House Prices - Advanced Regression Techniques

Introduction

In this project, our dataset will be coming from a Kaggle competition. In this competition, we are utilizing 79 explanatory variables to predict a final price of a house. These variables vary from the number of bedrooms to the type of foundation and even roof material. The goal of this competition is to utilize different regression techniques on our dataset and predict the sales price. In this competition, submissions are graded by checking for the Root-mean-squared error. This is a measure of accuracy, through the predicted sales price and the observed sales price.

Regression

Linear Regression put simply is the measurement of the influence of one or more variables on another variable. These variables can be continuously independent or dependent. In most cases, we use the independent variable to predict the value or sum of a dependent variable. Using linear regression helps us calculate the “line of best fit “ on a dataset. In order to do so, we use the least squares method to help us find the equation of the line. We know the equation for the slope is ‘y = mx + b ’. And the least squares method is calculated by the equation ->

$$m = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

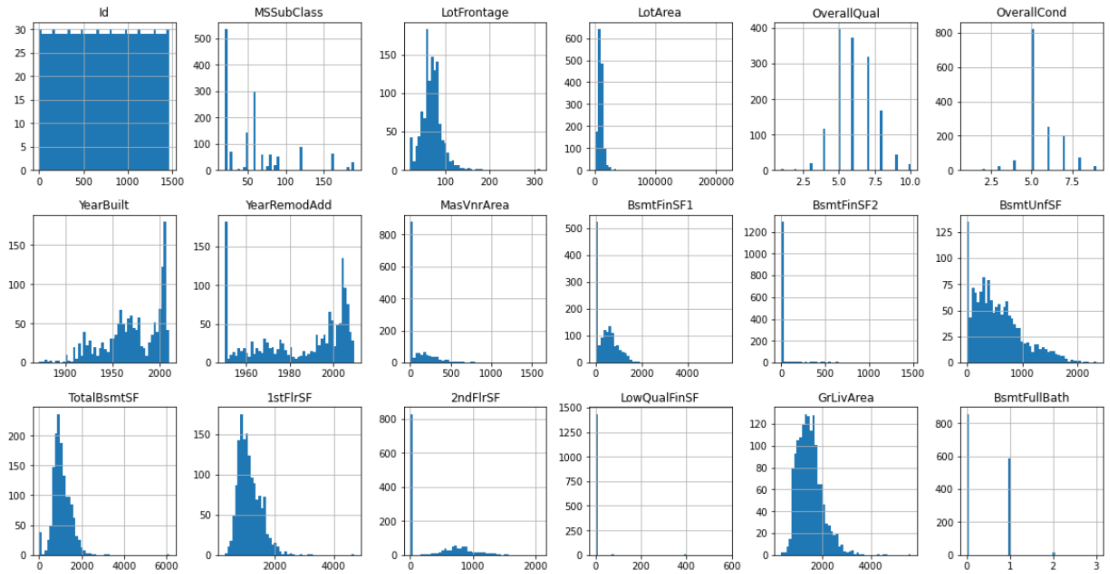
In this equation, m is the slope, b is the intercept and x’ and y’ are means of x and y values. For starting linear regression on our project we’ll first define the dependent and independent variables. Then create the scatter plot and create the linear regression line on the chart using the ordinary least squares model.

Data Understanding

Understanding the dataset took a reasonable amount of time. I was utilizing jupyter notebook to make visualizations made the process much smoother. Before diving in with juypter, I took a look into the many variables we have. Several variables that caught my attention were ‘Month Sold’, ‘Lot area’, and the ‘first-floor square feet. I thought that these variables were more crucial to the housing price because these are the keywords home buyers look for when buying. Looking at the dataset I decided to create a histogram in order to create a visualization to understand the data as a whole. After doing so I can now see the range of the data we have. From this visualization I understood the distribution of the data, some of the variables have a wider distribution than other variables.

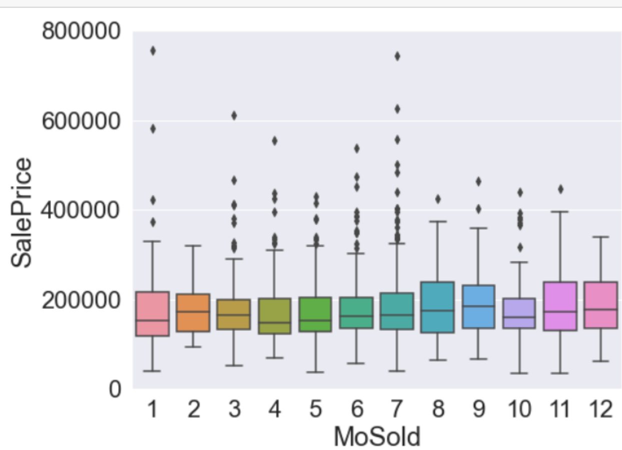
Data Visualization

In [60]:
```
train_df.hist(bins=50, figsize=(20,25))
plt.show()
```



In addition to a histogram, creating a boxplot was useful for seeing how similar the prices were despite the different variables. In my case, I compared the relationship with Saleprice and the month-sold variable.

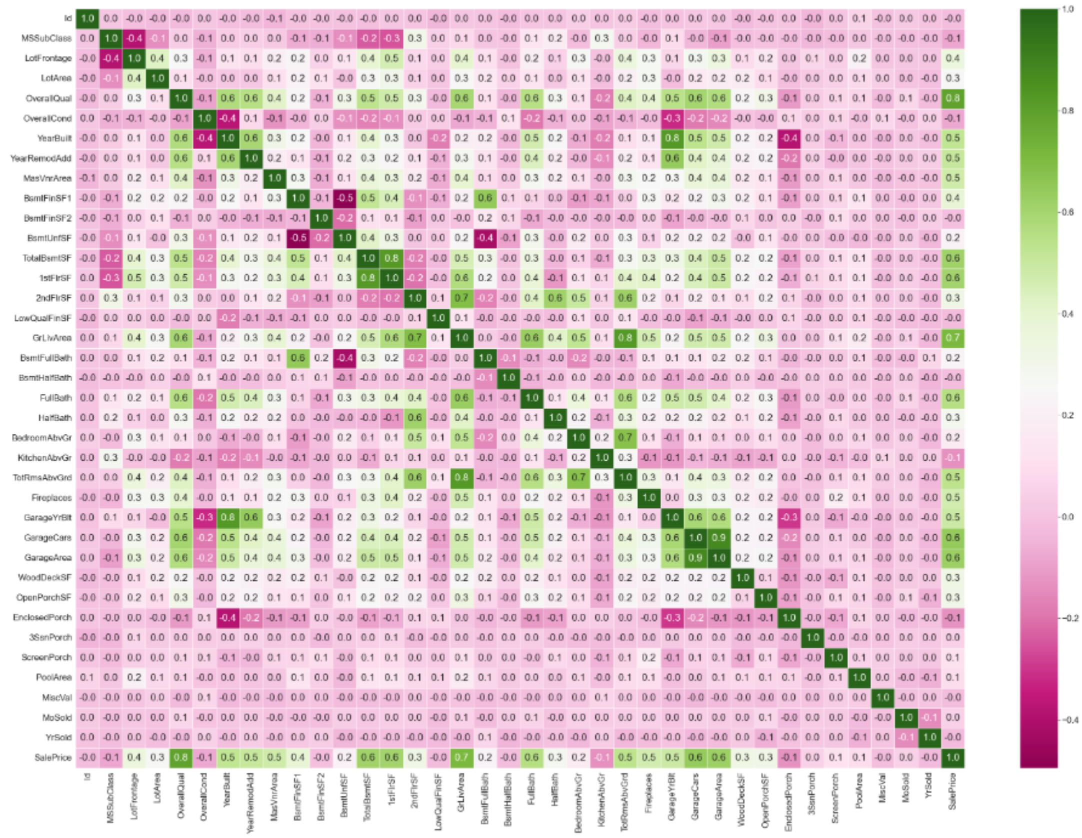
In [74]:
```
# month sold and saleprice boxplot

var = 'MoSold'
data = pd.concat([train_df['SalePrice'], train_df[var]], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
```



Creating a correlation chart was useful as well. Through this chart, we can tell that the variables ‘OverallQual’, ‘GrLivArea’, ‘TotalBsmtSF’, ‘FullBath’, ‘GarageCars’, and ‘GarageArea’ had the strongest relationship to the Sales Price.

In [160]:

```
import seaborn as sb
correlation_train=train_df[train_df.dtypes[train_df.dtypes != 'object'].index].corr()
sb.set(font_scale=2)
plt.figure(figsize = (50,35))
ax = sb.heatmap(correlation_train, annot=True,annot_kws={"size": 25},fmt='.1f',cmap='PiYG', linewidths=.5)
```



Pre-processing

Processing the data required removing the null variable. This required first checking and choosing whether to delete our insert values for the variables. In this case, I decided to remove the null variables to make the process easier for me.

#Checking for null values

```
In [80]: test_df.isnull().sum()
```

```
Out[80]: Id                 0
         MSSubClass         0
         MSZoning           4
         LotFrontage      227
         LotArea            0
                         ...
         MiscVal            0
         MoSold             0
         YrSold             0
         SaleType           1
         SaleCondition      0
         Length: 80, dtype: int64
```

```
In [81]: train_df.isnull().sum()
```

```
Out[81]: Id                 0
         MSSubClass         0
         MSZoning           0
         LotFrontage      259
         LotArea            0
                         ...
         MoSold             0
         YrSold             0
         SaleType           0
         SaleCondition      0
         SalePrice          0
         Length: 81, dtype: int64
```

```
In [88]: # Dropping all Null values in test dataset
         test_df = test_df.dropna(subset = ['LotFrontage'])
         test_df = test_df.dropna(subset = ['MSZoning'])
         test_df = test_df.dropna(subset = ['SaleType'])
         test_df.isnull().sum()
```

```
Out[88]: Id               0
         MSSubClass       0
         MSZoning         0
         LotFrontage      0
         LotArea          0
                         ..
         MiscVal          0
         MoSold           0
         YrSold           0
         SaleType         0
         SaleCondition    0
         Length: 80, dtype: int64
```

Modeling / Experiments

In my experiments, I will be looking at the variables Lot Area, Overall quality, and 1st-floor square. Creating a scatter plot will help us easily see the relationship between the values

Experiment 1:

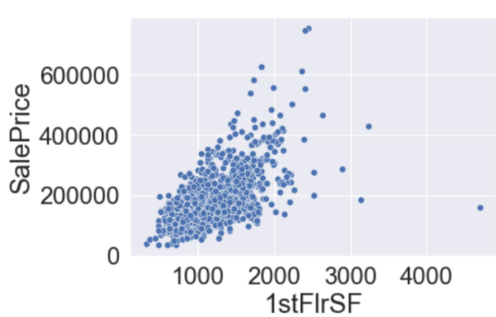
In this experiment evaluation required finding the root mean square error. This function calculates the transformation between values predicted by a model and actual values. Simply put, its another way to evaluate linear regression models.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Creating the model was a smooth process, I did had have issues calculating the rot mean square error because of this specific error; “**ValueError**: could not convert string to float: 'RL'”.WIth the first model I as able to find the coefeefincent and intercepts. Utilizing this would have helped me complete the root mean square error function

```
File ~\anaconda3\lib\site-packages\sklearn\utils\validation.py:746, in check_array(array, accept_sparse, accept_large_sparse, d
type, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, estimator)
    744         array = array.astype(dtype, casting="unsafe", copy=False)
    745     else:
--> 746         array = np.asarray(array, order=order, dtype=dtype)
    747 except ComplexWarning as complex_warning:
    748     raise ValueError(
    749         "Complex data not supported\n{}\n".format(array)
    750     ) from complex_warning

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:2064, in NDFrame.__array__(self, dtype)
   2063 def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
-> 2064     return np.asarray(self._values, dtype=dtype)

ValueError: could not convert string to float: 'RL'
```

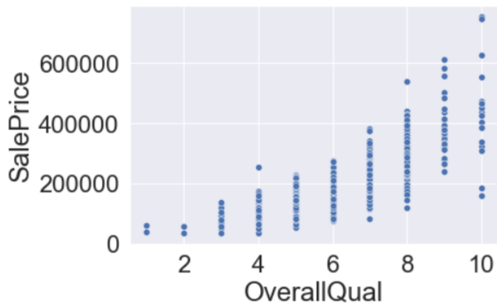
```
In [195]: sns.scatterplot(x='1stFlrSF', y='SalePrice', data=train_df)
Out[195]: <AxesSubplot:xlabel='1stFlrSF', ylabel='SalePrice'>
```



Experiment 2

In this part I wanted to check for regression between the ‘overall quallity’ variable and saleprice. Once the model was created, I could already guess a strong linear correlation given that the values are staple.

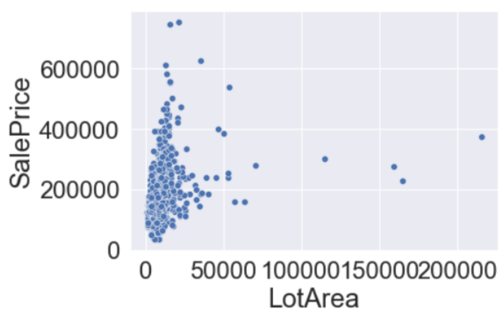
```
In [196]: sns.scatterplot(x='OverallQual', y='SalePrice', data = train_df)
Out[196]: <AxesSubplot:xlabel='OverallQual', ylabel='SalePrice'>
```



Experiment 3

In this part I wanted to check for regression between the ‘Lot Area’ variable and saleprice. I wasn’t able to get past with the similar error on this model either.

```
In [191]: sns.scatterplot(x='LotArea', y='SalePrice', data=train_df)
Out[191]: <AxesSubplot:xlabel='LotArea', ylabel='SalePrice'>
```



Impact

The impact of this project would be positive to new homebuyers. I think looking at such relationships will help new homebuyers decide what features to focus on based on their budget.

Conclusion

I’ve learned alot from this project. Personally my favorite part of this project was creating the visulaizations. My least favorite part was attempting the linear regression evaluations. I had a hard time getting through this part of the project. I believe my issue started with preprocessing and organinzing the data set. Although I didn’t reach the main goal of this project with my data set, I was able to see different examples on youtube and on kaggle of the liner regression model and root mean squared error. Doing so allowed me to understand the process and how it can be helpful in today's world.