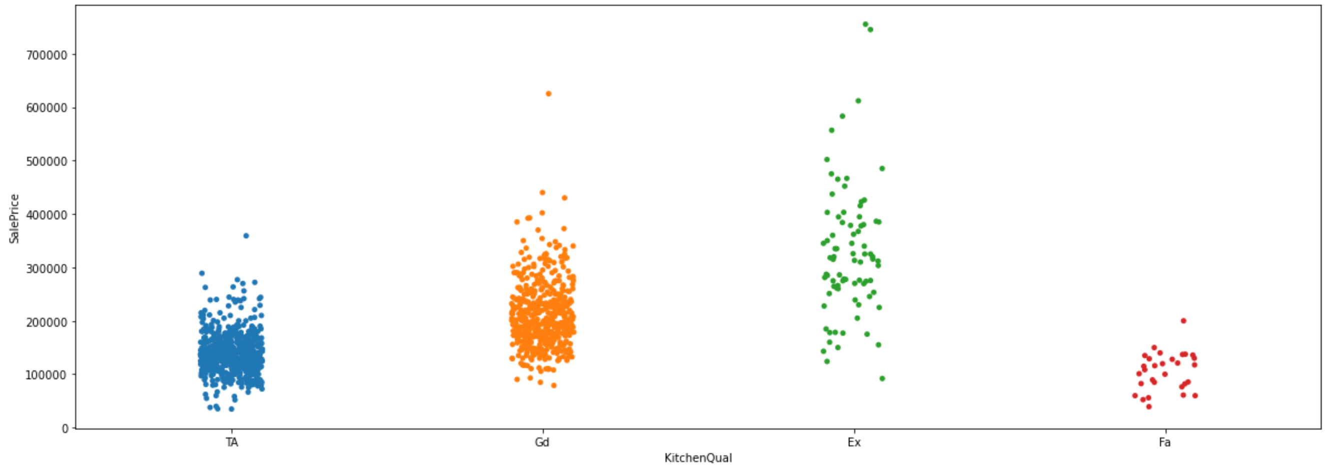

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Unf
Rec
BLQ
GLQ
ALQ
LwQ
BsmtFinType2

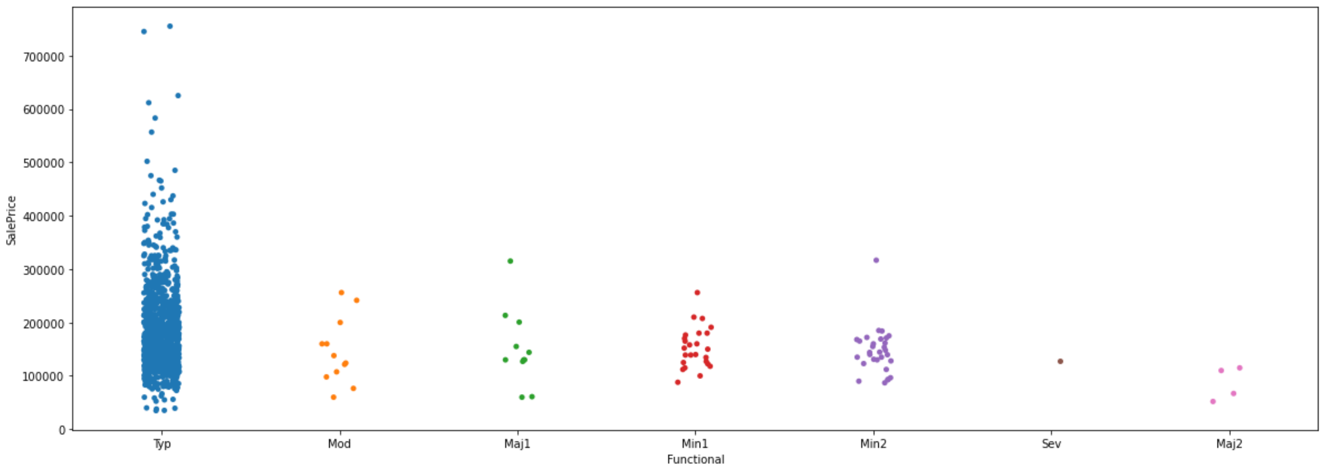

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
GasA
GasW
Floor
OthW
Wall
Grav
Heating


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Ex
Gd
Fa
Po
HeatingQC

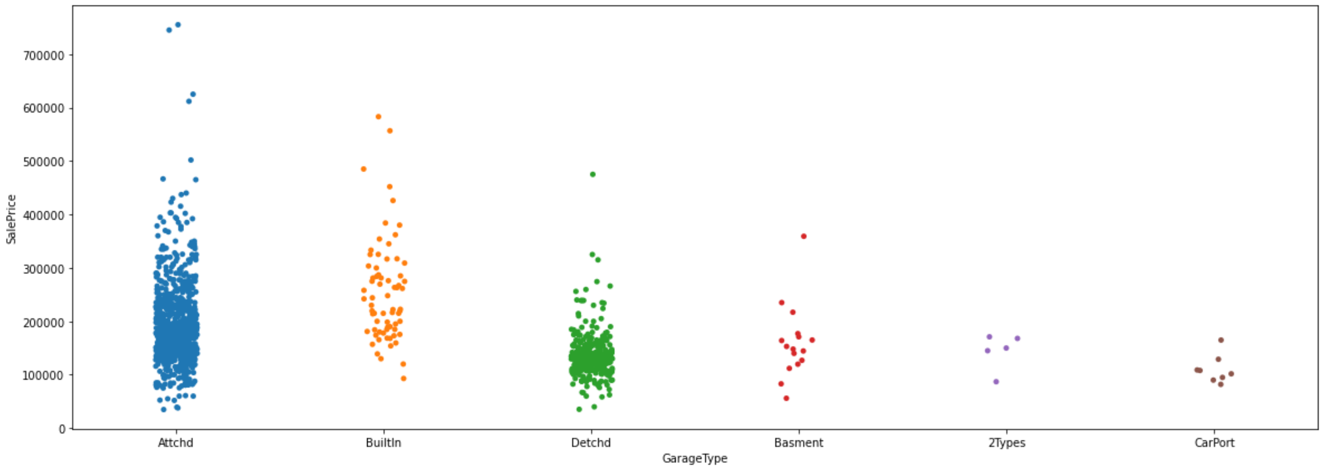

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Y
N
CentralAir


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
SBrkr
FuseA
FuseF
FuseP
Mix
Electrical

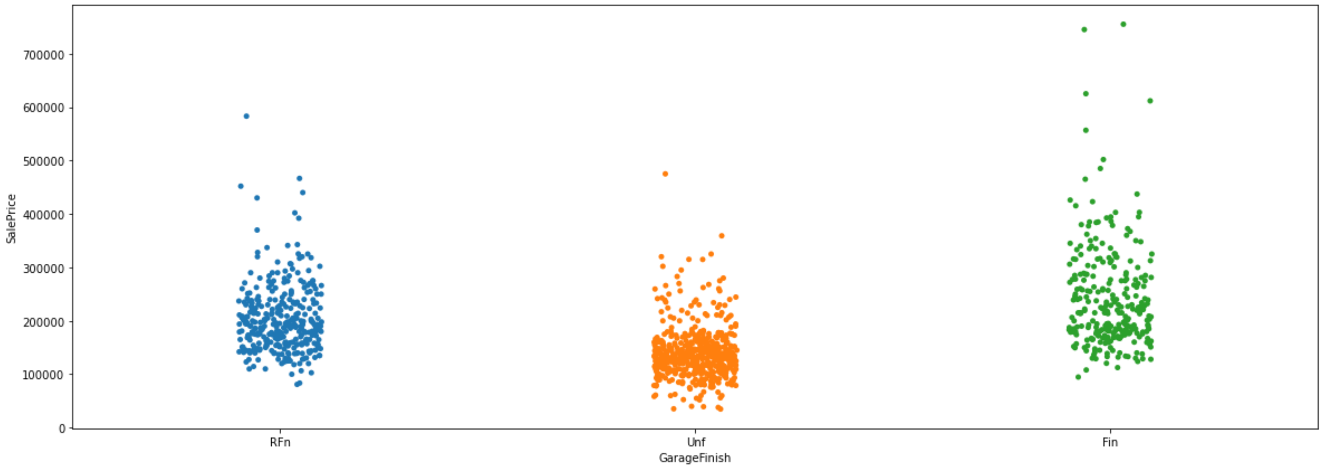

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Ex
Fa
KitchenQual



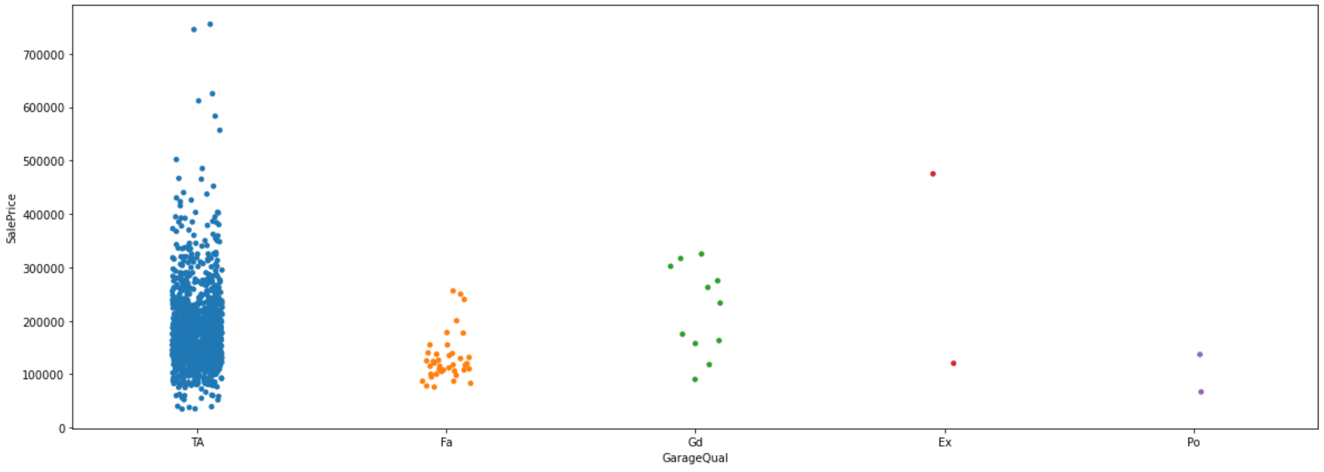
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Typ
Mod
Maj1
Min1
Min2
Sev
Maj2
Functional



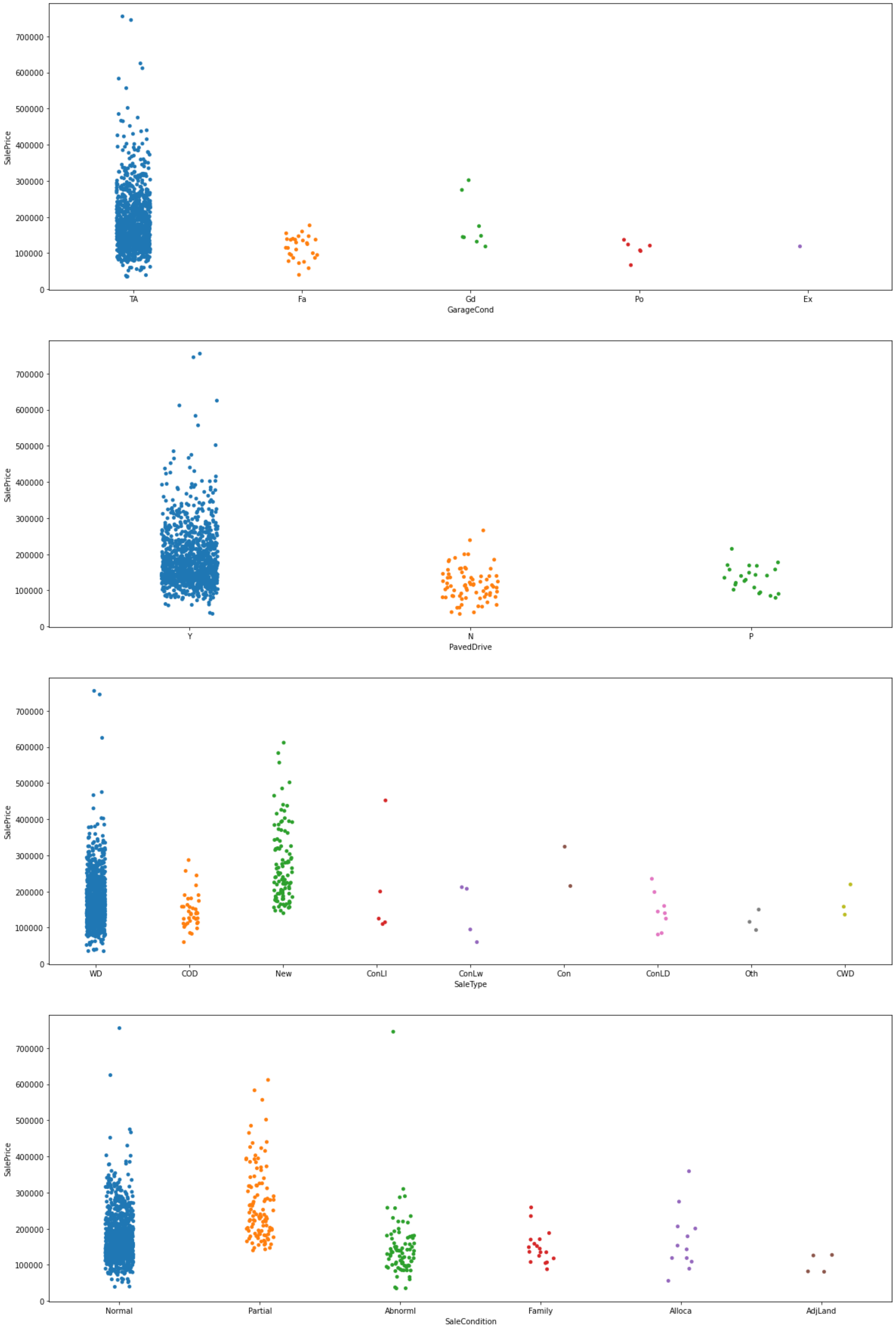
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Attchd
BuiltIn
Detchd
Basment
2Types
CarPort
GarageType



SalePrice
0
100000
200000
300000
400000
500000
600000
700000
RFn
Unf
Fin
GarageFinish



SalePrice
0
100000
200000
300000
400000
500000
600000
700000
TA
Fa
Gd
Ex
Po
GarageQual



SalePrice
0
100000
200000
300000
400000
500000
600000
700000
TA
Fa
Gd
Po
Ex
GarageCond
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Y
N
P
PavedDrive
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
WD
COD
New
ConLI
ConLw
Con
ConLD
Oth
CWD
SaleType
SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Normal
Partial
Abnorml
Family
Alloca
AdjLand
SaleCondition

Findings:

MSZoning -> With RL zoning the property have higher value
Street-> with Pave stype property have higher value
LotShape-> IR1 shape property have higher value
LandContour -> LVL property have higher value
LotConfig -> Corner property have higher value
LandSlope -> Gentle slope property have higher value
Neighborhood -> NoRidge property have higher value
Condition1 ->norm property have higher value
Condition2 ->norm property have higher value
BldgType -> 1Fam property have higher value
HouseStyle -> 2 story property have higher value
RoofStyle -> Gable and Hip stype property have higher value
RoofMatl -> Compshg and WdShngle type property have higher value
Exterior1st -> Brkcomm, Aspshnn style decreases the property value
Exterior2nd -> Hd board type property have higher value
MasVnrType -> BrkCmn type decreases the property value
ExterQual -> Gd and Ex quality have higher property value
Foundation -> Pconc foundation property have higher value
BsmtQual -> Ex quality property have higher value
BsmtCond -> Po quality property have low price
BsmtFinType1 -> GLQ type have higher property prices
BsmtFinType2 -> Unf type have higher price
Heating -> GasA heating system have higher property price
HeatingQC -> Houses with Fa HeatingQc price is low
CentralAir -> Houses with central air have higher cost
Electrical -> houses with FuseP and Mix have lower property value
KitchenQual -> Excelent kitchen quality can increase the Property value
GarageType -> Attached garage have higher property value
GarageQual -> Poor garage quality decreases the price of property
PavedDrive -> Paved drive hiuses have higher price
SaleType -> WD and New sale type can get higher price
SaleCondition -> having AdjLand have lower price

Interpretation of the Results

Results:

1) Large amount of null values are present in the dataset
2) Data Set is not normally distributed
3) Dataset have outliers in most of the variables
4) Dataset is not normalized
5) Dataset is highly skewed
6) Random Forest Algorithm is best suited for the current dataset

CONCLUSION

- Key Findings and Conclusions of the Study

  We found that to predict the House price using Data Science the best way after performing Data Cleaning is to use Random Forest Algorithm it provides 88% accuracy which is better than other Regression algorithms.

- Learning Outcomes of the Study in respect of Data Science

  In data science, there are various steps involved during Data analysis and cleaning. With the help of various Visualization tools like plots, Graphs we were able to perform the actions and observe different things. Like for finding the outliers we used Box Plot visualization, for finding the skewness and normalization we used Count Plot visualization, for finding skewness we visualized the skewness using Heat Map for the clear picture of how the variables are co-related to each other in the dataset. We used different metrics to check which model best fits the prediction for the dataset.

- Limitations of this work and Scope for Future Work

  Data was unbalanced if data was balanced more accurate and clear picture of the output -> result is dependent on the data

  Neural network classifier which are still unexplored & can be taken for future consideration