HOUSING : PRICE PREDICTION

Submitted by:
Arjita Saxena

ACKNOWLEDGMENT

I would like to express my sincere thanks to Ms. Swati Mahaseth for her timely and valuable support and guidance in completing my project.

I would also like to express my gratitude towards my family & members of Flip Robo Technologies for their kind co-operation and encouragement which helped me in completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me this great opportunity to do a project on ‘Housing Price Prediction’.

INTRODUCTION

- ## Business Problem Framing

  Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modeling, Market mix modeling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies.

- ## Conceptual Background of the Domain Problem

  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 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?

- ## Review of Literature

  We are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

- ## Motivation for the Problem Undertaken

  In today's world, everyone wishes for a house that suits their lifestyle and provides amenities according to their needs. House prices keep on changing very frequently which proves that house prices are often exaggerated. There are many factors that have to be taken into consideration for predicting house prices such as location, number of rooms, carpet area, how old the property is and other basic local amenities. Our aim is to use machine learning algorithms to develop models for predicting the house prices accordingly.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

In the project below are the mathematical, statistical and analytics modelling done:

- Data Pre-processing :
  To ensure high-quality data, it's crucial to pre-process it. To make the process easier, data pre-processing is divided into four stages: data cleaning, data integration, data reduction, and data transformation.

- Exploratory Data Analysis:
  Exploratory Data Analysis (EDA) is the crucial process of using summary statistics and graphical representations to perform preliminary investigations on data in order to uncover patterns, detect anomalies, test hypotheses, and verify assumptions.

- Data cleaning :
  Data cleaning is the process of ensuring data is correct, consistent and usable. You can clean data by identifying errors or corruptions, correcting or deleting them, or manually processing data as needed to prevent the same errors from occurring.

- Correlation :
  Correlation is a statistical measure. Correlation explains how one or more variables are related to each other. These variables can be input data features which have been used to forecast our target variable.

- Data Summary :
  The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

- Statistic summary :
  It is used to view some basic statistical details like percentile, mean, standard deviation etc. of a data frame or a series of numeric values. When this method is applied to a series of string, it returns a different output like count of values, unique values, top and frequency of occurrence in this case.

- Data Sources and their formats

```
# Checking dataset schema
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1168 entries, 0 to 1167
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1168 non-null   int64
 1   MSSubClass     1168 non-null   int64
 2   MSZoning       1168 non-null   object
 3   LotFrontage    954 non-null    float64
 4   LotArea        1168 non-null   int64
 5   Street         1168 non-null   object
 6   Alley          77 non-null     object
 7   LotShape       1168 non-null   object
 8   LandContour    1168 non-null   object
 9   Utilities      1168 non-null   object
 10  LotConfig      1168 non-null   object
 11  LandSlope      1168 non-null   object
 12  Neighborhood   1168 non-null   object
 13  Condition1     1168 non-null   object
 14  Condition2     1168 non-null   object
 15  BldgType       1168 non-null   object
 16  HouseStyle     1168 non-null   object
 17  OverallQual    1168 non-null   int64
 18  OverallCond    1168 non-null   int64
 19  YearBuilt      1168 non-null   int64
 20  YearRemodAdd   1168 non-null   int64
 21  RoofStyle      1168 non-null   object
 22  RoofMatl       1168 non-null   object
 23  Exterior1st    1168 non-null   object
 24  Exterior2nd    1168 non-null   object
 25  MasVnrType     1161 non-null   object
 26  MasVnrArea     1161 non-null   float64
 27  ExterQual      1168 non-null   object
 28  ExterCond      1168 non-null   object
 29  Foundation     1168 non-null   object
 30  BsmtQual       1138 non-null   object
```

```
31  BsmtCond       1138 non-null   object
32  BsmtExposure   1137 non-null   object
33  BsmtFinType1   1138 non-null   object
34  BsmtFinSF1     1168 non-null   int64
35  BsmtFinType2   1137 non-null   object
36  BsmtFinSF2     1168 non-null   int64
37  BsmtUnfSF      1168 non-null   int64
38  TotalBsmtSF    1168 non-null   int64
39  Heating        1168 non-null   object
40  HeatingQC      1168 non-null   object
41  CentralAir     1168 non-null   object
42  Electrical     1168 non-null   object
43  1stFlrSF       1168 non-null   int64
44  2ndFlrSF       1168 non-null   int64
45  LowQualFinSF   1168 non-null   int64
46  GrLivArea      1168 non-null   int64
47  BsmtFullBath   1168 non-null   int64
48  BsmtHalfBath   1168 non-null   int64
49  FullBath       1168 non-null   int64
50  HalfBath       1168 non-null   int64
51  BedroomAbvGr   1168 non-null   int64
52  KitchenAbvGr   1168 non-null   int64
53  KitchenQual    1168 non-null   object
54  TotRmsAbvGrd   1168 non-null   int64
55  Functional     1168 non-null   object
56  Fireplaces     1168 non-null   int64
57  FireplaceQu    617 non-null    object
58  GarageType     1104 non-null   object
59  GarageYrBlt    1104 non-null   float64
60  GarageFinish   1104 non-null   object
61  GarageCars     1168 non-null   int64
62  GarageArea     1168 non-null   int64
63  GarageQual     1104 non-null   object
64  GarageCond     1104 non-null   object
65  PavedDrive     1168 non-null   object
```

```
 66  WoodDeckSF     1168 non-null   int64
 67  OpenPorchSF    1168 non-null   int64
 68  EnclosedPorch  1168 non-null   int64
 69  3SsnPorch      1168 non-null   int64
 70  ScreenPorch    1168 non-null   int64
 71  PoolArea       1168 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          237 non-null    object
 74  MiscFeature    44 non-null     object
 75  MiscVal        1168 non-null   int64
 76  MoSold         1168 non-null   int64
 77  YrSold         1168 non-null   int64
 78  SaleType       1168 non-null   object
 79  SaleCondition  1168 non-null   object
 80  SalePrice      1168 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 739.2+ KB
```

The 'SalePrice' column in the dataset is the target column that is having numeric values; hence this is a Regression Model.

- Data Pre-processing Done
    - Changing columns data type to the relevant one
    - Dropping irrelevant columns that are not contributing much in the model learning
    - Removing or Replacing invalid/null values
    - Checking and removing duplicates present in the dataset
    - Treating outliers and skewness
    - Checked and treated Multi-colinearity
    - Scaling

- Data Inputs- Logic- Output Relationships

FEATURES :-
'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', 'SalePrice'.

EnclosedPorch has good correlation with YearBuilt, GarageYrBlt. OpenPorchSF has good correlation with OverallQual, EnclosedPorch has good correlation with YearBuilt, GarageYrBlt. OpenPorchSF has good correlation with OverallQual, FullBath, HalfBath, GarageCars, YearBuilt, TotRmsAbvGrd, GarageYrBlt, ExterQual, BsmtQual,GarageArea. Foundation is highly correlated with GarageYrBlt. Fireplaces and FireplaceQu are strong positively correlated.

Target column 'SalePrice' has strong positive linear correlation with OverallQual, FullBath, TotRmsAbvGrd, Fireplaces, GarageCars, YearBuilt, YearRemodAdd, GarageYrBlt, Foundation, BsmtQual, HeatingQC, KitchenQual, FireplaceQu, TotalBsmtSF, 1stFlrSF, GrLivArea, GarageArea, OpenPorchSF and has strong negative linear correlation with GarageType, GarageFinish. 'SalePrice' has good linear correlation with HalfBath, BsmtExposure, BsmtFinType1, WoodDeckSF.

- State the set of assumptions (if any) related to the problem under consideration
  - There are null values in the dataset and they are treated well as per the domain knowledge and understanding
  - For some features, there may be values which might not be realistic. We have observed them and treat them with a suitable explanation.
  - We came across outliers in some features but since the data in those particular features are important we dealt with it accordingly.
  - Skewness was found in a few features which we handled as per our understanding and relevant threshold.

- Hardware and Software Requirements and Tools Used

Hardware & Software requirements:

  - Modern Operating System (**Windows:** 7 or newer)
  - x86 64-bit CPU (Intel / AMD architecture)
  - Memory (RAM): Recommended 4 GB or above, 5 GB free disk space.
  - Processor: Minimum 1 GHz; Recommended 2GHz or more
  - Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
  - Hard Drive: Minimum 32 GB; Recommended 64 GB or more

Tools, libraries and packages used :

  - Tools : Jupyter Notebook (Anaconda), Python, PIP 2.7
  - Libraries : Pandas, Numpy, Matplotlib, Seaborn, SciPy, Scikit-learn
  - Statsmodels, imblearn, pickle, joblib.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  - Problem Formulation and understanding
  - Data Preparation and Pre-processing
  - Feature Engineering and Selection
  - Exploratory Data Analysis
  - Data Cleaning
  - Model Development
  - Hyper parameter Tuning and Cross-Validation
  - Model Evaluation
  - Conclusion

- Testing of Identified Approaches (Algorithms)
  - LinearRegression
  - DecisionTreeRegressor
  - KNeighborsRegressor
  - RandomForestRegressor
  - GradientBoostingRegressor

- Run and Evaluate selected models

```
LinearRegression()
Accuracy(Training) :  86.84477930437464  Accuracy(Test)  89.10520427685388
mean_absolute_error 18075.463027667374
mean_squared_error 647441777.9052033


DecisionTreeRegressor()
Accuracy(Training) :  100.0  Accuracy(Test)  75.7978512082764
mean_absolute_error 24938.776435045318
mean_squared_error 1438253882.039275


KNeighborsRegressor()
Accuracy(Training) :  86.47201535894533  Accuracy(Test)  85.75954242215478
mean_absolute_error 20954.74259818731
mean_squared_error 846263427.6653776


RandomForestRegressor()
Accuracy(Training) :  98.06050817030099  Accuracy(Test)  89.58443489961705
mean_absolute_error 17007.804561933535
mean_squared_error 618962682.5359138


GradientBoostingRegressor()
Accuracy(Training) :  97.33779985832217  Accuracy(Test)  91.64451999248149
mean_absolute_error 15896.000258603215
mean_squared_error 496538619.7949743
```

GradientBoostingRegressor() has the highest accuracy among all the models selected for model training.

- Key Metrics for success in solving problem under consideration

Cross validation Score is a technique applied on a fixed number of folds (or partitions) of the data is made to run the analysis on each fold, and then average the overall error estimate that is used to protect against over-fitting issues in a predictive model.

```
# Computing cross validation score of all the models used
from sklearn.model_selection import cross_val_score

for i in algo_list :
    print('CV mean of ',i,' is ',np.abs(cross_val_score(i,x2,y,cv=7).mean()*100))
```

```
CV mean of  LinearRegression()  is  85.8374622653784
CV mean of  DecisionTreeRegressor()  is  72.69230684434298
CV mean of  KNeighborsRegressor()  is  83.24220589982089
CV mean of  RandomForestRegressor()  is  87.3679210651431
CV mean of  GradientBoostingRegressor()  is  89.05638575499157
```

```
Finding out the difference of Accuracy and Cross validation mean of all the models used
   Accuracy CVmean Diff
LR   89       86    3
DTR  76       73    3
KNN  86       83    3
RFR  90       87    3
GBR  92       89    3
```

Looking at difference of accuracy and cv mean, LinearRegression, RandomForestRegressor and GradientBoostingRegressor are showing good accuracy and almost similar difference between accuracy and cv mean.

Hyper parameter Tuning is used for better accuracy and to avoid over fitting issues with best parameters on selected model.

```
# Using hyper parameter tuning on GradientBoostingRegressor model for better accuracy and to avoid overfitting issues

parameters = {'loss':['squared_error', 'absolute_error', 'huber', 'quantile'],
              'criterion':['friedman_mse', 'squared_error', 'mse', 'mae'],
              'max_features':['auto','sqrt','log2',None],
              'max_depth':[None,3,5,10]}

gb=GradientBoostingRegressor()
GCV=GridSearchCV(gb,parameters,cv=7)
GCV.fit(x_train,y_train)
GCV.best_params_
```

```
{'criterion': 'mse', 'loss': 'huber', 'max_depth': 3, 'max_features': 'sqrt'}
```

```
gb=GradientBoostingRegressor(criterion='mae',max_depth=3,max_features='log2')
gb.fit(x_train,y_train)
pred=gb.predict(x_test)
acc=r2_score(y_test,pred)
cv=cross_val_score(gb,x2,y,cv=7).mean()

print('Accuracy : ',acc,' CV mean : ',cv)
```

```
Accuracy :  0.9159722145447942  CV mean :  0.8935979364761242
```

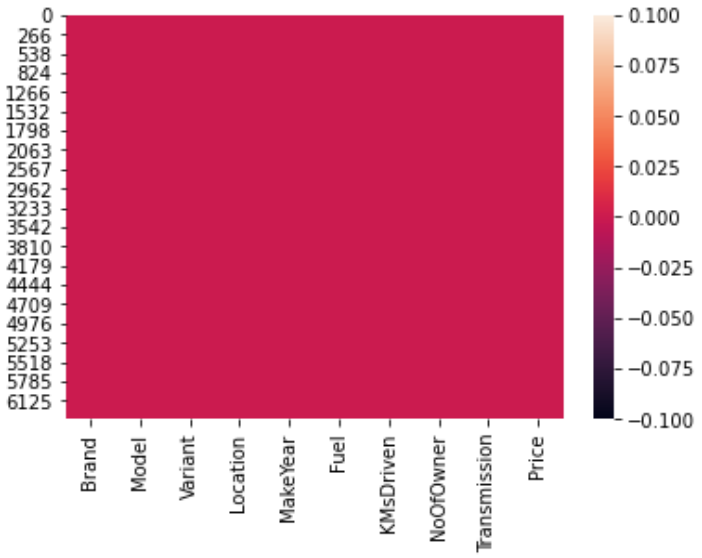
We are getting model accuracy as approx 92% and cv mean as approx 89% for model GradientBoostingRegressor, and it shows our model is performing good.

- Visualizations

  - **Checking for null values in the dataset**

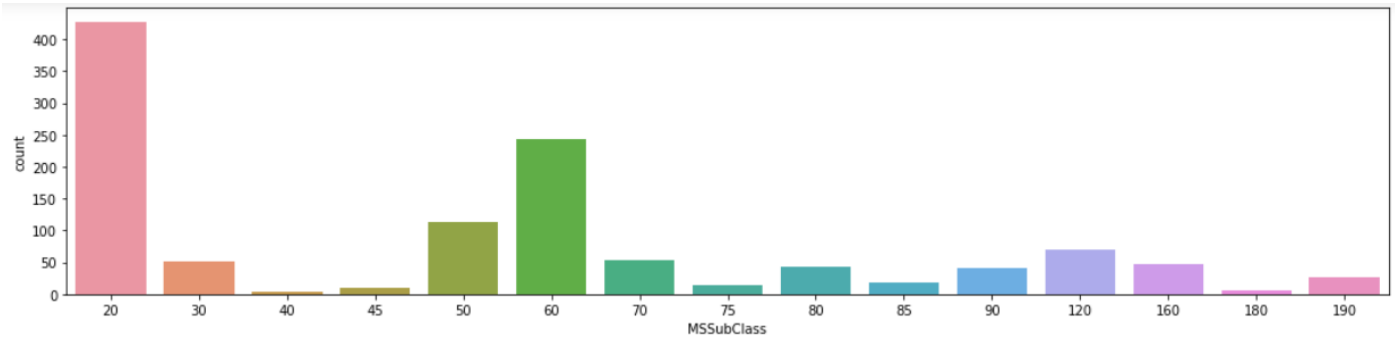
```
# Visualizing nulls
sns.heatmap(df.isnull())
```

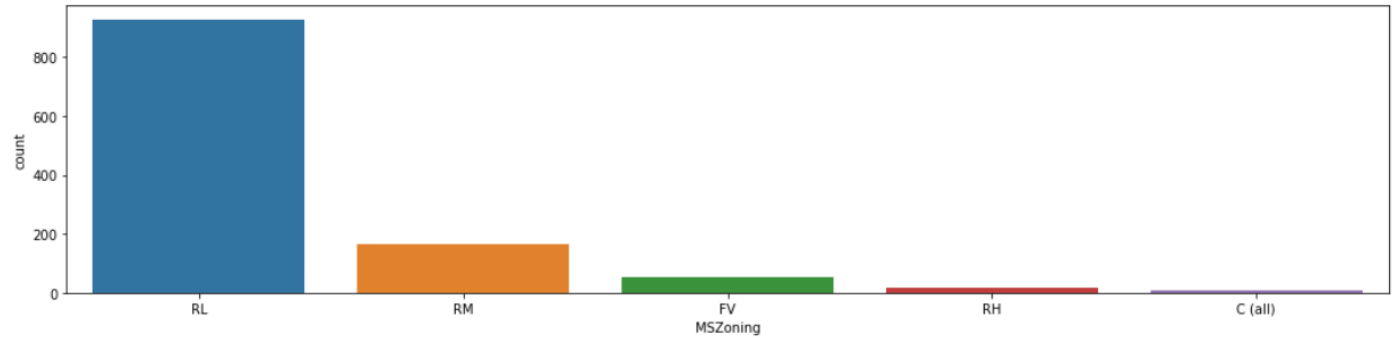
```
<AxesSubplot:>
```

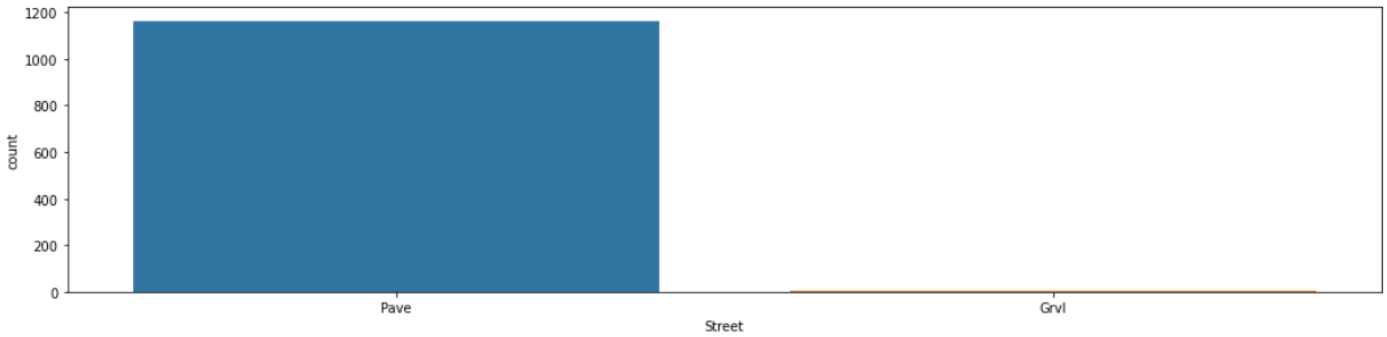


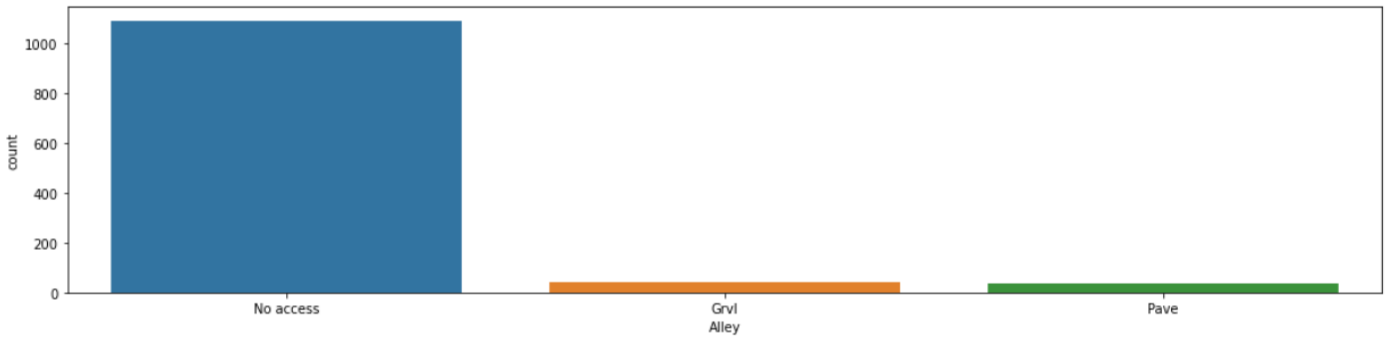
Heatmap showing no null values are present in the dataset.

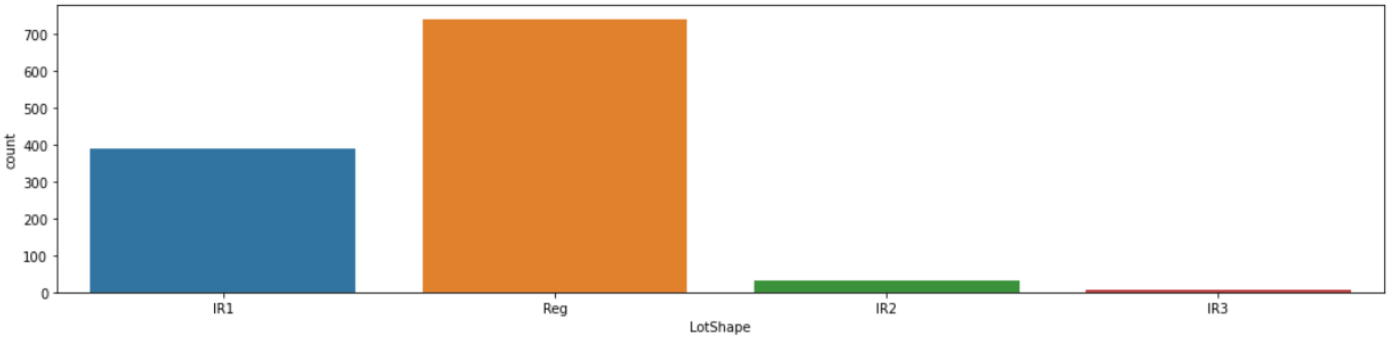
  - **Checking for the value counts using count plot**

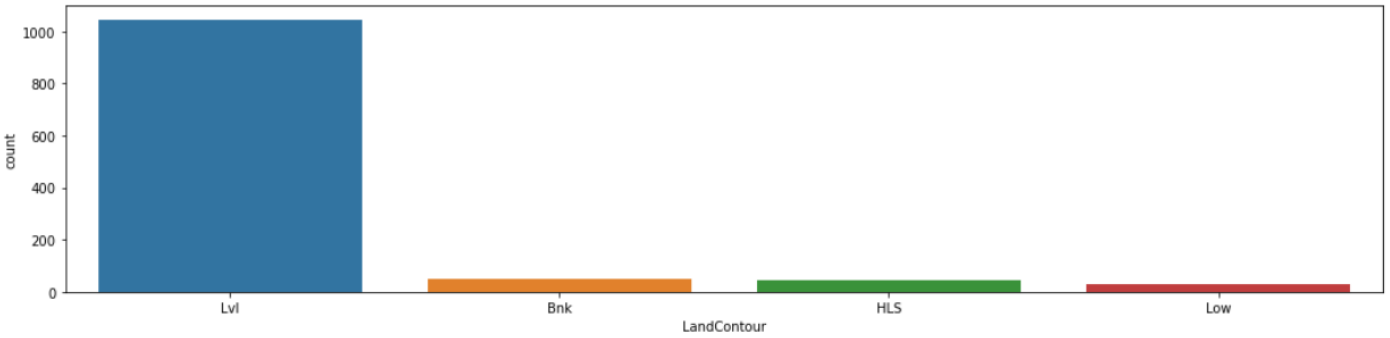


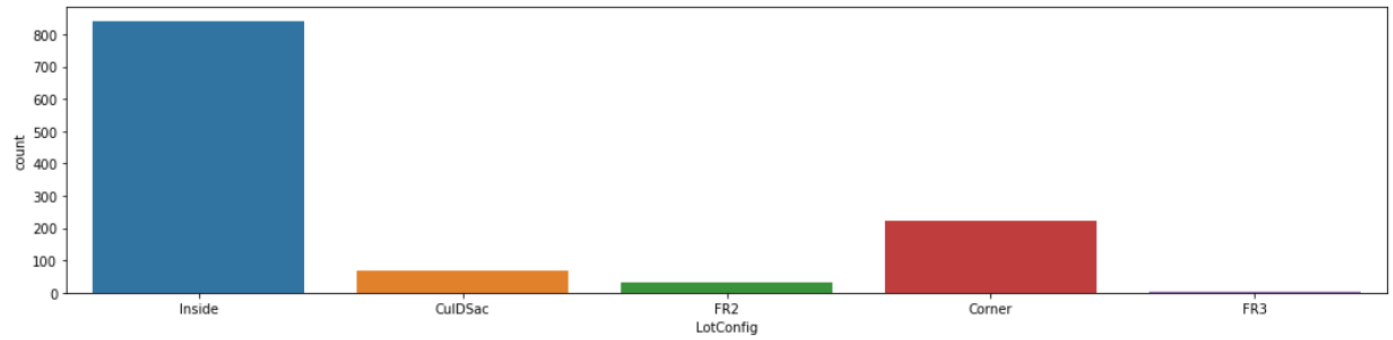


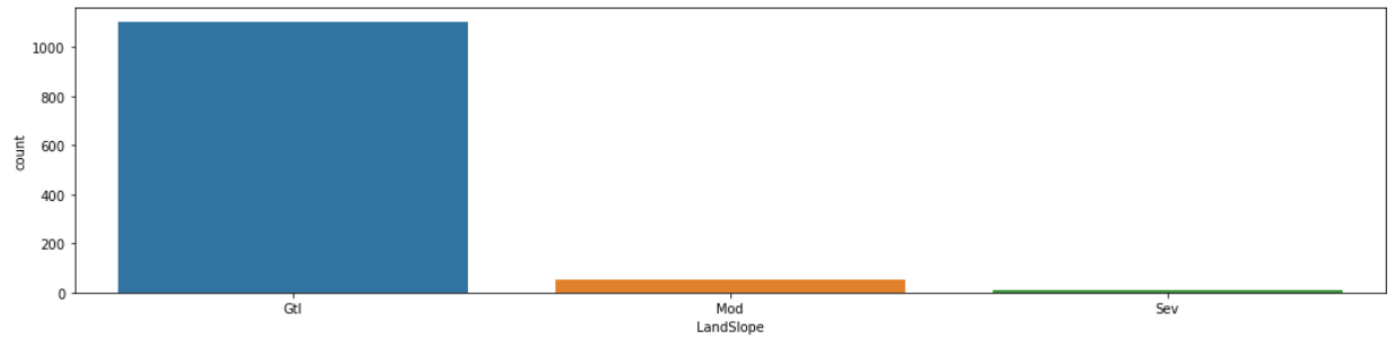

1200
1000
800
600
400
200
0
count
Pave
Grvl
Street


1000
800
600
400
200
0
count
No access
Grvl
Pave
Alley


700
600
500
400
300
200
100
0
count
IR1
Reg
IR2
IR3
LotShape


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

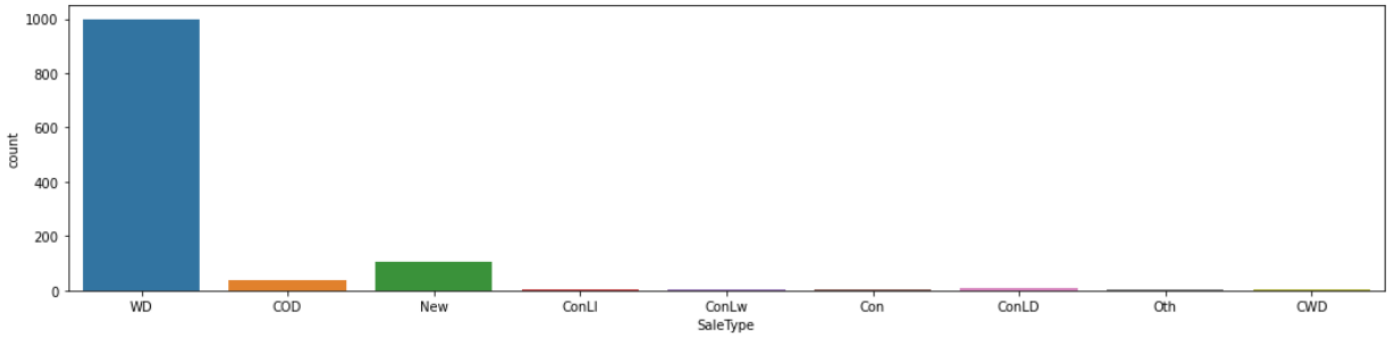


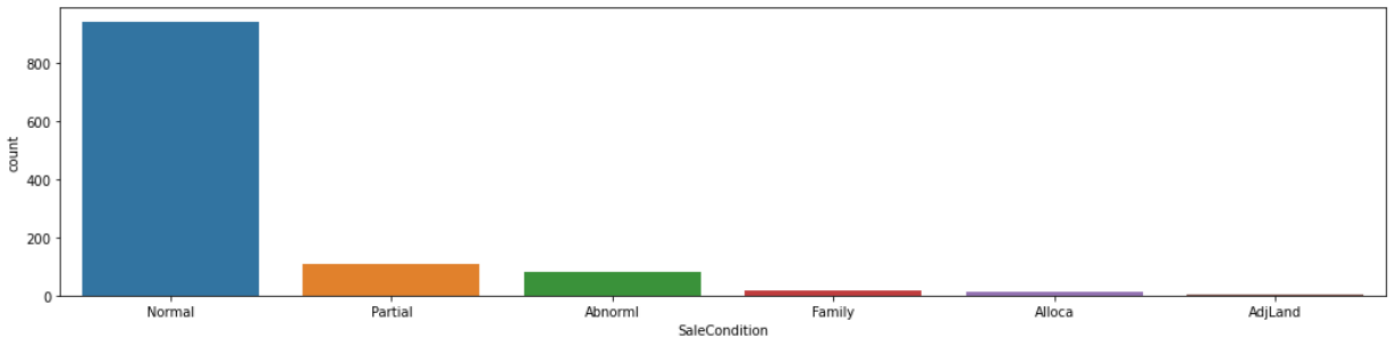


.

.

.





MSSubClass as 20 (1-STORY 1946 & NEWER ALL STYLES) is highly preferred followed by 60(2-STORY 1946 & NEWER). MSZoning is highly in RL (Residential Low Density) and least in C(all) Commercial. Street is majorly Pave. Houses with No alley access are in majority and other options are in similar counts. LotShape as Reg (Regular) is leading all others. LAndContour as Lvl (Near Flat/Level) is highly preferred. LotConfig as Inside is in majority and FR3 Frontage on 3 sides of property is least in count. Gtl-Gentle in LandSlope is mostly preferred. Condition 1 as Normal is in majority others are very less comparatively and same for Condition 2 as well. BldgType Type of dwelling as 1Fam Single-family Detached is leading all others. HouseStyle as 1Story One story is highly preferred followed by 2Story Two story

and 1.5Fin One and one-half story: 2nd level finished. OverallQual is mostly given Rates to overall material and finish of the house as Average followed by Above Average, Good, Very Good. OverallCond: Rates the overall condition of the house is Average for most of the houses. RoofStyle Gable is in majority among all other options. RoofMatl is highly chosen as CompShg Standard (Composite) Shingle. Exterior covering on house Exterior1st and Exterior2nd as VinylSd Vinyl Siding is highly preferred. ExterQual: Evaluates the quality of the material on the exterior, ExterCond: Evaluates the present condition of the material on the exterior and BsmtCond: Evaluates the general condition of the basement are rated as Average/Typical majorly. Type of foundation of houses are in CBlock Cinder Block and PConc Poured Contrete majorly. BsmtQual Evaluates the height of the basement is mostly Typical (80-89 inches) followed by Good (90-99 inches) and very few houses have no basement. BsmtExposure: Refers to walkout or garden level walls and majority of houses have No Exposure and least have No Basement. BsmtFinType1: Rating of basement finished area is Unfinshed for highest number of houses folloed by Good Living Quarters. BsmtFinType2: Rating of basement finished area (if multiple types) is highest for Unfinshed. Type of heating is highly by GasA (Gas forced warm air furnace). HeatingQC: Heating quality and condition is overall Excellent. CentralAir: Central air conditioning is there in majority of houses. Electrical: Electrical system SBrkr (Standard Circuit Breakers & Romex) is leading all others. LowQualFinSF: Low quality finished square feet (all floors) is 0 in high number. 3SsnPorch: Three season porch area in square feet and MiscVal: $Value of miscellaneous feature is 0 in majority.

BsmtFullBath: Basement full bathrooms are mostly 0 followed by 1. BsmtHalfBath: Basement half bathrooms are mostly 0 in count. FullBath: Full bathrooms above grade are 2 in most of the house data followed by 1. HalfBath: Half baths above grade are 0 in most of the houses, followed by 1 and least have 2. BedroomAbvGr are 3 in majority. KitchenAbvGr is 1 in majority. KitchenQual: Kitchen quality is highly Typical/Average followed by Good. TotRmsAbvGrd: Total rooms above grade (does not include bathrooms) are 6 followed by 7 then 5 and then 8. Functional: Home functionality (Assume typical unless deductions are warranted) is mostly Typical Functionality. Fireplaces: Number of fireplaces are 0 followed by 1 in majority of houses. FireplaceQu: Fireplace quality is majorly good in the houses where fireplaces are available but most houses have No Fireplace. GarageType: Garage location is mostly Attached to home followed by Detached from home. GarageFinish: Interior finish of the garage is majorly Unfinished followed by Rough Finished and then comes Finished least have No Garage. GarageCars: Size of garage in car capacity is mostly 2. GarageQual: Garage quality and GarageCond: Garage condition are mostly Typical/Average. PavedDrive: Paved driveway are mostly Paved. MoSold: Month Sold (MM) has highest count in 6 then 7 then 5. YrSold: Year Sold (YYYY) is highest in 2009.

- **Statistic Summary**

```
# Statistic Summary
df.describe()
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	...	GarageAre
count	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	...	1168.0000(
mean	56.767979	70.807363	10484.749144	6.104452	5.595890	1970.930651	1984.758562	444.726027	46.647260	569.721747	...	476.8604
std	41.940650	22.440317	8957.442311	1.390153	1.124343	30.145255	20.785185	462.664785	163.520016	449.375525	...	214.4667
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1875.000000	1950.000000	0.000000	0.000000	0.000000	...	0.0000(
25%	20.000000	60.000000	7621.500000	5.000000	5.000000	1954.000000	1966.000000	0.000000	0.000000	216.000000	...	338.0000(
50%	50.000000	70.000000	9522.500000	6.000000	5.000000	1972.000000	1993.000000	385.500000	0.000000	474.000000	...	480.0000(
75%	70.000000	79.250000	11515.500000	7.000000	6.000000	2000.000000	2004.000000	714.500000	0.000000	816.000000	...	576.0000(
max	190.000000	313.000000	164660.000000	10.000000	9.000000	2010.000000	2010.000000	5644.000000	1474.000000	2336.000000	...	1418.0000(

8 rows × 35 columns

Column MSSubClass has range from 20 to 190, LotFrontage ranges from 21 to 313, LotArea are from 1300 and 164660, OverallQual has values from 1-10. OverallCond has values from 1-9, YearBuilt has years from 1875 to 2010. YearRemodAdd has years from 1950-2010. BsmtFinSF1 ranges from 0 to 5644. BsmtUnfSF ranges from 0 to 1474. BsmtUnfSF ranges from 0 to 2336. GarageArea ranges from 0-1418, WoodDeckSF ranges from 0-857, OpenPorchSF has values from 0-547, EnclosedPorch has values from 0 to 552, 3SsnPorch contains values from 0-508, ScreenPorch has values from 0-480, YrSold has years from 2006 to 2010, SalePrice has values starting from 34900 to 755000 and so on..

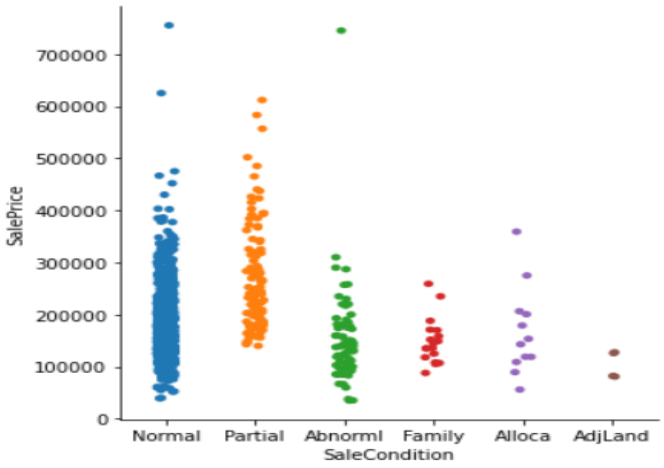
MSSubClass,LotArea,BsmtFinSF1,BsmtFinSF2,BsmtUnfSF,WoodDeckSF,OpenPorchSF,EnclosedPorch,SalePrice etc has mean>median i.e. right skewed. GarageArea has mean<median i.e. left skewed.

Standard deviation is high for many columns. Difference between 75 percentile and max value is high for many columns hence, outliers need to be treated.

- **Category Plots**

```
# Category plot for categorical data
sns.catplot(x='SaleCondition', y='SalePrice', data=df)
```

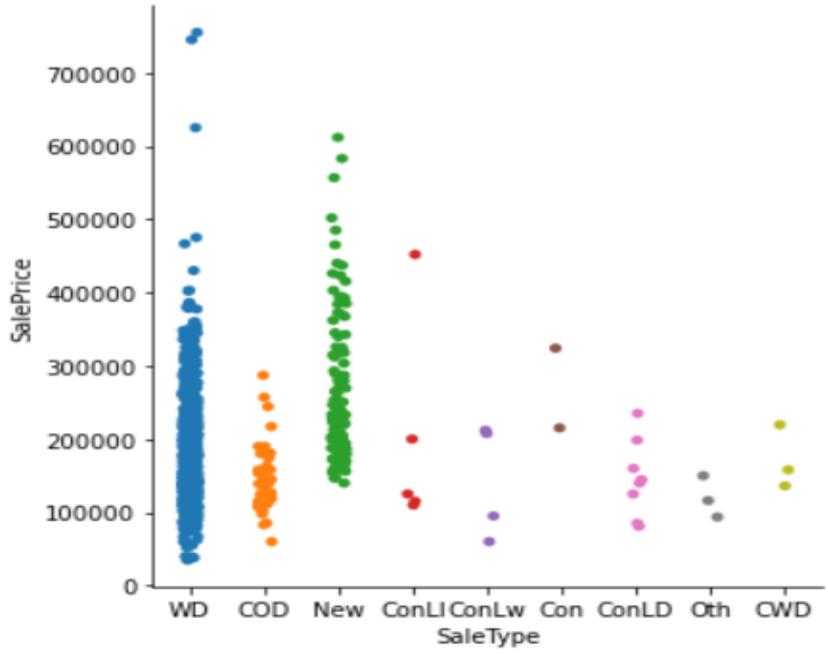
```
<seaborn.axisgrid.FacetGrid at 0x1f13fe390a0>
```



Houses are majorly of Normal Sale Condition and a few values are there for Adjoining Land Purchase. Starting and overall Sale Price is high for Partial Home not completed when last assessed (associated with New Homes) Sale Condition houses.

```
sns.catplot(x='SaleType',y='SalePrice',data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1f14169bd60>
```

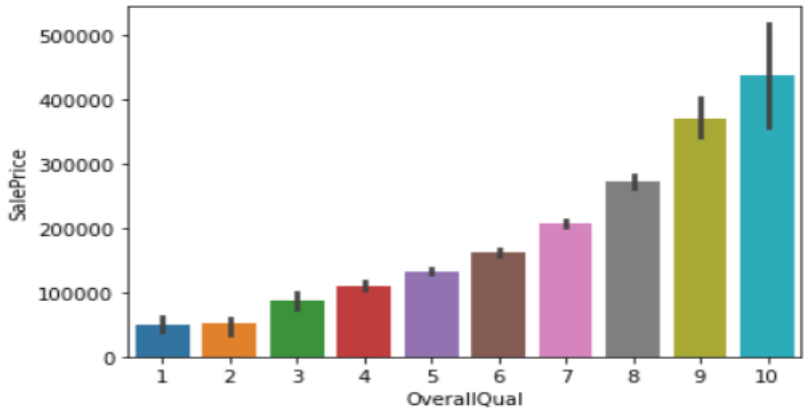


Sale Type WD: Warranty Deed-Conventional has majority of data and Sale Price in this is lowest and highest both comparatively. Next New Home just constructed and sold has goos amount of data with higher starting price w.r.t others.

- **Barplot showing relations between variables**

```
# Visualizing with barplot
sns.barplot(x='OverallQual',y='SalePrice',data=df)
```

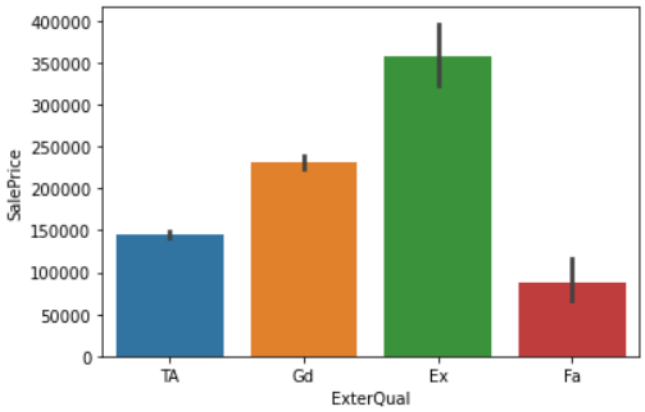
```
<AxesSubplot:xlabel='OverallQual', ylabel='SalePrice'>
```



Sale Price increases with increase in Rates the overall material and finish of the house.

```
#plt.figure(figsize=(20,6))
sns.barplot(x='ExterQual',y='SalePrice',data=df)
```

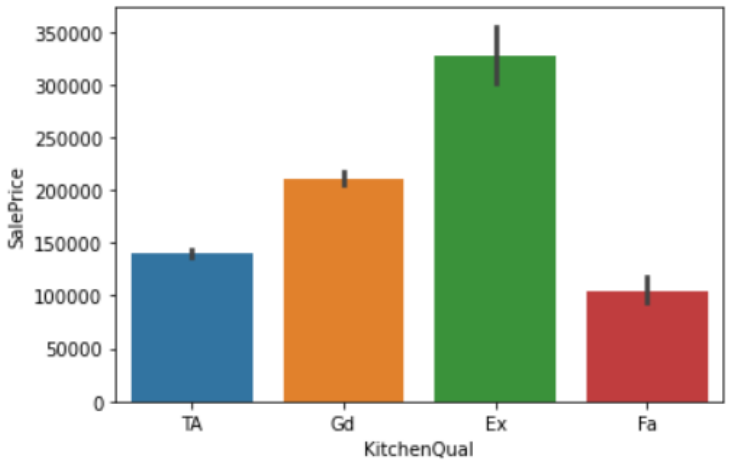
```
<AxesSubplot:xlabel='ExterQual', ylabel='SalePrice'>
```



Sale Price is highest for Excellent ExterQual: Evaluates the quality of the material on the exterior and lowest for Fair ExterQual.

```
sns.barplot(x='KitchenQual',y='SalePrice',data=df)
```

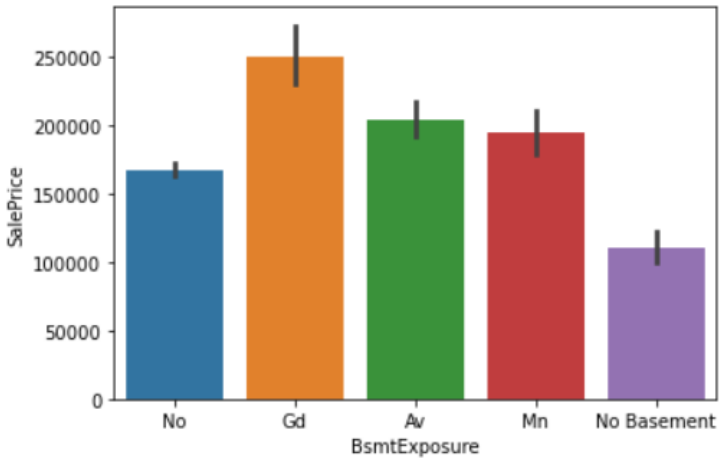
```
<AxesSubplot:xlabel='KitchenQual', ylabel='SalePrice'>
```



SalesPrice is highest for Excellent Kitchen quality and least for Fair Kitchen quality.

```
sns.barplot(x='BsmtExposure',y='SalePrice',data=df)
```

```
<AxesSubplot:xlabel='BsmtExposure', ylabel='SalePrice'>
```

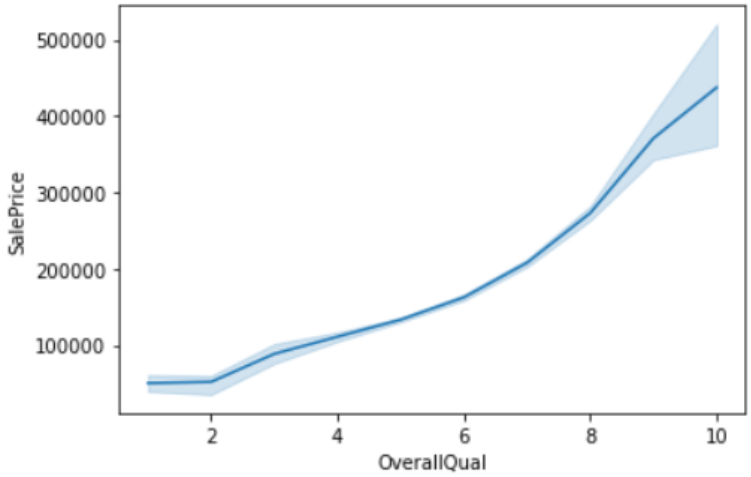


SalePrice is highest for Good Exposure in BsmtExposure: Refers to walkout or garden level walls and least for No Basement.

- **Lineplot showing relation between variables**

```
# Visualizing with lineplot
sns.lineplot(x='OverallQual',y='SalePrice',data=df)
```

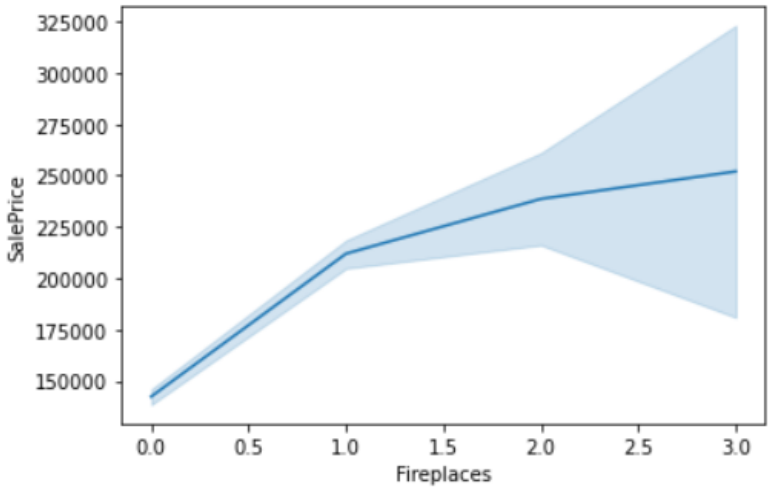
```
<AxesSubplot:xlabel='OverallQual', ylabel='SalePrice'>
```



SalePrice increases with increased rating of overall material and finish of the house.

```
sns.lineplot(x='Fireplaces',y='SalePrice',data=df)
```

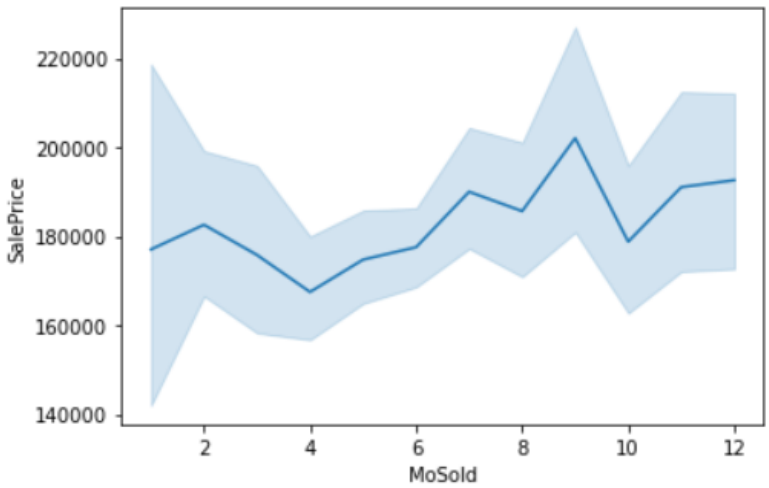
```
<AxesSubplot:xlabel='Fireplaces', ylabel='SalePrice'>
```



With increase in Fireplaces, SalePrice also increases.

```
sns.lineplot(x='MoSold',y='SalePrice',data=df)
```

```
<AxesSubplot:xlabel='MoSold', ylabel='SalePrice'>
```

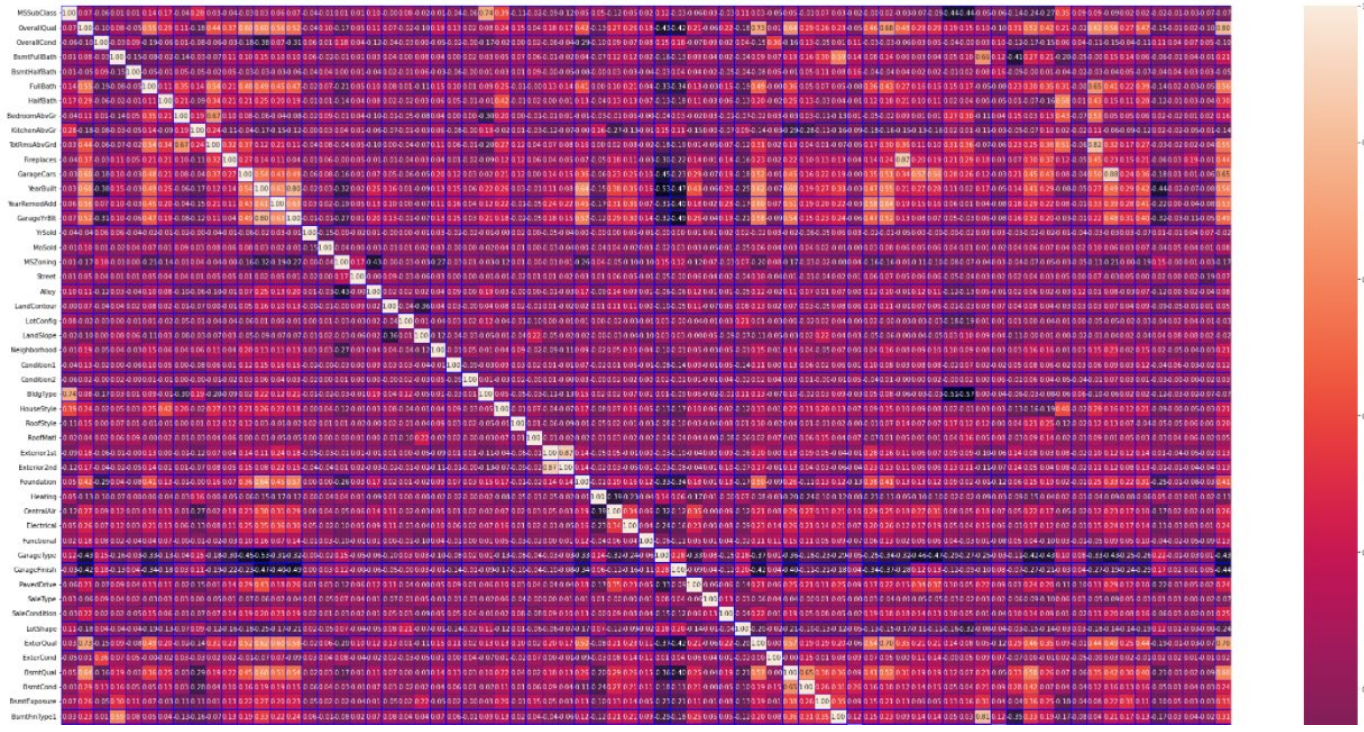


SalePrice is highest in 9th month and lowest in April.

- **Visualizing Correlation**

```
# Visualizing correlation
plt.figure(figsize=(40,30))
sns.heatmap(dff.corr(),annot=True,linewidths=0.5,linecolor='b',fmt='.2f')
```

```
<AxesSubplot:>
```



EnclosedPorch has good correlation with YearBuilt, GarageYrBlt. OpenPorchSF has good correlation with OverallQual, EnclosedPorch has good correlation with YearBuilt, GarageYrBlt. OpenPorchSF has good correlation with OverallQual, FullBath, HalfBath, GarageCars, YearBuilt, TotRmsAbvGrd, GarageYrBlt, ExterQual, BsmtQual,GarageArea. Target column 'SalePrice' has strong positive linear correlation with OverallQual, FullBath, TotRmsAbvGrd, Fireplaces, GarageCars, YearBuilt, YearRemodAdd, GarageYrBlt,

Foundation, BsmtQual, HeatingQC, KitchenQual, FireplaceQu, TotalBsmtSF, 1stFlrSF, GrLivArea, GarageArea, OpenPorchSF and has strong negative linear correlation with GarageType, GarageFinish.
SalePrice has good linear correlation with HalfBath, BsmtExposure, BsmtFinType1, WoodDeckSF.

- Interpretation of the Results

AUC-ROC Curve

AUC-ROC is not available for regression problems, because there is no cut-off value for this algorithm, and ROC AUC is only calculable in the case if the algorithm returns a continuous probability value (and only 1 value) for an unseen element.

Saving the model

```
# Saved the model in .pkl file
import pickle #import joblib
file='house_price_prediction.pkl'
pickle.dump(gb,open(file,'wb'))
```

```
# Loading the model for checking score on actual and predicted test sets
load_model=pickle.load(open(file,'rb'))
result=load_model.score(x_test,y_test)
result
```

```
0.9159722145447942
```

```
# Comparison dataframe having Original and Predicted values
actual=np.array(y_test)
predicted=np.array(gb.predict(x_test))
conclusion=pd.DataFrame({'Orignal':actual,'Predicted':predicted.astype(int)})
conclusion
```

	Orignal	Predicted
0	280000	273342
1	176500	180543
2	105000	129302
3	262280	246695
4	184100	215682
...	...	...
326	124500	131335
327	174000	173482
328	116050	114976
329	193500	210710
330	141000	116652

331 rows × 2 columns

CONCLUSION

- ## Key Findings and Conclusions of the Study

  We are getting good accuracy of training and testing the given dataset after cleaning the data using GradientBoostingRegressor for model development and evaluation.

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

  It helped in developing relevant programming abilities, demonstrated proficiency with statistical analysis of data, developed the ability to build and assess data-based models, executed statistical analyses and demonstrated skill in data management.

  Supervised learning algorithms are employed where the training data has output variables corresponding to the input variables. The algorithm analyses the input data and learns a function to map the relationship between the input and output variables. Supervised learning can further be classified into Regression, Classification, Forecasting, and Anomaly Detection.

  Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems. Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any Boolean function on discrete attributes using the decision tree.

  K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique used for both classification and regression problems. KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows.

  Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as **bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

- Limitations of this work and Scope for Future Work

  Data will change with time and hence relation between them will also change. For better performance, we can train data on clusters of data rather than the whole dataset.

  In future this machine learning model may collaborate with various other sources which can provide real time data for price prediction. Also we may add large historical data of house price which can help to improve accuracy of the machine learning model. We can have a website and/or application as user interface for interacting with user.