Fliprobo

# House Price Prediction Model

Report


Submitted by:

**Arjun Verma,**

**Intern Data Scientist**

# ACKNOWLEDGMENT

***I would like to express my greatest appreciation to the all individuals who have helped and supported me throughout the project. I am thankful to Fliprobo team for their ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project.***

***A special acknowledgement goes to my institute Datatrained who helped me in completing the project and learning concepts.***

***I wish to thank my parents as well for their undivided support and interest who inspired me and encouraged me to go my own way, without whom I would be unable to complete my project.***



Below following are the other references:

www.towardsdatascience.com

www.medium.com

www.stackoverflow.com

Datatrained lectures

# 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

**We are required to model the price of houses with the available independent variables. This model will 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.**

**Solution we find that building a machine learning model that can predict upcoming actual price of the houses from previous house prices dataset. Here we implement 5 models and find out best machine learning models.**

## ➢ Review of Literature

1. Here we find two dataset one for training model and another one for prediction upcoming selling houses prices.

2. Prices of house are depends on the various features which we will show later observations how different feature impact house prices.

3. For building a best model for prediction we did EDA and several mandatory requirement procedures for enhancing and improving model accuracy to predict house prices.

## ➢ Motivation for the Problem Undertaken

**Genuinely it's a need of the real states services to complete their goal with higher revenue and low expenditure. Hence this model can brings higher revenue because we can predict upcoming selling property prices and bid a price to the seller with lower amount, before their publishment of house prices.**

## ➢ Mathematical/ Analytical Modeling of the Problem

**Data is statistically analysed through variance inflation factor. Analysed through correlation and multicollinearity. Graphical modelling done through seaborn and matplotlib to understanding how different features impact dataset.**

## Data Sources and their formats

**Datasets are provided by fliprobo for building machine learning model to predict house price based on given parameter.**

**Dataset are in two parts one is for building model and second one is for predict price.**

**Train dataset: Dataset is having 1168 rows and 81 columns including target.**

**Test dataset: Dataset is having 292 rows and 80 columns**

**The information about features are as follows**

'Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', FullBath, 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType', 'SaleCondition'

- MSSubClass: Identifies the type of dwelling involved in the sale.
- MSZoning: Identifies the general zoning classification of the sale.
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access to property
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access to property
- Alley: Type of alley access to property
- LotShape: General shape of property

- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to various conditions
- Condition2: Proximity to various conditions (if more than one is present)
- BldgType: Type of dwelling
- HouseStyle: Style of dwelling
- OverallQual: Rates the overall material and finish of the house
- OverallCond: Rates the overall condition of the house
- YearBuilt: Original construction date
- YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)
- RoofStyle: Type of roof
- RoofMatl: Roof material
- Exterior1st: Exterior covering on house
- Exterior2nd: Exterior covering on house (if more than one material)
- MasVnrType: Masonry veneer type
- MasVnrArea: Masonry veneer area in square feet
- ExterQual: Evaluates the quality of the material on the exterior
- ExterCond: Evaluates the present condition of the material on the exterior
- Foundation: Type of foundation
- BsmtQual: Evaluates the height of the basement
- BsmtCond: Evaluates the general condition of the basement
- BsmtExposure: Refers to walkout or garden level walls

- BsmtFinSF2: Type 2 finished square feet

- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area
- Heating: Type of heating
- HeatingQC: Heating quality and condition
- CentralAir: Central air conditioning
- Electrical: Electrical system
- 1stFlrSF: First Floor square feet
- 2ndFlrSF: Second floor square feet
- LowQualFinSF: Low quality finished square feet (all floors)
- G LivArea: Above grade (ground) living area square feet
- Bs mtFullBath: 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
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- Functional: Home functionality (Assume typical unless deductions are warranted)

- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
- GarageType: Garage location
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- GarageQual: Garage quality

- GarageCond: Garage condition
- PavedDrive: Paved driveway
- 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
- Fence: Fence quality
- MiscFeature: Miscellaneous feature not covered in other categories
- MiscVal: $Value of miscellaneous feature
- MoSold: Month Sold (MM)
- YrSold: Year Sold (YYYY)
- SaleType: Type of sale
- SaleCondition: Condition of sale



```
df_train.head() # checking first 5 rows
```

| | Id | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley | LotShape | LandContour | Utilities | LotConfig | LandSlope | Neighborhood | Condition1 | Cond |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 127 | 120 | RL | NaN | 4928 | Pave | NaN | IR1 | Lvl | AllPub | Inside | Gtl | NPkVill | Norm | |
| 1 | 889 | 20 | RL | 95.0 | 15865 | Pave | NaN | IR1 | Lvl | AllPub | Inside | Mod | NAmes | Norm | |
| 2 | 793 | 60 | RL | 92.0 | 9920 | Pave | NaN | IR1 | Lvl | AllPub | CulDSac | Gtl | NoRidge | Norm | |
| 3 | 110 | 20 | RL | 105.0 | 11751 | Pave | NaN | IR1 | Lvl | AllPub | Inside | Gtl | NWAmes | Norm | |
| 4 | 422 | 20 | RL | NaN | 16635 | Pave | NaN | IR1 | Lvl | AllPub | FR2 | Gtl | NWAmes | Norm | |

| 1 | Condition2 | BldgType | HouseStyle | OverallQual | OverallCond | YearBuilt | YearRemodAdd | RoofStyle | RoofMatl | Exterior1st | Exterior2nd | MasVnrType | MasVnrAre |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | Norm | TwnhsE | 1Story | 6 | 5 | 1976 | 1976 | Gable | CompShg | Plywood | Plywood | None | 0. |
| n | Norm | 1Fam | 1Story | 8 | 6 | 1970 | 1970 | Flat | Tar&Grv | Wd Sdng | Wd Sdng | None | 0. |
| n | Norm | 1Fam | 2Story | 7 | 5 | 1996 | 1997 | Gable | CompShg | MetalSd | MetalSd | None | 0. |
| n | Norm | 1Fam | 1Story | 6 | 6 | 1977 | 1977 | Hip | CompShg | Plywood | Plywood | BrkFace | 480. |
| n | Norm | 1Fam | 1Story | 6 | 7 | 1977 | 2000 | Gable | CompShg | CemntBd | CmentBd | Stone | 126. |

| MasVnrArea | ExterQual | ExterCond | Foundation | BsmtQual | BsmtCond | BsmtExposure | BsmtFinType1 | BsmtFinSF1 | BsmtFinType2 | BsmtFinSF2 | BsmtUnfSF | To |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0 | TA | TA | CBlock | Gd | TA | No | ALQ | 120 | Unf | 0 | 958 | |
| 0.0 | Gd | Gd | PConc | TA | Gd | Gd | ALQ | 351 | Rec | 823 | 1043 | |
| 0.0 | Gd | TA | PConc | Gd | TA | Av | GLQ | 862 | Unf | 0 | 255 | |
| 480.0 | TA | TA | CBlock | Gd | TA | No | BLQ | 705 | Unf | 0 | 1139 | |
| 126.0 | Gd | TA | CBlock | Gd | TA | No | ALQ | 1246 | Unf | 0 | 356 | |

| TotalBsmtSF | Heating | HeatingQC | CentralAir | Electrical | 1stFlrSF | 2ndFlrSF | LowQualFinSF | GrLivArea | BsmtFullBath | BsmtHalfBath | FullBath | HalfBath | Bedroor |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1078 | GasA | TA | Y | SBrkr | 958 | 0 | 0 | 958 | 0 | 0 | 2 | 0 | |
| 2217 | GasA | Ex | Y | SBrkr | 2217 | 0 | 0 | 2217 | 1 | 0 | 2 | 0 | |
| 1117 | GasA | Ex | Y | SBrkr | 1127 | 886 | 0 | 2013 | 1 | 0 | 2 | 1 | |
| 1844 | GasA | Ex | Y | SBrkr | 1844 | 0 | 0 | 1844 | 0 | 0 | 2 | 0 | |
| 1602 | GasA | Gd | Y | SBrkr | 1602 | 0 | 0 | 1602 | 0 | 1 | 2 | 0 | |

| BedroomAbvGr | KitchenAbvGr | KitchenQual | TotRmsAbvGrd | Functional | Fireplaces | FireplaceQu | GarageType | GarageYrBlt | GarageFinish | GarageCars |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | TA | 5 | Typ | 1 | TA | Attchd | 1977.0 | RFn | 2 |
| 4 | 1 | Gd | 8 | Typ | 1 | TA | Attchd | 1970.0 | Unf | 2 |
| 3 | 1 | TA | 8 | Typ | 1 | TA | Attchd | 1997.0 | Unf | 2 |
| 3 | 1 | TA | 7 | Typ | 1 | TA | Attchd | 1977.0 | RFn | 2 |
| 3 | 1 | Gd | 8 | Typ | 1 | TA | Attchd | 1977.0 | Fin | 2 |

| GarageArea | GarageQual | GarageCond | PavedDrive | WoodDeckSF | OpenPorchSF | EnclosedPorch | 3SsnPorch | ScreenPorch | PoolArea | PoolQC | Fence |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 440 | TA | TA | Y | 0 | 205 | 0 | 0 | 0 | 0 | NaN | NaN |
| 621 | TA | TA | Y | 81 | 207 | 0 | 0 | 224 | 0 | NaN | NaN |
| 455 | TA | TA | Y | 180 | 130 | 0 | 0 | 0 | 0 | NaN | NaN |
| 546 | TA | TA | Y | 0 | 122 | 0 | 0 | 0 | 0 | NaN | MnPrv |
| 529 | TA | TA | Y | 240 | 0 | 0 | 0 | 0 | 0 | NaN | NaN |

| SaleCondition | SalePrice |
|---|---|
| Normal | 128000 |
| Normal | 268000 |
| Normal | 269790 |
| Normal | 190000 |
| Normal | 215000 |

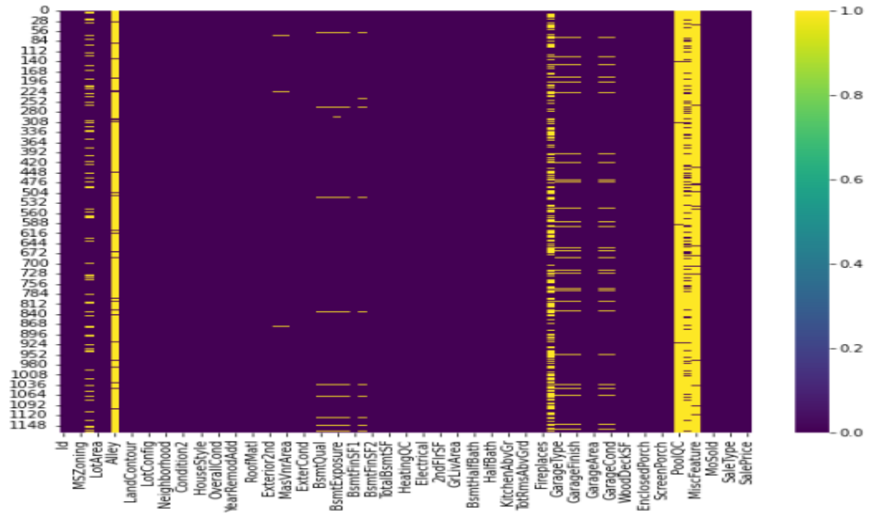
# Dataset Information

- These columns 'MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities', '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' are of **object** types.
- These columns 'Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual','OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath','HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd','Fireplaces', 'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF','OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SalePrice' are of **numerical** types.

### Checking Null Values of the dataset

```
plt.figure(figsize = (10,8))
sns.heatmap(df_train.isnull(), cmap = 'viridis')  #checking null values visually
```

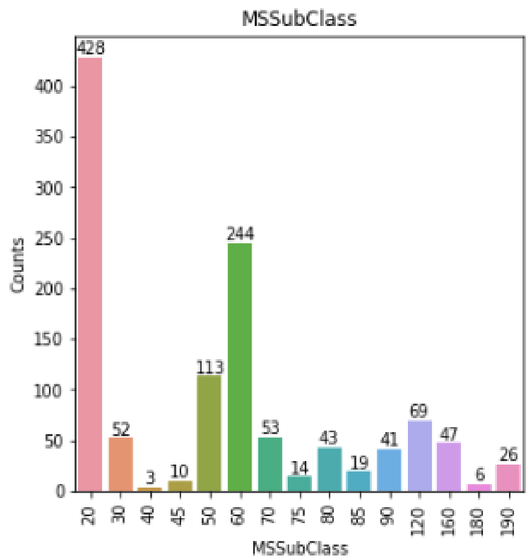
```
<AxesSubplot:>
```

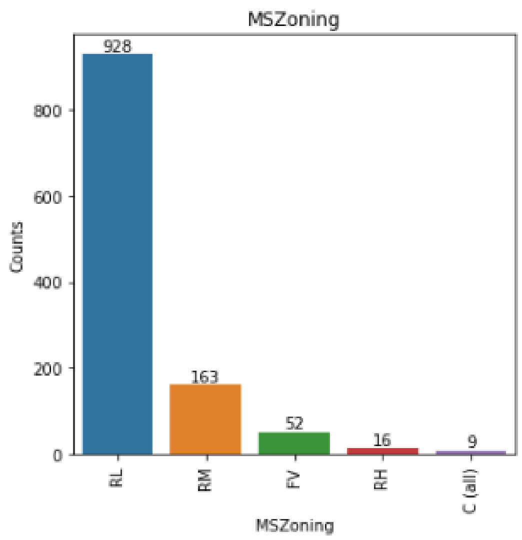



- ➢ Highlighted yellowish columns having null values in the dataset.
- ➢ We dealed with all null values filling values with mode.
- ➢ EDA done to clean for better understandings.

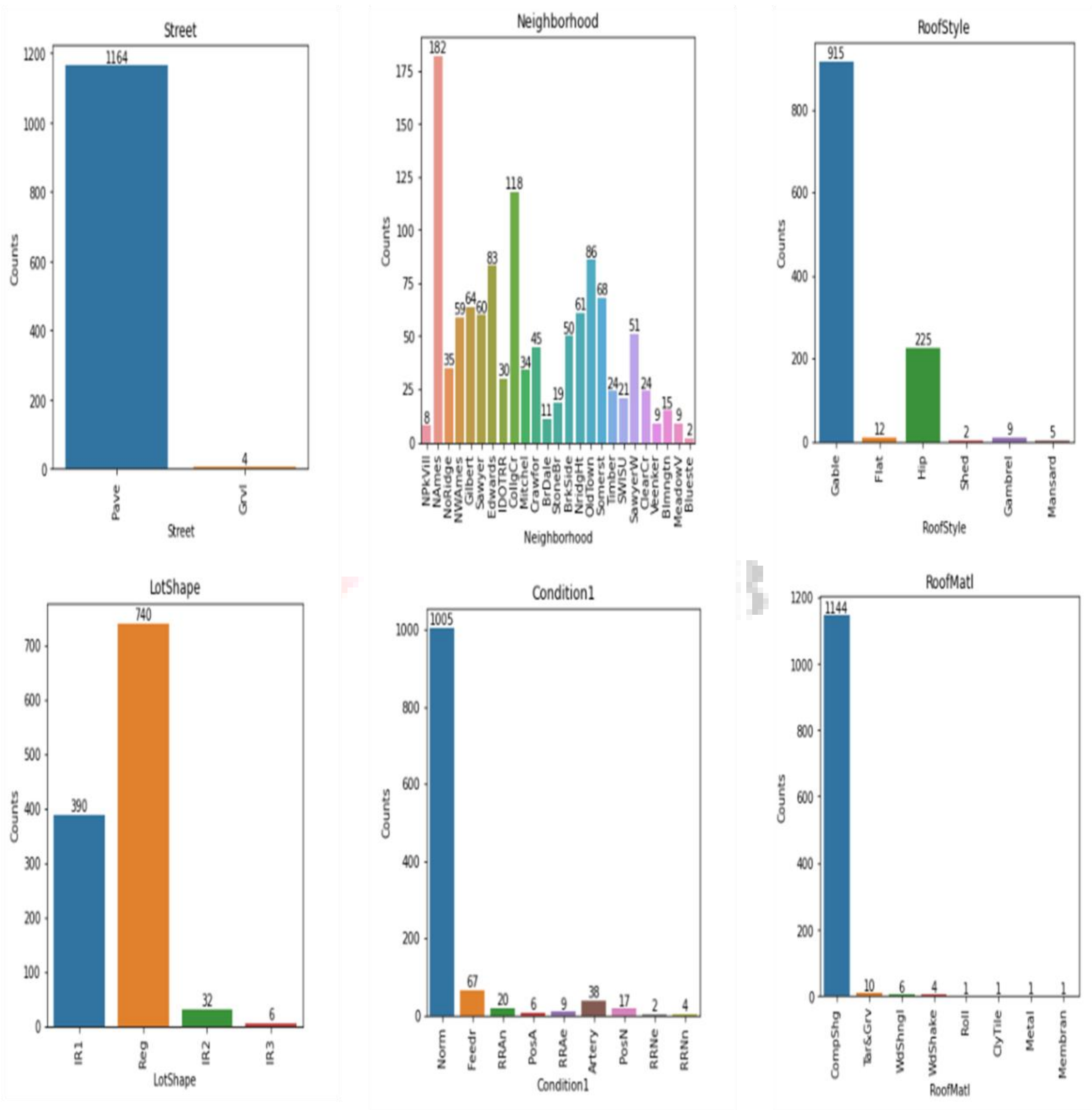
### Visualizing Parameters for better understandings

Bar Plot

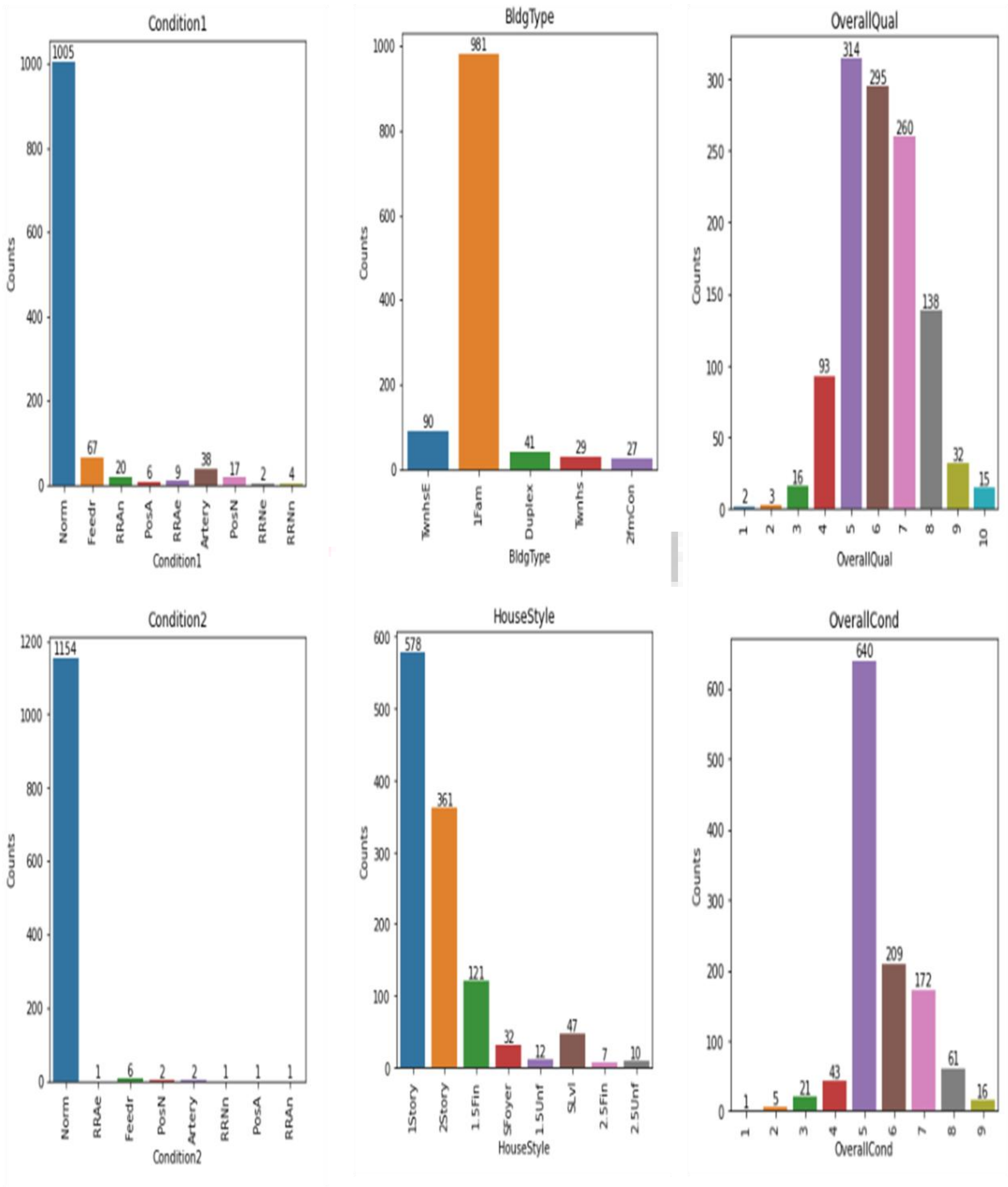


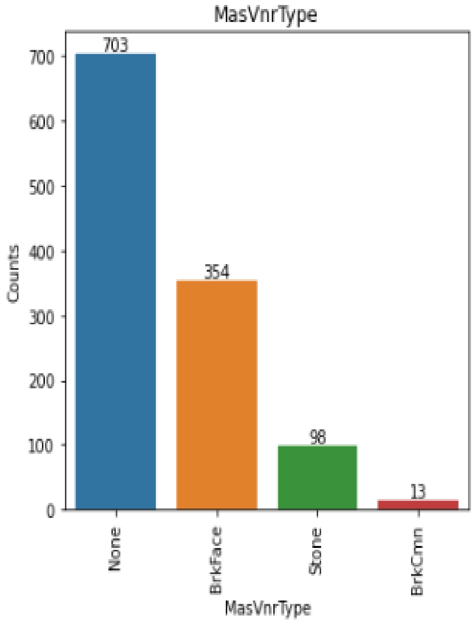



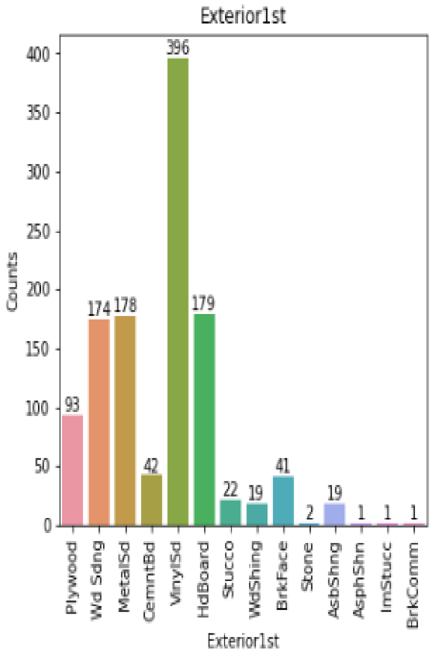
## Visualization

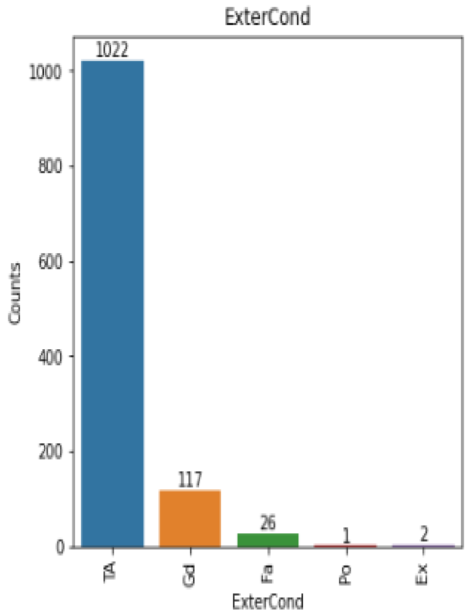


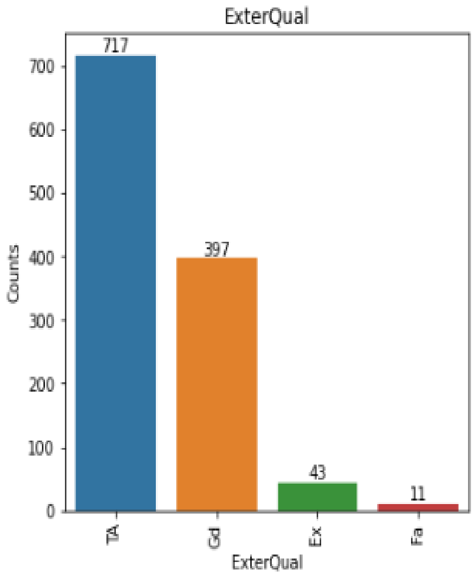
# Visualization

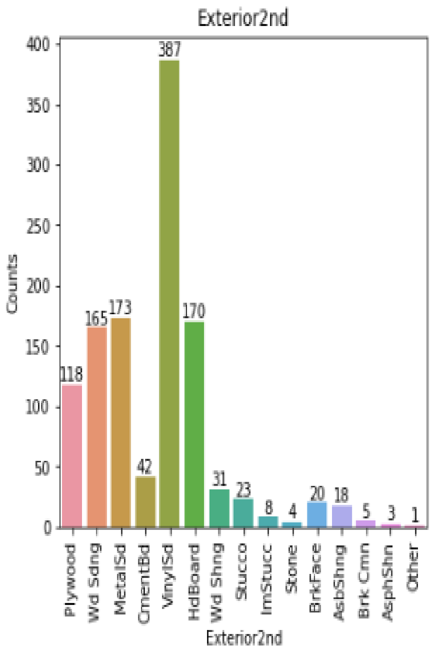


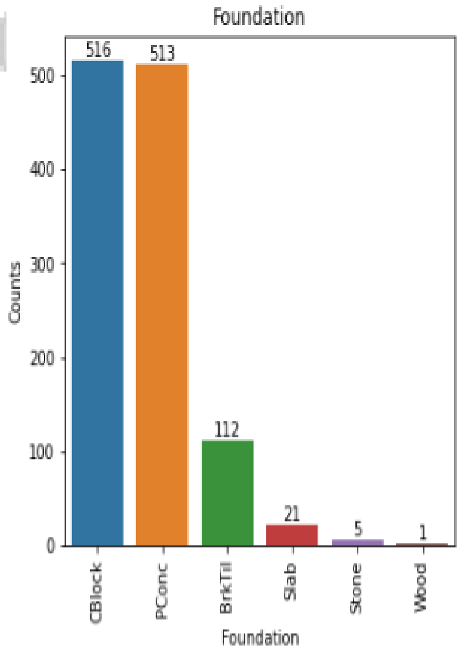

MasVnrType
Counts
MasVnrType
None
BrkFace
Stone
BrkCmn
703
354
98
13


Exterior1st
Counts
Exterior1st
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
93
174
178
42
396
179
22
19
41
2
19
1
1
1

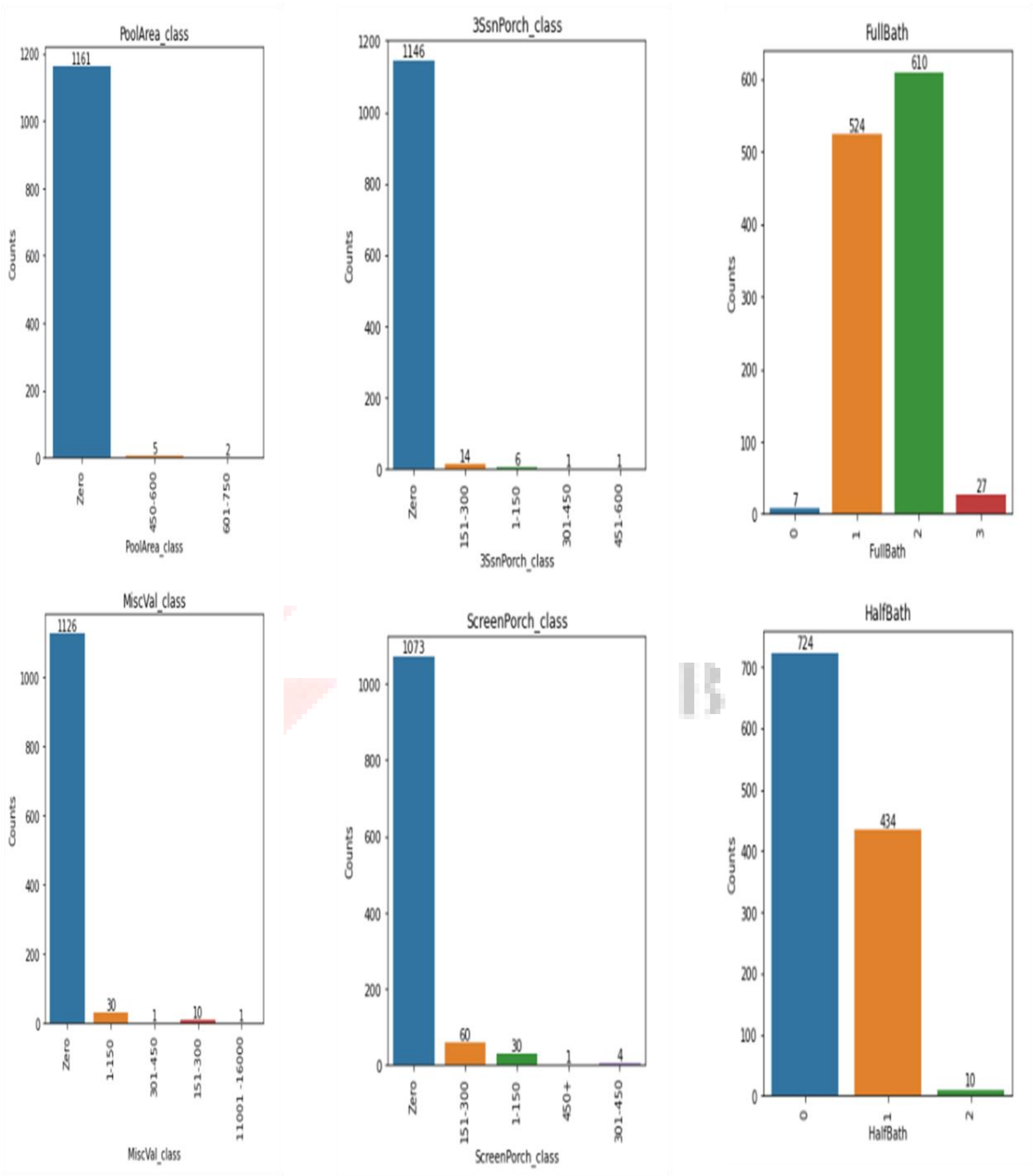

ExterCond
Counts
ExterCond
TA
Gd
Fa
Po
Ex
1022
117
26
1
2


ExterQual
Counts
ExterQual
TA
Gd
Ex
Fa
717
397
43
11

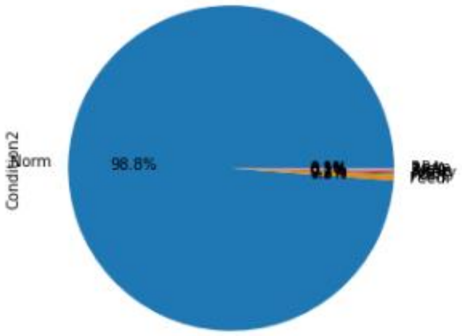

Exterior2nd
Counts
Exterior2nd
Plywood
Wd Sdng
MetalSd
CmentBd
VinylSd
HdBoard
Wd Shng
Stucco
ImStucc
Stone
BrkFace
AsbShng
Brk Cmn
AsphShn
Other
118
165
173
42
387
170
31
23
8
4
20
18
5
3
1

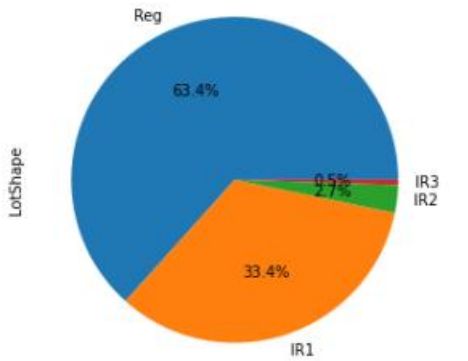

Foundation
Counts
Foundation
CBlock
PConc
BrkTil
Slab
Stone
Wood
516
513
112
21
5
1

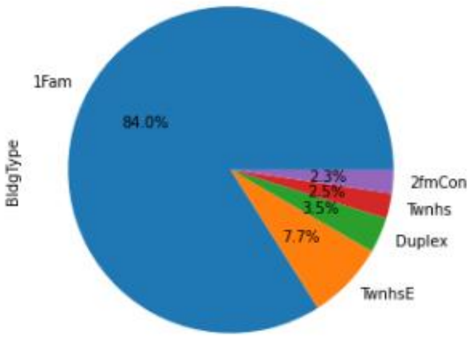
# Visualization

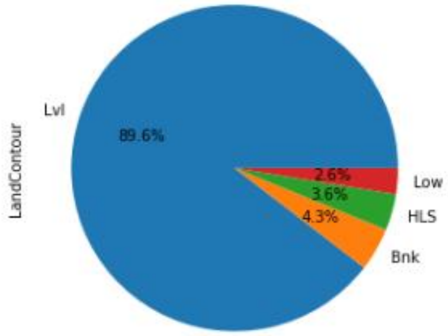


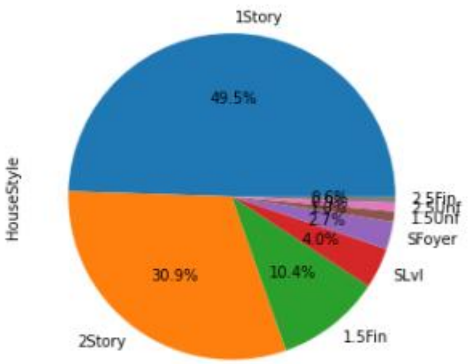
**Pie Chart**

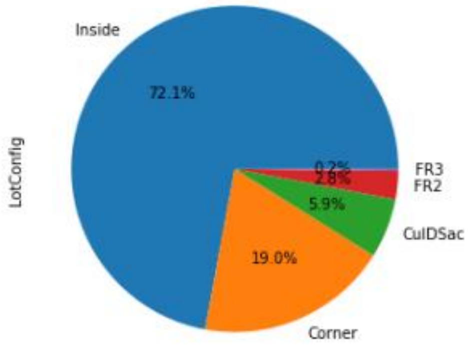

Condition2
Norm
98.8%

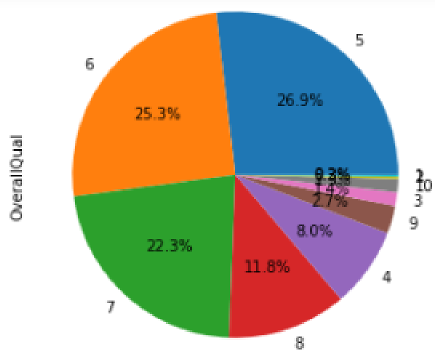

LotShape
Reg
63.4%
0.5%
2.7%
IR3
IR2
33.4%
IR1

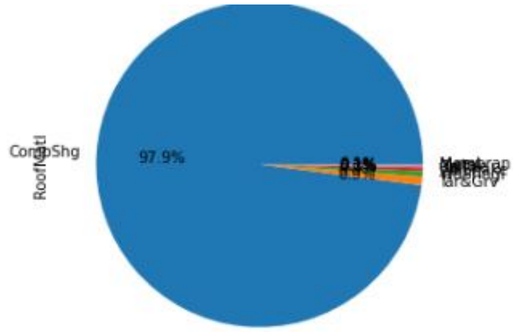

BldgType
1Fam
84.0%
2.3%
2.5%
3.5%
7.7%
2fmCon
Twnhs
Duplex
TwnhsE

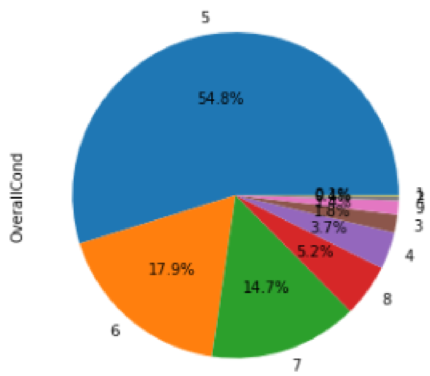

LandContour
Lvl
89.6%
2.6%
3.6%
4.3%
Low
HLS
Bnk

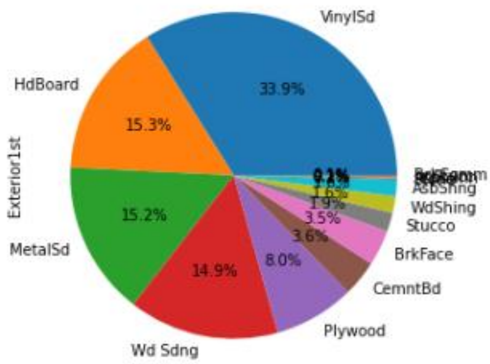

HouseStyle
1Story
49.5%
2.7%
4.0%
SFoyer
SLvl
30.9%
10.4%
2Story
1.5Fin

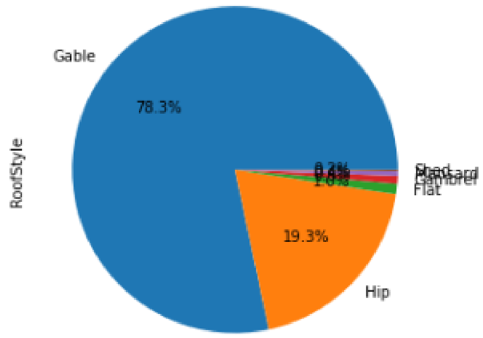

LotConfig
Inside
72.1%
0.2%
2.8%
FR3
FR2
5.9%
CulDSac
19.0%
Corner

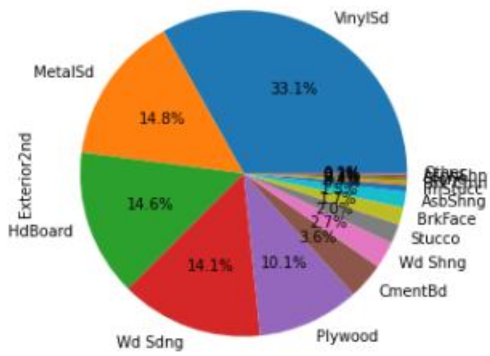

OverallQual
5
6
7
8
4
9
3
10
26.9%
25.3%
22.3%
11.8%
8.0%
2.7%
1.4%

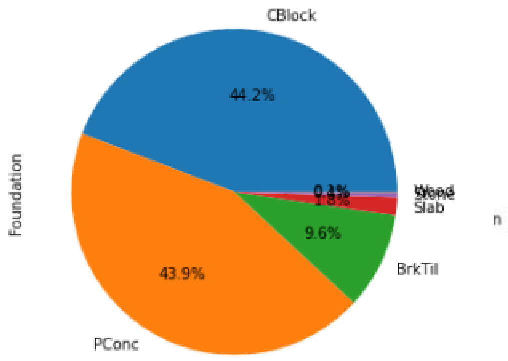

RoofMatl
CompShg
97.9%
Tar&GrV

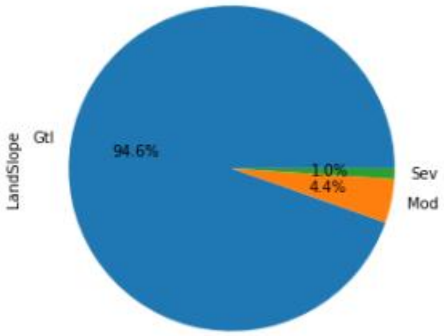

OverallCond
5
6
7
8
4
3
54.8%
17.9%
14.7%
5.2%
3.7%
1.8%

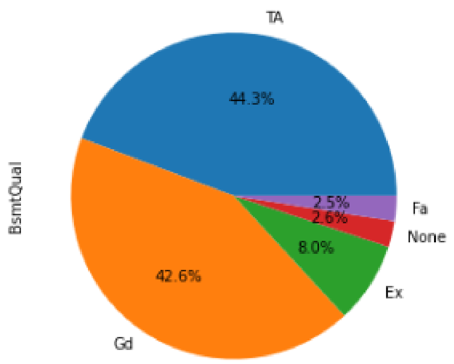

Exterior1st
VinylSd
HdBoard
MetalSd
Wd Sdng
Plywood
CemntBd
BrkFace
Stucco
WdShing
AsbShng
33.9%
15.3%
15.2%
14.9%
8.0%
3.6%
3.5%
1.9%
1.6%

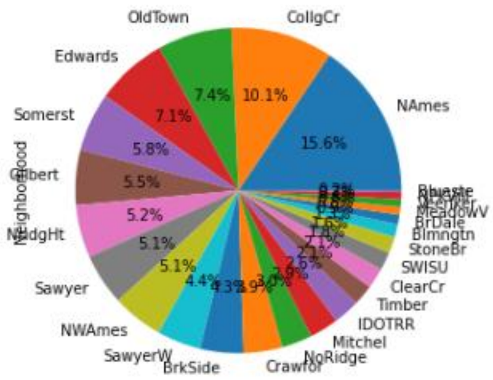

RoofStyle
Gable
Hip
Flat
Gambrel
Shed
78.3%
19.3%
1.0%

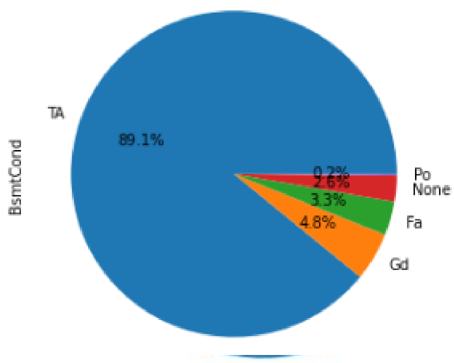

Exterior2nd
VinylSd
MetalSd
HdBoard
Wd Sdng
Plywood
CmentBd
Wd Shng
Stucco
BrkFace
AsbShng
33.1%
14.8%
14.6%
14.1%
10.1%
3.6%
2.7%
2.0%

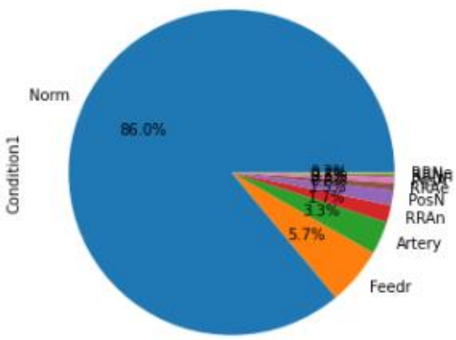

Foundation
CBlock
44.2%
PConc
43.9%
BrkTil
9.6%
Slab
1.6%
Stone
Wood

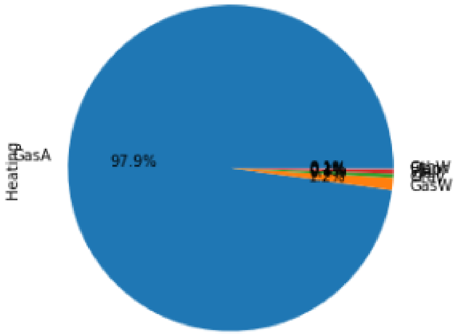

LandSlope
Gtl
94.6%
Mod
4.4%
Sev
1.0%

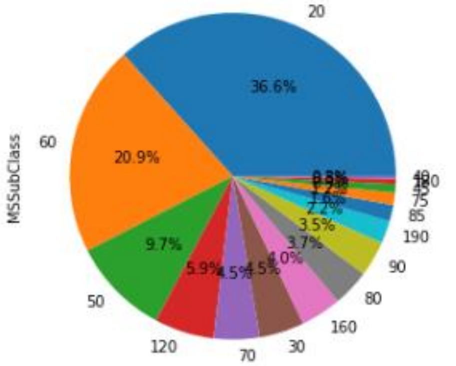

BsmtQual
TA
44.3%
Gd
42.6%
Ex
8.0%
None
2.6%
Fa
2.5%

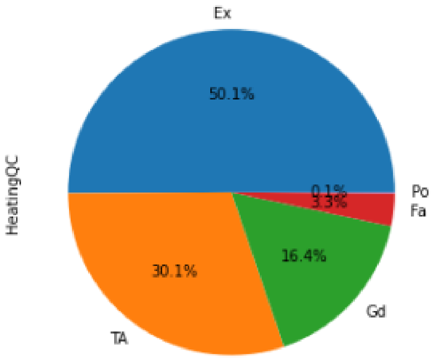

Neighborhood
NAmes
15.6%
CollgCr
10.1%
OldTown
7.4%
Edwards
7.1%
Somerst
5.8%
Gilbert
5.5%
NridgHt
5.2%
Sawyer
5.1%
NWAmes
5.1%
SawyerW
4.4%
BrkSide
Crawfor
NoRidge
Mitchel
IDOTRR
Timber
ClearCr
SWISU
StoneBr
Blmngtn
MeadowV
BrDale

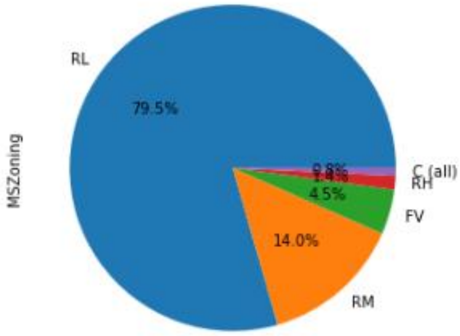

BsmtCond
TA
89.1%
Gd
4.8%
Fa
3.3%
None
2.6%
Po
0.2%

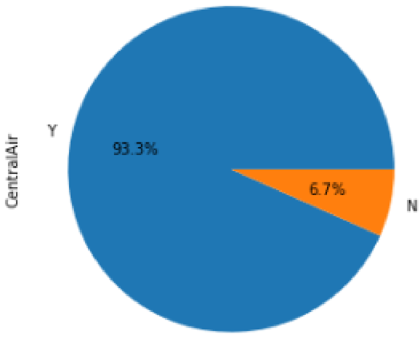

Condition1
Norm
86.0%
Feedr
5.7%
Artery
3.3%
RRAn
1.7%
PosN
RRAe

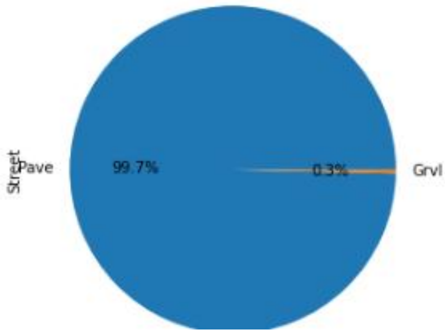

Heating
GasA
97.9%
GasW
1.2%

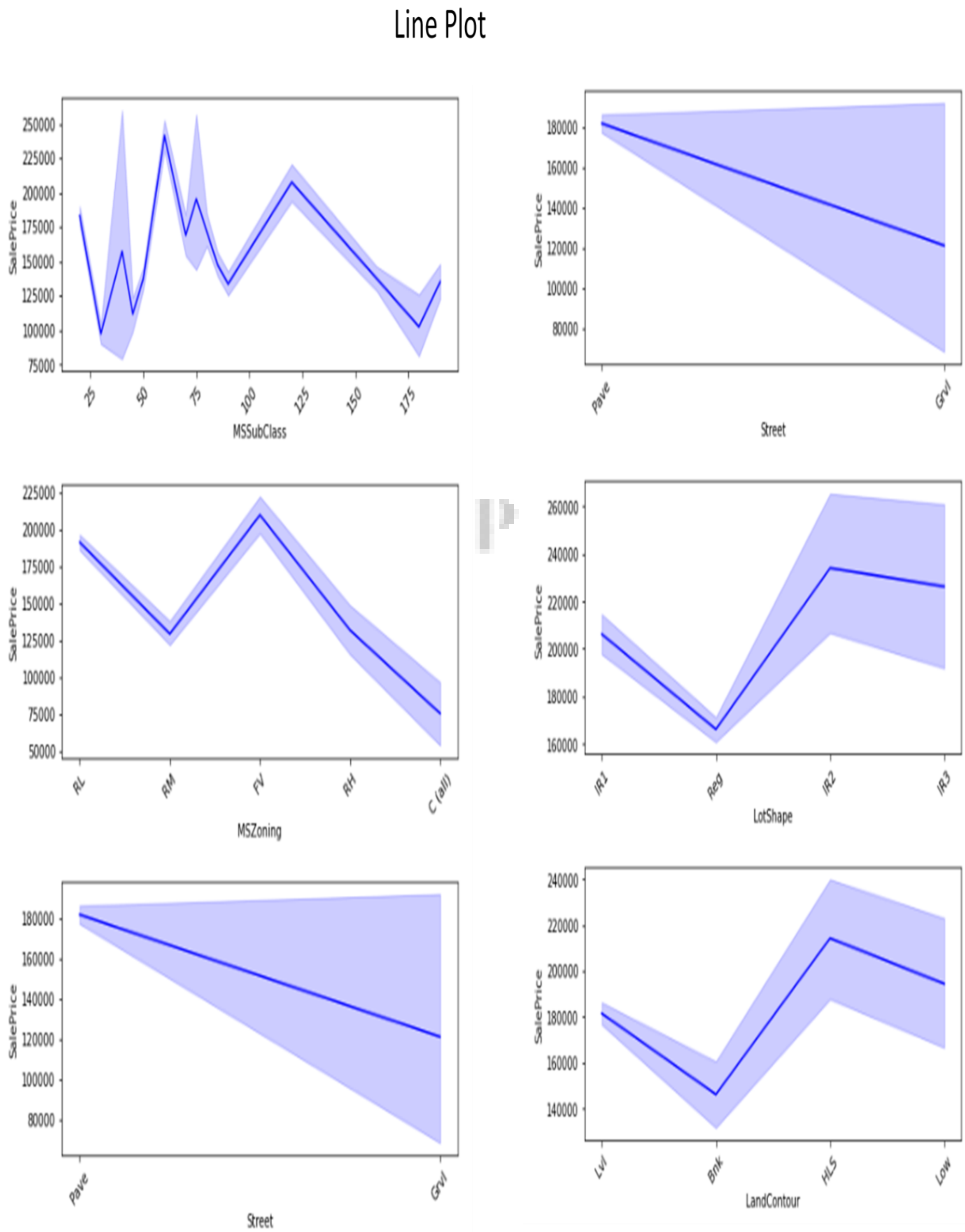

MSSubClass
20
36.6%
60
20.9%
50
9.7%
120
5.9%
70
30
160
80
90
190
85
75
3.5%
3.7%
4.0%


HeatingQC
Ex
50.1%
Po
Fa
0.1%
3.3%
16.4%
Gd
30.1%
TA

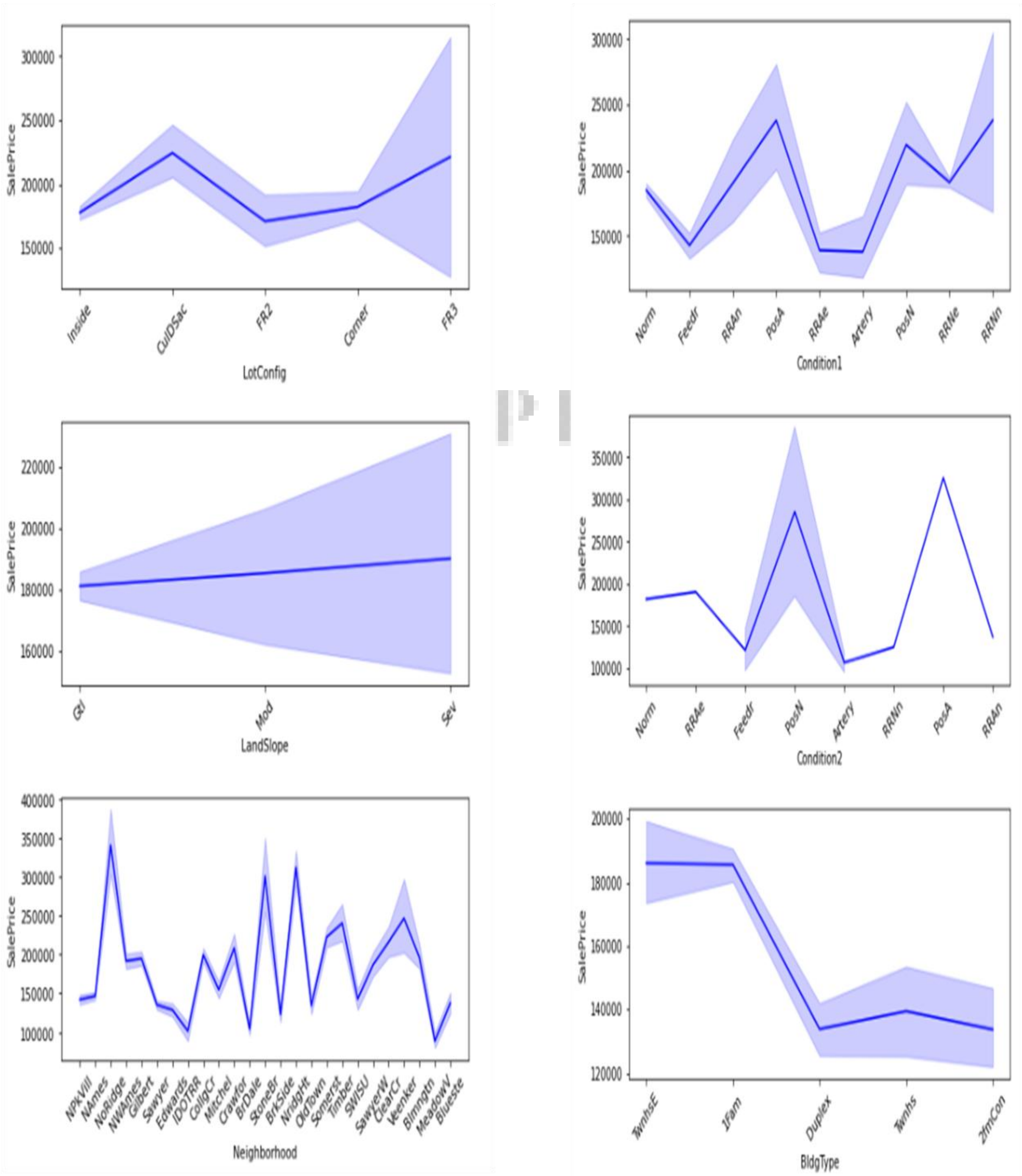

MSZoning
RL
79.5%
C (all)
RH
4.5%
FV
14.0%
RM


CentralAir
Y
93.3%
6.7%
N

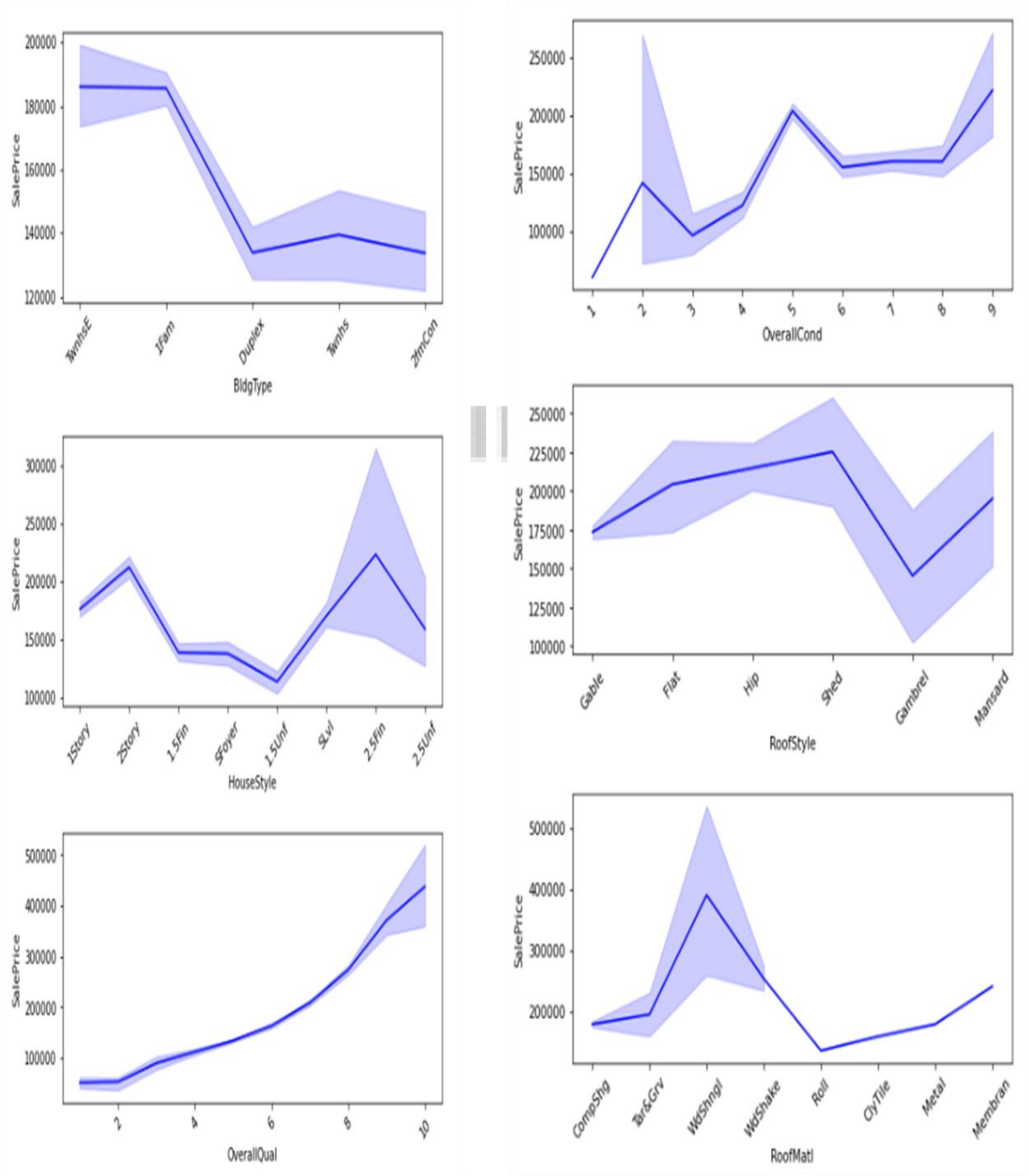

Street
Pave
99.7%
0.3%
Grvl


Line Plot
SalePrice
MSSubClass
250000
225000
200000
175000
150000
125000
100000
75000
25
50
75
100
125
150
175
Street
Pave
Grvl
MSZoning
RL
RM
FV
RH
C (all)
LotShape
IR1
Reg
IR2
IR3
LandContour
Lvl
Bnk
HLS
Low

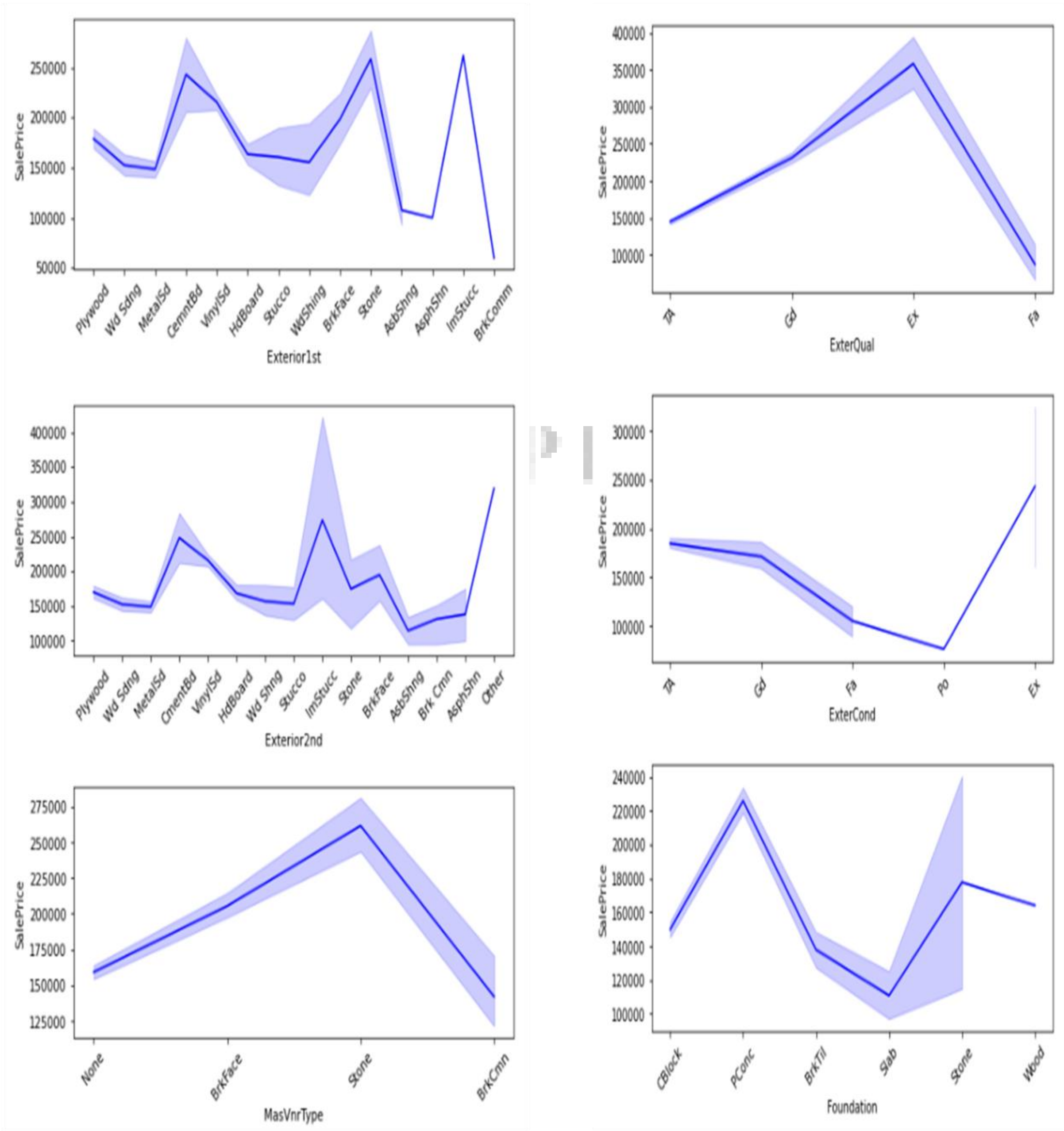
## Line Plot


SalePrice
LotConfig
Inside
CulDSac
FR2
Corner
FR3
Condition1
Norm
Feedr
RRAn
PosA
RRAe
Artery
PosN
RRNe
RRNn
LandSlope
Gtl
Mod
Sev
Condition2
Norm
RRAe
Feedr
PosN
Artery
RRNn
PosA
RRAn
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
BldgType
TwnhsE
1Fam
Duplex
Twnhs
2fmCon

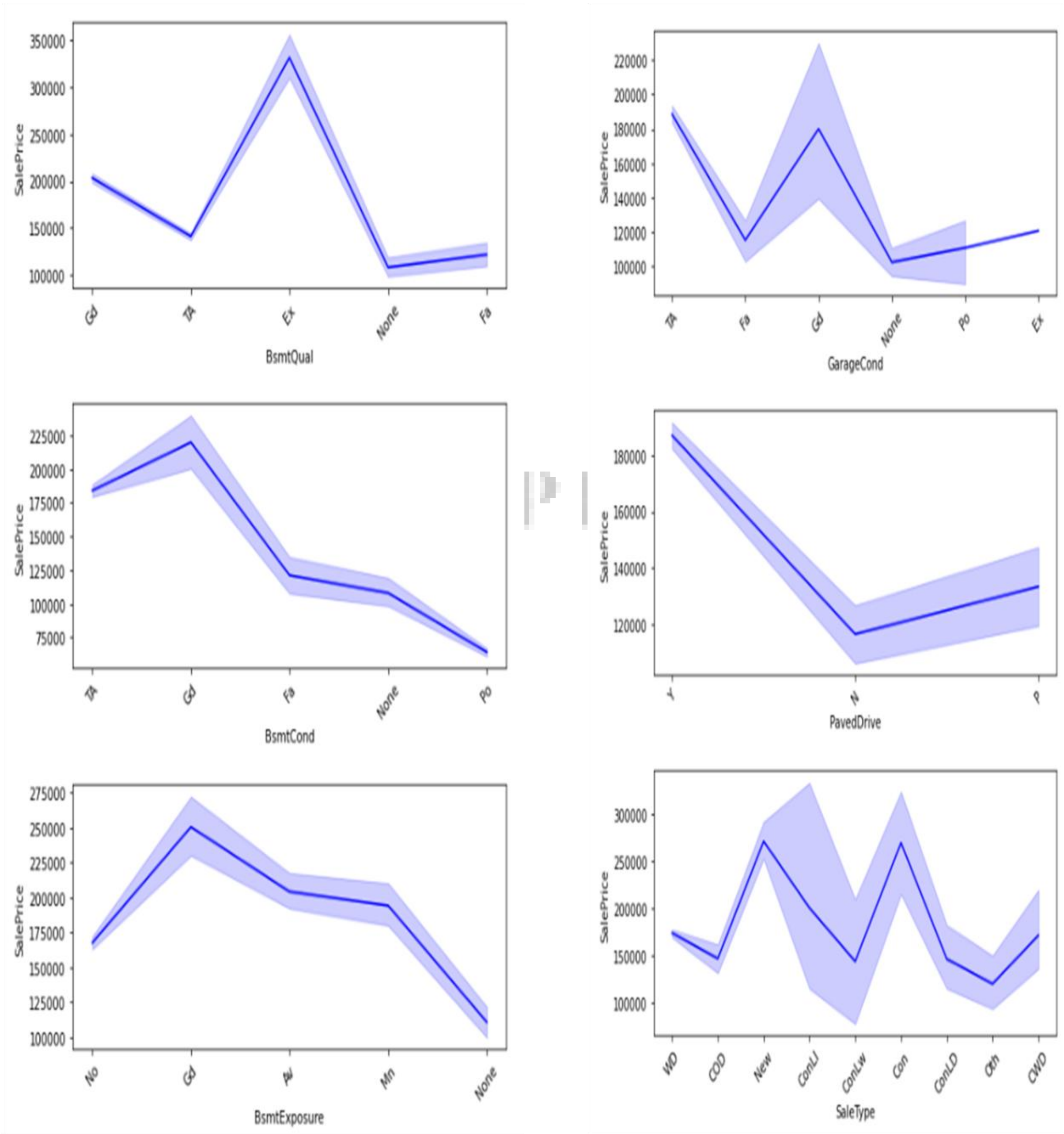
# Line Plot

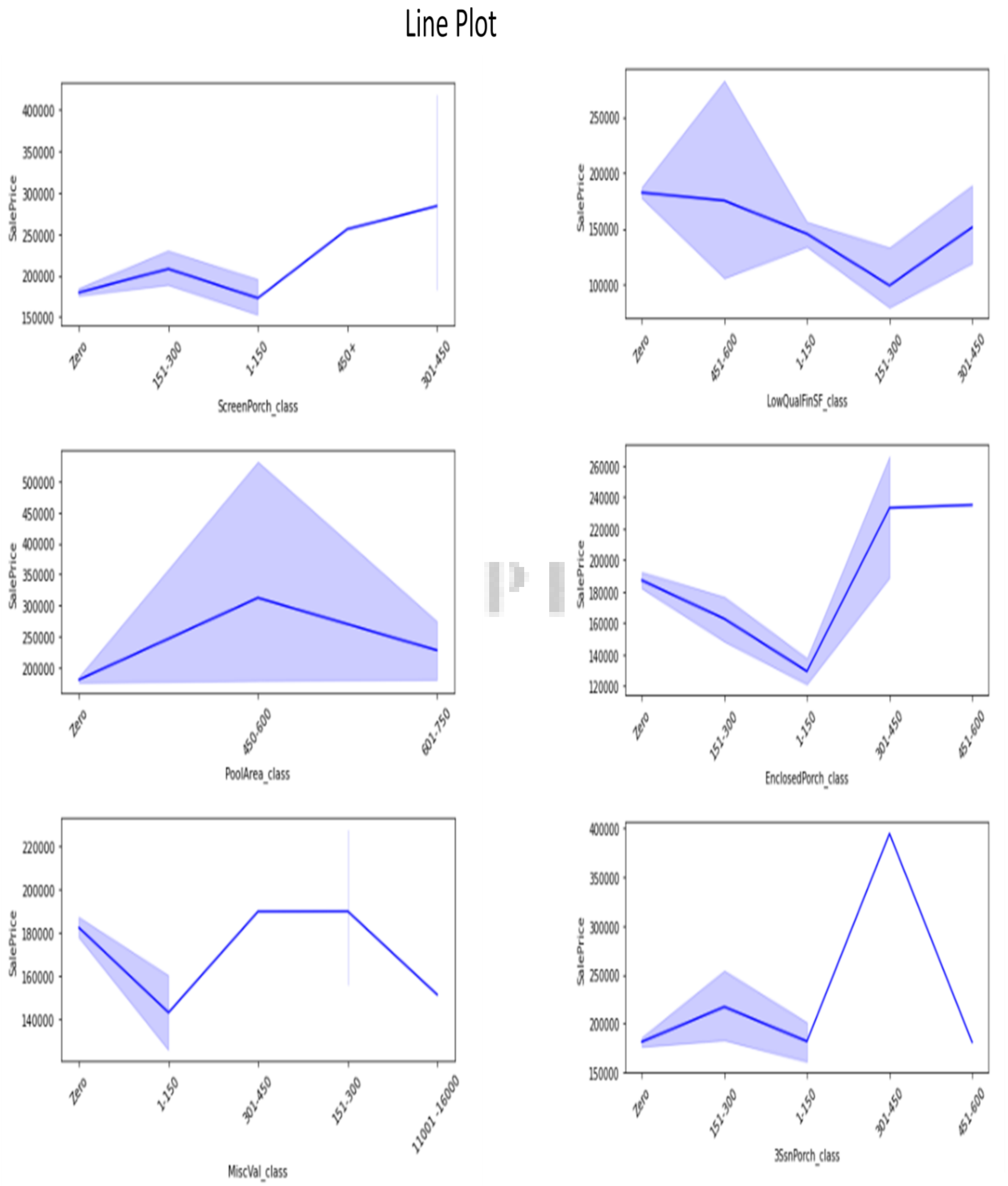

SalePrice
BldgType
TwnhsE
1Fam
Duplex
Twnhs
2fmCon
OverallCond
HouseStyle
1Story
2Story
1.5Fin
SFoyer
1.5Unf
SLvl
2.5Fin
2.5Unf
RoofStyle
Gable
Flat
Hip
Shed
Gambrel
Mansard
OverallQual
RoofMatl
CompShg
Tar&Grv
WdShngl
WdShake
Roll
ClyTile
Metal
Membran

## Line Plot

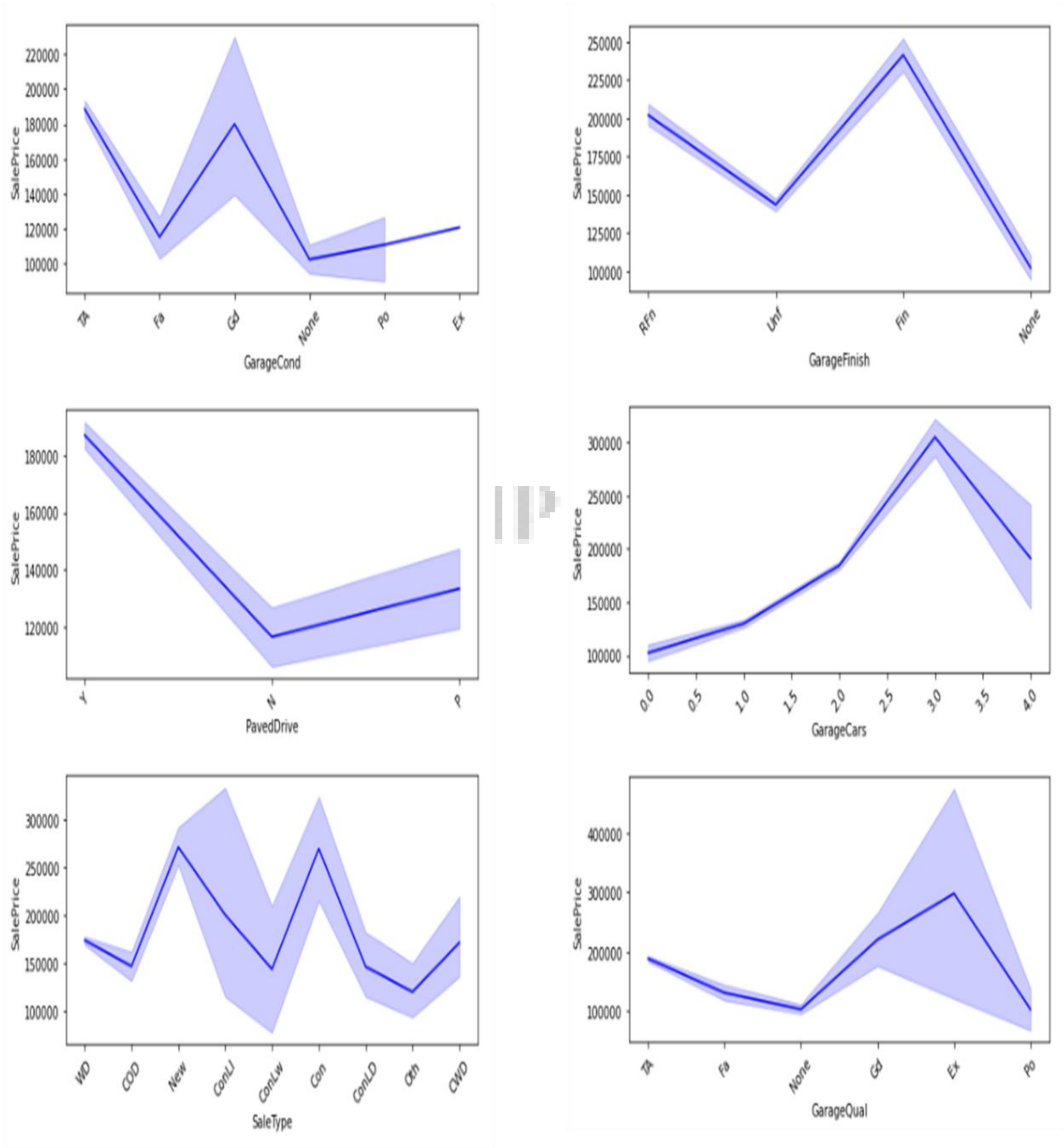

SalePrice
Exterior1st
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
ExterQual
TA
Gd
Ex
Fa
Exterior2nd
Plywood
Wd Sdng
MetalSd
CmentBd
VinylSd
HdBoard
Wd Shng
Stucco
ImStucc
Stone
BrkFace
AsbShng
Brk Cmn
AsphShn
Other
ExterCond
TA
Gd
Fa
Po
Ex
MasVnrType
None
BrkFace
Stone
BrkCmn
Foundation
CBlock
PConc
BrkTil
Slab
Stone
Wood

## Line Plot

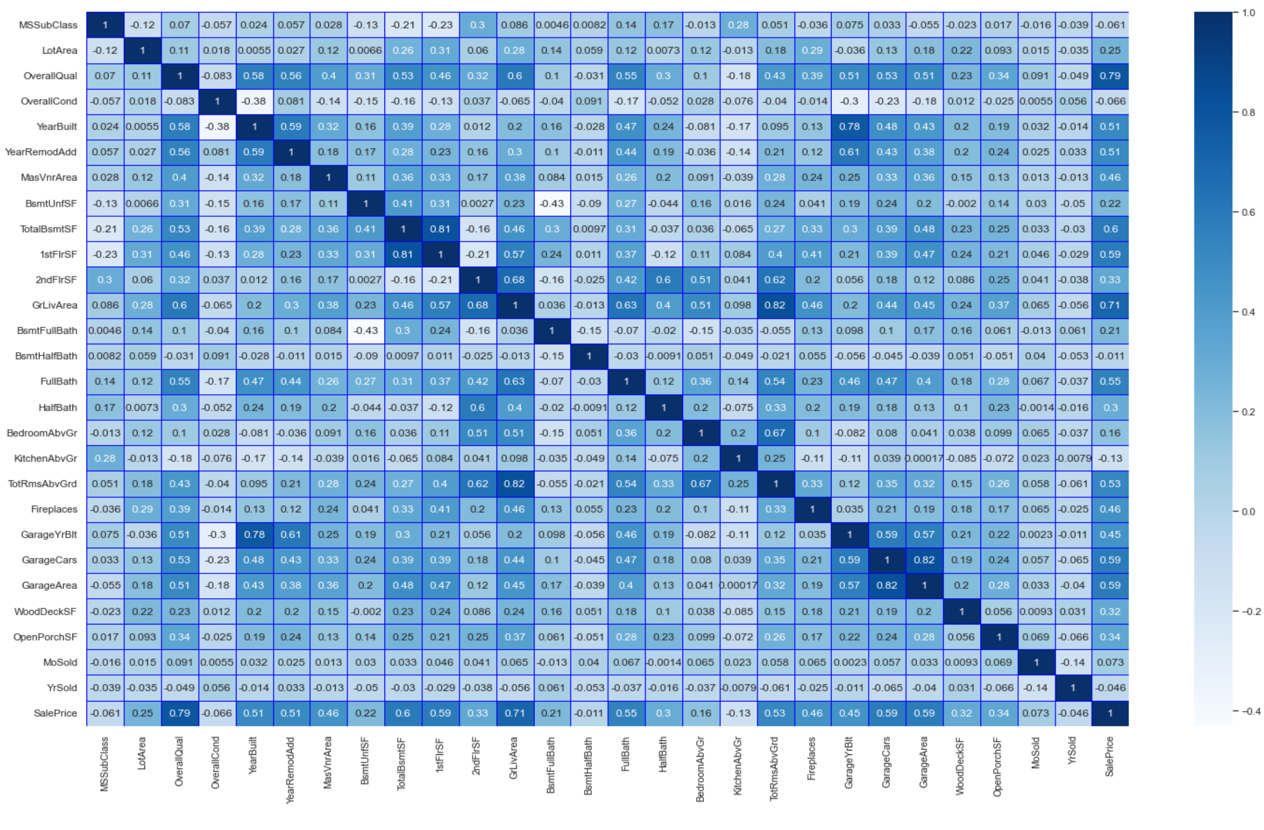

SalePrice
BsmtQual
Gd TA Ex None Fa
GarageCond
TA Fa Gd None Po Ex
BsmtCond
TA Gd Fa None Po
PavedDrive
Y N P
BsmtExposure
No Gd Av Mn None
SaleType
WD COD New ConLI ConLw Con ConLD Oth CWD


Line Plot
SalePrice
ScreenPorch_class
Zero
151-300
1-150
450+
301-450
LowQualFinSF_class
Zero
451-600
1-150
151-300
301-450
PoolArea_class
Zero
450-600
601-750
EnclosedPorch_class
Zero
151-300
1-150
301-450
451-600
MiscVal_class
Zero
1-150
301-450
151-300
11001-16000
3SsnPorch_class
Zero
151-300
1-150
301-450
451-600

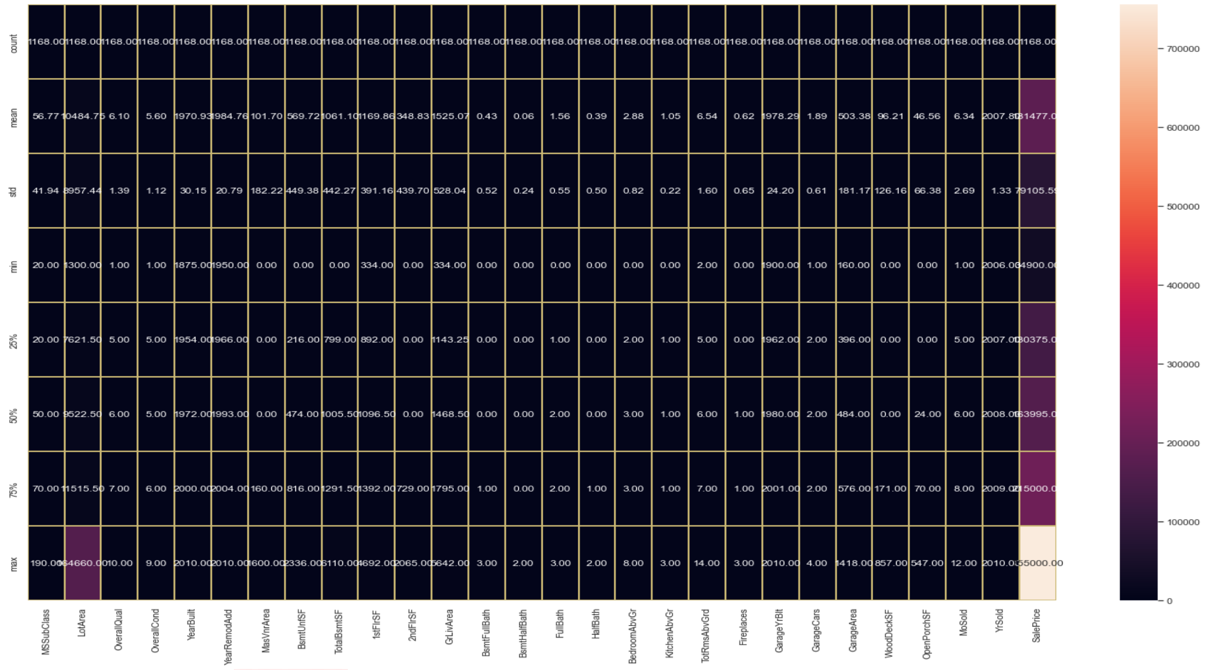
# Line Plot


SalePrice
GarageCond
TA
Fa
Gd
None
Po
Ex
GarageFinish
RFn
Unf
Fin
None
PavedDrive
Y
N
P
GarageCars
SaleType
WD
COD
New
ConLI
ConLw
Con
ConLD
Oth
CWD
GarageQual

# Correlation of the Dataset



## Describe of the Dataset




**Converting objects dataset into numerical form we are using Ordinal Encoder**

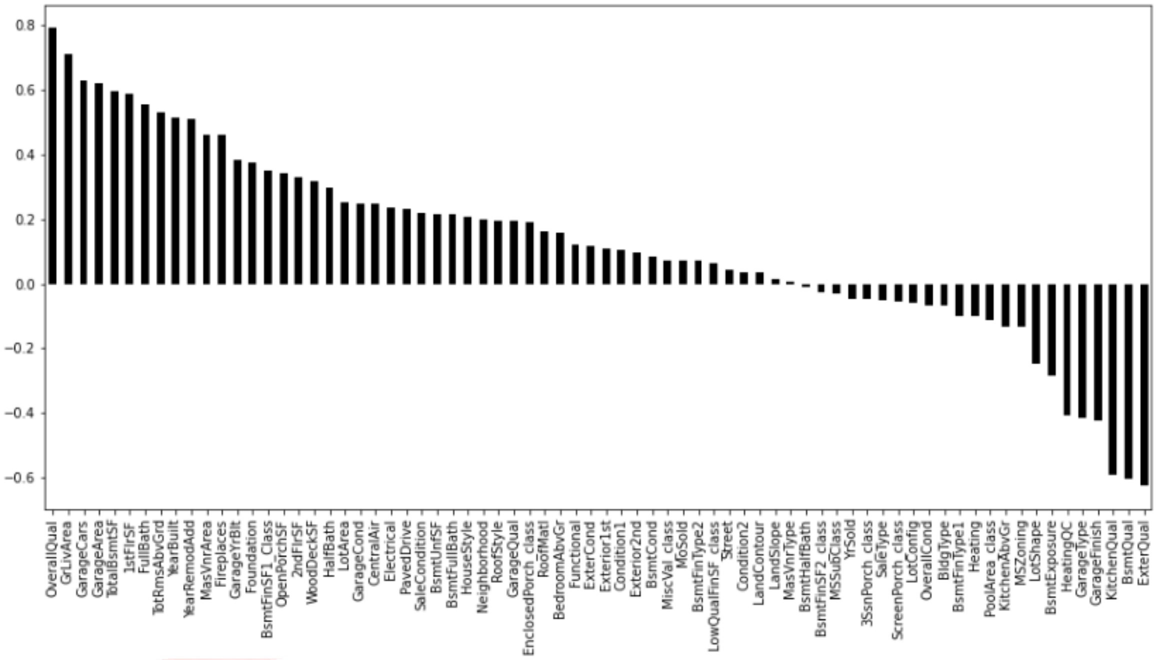
```
from sklearn.preprocessing import OrdinalEncoder
onc = OrdinalEncoder()
```

```
for i in df_train.select_dtypes(include = 'object').columns:
    df_train[i] = onc.fit_transform(df_train[i].values.reshape(-1,1))

for i in df_test.select_dtypes(include = 'object').columns:
    df_test[i] = onc.fit_transform(df_test[i].values.reshape(-1,1))
```

# Outliers

**We have applied Z score and Interquartile method for outlier removal but both shows very high amount of data loss upto 50 percent hence we can't consider it.**

# Checking Positive and Negative Correlation






# Dividing data for feature selection

```
x = df_train.drop('SalePrice', axis = 1)
y = df_train['SalePrice']
```

```
print('shape of x', x.shape)
print('Shape of y', y.shape)
```

```
shape of x (1168, 72)
Shape of y (1168,)
```

## Checking Mutlicollinearity

```
import statsmodels.api as sm
from scipy import stats
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
def calc_vif(x):

    vif = pd.DataFrame()
    vif['Variance'] = x.columns
    vif["VIF Factor"] = [variance_inflation_factor(x.values, i) for i in range(x.shape[1])]
    return vif
```

```
corr_col2 = ['LotArea', 'MasVnrArea', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF',
calc_vif(x[corr_col2])
```

| | Variance | VIF Factor |
|---|---|---|
| 0 | LotArea | 2.760564 |
| 1 | MasVnrArea | 1.605672 |
| 2 | BsmtUnfSF | 3.263419 |
| 3 | TotalBsmtSF | 24.041022 |
| 4 | 1stFlrSF | 26.744095 |
| 5 | 2ndFlrSF | 1.957954 |
| 6 | GarageArea | 8.799739 |
| 7 | WoodDeckSF | 1.783846 |
| 8 | OpenPorchSF | 1.752684 |
| 9 | MoSold | 4.970495 |

**Vif are morely like under acceptable zone as lower numerical dataset.**

## Removing Skewness

### Using Power Transformer method

```
from sklearn.preprocessing import PowerTransformer
pw = PowerTransformer('yeo-johnson')
```

```
x[corr_col2] = pw.fit_transform(x[corr_col2])
x[corr_col2].skew()
```

```
LotArea        0.032509
MasVnrArea     0.439526
BsmtUnfSF     -0.284390
TotalBsmtSF    0.286779
1stFlrSF      -0.002391
2ndFlrSF       0.280208
GarageArea    -0.320370
WoodDeckSF     0.113026
OpenPorchSF   -0.002749
MoSold        -0.035838
dtype: float64
```

## Standard Scalling

```
: from sklearn.preprocessing import StandardScaler
  sc = StandardScaler()
```

```
: scaler_col = ['LotArea', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GarageY
```

```
: x[scaler_col] = sc.fit_transform(x[scaler_col])
```

```
: df_test[scaler_col] = sc.fit_transform(df_test[scaler_col])
```

## Feature Selection

## Model Building and Results

```
# models                    R2_score_train_score    R2_score_test_score     CV score            CV_state

# KNeighborsRegressor       70.52780127415791       67.92035167996285       70.11645830235842      2
# DecisionTreeRegressor     100.0                   72.07707919431523       71.11165222915994      9
# XGBRegressor              99.99656969902267       88.74409777301683       84.2703914308628       3
# GradientBoostingRegressor 96.73484792121033       87.89327874400033       86.71369922910603      3
# LGBMRegressor             95.6832603079067        87.80425114146232       84.37347354281358      3
```

**- KNeighborsRegressor : Model shows low r2 score in training and testing accuracy hence we cannot consider it.**

**- DecisionTreeRegressor: Same as DecisionTreeRegressor shows very much difference in training and testing accuracy hence we cannot consider it. Model becames underfit.**

**- XGBRegressor: Same as above two model it shows very much difference in training and testing accuracy hence we cannot consider it. Model becames underfit.**

**- GradientBoostingRegressor: GradientBoostingRegressor shows closer R2 testing and training score but still training score it much greater than it testings R2 score which makes model underfit also R2 score not good yet from all models hence we can't consider it.**

**- LGBMRegressor : Moder shows very close R2 score of testing and training also CV score is also good hence we can consider it for model building.**

# Final Model LGBM Regressor

### Ensemble Method

## Hyper Parameter Tuning

```
model = LGBMRegressor()
# using hyper parameter tuning for Ridge regression to find out best criterion

# param (boosting_type: str = 'gbdt',
#     num_leaves: int = 31,
#     max_depth: int = -1,
#     learning_rate: float = 0.1,
#     n_estimators: int = 100,
#     subsample_for_bin: int = 200000,
#     objective: Union[str, Callable, NoneType] = None,
#     class_weight: Union[Dict, str, NoneType] = None,
#     min_split_gain: float = 0.0,
#     min_child_weight: float = 0.001,
#     min_child_samples: int = 20,
#     subsample: float = 1.0,
#     subsample_freq: int = 0,
#     colsample_bytree: float = 1.0,
#     reg_alpha: float = 0.0,
#     reg_lambda: float = 0.0,
#     random_state: Union[numpy.random.mtrand.RandomState, int, NoneType] = None,
#     n_jobs: int = -1,
#     silent: Union[bool, str] = 'warn',
#     importance_type: str = 'split',)
# by default params

param = {'boosting_type': ['gbdt', 'dart'], 'num_leaves': [31,41], 'max_depth': [-1,0], 'learning_rate': [0.1, 1],
        'n_estimators': [100], 'colsample_bytree': [1.0, 0.1],}
# using only important parameters.

gd = GridSearchCV(model, param_grid=param, cv = 8)
gd.fit(x, y)
gd.best_params_
```

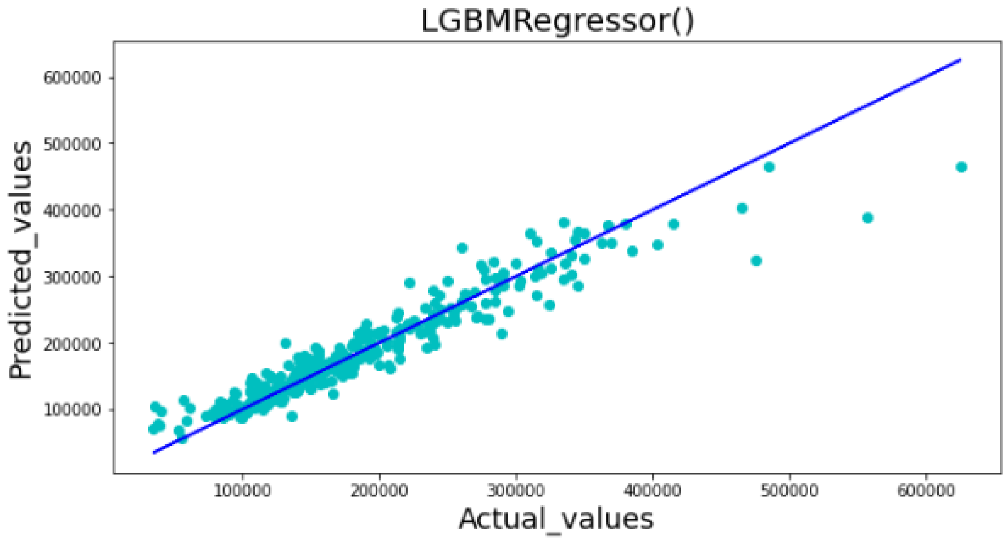
```
{'boosting_type': 'gbdt',
 'colsample_bytree': 0.1,
 'learning_rate': 0.1,
 'max_depth': -1,
 'n_estimators': 100,
 'num_leaves': 41}
```

```
final_model = LGBMRegressor(boosting_type = 'gbdt', colsample_bytree= 0.1, learning_rate=0.1, max_depth = -1, n_estimators=100, r
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.30, random_state = 135)
final_model.fit(x_train, y_train)
pred_train = final_model.predict(x_train)
pred_test = final_model.predict(x_test)
print("At random state", 135 , "model giving best accuracy score","\n")
Train_accuracy = r2_score(pred_train, y_train)
Test_accuracy = r2_score(pred_test, y_test)

print('Training accuracy:- ', Train_accuracy*100)
print('Testing accuracy:- ', Test_accuracy*100)
print("\n")
print('-------------------------------------')
print('Mean squared error:- ', mean_squared_error(pred_test, y_test) )
print('Mean absolute error:- ', mean_absolute_error(pred_test, y_test) )
print('Root Mean squared error:-',np.sqrt(mean_squared_error(pred_test, y_test)))

plt.figure(figsize = (10, 5))
plt.scatter(x = y_test, y = pred_test, color = 'c')
plt.plot(y_test, y_test, color = 'b')
plt.xlabel('Actual_values', fontsize= 18 )
plt.ylabel('Predicted_values', fontsize = 18)
plt.title(str(model), fontsize = 20)
```

```
At random state 135 model giving best accuracy score

Training accuracy:-  91.65038262533558
Testing accuracy:-  87.79961573185521


--------------------------------------
Mean squared error:-  676201558.6377157
Mean absolute error:-  16769.15395469789
Root Mean squared error:- 26003.875838761338
```

```
: Text(0.5, 1.0, 'LGBMRegressor()')
```




### Cross Val Score

```
: cross_val_score(final_model, x, y, cv = 3).mean()
```

```
: 0.844315554673861
```

# Model Deployment

### Deploy Model

```
import pickle

filename = "Housingprice.pkl"
pickle.dump(final_model, open(filename, 'wb'))
```

### Loading Model

```
load = pickle.load(open('Housingprice.pkl', 'rb'))
result = load.score(x_test, y_test)
print(result)
```

```
0.9046302975945946
```

```
conclusion = pd.DataFrame()
conclusion['Predicted House price'] = np.array(final_model.predict(x_test))
conclusion['Actual House price'] = np.array(y_test)
```

```
conclusion.sample(10)
```

| | Predicted House price | Actual House price |
|---|---|---|
| 202 | 122236.067578 | 112500 |
| 178 | 128810.668947 | 119200 |
| 228 | 136271.277249 | 140000 |
| 42 | 389302.622393 | 556581 |
| 89 | 234721.661016 | 225000 |
| 276 | 240859.071317 | 272000 |
| 253 | 316799.002176 | 274000 |
| 119 | 214577.141973 | 193000 |

# Prediction for test dataset

### Prediction for test dataset

```
df_test.head(1)
```

| | MSSubClass | MSZoning | LotArea | Street | LotShape | LandContour | LotConfig | LandSlope | Neighborhood | Condition1 | Condition2 | BldgType | HouseStyle | Ove |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 2.0 | 0.263894 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 21.0 | 2.0 | 0.0 | 0.0 | 2.0 | |

```
predicted_price =  pickle.load(open('Housingprice.pkl', 'rb'))  # Loading price_predictor model
```

```
predicted_house_price = np.array(predicted_price.predict(df_test))
```

```
df_test['Predicted_House_price'] = predicted_house_price
```

```
df_test.sample(10)
```

| | MSSubClass | MSZoning | LotArea | Street | LotShape | LandContour | LotConfig | LandSlope | Neighborhood | Condition1 | Condition2 | BldgType | HouseStyle |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 124 | 4.0 | 2.0 | 0.021405 | 1.0 | 3.0 | 0.0 | 4.0 | 0.0 | 11.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 148 | 0.0 | 3.0 | 0.154560 | 1.0 | 3.0 | 3.0 | 4.0 | 0.0 | 16.0 | 2.0 | 0.0 | 0.0 | 2.0 |
| 263 | 2.0 | 3.0 | -0.394140 | 1.0 | 3.0 | 3.0 | 0.0 | 0.0 | 16.0 | 2.0 | 0.0 | 0.0 | 2.0 |
| 182 | 0.0 | 2.0 | 0.000816 | 1.0 | 0.0 | 1.0 | 4.0 | 0.0 | 22.0 | 2.0 | 0.0 | 0.0 | 2.0 |
| 183 | 12.0 | 1.0 | -0.606646 | 1.0 | 3.0 | 3.0 | 2.0 | 0.0 | 20.0 | 2.0 | 0.0 | 3.0 | 5.0 |
| 21 | 0.0 | 2.0 | -0.183963 | 1.0 | 3.0 | 3.0 | 4.0 | 0.0 | 18.0 | 2.0 | 0.0 | 0.0 | 2.0 |
| 23 | 14.0 | 3.0 | -0.379111 | 1.0 | 3.0 | 3.0 | 4.0 | 0.0 | 8.0 | 2.0 | 0.0 | 1.0 | 5.0 |
| 129 | 14.0 | 3.0 | -0.465076 | 1.0 | 3.0 | 3.0 | 4.0 | 0.0 | 16.0 | 2.0 | 0.0 | 1.0 | 5.0 |
| 86 | 1.0 | 2.0 | 0.011636 | 1.0 | 3.0 | 3.0 | 4.0 | 0.0 | 16.0 | 2.0 | 0.0 | 0.0 | 2.0 |
| 35 | 8.0 | 2.0 | -0.050958 | 1.0 | 0.0 | 3.0 | 4.0 | 0.0 | 7.0 | 2.0 | 0.0 | 0.0 | 7.0 |

# ➢ Hardware and Software Requirements and Tools Used

**Operating System: Window 11**

**RAM: 8 GB**

**Processor: i5 10th Generation**

**Software: Jupyter Notebook**

**Python Libraries: Mainly**

**Pandas: This library used for dataframe operations .**

**Numpy: This library gives statistical computation for smooth functioning .**

**Matplotlib: Used for visualization.**

**Seaborn: This library is also used for visualization.**

**Sklearn: This library having so many machine learning module and we can import them from this library.**

**Pickle: This is used for deploying the model.**

**Xgboost: Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library**

**Lightgbm: Light version of Gradient Boosting Machine.**

# CONCLUSION

## ➢ Key Findings and Conclusions of the Study

**This project has built a model that can predict upcoming Sale Prices of House. For this company can reduces loses in Investment. The challenge behind Sale Price finding in machine learning is the number of features in dataset. Also some other issues like imputation understandings and so many values are zeros.**

## ➢ Learning Outcomes of the Study in respect of Data Science

**Data cleaning is the most important part in this model building as we see above there are so many NULL values we fill with imputation and ranges some of column dataset for better observations. This project has gives so much information about parameters that how a single parameter can increase or decrease prices of house.**

## ➢ Limitations of this work and Scope for Future Work

**Model work with similar parameters as we build the whole model if some of the parameters missed then we need to train model with remains parameter after that we can predict upcoming sales of houses hence we need to up to date all the parameters as per training dataset.**