HOUSING: PRICE PREDICTION

Submitted by:

Shivani Kataria

ACKNOWLEDGMENT

Resources that helped me and guided me in completion of the project is as follows:
DataTrained Documents & Trainings
https://imbalanced-learn.org/stable/over_sampling.html
machinelearningmastery.com/ran...for-imbalanced-classification/
stackoverflow for error handlings.

INTRODUCTION

- Business Problem Framing

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

- Conceptual Background of the Domain Problem

  A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below.

  The company is looking at prospective properties to buy houses to enter the market.

Review of Literature

- required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm

and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

- Motivation for the Problem Undertaken

To analyze and have more knowledge of the repective domain and hence forth predicting the Sales Prices of Houses was motivational and made learning of some concepts like
Hyper parameter tuning,
Cleaning of Dataset,
Binning of Year columns
Ordinal encodings and nominal encodings.
Handling two types of dataset and predicting the output variable.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  a. Some of the basic mathematical calculations were used such as for calculating the data loss percentage.
  b. Statistical methods of z score for handling outliers , concept of skewness, correlation between variables.
  c. and analytics modelling with different classification type models for accuracy checking and predicting the output .

- Data Sources and their formats

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

  MSZoning: Identifies the general zoning classification of the sale.

  LotFrontage: Linear feet of street connected to property

  LotArea: Lot size in square feet

  LotShape: General shape of property

  LotConfig: Lot configuration

LandSlope: Slope of property

Neighbourhood: Physical locations within Ames city limits

Condition1: Proximity to various conditions

Condition2: Proximity to various condition

HouseStyle: Style of dwelling

OverallQual: Rates the overall material and finish of the house

OverallCond: Rates the overall condition of the house

YearBuilt: Original construction date

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

RoofStyle: Type of roof

RoofMatl: Roof material

Exterior1st: Exterior covering on house

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

MasVnrType: Masonry veneer type

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

Foundation: Type of foundation

BsmtCond: Evaluates the general condition of the basement

BsmtExposure: Refers to walkout or garden level walls

BsmtFinSF1: Type 1 finished square feet

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

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

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

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

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

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

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

GarageType: Garage location

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

Fence: Fence quality

MiscFeature: Miscellaneous feature not covered in other categories

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

SaleType: Type of sale

SaleCondition: Condition of sale

- Data Preprocessing Done

Removal of unwanted columns were handled.
Tried to remove Outliers.
Also Skewness were checked for making to normal distribution.
Datatypes were identified .
Null datas were checked .

- Data Inputs- Logic- Output Relationships

Data contains 1460 entries each having 81 variables.

Data contains Null values. It was handled using the domain knowledge and own understanding.

Extensive EDA has been performed to gain relationships of important variable and price.

Data contains numerical as well as categorical variable. Building Machine Learning models, apply regularization and determined the optimal values of Hyper Parameters.

- Hardware and Software Requirements and Tools Used

d. System with 4GB ram,
e. Jupyter notebook for coding,

f. Excel,
g. Word document,
h. Anacondas full set installations
i. With libraries like:
j. Pandas
k. numpy
l. matplotlib
m. seaborn
n. sklearn

Model/s Development and Evaluation

Removing the unwanted columns which affect the accuracy.
To findout the outliers we are doing the below plots with boxplot .
We use special method to remove outliers since columns are with skewness and not normally distributed.
Tried with Z score to remove outliers.
Trying to reduce skewness with the help of Power Transform method.
Checking for the best random state for better accuracy.
Standardistion of the x,y variables.
Model Bulding with different kind of model for better accuracy.

- Testing of Identified Approaches (Algorithms)

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20,random_state=maximum_randomstate)

- Run and Evaluate selected models
- lm=LinearRegression()
- lm.fit(x_train,y_train)

- pred=lm.predict(x_test)
- print("Coefficient : ",lm.coef_)
- print("Intercept : ",lm.intercept_)
- print("Score : ",lm.score(x_train,y_train))
- print(' ')
- print("error")
- print("Mean absolute error :",mean_absolute_error(y_test,pred))
- print("Mean squared error :",mean_squared_error(y_test,pred))
- print("Root mean squared error:",np.sqrt(mean_squared_error(y_test,pred)))
- print(' ')
- #r2 score ------> coefficient of determination
- #i.e. change coming in y whenever x is being changed.
- from sklearn.metrics import r2_score
- print("r2 score : ",r2_score(y_test,pred)

```
ls=Lasso(alpha=0.0001)
   ls.fit(x_train,y_train)
   pred=ls.predict(x_test)
   print("Coefficient : ",ls.coef_)
   print("Intercept   : ",ls.intercept_)
   print("Score       : ",ls.score(x_train,y_train))

rf=RandomForestRegressor()
```

```
rf.fit(x_train,y_train)

pred=rf.predict(x_test)

rf.score(x_train,y_train)

#print("Coefficient : ",rf.coef_)

# print("Intercept   : ",rf.intercept_)

print("Score       : ",rf.score(x_train,y_train))

print(' ')

print("error")

print("Mean absolute
error    :",mean_absolute_error(y_test,pred))

print("Mean squared
error     :",mean_squared_error(y_test,pred))

print("Root mean squared
error:",np.sqrt(mean_squared_error(y_test,pred)))


gbr=GradientBoostingRegressor()

gbr.fit(x_train,y_train)

pred=gbr.predict(x_test)

gbr.score(x_train,y_train)

#print("Coefficient : ",rf.coef_)

# print("Intercept   : ",rf.intercept_)

print("Score       : ",gbr.score(x_train,y_train))
```

```
print(' ')
print("error")
print("Mean absolute
error    :",mean_absolute_error(y_test,pred))
print("Mean squared
error     :",mean_squared_error(y_test,pred))
print("Root mean squared
error:",np.sqrt(mean_squared_error(y_test,pred)))
print(' ')
#r2 score ------> coefficient of determination
#i.e. change coming in y whenever x is being changed.
from sklearn.metrics import r2_score
print("r2 score : ",r2_score(y_test,pred))

adb=AdaBoostRegressor()
adb.fit(x_train,y_train)
pred=adb.predict(x_test)
adb.score(x_train,y_train)
print("Score       : ",adb.score(x_train,y_train))
print(' ')
print("error")
```

```
print("Mean absolute
error    :",mean_absolute_error(y_test,pred))

print("Mean squared
error     :",mean_squared_error(y_test,pred))

print("Root mean squared
error:",np.sqrt(mean_squared_error(y_test,pred)))

print(' ')

from sklearn.metrics import r2_score

print("r2 score : ",r2_score(y_test,pred))
```

- Key Metrics for success in solving problem under consideration

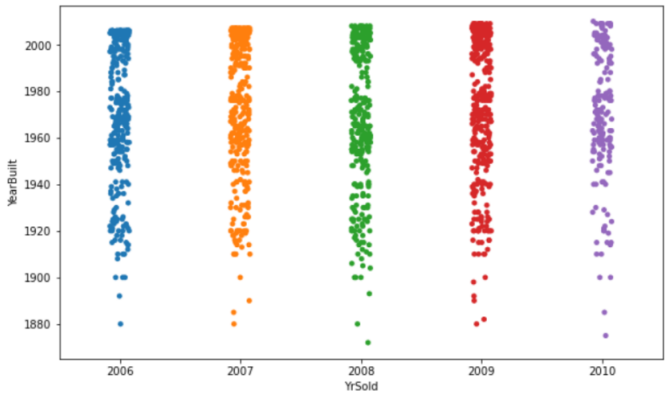
```
scr=cross_val_score(rf,x,y,cv=5,scoring="r2")

print("Cross Validation Score of RandomForestRegressor
Model is : ", scr.mean())


scr=cross_val_score(gbr,x,y,cv=5,scoring='r2')

print("Cross Validation Score of
GradientBoostingRegressor Model is : ", scr.mean())


parameters = {'learning_rate': [0.01,0.02,0.03,0.04],
```

```
                'subsample'    : [0.9, 0.5, 0.2, 0.1],
                'n_estimators' : [100,500,1000, 1500]}
gcv=GridSearchCV(GradientBoostingRegressor(),
parameters,cv=5,scoring="r2")

gcv.fit(x_train,y_train)
gcv.best_params_

mod2=GradientBoostingRegressor(learning_rate=0.01,n
_estimators=500,subsample=
0.1,random_state=maximum_randomstate)
mod2.fit(x_train,y_train)
pred=mod2.predict(x_test)
print(r2_score(y_test,pred)*100)
```

- Visualizations

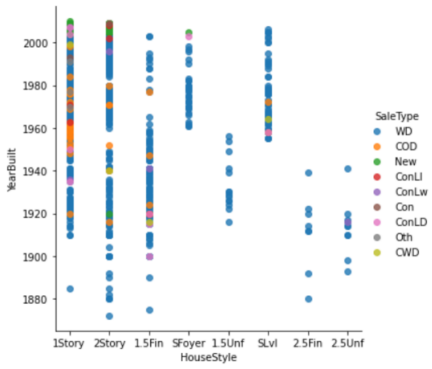
In [85]:
```
plt.figure(figsize=(10,6))
sns.stripplot(x="YrSold", y="YearBuilt", data=df_concat, jitter=.08)
```

Out[85]: <AxesSubplot:xlabel='YrSold', ylabel='YearBuilt'>



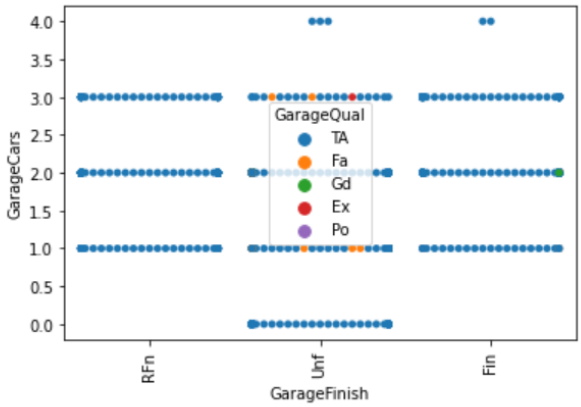
In [84]:
```
sns.lmplot(x ='HouseStyle', y ='YearBuilt',fit_reg = False, hue = 'SaleType',data = df_concat)
```

Out[84]: <seaborn.axisgrid.FacetGrid at 0x19d5a436610>



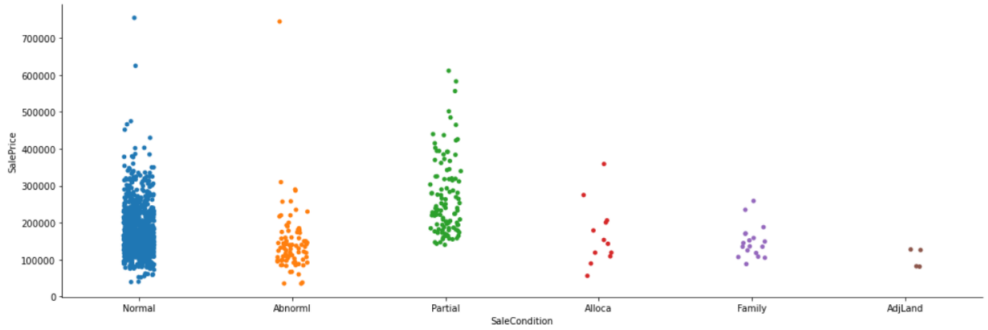
In [83]:
```
sns.swarmplot(x="GarageFinish", y="GarageCars",hue='GarageQual', data=df_concat)

plt.xticks(rotation=90)
```

Out[83]:
```
(array([0, 1, 2]), [Text(0, 0, 'RFn'), Text(1, 0, 'Unf'), Text(2, 0, 'Fin')])
```



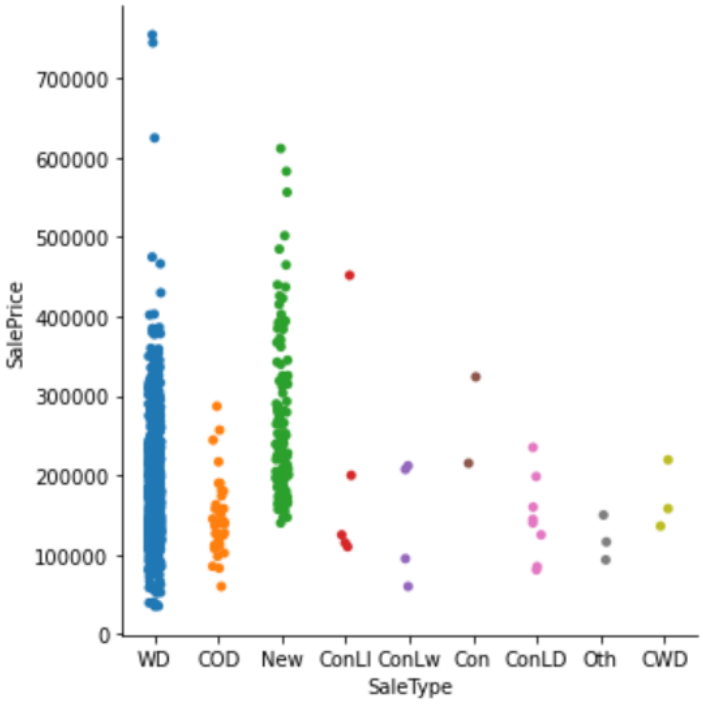
In [82]:
```
sns.catplot(x='SaleCondition',y='SalePrice',data = df_concat.sort_values("SalePrice", ascending = False), height = 5, aspect = 3)
```

Out[82]:
```
<seaborn.axisgrid.FacetGrid at 0x19d5b88d6d0>
```

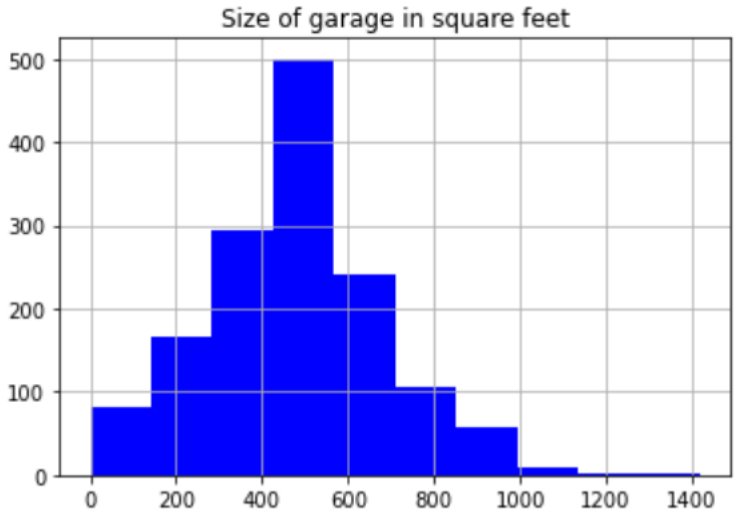


In [81]:
```
sns.catplot(x='SaleType',y='SalePrice',data=df_concat)
```

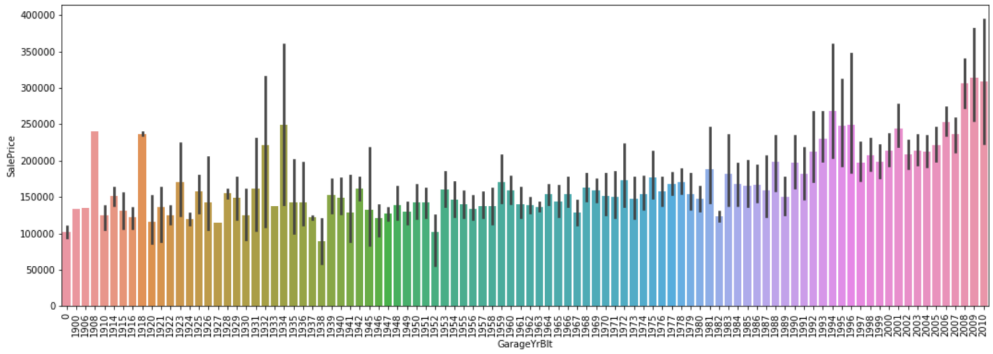
Out[81]:
```
<seaborn.axisgrid.FacetGrid at 0x19d5bdfca30>
```



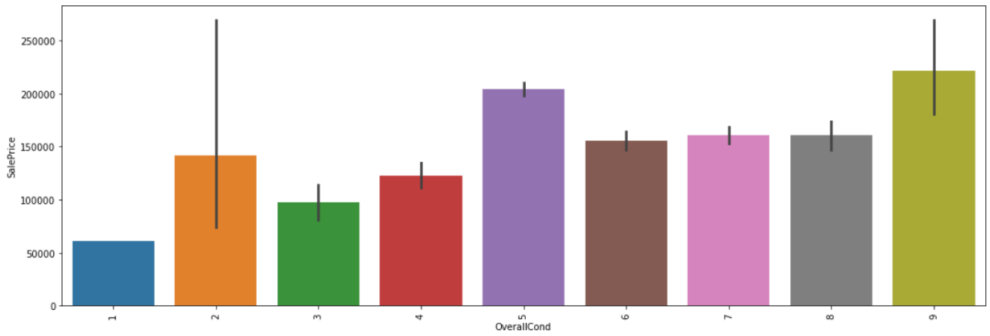
In [68]:
```
df_concat["GarageArea"].hist(grid=True,color='blue')
plt.title("Size of garage in square feet")
plt.show()
```



In [66]:
```
plt.figure(figsize=(18,6))
sns.barplot(x="GarageYrBlt",y="SalePrice",data=df_concat)
plt.xticks(rotation=90)
plt.show()
```

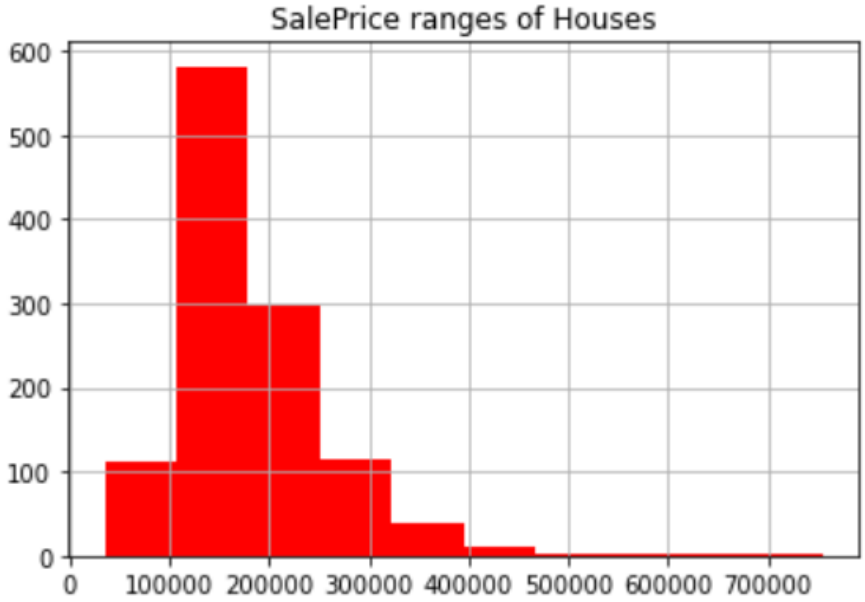


In [41]:
```
plt.figure(figsize=(18,6))
sns.barplot(x="OverallCond",y="SalePrice",data=df_concat)
plt.xticks(rotation=90)
plt.show()
```



EDA

```
df_train["SalePrice"].hist(grid=True,color='red')
plt.title("SalePrice ranges of Houses")
plt.show()
```



- Interpretation of the Results

  we have got Best accuracy with GradientBoostingRegressor model. We have saved and loaded that for checking purpose and predicted in above steps..

CONCLUSION

we have got Best accuracy with GradientBoostingRegressor model with score of 89%. We have saved and loaded that for checking purpose and predicted in above steps.