In [1]:
```
# Assignment: DSC680 - Project 2 – House Price Prediction
# Name: Bezawada, Sashidhar
# Date: 2024-04-27
# Milestone 2 : Draft of White Paper
```

In [2]:
```
# Import libraries
import pandas as pd
import numpy as np
import os
from __future__ import print_function, division
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import statistics
import plotly.express as px

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (accuracy_score, log_loss, classification_report)
```

# 1. Import the data frame and ensure that the data is loaded properly

In [3]:
```
#Load the dataset as a Pandas data frame
#Read train.csv
df_train = pd.read_csv("datasets/train.csv")
#Display the first ten rows of data
df_train.info()
df_train.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1460 non-null   int64  
 1   MSSubClass     1460 non-null   int64  
 2   MSZoning       1460 non-null   object 
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64  
 5   Street         1460 non-null   object 
 6   Alley          91 non-null     object 
 7   LotShape       1460 non-null   object 
 8   LandContour    1460 non-null   object 
 9   Utilities      1460 non-null   object 
 10  LotConfig      1460 non-null   object 
 11  LandSlope      1460 non-null   object 
 12  Neighborhood   1460 non-null   object 
 13  Condition1     1460 non-null   object 
 14  Condition2     1460 non-null   object 
 15  BldgType       1460 non-null   object 
 16  HouseStyle     1460 non-null   object 
 17  OverallQual    1460 non-null   int64  
 18  OverallCond    1460 non-null   int64  
 19  YearBuilt      1460 non-null   int64  
 20  YearRemodAdd   1460 non-null   int64  
 21  RoofStyle      1460 non-null   object 
 22  RoofMatl       1460 non-null   object 
 23  Exterior1st    1460 non-null   object 
 24  Exterior2nd    1460 non-null   object 
 25  MasVnrType     588 non-null    object 
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object 
 28  ExterCond      1460 non-null   object 
 29  Foundation     1460 non-null   object 
 30  BsmtQual       1423 non-null   object 
 31  BsmtCond       1423 non-null   object 
 32  BsmtExposure   1422 non-null   object 
 33  BsmtFinType1   1423 non-null   object 
 34  BsmtFinSF1     1460 non-null   int64  
 35  BsmtFinType2   1422 non-null   object 
 36  BsmtFinSF2     1460 non-null   int64  
 37  BsmtUnfSF      1460 non-null   int64  
 38  TotalBsmtSF    1460 non-null   int64  
 39  Heating        1460 non-null   object 
 40  HeatingQC      1460 non-null   object 
 41  CentralAir     1460 non-null   object 
 42  Electrical     1459 non-null   object 
 43  1stFlrSF       1460 non-null   int64  
 44  2ndFlrSF       1460 non-null   int64  
```

```
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

Out[3]:

| | Id | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley | LotShape | LandContour | Utilities | ... | PoolArea | PoolQC | Fence | MiscFeature | MiscVal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 60 | RL | 65.0 | 8450 | Pave | NaN | Reg | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **1** | 2 | 20 | RL | 80.0 | 9600 | Pave | NaN | Reg | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **2** | 3 | 60 | RL | 68.0 | 11250 | Pave | NaN | IR1 | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **3** | 4 | 70 | RL | 60.0 | 9550 | Pave | NaN | IR1 | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **4** | 5 | 60 | RL | 84.0 | 14260 | Pave | NaN | IR1 | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **5** | 6 | 50 | RL | 85.0 | 14115 | Pave | NaN | IR1 | Lvl | AllPub | ... | 0 | NaN | MnPrv | Shed | 700 |
| **6** | 7 | 20 | RL | 75.0 | 10084 | Pave | NaN | Reg | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **7** | 8 | 60 | RL | NaN | 10382 | Pave | NaN | IR1 | Lvl | AllPub | ... | 0 | NaN | NaN | Shed | 350 |
| **8** | 9 | 50 | RM | 51.0 | 6120 | Pave | NaN | Reg | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |
| **9** | 10 | 190 | RL | 50.0 | 7420 | Pave | NaN | Reg | Lvl | AllPub | ... | 0 | NaN | NaN | NaN | 0 |

10 rows × 81 columns

In [4]:

```
# Attributes and Descriptions

#SalePrice  :  the property's sale price in dollars. This is the target variable that you're trying to predict.
#MSSubClass :  The building class
#MSZoning :  The general zoning classification
#LotFrontage :  Linear feet of street connected to property
#LotArea :  Lot size in square feet
#Street  :  Type of road access
#Alley  :  Type of alley access
#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 main road or railroad
#Condition2 :  Proximity to main road or railroad (if a second is present)
#BldgType :  Type of dwelling
#HouseStyle :  Style of dwelling
#OverallQual :  Overall material and finish quality
#OverallCond :  Overall condition rating
#YearBuilt :  Original construction date
#YearRemodAdd :  Remodel date
#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 :  Exterior material quality
#ExterCond :  Present condition of the material on the exterior
#Foundation :  Type of foundation
#BsmtQual :  Height of the basement
#BsmtCond :  General condition of the basement
#BsmtExposure :  Walkout or garden level basement walls
#BsmtFinType1 :  Quality of basement finished area
#BsmtFinSF1 :  Type 1 finished square feet
#BsmtFinType2 :  Quality of second finished area (if present)
#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)
#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 :  Number of bedrooms above basement level
#Kitchen :  Number of kitchens
#KitchenQual :  Kitchen quality
#TotRmsAbvGrd :  Total rooms above grade (does not include bathrooms)
#Functional :  Home functionality rating
#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
#YrSold  :  Year Sold
#SaleType :  Type of sale
#SaleCondition :  Condition of sale
```

There are 80 feature columns. Using these features your model has to predict the house sale price indicated by the label column named SalePrice. We will drop the Id column as it is not necessary for model training.

In [5]:
```
for col in df_train.select_dtypes('O').columns:
    print('We have {} Unique values. Values in {} Column : {}'.format(len(df_train[col].unique()),col,df_train[col].unique()))
    print('__'*30)
```

```
We have 5 Unique values. Values in MSZoning Column : ['RL' 'RM' 'C (all)' 'FV' 'RH']
____________________________________________________________
We have 2 Unique values. Values in Street Column : ['Pave' 'Grvl']
____________________________________________________________
We have 3 Unique values. Values in Alley Column : [nan 'Grvl' 'Pave']
____________________________________________________________
We have 4 Unique values. Values in LotShape Column : ['Reg' 'IR1' 'IR2' 'IR3']
____________________________________________________________
We have 4 Unique values. Values in LandContour Column : ['Lvl' 'Bnk' 'Low' 'HLS']
____________________________________________________________
We have 2 Unique values. Values in Utilities Column : ['AllPub' 'NoSeWa']
____________________________________________________________
We have 5 Unique values. Values in LotConfig Column : ['Inside' 'FR2' 'Corner' 'CulDSac' 'FR3']
____________________________________________________________
We have 3 Unique values. Values in LandSlope Column : ['Gtl' 'Mod' 'Sev']
____________________________________________________________
We have 25 Unique values. Values in Neighborhood Column : ['CollgCr' 'Veenker' 'Crawfor' 'NoRidge' 'Mitchel' 'Somerst' 'NWAmes'
 'OldTown' 'BrkSide' 'Sawyer' 'NridgHt' 'NAmes' 'SawyerW' 'IDOTRR'
 'MeadowV' 'Edwards' 'Timber' 'Gilbert' 'StoneBr' 'ClearCr' 'NPkVill'
 'Blmngtn' 'BrDale' 'SWISU' 'Blueste']
____________________________________________________________
We have 9 Unique values. Values in Condition1 Column : ['Norm' 'Feedr' 'PosN' 'Artery' 'RRAe' 'RRNn' 'RRAn' 'PosA' 'RRNe']
____________________________________________________________
We have 8 Unique values. Values in Condition2 Column : ['Norm' 'Artery' 'RRNn' 'Feedr' 'PosN' 'PosA' 'RRAn' 'RRAe']
____________________________________________________________
We have 5 Unique values. Values in BldgType Column : ['1Fam' '2fmCon' 'Duplex' 'TwnhsE' 'Twnhs']
____________________________________________________________
We have 8 Unique values. Values in HouseStyle Column : ['2Story' '1Story' '1.5Fin' '1.5Unf' 'SFoyer' 'SLvl' '2.5Unf' '2.5Fin']
____________________________________________________________
We have 6 Unique values. Values in RoofStyle Column : ['Gable' 'Hip' 'Gambrel' 'Mansard' 'Flat' 'Shed']
____________________________________________________________
We have 8 Unique values. Values in RoofMatl Column : ['CompShg' 'WdShngl' 'Metal' 'WdShake' 'Membran' 'Tar&Grv' 'Roll'
 'ClyTile']
____________________________________________________________
We have 15 Unique values. Values in Exterior1st Column : ['VinylSd' 'MetalSd' 'Wd Sdng' 'HdBoard' 'BrkFace' 'WdShing' 'CemntBd'
 'Plywood' 'AsbShng' 'Stucco' 'BrkComm' 'AsphShn' 'Stone' 'ImStucc'
 'CBlock']
____________________________________________________________
We have 16 Unique values. Values in Exterior2nd Column : ['VinylSd' 'MetalSd' 'Wd Shng' 'HdBoard' 'Plywood' 'Wd Sdng' 'CmentBd'
 'BrkFace' 'Stucco' 'AsbShng' 'Brk Cmn' 'ImStucc' 'AsphShn' 'Stone'
 'Other' 'CBlock']
____________________________________________________________
We have 4 Unique values. Values in MasVnrType Column : ['BrkFace' nan 'Stone' 'BrkCmn']
____________________________________________________________
We have 4 Unique values. Values in ExterQual Column : ['Gd' 'TA' 'Ex' 'Fa']
____________________________________________________________
We have 5 Unique values. Values in ExterCond Column : ['TA' 'Gd' 'Fa' 'Po' 'Ex']
____________________________________________________________
We have 6 Unique values. Values in Foundation Column : ['PConc' 'CBlock' 'BrkTil' 'Wood' 'Slab' 'Stone']
____________________________________________________________
```

```
We have 5 Unique values. Values in BsmtQual Column : ['Gd' 'TA' 'Ex' nan 'Fa']
____________________________________________________________
We have 5 Unique values. Values in BsmtCond Column : ['TA' 'Gd' nan 'Fa' 'Po']
____________________________________________________________
We have 5 Unique values. Values in BsmtExposure Column : ['No' 'Gd' 'Mn' 'Av' nan]
____________________________________________________________
We have 7 Unique values. Values in BsmtFinType1 Column : ['GLQ' 'ALQ' 'Unf' 'Rec' 'BLQ' nan 'LwQ']
____________________________________________________________
We have 7 Unique values. Values in BsmtFinType2 Column : ['Unf' 'BLQ' nan 'ALQ' 'Rec' 'LwQ' 'GLQ']
____________________________________________________________
We have 6 Unique values. Values in Heating Column : ['GasA' 'GasW' 'Grav' 'Wall' 'OthW' 'Floor']
____________________________________________________________
We have 5 Unique values. Values in HeatingQC Column : ['Ex' 'Gd' 'TA' 'Fa' 'Po']
____________________________________________________________
We have 2 Unique values. Values in CentralAir Column : ['Y' 'N']
____________________________________________________________
We have 6 Unique values. Values in Electrical Column : ['SBrkr' 'FuseF' 'FuseA' 'FuseP' 'Mix' nan]
____________________________________________________________
We have 4 Unique values. Values in KitchenQual Column : ['Gd' 'TA' 'Ex' 'Fa']
____________________________________________________________
We have 7 Unique values. Values in Functional Column : ['Typ' 'Min1' 'Maj1' 'Min2' 'Mod' 'Maj2' 'Sev']
____________________________________________________________
We have 6 Unique values. Values in FireplaceQu Column : [nan 'TA' 'Gd' 'Fa' 'Ex' 'Po']
____________________________________________________________
We have 7 Unique values. Values in GarageType Column : ['Attchd' 'Detchd' 'BuiltIn' 'CarPort' nan 'Basment' '2Types']
____________________________________________________________
We have 4 Unique values. Values in GarageFinish Column : ['RFn' 'Unf' 'Fin' nan]
____________________________________________________________
We have 6 Unique values. Values in GarageQual Column : ['TA' 'Fa' 'Gd' nan 'Ex' 'Po']
____________________________________________________________
We have 6 Unique values. Values in GarageCond Column : ['TA' 'Fa' nan 'Gd' 'Po' 'Ex']
____________________________________________________________
We have 3 Unique values. Values in PavedDrive Column : ['Y' 'N' 'P']
____________________________________________________________
We have 4 Unique values. Values in PoolQC Column : [nan 'Ex' 'Fa' 'Gd']
____________________________________________________________
We have 5 Unique values. Values in Fence Column : [nan 'MnPrv' 'GdWo' 'GdPrv' 'MnWw']
____________________________________________________________
We have 5 Unique values. Values in MiscFeature Column : [nan 'Shed' 'Gar2' 'Othr' 'TenC']
____________________________________________________________
We have 9 Unique values. Values in SaleType Column : ['WD' 'New' 'COD' 'ConLD' 'ConLI' 'CWD' 'ConLw' 'Con' 'Oth']
____________________________________________________________
We have 6 Unique values. Values in SaleCondition Column : ['Normal' 'Abnorml' 'Partial' 'AdjLand' 'Alloca' 'Family']
____________________________________________________________
```

## Exploratory Data Analysis

In [6]:
```
#Check for Duplicates
have_duplicate_rows = df_train.duplicated().any()
have_duplicate_rows
```

Out[6]:
```
False
```

In [7]:
```
#Check for Missing Values
missing_df = df_train.isnull().sum().to_frame().rename(columns={0:"Total No. of Missing Values"})
missing_df["% of Missing Values"] = round((missing_df["Total No. of Missing Values"]/len(df_train))*100,2)
missing_df
```

Out[7]:

| | Total No. of Missing Values | % of Missing Values |
|---|---|---|
| **Id** | 0 | 0.00 |
| **MSSubClass** | 0 | 0.00 |
| **MSZoning** | 0 | 0.00 |
| **LotFrontage** | 259 | 17.74 |
| **LotArea** | 0 | 0.00 |
| **...** | ... | ... |
| **MoSold** | 0 | 0.00 |
| **YrSold** | 0 | 0.00 |
| **SaleType** | 0 | 0.00 |
| **SaleCondition** | 0 | 0.00 |
| **SalePrice** | 0 | 0.00 |

81 rows × 2 columns

In [8]:
```
#Check for Unique Values
null_train_data = round(100*(df_train.isnull().sum().sort_values(ascending=False)/len(df_train.index)),2)\
                    .to_frame().rename(columns={0:'Train Null values percentage'})[:20]
null_train_data
```

Out[8]:

| | Train Null values percentage |
|---|---|
| **PoolQC** | 99.52 |
| **MiscFeature** | 96.30 |
| **Alley** | 93.77 |
| **Fence** | 80.75 |
| **MasVnrType** | 59.73 |
| **FireplaceQu** | 47.26 |
| **LotFrontage** | 17.74 |
| **GarageYrBlt** | 5.55 |
| **GarageCond** | 5.55 |
| **GarageType** | 5.55 |
| **GarageFinish** | 5.55 |
| **GarageQual** | 5.55 |
| **BsmtFinType2** | 2.60 |
| **BsmtExposure** | 2.60 |
| **BsmtQual** | 2.53 |
| **BsmtCond** | 2.53 |
| **BsmtFinType1** | 2.53 |
| **MasVnrArea** | 0.55 |
| **Electrical** | 0.07 |
| **Id** | 0.00 |

In [9]:

```
df_catcols = df_train.select_dtypes(include=['object', 'O']).columns

df_numcols = df_train.select_dtypes(include=['int64', 'float64'])

print(f"Feature | # 0 Values | # Null Values | # Unique Values ")
print("="*60)
for feature in df_numcols:
    zero_values = (df_train[feature] == 0).sum()
    null_values = df_train[feature].isnull().sum()
    unique_values = len(df_train[feature].unique())
```

```
    print(f"{feature} | {zero_values} | {zero_values} | {unique_values} ")
```

```
Feature | # 0 Values | # Null Values | # Unique Values
============================================================
Id | 0 | 0 | 1460
MSSubClass | 0 | 0 | 15
LotFrontage | 0 | 0 | 111
LotArea | 0 | 0 | 1073
OverallQual | 0 | 0 | 10
OverallCond | 0 | 0 | 9
YearBuilt | 0 | 0 | 112
YearRemodAdd | 0 | 0 | 61
MasVnrArea | 861 | 861 | 328
BsmtFinSF1 | 467 | 467 | 637
BsmtFinSF2 | 1293 | 1293 | 144
BsmtUnfSF | 118 | 118 | 780
TotalBsmtSF | 37 | 37 | 721
1stFlrSF | 0 | 0 | 753
2ndFlrSF | 829 | 829 | 417
LowQualFinSF | 1434 | 1434 | 24
GrLivArea | 0 | 0 | 861
BsmtFullBath | 856 | 856 | 4
BsmtHalfBath | 1378 | 1378 | 3
FullBath | 9 | 9 | 4
HalfBath | 913 | 913 | 3
BedroomAbvGr | 6 | 6 | 8
KitchenAbvGr | 1 | 1 | 4
TotRmsAbvGrd | 0 | 0 | 12
Fireplaces | 690 | 690 | 4
GarageYrBlt | 0 | 0 | 98
GarageCars | 81 | 81 | 5
GarageArea | 81 | 81 | 441
WoodDeckSF | 761 | 761 | 274
OpenPorchSF | 656 | 656 | 202
EnclosedPorch | 1252 | 1252 | 120
3SsnPorch | 1436 | 1436 | 20
ScreenPorch | 1344 | 1344 | 76
PoolArea | 1453 | 1453 | 8
MiscVal | 1408 | 1408 | 21
MoSold | 0 | 0 | 12
YrSold | 0 | 0 | 5
SalePrice | 0 | 0 | 663
```

In [10]:

```
category_cols = df_train.select_dtypes(include="O").columns

for column in category_cols:
    print('Unique values of ', column, set(df_train[column]))
    print("-"*127)
```

```
Unique values of  MSZoning {'RL', 'RM', 'C (all)', 'FV', 'RH'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Street {'Grvl', 'Pave'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Alley {'Pave', nan, 'Grvl'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  LotShape {'IR3', 'Reg', 'IR2', 'IR1'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  LandContour {'Lvl', 'Bnk', 'HLS', 'Low'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Utilities {'AllPub', 'NoSeWa'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  LotConfig {'FR3', 'Corner', 'Inside', 'FR2', 'CulDSac'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  LandSlope {'Gtl', 'Mod', 'Sev'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Neighborhood {'NoRidge', 'Blueste', 'NWAmes', 'NridgHt', 'SWISU', 'Gilbert', 'CollgCr', 'Sawyer', 'SawyerW', 'E
dwards', 'BrkSide', 'Blmngtn', 'Veenker', 'NAmes', 'MeadowV', 'NPkVill', 'Crawfor', 'OldTown', 'Mitchel', 'StoneBr', 'Somerst',
'IDOTRR', 'Timber', 'ClearCr', 'BrDale'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Condition1 {'PosN', 'RRNe', 'Norm', 'PosA', 'RRAe', 'RRAn', 'RRNn', 'Feedr', 'Artery'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Condition2 {'PosN', 'Norm', 'PosA', 'RRAn', 'RRAe', 'RRNn', 'Feedr', 'Artery'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  BldgType {'TwnhsE', 'Twnhs', '1Fam', '2fmCon', 'Duplex'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  HouseStyle {'SLvl', '2.5Fin', '1Story', '2.5Unf', '1.5Fin', '2Story', 'SFoyer', '1.5Unf'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  RoofStyle {'Flat', 'Shed', 'Gable', 'Mansard', 'Gambrel', 'Hip'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  RoofMatl {'Membran', 'CompShg', 'WdShngl', 'Roll', 'Tar&Grv', 'ClyTile', 'Metal', 'WdShake'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Exterior1st {'AsphShn', 'BrkFace', 'Stucco', 'Stone', 'VinylSd', 'BrkComm', 'MetalSd', 'CBlock', 'Plywood', 'Wd
Sdng', 'AsbShng', 'WdShing', 'ImStucc', 'CemntBd', 'HdBoard'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Exterior2nd {'AsphShn', 'Other', 'Brk Cmn', 'BrkFace', 'Stucco', 'Stone', 'VinylSd', 'MetalSd', 'CBlock', 'Plyw
ood', 'Wd Sdng', 'AsbShng', 'Wd Shng', 'ImStucc', 'CmentBd', 'HdBoard'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  MasVnrType {'Stone', 'BrkCmn', nan, 'BrkFace'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  ExterQual {'Gd', 'Ex', 'Fa', 'TA'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  ExterCond {'Gd', 'Po', 'TA', 'Ex', 'Fa'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  Foundation {'Stone', 'Wood', 'Slab', 'BrkTil', 'PConc', 'CBlock'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  BsmtQual {'Gd', nan, 'TA', 'Ex', 'Fa'}
------------------------------------------------------------------------------------------------------------------------
Unique values of  BsmtCond {'Gd', nan, 'Po', 'TA', 'Fa'}
------------------------------------------------------------------------------------------------------------------------
```

```
Unique values of  BsmtExposure {'Gd', nan, 'Mn', 'No', 'Av'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  BsmtFinType1 {nan, 'BLQ', 'LwQ', 'Unf', 'GLQ', 'ALQ', 'Rec'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  BsmtFinType2 {nan, 'BLQ', 'LwQ', 'Unf', 'GLQ', 'ALQ', 'Rec'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  Heating {'Floor', 'GasA', 'Wall', 'GasW', 'Grav', 'OthW'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  HeatingQC {'Gd', 'Po', 'TA', 'Ex', 'Fa'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  CentralAir {'N', 'Y'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  Electrical {nan, 'SBrkr', 'FuseP', 'FuseF', 'Mix', 'FuseA'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  KitchenQual {'Gd', 'Ex', 'Fa', 'TA'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  Functional {'Mod', 'Sev', 'Min2', 'Maj2', 'Min1', 'Maj1', 'Typ'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  FireplaceQu {'Gd', nan, 'Po', 'TA', 'Ex', 'Fa'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  GarageType {'Attchd', nan, 'CarPort', 'Basment', 'BuiltIn', 'Detchd', '2Types'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  GarageFinish {'Fin', 'RFn', nan, 'Unf'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  GarageQual {'Gd', nan, 'Po', 'TA', 'Ex', 'Fa'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  GarageCond {'Gd', 'Po', nan, 'TA', 'Ex', 'Fa'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  PavedDrive {'N', 'P', 'Y'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  PoolQC {'Gd', 'Ex', nan, 'Fa'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  Fence {'GdPrv', nan, 'GdWo', 'MnPrv', 'MnWw'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  MiscFeature {nan, 'Shed', 'TenC', 'Othr', 'Gar2'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  SaleType {'ConLD', 'WD', 'COD', 'ConLw', 'CWD', 'New', 'Oth', 'Con', 'ConLI'}
----------------------------------------------------------------------------------------------------------------------------
Unique values of  SaleCondition {'AdjLand', 'Normal', 'Alloca', 'Partial', 'Abnorml', 'Family'}
----------------------------------------------------------------------------------------------------------------------------
```

In [11]:
```
#Describing each field in the dataset *( Transpose the table)
df_train.describe(include="O").T
```

Out[11]:

| | count | unique | top | freq |
|---|---|---|---|---|
| **MSZoning** | 1460 | 5 | RL | 1151 |
| **Street** | 1460 | 2 | Pave | 1454 |
| **Alley** | 91 | 2 | Grvl | 50 |
| **LotShape** | 1460 | 4 | Reg | 925 |
| **LandContour** | 1460 | 4 | Lvl | 1311 |
| **Utilities** | 1460 | 2 | AllPub | 1459 |
| **LotConfig** | 1460 | 5 | Inside | 1052 |
| **LandSlope** | 1460 | 3 | Gtl | 1382 |
| **Neighborhood** | 1460 | 25 | NAmes | 225 |
| **Condition1** | 1460 | 9 | Norm | 1260 |
| **Condition2** | 1460 | 8 | Norm | 1445 |
| **BldgType** | 1460 | 5 | 1Fam | 1220 |
| **HouseStyle** | 1460 | 8 | 1Story | 726 |
| **RoofStyle** | 1460 | 6 | Gable | 1141 |
| **RoofMatl** | 1460 | 8 | CompShg | 1434 |
| **Exterior1st** | 1460 | 15 | VinylSd | 515 |
| **Exterior2nd** | 1460 | 16 | VinylSd | 504 |
| **MasVnrType** | 588 | 3 | BrkFace | 445 |
| **ExterQual** | 1460 | 4 | TA | 906 |
| **ExterCond** | 1460 | 5 | TA | 1282 |
| **Foundation** | 1460 | 6 | PConc | 647 |
| **BsmtQual** | 1423 | 4 | TA | 649 |
| **BsmtCond** | 1423 | 4 | TA | 1311 |
| **BsmtExposure** | 1422 | 4 | No | 953 |
| **BsmtFinType1** | 1423 | 6 | Unf | 430 |
| **BsmtFinType2** | 1422 | 6 | Unf | 1256 |
| **Heating** | 1460 | 6 | GasA | 1428 |
| **HeatingQC** | 1460 | 5 | Ex | 741 |

| | count | unique | top | freq |
|---|---|---|---|---|
| **CentralAir** | 1460 | 2 | Y | 1365 |
| **Electrical** | 1459 | 5 | SBrkr | 1334 |
| **KitchenQual** | 1460 | 4 | TA | 735 |
| **Functional** | 1460 | 7 | Typ | 1360 |
| **FireplaceQu** | 770 | 5 | Gd | 380 |
| **GarageType** | 1379 | 6 | Attchd | 870 |
| **GarageFinish** | 1379 | 3 | Unf | 605 |
| **GarageQual** | 1379 | 5 | TA | 1311 |
| **GarageCond** | 1379 | 5 | TA | 1326 |
| **PavedDrive** | 1460 | 3 | Y | 1340 |
| **PoolQC** | 7 | 3 | Gd | 3 |
| **Fence** | 281 | 4 | MnPrv | 157 |
| **MiscFeature** | 54 | 4 | Shed | 49 |
| **SaleType** | 1460 | 9 | WD | 1267 |
| **SaleCondition** | 1460 | 6 | Normal | 1198 |

In [12]:

```
#Describing each field in the dataset *( Transpose the table)
df_train.describe(exclude="O").T
```

Out[12]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Id** | 1460.0 | 730.500000 | 421.610009 | 1.0 | 365.75 | 730.5 | 1095.25 | 1460.0 |
| **MSSubClass** | 1460.0 | 56.897260 | 42.300571 | 20.0 | 20.00 | 50.0 | 70.00 | 190.0 |
| **LotFrontage** | 1201.0 | 70.049958 | 24.284752 | 21.0 | 59.00 | 69.0 | 80.00 | 313.0 |
| **LotArea** | 1460.0 | 10516.828082 | 9981.264932 | 1300.0 | 7553.50 | 9478.5 | 11601.50 | 215245.0 |
| **OverallQual** | 1460.0 | 6.099315 | 1.382997 | 1.0 | 5.00 | 6.0 | 7.00 | 10.0 |
| **OverallCond** | 1460.0 | 5.575342 | 1.112799 | 1.0 | 5.00 | 5.0 | 6.00 | 9.0 |
| **YearBuilt** | 1460.0 | 1971.267808 | 30.202904 | 1872.0 | 1954.00 | 1973.0 | 2000.00 | 2010.0 |
| **YearRemodAdd** | 1460.0 | 1984.865753 | 20.645407 | 1950.0 | 1967.00 | 1994.0 | 2004.00 | 2010.0 |
| **MasVnrArea** | 1452.0 | 103.685262 | 181.066207 | 0.0 | 0.00 | 0.0 | 166.00 | 1600.0 |
| **BsmtFinSF1** | 1460.0 | 443.639726 | 456.098091 | 0.0 | 0.00 | 383.5 | 712.25 | 5644.0 |
| **BsmtFinSF2** | 1460.0 | 46.549315 | 161.319273 | 0.0 | 0.00 | 0.0 | 0.00 | 1474.0 |
| **BsmtUnfSF** | 1460.0 | 567.240411 | 441.866955 | 0.0 | 223.00 | 477.5 | 808.00 | 2336.0 |
| **TotalBsmtSF** | 1460.0 | 1057.429452 | 438.705324 | 0.0 | 795.75 | 991.5 | 1298.25 | 6110.0 |
| **1stFlrSF** | 1460.0 | 1162.626712 | 386.587738 | 334.0 | 882.00 | 1087.0 | 1391.25 | 4692.0 |
| **2ndFlrSF** | 1460.0 | 346.992466 | 436.528436 | 0.0 | 0.00 | 0.0 | 728.00 | 2065.0 |
| **LowQualFinSF** | 1460.0 | 5.844521 | 48.623081 | 0.0 | 0.00 | 0.0 | 0.00 | 572.0 |
| **GrLivArea** | 1460.0 | 1515.463699 | 525.480383 | 334.0 | 1129.50 | 1464.0 | 1776.75 | 5642.0 |
| **BsmtFullBath** | 1460.0 | 0.425342 | 0.518911 | 0.0 | 0.00 | 0.0 | 1.00 | 3.0 |
| **BsmtHalfBath** | 1460.0 | 0.057534 | 0.238753 | 0.0 | 0.00 | 0.0 | 0.00 | 2.0 |
| **FullBath** | 1460.0 | 1.565068 | 0.550916 | 0.0 | 1.00 | 2.0 | 2.00 | 3.0 |
| **HalfBath** | 1460.0 | 0.382877 | 0.502885 | 0.0 | 0.00 | 0.0 | 1.00 | 2.0 |
| **BedroomAbvGr** | 1460.0 | 2.866438 | 0.815778 | 0.0 | 2.00 | 3.0 | 3.00 | 8.0 |
| **KitchenAbvGr** | 1460.0 | 1.046575 | 0.220338 | 0.0 | 1.00 | 1.0 | 1.00 | 3.0 |
| **TotRmsAbvGrd** | 1460.0 | 6.517808 | 1.625393 | 2.0 | 5.00 | 6.0 | 7.00 | 14.0 |
| **Fireplaces** | 1460.0 | 0.613014 | 0.644666 | 0.0 | 0.00 | 1.0 | 1.00 | 3.0 |
| **GarageYrBlt** | 1379.0 | 1978.506164 | 24.689725 | 1900.0 | 1961.00 | 1980.0 | 2002.00 | 2010.0 |
| **GarageCars** | 1460.0 | 1.767123 | 0.747315 | 0.0 | 1.00 | 2.0 | 2.00 | 4.0 |
| **GarageArea** | 1460.0 | 472.980137 | 213.804841 | 0.0 | 334.50 | 480.0 | 576.00 | 1418.0 |

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **WoodDeckSF** | 1460.0 | 94.244521 | 125.338794 | 0.0 | 0.00 | 0.0 | 168.00 | 857.0 |
| **OpenPorchSF** | 1460.0 | 46.660274 | 66.256028 | 0.0 | 0.00 | 25.0 | 68.00 | 547.0 |
| **EnclosedPorch** | 1460.0 | 21.954110 | 61.119149 | 0.0 | 0.00 | 0.0 | 0.00 | 552.0 |
| **3SsnPorch** | 1460.0 | 3.409589 | 29.317331 | 0.0 | 0.00 | 0.0 | 0.00 | 508.0 |
| **ScreenPorch** | 1460.0 | 15.060959 | 55.757415 | 0.0 | 0.00 | 0.0 | 0.00 | 480.0 |
| **PoolArea** | 1460.0 | 2.758904 | 40.177307 | 0.0 | 0.00 | 0.0 | 0.00 | 738.0 |
| **MiscVal** | 1460.0 | 43.489041 | 496.123024 | 0.0 | 0.00 | 0.0 | 0.00 | 15500.0 |
| **MoSold** | 1460.0 | 6.321918 | 2.703626 | 1.0 | 5.00 | 6.0 | 8.00 | 12.0 |
| **YrSold** | 1460.0 | 2007.815753 | 1.328095 | 2006.0 | 2007.00 | 2008.0 | 2009.00 | 2010.0 |
| **SalePrice** | 1460.0 | 180921.195890 | 79442.502883 | 34900.0 | 129975.00 | 163000.0 | 214000.00 | 755000.0 |

In [13]:
```
for col in df_train.select_dtypes('O').columns:
    print('We have {} Unique values. Values in {} Column : {}'.format(len(df_train[col].unique()),col,df_train[col].unique()))
    print('__'*30)
```

```
We have 5 Unique values. Values in MSZoning Column : ['RL' 'RM' 'C (all)' 'FV' 'RH']
____________________________________________________________
We have 2 Unique values. Values in Street Column : ['Pave' 'Grvl']
____________________________________________________________
We have 3 Unique values. Values in Alley Column : [nan 'Grvl' 'Pave']
____________________________________________________________
We have 4 Unique values. Values in LotShape Column : ['Reg' 'IR1' 'IR2' 'IR3']
____________________________________________________________
We have 4 Unique values. Values in LandContour Column : ['Lvl' 'Bnk' 'Low' 'HLS']
____________________________________________________________
We have 2 Unique values. Values in Utilities Column : ['AllPub' 'NoSeWa']
____________________________________________________________
We have 5 Unique values. Values in LotConfig Column : ['Inside' 'FR2' 'Corner' 'CulDSac' 'FR3']
____________________________________________________________
We have 3 Unique values. Values in LandSlope Column : ['Gtl' 'Mod' 'Sev']
____________________________________________________________
We have 25 Unique values. Values in Neighborhood Column : ['CollgCr' 'Veenker' 'Crawfor' 'NoRidge' 'Mitchel' 'Somerst' 'NWAmes'
 'OldTown' 'BrkSide' 'Sawyer' 'NridgHt' 'NAmes' 'SawyerW' 'IDOTRR'
 'MeadowV' 'Edwards' 'Timber' 'Gilbert' 'StoneBr' 'ClearCr' 'NPkVill'
 'Blmngtn' 'BrDale' 'SWISU' 'Blueste']
____________________________________________________________
We have 9 Unique values. Values in Condition1 Column : ['Norm' 'Feedr' 'PosN' 'Artery' 'RRAe' 'RRNn' 'RRAn' 'PosA' 'RRNe']
____________________________________________________________
We have 8 Unique values. Values in Condition2 Column : ['Norm' 'Artery' 'RRNn' 'Feedr' 'PosN' 'PosA' 'RRAn' 'RRAe']
____________________________________________________________
We have 5 Unique values. Values in BldgType Column : ['1Fam' '2fmCon' 'Duplex' 'TwnhsE' 'Twnhs']
____________________________________________________________
We have 8 Unique values. Values in HouseStyle Column : ['2Story' '1Story' '1.5Fin' '1.5Unf' 'SFoyer' 'SLvl' '2.5Unf' '2.5Fin']
____________________________________________________________
We have 6 Unique values. Values in RoofStyle Column : ['Gable' 'Hip' 'Gambrel' 'Mansard' 'Flat' 'Shed']
____________________________________________________________
We have 8 Unique values. Values in RoofMatl Column : ['CompShg' 'WdShngl' 'Metal' 'WdShake' 'Membran' 'Tar&Grv' 'Roll'
 'ClyTile']
____________________________________________________________
We have 15 Unique values. Values in Exterior1st Column : ['VinylSd' 'MetalSd' 'Wd Sdng' 'HdBoard' 'BrkFace' 'WdShing' 'CemntBd'
 'Plywood' 'AsbShng' 'Stucco' 'BrkComm' 'AsphShn' 'Stone' 'ImStucc'
 'CBlock']
____________________________________________________________
We have 16 Unique values. Values in Exterior2nd Column : ['VinylSd' 'MetalSd' 'Wd Shng' 'HdBoard' 'Plywood' 'Wd Sdng' 'CmentBd'
 'BrkFace' 'Stucco' 'AsbShng' 'Brk Cmn' 'ImStucc' 'AsphShn' 'Stone'
 'Other' 'CBlock']
____________________________________________________________
We have 4 Unique values. Values in MasVnrType Column : ['BrkFace' nan 'Stone' 'BrkCmn']
____________________________________________________________
We have 4 Unique values. Values in ExterQual Column : ['Gd' 'TA' 'Ex' 'Fa']
____________________________________________________________
We have 5 Unique values. Values in ExterCond Column : ['TA' 'Gd' 'Fa' 'Po' 'Ex']
____________________________________________________________
We have 6 Unique values. Values in Foundation Column : ['PConc' 'CBlock' 'BrkTil' 'Wood' 'Slab' 'Stone']
____________________________________________________________
```

```
We have 5 Unique values. Values in BsmtQual Column : ['Gd' 'TA' 'Ex' nan 'Fa']
____________________________________________________________
We have 5 Unique values. Values in BsmtCond Column : ['TA' 'Gd' nan 'Fa' 'Po']
____________________________________________________________
We have 5 Unique values. Values in BsmtExposure Column : ['No' 'Gd' 'Mn' 'Av' nan]
____________________________________________________________
We have 7 Unique values. Values in BsmtFinType1 Column : ['GLQ' 'ALQ' 'Unf' 'Rec' 'BLQ' nan 'LwQ']
____________________________________________________________
We have 7 Unique values. Values in BsmtFinType2 Column : ['Unf' 'BLQ' nan 'ALQ' 'Rec' 'LwQ' 'GLQ']
____________________________________________________________
We have 6 Unique values. Values in Heating Column : ['GasA' 'GasW' 'Grav' 'Wall' 'OthW' 'Floor']
____________________________________________________________
We have 5 Unique values. Values in HeatingQC Column : ['Ex' 'Gd' 'TA' 'Fa' 'Po']
____________________________________________________________
We have 2 Unique values. Values in CentralAir Column : ['Y' 'N']
____________________________________________________________
We have 6 Unique values. Values in Electrical Column : ['SBrkr' 'FuseF' 'FuseA' 'FuseP' 'Mix' nan]
____________________________________________________________
We have 4 Unique values. Values in KitchenQual Column : ['Gd' 'TA' 'Ex' 'Fa']
____________________________________________________________
We have 7 Unique values. Values in Functional Column : ['Typ' 'Min1' 'Maj1' 'Min2' 'Mod' 'Maj2' 'Sev']
____________________________________________________________
We have 6 Unique values. Values in FireplaceQu Column : [nan 'TA' 'Gd' 'Fa' 'Ex' 'Po']
____________________________________________________________
We have 7 Unique values. Values in GarageType Column : ['Attchd' 'Detchd' 'BuiltIn' 'CarPort' nan 'Basment' '2Types']
____________________________________________________________
We have 4 Unique values. Values in GarageFinish Column : ['RFn' 'Unf' 'Fin' nan]
____________________________________________________________
We have 6 Unique values. Values in GarageQual Column : ['TA' 'Fa' 'Gd' nan 'Ex' 'Po']
____________________________________________________________
We have 6 Unique values. Values in GarageCond Column : ['TA' 'Fa' nan 'Gd' 'Po' 'Ex']
____________________________________________________________
We have 3 Unique values. Values in PavedDrive Column : ['Y' 'N' 'P']
____________________________________________________________
We have 4 Unique values. Values in PoolQC Column : [nan 'Ex' 'Fa' 'Gd']
____________________________________________________________
We have 5 Unique values. Values in Fence Column : [nan 'MnPrv' 'GdWo' 'GdPrv' 'MnWw']
____________________________________________________________
We have 5 Unique values. Values in MiscFeature Column : [nan 'Shed' 'Gar2' 'Othr' 'TenC']
____________________________________________________________
We have 9 Unique values. Values in SaleType Column : ['WD' 'New' 'COD' 'ConLD' 'ConLI' 'CWD' 'ConLw' 'Con' 'Oth']
____________________________________________________________
We have 6 Unique values. Values in SaleCondition Column : ['Normal' 'Abnorml' 'Partial' 'AdjLand' 'Alloca' 'Family']
____________________________________________________________
```

## Data Transformations

In [14]:
```
#Deleting the following Columns as their values is same across all observations
#columns=['Alley','MasVnrType','FireplaceQu','PoolQC','Fence','MiscFeature']
columns=["Id"]
df_train=df_train.drop(columns,axis=1)
```

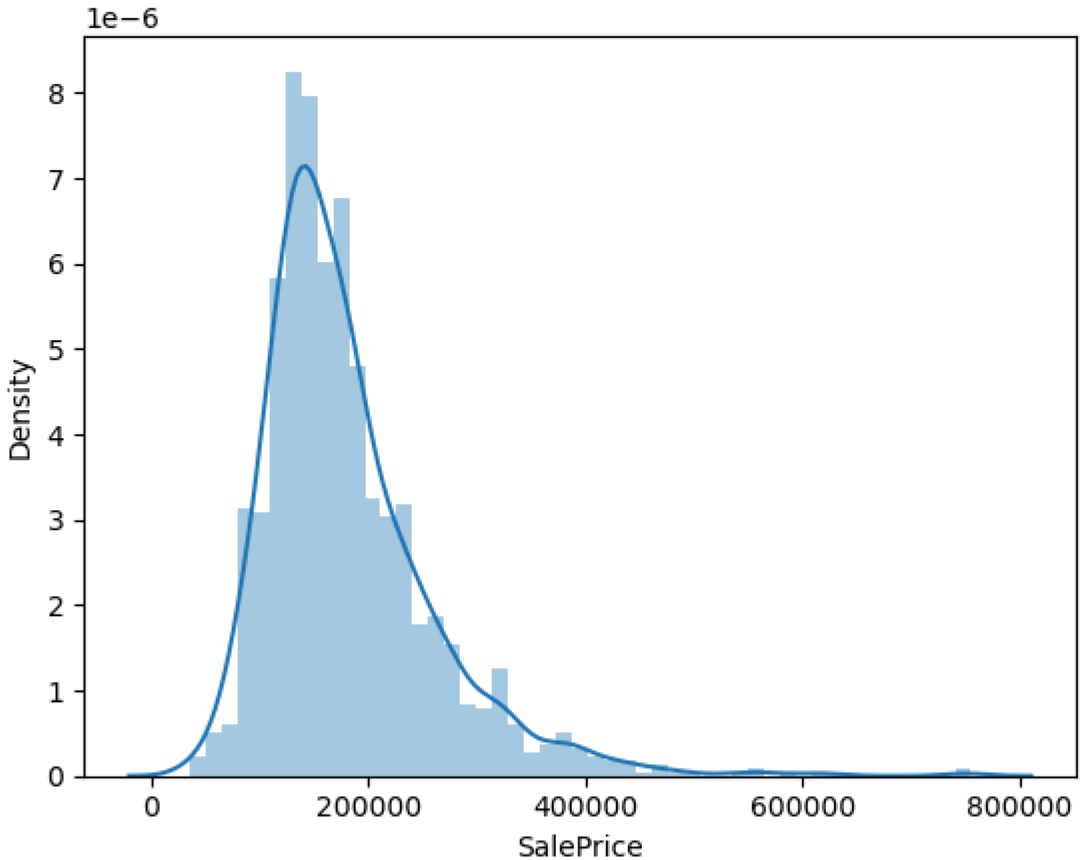
In [15]:
```
#Replacing "Y" and "N" in traget variable with 1 and 0

df_train.CentralAir.replace({"Y":1,"N":0}, inplace=True)
```

# Charts

In [16]:
```
sns.distplot(df_train['SalePrice']);
df_train['SalePrice'].describe()
```

```
C:\Users\bsash\AppData\Local\Temp\ipykernel_36268\1588422418.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_train['SalePrice']);
```

Out[16]:
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```




Since the Saleprice figures are skewed towards left, we will apply the log transformation to obtain a centralized data

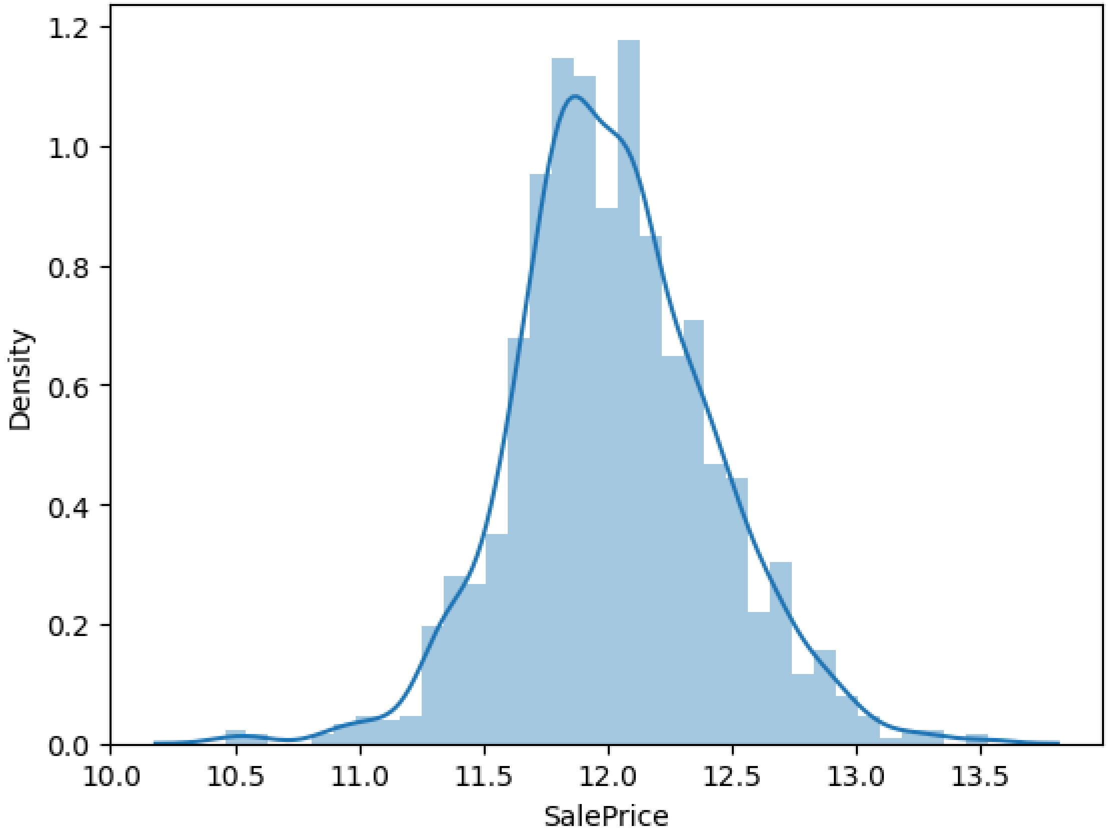
In [17]:

```
#Log Transformation
sns.distplot(np.log1p(df_train['SalePrice']))
```

```
C:\Users\bsash\AppData\Local\Temp\ipykernel_36268\2195387701.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(np.log1p(df_train['SalePrice']))
```

Out[17]:

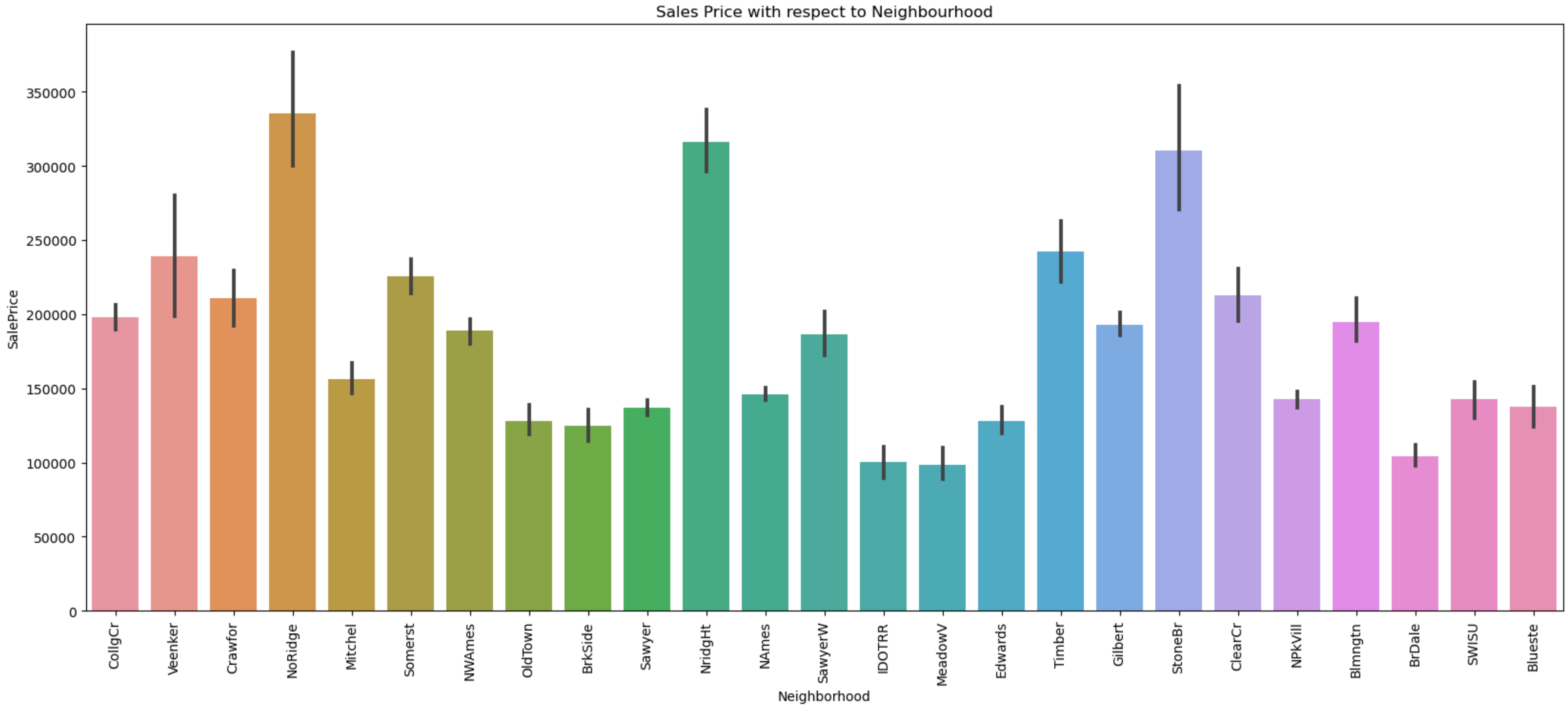
```
<Axes: xlabel='SalePrice', ylabel='Density'>
```




In [18]:

```
# Visualization to show  'Sale Price' with respect to 'Neighborhood'

plt.figure(figsize=(20, 8))
sns.barplot(x="Neighborhood", y="SalePrice", data= df_train)
plt.title("Sales Price with respect to Neighbourhood")
plt.xticks(rotation=90)
```

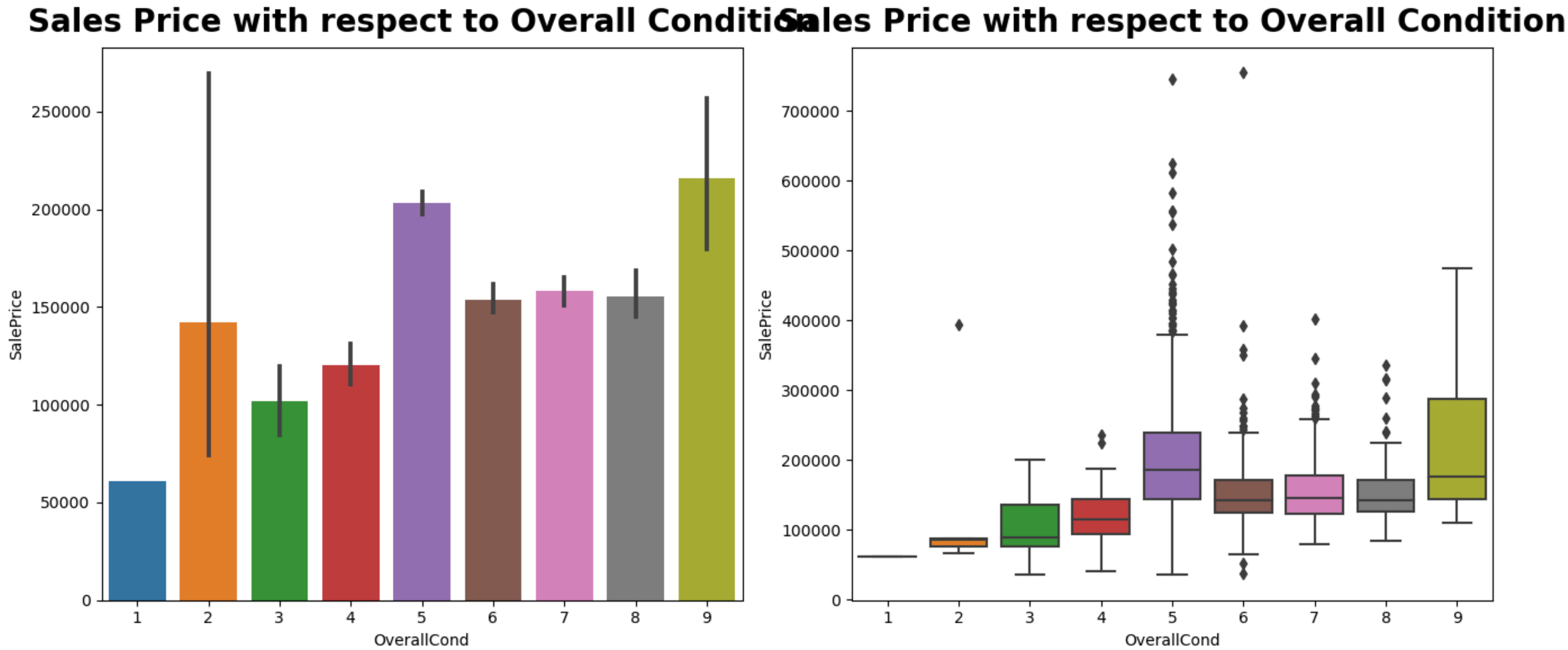
Out[18]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24]),
 [Text(0, 0, 'CollgCr'),
  Text(1, 0, 'Veenker'),
  Text(2, 0, 'Crawfor'),
  Text(3, 0, 'NoRidge'),
  Text(4, 0, 'Mitchel'),
  Text(5, 0, 'Somerst'),
  Text(6, 0, 'NWAmes'),
  Text(7, 0, 'OldTown'),
  Text(8, 0, 'BrkSide'),
  Text(9, 0, 'Sawyer'),
  Text(10, 0, 'NridgHt'),
  Text(11, 0, 'NAmes'),
  Text(12, 0, 'SawyerW'),
  Text(13, 0, 'IDOTRR'),
  Text(14, 0, 'MeadowV'),
  Text(15, 0, 'Edwards'),
  Text(16, 0, 'Timber'),
  Text(17, 0, 'Gilbert'),
  Text(18, 0, 'StoneBr'),
  Text(19, 0, 'ClearCr'),
  Text(20, 0, 'NPkVill'),
  Text(21, 0, 'Blmngtn'),
  Text(22, 0, 'BrDale'),
  Text(23, 0, 'SWISU'),
  Text(24, 0, 'Blueste')])
```




Properties in some of the Neighborhoods are high priced.

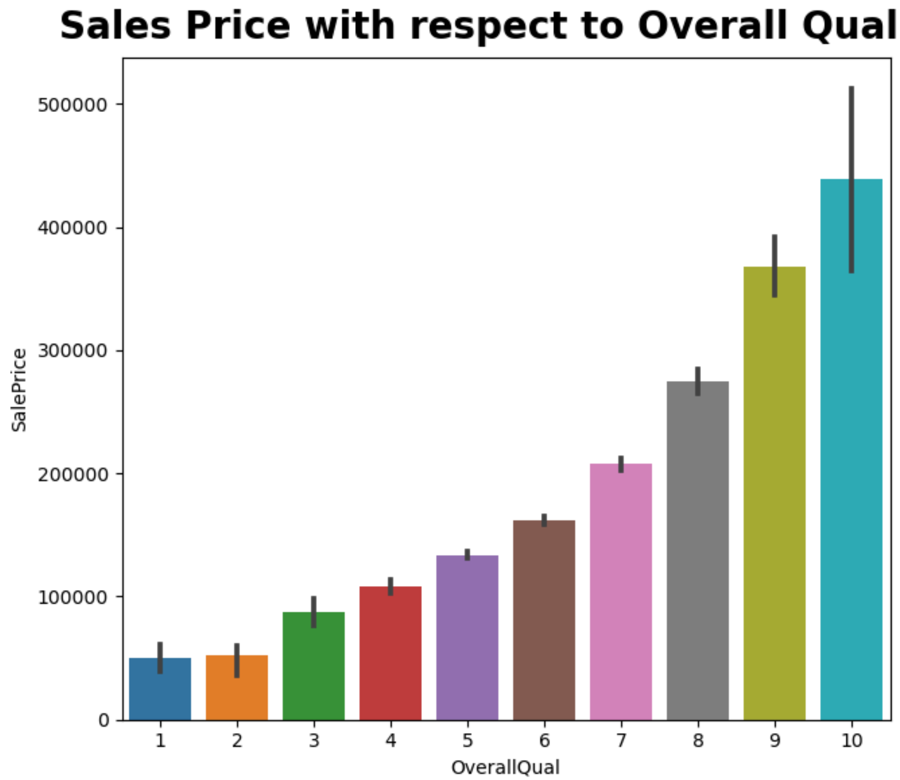
In [19]:

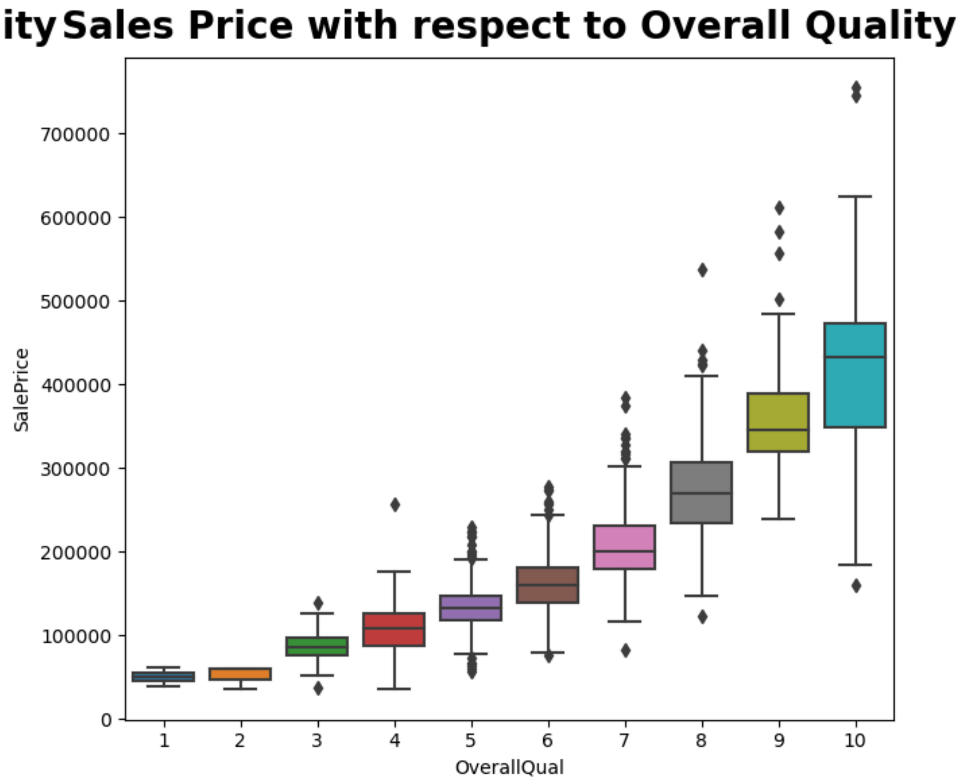
```
# Visualization to show  'overall condition' with respect to 'Saleprice'
plt.figure(figsize=(13.5,6))
plt.subplot(1,2,1)
sns.barplot(x="OverallCond",y="SalePrice",data=df_train)
plt.title("Sales Price with respect to Overall Condition",fontweight="black",size=20,pad=10)


#Visualization to show Employee Distribution by Age & Attrition.
plt.subplot(1,2,2)
sns.boxplot(x="OverallCond",y="SalePrice",data=df_train)
plt.title("Sales Price with respect to Overall Condition",fontweight="black",size=20,pad=10)
plt.tight_layout()
plt.show()
```



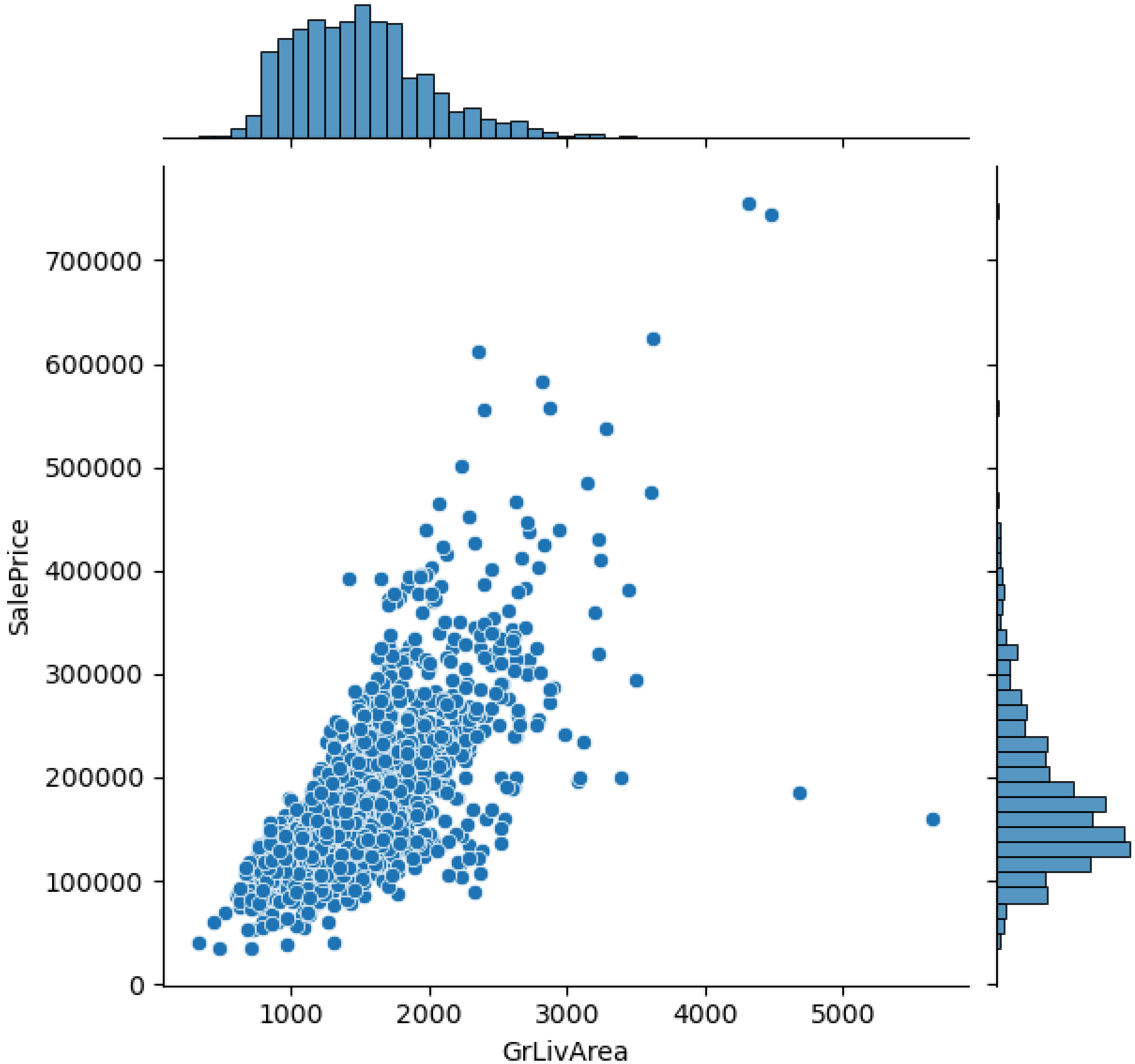

In [20]:

```python
# Visualization to show  'overall quality' with respect to 'Saleprice'
plt.figure(figsize=(13.5,6))
plt.subplot(1,2,1)
sns.barplot(x="OverallQual",y="SalePrice",data=df_train)
plt.title("Sales Price with respect to Overall Quality",fontweight="black",size=20,pad=10)


#Visualization to show Employee Distribution by Age & Attrition.
plt.subplot(1,2,2)
sns.boxplot(x="OverallQual",y="SalePrice",data=df_train)
plt.title("Sales Price with respect to Overall Quality",fontweight="black",size=20,pad=10)
plt.tight_layout()
plt.show()
```




In [21]:
```
# Analyse some important numeric columns
sns.jointplot(x='GrLivArea', y='SalePrice', data=df_train)
plt.show()
```




In [22]:

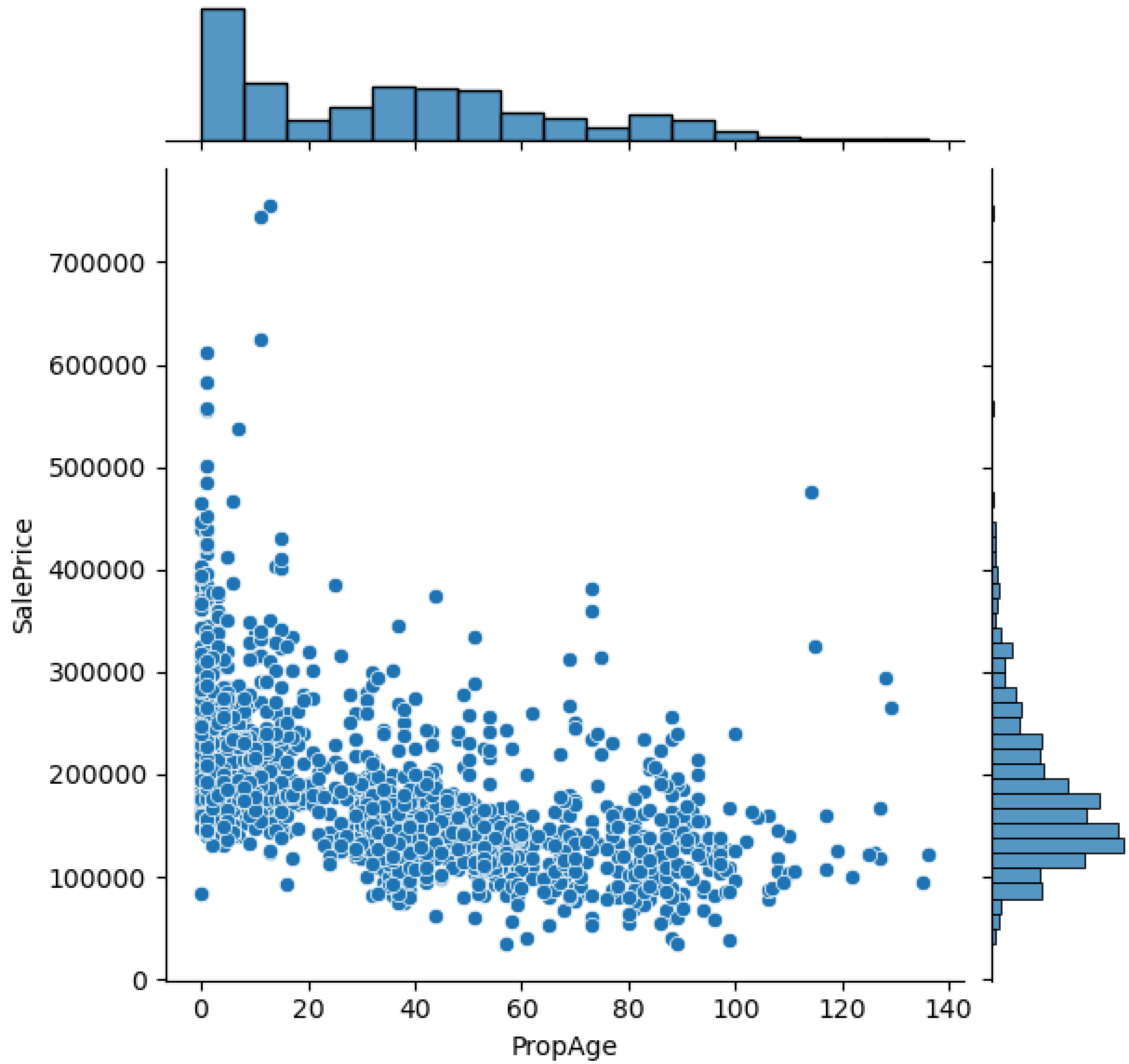
```
#Derive a column for 'Age of the property' when it was sold: Name it as 'PropAge'
df_train['PropAge'] = (df_train['YrSold'] - df_train['YearBuilt'])
cols1=['PropAge','YrSold','YearBuilt']
df_train[cols1].head()
```

Out[22]:

| | PropAge | YrSold | YearBuilt |
|---|---|---|---|
| **0** | 5 | 2008 | 2003 |
| **1** | 31 | 2007 | 1976 |
| **2** | 7 | 2008 | 2001 |
| **3** | 91 | 2006 | 1915 |
| **4** | 8 | 2008 | 2000 |

In [23]:

```
# PropAge vs SalePrice
sns.jointplot(x = df_train['PropAge'], y = df_train['SalePrice'])
plt.show()
```




Increase in Property Age shows a decreasing saleprice trend i.e newer the property, high is the value. Now we can drop the column Month sold and Year Sold, Year built and Year remodelled since it will not be required further

In [24]:
```
df_train = df_train.drop(['MoSold','YrSold','YearBuilt','YearRemodAdd'], axis = 1)
df_train.head()
```

Out[24]:

| | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley | LotShape | LandContour | Utilities | LotConfig | ... | ScreenPorch | PoolArea | PoolQC | Fence |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 60 | RL | 65.0 | 8450 | Pave | NaN | Reg | Lvl | AllPub | Inside | ... | 0 | 0 | NaN | NaN |
| **1** | 20 | RL | 80.0 | 9600 | Pave | NaN | Reg | Lvl | AllPub | FR2 | ... | 0 | 0 | NaN | NaN |
| **2** | 60 | RL | 68.0 | 11250 | Pave | NaN | IR1 | Lvl | AllPub | Inside | ... | 0 | 0 | NaN | NaN |
| **3** | 70 | RL | 60.0 | 9550 | Pave | NaN | IR1 | Lvl | AllPub | Corner | ... | 0 | 0 | NaN | NaN |
| **4** | 60 | RL | 84.0 | 14260 | Pave | NaN | IR1 | Lvl | AllPub | FR2 | ... | 0 | 0 | NaN | NaN |

5 rows × 77 columns

In [25]:
```
df_train.Street.value_counts()
```

Out[25]:
```
Street
Pave    1454
Grvl       6
Name: count, dtype: int64
```

In [26]:
```
df_train.Utilities.value_counts()
```

Out[26]:
```
Utilities
AllPub    1459
NoSeWa       1
Name: count, dtype: int64
```

In [27]:
```
# We can also drop columns that show very low variance and thus not required for predictions
df_train = df_train.drop(['Street','Utilities'], axis = 1)
df_train.head()
```

Out[27]:

| | MSSubClass | MSZoning | LotFrontage | LotArea | Alley | LotShape | LandContour | LotConfig | LandSlope | Neighborhood | ... | ScreenPorch | PoolArea | PoolQ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 60 | RL | 65.0 | 8450 | NaN | Reg | Lvl | Inside | Gtl | CollgCr | ... | 0 | 0 | Na |
| **1** | 20 | RL | 80.0 | 9600 | NaN | Reg | Lvl | FR2 | Gtl | Veenker | ... | 0 | 0 | Na |
| **2** | 60 | RL | 68.0 | 11250 | NaN | IR1 | Lvl | Inside | Gtl | CollgCr | ... | 0 | 0 | Na |
| **3** | 70 | RL | 60.0 | 9550 | NaN | IR1 | Lvl | Corner | Gtl | Crawfor | ... | 0 | 0 | Na |
| **4** | 60 | RL | 84.0 | 14260 | NaN | IR1 | Lvl | FR2 | Gtl | NoRidge | ... | 0 | 0 | Na |

5 rows × 75 columns

In [28]:
```
#These Columns were having high null values, there was very little variance in the data. So we have decided to drop these columns.
df_train = df_train.drop(['PoolQC', 'Alley', 'MiscFeature','Fence', 'MasVnrType', 'FireplaceQu','LotFrontage','GarageYrBlt','MasVr
```

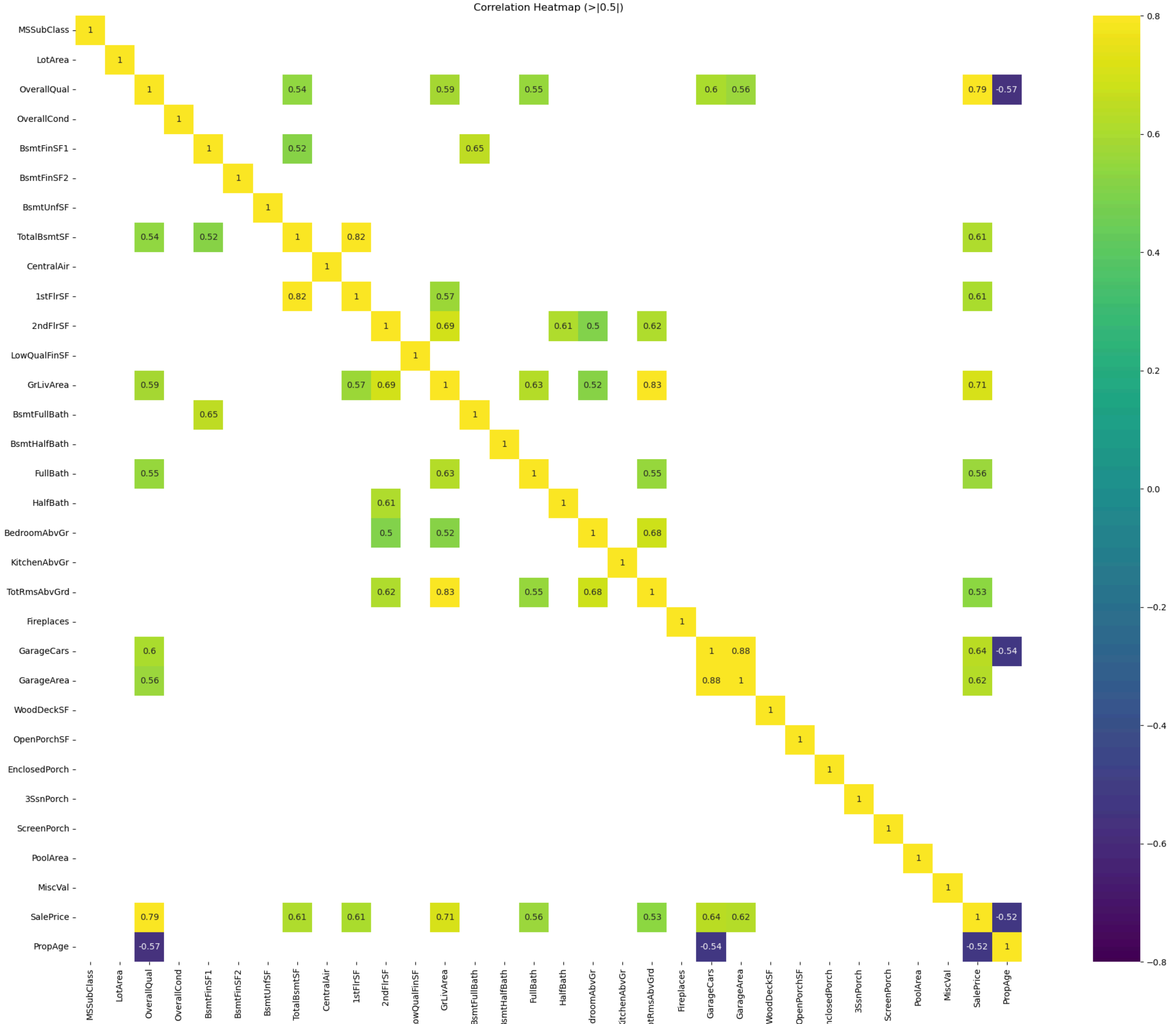
In [29]:
```
df_train_numeric = df_train.select_dtypes(include=['float64', 'int64'])
df_train_numeric.head()
```

Out[29]:

| | MSSubClass | LotArea | OverallQual | OverallCond | BsmtFinSF1 | BsmtFinSF2 | BsmtUnfSF | TotalBsmtSF | CentralAir | 1stFlrSF | ... | GarageArea | WoodDeckSF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 60 | 8450 | 7 | 5 | 706 | 0 | 150 | 856 | 1 | 856 | ... | 548 | 0 |
| **1** | 20 | 9600 | 6 | 8 | 978 | 0 | 284 | 1262 | 1 | 1262 | ... | 460 | 298 |
| **2** | 60 | 11250 | 7 | 5 | 486 | 0 | 434 | 920 | 1 | 920 | ... | 608 | 0 |
| **3** | 70 | 9550 | 7 | 5 | 216 | 0 | 540 | 756 | 1 | 961 | ... | 642 | 0 |
| **4** | 60 | 14260 | 8 | 5 | 655 | 0 | 490 | 1145 | 1 | 1145 | ... | 836 | 192 |

5 rows × 32 columns

In [30]:
```
# correlation matrix
corr_matrix = df_train_numeric.corr()

# Create a mask for correlations greater than 0.5 or less than -0.5
mask = (corr_matrix > 0.5) | (corr_matrix < -0.5)

# Plotting the heatmap using Matplotlib and Seaborn
plt.figure(figsize=(30,20))
sns.heatmap(corr_matrix[mask], vmin=-0.8, vmax=0.8, square=True, annot=True, cmap='viridis')

# Customize the plot
plt.title("Correlation Heatmap (>|0.5|)")
plt.show()
```

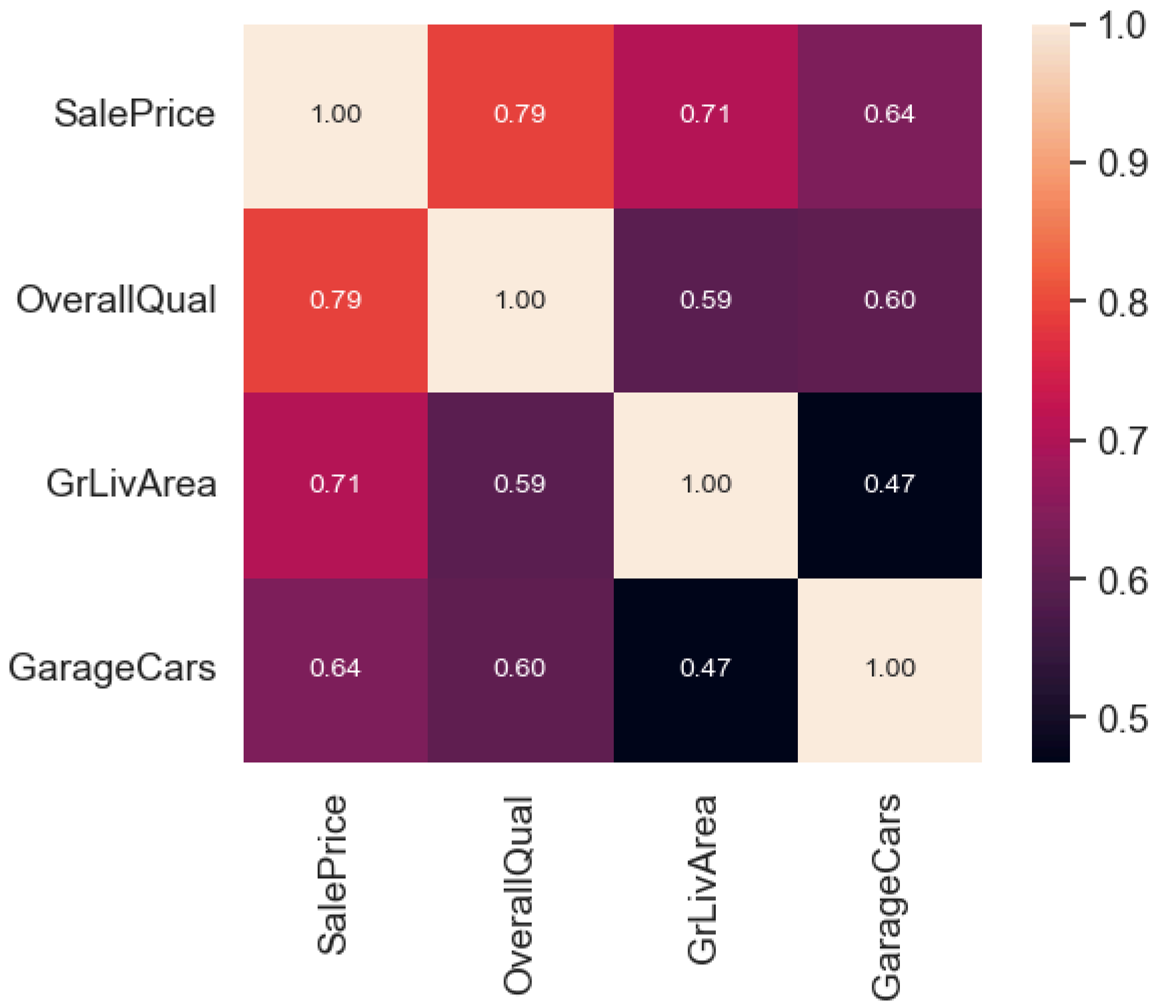

Correlation Heatmap (>|0.5|)
MSSubClass
LotArea
OverallQual
OverallCond
BsmtFinSF1
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
CentralAir
1stFlrSF
2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
TotRmsAbvGrd
Fireplaces
GarageCars
GarageArea
WoodDeckSF
OpenPorchSF
EnclosedPorch
3SsnPorch
ScreenPorch
PoolArea
MiscVal
SalePrice
PropAge
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8

This heatmap drives intuitions, that values greater then 0.6 or values less then -0.6 should be shown in the plot, also it helps in observing the strong or weak correlation accordingly.

Before dropping these columns, we will first check their predictive power

In [31]:
```
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars']

df_train_corr=df_train[cols]
```

In [32]:
```
#saleprice correlation matrix
corrmat = df_train_corr.corr()
k = 5 #number of variables for heatmap
cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index
cm = np.corrcoef(df_train[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'size': 10}, yticklabels=cols.values, xticklabels=c
plt.show()
```

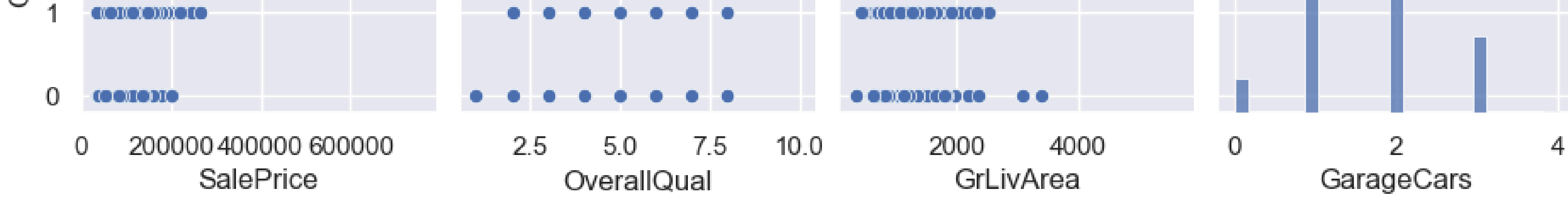



Correlation plots are instrumental in house price prediction machine learning. They help identify important features by revealing their strength of association with house prices. By prioritizing highly correlated features, correlation plots streamline feature selection and enhance model interpretability. Using this I could understand how much a feature affects the final price of the house. However a major problem with this data set is the fact that both variables are in the dataset which could lead to multicollinearity.

In [33]:
```
sns.set()
sns.pairplot(df_train_corr, size = 2.5)
plt.show();
```

```
C:\Users\bsash\anaconda3\Lib\site-packages\seaborn\axisgrid.py:2095: UserWarning: The `size` parameter has been renamed to `heigh
t`; please update your code.
  warnings.warn(msg, UserWarning)
C:\Users\bsash\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

SalePrice
600000
400000
200000
0
OverallQual
10
8
6
4
2
GrLivArea
5000
4000
3000
2000
1000
GarageCars
4
3
2




Scatter plots are essential visual tools in the realm of house price prediction. They helped me understand how different features relate to house prices by visualizing the relationships between them. Scatter plots helped me detecting any patterns, identify outliers, assess correlation, explore feature engineering possibilities, and evaluate model performance. This can be seen in the next phase of this asisgnment as I used the scatter plots to find outliers and hopefully it resulted in a more accurate representation of the housing market.

## One-Hot Encoding for Non-Numerical Variables

In [34]:
```
#Describing each field in the dataset *( Transpose the table)
df_train.describe(exclude=['float64', 'int64']).T
```

Out[34]:

| | count | unique | top | freq |
|---|---|---|---|---|
| **MSZoning** | 1460 | 5 | RL | 1151 |
| **LotShape** | 1460 | 4 | Reg | 925 |
| **LandContour** | 1460 | 4 | Lvl | 1311 |
| **LotConfig** | 1460 | 5 | Inside | 1052 |
| **LandSlope** | 1460 | 3 | Gtl | 1382 |
| **Neighborhood** | 1460 | 25 | NAmes | 225 |
| **Condition1** | 1460 | 9 | Norm | 1260 |
| **Condition2** | 1460 | 8 | Norm | 1445 |
| **BldgType** | 1460 | 5 | 1Fam | 1220 |
| **HouseStyle** | 1460 | 8 | 1Story | 726 |
| **RoofStyle** | 1460 | 6 | Gable | 1141 |
| **RoofMatl** | 1460 | 8 | CompShg | 1434 |
| **Exterior1st** | 1460 | 15 | VinylSd | 515 |
| **Exterior2nd** | 1460 | 16 | VinylSd | 504 |
| **ExterQual** | 1460 | 4 | TA | 906 |
| **ExterCond** | 1460 | 5 | TA | 1282 |
| **Foundation** | 1460 | 6 | PConc | 647 |
| **BsmtQual** | 1423 | 4 | TA | 649 |
| **BsmtCond** | 1423 | 4 | TA | 1311 |
| **BsmtExposure** | 1422 | 4 | No | 953 |
| **BsmtFinType1** | 1423 | 6 | Unf | 430 |
| **BsmtFinType2** | 1422 | 6 | Unf | 1256 |
| **Heating** | 1460 | 6 | GasA | 1428 |
| **HeatingQC** | 1460 | 5 | Ex | 741 |
| **Electrical** | 1459 | 5 | SBrkr | 1334 |
| **KitchenQual** | 1460 | 4 | TA | 735 |
| **Functional** | 1460 | 7 | Typ | 1360 |
| **GarageType** | 1379 | 6 | Attchd | 870 |

| | count | unique | top | freq |
|---|---|---|---|---|
| **GarageFinish** | 1379 | 3 | Unf | 605 |
| **GarageQual** | 1379 | 5 | TA | 1311 |
| **GarageCond** | 1379 | 5 | TA | 1326 |
| **PavedDrive** | 1460 | 3 | Y | 1340 |
| **SaleType** | 1460 | 9 | WD | 1267 |
| **SaleCondition** | 1460 | 6 | Normal | 1198 |

In [35]:
```
from sklearn.preprocessing import LabelEncoder
```

In [36]:
```
#diagnosis
cat_cols = ['MSZoning','LotShape','LandContour','LotConfig','LandSlope','Neighborhood','Condition1','Condition2','BldgType','House

#turning categories into their numerical counterparts using LabelEncoder
for var in cat_cols:
    number = LabelEncoder()
    df_train[var+"cat"] = number.fit_transform(df_train[var].astype('str'))
```

In [37]:
```
#Describing each field in the dataset *( Transpose the table)
df_train.describe(exclude="O").T
```

Out[37]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **MSSubClass** | 1460.0 | 56.897260 | 42.300571 | 20.0 | 20.0 | 50.0 | 70.00 | 190.0 |
| **LotArea** | 1460.0 | 10516.828082 | 9981.264932 | 1300.0 | 7553.5 | 9478.5 | 11601.50 | 215245.0 |
| **OverallQual** | 1460.0 | 6.099315 | 1.382997 | 1.0 | 5.0 | 6.0 | 7.00 | 10.0 |
| **OverallCond** | 1460.0 | 5.575342 | 1.112799 | 1.0 | 5.0 | 5.0 | 6.00 | 9.0 |
| **BsmtFinSF1** | 1460.0 | 443.639726 | 456.098091 | 0.0 | 0.0 | 383.5 | 712.25 | 5644.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **GarageQualcat** | 1460.0 | 3.927397 | 0.647822 | 0.0 | 4.0 | 4.0 | 4.00 | 5.0 |
| **GarageCondcat** | 1460.0 | 3.960959 | 0.566832 | 0.0 | 4.0 | 4.0 | 4.00 | 5.0 |
| **PavedDrivecat** | 1460.0 | 1.856164 | 0.496592 | 0.0 | 2.0 | 2.0 | 2.00 | 2.0 |
| **SaleTypecat** | 1460.0 | 7.513014 | 1.552100 | 0.0 | 8.0 | 8.0 | 8.00 | 8.0 |
| **SaleConditioncat** | 1460.0 | 3.770548 | 1.100854 | 0.0 | 4.0 | 4.0 | 4.00 | 5.0 |

66 rows × 8 columns

In [38]:

```
#creating a list of only numerical values
df_train_num = df_train.select_dtypes(exclude="O").select_dtypes(exclude=["category"])
df_train_num.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 66 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   MSSubClass        1460 non-null   int64
 1   LotArea           1460 non-null   int64
 2   OverallQual       1460 non-null   int64
 3   OverallCond       1460 non-null   int64
 4   BsmtFinSF1        1460 non-null   int64
 5   BsmtFinSF2        1460 non-null   int64
 6   BsmtUnfSF         1460 non-null   int64
 7   TotalBsmtSF       1460 non-null   int64
 8   CentralAir        1460 non-null   int64
 9   1stFlrSF          1460 non-null   int64
 10  2ndFlrSF          1460 non-null   int64
 11  LowQualFinSF      1460 non-null   int64
 12  GrLivArea         1460 non-null   int64
 13  BsmtFullBath      1460 non-null   int64
 14  BsmtHalfBath      1460 non-null   int64
 15  FullBath          1460 non-null   int64
 16  HalfBath          1460 non-null   int64
 17  BedroomAbvGr      1460 non-null   int64
 18  KitchenAbvGr      1460 non-null   int64
 19  TotRmsAbvGrd      1460 non-null   int64
 20  Fireplaces        1460 non-null   int64
 21  GarageCars        1460 non-null   int64
 22  GarageArea        1460 non-null   int64
 23  WoodDeckSF        1460 non-null   int64
 24  OpenPorchSF       1460 non-null   int64
 25  EnclosedPorch     1460 non-null   int64
 26  3SsnPorch         1460 non-null   int64
 27  ScreenPorch       1460 non-null   int64
 28  PoolArea          1460 non-null   int64
 29  MiscVal           1460 non-null   int64
 30  SalePrice         1460 non-null   int64
 31  PropAge           1460 non-null   int64
 32  MSZoningcat       1460 non-null   int32
 33  LotShapecat       1460 non-null   int32
 34  LandContourcat    1460 non-null   int32
 35  LotConfigcat      1460 non-null   int32
 36  LandSlopecat      1460 non-null   int32
 37  Neighborhoodcat   1460 non-null   int32
 38  Condition1cat     1460 non-null   int32
 39  Condition2cat     1460 non-null   int32
 40  BldgTypecat       1460 non-null   int32
 41  HouseStylecat     1460 non-null   int32
 42  RoofStylecat      1460 non-null   int32
 43  RoofMatlcat       1460 non-null   int32
 44  Exterior1stcat    1460 non-null   int32
```

```
 45  Exterior2ndcat    1460 non-null   int32
 46  ExterQualcat      1460 non-null   int32
 47  ExterCondcat      1460 non-null   int32
 48  Foundationcat     1460 non-null   int32
 49  BsmtQualcat       1460 non-null   int32
 50  BsmtCondcat       1460 non-null   int32
 51  BsmtExposurecat   1460 non-null   int32
 52  BsmtFinType1cat   1460 non-null   int32
 53  BsmtFinType2cat   1460 non-null   int32
 54  Heatingcat        1460 non-null   int32
 55  HeatingQCcat      1460 non-null   int32
 56  Electricalcat     1460 non-null   int32
 57  KitchenQualcat    1460 non-null   int32
 58  Functionalcat     1460 non-null   int32
 59  GarageTypecat     1460 non-null   int32
 60  GarageFinishcat   1460 non-null   int32
 61  GarageQualcat     1460 non-null   int32
 62  GarageCondcat     1460 non-null   int32
 63  PavedDrivecat     1460 non-null   int32
 64  SaleTypecat       1460 non-null   int32
 65  SaleConditioncat  1460 non-null   int32
dtypes: int32(34), int64(32)
memory usage: 559.0 KB
```

In [39]:

```
#Check for Missing Values
missing_df = df_train_num.isnull().sum().to_frame().rename(columns={0:"Total No. of Missing Values"})
missing_df["% of Missing Values"] = round((missing_df["Total No. of Missing Values"]/len(df_train_num))*100,2)
missing_df
```

Out[39]:

| | Total No. of Missing Values | % of Missing Values |
|---|---|---|
| **MSSubClass** | 0 | 0.0 |
| **LotArea** | 0 | 0.0 |
| **OverallQual** | 0 | 0.0 |
| **OverallCond** | 0 | 0.0 |
| **BsmtFinSF1** | 0 | 0.0 |
| **...** | ... | ... |
| **GarageQualcat** | 0 | 0.0 |
| **GarageCondcat** | 0 | 0.0 |
| **PavedDrivecat** | 0 | 0.0 |
| **SaleTypecat** | 0 | 0.0 |
| **SaleConditioncat** | 0 | 0.0 |

66 rows × 2 columns

In [40]:

```
#Check for Unique Values
null_train_data = round(100*(df_train_num.isnull().sum().sort_values(ascending=False)/len(df_train_num.index)),2)\
                    .to_frame().rename(columns={0:'Train Null values percentage'})[:20]
null_train_data
```

Out[40]:

| | Train Null values percentage |
|---|---|
| MSSubClass | 0.0 |
| BsmtQualcat | 0.0 |
| LotConfigcat | 0.0 |
| LandSlopecat | 0.0 |
| Neighborhoodcat | 0.0 |
| Condition1cat | 0.0 |
| Condition2cat | 0.0 |
| BldgTypecat | 0.0 |
| HouseStylecat | 0.0 |
| RoofStylecat | 0.0 |
| RoofMatlcat | 0.0 |
| Exterior1stcat | 0.0 |
| Exterior2ndcat | 0.0 |
| ExterQualcat | 0.0 |
| ExterCondcat | 0.0 |
| Foundationcat | 0.0 |
| BsmtCondcat | 0.0 |
| LotArea | 0.0 |
| BsmtExposurecat | 0.0 |
| BsmtFinType1cat | 0.0 |

# Building a model and evaluating

Having performed some exploratory data analysis and simple feature engineering as well as having ensured that all categorical values are encoded, we are now ready to proceed onto building our models.

Will evaluate using different learning models.

In [41]:

```
# Splitting the data into train and test
from sklearn.model_selection import train_test_split
```

```python
x = df_train_num.drop(labels=['SalePrice'],axis=1)
y = df_train_num['SalePrice']

x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.7, test_size=0.3, random_state=50)
```

In [42]:
```python
# Sklearn regression algorithms
from math import sqrt
from sklearn import linear_model
from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder

import warnings

from sklearn.datasets import make_classification
from numpy import mean


from sklearn import metrics
```

# Linear regression

Linear regression seeks to find the best-fitting straight line that describes the relationship between the input variables and the output variable. This line is determined by minimizing the difference between the predicted values from the model and the actual values observed in the data.However this model assumes that the model is a linear relationship, however in most real-life situations it is hard to draw a accurate conclusion soley on the use of a linear relationship between two variables

In [43]:
```python
# Use the Linear Regeression :
classifier = linear_model.LinearRegression()
```

In [44]:
```python
classifier.fit(x_train,y_train)
y_predict = classifier.predict(x_test)
```

In [45]:
```python
y_predict[:10]
```

Out[45]:
```
array([243398.69844144, 197387.04153552, 112712.12705638, 119282.1390447 ,
       277571.96555069, 180368.86804607, 153700.86242099, 154961.89101866,
       127573.51658206, 208980.06904305])
```

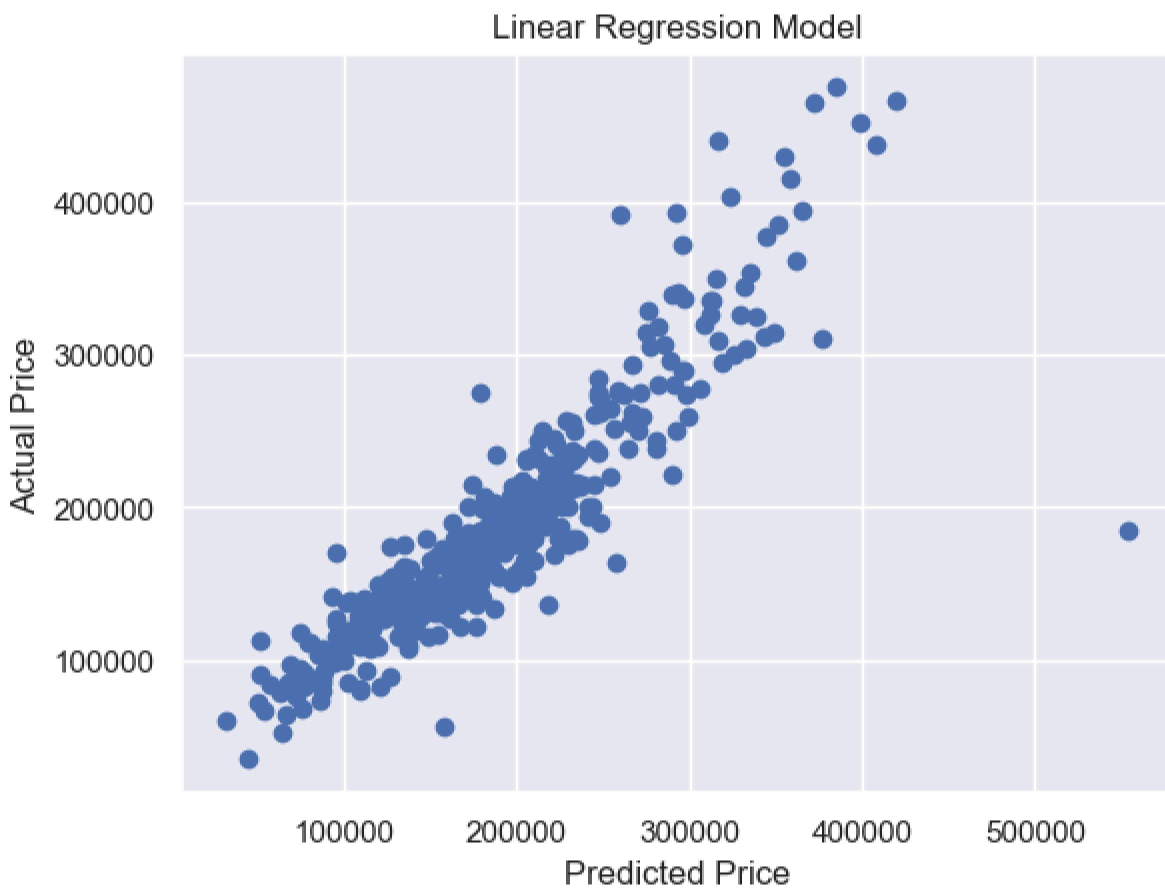
In [46]:
```python
accuracy_score = classifier.score(x_test,y_test)
print("Accuracy score          :", accuracy_score)
print('RMSE                    :', np.sqrt(metrics.mean_squared_error(y_test, y_predict)))
```

```
Accuracy score          : 0.8106186288143936
RMSE                    : 33088.260858499554
```

The linear regression model shows a adequate RSME and accuracy score. The RSME does however seem quite high as it is almost 33088 suggesting that there are some large errors

In [47]:
```
# alpha helps to show overlapping data
plt.scatter(y_predict, y_test, color = 'b')
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('Linear Regression Model')
```

Out[47]:
```
Text(0.5, 1.0, 'Linear Regression Model')
```




THe non-linear model I chose is the Random Forest, this is because random forest is a machine learning algorithm that builds multiple decision trees during training, each trained on a random subset of the data and using a random subset of features at each split. It combines the predictions of these trees (either by voting for classification or averaging for regression) to produce a final prediction. Random Forest is known for its high accuracy, resistance to overfitting, and ability to handle noisy data effectively.

## Random Forest Regressor

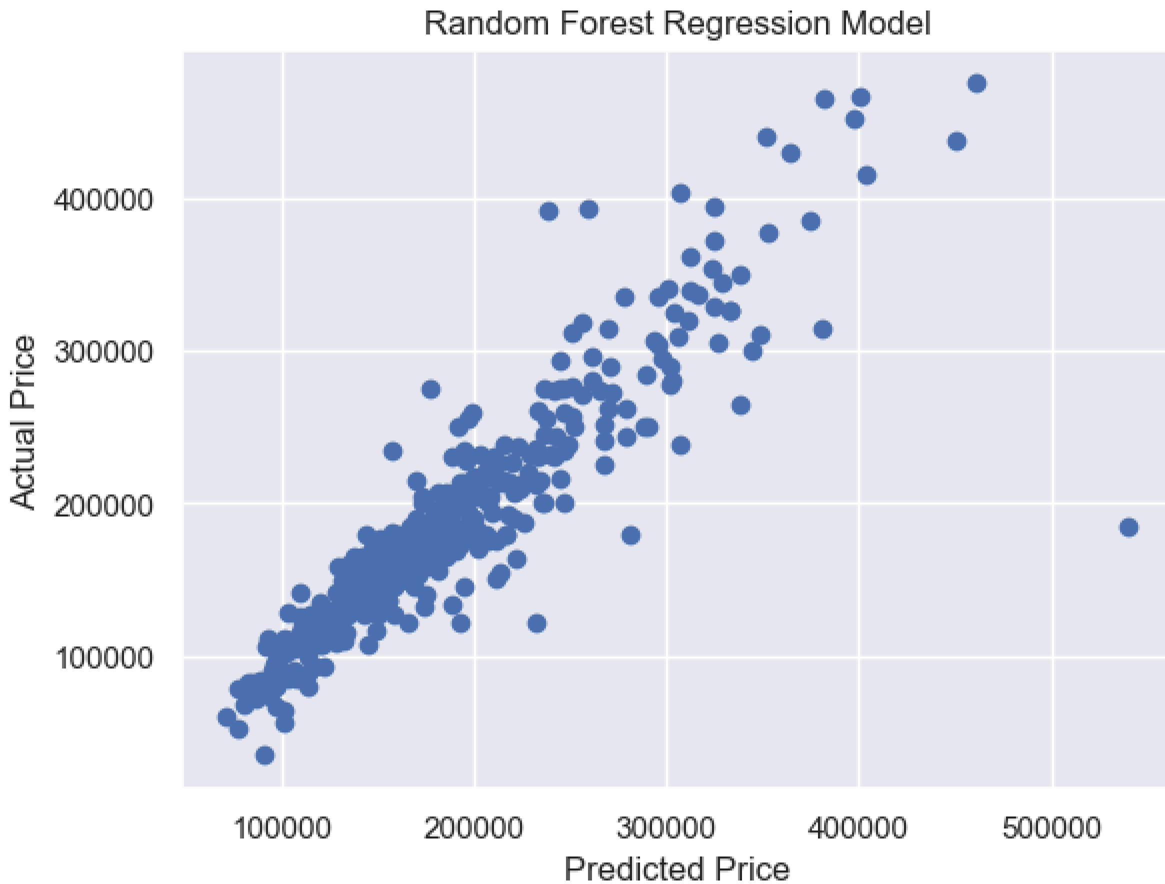
In [48]:
```
from sklearn.ensemble import RandomForestRegressor
classifier=RandomForestRegressor()
classifier.fit(x_train,y_train)
y_predict=classifier.predict(x_test)
```

In [49]:
```
accuracy_score = classifier.score(x_test,y_test)
print("Accuracy score          :", accuracy_score)
print('RMSE                    :', np.sqrt(metrics.mean_squared_error(y_test, y_predict)))
```

```
Accuracy score          : 0.8304162115014839
RMSE                    : 31311.039004774546
```

In [50]:
```
# alpha helps to show overlapping data
plt.scatter(y_predict, y_test, color = 'b')
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('Random Forest Regression Model')
```

Out[50]:
```
Text(0.5, 1.0, 'Random Forest Regression Model')
```




## GradientBoosting Regressor

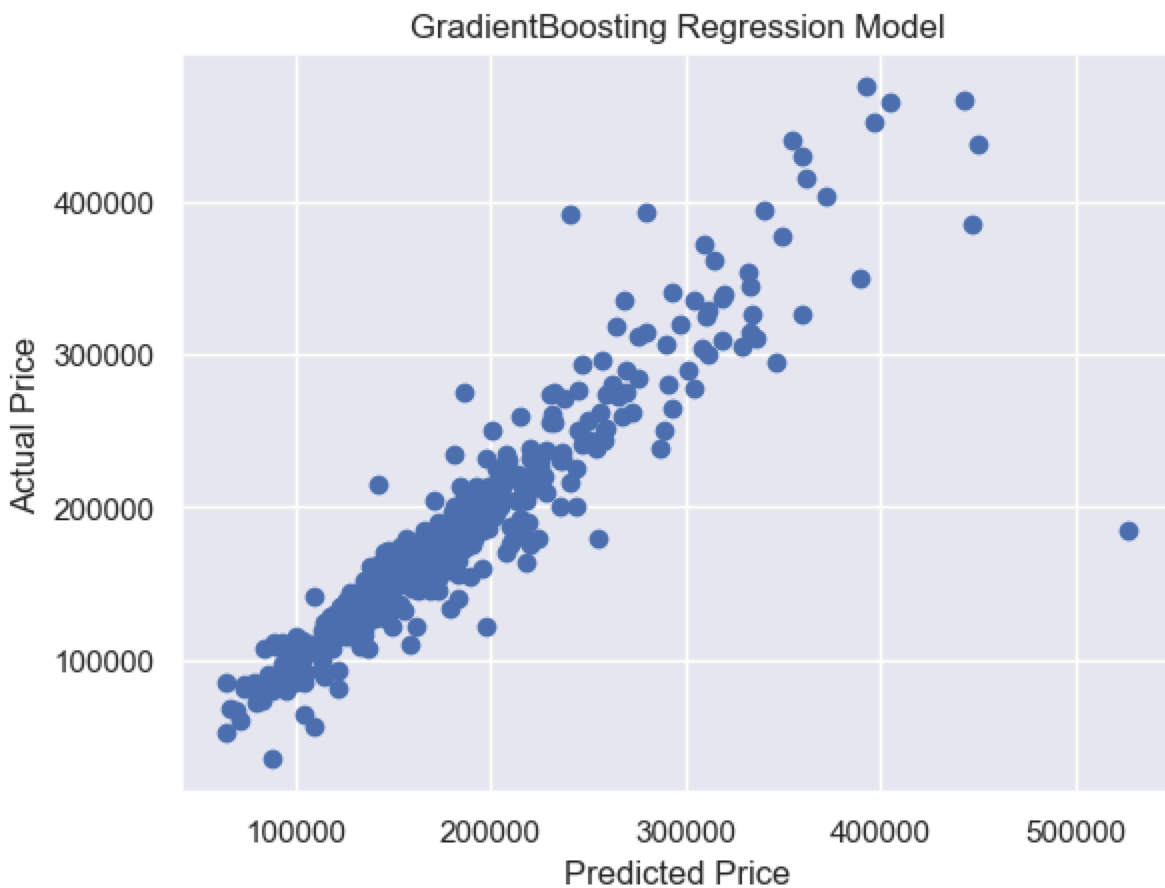
In [51]:
```
from sklearn.ensemble import GradientBoostingRegressor
classifier=GradientBoostingRegressor()
classifier.fit(x_train,y_train)
y_predict=classifier.predict(x_test)
```

In [52]:
```
accuracy_score = classifier.score(x_test,y_test)
print("Accuracy score          :", accuracy_score)
print('RMSE                    :', np.sqrt(metrics.mean_squared_error(y_test, y_predict)))
```

```
Accuracy score          : 0.8596520212596312
RMSE                    : 28484.486443174224
```

In [53]:
```
# alpha helps to show overlapping data
plt.scatter(y_predict, y_test, color = 'b')
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('GradientBoosting Regression Model')
```

Out[53]: Text(0.5, 1.0, 'GradientBoosting Regression Model')




## CatBoostRegressor

In [59]:
```
#pip install catboost
```

In [55]:
```
from catboost import CatBoostRegressor
classifier=CatBoostRegressor()
classifier.fit(x_train,y_train)
y_predict=classifier.predict(x_test)
```

```
Learning rate set to 0.041078
0:	learn: 78864.0153391	total: 155ms	remaining: 2m 35s
1:	learn: 76824.8299083	total: 161ms	remaining: 1m 20s
2:	learn: 74856.0998091	total: 167ms	remaining: 55.4s
3:	learn: 73082.4951374	total: 173ms	remaining: 43.1s
4:	learn: 71449.8163162	total: 181ms	remaining: 36.1s
5:	learn: 69531.5871880	total: 189ms	remaining: 31.2s
6:	learn: 67932.8137107	total: 196ms	remaining: 27.8s
7:	learn: 66492.0220364	total: 203ms	remaining: 25.2s
8:	learn: 64896.8308637	total: 210ms	remaining: 23.1s
9:	learn: 63448.3549950	total: 220ms	remaining: 21.8s
10:	learn: 62023.2143794	total: 225ms	remaining: 20.2s
11:	learn: 60664.9056151	total: 228ms	remaining: 18.8s
12:	learn: 59301.6234201	total: 234ms	remaining: 17.8s
13:	learn: 57981.2056392	total: 239ms	remaining: 16.8s
14:	learn: 56773.4508706	total: 245ms	remaining: 16.1s
15:	learn: 55432.3320375	total: 253ms	remaining: 15.6s
16:	learn: 54224.5567980	total: 261ms	remaining: 15.1s
17:	learn: 53042.8482533	total: 268ms	remaining: 14.6s
18:	learn: 51945.7423794	total: 274ms	remaining: 14.1s
19:	learn: 50796.0057868	total: 281ms	remaining: 13.8s
20:	learn: 49932.5804049	total: 289ms	remaining: 13.5s
21:	learn: 48966.0560175	total: 296ms	remaining: 13.2s
22:	learn: 47974.5159658	total: 302ms	remaining: 12.8s
23:	learn: 47082.3212541	total: 309ms	remaining: 12.6s
24:	learn: 46132.9682992	total: 314ms	remaining: 12.3s
25:	learn: 45259.4522350	total: 321ms	remaining: 12s
26:	learn: 44531.6080250	total: 327ms	remaining: 11.8s
27:	learn: 43751.8453380	total: 334ms	remaining: 11.6s
28:	learn: 42959.9258640	total: 341ms	remaining: 11.4s
29:	learn: 42113.1336584	total: 349ms	remaining: 11.3s
30:	learn: 41453.7601454	total: 356ms	remaining: 11.1s
31:	learn: 40754.7430398	total: 362ms	remaining: 10.9s
32:	learn: 40094.4462832	total: 370ms	remaining: 10.8s
33:	learn: 39483.4238886	total: 377ms	remaining: 10.7s
34:	learn: 38876.1830212	total: 382ms	remaining: 10.5s
35:	learn: 38262.4377835	total: 388ms	remaining: 10.4s
36:	learn: 37692.9419898	total: 394ms	remaining: 10.2s
37:	learn: 37188.9308207	total: 400ms	remaining: 10.1s
38:	learn: 36632.8383772	total: 407ms	remaining: 10s
39:	learn: 36130.8681971	total: 413ms	remaining: 9.92s
40:	learn: 35659.7744123	total: 419ms	remaining: 9.79s
41:	learn: 35216.2569643	total: 434ms	remaining: 9.89s
42:	learn: 34739.7763703	total: 440ms	remaining: 9.79s
43:	learn: 34365.1423835	total: 447ms	remaining: 9.7s
44:	learn: 33856.0510891	total: 453ms	remaining: 9.61s
45:	learn: 33338.3102581	total: 460ms	remaining: 9.54s
46:	learn: 32836.7644395	total: 465ms	remaining: 9.43s
47:	learn: 32416.6460796	total: 471ms	remaining: 9.34s
48:	learn: 32009.0939540	total: 476ms	remaining: 9.24s
```

```
49:	learn: 31599.4847664	total: 480ms	remaining: 9.12s
50:	learn: 31198.3947008	total: 486ms	remaining: 9.03s
51:	learn: 30891.9371455	total: 492ms	remaining: 8.96s
52:	learn: 30534.0658805	total: 497ms	remaining: 8.89s
53:	learn: 30167.6337196	total: 506ms	remaining: 8.86s
54:	learn: 29857.7139524	total: 513ms	remaining: 8.81s
55:	learn: 29543.7078088	total: 521ms	remaining: 8.78s
56:	learn: 29174.4319033	total: 527ms	remaining: 8.71s
57:	learn: 28875.0377421	total: 532ms	remaining: 8.64s
58:	learn: 28623.9513867	total: 539ms	remaining: 8.59s
59:	learn: 28348.3708006	total: 546ms	remaining: 8.56s
60:	learn: 28033.2817621	total: 552ms	remaining: 8.5s
61:	learn: 27762.0916823	total: 560ms	remaining: 8.48s
62:	learn: 27507.0630057	total: 567ms	remaining: 8.43s
63:	learn: 27300.3307534	total: 572ms	remaining: 8.37s
64:	learn: 27037.0750996	total: 579ms	remaining: 8.33s
65:	learn: 26836.9055382	total: 584ms	remaining: 8.27s
66:	learn: 26612.6765520	total: 589ms	remaining: 8.2s
67:	learn: 26437.8967621	total: 595ms	remaining: 8.16s
68:	learn: 26274.0211746	total: 603ms	remaining: 8.13s
69:	learn: 26089.3282168	total: 609ms	remaining: 8.08s
70:	learn: 25947.6212097	total: 616ms	remaining: 8.05s
71:	learn: 25749.4654688	total: 622ms	remaining: 8.02s
72:	learn: 25599.8088651	total: 631ms	remaining: 8.01s
73:	learn: 25424.6965178	total: 638ms	remaining: 7.98s
74:	learn: 25231.6435726	total: 646ms	remaining: 7.96s
75:	learn: 25048.4304718	total: 655ms	remaining: 7.96s
76:	learn: 24893.5152235	total: 662ms	remaining: 7.93s
77:	learn: 24728.3712339	total: 668ms	remaining: 7.89s
78:	learn: 24502.6221502	total: 675ms	remaining: 7.87s
79:	learn: 24352.1461650	total: 684ms	remaining: 7.87s
80:	learn: 24176.3957608	total: 689ms	remaining: 7.82s
81:	learn: 24021.8287562	total: 695ms	remaining: 7.78s
82:	learn: 23878.4998530	total: 702ms	remaining: 7.75s
83:	learn: 23751.9052547	total: 711ms	remaining: 7.75s
84:	learn: 23600.6755632	total: 719ms	remaining: 7.74s
85:	learn: 23478.1193434	total: 728ms	remaining: 7.74s
86:	learn: 23322.4052826	total: 737ms	remaining: 7.74s
87:	learn: 23208.6374276	total: 745ms	remaining: 7.72s
88:	learn: 23078.5079269	total: 752ms	remaining: 7.7s
89:	learn: 22953.5796611	total: 761ms	remaining: 7.69s
90:	learn: 22848.6848613	total: 769ms	remaining: 7.68s
91:	learn: 22713.3727922	total: 778ms	remaining: 7.67s
92:	learn: 22677.4695921	total: 784ms	remaining: 7.65s
93:	learn: 22547.8722861	total: 794ms	remaining: 7.66s
94:	learn: 22389.6009784	total: 802ms	remaining: 7.64s
95:	learn: 22268.2689622	total: 809ms	remaining: 7.62s
96:	learn: 22129.2206614	total: 816ms	remaining: 7.59s
97:	learn: 22004.3103531	total: 822ms	remaining: 7.57s
98:	learn: 21882.9385055	total: 829ms	remaining: 7.54s
```

```
99:     learn: 21793.3785713    total: 835ms    remaining: 7.52s
100:    learn: 21706.6728038    total: 842ms    remaining: 7.49s
101:    learn: 21615.4486904    total: 849ms    remaining: 7.47s
102:    learn: 21525.1133183    total: 856ms    remaining: 7.45s
103:    learn: 21441.6582808    total: 862ms    remaining: 7.43s
104:    learn: 21316.7259279    total: 869ms    remaining: 7.4s
105:    learn: 21197.8045394    total: 876ms    remaining: 7.39s
106:    learn: 21115.5973351    total: 881ms    remaining: 7.35s
107:    learn: 21019.4588956    total: 892ms    remaining: 7.36s
108:    learn: 20944.3577495    total: 901ms    remaining: 7.36s
109:    learn: 20846.4860385    total: 908ms    remaining: 7.34s
110:    learn: 20775.9002164    total: 916ms    remaining: 7.33s
111:    learn: 20696.6179485    total: 923ms    remaining: 7.31s
112:    learn: 20598.0451622    total: 930ms    remaining: 7.3s
113:    learn: 20518.3753987    total: 936ms    remaining: 7.28s
114:    learn: 20436.2978770    total: 942ms    remaining: 7.25s
115:    learn: 20367.7375972    total: 948ms    remaining: 7.22s
116:    learn: 20262.9349184    total: 956ms    remaining: 7.21s
117:    learn: 20190.8047773    total: 963ms    remaining: 7.19s
118:    learn: 20115.0898681    total: 970ms    remaining: 7.18s
119:    learn: 20035.2089913    total: 977ms    remaining: 7.16s
120:    learn: 19945.3052721    total: 985ms    remaining: 7.15s
121:    learn: 19889.3235478    total: 993ms    remaining: 7.14s
122:    learn: 19814.3288555    total: 1s       remaining: 7.13s
123:    learn: 19741.7021682    total: 1s       remaining: 7.11s
124:    learn: 19677.5915978    total: 1.01s    remaining: 7.09s
125:    learn: 19595.7007200    total: 1.02s    remaining: 7.07s
126:    learn: 19553.3013738    total: 1.02s    remaining: 7.04s
127:    learn: 19485.6581557    total: 1.03s    remaining: 7.03s
128:    learn: 19430.8470685    total: 1.04s    remaining: 7.02s
129:    learn: 19377.5608338    total: 1.05s    remaining: 7.01s
130:    learn: 19341.6606081    total: 1.05s    remaining: 6.99s
131:    learn: 19295.0811816    total: 1.06s    remaining: 6.98s
132:    learn: 19242.9052543    total: 1.07s    remaining: 6.98s
133:    learn: 19211.9781149    total: 1.08s    remaining: 6.97s
134:    learn: 19153.6566396    total: 1.08s    remaining: 6.96s
135:    learn: 19104.0609265    total: 1.09s    remaining: 6.94s
136:    learn: 19069.5307624    total: 1.1s     remaining: 6.92s
137:    learn: 19036.5557451    total: 1.11s    remaining: 6.92s
138:    learn: 18998.2437688    total: 1.11s    remaining: 6.9s
139:    learn: 18933.1467217    total: 1.12s    remaining: 6.89s
140:    learn: 18888.8709404    total: 1.13s    remaining: 6.87s
141:    learn: 18842.2507336    total: 1.13s    remaining: 6.86s
142:    learn: 18766.3603219    total: 1.14s    remaining: 6.85s
143:    learn: 18688.3165197    total: 1.15s    remaining: 6.83s
144:    learn: 18614.6832623    total: 1.16s    remaining: 6.82s
145:    learn: 18580.4983024    total: 1.16s    remaining: 6.79s
146:    learn: 18532.1603629    total: 1.17s    remaining: 6.78s
147:    learn: 18484.3946554    total: 1.17s    remaining: 6.75s
148:    learn: 18436.3536644    total: 1.18s    remaining: 6.74s
```

```
149:	learn: 18368.7057444	total: 1.19s	remaining: 6.72s
150:	learn: 18335.7553644	total: 1.19s	remaining: 6.68s
151:	learn: 18250.0296067	total: 1.2s	remaining: 6.67s
152:	learn: 18205.3532946	total: 1.2s	remaining: 6.65s
153:	learn: 18153.3265455	total: 1.21s	remaining: 6.64s
154:	learn: 18106.7793187	total: 1.22s	remaining: 6.64s
155:	learn: 18043.8613971	total: 1.23s	remaining: 6.63s
156:	learn: 17987.0603533	total: 1.23s	remaining: 6.61s
157:	learn: 17926.6744689	total: 1.24s	remaining: 6.59s
158:	learn: 17884.9316011	total: 1.24s	remaining: 6.58s
159:	learn: 17839.4931430	total: 1.25s	remaining: 6.55s
160:	learn: 17784.8111163	total: 1.25s	remaining: 6.54s
161:	learn: 17753.0839737	total: 1.26s	remaining: 6.52s
162:	learn: 17715.5745673	total: 1.27s	remaining: 6.5s
163:	learn: 17668.3007272	total: 1.27s	remaining: 6.49s
164:	learn: 17619.8720762	total: 1.28s	remaining: 6.47s
165:	learn: 17562.5889821	total: 1.28s	remaining: 6.45s
166:	learn: 17525.7983470	total: 1.29s	remaining: 6.43s
167:	learn: 17516.9224470	total: 1.29s	remaining: 6.4s
168:	learn: 17490.8676350	total: 1.3s	remaining: 6.39s
169:	learn: 17451.3598271	total: 1.3s	remaining: 6.38s
170:	learn: 17397.4441501	total: 1.31s	remaining: 6.37s
171:	learn: 17349.7314468	total: 1.32s	remaining: 6.36s
172:	learn: 17301.3053678	total: 1.33s	remaining: 6.35s
173:	learn: 17251.4945486	total: 1.34s	remaining: 6.34s
174:	learn: 17212.0660983	total: 1.34s	remaining: 6.33s
175:	learn: 17162.3299607	total: 1.35s	remaining: 6.31s
176:	learn: 17110.0081777	total: 1.35s	remaining: 6.3s
177:	learn: 17066.4557184	total: 1.36s	remaining: 6.28s
178:	learn: 17009.2413131	total: 1.37s	remaining: 6.27s
179:	learn: 16999.4885907	total: 1.38s	remaining: 6.29s
180:	learn: 16963.9465296	total: 1.39s	remaining: 6.29s
181:	learn: 16930.6884288	total: 1.4s	remaining: 6.28s
182:	learn: 16898.4622618	total: 1.4s	remaining: 6.27s
183:	learn: 16890.7693317	total: 1.41s	remaining: 6.26s
184:	learn: 16851.7802672	total: 1.42s	remaining: 6.27s
185:	learn: 16822.4211256	total: 1.43s	remaining: 6.26s
186:	learn: 16814.1570671	total: 1.44s	remaining: 6.25s
187:	learn: 16790.7969652	total: 1.45s	remaining: 6.26s
188:	learn: 16759.0204735	total: 1.46s	remaining: 6.25s
189:	learn: 16733.0442272	total: 1.46s	remaining: 6.24s
190:	learn: 16692.7201633	total: 1.47s	remaining: 6.24s
191:	learn: 16675.6557732	total: 1.48s	remaining: 6.24s
192:	learn: 16623.5684477	total: 1.49s	remaining: 6.23s
193:	learn: 16609.5245621	total: 1.5s	remaining: 6.23s
194:	learn: 16594.2229086	total: 1.51s	remaining: 6.23s
195:	learn: 16586.2965899	total: 1.52s	remaining: 6.22s
196:	learn: 16560.1546353	total: 1.53s	remaining: 6.23s
197:	learn: 16546.8833825	total: 1.54s	remaining: 6.22s
198:	learn: 16513.8392910	total: 1.54s	remaining: 6.22s
```

```
199:	learn: 16474.7295642	total: 1.55s	remaining: 6.22s
200:	learn: 16414.9951969	total: 1.56s	remaining: 6.22s
201:	learn: 16408.0482145	total: 1.57s	remaining: 6.22s
202:	learn: 16397.6301681	total: 1.58s	remaining: 6.22s
203:	learn: 16385.3290187	total: 1.59s	remaining: 6.22s
204:	learn: 16377.7012962	total: 1.6s	remaining: 6.23s
205:	learn: 16372.0443026	total: 1.61s	remaining: 6.22s
206:	learn: 16351.0451933	total: 1.62s	remaining: 6.22s
207:	learn: 16292.8077042	total: 1.63s	remaining: 6.21s
208:	learn: 16284.7143483	total: 1.64s	remaining: 6.2s
209:	learn: 16276.1803659	total: 1.65s	remaining: 6.19s
210:	learn: 16268.8685090	total: 1.65s	remaining: 6.18s
211:	learn: 16263.1847209	total: 1.66s	remaining: 6.17s
212:	learn: 16256.8499816	total: 1.67s	remaining: 6.15s
213:	learn: 16249.5187523	total: 1.67s	remaining: 6.14s
214:	learn: 16223.4190337	total: 1.68s	remaining: 6.14s
215:	learn: 16182.8044628	total: 1.69s	remaining: 6.13s
216:	learn: 16148.9755757	total: 1.7s	remaining: 6.12s
217:	learn: 16116.7436037	total: 1.7s	remaining: 6.11s
218:	learn: 16099.2166823	total: 1.71s	remaining: 6.11s
219:	learn: 16083.9627142	total: 1.72s	remaining: 6.11s
220:	learn: 16077.3816311	total: 1.73s	remaining: 6.1s
221:	learn: 16045.7364206	total: 1.74s	remaining: 6.09s
222:	learn: 15992.1479963	total: 1.75s	remaining: 6.09s
223:	learn: 15958.0613187	total: 1.76s	remaining: 6.09s
224:	learn: 15914.8073289	total: 1.77s	remaining: 6.09s
225:	learn: 15903.9764553	total: 1.77s	remaining: 6.08s
226:	learn: 15875.3065550	total: 1.78s	remaining: 6.08s
227:	learn: 15839.8353903	total: 1.79s	remaining: 6.07s
228:	learn: 15797.4851537	total: 1.8s	remaining: 6.06s
229:	learn: 15775.3580429	total: 1.81s	remaining: 6.05s
230:	learn: 15733.6456792	total: 1.82s	remaining: 6.05s
231:	learn: 15728.7208842	total: 1.82s	remaining: 6.04s
232:	learn: 15682.7115539	total: 1.83s	remaining: 6.03s
233:	learn: 15655.3089235	total: 1.84s	remaining: 6.03s
234:	learn: 15613.0373781	total: 1.85s	remaining: 6.02s
235:	learn: 15583.8988138	total: 1.86s	remaining: 6.01s
236:	learn: 15579.4824407	total: 1.86s	remaining: 6s
237:	learn: 15548.9510238	total: 1.87s	remaining: 6s
238:	learn: 15534.8923313	total: 1.88s	remaining: 6s
239:	learn: 15517.9460954	total: 1.89s	remaining: 5.99s
240:	learn: 15495.9124097	total: 1.9s	remaining: 5.97s
241:	learn: 15445.7827105	total: 1.9s	remaining: 5.96s
242:	learn: 15413.9199446	total: 1.91s	remaining: 5.95s
243:	learn: 15399.2108149	total: 1.91s	remaining: 5.93s
244:	learn: 15369.5055205	total: 1.92s	remaining: 5.92s
245:	learn: 15360.0512843	total: 1.93s	remaining: 5.91s
246:	learn: 15355.1254652	total: 1.93s	remaining: 5.89s
247:	learn: 15308.4539306	total: 1.94s	remaining: 5.88s
248:	learn: 15258.4840205	total: 1.95s	remaining: 5.87s
```

```
249:	learn: 15243.1929942	total: 1.95s	remaining: 5.86s
250:	learn: 15207.2698625	total: 1.96s	remaining: 5.85s
251:	learn: 15182.5594836	total: 1.97s	remaining: 5.84s
252:	learn: 15154.4768667	total: 1.98s	remaining: 5.83s
253:	learn: 15116.5084770	total: 1.98s	remaining: 5.83s
254:	learn: 15084.6654006	total: 1.99s	remaining: 5.81s
255:	learn: 15064.1252530	total: 2s	remaining: 5.8s
256:	learn: 15019.7271907	total: 2s	remaining: 5.79s
257:	learn: 15015.4938552	total: 2.01s	remaining: 5.78s
258:	learn: 15007.7957512	total: 2.02s	remaining: 5.77s
259:	learn: 15000.2496178	total: 2.02s	remaining: 5.76s
260:	learn: 14964.1943283	total: 2.03s	remaining: 5.75s
261:	learn: 14917.5348686	total: 2.04s	remaining: 5.75s
262:	learn: 14909.8526544	total: 2.05s	remaining: 5.74s
263:	learn: 14871.1888771	total: 2.06s	remaining: 5.73s
264:	learn: 14832.1783885	total: 2.06s	remaining: 5.72s
265:	learn: 14783.5763959	total: 2.07s	remaining: 5.7s
266:	learn: 14777.3322714	total: 2.07s	remaining: 5.69s
267:	learn: 14741.7933166	total: 2.08s	remaining: 5.68s
268:	learn: 14708.1018878	total: 2.09s	remaining: 5.67s
269:	learn: 14681.2643083	total: 2.09s	remaining: 5.66s
270:	learn: 14642.3780340	total: 2.1s	remaining: 5.65s
271:	learn: 14615.5485372	total: 2.11s	remaining: 5.64s
272:	learn: 14587.3060733	total: 2.12s	remaining: 5.63s
273:	learn: 14541.8862144	total: 2.12s	remaining: 5.62s
274:	learn: 14513.7679591	total: 2.13s	remaining: 5.61s
275:	learn: 14509.3838585	total: 2.13s	remaining: 5.6s
276:	learn: 14478.4618787	total: 2.14s	remaining: 5.59s
277:	learn: 14471.4604653	total: 2.15s	remaining: 5.58s
278:	learn: 14455.1615247	total: 2.16s	remaining: 5.57s
279:	learn: 14432.2480778	total: 2.16s	remaining: 5.56s
280:	learn: 14395.5438388	total: 2.17s	remaining: 5.55s
281:	learn: 14355.5344017	total: 2.18s	remaining: 5.54s
282:	learn: 14319.3885873	total: 2.18s	remaining: 5.53s
283:	learn: 14313.1260768	total: 2.19s	remaining: 5.52s
284:	learn: 14291.0759971	total: 2.2s	remaining: 5.51s
285:	learn: 14285.9474981	total: 2.2s	remaining: 5.5s
286:	learn: 14261.0406462	total: 2.21s	remaining: 5.49s
287:	learn: 14226.5332266	total: 2.23s	remaining: 5.52s
288:	learn: 14187.7114265	total: 2.25s	remaining: 5.53s
289:	learn: 14167.6153762	total: 2.25s	remaining: 5.51s
290:	learn: 14160.5223401	total: 2.26s	remaining: 5.51s
291:	learn: 14122.8478560	total: 2.27s	remaining: 5.5s
292:	learn: 14093.9779118	total: 2.27s	remaining: 5.49s
293:	learn: 14071.9971530	total: 2.28s	remaining: 5.47s
294:	learn: 14045.7944882	total: 2.29s	remaining: 5.46s
295:	learn: 14024.6604092	total: 2.29s	remaining: 5.45s
296:	learn: 14011.5926725	total: 2.3s	remaining: 5.45s
297:	learn: 13987.3883451	total: 2.31s	remaining: 5.44s
298:	learn: 13943.6841315	total: 2.32s	remaining: 5.44s
```

```
299:	learn: 13910.1901308	total: 2.33s	remaining: 5.42s
300:	learn: 13885.8274342	total: 2.33s	remaining: 5.42s
301:	learn: 13858.2820129	total: 2.34s	remaining: 5.41s
302:	learn: 13828.4068713	total: 2.35s	remaining: 5.4s
303:	learn: 13788.0596850	total: 2.35s	remaining: 5.39s
304:	learn: 13756.0410348	total: 2.36s	remaining: 5.39s
305:	learn: 13730.3943968	total: 2.37s	remaining: 5.38s
306:	learn: 13725.6297816	total: 2.38s	remaining: 5.37s
307:	learn: 13686.2556730	total: 2.39s	remaining: 5.36s
308:	learn: 13657.4670234	total: 2.39s	remaining: 5.36s
309:	learn: 13632.9511412	total: 2.4s	remaining: 5.34s
310:	learn: 13608.3975990	total: 2.41s	remaining: 5.34s
311:	learn: 13580.8109287	total: 2.42s	remaining: 5.33s
312:	learn: 13561.5645151	total: 2.42s	remaining: 5.32s
313:	learn: 13537.1710620	total: 2.43s	remaining: 5.31s
314:	learn: 13514.3421829	total: 2.44s	remaining: 5.3s
315:	learn: 13489.7432498	total: 2.45s	remaining: 5.3s
316:	learn: 13467.2121905	total: 2.46s	remaining: 5.29s
317:	learn: 13449.0724851	total: 2.46s	remaining: 5.29s
318:	learn: 13441.1366382	total: 2.47s	remaining: 5.28s
319:	learn: 13411.2303495	total: 2.48s	remaining: 5.27s
320:	learn: 13377.7962128	total: 2.49s	remaining: 5.26s
321:	learn: 13349.8173845	total: 2.5s	remaining: 5.26s
322:	learn: 13327.0165520	total: 2.51s	remaining: 5.25s
323:	learn: 13295.3287611	total: 2.52s	remaining: 5.25s
324:	learn: 13274.5740682	total: 2.52s	remaining: 5.24s
325:	learn: 13228.3516933	total: 2.53s	remaining: 5.24s
326:	learn: 13212.0492360	total: 2.54s	remaining: 5.23s
327:	learn: 13173.3831852	total: 2.55s	remaining: 5.22s
328:	learn: 13164.4080867	total: 2.56s	remaining: 5.22s
329:	learn: 13144.0208675	total: 2.57s	remaining: 5.22s
330:	learn: 13126.8841899	total: 2.58s	remaining: 5.22s
331:	learn: 13106.4437285	total: 2.59s	remaining: 5.22s
332:	learn: 13082.8894711	total: 2.6s	remaining: 5.21s
333:	learn: 13064.9157206	total: 2.61s	remaining: 5.21s
334:	learn: 13039.0952659	total: 2.62s	remaining: 5.21s
335:	learn: 13036.1767696	total: 2.63s	remaining: 5.2s
336:	learn: 13034.0734925	total: 2.64s	remaining: 5.2s
337:	learn: 13019.1882487	total: 2.65s	remaining: 5.19s
338:	learn: 13003.7034216	total: 2.66s	remaining: 5.18s
339:	learn: 13000.8878378	total: 2.67s	remaining: 5.18s
340:	learn: 12982.8441633	total: 2.67s	remaining: 5.17s
341:	learn: 12976.0996341	total: 2.68s	remaining: 5.16s
342:	learn: 12952.5342977	total: 2.69s	remaining: 5.15s
343:	learn: 12932.6683476	total: 2.7s	remaining: 5.14s
344:	learn: 12930.4688891	total: 2.71s	remaining: 5.14s
345:	learn: 12918.6449073	total: 2.72s	remaining: 5.13s
346:	learn: 12898.7029840	total: 2.73s	remaining: 5.13s
347:	learn: 12874.2410105	total: 2.74s	remaining: 5.13s
348:	learn: 12857.6097098	total: 2.74s	remaining: 5.12s
```

```
349:    learn: 12833.6922322    total: 2.75s    remaining: 5.11s
350:    learn: 12794.3307117    total: 2.76s    remaining: 5.1s
351:    learn: 12778.0016559    total: 2.77s    remaining: 5.1s
352:    learn: 12753.8399654    total: 2.78s    remaining: 5.09s
353:    learn: 12732.0495678    total: 2.79s    remaining: 5.09s
354:    learn: 12711.5330219    total: 2.8s     remaining: 5.08s
355:    learn: 12677.6212552    total: 2.81s    remaining: 5.07s
356:    learn: 12659.6312954    total: 2.81s    remaining: 5.07s
357:    learn: 12651.4302901    total: 2.82s    remaining: 5.05s
358:    learn: 12632.1560333    total: 2.83s    remaining: 5.05s
359:    learn: 12626.1594674    total: 2.83s    remaining: 5.04s
360:    learn: 12601.8064578    total: 2.84s    remaining: 5.03s
361:    learn: 12578.9934645    total: 2.85s    remaining: 5.02s
362:    learn: 12560.0122904    total: 2.85s    remaining: 5.01s
363:    learn: 12539.1347835    total: 2.86s    remaining: 5s
364:    learn: 12525.0587063    total: 2.87s    remaining: 4.99s
365:    learn: 12520.2500107    total: 2.87s    remaining: 4.98s
366:    learn: 12502.9657227    total: 2.88s    remaining: 4.97s
367:    learn: 12473.3359145    total: 2.89s    remaining: 4.96s
368:    learn: 12458.1440351    total: 2.89s    remaining: 4.95s
369:    learn: 12435.9913540    total: 2.9s     remaining: 4.94s
370:    learn: 12425.3262583    total: 2.91s    remaining: 4.93s
371:    learn: 12395.8718986    total: 2.91s    remaining: 4.92s
372:    learn: 12370.1663771    total: 2.92s    remaining: 4.91s
373:    learn: 12355.7721122    total: 2.93s    remaining: 4.9s
374:    learn: 12332.8289595    total: 2.94s    remaining: 4.89s
375:    learn: 12314.1884192    total: 2.94s    remaining: 4.88s
376:    learn: 12292.3758779    total: 2.95s    remaining: 4.88s
377:    learn: 12245.3353624    total: 2.96s    remaining: 4.87s
378:    learn: 12224.8821955    total: 2.96s    remaining: 4.86s
379:    learn: 12208.7178313    total: 2.97s    remaining: 4.85s
380:    learn: 12187.9826766    total: 2.98s    remaining: 4.84s
381:    learn: 12160.4522329    total: 2.99s    remaining: 4.83s
382:    learn: 12150.6614488    total: 2.99s    remaining: 4.82s
383:    learn: 12137.8761855    total: 3s       remaining: 4.81s
384:    learn: 12103.3062280    total: 3.01s    remaining: 4.8s
385:    learn: 12081.0508279    total: 3.01s    remaining: 4.79s
386:    learn: 12068.2555980    total: 3.02s    remaining: 4.79s
387:    learn: 12054.1195847    total: 3.04s    remaining: 4.79s
388:    learn: 12044.6853409    total: 3.04s    remaining: 4.78s
389:    learn: 12035.9618485    total: 3.05s    remaining: 4.78s
390:    learn: 12025.2906132    total: 3.06s    remaining: 4.77s
391:    learn: 12008.6589606    total: 3.07s    remaining: 4.76s
392:    learn: 11989.6339093    total: 3.08s    remaining: 4.75s
393:    learn: 11963.6775880    total: 3.08s    remaining: 4.74s
394:    learn: 11928.1756653    total: 3.09s    remaining: 4.73s
395:    learn: 11917.1690141    total: 3.1s     remaining: 4.72s
396:    learn: 11890.1373366    total: 3.1s     remaining: 4.71s
397:    learn: 11866.4351563    total: 3.11s    remaining: 4.7s
398:    learn: 11855.4390622    total: 3.12s    remaining: 4.7s
```

```
399:	learn: 11826.9666909	total: 3.12s	remaining: 4.68s
400:	learn: 11797.7053987	total: 3.13s	remaining: 4.67s
401:	learn: 11782.8575421	total: 3.14s	remaining: 4.67s
402:	learn: 11762.6841232	total: 3.14s	remaining: 4.66s
403:	learn: 11752.5688454	total: 3.15s	remaining: 4.65s
404:	learn: 11724.9305873	total: 3.16s	remaining: 4.64s
405:	learn: 11700.5326931	total: 3.17s	remaining: 4.63s
406:	learn: 11683.0661220	total: 3.17s	remaining: 4.63s
407:	learn: 11666.3488547	total: 3.18s	remaining: 4.62s
408:	learn: 11653.3542324	total: 3.19s	remaining: 4.61s
409:	learn: 11631.8996391	total: 3.2s	remaining: 4.6s
410:	learn: 11621.1621306	total: 3.2s	remaining: 4.59s
411:	learn: 11600.5991605	total: 3.21s	remaining: 4.58s
412:	learn: 11573.7967686	total: 3.22s	remaining: 4.57s
413:	learn: 11561.1673052	total: 3.22s	remaining: 4.56s
414:	learn: 11549.4713028	total: 3.23s	remaining: 4.55s
415:	learn: 11530.3182315	total: 3.24s	remaining: 4.54s
416:	learn: 11511.6022680	total: 3.25s	remaining: 4.54s
417:	learn: 11506.8895337	total: 3.25s	remaining: 4.53s
418:	learn: 11492.5926981	total: 3.26s	remaining: 4.52s
419:	learn: 11473.7046260	total: 3.27s	remaining: 4.52s
420:	learn: 11454.8721879	total: 3.28s	remaining: 4.51s
421:	learn: 11436.0237534	total: 3.29s	remaining: 4.5s
422:	learn: 11419.2839970	total: 3.3s	remaining: 4.5s
423:	learn: 11400.6401503	total: 3.31s	remaining: 4.49s
424:	learn: 11397.6573487	total: 3.32s	remaining: 4.49s
425:	learn: 11380.4756026	total: 3.33s	remaining: 4.48s
426:	learn: 11371.8449431	total: 3.33s	remaining: 4.47s
427:	learn: 11349.4849203	total: 3.34s	remaining: 4.47s
428:	learn: 11331.7612774	total: 3.35s	remaining: 4.46s
429:	learn: 11308.3811262	total: 3.36s	remaining: 4.45s
430:	learn: 11286.4409540	total: 3.36s	remaining: 4.44s
431:	learn: 11269.8024609	total: 3.37s	remaining: 4.43s
432:	learn: 11253.2266388	total: 3.38s	remaining: 4.42s
433:	learn: 11239.0417330	total: 3.38s	remaining: 4.41s
434:	learn: 11217.6646215	total: 3.39s	remaining: 4.4s
435:	learn: 11201.3200591	total: 3.4s	remaining: 4.39s
436:	learn: 11195.4302613	total: 3.4s	remaining: 4.38s
437:	learn: 11162.4266239	total: 3.41s	remaining: 4.38s
438:	learn: 11144.0440759	total: 3.42s	remaining: 4.37s
439:	learn: 11126.3175061	total: 3.43s	remaining: 4.36s
440:	learn: 11097.1362948	total: 3.44s	remaining: 4.35s
441:	learn: 11080.5258752	total: 3.44s	remaining: 4.34s
442:	learn: 11057.4796846	total: 3.45s	remaining: 4.34s
443:	learn: 11036.9699382	total: 3.46s	remaining: 4.33s
444:	learn: 11020.1514951	total: 3.46s	remaining: 4.32s
445:	learn: 10989.4129749	total: 3.47s	remaining: 4.31s
446:	learn: 10969.1760249	total: 3.48s	remaining: 4.3s
447:	learn: 10956.8924385	total: 3.49s	remaining: 4.3s
448:	learn: 10946.6961123	total: 3.49s	remaining: 4.29s
```

```
449:	learn: 10941.6317788	total: 3.5s	remaining: 4.28s
450:	learn: 10925.3875261	total: 3.51s	remaining: 4.27s
451:	learn: 10918.7350482	total: 3.52s	remaining: 4.26s
452:	learn: 10899.4873293	total: 3.52s	remaining: 4.25s
453:	learn: 10887.2665504	total: 3.53s	remaining: 4.25s
454:	learn: 10864.4672235	total: 3.54s	remaining: 4.24s
455:	learn: 10843.5928618	total: 3.55s	remaining: 4.23s
456:	learn: 10809.1184525	total: 3.55s	remaining: 4.22s
457:	learn: 10794.6528823	total: 3.56s	remaining: 4.21s
458:	learn: 10776.4685054	total: 3.57s	remaining: 4.21s
459:	learn: 10755.0654898	total: 3.57s	remaining: 4.2s
460:	learn: 10735.3030200	total: 3.58s	remaining: 4.19s
461:	learn: 10731.7144198	total: 3.59s	remaining: 4.18s
462:	learn: 10702.3021476	total: 3.6s	remaining: 4.17s
463:	learn: 10681.3133491	total: 3.61s	remaining: 4.17s
464:	learn: 10666.1953544	total: 3.62s	remaining: 4.16s
465:	learn: 10663.4435994	total: 3.62s	remaining: 4.15s
466:	learn: 10635.9950604	total: 3.63s	remaining: 4.14s
467:	learn: 10627.6919890	total: 3.64s	remaining: 4.13s
468:	learn: 10608.8976223	total: 3.65s	remaining: 4.13s
469:	learn: 10596.7831101	total: 3.65s	remaining: 4.12s
470:	learn: 10578.2675591	total: 3.66s	remaining: 4.11s
471:	learn: 10571.3179304	total: 3.67s	remaining: 4.11s
472:	learn: 10558.8318854	total: 3.68s	remaining: 4.1s
473:	learn: 10555.2945741	total: 3.69s	remaining: 4.09s
474:	learn: 10529.5763308	total: 3.7s	remaining: 4.09s
475:	learn: 10519.9959721	total: 3.7s	remaining: 4.08s
476:	learn: 10508.2876112	total: 3.71s	remaining: 4.07s
477:	learn: 10505.5130270	total: 3.71s	remaining: 4.06s
478:	learn: 10497.5690383	total: 3.72s	remaining: 4.05s
479:	learn: 10483.5265383	total: 3.73s	remaining: 4.04s
480:	learn: 10469.1037650	total: 3.74s	remaining: 4.03s
481:	learn: 10445.8583586	total: 3.75s	remaining: 4.03s
482:	learn: 10435.4718807	total: 3.75s	remaining: 4.02s
483:	learn: 10419.4721540	total: 3.76s	remaining: 4.01s
484:	learn: 10410.9827492	total: 3.77s	remaining: 4s
485:	learn: 10400.6926211	total: 3.78s	remaining: 3.99s
486:	learn: 10380.7183991	total: 3.78s	remaining: 3.99s
487:	learn: 10361.2751016	total: 3.79s	remaining: 3.98s
488:	learn: 10347.2657576	total: 3.8s	remaining: 3.97s
489:	learn: 10331.9790208	total: 3.81s	remaining: 3.96s
490:	learn: 10309.9007502	total: 3.81s	remaining: 3.95s
491:	learn: 10288.9463071	total: 3.82s	remaining: 3.94s
492:	learn: 10269.3912699	total: 3.83s	remaining: 3.94s
493:	learn: 10250.1054914	total: 3.84s	remaining: 3.93s
494:	learn: 10223.6217185	total: 3.85s	remaining: 3.92s
495:	learn: 10201.8216806	total: 3.85s	remaining: 3.92s
496:	learn: 10199.5899564	total: 3.86s	remaining: 3.91s
497:	learn: 10174.8855136	total: 3.87s	remaining: 3.9s
498:	learn: 10145.4533720	total: 3.88s	remaining: 3.89s
```

```
499:	learn: 10124.9904039	total: 3.88s	remaining: 3.88s
500:	learn: 10103.6233036	total: 3.89s	remaining: 3.88s
501:	learn: 10091.3249432	total: 3.9s	remaining: 3.87s
502:	learn: 10060.8060127	total: 3.9s	remaining: 3.86s
503:	learn: 10046.2434886	total: 3.91s	remaining: 3.85s
504:	learn: 10040.1507612	total: 3.92s	remaining: 3.84s
505:	learn: 10020.8430313	total: 3.93s	remaining: 3.83s
506:	learn: 9995.2791118	total: 3.93s	remaining: 3.82s
507:	learn: 9978.4784703	total: 3.94s	remaining: 3.81s
508:	learn: 9954.4625553	total: 3.94s	remaining: 3.81s
509:	learn: 9936.3587160	total: 3.95s	remaining: 3.8s
510:	learn: 9934.2974710	total: 3.96s	remaining: 3.79s
511:	learn: 9926.3733469	total: 3.96s	remaining: 3.78s
512:	learn: 9922.4126031	total: 3.97s	remaining: 3.77s
513:	learn: 9907.6080612	total: 3.98s	remaining: 3.76s
514:	learn: 9902.0335576	total: 3.99s	remaining: 3.75s
515:	learn: 9881.8682999	total: 3.99s	remaining: 3.75s
516:	learn: 9869.9186745	total: 4s	remaining: 3.74s
517:	learn: 9849.8660557	total: 4.01s	remaining: 3.73s
518:	learn: 9834.9313116	total: 4.02s	remaining: 3.72s
519:	learn: 9823.2790616	total: 4.03s	remaining: 3.72s
520:	learn: 9806.8391500	total: 4.03s	remaining: 3.71s
521:	learn: 9796.0265272	total: 4.04s	remaining: 3.7s
522:	learn: 9793.9117994	total: 4.04s	remaining: 3.69s
523:	learn: 9778.8448038	total: 4.05s	remaining: 3.68s
524:	learn: 9770.3576455	total: 4.06s	remaining: 3.67s
525:	learn: 9755.9110060	total: 4.06s	remaining: 3.66s
526:	learn: 9749.0113486	total: 4.07s	remaining: 3.65s
527:	learn: 9730.6287736	total: 4.08s	remaining: 3.64s
528:	learn: 9720.3783538	total: 4.08s	remaining: 3.63s
529:	learn: 9703.5293442	total: 4.09s	remaining: 3.63s
530:	learn: 9687.7533611	total: 4.09s	remaining: 3.62s
531:	learn: 9672.6071745	total: 4.1s	remaining: 3.61s
532:	learn: 9661.9671357	total: 4.11s	remaining: 3.6s
533:	learn: 9659.0637955	total: 4.11s	remaining: 3.59s
534:	learn: 9650.7683026	total: 4.12s	remaining: 3.58s
535:	learn: 9637.3197606	total: 4.13s	remaining: 3.57s
536:	learn: 9624.6104746	total: 4.13s	remaining: 3.56s
537:	learn: 9612.8919457	total: 4.14s	remaining: 3.56s
538:	learn: 9603.4847548	total: 4.15s	remaining: 3.55s
539:	learn: 9595.4196266	total: 4.16s	remaining: 3.54s
540:	learn: 9583.2923198	total: 4.16s	remaining: 3.53s
541:	learn: 9565.0052115	total: 4.17s	remaining: 3.52s
542:	learn: 9556.3898776	total: 4.17s	remaining: 3.51s
543:	learn: 9544.1081456	total: 4.18s	remaining: 3.5s
544:	learn: 9533.7123686	total: 4.19s	remaining: 3.5s
545:	learn: 9517.1956515	total: 4.2s	remaining: 3.49s
546:	learn: 9501.3143060	total: 4.2s	remaining: 3.48s
547:	learn: 9487.4345471	total: 4.21s	remaining: 3.47s
548:	learn: 9477.9329476	total: 4.22s	remaining: 3.46s
```

```
549:	learn: 9475.8545718	total: 4.22s	remaining: 3.46s
550:	learn: 9459.8636724	total: 4.23s	remaining: 3.45s
551:	learn: 9458.0977140	total: 4.24s	remaining: 3.44s
552:	learn: 9453.8295793	total: 4.25s	remaining: 3.44s
553:	learn: 9441.1293649	total: 4.26s	remaining: 3.43s
554:	learn: 9416.8763223	total: 4.27s	remaining: 3.42s
555:	learn: 9405.8367216	total: 4.28s	remaining: 3.42s
556:	learn: 9392.4348290	total: 4.28s	remaining: 3.41s
557:	learn: 9377.2464789	total: 4.29s	remaining: 3.4s
558:	learn: 9358.4542720	total: 4.3s	remaining: 3.39s
559:	learn: 9337.8128215	total: 4.3s	remaining: 3.38s
560:	learn: 9332.4640782	total: 4.31s	remaining: 3.37s
561:	learn: 9324.5589066	total: 4.32s	remaining: 3.37s
562:	learn: 9312.7508402	total: 4.33s	remaining: 3.36s
563:	learn: 9295.1326833	total: 4.33s	remaining: 3.35s
564:	learn: 9282.6838735	total: 4.34s	remaining: 3.34s
565:	learn: 9269.8727251	total: 4.35s	remaining: 3.33s
566:	learn: 9256.6859966	total: 4.35s	remaining: 3.32s
567:	learn: 9237.3421333	total: 4.36s	remaining: 3.32s
568:	learn: 9235.7589806	total: 4.37s	remaining: 3.31s
569:	learn: 9218.0692777	total: 4.38s	remaining: 3.3s
570:	learn: 9197.2324834	total: 4.38s	remaining: 3.29s
571:	learn: 9187.4170253	total: 4.39s	remaining: 3.29s
572:	learn: 9179.9835132	total: 4.4s	remaining: 3.28s
573:	learn: 9161.4215670	total: 4.4s	remaining: 3.27s
574:	learn: 9134.7819212	total: 4.41s	remaining: 3.26s
575:	learn: 9124.7400355	total: 4.42s	remaining: 3.25s
576:	learn: 9111.2805506	total: 4.43s	remaining: 3.25s
577:	learn: 9103.8837209	total: 4.43s	remaining: 3.24s
578:	learn: 9094.6874648	total: 4.44s	remaining: 3.23s
579:	learn: 9078.6760677	total: 4.45s	remaining: 3.22s
580:	learn: 9072.3600128	total: 4.46s	remaining: 3.21s
581:	learn: 9056.7415099	total: 4.46s	remaining: 3.21s
582:	learn: 9054.7403959	total: 4.47s	remaining: 3.2s
583:	learn: 9045.8160669	total: 4.48s	remaining: 3.19s
584:	learn: 9032.4848520	total: 4.49s	remaining: 3.18s
585:	learn: 9017.9927878	total: 4.5s	remaining: 3.18s
586:	learn: 9013.8306426	total: 4.5s	remaining: 3.17s
587:	learn: 9000.0213245	total: 4.51s	remaining: 3.16s
588:	learn: 8988.4492738	total: 4.52s	remaining: 3.15s
589:	learn: 8987.0620248	total: 4.53s	remaining: 3.15s
590:	learn: 8972.0873510	total: 4.53s	remaining: 3.14s
591:	learn: 8956.9477577	total: 4.54s	remaining: 3.13s
592:	learn: 8951.7896977	total: 4.54s	remaining: 3.12s
593:	learn: 8950.2748671	total: 4.55s	remaining: 3.11s
594:	learn: 8927.7524721	total: 4.56s	remaining: 3.1s
595:	learn: 8926.3154902	total: 4.57s	remaining: 3.1s
596:	learn: 8909.8577706	total: 4.57s	remaining: 3.09s
597:	learn: 8890.5268844	total: 4.58s	remaining: 3.08s
598:	learn: 8886.0395709	total: 4.59s	remaining: 3.07s
```

```
599:	learn: 8873.1840606	total: 4.6s	remaining: 3.06s
600:	learn: 8867.1597875	total: 4.6s	remaining: 3.06s
601:	learn: 8855.3491773	total: 4.61s	remaining: 3.05s
602:	learn: 8844.4627468	total: 4.62s	remaining: 3.04s
603:	learn: 8824.6777017	total: 4.62s	remaining: 3.03s
604:	learn: 8803.9255017	total: 4.63s	remaining: 3.02s
605:	learn: 8796.6432271	total: 4.64s	remaining: 3.02s
606:	learn: 8782.4508450	total: 4.64s	remaining: 3.01s
607:	learn: 8760.9433992	total: 4.65s	remaining: 3s
608:	learn: 8751.2953737	total: 4.66s	remaining: 2.99s
609:	learn: 8741.1485284	total: 4.67s	remaining: 2.98s
610:	learn: 8731.7913815	total: 4.67s	remaining: 2.98s
611:	learn: 8730.5142655	total: 4.68s	remaining: 2.97s
612:	learn: 8719.5455268	total: 4.69s	remaining: 2.96s
613:	learn: 8703.7842872	total: 4.7s	remaining: 2.95s
614:	learn: 8702.5483188	total: 4.7s	remaining: 2.94s
615:	learn: 8690.3132492	total: 4.71s	remaining: 2.94s
616:	learn: 8682.6716728	total: 4.72s	remaining: 2.93s
617:	learn: 8662.7518480	total: 4.73s	remaining: 2.92s
618:	learn: 8661.5445789	total: 4.74s	remaining: 2.92s
619:	learn: 8648.4235916	total: 4.75s	remaining: 2.91s
620:	learn: 8647.1234256	total: 4.76s	remaining: 2.9s
621:	learn: 8645.2627429	total: 4.76s	remaining: 2.9s
622:	learn: 8634.8389132	total: 4.77s	remaining: 2.89s
623:	learn: 8618.5754515	total: 4.78s	remaining: 2.88s
624:	learn: 8604.2851438	total: 4.79s	remaining: 2.88s
625:	learn: 8593.0235579	total: 4.81s	remaining: 2.87s
626:	learn: 8576.4566123	total: 4.82s	remaining: 2.87s
627:	learn: 8565.8934861	total: 4.83s	remaining: 2.86s
628:	learn: 8545.8620917	total: 4.83s	remaining: 2.85s
629:	learn: 8544.0982156	total: 4.84s	remaining: 2.84s
630:	learn: 8533.5271967	total: 4.85s	remaining: 2.83s
631:	learn: 8532.4353388	total: 4.85s	remaining: 2.83s
632:	learn: 8523.2087711	total: 4.86s	remaining: 2.82s
633:	learn: 8505.7384374	total: 4.87s	remaining: 2.81s
634:	learn: 8500.6171903	total: 4.87s	remaining: 2.8s
635:	learn: 8491.4890070	total: 4.88s	remaining: 2.79s
636:	learn: 8482.2149735	total: 4.89s	remaining: 2.78s
637:	learn: 8478.8402339	total: 4.89s	remaining: 2.78s
638:	learn: 8462.0000684	total: 4.9s	remaining: 2.77s
639:	learn: 8446.2194755	total: 4.91s	remaining: 2.76s
640:	learn: 8429.2060988	total: 4.92s	remaining: 2.75s
641:	learn: 8423.1726930	total: 4.92s	remaining: 2.75s
642:	learn: 8412.4171504	total: 4.93s	remaining: 2.74s
643:	learn: 8397.3097794	total: 4.94s	remaining: 2.73s
644:	learn: 8376.6100054	total: 4.95s	remaining: 2.72s
645:	learn: 8359.6866601	total: 4.95s	remaining: 2.71s
646:	learn: 8348.2342407	total: 4.96s	remaining: 2.71s
647:	learn: 8344.6691261	total: 4.97s	remaining: 2.7s
648:	learn: 8333.4129143	total: 4.97s	remaining: 2.69s
```

```
649:	learn: 8322.6325361	total: 4.98s	remaining: 2.68s
650:	learn: 8313.5568889	total: 4.99s	remaining: 2.67s
651:	learn: 8305.2683683	total: 5s	remaining: 2.67s
652:	learn: 8289.4860020	total: 5.01s	remaining: 2.66s
653:	learn: 8278.2429840	total: 5.02s	remaining: 2.65s
654:	learn: 8277.0211759	total: 5.02s	remaining: 2.65s
655:	learn: 8268.5910312	total: 5.03s	remaining: 2.64s
656:	learn: 8257.8473171	total: 5.04s	remaining: 2.63s
657:	learn: 8248.3149562	total: 5.04s	remaining: 2.62s
658:	learn: 8245.3113193	total: 5.05s	remaining: 2.61s
659:	learn: 8226.8170085	total: 5.06s	remaining: 2.61s
660:	learn: 8216.3537697	total: 5.07s	remaining: 2.6s
661:	learn: 8207.4241900	total: 5.07s	remaining: 2.59s
662:	learn: 8199.8029452	total: 5.08s	remaining: 2.58s
663:	learn: 8181.7777482	total: 5.08s	remaining: 2.57s
664:	learn: 8170.0525174	total: 5.09s	remaining: 2.56s
665:	learn: 8168.8649420	total: 5.1s	remaining: 2.56s
666:	learn: 8156.5691581	total: 5.11s	remaining: 2.55s
667:	learn: 8151.6825203	total: 5.11s	remaining: 2.54s
668:	learn: 8131.8922007	total: 5.12s	remaining: 2.53s
669:	learn: 8122.5348408	total: 5.13s	remaining: 2.52s
670:	learn: 8109.9358187	total: 5.13s	remaining: 2.52s
671:	learn: 8100.3609595	total: 5.14s	remaining: 2.51s
672:	learn: 8089.3430771	total: 5.15s	remaining: 2.5s
673:	learn: 8078.0757555	total: 5.15s	remaining: 2.49s
674:	learn: 8077.1471250	total: 5.16s	remaining: 2.48s
675:	learn: 8069.7028607	total: 5.17s	remaining: 2.48s
676:	learn: 8058.6108511	total: 5.17s	remaining: 2.47s
677:	learn: 8052.0651343	total: 5.18s	remaining: 2.46s
678:	learn: 8041.7784601	total: 5.19s	remaining: 2.46s
679:	learn: 8028.7132145	total: 5.2s	remaining: 2.45s
680:	learn: 8018.0082803	total: 5.21s	remaining: 2.44s
681:	learn: 8006.5612369	total: 5.22s	remaining: 2.44s
682:	learn: 7993.7991152	total: 5.24s	remaining: 2.43s
683:	learn: 7990.9421098	total: 5.25s	remaining: 2.42s
684:	learn: 7979.8956235	total: 5.26s	remaining: 2.42s
685:	learn: 7971.4233987	total: 5.27s	remaining: 2.41s
686:	learn: 7965.0967840	total: 5.28s	remaining: 2.4s
687:	learn: 7957.9029457	total: 5.29s	remaining: 2.4s
688:	learn: 7945.1748624	total: 5.3s	remaining: 2.39s
689:	learn: 7933.7203939	total: 5.31s	remaining: 2.38s
690:	learn: 7927.3380798	total: 5.31s	remaining: 2.38s
691:	learn: 7926.4988747	total: 5.32s	remaining: 2.37s
692:	learn: 7910.3401716	total: 5.33s	remaining: 2.36s
693:	learn: 7891.7172860	total: 5.34s	remaining: 2.35s
694:	learn: 7874.7666004	total: 5.35s	remaining: 2.35s
695:	learn: 7864.6765535	total: 5.36s	remaining: 2.34s
696:	learn: 7856.4935837	total: 5.37s	remaining: 2.33s
697:	learn: 7845.6933830	total: 5.38s	remaining: 2.33s
698:	learn: 7836.7220385	total: 5.39s	remaining: 2.32s
```

```
699:	learn: 7824.7166364	total: 5.4s	remaining: 2.31s
700:	learn: 7823.3367325	total: 5.4s	remaining: 2.3s
701:	learn: 7810.1933146	total: 5.41s	remaining: 2.3s
702:	learn: 7801.6033728	total: 5.42s	remaining: 2.29s
703:	learn: 7787.8071318	total: 5.43s	remaining: 2.28s
704:	learn: 7779.4765796	total: 5.44s	remaining: 2.28s
705:	learn: 7769.7097641	total: 5.45s	remaining: 2.27s
706:	learn: 7759.2749897	total: 5.46s	remaining: 2.26s
707:	learn: 7752.2547079	total: 5.47s	remaining: 2.25s
708:	learn: 7750.9824590	total: 5.47s	remaining: 2.25s
709:	learn: 7745.8477592	total: 5.48s	remaining: 2.24s
710:	learn: 7734.4722760	total: 5.49s	remaining: 2.23s
711:	learn: 7733.5915887	total: 5.5s	remaining: 2.23s
712:	learn: 7713.5413456	total: 5.51s	remaining: 2.22s
713:	learn: 7712.4765071	total: 5.52s	remaining: 2.21s
714:	learn: 7701.0707752	total: 5.53s	remaining: 2.2s
715:	learn: 7692.2514536	total: 5.54s	remaining: 2.2s
716:	learn: 7679.5468627	total: 5.55s	remaining: 2.19s
717:	learn: 7672.2853098	total: 5.55s	remaining: 2.18s
718:	learn: 7663.3945702	total: 5.56s	remaining: 2.17s
719:	learn: 7659.9156956	total: 5.57s	remaining: 2.17s
720:	learn: 7643.6130074	total: 5.58s	remaining: 2.16s
721:	learn: 7628.2990798	total: 5.59s	remaining: 2.15s
722:	learn: 7612.5649791	total: 5.59s	remaining: 2.14s
723:	learn: 7601.8190241	total: 5.6s	remaining: 2.14s
724:	learn: 7585.5050186	total: 5.61s	remaining: 2.13s
725:	learn: 7573.9801758	total: 5.62s	remaining: 2.12s
726:	learn: 7572.9588293	total: 5.63s	remaining: 2.11s
727:	learn: 7559.3143306	total: 5.64s	remaining: 2.1s
728:	learn: 7549.5571270	total: 5.65s	remaining: 2.1s
729:	learn: 7538.6975833	total: 5.65s	remaining: 2.09s
730:	learn: 7529.8553423	total: 5.66s	remaining: 2.08s
731:	learn: 7523.8245555	total: 5.67s	remaining: 2.08s
732:	learn: 7512.0473000	total: 5.68s	remaining: 2.07s
733:	learn: 7505.6209953	total: 5.69s	remaining: 2.06s
734:	learn: 7496.7524937	total: 5.7s	remaining: 2.05s
735:	learn: 7486.6687671	total: 5.7s	remaining: 2.05s
736:	learn: 7478.1160450	total: 5.71s	remaining: 2.04s
737:	learn: 7463.9231550	total: 5.72s	remaining: 2.03s
738:	learn: 7463.0570436	total: 5.73s	remaining: 2.02s
739:	learn: 7451.5397655	total: 5.74s	remaining: 2.02s
740:	learn: 7443.2021972	total: 5.75s	remaining: 2.01s
741:	learn: 7435.0824603	total: 5.76s	remaining: 2s
742:	learn: 7434.2171586	total: 5.76s	remaining: 1.99s
743:	learn: 7427.8441364	total: 5.77s	remaining: 1.99s
744:	learn: 7415.5378033	total: 5.78s	remaining: 1.98s
745:	learn: 7402.2866116	total: 5.79s	remaining: 1.97s
746:	learn: 7389.8464313	total: 5.8s	remaining: 1.96s
747:	learn: 7382.4003309	total: 5.8s	remaining: 1.96s
748:	learn: 7381.4079400	total: 5.81s	remaining: 1.95s
```

```
749:	learn: 7376.4308037	total: 5.82s	remaining: 1.94s
750:	learn: 7361.3979130	total: 5.83s	remaining: 1.93s
751:	learn: 7350.0844809	total: 5.84s	remaining: 1.92s
752:	learn: 7340.0829082	total: 5.84s	remaining: 1.92s
753:	learn: 7331.8914507	total: 5.85s	remaining: 1.91s
754:	learn: 7319.3550900	total: 5.86s	remaining: 1.9s
755:	learn: 7311.8457997	total: 5.86s	remaining: 1.89s
756:	learn: 7305.6479913	total: 5.87s	remaining: 1.88s
757:	learn: 7299.9811981	total: 5.88s	remaining: 1.88s
758:	learn: 7295.0459601	total: 5.89s	remaining: 1.87s
759:	learn: 7291.3868169	total: 5.89s	remaining: 1.86s
760:	learn: 7277.0957886	total: 5.9s	remaining: 1.85s
761:	learn: 7269.4603151	total: 5.91s	remaining: 1.84s
762:	learn: 7251.3854929	total: 5.91s	remaining: 1.84s
763:	learn: 7237.5714793	total: 5.92s	remaining: 1.83s
764:	learn: 7224.4101093	total: 5.93s	remaining: 1.82s
765:	learn: 7223.7354944	total: 5.94s	remaining: 1.81s
766:	learn: 7213.2289256	total: 5.95s	remaining: 1.81s
767:	learn: 7203.7142359	total: 5.96s	remaining: 1.8s
768:	learn: 7195.0027989	total: 5.96s	remaining: 1.79s
769:	learn: 7186.0334485	total: 5.97s	remaining: 1.78s
770:	learn: 7178.7446938	total: 5.97s	remaining: 1.77s
771:	learn: 7170.2276236	total: 5.98s	remaining: 1.77s
772:	learn: 7164.3588565	total: 5.99s	remaining: 1.76s
773:	learn: 7149.7800314	total: 6s	remaining: 1.75s
774:	learn: 7140.5973811	total: 6.01s	remaining: 1.74s
775:	learn: 7129.7948412	total: 6.01s	remaining: 1.74s
776:	learn: 7119.7759428	total: 6.02s	remaining: 1.73s
777:	learn: 7111.0138610	total: 6.03s	remaining: 1.72s
778:	learn: 7097.3959326	total: 6.04s	remaining: 1.71s
779:	learn: 7085.2307000	total: 6.04s	remaining: 1.71s
780:	learn: 7084.4417548	total: 6.05s	remaining: 1.7s
781:	learn: 7083.7018054	total: 6.06s	remaining: 1.69s
782:	learn: 7078.1996917	total: 6.07s	remaining: 1.68s
783:	learn: 7067.7515080	total: 6.08s	remaining: 1.67s
784:	learn: 7055.7953825	total: 6.08s	remaining: 1.67s
785:	learn: 7046.9030771	total: 6.09s	remaining: 1.66s
786:	learn: 7041.6700427	total: 6.1s	remaining: 1.65s
787:	learn: 7032.2676982	total: 6.11s	remaining: 1.64s
788:	learn: 7031.6194846	total: 6.11s	remaining: 1.63s
789:	learn: 7025.4044804	total: 6.12s	remaining: 1.63s
790:	learn: 7015.1860343	total: 6.13s	remaining: 1.62s
791:	learn: 7005.4245563	total: 6.13s	remaining: 1.61s
792:	learn: 6992.7762886	total: 6.14s	remaining: 1.6s
793:	learn: 6983.3844955	total: 6.15s	remaining: 1.59s
794:	learn: 6978.4694703	total: 6.16s	remaining: 1.59s
795:	learn: 6963.8740974	total: 6.16s	remaining: 1.58s
796:	learn: 6954.4179164	total: 6.17s	remaining: 1.57s
797:	learn: 6944.9589307	total: 6.18s	remaining: 1.56s
798:	learn: 6944.2858923	total: 6.18s	remaining: 1.55s
```

```
799:	learn: 6943.6332766	total: 6.19s	remaining: 1.55s
800:	learn: 6936.2691531	total: 6.2s	remaining: 1.54s
801:	learn: 6930.7293010	total: 6.2s	remaining: 1.53s
802:	learn: 6922.9452142	total: 6.21s	remaining: 1.52s
803:	learn: 6915.0754160	total: 6.22s	remaining: 1.52s
804:	learn: 6900.6538435	total: 6.23s	remaining: 1.51s
805:	learn: 6890.9893121	total: 6.24s	remaining: 1.5s
806:	learn: 6883.2675033	total: 6.24s	remaining: 1.49s
807:	learn: 6875.3539642	total: 6.25s	remaining: 1.49s
808:	learn: 6874.8113418	total: 6.26s	remaining: 1.48s
809:	learn: 6868.4733730	total: 6.27s	remaining: 1.47s
810:	learn: 6867.8540397	total: 6.28s	remaining: 1.46s
811:	learn: 6860.0930225	total: 6.29s	remaining: 1.46s
812:	learn: 6848.4692473	total: 6.3s	remaining: 1.45s
813:	learn: 6834.7486489	total: 6.31s	remaining: 1.44s
814:	learn: 6817.9845396	total: 6.32s	remaining: 1.43s
815:	learn: 6812.2580269	total: 6.33s	remaining: 1.43s
816:	learn: 6804.3971340	total: 6.34s	remaining: 1.42s
817:	learn: 6798.4369285	total: 6.34s	remaining: 1.41s
818:	learn: 6788.4881478	total: 6.35s	remaining: 1.4s
819:	learn: 6777.5171471	total: 6.36s	remaining: 1.4s
820:	learn: 6776.8140179	total: 6.37s	remaining: 1.39s
821:	learn: 6770.0764769	total: 6.38s	remaining: 1.38s
822:	learn: 6766.0742370	total: 6.39s	remaining: 1.37s
823:	learn: 6752.0307922	total: 6.4s	remaining: 1.37s
824:	learn: 6733.6553372	total: 6.41s	remaining: 1.36s
825:	learn: 6719.8443249	total: 6.42s	remaining: 1.35s
826:	learn: 6716.0787815	total: 6.42s	remaining: 1.34s
827:	learn: 6706.9733693	total: 6.43s	remaining: 1.34s
828:	learn: 6695.5721659	total: 6.44s	remaining: 1.33s
829:	learn: 6688.3172075	total: 6.45s	remaining: 1.32s
830:	learn: 6681.9658681	total: 6.46s	remaining: 1.31s
831:	learn: 6676.0736350	total: 6.46s	remaining: 1.3s
832:	learn: 6661.9297331	total: 6.47s	remaining: 1.3s
833:	learn: 6656.1990735	total: 6.48s	remaining: 1.29s
834:	learn: 6641.1548769	total: 6.49s	remaining: 1.28s
835:	learn: 6624.2800063	total: 6.5s	remaining: 1.27s
836:	learn: 6618.6037187	total: 6.5s	remaining: 1.27s
837:	learn: 6610.0237786	total: 6.51s	remaining: 1.26s
838:	learn: 6600.4454433	total: 6.52s	remaining: 1.25s
839:	learn: 6591.1353308	total: 6.53s	remaining: 1.24s
840:	learn: 6584.3056952	total: 6.53s	remaining: 1.24s
841:	learn: 6579.2934688	total: 6.54s	remaining: 1.23s
842:	learn: 6567.4679659	total: 6.55s	remaining: 1.22s
843:	learn: 6555.6084893	total: 6.56s	remaining: 1.21s
844:	learn: 6548.8012928	total: 6.56s	remaining: 1.2s
845:	learn: 6537.3229487	total: 6.57s	remaining: 1.2s
846:	learn: 6527.6642672	total: 6.58s	remaining: 1.19s
847:	learn: 6527.1544576	total: 6.59s	remaining: 1.18s
848:	learn: 6519.1418264	total: 6.59s	remaining: 1.17s
```

```
849:	learn: 6513.9268299	total: 6.6s	remaining: 1.17s
850:	learn: 6513.2666073	total: 6.61s	remaining: 1.16s
851:	learn: 6502.1521248	total: 6.62s	remaining: 1.15s
852:	learn: 6501.6495395	total: 6.63s	remaining: 1.14s
853:	learn: 6498.9323451	total: 6.63s	remaining: 1.13s
854:	learn: 6483.9524029	total: 6.64s	remaining: 1.13s
855:	learn: 6477.4952827	total: 6.65s	remaining: 1.12s
856:	learn: 6473.3733976	total: 6.66s	remaining: 1.11s
857:	learn: 6461.6075382	total: 6.67s	remaining: 1.1s
858:	learn: 6454.5875861	total: 6.67s	remaining: 1.09s
859:	learn: 6447.1507878	total: 6.68s	remaining: 1.09s
860:	learn: 6436.8477988	total: 6.7s	remaining: 1.08s
861:	learn: 6434.0858547	total: 6.7s	remaining: 1.07s
862:	learn: 6427.2884979	total: 6.71s	remaining: 1.06s
863:	learn: 6416.7556202	total: 6.72s	remaining: 1.06s
864:	learn: 6407.0099403	total: 6.73s	remaining: 1.05s
865:	learn: 6396.6958852	total: 6.74s	remaining: 1.04s
866:	learn: 6383.7954715	total: 6.74s	remaining: 1.03s
867:	learn: 6377.7792857	total: 6.75s	remaining: 1.03s
868:	learn: 6367.5060758	total: 6.76s	remaining: 1.02s
869:	learn: 6362.4368897	total: 6.76s	remaining: 1.01s
870:	learn: 6359.4999990	total: 6.77s	remaining: 1s
871:	learn: 6354.4814920	total: 6.78s	remaining: 995ms
872:	learn: 6343.7185538	total: 6.79s	remaining: 988ms
873:	learn: 6338.9433751	total: 6.79s	remaining: 980ms
874:	learn: 6332.0731429	total: 6.8s	remaining: 972ms
875:	learn: 6323.9285423	total: 6.81s	remaining: 964ms
876:	learn: 6318.2525960	total: 6.82s	remaining: 957ms
877:	learn: 6311.3334488	total: 6.83s	remaining: 949ms
878:	learn: 6305.5765754	total: 6.83s	remaining: 941ms
879:	learn: 6296.9861839	total: 6.84s	remaining: 933ms
880:	learn: 6288.7152589	total: 6.85s	remaining: 926ms
881:	learn: 6278.8512312	total: 6.86s	remaining: 918ms
882:	learn: 6275.3894718	total: 6.87s	remaining: 910ms
883:	learn: 6274.9251886	total: 6.87s	remaining: 902ms
884:	learn: 6267.2410925	total: 6.88s	remaining: 894ms
885:	learn: 6256.3106264	total: 6.89s	remaining: 886ms
886:	learn: 6246.0778465	total: 6.89s	remaining: 878ms
887:	learn: 6235.0799289	total: 6.9s	remaining: 871ms
888:	learn: 6222.2627894	total: 6.91s	remaining: 863ms
889:	learn: 6216.1545884	total: 6.92s	remaining: 855ms
890:	learn: 6209.1163003	total: 6.93s	remaining: 847ms
891:	learn: 6202.4351873	total: 6.93s	remaining: 840ms
892:	learn: 6187.3844285	total: 6.94s	remaining: 832ms
893:	learn: 6183.4731693	total: 6.96s	remaining: 825ms
894:	learn: 6168.9881051	total: 6.97s	remaining: 817ms
895:	learn: 6159.7928847	total: 6.97s	remaining: 809ms
896:	learn: 6151.2195866	total: 6.98s	remaining: 802ms
897:	learn: 6142.8574947	total: 6.99s	remaining: 794ms
898:	learn: 6132.3766153	total: 7s	remaining: 786ms
```

```
899:	learn: 6125.1696417	total: 7s	remaining: 778ms
900:	learn: 6113.5794375	total: 7.01s	remaining: 770ms
901:	learn: 6109.2681946	total: 7.02s	remaining: 763ms
902:	learn: 6101.1545560	total: 7.02s	remaining: 755ms
903:	learn: 6100.7056085	total: 7.03s	remaining: 747ms
904:	learn: 6093.9542197	total: 7.04s	remaining: 739ms
905:	learn: 6081.2252537	total: 7.05s	remaining: 731ms
906:	learn: 6080.8159630	total: 7.05s	remaining: 723ms
907:	learn: 6071.1523466	total: 7.06s	remaining: 716ms
908:	learn: 6070.6185491	total: 7.07s	remaining: 708ms
909:	learn: 6064.4470438	total: 7.08s	remaining: 700ms
910:	learn: 6056.4026717	total: 7.08s	remaining: 692ms
911:	learn: 6043.2942411	total: 7.09s	remaining: 685ms
912:	learn: 6036.8873489	total: 7.1s	remaining: 677ms
913:	learn: 6034.7591320	total: 7.11s	remaining: 669ms
914:	learn: 6028.1466449	total: 7.12s	remaining: 661ms
915:	learn: 6020.0505385	total: 7.12s	remaining: 653ms
916:	learn: 6018.8052722	total: 7.13s	remaining: 645ms
917:	learn: 6013.2309782	total: 7.14s	remaining: 637ms
918:	learn: 6012.8039886	total: 7.14s	remaining: 630ms
919:	learn: 6004.4345456	total: 7.15s	remaining: 622ms
920:	learn: 5998.8567948	total: 7.16s	remaining: 614ms
921:	learn: 5992.1396754	total: 7.16s	remaining: 606ms
922:	learn: 5985.5274768	total: 7.17s	remaining: 598ms
923:	learn: 5984.9940952	total: 7.18s	remaining: 591ms
924:	learn: 5979.9598877	total: 7.19s	remaining: 583ms
925:	learn: 5977.0965322	total: 7.2s	remaining: 575ms
926:	learn: 5973.4938400	total: 7.2s	remaining: 567ms
927:	learn: 5967.1153950	total: 7.21s	remaining: 560ms
928:	learn: 5958.0183181	total: 7.25s	remaining: 554ms
929:	learn: 5954.6433968	total: 7.26s	remaining: 546ms
930:	learn: 5948.9764092	total: 7.27s	remaining: 539ms
931:	learn: 5939.4988563	total: 7.28s	remaining: 531ms
932:	learn: 5939.0592796	total: 7.28s	remaining: 523ms
933:	learn: 5932.7669828	total: 7.29s	remaining: 515ms
934:	learn: 5921.7831019	total: 7.3s	remaining: 507ms
935:	learn: 5914.0014102	total: 7.31s	remaining: 500ms
936:	learn: 5908.0548103	total: 7.31s	remaining: 492ms
937:	learn: 5902.2347880	total: 7.32s	remaining: 484ms
938:	learn: 5901.7304064	total: 7.33s	remaining: 476ms
939:	learn: 5892.9998274	total: 7.33s	remaining: 468ms
940:	learn: 5887.0941490	total: 7.34s	remaining: 460ms
941:	learn: 5886.6505369	total: 7.35s	remaining: 452ms
942:	learn: 5880.9768836	total: 7.36s	remaining: 445ms
943:	learn: 5871.1646907	total: 7.36s	remaining: 437ms
944:	learn: 5865.1664564	total: 7.37s	remaining: 429ms
945:	learn: 5858.3766046	total: 7.38s	remaining: 421ms
946:	learn: 5857.9857367	total: 7.38s	remaining: 413ms
947:	learn: 5847.5222754	total: 7.39s	remaining: 405ms
948:	learn: 5836.7733413	total: 7.4s	remaining: 398ms
```

```
949:	learn: 5829.7396683	total: 7.4s	remaining: 390ms
950:	learn: 5818.0657348	total: 7.41s	remaining: 382ms
951:	learn: 5814.8356846	total: 7.42s	remaining: 374ms
952:	learn: 5812.7853104	total: 7.43s	remaining: 366ms
953:	learn: 5808.2214109	total: 7.43s	remaining: 358ms
954:	learn: 5800.3768335	total: 7.44s	remaining: 351ms
955:	learn: 5794.9708222	total: 7.45s	remaining: 343ms
956:	learn: 5790.8592058	total: 7.46s	remaining: 335ms
957:	learn: 5782.3161323	total: 7.46s	remaining: 327ms
958:	learn: 5774.1081775	total: 7.47s	remaining: 319ms
959:	learn: 5773.6459852	total: 7.48s	remaining: 312ms
960:	learn: 5759.4885855	total: 7.49s	remaining: 304ms
961:	learn: 5748.3285370	total: 7.5s	remaining: 296ms
962:	learn: 5740.4602178	total: 7.51s	remaining: 288ms
963:	learn: 5737.9352958	total: 7.51s	remaining: 281ms
964:	learn: 5735.5930394	total: 7.52s	remaining: 273ms
965:	learn: 5727.9043839	total: 7.53s	remaining: 265ms
966:	learn: 5723.8340790	total: 7.53s	remaining: 257ms
967:	learn: 5719.5251352	total: 7.54s	remaining: 249ms
968:	learn: 5719.0768175	total: 7.55s	remaining: 242ms
969:	learn: 5713.8383728	total: 7.56s	remaining: 234ms
970:	learn: 5699.5606136	total: 7.56s	remaining: 226ms
971:	learn: 5695.6430549	total: 7.57s	remaining: 218ms
972:	learn: 5695.0677962	total: 7.58s	remaining: 210ms
973:	learn: 5691.4416542	total: 7.58s	remaining: 202ms
974:	learn: 5684.7085856	total: 7.59s	remaining: 195ms
975:	learn: 5672.3426330	total: 7.6s	remaining: 187ms
976:	learn: 5662.8819202	total: 7.6s	remaining: 179ms
977:	learn: 5654.2602235	total: 7.61s	remaining: 171ms
978:	learn: 5649.4501153	total: 7.62s	remaining: 163ms
979:	learn: 5646.2623989	total: 7.62s	remaining: 156ms
980:	learn: 5642.2572092	total: 7.63s	remaining: 148ms
981:	learn: 5636.3494739	total: 7.63s	remaining: 140ms
982:	learn: 5629.1801576	total: 7.64s	remaining: 132ms
983:	learn: 5622.4477708	total: 7.65s	remaining: 124ms
984:	learn: 5612.4268946	total: 7.66s	remaining: 117ms
985:	learn: 5608.0437291	total: 7.66s	remaining: 109ms
986:	learn: 5601.0800603	total: 7.67s	remaining: 101ms
987:	learn: 5600.7279403	total: 7.68s	remaining: 93.3ms
988:	learn: 5597.5422912	total: 7.69s	remaining: 85.5ms
989:	learn: 5589.6119644	total: 7.69s	remaining: 77.7ms
990:	learn: 5582.7931937	total: 7.7s	remaining: 69.9ms
991:	learn: 5582.3347656	total: 7.71s	remaining: 62.2ms
992:	learn: 5582.0052518	total: 7.72s	remaining: 54.4ms
993:	learn: 5575.1620552	total: 7.73s	remaining: 46.6ms
994:	learn: 5571.1729321	total: 7.73s	remaining: 38.9ms
995:	learn: 5567.7584596	total: 7.74s	remaining: 31.1ms
996:	learn: 5562.9326863	total: 7.75s	remaining: 23.3ms
997:	learn: 5557.4777975	total: 7.76s	remaining: 15.5ms
```

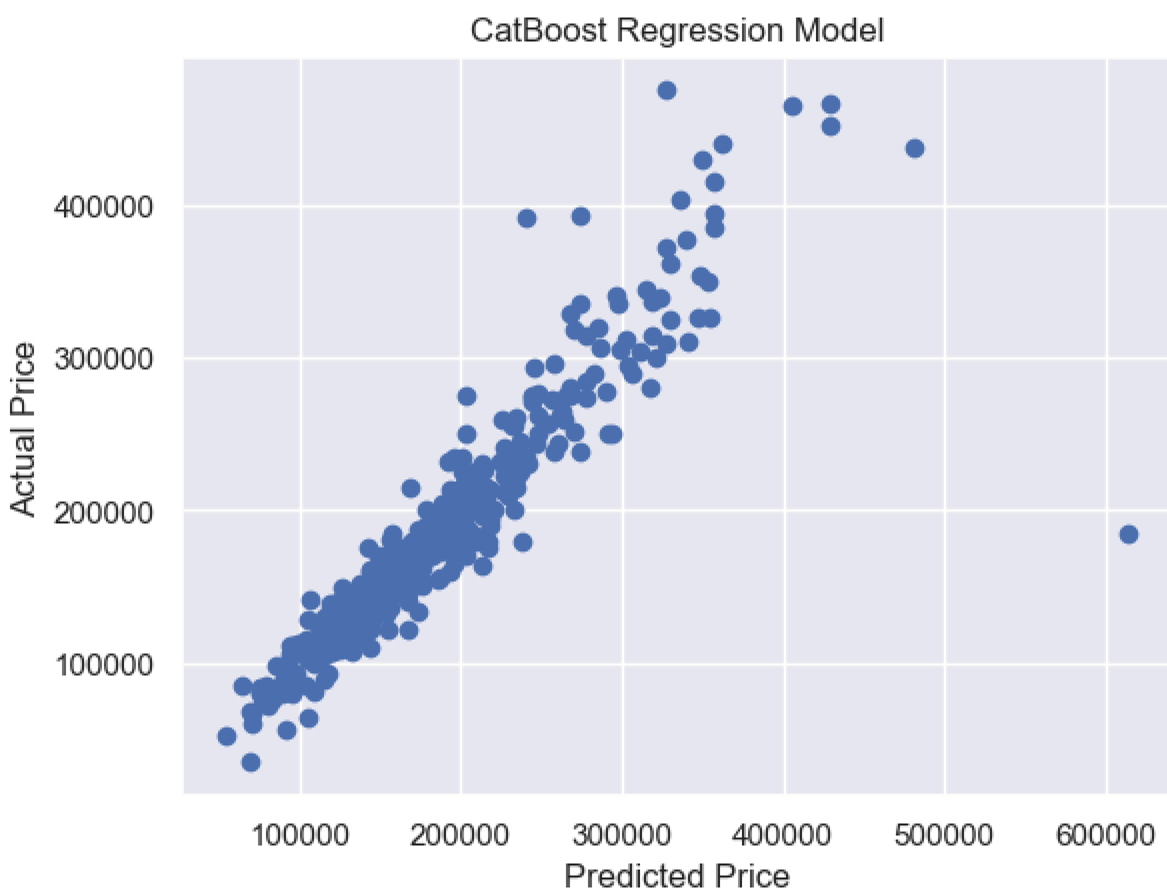
```
998:	learn: 5551.4630824	total: 7.77s	remaining: 7.78ms
999:	learn: 5548.3047057	total: 7.78s	remaining: 0us
```

In [56]:
```python
accuracy_score = classifier.score(x_test,y_test)
print("Accuracy score           :", accuracy_score)
print('RMSE                     :', np.sqrt(metrics.mean_squared_error(y_test, y_predict)))
```

```
Accuracy score           : 0.8411645693158317
RMSE                     : 30302.53804008603
```

In [57]:
```python
# alpha helps to show overlapping data
plt.scatter(y_predict, y_test, color = 'b')
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('CatBoost Regression Model')
```

Out[57]:
```
Text(0.5, 1.0, 'CatBoost Regression Model')
```




## Comparing all the models based on Model Performance

In [58]:
```python
comparison_frame = pd.DataFrame({'Model':['Accuracy score                :',
                                          'RMSE                          :'],
                                 'Linear Regression':[0.8106,33088.26],
                                 'Random Forest':[0.8237,31919.91],
                                 'Gradient Boosting':[0.8554,28911.11],
                                 'CatBoost':[0.8411,30302.53]})


comparison_frame
```

Out[58]:

| | Model | Linear Regression | Random Forest | Gradient Boosting | CatBoost |
|---|---|---|---|---|---|
| **0** | Accuracy score : | 0.8106 | 0.8237 | 0.8554 | 0.8411 |
| **1** | RMSE : | 33088.2600 | 31919.9100 | 28911.1100 | 30302.5300 |

## Results interpretation :

Accuracy Score : Examining the updated accuracy_score reveals distinct performance characteristics among the algorithms. GradientBoosting Regressor stands out with an impressive accuracy score of 0.8554, signifying its exceptional ability to explain 85.54% of the variance in the target variable. Maintaining a robust position, CatBoost Regressor yields an accuracy score of 0.8411, showcasing reliable predictive capabilities with an explanation of approximately 84.11% of the variance. Random Forest, though slightly behind with an accuracy score of 0.8237, still provides a strong explanation of around 82.37% of the variance. Linear regression has the lowest of all with an accuracy score of 0.8106. These results underscore the nuanced strengths of each algorithm, with GradientBoosting demonstrating superior accuracy.

RMSE : GradientBoosting outperforms all 4 in terms of RMSE, achieving the lowest score of 28911.1100. This suggests that GradientBoosting provides more accurate predictions.

## So, the best result is obtained by Gradient Boosted Classifier.