

Predictive Analytics for Real Estate Selling Price

Group Project – Team 1

CIS 9557 – Business Analytics – Section QMWA

Phase #3 – 12/09/19



Shobhit Ratan (shobhit.ratan@baruchmail.cuny.edu)

Sabbir Ahmed (sabbir.ahmed1@baruchmail.cuny.edu)

Rosario Campoverde (rosario.campoverdeperez@baruchmail.cuny.edu)

Trisha Gangadeen (Trisha.Gangadeen@baruchmail.cuny.edu)

Yifan Huang (yifan.huang1@baruchmail.cuny.edu)

Valeriia Vodianova (valeriia.vodianova@baruchmail.cuny.edu)

Daniel Xu (chaodi.xu@baruchmail.cuny.edu)

Table of Contents

Project Description	4
Business Problem	4
Data Available	4
Target Variable Predictive Analytics Task.....	4
Exploratory data analysis	5
Frequency of the target variable	5
Missing values, duplicates	7
Relationship between variables	8
'SalePrice' vs Other Attributes.....	9
Outliers.....	20
Baseline Model	21
Generalized Linear Model – Split Validation – Default Parameters	22
Deep Learning – Split Validation – Default Parameters	22
Random Forest – Split Validation – Default Parameters	22
Decision Tree – Split Validation – Default Parameters.....	23
Gradient Boosted Trees – Split Validation – Default Parameters	23
Generalized Linear Model – Cross Validation – Default Parameters.....	23
Random Forest – Cross Validation – Default Parameters.....	24
Decision Tree – Cross Validation – Default Parameters	24
Gradient Boosted Trees – Split Validation – Default Parameters	25
Comparison of Split Validation and Cross Validation Models.....	25
Most Important Features	26
Parameter Optimization.....	28
Generalized Linear Model – Best Model Performance- Root_Mean_Squared_Error (RMSE)28	
Generalized Linear Model – Best Model Performance – Correlation (Corr.)	28
Deep Learning – Best Model Performance – Root_Mean_Squared_Error (RMSE)	29
Deep Learning – Best Model Performance – Correlation (Corr.).....	29
Random Forest – Best Model Performance – Root_Mean_Squared_Error (RMSE)	30
Random Forest – Best Model Performance – Correlation (Corr.).....	30
Decision Tree – Best Model Performance – Root_Mean_Squared_Error (RMSE).....	31
Decision Tree – Best Model Performance – Correlation (Corr.)	31
Gradient Boosted Trees – Best Model Performance - Root_Mean _Squared_Error (RMSE)32	
Gradient Boosted Trees – Best Model Performance – Correlation (Corr.)	32
Model Building	33
Analysis and Recommendations	33
Appendix.....	33
House Prices Dataset Attributes Description	35
Generalized Linear Model – Default Parameters	37

Generalized Linear Model – Changed the Missing values handling from MeanImpute to Skip	38
Generalized Linear Model – Changed the Max Iterations from 0 to 5.....	38
Generalized Linear Model – Changed the lambda from 0 to 0.5.	38
Generalized Linear Model – Changed the lambda from 0.5 to 0.75.	39
Generalized Linear Model – Changed the lambda from 0.75 to 0.25	39
Generalized Linear Model – Changed the lambda from 0.25 to 0.1	40
Generalized Linear Model – Changed the lambda from 0.1 to 0.01	40
Generalized Linear Model – Changed the lambda from 0.1 to 0.001	40
Deep Learning – Default Parameters	41
Deep Learning – Changed the activation function from Tanh to Rectifier.....	42
Deep Learning – Changed the activation function from Maxout to ExpRectifier	43
Deep Learning – Increased the epochs from 7 to 10	43
Deep Learning – Decreased the epochs from 7 to 4.....	44
Random Forest – Default Parameters	45
Random Forest – Changed the maximal depth from 10 to 15.....	46
Random Forest – Changed the maximal depth from 10 to 7.....	47
Random Forest – Changed the number of trees from 100 to 125.....	47
Random Forest – Changed the number of trees from 100 to 75.....	48
Decision Tree – Default Parameters.....	49
Decision Tree – Disabled Apply Prepruning button with everything else remaining same.....	50
Decision Tree – Changed the maximal depth from 10 to 15	50
Decision Tree – Changed the maximal depth from 10 to 20	50
Decision Tree – Changed the minimal gain from 0.01 to 0.02.....	51
Decision Tree – Changed the minimal gain from 0.01 to 0.005.....	51
Decision Tree – Changed the minimal leaf size from 2 to 5	51
Decision Tree – Changed the minimal leaf size from 5 to 8	51
Decision Tree – Changed the minimal size for split from 4 to 5	52
Decision Tree – Changed the minimal size for split from 4 to 3	52
Decision Tree – Changed the number of prepruning alternatives from 3 to 4	52
Decision Tree – Changed the number of prepruning alternatives from 4 to 2	52
Gradient Boosted Trees – Increased the number of trees from 100 to 200.....	55
Gradient Boosted Trees – Increased the number of trees from 200 to 300.....	56
Gradient Boosted Trees – Increased the number of trees from 300 to 400.....	56
Gradient Boosted Trees – Increased the number of trees from 400 to 500.....	57
Gradient Boosted Trees – Increased the number of trees from 500 to 600.....	57
Gradient Boosted Trees – Increased the number of trees from 600 to 700.....	58
Gradient Boosted Trees – Increased the number of trees from 700 to 800.....	58
Gradient Boosted Trees – Increased the number of trees from 1300 to 1400.....	61
Gradient Boosted Trees – Decreased the number of trees from 1300 to 1250	62

Gradient Boosted Trees – Increased the maximal depth from 10 to 15.....62

Gradient Boosted Trees - Increased the maximal depth from 15 to 2063

Project Description

Buying real estate is a tedious and stressful process. Potential homeowner's biggest concern is fair value / safe bidding price while purchasing a house. More often than not home buyers end up bidding more than the fair market value of the real estate. We will analyze a dataset to understand key attributes that influence the housing real estate price. This dataset contains 79 explanatory variables describing every aspect of residential homes in Ames, Iowa. We plan to use '**Regression**' to predict the '**SalePrice**' of the houses.

Business Problem

According to NerdWallet's 2019 Home Buyer Report, approximately 45% of Americans who have purchased a home in the last 5 years offered more than the asking price before having their offer accepted (<https://www.nerdwallet.com/blog/2019-home-buyer-report/>). As a result, people end up losing money. According to forecasts, only 36% of Americans plan to buy a house in the next 5 years. Prospective home buyers are staying away from the real estate market view fear of overpricing and risk of foreclosure.

We believe that our predictive analysis will provide prospective buyers a fair price for real estate, thus preventing overbidding while purchasing their dream house. Availability of fair prices will manifest in monetary savings for home buyers, reduce foreclosure rates and boost customer sentiment in real estate market.

Data Available

Our data set is available on following Web link:

<https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

This dataset contains **80** attribute and **1,460** instances, which represent every house in Ames, Ohio. In the beginning of our analysis, we will use 78 out of 80 attributes. However, after exploratory data analysis only the most important attributes will be used for further analysis. House price dataset attributes are enclosed at Appendix 1.

The dataset is by and large clean and only 7.5% of the attributes have missing information. Distribution of this missing data (NA) is as follows:

- 18% of the rows in LotFrontage
- 94% of the rows in Alley
- 47% of the rows in FireplaceQu
- 6% of the rows in GarageYrBlt,
- 100% of the rows in PoolQC
- 81% of the rows in Fence.

Target Variable | Predictive Analytics Task

Our **Target variable** will be '**SalePrice**' for all the houses in Ames, Iowa. We plan to use '**Regression**' to predict the '**SalePrice**' of the houses.

Exploratory data analysis

Frequency of the target variable

The target variable 'SalePrice' follows a positively skewed distribution as shown in Figure 1.

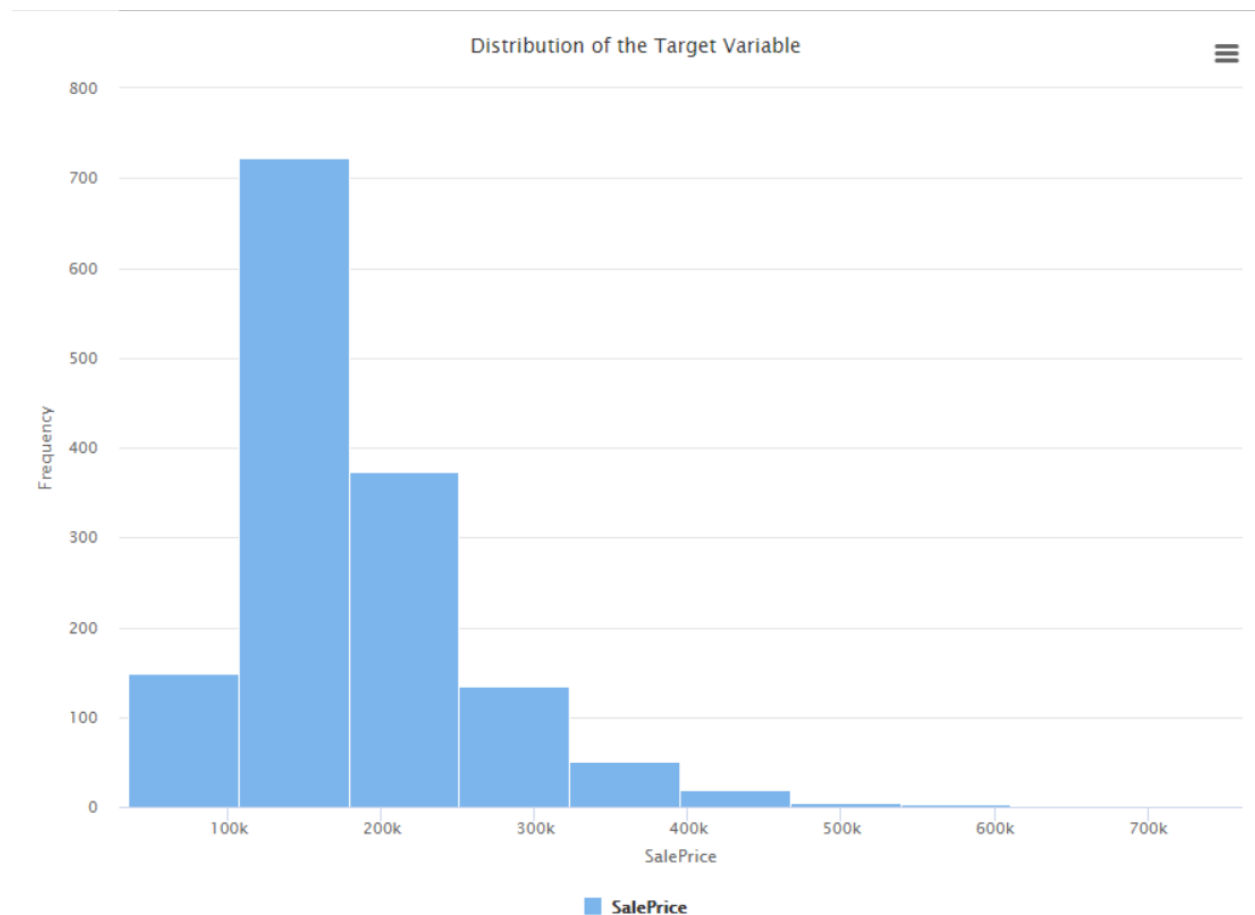


Figure 1: Distribution of the Target Variable: 'SalePrice'

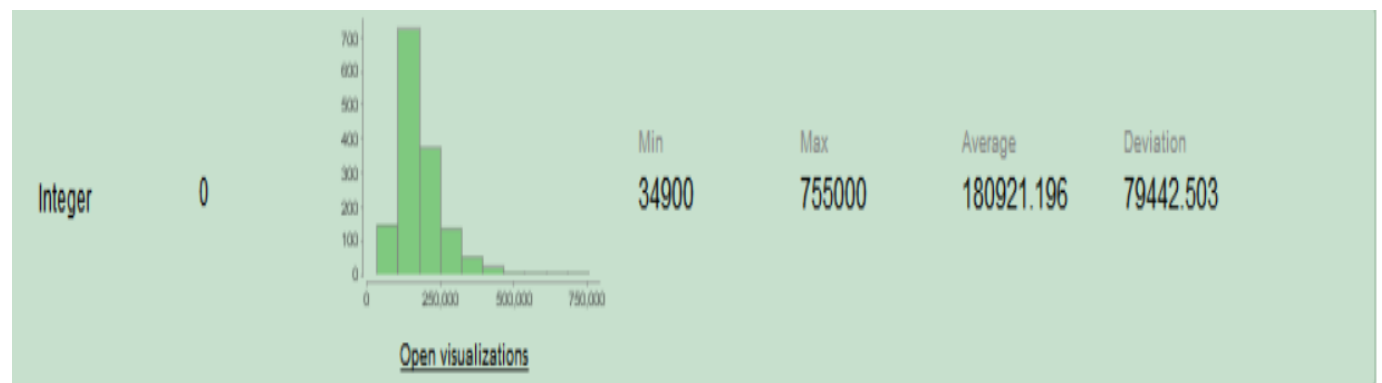


Figure 2: Descriptive Statistics: 'SalePrice'

According to Figure 1 and 2, the data is positively skewed with a few extreme values. The frequency of the distribution is as follows:

SI	Price Range (\$)	Frequency
1	34,900 – 106,910	148
2	106,910 – 178,920	723
3	178,920 – 250,930	373
4	250,930 – 322,940	135
5	322,940 – 394,950	51
6	394,950 – 466,960	19
7	466,960 – 538,970	4
8	538,970 - 610,980	3

Table 1: Frequency of the Target Variable: 'SalePrice'

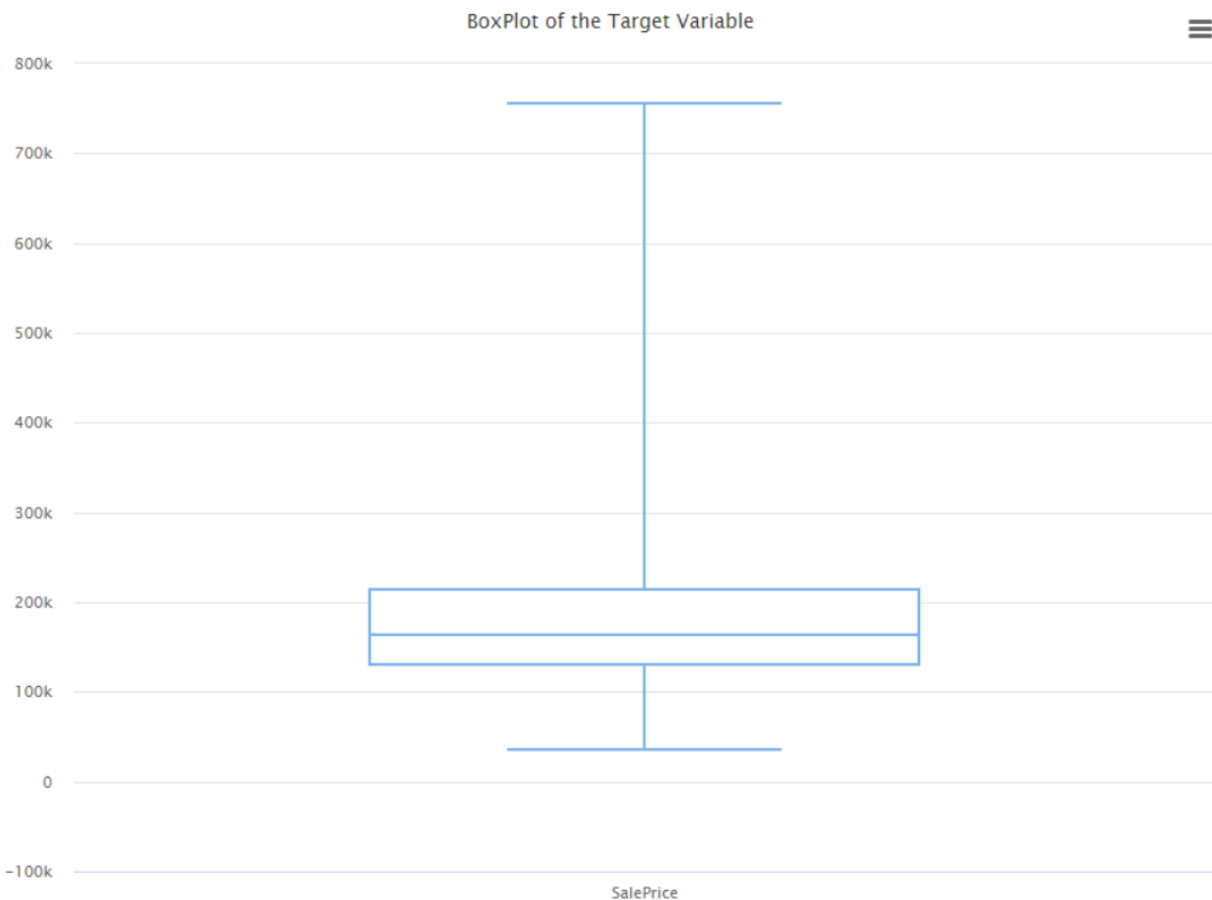


Figure 3: Boxplot Analysis of the Target Variable: 'SalePrice'

The maximum Sales Price is \$755,000 and the minimum is \$34,900. The average Sales price is \$180,921.195 and the median is \$163,000.

Missing values, duplicates

As stated in Phase I, only 7.5% of the attributes have missing information. Those attributes had minimal entropy and they didn't offer any information gain. Thus, we didn't consider the following attributes for further analysis:

SI	Attributes	Description
1	3SsnPorch	Three season porch area in square feet.
2	Alley	Type of alley access
3	BsmtHalfBath	Basement half bathrooms
4	CentralAir	Central air conditioning
5	Condition1	Proximity to main road or railroad
6	Condition2	Proximity to main road or railroad (if a second is present)
7	Electrical	Electrical system
8	Functional	Home functionality rating
9	GarageCond	Garage condition
10	Heating	Type of heating
11	ID	Identification Number
12	KitchenAbvGrd	Number of kitchens
13	LandSlope	Slope of property
14	LowQualFinSF	Low quality finished square feet (all floors)
15	MiscFeature	Miscellaneous features not covered in other categories
16	MiscVal	\$Value of miscellaneous features
17	PavedDrive	Paved driveway
18	PoolArea	Pool area in square feet
19	PoolQc	Pool quality
20	RoofMatl	Roof material
21	ScreenPorch	Screen porch area in square feet
22	Street	Type of road access
23	Utilities	Types of utilities available

Table 2: Attributes filtered out in the 'Select Attributes' Process

Relationship between variables

Firstly, we evaluated the relationship between variables by creating the correlation matrix. We began by viewing the correlation matrix.

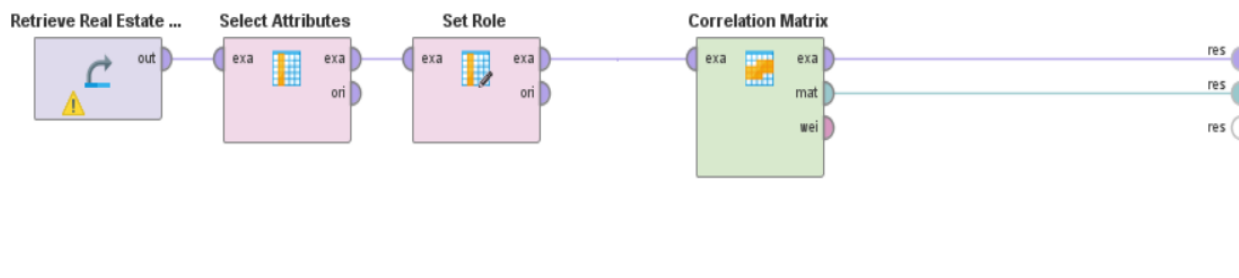


Figure 4: Correlation Matrix Process

We used RapidMiner Correlation Matrix operator to gauge the relationship strength between pairs of attributes and help us identify which variables serve as the best predictors in calculating our target variable. In Figure 4 you can see how this process was set up. We decided to remove attributes that have no impact on the final result, which are shown in Table 2. Additionally, we also removed all the nominal attributes since we can't measure their correlation.

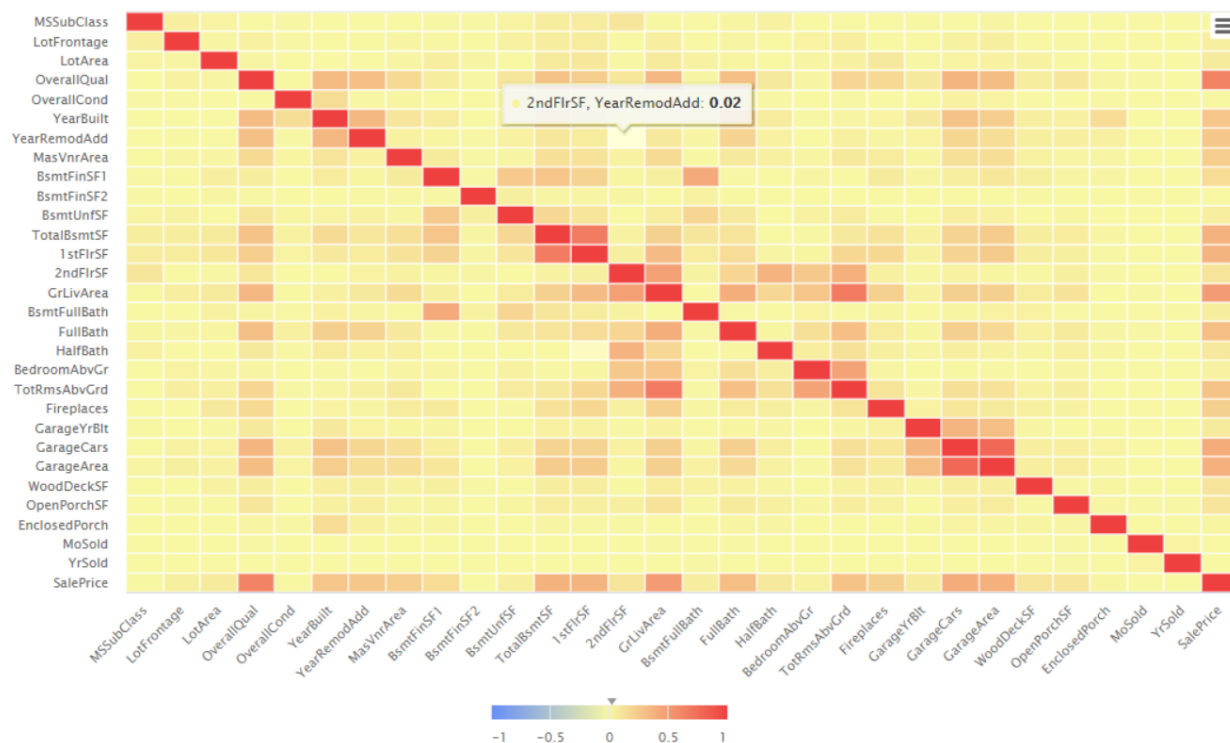


Figure 5: Correlation Matrix Heat Map

'SalePrice' vs Other Attributes

Since 'SalePrice' is our target variable, we first explored this attribute relative to others. The darker shadings represent a higher correlation, either positive or negative.

Attri... ↓	MSSub...	LotFron...	LotArea	Overall...	Overall...	YearBuilt	YearRe...	MasVnr...	BsmtFin...	BsmtFin...	BsmtUn...	TotalBs...	1stFirSF
SalePrice	0.007	0.044	0.070	0.626	0.006	0.273	0.257	0.223	0.149	0.000	0.046	0.376	0.367

2ndFirSF	GrLivAr...	BsmtFul...	FullBath	HalfBath	Bedroo...	TotRms...	Fireplac...	Garage...	Garage...	Garage...	WoodD...	OpenPo...	Enclose...
0.102	0.502	0.052	0.314	0.081	0.028	0.285	0.218	0.068	0.410	0.389	0.105	0.100	0.017

MoSold	YrSold
0.002	0.001

Figure 6: Target Variable vs. Other Attributes

According to Figure 6, the darker shades of blue represents a higher correlation which is closer to 1 or -1. We noticed that in relation to 'SalePrice', some attributes that had medium correlation were OverallQual, GrLivArea, GarageCars and GarageArea. Rest of the variables have a weak correlation. The scatterplots for the highlighted attributes in Figure 6 are as follows:

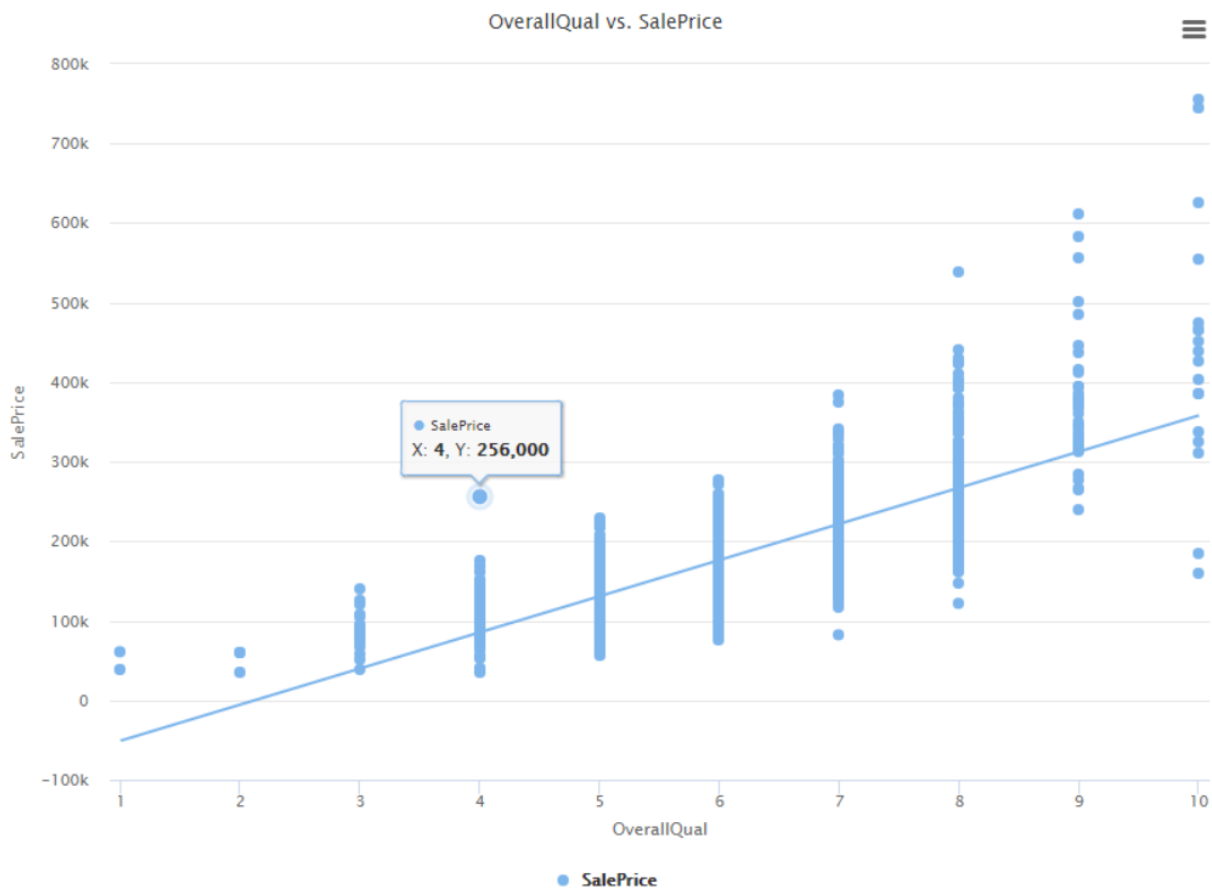


Figure 7: OverallQual vs. SalePrice

There is a positive correlation between SalePrice and OverallQual with a higher tendency for SalePrice increase when OverallQual increases, however sometimes SalePrice decreases when OverallQual increases. The correlation is not very strong.

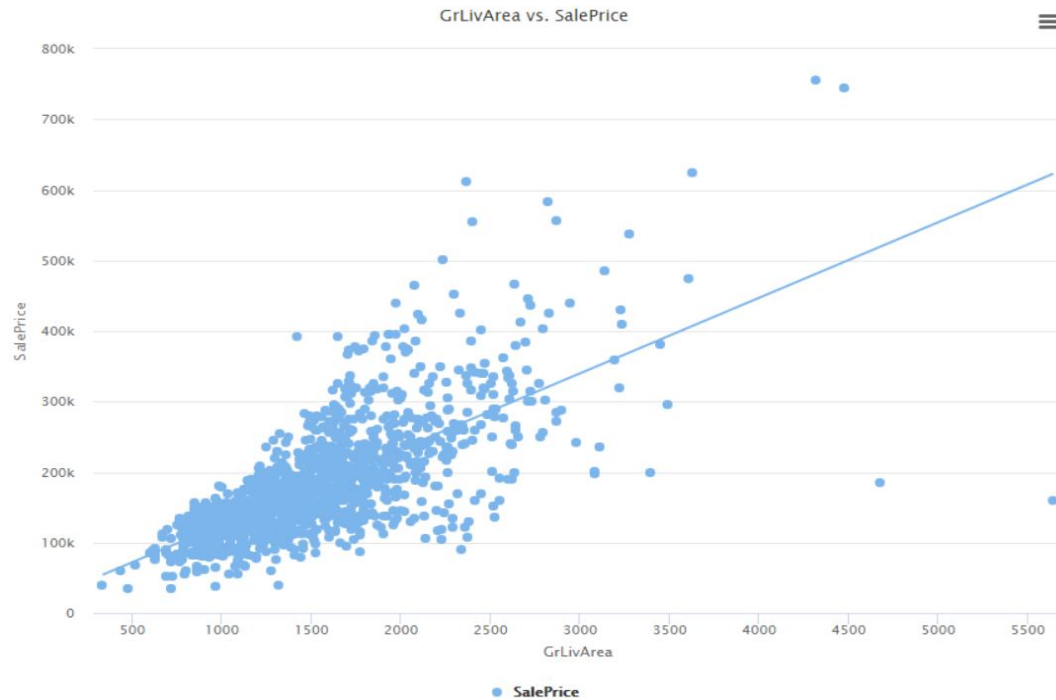


Figure 8: GrLivArea vs. SalePrice

There is a positive correlation between SalePrice and GrLivArea characterized by oval shape which tilted in positive direction. There is a tendency for SalePrice increasing when GrLivArea increases, however there are some outlier as SalePrice decreases when GrLivArea increases. The correlation is not really strong.



Figure 9: 1stFlrSF vs. SalePrice

There is a positive correlation between SalePrice and 1stflrSF. There is a tendency for SalePrice increasing when 1stflrSF increases, with few outliers. The correlation is not really strong.

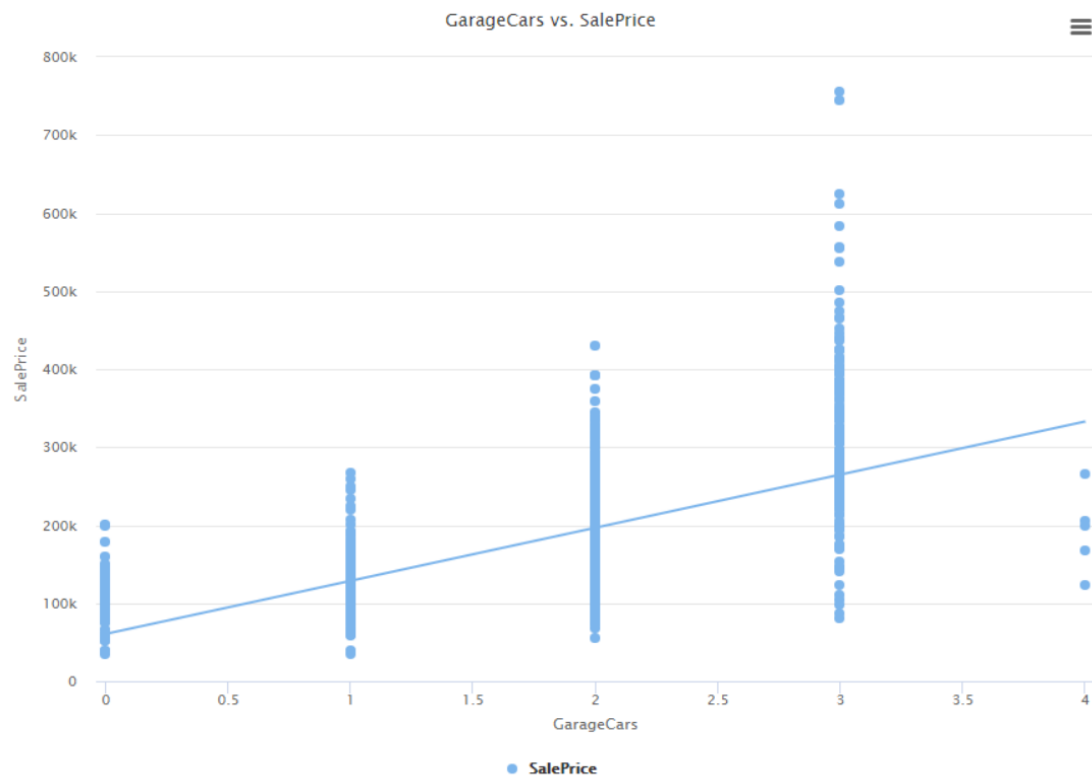


Figure 10: GarageCars vs. SalePrice

There is a positive correlation between GarageCars and SalePrice as SalePrice increases when GarageCars value increases. However after certain point SalePrice doesn't increase as the GarageCars value increases.

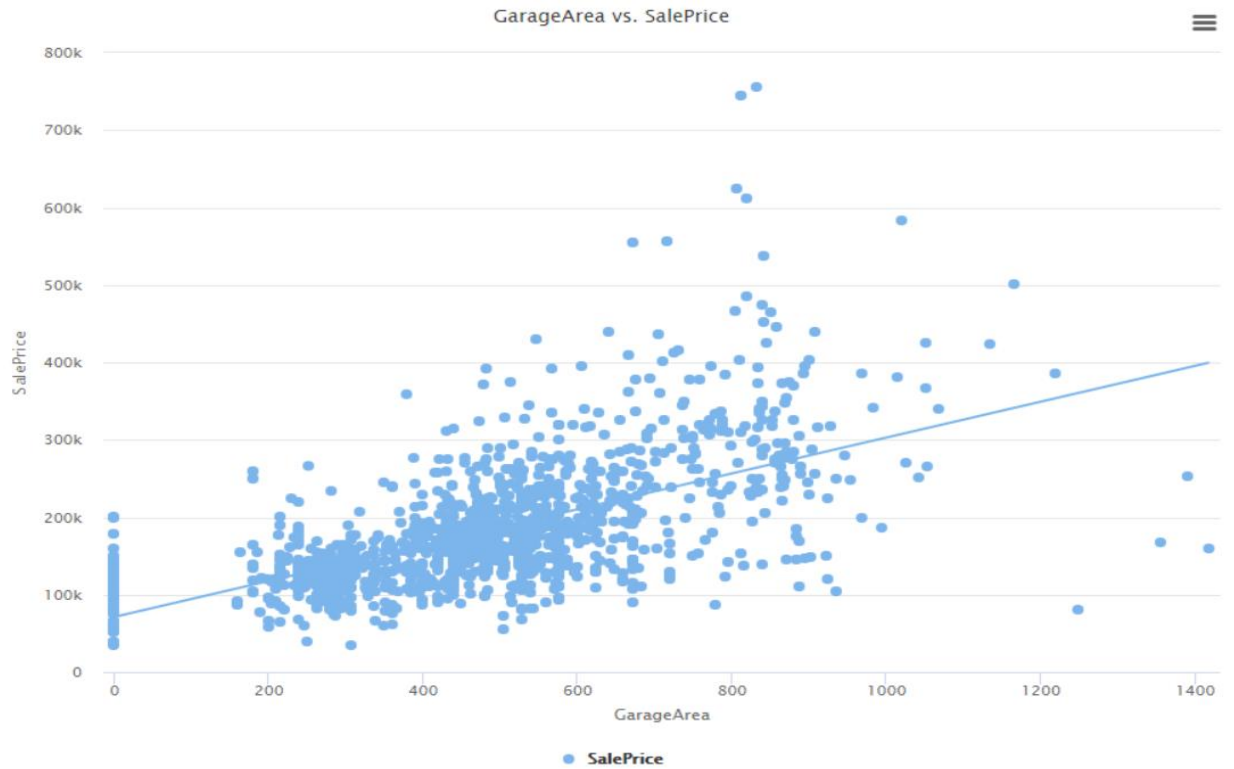


Figure 11: GarageArea vs. SalePrice

There is a positive correlation between SalePrice and GarageArea. There is a gap between 0 and 200 GarageArea, that could be because the minimal garage square footage is 200. There is a tendency for SalePrice increasing when GarageArea increases, however sometimes SalePrice decreases when GarageArea increases. The correlation is not really strong and becomes even weaker with increasing in GargeArea.

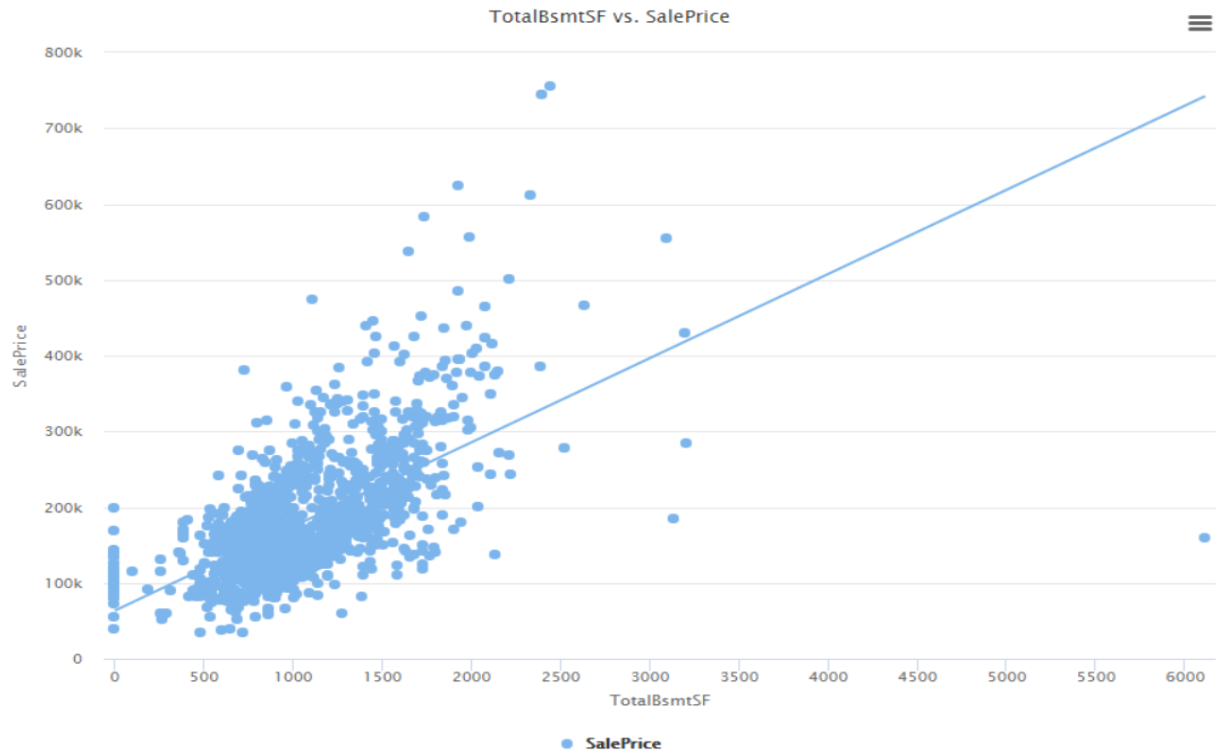


Figure 12: TotalBsmtSF vs. SalePrice

Relationship between SalePrice and TotalBsmtSF is characterized by oval shape. The oval is tilted in positive direction. There is a positive correlation between SalePrice and TotalBsmtSF. There is a tendency for SalePrice increasing when TotalBsmtSF increases, however sometimes SalePrice decreases when TotalBsmtSF increases. The correlation is not really strong.

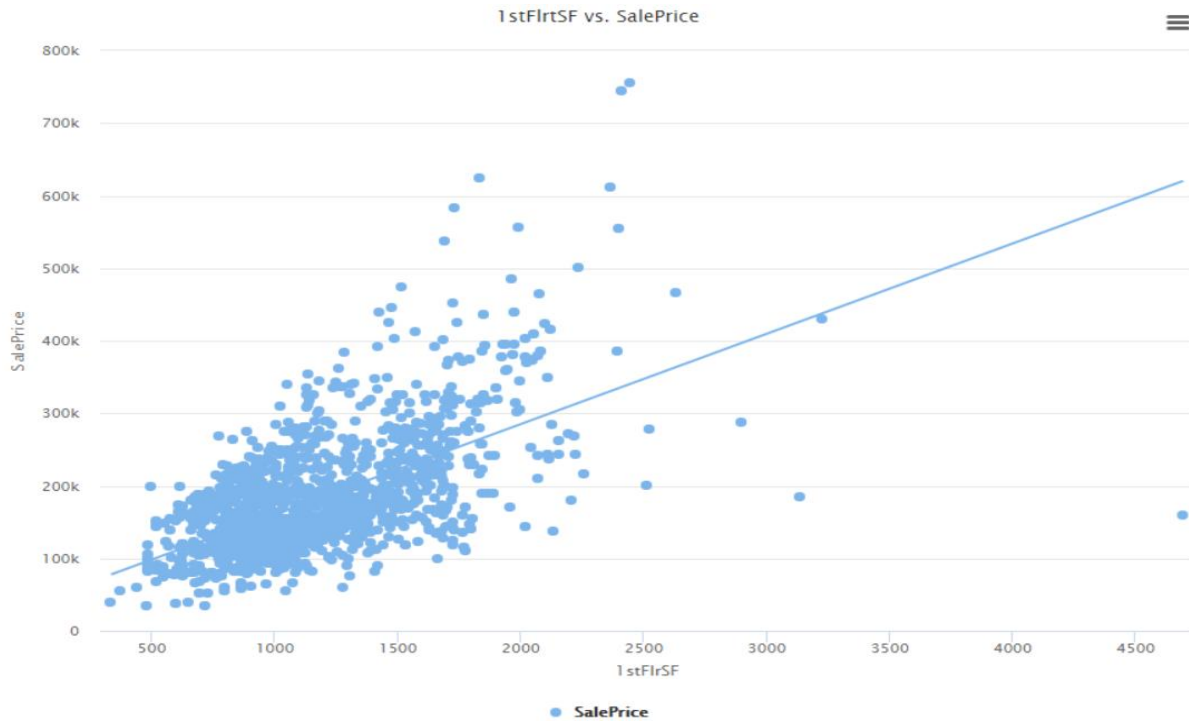


Figure 13: 1stFlrSF vs. SalePrice

There is a positive correlation between SalePrice and 1stflrSF. There is a tendency for SalePrice increasing when 1stflrSF increases, however sometimes SalePrice decreases when 1stflrSF increases. The correlation is not really strong.

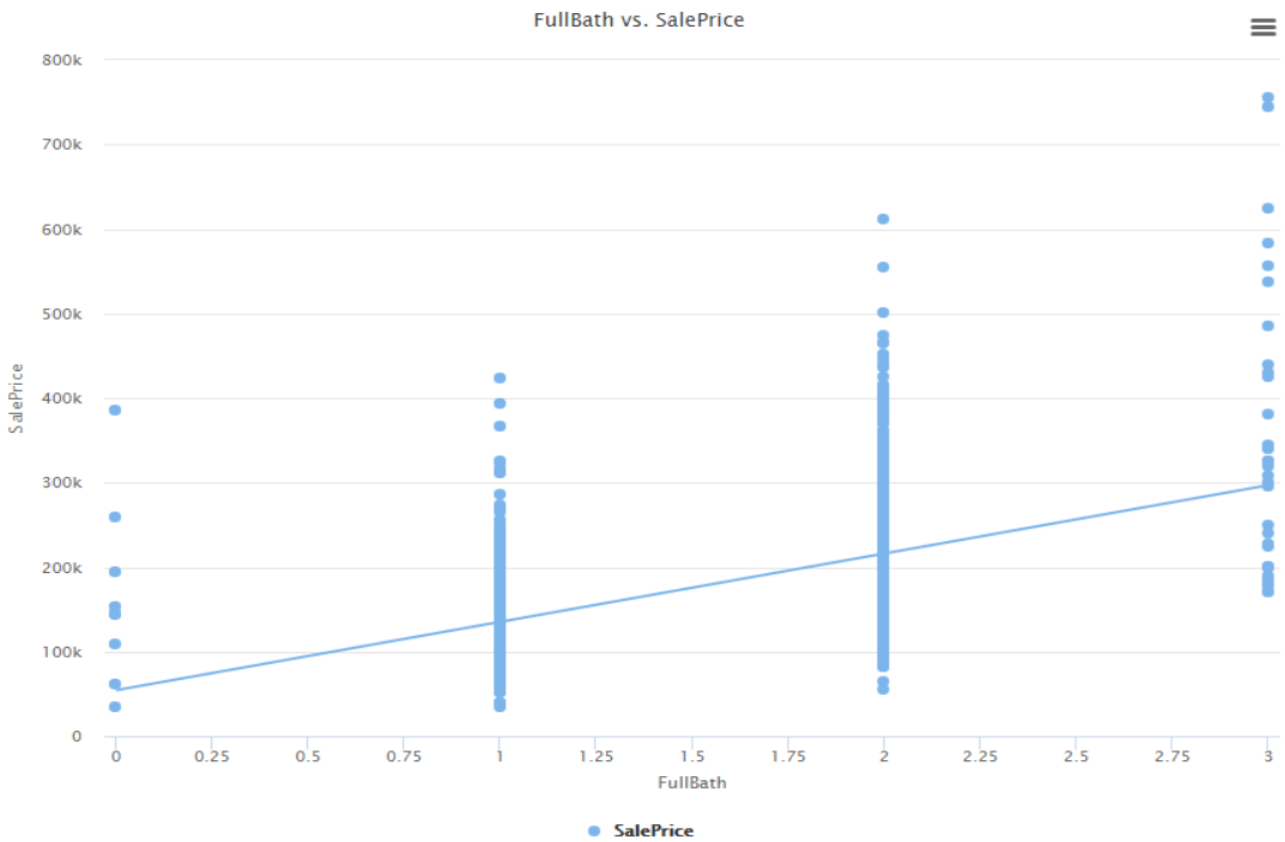


Figure 14: FullBath vs. SalePrice

There is a positive correlation between SalePrice and FullBath. There is a strong tendency for SalePrice increase between FullBath value of 1 and 2, but correlation weakens as FullBath value increases to 3.

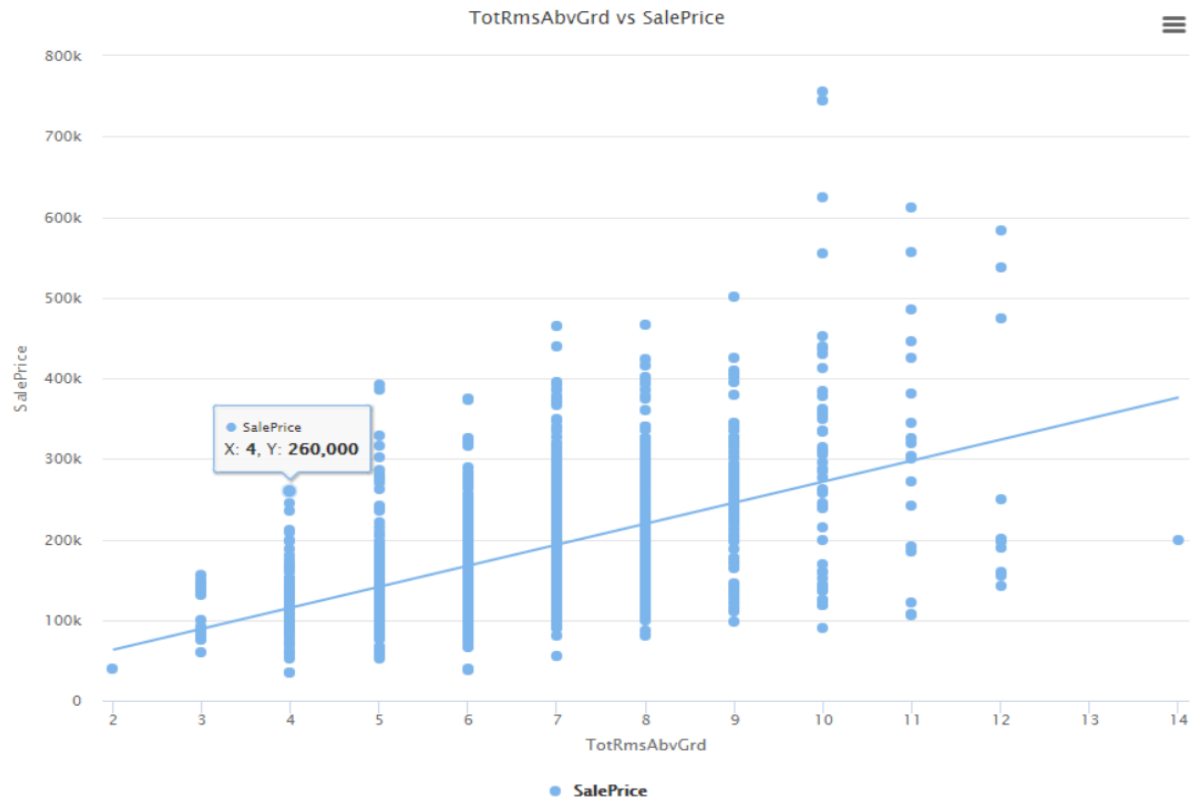


Figure 15: TotRmsAbvGrd vs. SalePrice

There is a positive correlation between SalePrice and TotRmsAbvGrd. The SalePrice increases when TotRmsAbvGrd increases, however after TotRmsAbvGrd value reaches 10, correlation starts to weaken.

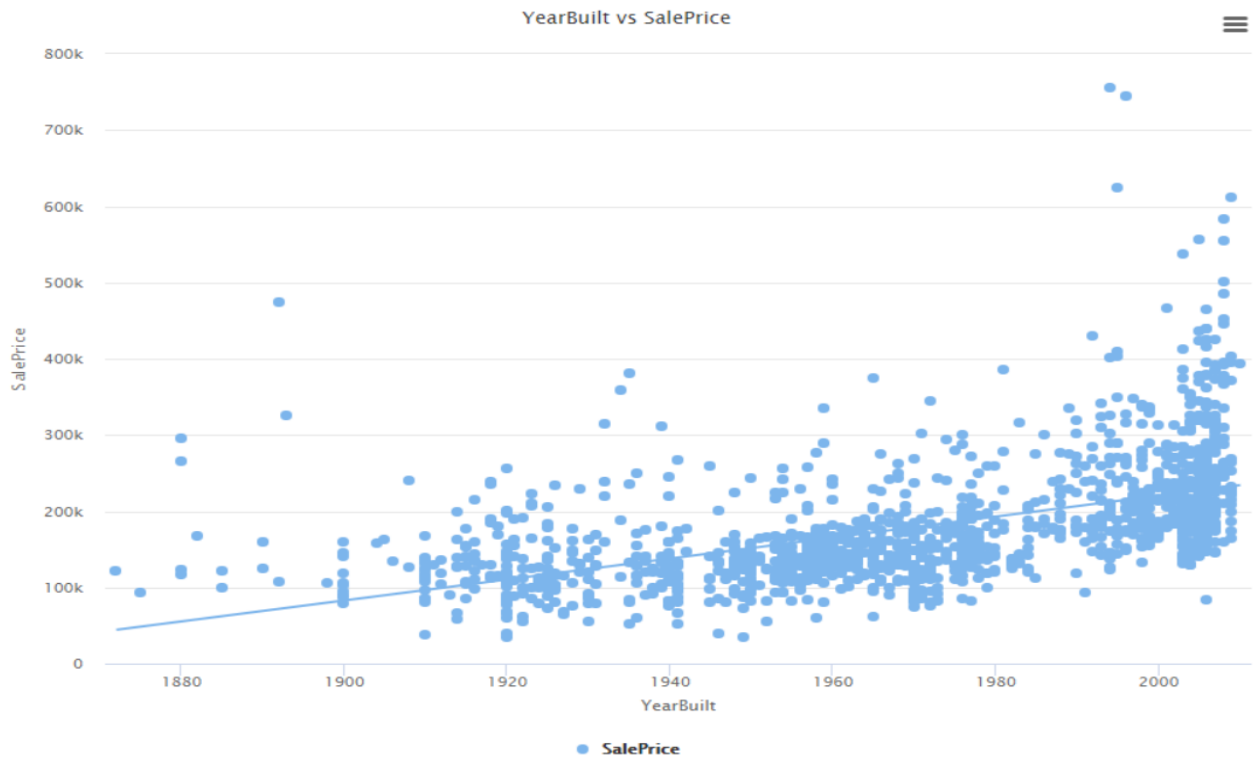


Figure 16: YearBuilt vs. SalePrice

There is a positive correlation between SalePrice and YearBuilt. The SalePrice increases when YearBuilt increases. However, the correlation is not strong as there is a lot of variation between the sale prices as the YearBuilt increases.

Outliers

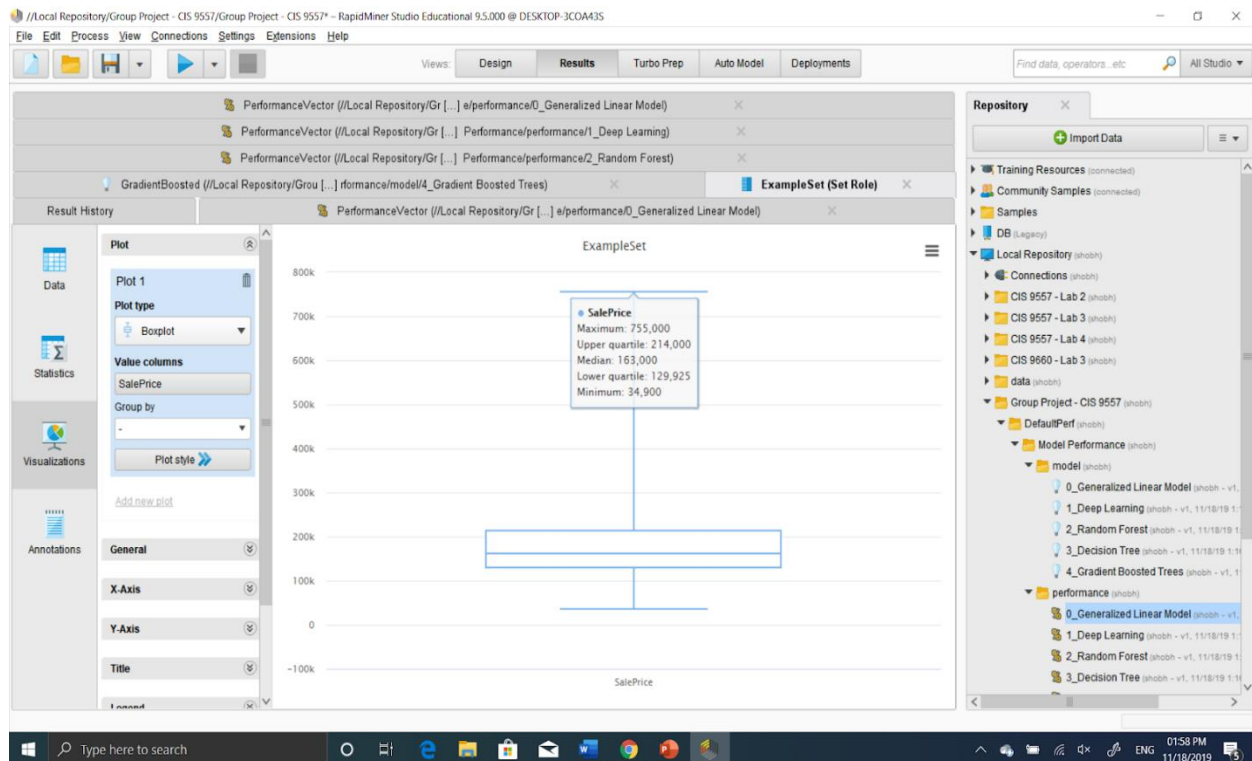


Figure 17: Boxplot Analysis of the Target Variable: 'SalePrice'

$$\text{IQR} = \text{Q3} - \text{Q1}$$

$$= 214,000 - 129,925$$

$$= 84,075$$

$$\text{Outliers Upper Bound} = \text{Q3} + (1.5 * \text{IQR})$$

$$= 214,000 + (1.5 * 84,075)$$

$$= 340,112.5$$

Thus, all houses with a 'SalePrice' above \$340,112.5 are outliers.

Baseline Model

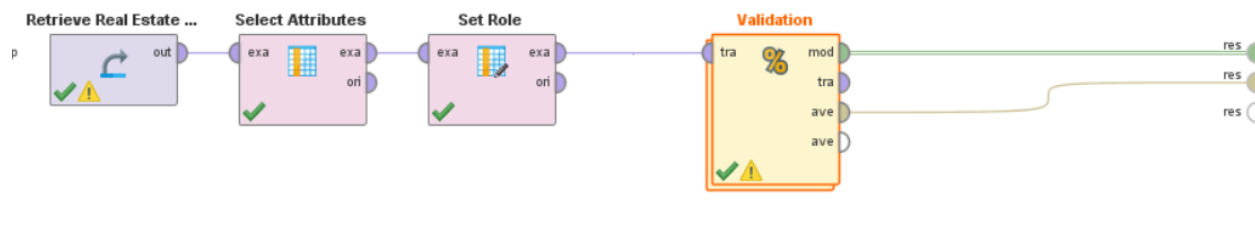


Figure 18: Baseline Model Process

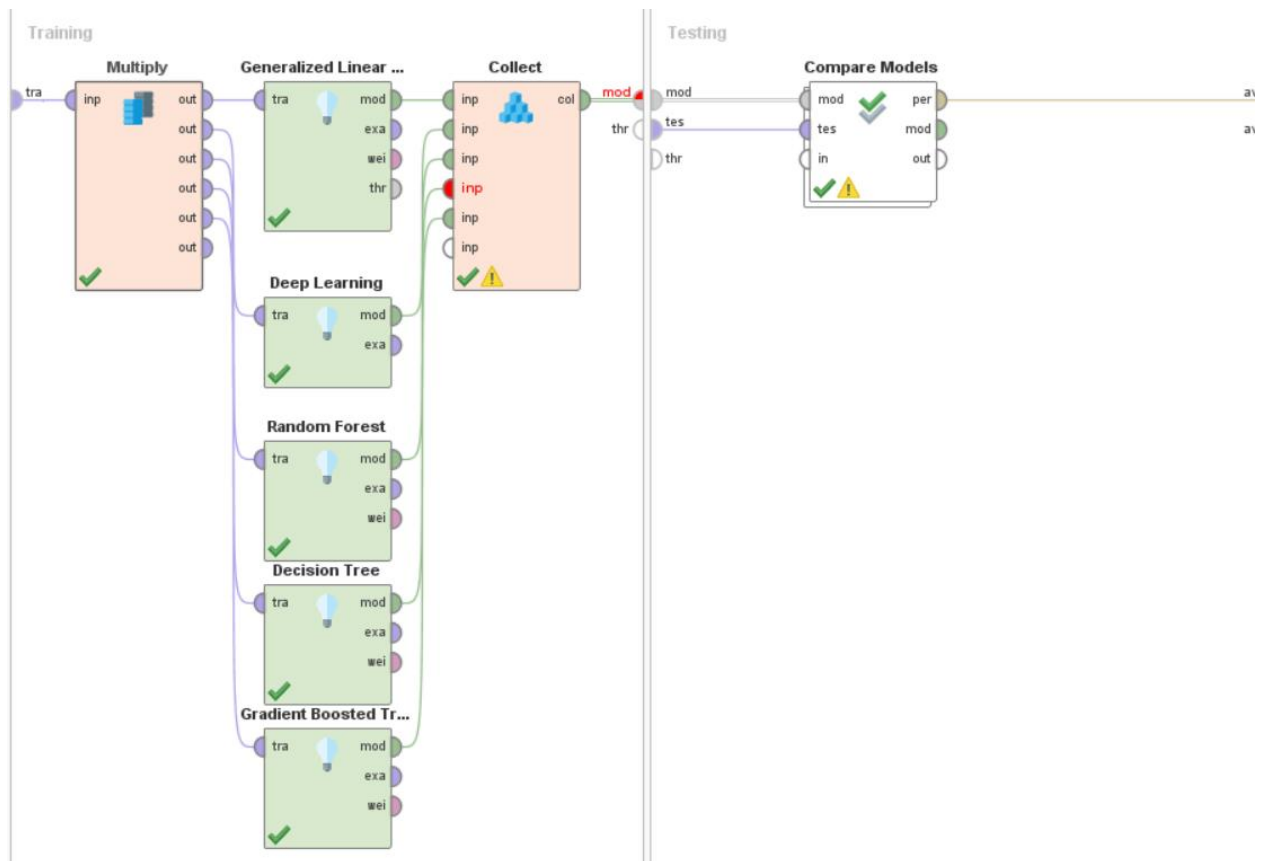


Figure 19: Baseline Model - Validation



Figure 20: Baseline Model - Compare Models

Generalized Linear Model – Split Validation – Default Parameters

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 83383.279 +/- 0.000
squared_error: 6952771223.983 +/- 23959502950.441
correlation: 0.901
squared_correlation: 0.812
prediction_average: 178462.575 +/- 83371.214
```

Figure 21: Generalized Linear Model - Model Performance - Split Validation - Default Parameters

Deep Learning – Split Validation – Default Parameters

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 28772.881 +/- 0.000
squared_error: 827878657.160 +/- 6153113026.846
correlation: 0.941
squared_correlation: 0.885
prediction_average: 178462.575 +/- 83371.214
```

Figure 22: Deep Learning - Model Performance - Split Validation - Default Parameters

Random Forest – Split Validation – Default Parameters

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 34192.757 +/- 0.000
squared_error: 1169144617.927 +/- 7418768775.373
correlation: 0.926
squared_correlation: 0.857
prediction_average: 178462.575 +/- 83371.214
```

Figure 23: Random Forest - Model Performance - Split Validation - Default Parameters

Decision Tree – Split Validation – Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 52260.019 +/- 0.000  
squared_error: 2731109591.608 +/- 13706114534.989  
correlation: 0.787  
squared_correlation: 0.620  
prediction_average: 178462.575 +/- 83371.214
```

Figure 24: Decision Tree - Model Performance - Split Validation - Default Parameters

Gradient Boosted Trees – Split Validation – Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 45893.305 +/- 0.000  
squared_error: 2106195469.084 +/- 10429740083.333  
correlation: 0.910  
squared_correlation: 0.828  
prediction_average: 178462.575 +/- 83371.214
```

Figure 25: Gradient Boosted Trees - Model Performance - Split Validation - Default Parameters

Generalized Linear Model – Cross Validation – Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 97060.309 +/- 0.000  
squared_error: 9420703498.443 +/- 37610108247.125  
correlation: 0.888  
squared_correlation: 0.788  
prediction_average: 183807.384 +/- 97072.003
```

Figure 26: Generalized Linear Model - Model Performance - Cross Validation - Default Parameters

Deep Learning – Cross Validation – Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 38826.442 +/- 0.000  
squared_error: 1507492607.144 +/- 10341044124.906  
correlation: 0.942  
squared_correlation: 0.887  
prediction_average: 183807.384 +/- 97072.003
```

Figure 27: Deep Learning - Model Performance - Cross Validation - Default Parameters

Random Forest – Cross Validation – Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 45889.443 +/- 0.000  
squared_error: 2105841016.536 +/- 13248524680.026  
correlation: 0.909  
squared_correlation: 0.827  
prediction_average: 183807.384 +/- 97072.003
```

Figure 28: Random Forest - Model Performance - Cross Validation - Default Parameters

Decision Tree – Cross Validation – Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 53195.260 +/- 0.000  
squared_error: 2829735737.775 +/- 10652274709.339  
correlation: 0.837  
squared_correlation: 0.701  
prediction_average: 183807.384 +/- 97072.003
```

Figure 29: Decision Tree - Model Performance - Cross Validation - Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 54490.170 +/- 0.000  
squared_error: 2969178653.102 +/- 16049857820.279  
correlation: 0.929  
squared_correlation: 0.863  
prediction_average: 183807.384 +/- 97072.003
```

Figure 30: Gradient Boosted Trees - Model Performance - Cross Validation - Default Parameters

Comparison of Split Validation and Cross Validation Models

The Comparison of root_mean_squared_error for both Split Validation and Cross Validation Models is as follows:

Model	Split Validation	Cross Validation
Generalized Linear Model	83383.279	97060.309
Deep Learning	28772.881	38826.442
Random Forest	34192.757	45889.443
Decision Tree	52260.019	53195.260
Gradient Boosted Trees	45893.305	54490.170

Table 3: Comparison of root mean squared error between Split Validation and Cross Validation Models

The goal of Regression Analysis is to minimize the root_mean_squared_error. Thus, as shown in Table 3, Split Validation Models achieve the best results.

The Comparison of correlation for both Split Validation and Cross Validation Models is as follows:

Model	Split Validation	Cross Validation
Generalized Linear Model	0.901	0.888
Deep Learning	0.941	0.942
Random Forest	0.926	0.909
Decision Tree	0.787	0.837
Gradient Boosted Trees	0.910	0.929

Table 4: Comparison of correlation between Split Validation and Cross Validation Models

The goal of Regression Analysis is also to maximize the correlation. As shown in Table 4, Cross Validation achieves better results for Deep Learning, Decision Tree and Gradient Boosted Trees. However, we will continue to use the Split Validation results as the base model because it performs significantly better when comparing the root_mean_squared_error and that is of greater interest for predicting 'SalePrice' of the homes in Aimes, Iowa.

Most Important Features

```

Model Metrics Type: Regression
Description: N/A
model id: rm-h2o-model-gradient_boosted_trees-159510
frame id: rm-h2o-frame-gradient_boosted_trees-335153
MSE: 1.12724838E9
R^2: 0.81297004
mean residual deviance: 1.12724838E9
Variable Importances:
  Variable      Relative Importance Scaled Importance Percentage
OverallQual 155423608930304.000000      1.000000      0.616161
Neighborhood 31690724474880.000000      0.203899      0.125635
GrLivArea 17435257208832.000000      0.112179      0.069120
1stFlrSF 6364812804096.000000      0.040951      0.025233
GarageCars 6334495326208.000000      0.040756      0.025112
TotalBsmSF 5484654886912.000000      0.035288      0.021743
GarageArea 4181399961600.000000      0.026903      0.016577
BedroomAbvGr 2944137953280.000000      0.018943      0.011672
BsmFinSF1 2319896543232.000000      0.014926      0.009197
TotRmsAbvGrd 2127474196480.000000      0.013688      0.008434
---
EnclosedPorch 9901284352.000000      0.000064      0.000039
Fence 7093223424.000000      0.000046      0.000028
LotShape 6341193216.000000      0.000041      0.000025
LotConfig 6082092544.000000      0.000039      0.000024
ExterCond 4007680256.000000      0.000026      0.000016
BsmFinType2 3385163264.000000      0.000022      0.000013
LandContour 0.000000      0.000000      0.000000
BldgType 0.000000      0.000000      0.000000
RoofStyle 0.000000      0.000000      0.000000
BsmFinSF2 0.000000      0.000000      0.000000
Model Summary:
Number of Trees Model Size in Bytes Min. Depth Max. Depth Mean Depth Min. Leaves Max. Leaves Mean Leaves
      100      95569      10      10      10.00000      69      85      77.07000

```

Figure 31: Important Features

SI	Most Important Variables	Description
1	OverallQual	Overall material and finish quality
2	Neighborhood	Physical location within Aimes city limits
3	GrLivArea	Above grade (ground) living area square feet
4	1stFlrSF	First floor square feet
5	GarageCars	Size of garage in car capacity
6	Total BsmSF	Total square feet of basement area
7	GarageArea	Size of garage in square feet
8	BedroomAbvGrd	Number of bedrooms above basement level

9	BsmtFinSF1	Type 1 finished square feet
10	TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
11	EnclosedPorch	Enclosed porch area in square feet
12	Fence	Fence quality
13	LotShape	General shape of property
14	LotConfig	Lot configuration
15	ExterCond	Present condition of the material on the exterior
16	BsmtFinType2	Quality of second finished area (if present)
17	LandContour	Flatness of the property

Table 5: Description List of Important Features

Parameter Optimization

Generalized Linear Model – Best Model Performance- Root_Mean_Squared_Error (RMSE)

SI	Parameters	RMSE	Ref.
1	Default Parameters (Figure 32)	83383.279	Fig. 33
2	Changed the Missing values handling from MeanImpute to Skip	83383.279	Fig. 33
3	Changed the Max Iterations from 0 to 5	83383.279	Fig. 33
4	Changed the lambda from 0 to 0.5	32692.868	Fig. 34
5	Changed the lambda from 0.5 to 0.75	33426.282	Fig. 35
6	Changed the lambda from 0.75 to 0.25	31788.822	Fig. 36
7	Changed the lambda from 0.25 to 0.1	30971.026	Fig. 37
8	Changed the lambda from 0.1 to 0.01	30971.026	Fig. 37
9	Changed the lambda from 0.01 to 0.001	31529.105	Fig. 38

Table 6: Generalized Linear Model - Best Model Performance - Root Mean Squared Error

Generalized Linear Model – Best Model Performance – Correlation (Corr.)

SI	Parameters	Corr.	Ref.
1	Default Parameters (Figure 32)	0.901	Fig. 33
2	Changed the Missing values handling from MeanImpute to Skip	0.901	Fig. 33
3	Changed the Max Iterations from 0 to 5	0.901	Fig. 33
4	Changed the lambda from 0 to 0.5	0.926	Fig. 34
5	Changed the lambda from 0.5 to 0.75	0.924	Fig. 35
6	Changed the lambda from 0.75 to 0.25	0.928	Fig. 36
7	Changed the lambda from 0.25 to 0.1	0.931	Fig. 37
8	Changed the lambda from 0.1 to 0.01	0.931	Fig. 37
9	Changed the lambda from 0.01 to 0.001	0.926	Fig. 38

Table 7: Generalized Linear Model - Best Model Performance - Correlation

Therefore, the best model performance is achieved when lambda is 0.01.

(Note: The reference figures are attached at the appendix)

Deep Learning – Best Model Performance – Root_Mean_Squared_Error (RMSE)

SI	Parameters	RMSE	Ref.
1	Default Parameters (Figure 40)	28772.881	Fig. 41
2	Changed the activation function from Tanh to Rectifier	31956.408	Fig. 42
3	Changed the activation function from Rectifier to Maxout	32817.658	Fig. 43
4	Changed the activation function from Maxout to ExpRectifier	29077.236	Fig. 44
5	Increased the epochs from 7 to 10	29623.893	Fig. 45
6	Decreased the epochs from 7 to 4	30769.916	Fig. 46

Table 8: Deep Learning - Best Model Performance - Root Mean Squared Error

Deep Learning – Best Model Performance – Correlation (Corr.)

SI	Parameters	Corr.	Ref.
1	Default Parameters (Figure 40)	0.941	Fig. 41
2	Changed the activation function from Tanh to Rectifier	0.928	Fig. 42
3	Changed the activation function from Rectifier to Maxout	0.920	Fig. 43
4	Changed the activation function from Maxout to ExpRectifier	0.941	Fig. 44
5	Increased the epochs from 7 to 10	0.940	Fig. 45
6	Decreased the epochs from 7 to 4	0.944	Fig. 46

Table 9: Deep Learning - Best Model Performance - Correlation

Thus, the Tanh activation function has the best model performance because it has the highest correlation and the lowest root_mean_squared_error and we will use that as the base model for further optimization. We will continue to use 7 epochs as the root_mean_squared_error is lower than when epochs is 4 and there isn't any drastic increase between the correlations.

Random Forest – Best Model Performance – Root_Mean_Squared_Error (RMSE)

SI	Parameters	RMSE	Ref.
1	Default Parameters (Figure 47)	34192.757	Fig. 48
2	Changed the maximal depth from 10 to 15	34132.495	Fig. 49
3	Changed the maximal depth from 10 to 7	34685.970	Fig. 50
4	Changed the number of trees from 100 to 125	34324.423	Fig. 51
5	Changed the number of trees from 100 to 75	33920.910	Fig. 52

Table 10: Random Forest - Best Model Performance - Root_Mean_Squared_Error

Random Forest – Best Model Performance – Correlation (Corr.)

SI	Parameters	Corr.	Ref.
1	Default Parameters (Figure 47)	0.926	Fig. 48
2	Changed the maximal depth from 10 to 15	0.926	Fig. 49
3	Changed the maximal depth from 10 to 7	0.925	Fig. 50
4	Changed the number of trees from 100 to 125	0.926	Fig. 51
5	Changed the number of trees from 100 to 75	0.926	Fig. 52

Table 11: Random Forest - Best Model Performance - Correlation

Thus, after reviewing Figures 48 – 52, we decided to use the Random Forest Model with a maximal depth of 10 and 75 number of trees.

Decision Tree – Best Model Performance – Root_Mean_Squared_Error (RMSE)

SI	Parameters	RMSE	Ref.
1	Default Parameters (Figure 53)	53195.260	Fig. 54
2	Disabled Apply Prepruning button with everything else remaining the same	53351.584	Fig. 55
3	Changed the maximal depth from 10 to 15	52242.401	Fig. 56
4	Changed the maximal depth from 10 to 20	52242.748	Fig. 57
5	Changed the minimal gain from 0.01 to 0.02	52242.401	Fig. 57
6	Changed the minimal gain from 0.01 to 0.005	52242.401	Fig. 57
7	Changed the minimal leaf size from 2 to 5	36510.800	Fig. 58
8	Changed the minimal leaf size from 5 to 8	41237.062	Fig. 59
9	Changed the minimal size for split from 4 to 5	36510.800	Fig. 58
10	Changed the minimal size for split from 4 to 3	36510.800	Fig. 58
11	Changed the number of prepruning alternatives from 3 to 4	36499.823	Fig. 60
12	Changed the number of prepruning alternatives from 4 to 2	37782.179	Fig. 61
13	Changed the number of prepruning alternatives from 4 to 5	36273.217	Fig. 62
14	Changed the number of prepruning alternatives from 5 to 6	36273.155	Fig. 63
15	Changed the number of prepruning alternatives from 6 to 7	36451.711	Fig. 64

Table 12: Decision Tree - Best Model Performance - Root Mean Squared Error

Decision Tree – Best Model Performance – Correlation (Corr.)

SI	Parameters	Corr.	Ref.
1	Default Parameters (Figure 53)	0.837	Fig. 54
2	Disabled Apply Prepruning button with everything else remaining the same	0.789	Fig. 55
3	Changed the maximal depth from 10 to 15	0.787	Fig. 56
4	Changed the maximal depth from 10 to 20	0.787	Fig. 57
5	Changed the minimal gain from 0.01 to 0.02	0.787	Fig. 57
6	Changed the minimal gain from 0.01 to 0.005	0.787	Fig. 57
7	Changed the minimal leaf size from 2 to 5	0.900	Fig. 58
8	Changed the minimal leaf size from 5 to 8	0.872	Fig. 59
9	Changed the minimal size for split from 4 to 5	0.900	Fig. 58
10	Changed the minimal size for split from 4 to 3	0.900	Fig. 58
11	Changed the number of prepruning alternatives from 3 to 4	0.900	Fig. 60
12	Changed the number of prepruning alternatives from 4 to 2	0.893	Fig. 61
13	Changed the number of prepruning alternatives from 4 to 5	0.901	Fig. 62
14	Changed the number of prepruning alternatives from 5 to 6	0.901	Fig. 63
15	Changed the number of prepruning alternatives from 6 to 7	0.900	Fig. 64

Table 13: Decision Tree - Best Model Performance - Correlation

Thus, after reviewing figures 54-64, we decided to use the Decision tree with maximal depth of 15, minimal gain of 0.01, minimal leaf size of 5, minimal size for split of 4 and number of prepruning alternatives of 6.

Gradient Boosted Trees – Best Model Performance - Root_Mean _Squared_Error (RMSE)

SI	Parameters	RMSE	Ref.
1	Default Parameters (Figure 65)	45893.305	Fig. 66
2	Increased the number of trees from 100 to 200	35516.679	Fig. 67
3	Increased the number of trees from 200 to 300	32008.256	Fig. 68
4	Increased the number of trees from 300 to 400	30866.547	Fig. 69
5	Increased the number of trees from 400 to 500	30650.179	Fig. 70
6	Increased the number of trees from 500 to 600	30460.651	Fig. 71
7	Increased the number of trees from 600 to 700	30325.758	Fig. 72
8	Increased the number of trees from 700 to 800	30253.130	Fig. 73
9	Increased the number of trees from 800 to 900	30215.662	Fig. 74
10	Increased the number of trees from 900 to 1000	30199.711	Fig. 75
11	Increased the number of trees from 1000 to 1100	30196.595	Fig. 76
12	Increased the number of trees from 1100 to 1200	30196.268	Fig. 77
13	Increased the number of trees from 1200 to 1300	30193.040	Fig. 78
14	Increased the number of trees from 1300 to 1400	30203.444	Fig. 79
15	Decreased the number of trees from 1300 to 1250	30192.611	Fig. 80
16	Increased the maximal depth from 10 to 15	30136.040	Fig. 81
17	Increased the maximal depth from 15 to 20	30157.010	Fig. 82

Table 14: Gradient Boosted Trees - Best Model Performance - Root Mean Squared Error

Gradient Boosted Trees – Best Model Performance – Correlation (Corr.)

SI	Parameters	Corr	Ref.
1	Default Parameters (Figure 65)	0.910	Fig. 66
2	Increased the number of trees from 100 to 200	0.921	Fig. 67
3	Increased the number of trees from 200 to 300	0.929	Fig. 68
4	Increased the number of trees from 300 to 400	0.932	Fig. 69
5	Increased the number of trees from 400 to 500	0.932	Fig. 70
6	Increased the number of trees from 500 to 600	0.933	Fig. 71
7	Increased the number of trees from 600 to 700	0.933	Fig. 72
8	Increased the number of trees from 700 to 800	0.934	Fig. 73
9	Increased the number of trees from 800 to 900	0.934	Fig. 74
10	Increased the number of trees from 900 to 1000	0.934	Fig. 75
11	Increased the number of trees from 1000 to 1100	0.934	Fig. 76
12	Increased the number of trees from 1100 to 1200	0.934	Fig. 77
13	Increased the number of trees from 1200 to 1300	0.934	Fig. 78
14	Increased the number of trees from 1300 to 1400	0.934	Fig. 79
15	Decreased the number of trees from 1300 to 1250	0.934	Fig. 80
16	Increased the maximal depth from 10 to 15	0.934	Fig. 81
17	Increased the maximal depth from 15 to 20	0.934	Fig. 82

Table 15: Gradient Boosted Trees - Best Model Performance - Correlation

Thus, after reviewing Figures 66-82, we decided to use the Gradient Boosted Trees model with 1250 number of trees and maximal depth of 15.

Model Building

Based on Tables 6-15, the best models are as follows:

SI	Models	RMSE	Ref.
1	Generalized Linear Model	30971.026	Fig. 37
2	Deep Learning	28772.881	Fig. 41
3	Random Forest	33920.910	Fig. 52
4	Decision Tree	36273.155	Fig. 63
5	Gradient Boosted Trees	30136.040	Fig. 81

Table 16: Comparison of Best Models - RMSE

SI	Models	Corr.	Ref.
1	Generalized Linear Model	0.931	Fig. 37
2	Deep Learning	0.941	Fig. 41
3	Random Forest	0.926	Fig. 52
4	Decision Tree	0.901	Fig. 63
5	Gradient Boosted Trees	0.934	Fig. 81

Table 17: Comparison of Best Models - Correlation

Based on Tables 16 – 17, the best models are Deep Learning, Gradient Boosted Trees and Generalized Linear Model. As shown in Table 8, we couldn't improve the model any further. The highest model improvement was done for Generalized Linear Model as we were able decrease the RMSE from 83383.279 to 30971.026. The highest model improvement on the basis of correlation was achieved by Decision Tree, from 0.837 to 0.901.

Analysis and Recommendations

Our analysis focused on understanding the top characteristics that factored into the selling price of real estate in Ames, Iowa between 2006 and 2010. Our general assumptions *prior* to exploring the data and building a prediction model was that the primary factors in determining the selling price of a property was the age of the property (year built), its overall condition, size, and neighborhood. Another initial assumption, which our analysis confirmed, was that a higher number of sales occurred during the middle part of the year.

Early exploration of the data indicated that this was only partly true. Specifically, amongst the top predictors of the price of real estate was the overall quality of the property, *not* the age of the property, followed by neighborhood, the size of the general living area (defined as the area above the ground). Also, buyers seemingly factored in characteristics of the garage when ultimately agreeing on a final price. Through this analysis, we were able to identify which characteristics of a house were most significant in determining the value that buyers place on residential properties. The results of our regression analysis revealed that there were some similarities and some differences regarding which criteria were significant in determination of sales price. Attributes such as OverallQual, GrLivArea, GarageCars and Garage Area have a strong correlation with SalePrice.

The exploratory analysis showed a high correlation between the sales price and the overall quality. An explanation for the higher correlation between the sales price and overall quality versus the lower correlation between the sales price and overall condition could lie in what information the attributes, *overall quality* and *overall condition*, were capturing. In other words, the overall quality of the property was defined in terms of the condition of the materials and finishes used. Poor overall quality could mean that the buyers would have to invest in making upgrades themselves after closing. The overall condition attribute pertained to the general condition of the house. Further exploration could be done in this space to better understand the nuanced differences between these two attributes as both were measured on a scale from 1-10.

Residents of Ames prefer floor level houses for the most part and they will decide over a house with basement only depending on the area in square feet of it. In addition, they do care about garage areas and capacity when paying for a house since most of residents own at least one car.

Due to heterogeneity, predicting real estate prices is not an easy task, therefore we came up with different models that give us the best performance for our target variable 'SalePrice'. Models such as Generalized Linear Model, Deep Learning, Decision Tree, Gradient Boosted Trees and Random Forest provided good insights to take into consideration to solve this business problem.

Based on these notable results, we recommend the following:

- To use the Deep Learning Model, Gradient Boosted Trees and Generalized Linear Model with split validation 70/30 to predict the real estate prices.
- To run a new process with updated information (current years) to compare the trend our target variable has followed throughout these years.
- To analyze economic shocks that may have occurred in that period (2006-2010) to filter out the probability that unexpected events had affected the real estate economy in Ames, Iowa and thus, had influenced the performance of the predicted variables.

Appendix


House Prices Dataset Attributes Description

SI	Attributes	Description
1	SalePrice	The property's sale price in dollars. This is the target variable that we are trying to predict
2	MSSubClass	The building class
3	MSZoning	The general zoning classification
4	LotFrontage	Linear feet of street connected to property
5	LotArea	Lot size in square feet
6	Street	Type of road access
7	Alley	Type of alley access
8	LotShape	General shape of property
9	LandContour	Flatness of the property
10	Utilities	Type of utilities available
11	LotConfig	Lot configuration
12	LandSlope	Slope of property
13	Neighborhood	Physical location within Ames city limits
14	Condition1	Proximity to main road or railroad
15	Condition2	Proximity to main road or railroad (if a second is present)
16	BldgType	Type of dwelling
17	HouseStyle	Style of dwelling
18	OverallQual	Overall material and finish quality
19	OverallCond	Overall condition rating
20	YearBuilt	Original construction date
21	YearRemodAdd	Remodel date
22	RoofStyle	Type of roof
23	RoofMatl	Roof material
24	Exterior1st	Exterior covering on house
25	Exterior2nd	Exterior covering on house (if more than one material)
26	MasVnrType	Masonry veneer type
27	MasVnrArea	Masonry veneer area in square feet
28	ExterQual	Exterior material quality
29	ExterCond	Present condition of the material on the exterior
30	Foundation	Type of foundation
31	BsmtQual	Height of the basement
32	BsmtCond	General condition of the basement
33	BsmtExposure	Walkout or garden level basement walls
34	BsmtFinType1	Quality of basement finished area
35	BsmtFinSF1	Type 1 finished square feet
36	BsmtFinType2	Quality of second finished area (if present)
37	BsmtFinSF2	Type 2 finished square feet
38	BsmtUnfSF	Unfinished square feet of basement area
39	TotalBsmtSF	Total square feet of basement area
40	Heating	Type of heating
41	HeatingQC	Heating quality and condition

42	CentralAir	Central air conditioning
43	Electrical	Electrical system
44	1stFlrSF	First floor square feet
45	2ndFlrSF	Second floor square feet
46	LowQualFinSF	Low quality finished square feet (all floors)
47	GrLivArea	Above grade (ground) living area square feet
48	BsmtFullBath	Basement full bathrooms
49	BsmtHalfBath	Basement half bathrooms
50	FullBath	Full bathrooms above grade
51	HalfBath	Half bathrooms above grade
52	Bedroom	Number of bedrooms above basement level
53	KitchenAbvGrd	Number of kitchens
54	KitchenQual	Kitchen quality
55	TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
56	Functional	Home functionality rating
57	Fireplaces	Number of fireplaces
58	FireplaceQu	Fireplace quality
59	GarageType	GarageLocation
60	GarageYearBlt	Year garage was built
61	GarageFinish	Interior finish of the garage
62	GarageCars	Size of garage in car capacity
63	GarageArea	Size of garage in square feet
64	GarageQual	Garage quality
65	GarageCond	Garage condition
66	PavedDrive	Paved driveway
67	WoodDeckSF	Wood deck area in square feet
68	OpenPorchSF	Open porch area in square feet
69	EnclosedPorch	Enclosed porch area in square feet
70	3SsnPorch	Three season porch area in square feet
71	ScreenPorch	Screen porch area in square feet
72	PoolArea	Pool area in square feet
73	PoolQC	Pool quality
74	Fence	Fence quality
75	MiscFeature	Miscellaneous feature not covered in other categories
76	MiscVal	\$Value of miscellaneous feature
77	MoSold	MonthSold
78	YrSold	Year Sold
79	SaleType	Type of Sale
80	SaleCondition	Condition of Sale

Generalized Linear Model – Default Parameters

Parameters ✕

 **Generalized Linear Model**

family ⓘ

solver ⓘ

☐ reproducible ⓘ

☒ use regularization ⓘ

lambda ⓘ

☐ lambda search ⓘ

alpha ⓘ

☒ standardize ⓘ

☐ non-negative coefficients ⓘ

☒ add intercept ⓘ

☐ remove collinear columns ⓘ

missing values handli... ⓘ

max iterations ⓘ

☐ specify beta constraints ⓘ

Figure 32: Generalized Linear Model - Default Parameters

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 83383.279 +/- 0.000
squared_error: 6952771223.983 +/- 23959502950.441
correlation: 0.901
squared_correlation: 0.812
prediction_average: 178462.575 +/- 83371.214
```

Figure 33: Generalized Linear Model - Model Performance - Default Parameters

Generalized Linear Model – Changed the Missing values handling from MeanImpute to Skip

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 83383.279 +/- 0.000
squared_error: 6952771223.983 +/- 23959502950.441
correlation: 0.901
squared_correlation: 0.812
prediction_average: 178462.575 +/- 83371.214
```

Figure 34: Generalized Linear Model - Model Performance - Changed the Missing values handling from MeanImpute to Skip

Same performance as the Default Parameters.

Generalized Linear Model – Changed the Max Iterations from 0 to 5.
Same performance as the Default Parameters.

Generalized Linear Model – Changed the lambda from 0 to 0.5.

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 32692.868 +/- 0.000
squared_error: 1068823606.080 +/- 7263001246.832
correlation: 0.926
squared_correlation: 0.857
prediction_average: 178462.575 +/- 83371.214
```

Figure 35: Generalized Linear Model - Model Performance - Changed the lambda from 0 to 0.5

The correlation has increased from 0.901 to 0.926 and the root_mean_squared_error has decreased from 83383.279 to 32692.868, which indicates that the spread around the line of best fit has decreased. Our aim is to decrease the Root Mean Squared Error and increase the correlation for the model.

Generalized Linear Model – Changed the lambda from 0.5 to 0.75.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 33426.282 +/- 0.000  
squared_error: 1117316361.314 +/- 7571764964.596  
correlation: 0.924  
squared_correlation: 0.853  
prediction_average: 178462.575 +/- 83371.214
```

Figure 36: Generalized Linear Model - Model Performance - Changed the lambda from 0.5 to 0.75

The correlation has decreased from 0.926 to 0.924 and the root_mean_squared_error has increased from 32692.868 to 33426.282.

Generalized Linear Model – Changed the lambda from 0.75 to 0.25

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 31788.822 +/- 0.000  
squared_error: 1010529215.478 +/- 6853703719.401  
correlation: 0.928  
squared_correlation: 0.862  
prediction_average: 178462.575 +/- 83371.214
```

Figure 37: Generalized Linear Model - Model Performance - Changed the lambda from 0.75 to 0.25

The correlation has increased from 0.924 to 0.928 and the root_mean_squared_error has decreased from 33426.282 to 31788.822. Thus, the general trend is as the lambda is decreasing the model performance is also increasing.

Generalized Linear Model – Changed the lambda from 0.25 to 0.1

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30971.026 +/- 0.000  
squared_error: 959204457.275 +/- 6439525781.684  
correlation: 0.931  
squared_correlation: 0.866  
prediction_average: 178462.575 +/- 83371.214
```

Figure 38: Generalized Linear Model - Model Performance - Changed the lambda from 0.25 to 0.1

The correlation has increased from 0.928 to 0.931 and the root_mean_squared_error has decreased from 31788.822 to 30971.026.

Generalized Linear Model – Changed the lambda from 0.1 to 0.01

The model performance has remained the same. Thus, it can be inferred that after a certain point the model stops improving.

Generalized Linear Model – Changed the lambda from 0.1 to 0.001

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 31529.105 +/- 0.000  
squared_error: 994084445.264 +/- 5699748168.910  
correlation: 0.926  
squared_correlation: 0.857  
prediction_average: 178462.575 +/- 83371.214
```

Figure 39: Generalized Linear Model - Model Performance - Changed the lambda from 0.1 to 0.001

The correlation has decreased from 0.931 to 0.926 and the root_mean_squared_error has increased from 30971.026 to 31529.105. **Therefore, the highest model performance is achieved when lambda is 0.01.**

Deep Learning – Default Parameters

Parameters [X]

Deep Learning

activation: Tanh [i]

hidden layer sizes: [Edit Enumeration...] [i]

☒ reproducible (uses 1 thread) [i]

☒ use local random seed [i]

local random seed: 1992 [i]

epochs: 7.0 [i]

☒ compute variable importances [i]

[] [i]

[Hide advanced parameters](#)

[Change compatibility \(9.3.001\)](#)

Parameters [X]

Deep Learning

☒ adaptive rate [i]

epsilon: 1.0E-8 [i]

rho: 0.99 [i]

☒ standardize [i]

L1: 1.0E-5 [i]

L2: 0.0 [i]

max w2: 10.0 [i]

[] [i]

[Hide advanced parameters](#)

[Change compatibility \(9.3.001\)](#)

Figure 40: Deep Learning - Default Parameters

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 28772.881 +/- 0.000
squared_error: 827878657.160 +/- 6153113026.846
correlation: 0.941
squared_correlation: 0.885
prediction_average: 178462.575 +/- 83371.214
```

Figure 41: Deep Learning - Model Performance - Default Parameters

Deep Learning – Changed the activation function from Tanh to Rectifier

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 31956.408 +/- 0.000
squared_error: 1021212031.667 +/- 7503567891.966
correlation: 0.928
squared_correlation: 0.862
prediction_average: 178462.575 +/- 83371.214
```

Figure 42: Deep Learning - Model Performance - Default Parameters - Changed the activation function from Tanh to Rectifier

The correlation has decreased from 0.941 to 0.928 and the root_mean_squared_error has increased from 28772.881 to 31956.408.

Deep Learning – Changed the activation function from Rectifier to Maxout

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 32817.658 +/- 0.000
squared_error: 1076998660.374 +/- 7483265266.715
correlation: 0.920
squared_correlation: 0.847
prediction_average: 178462.575 +/- 83371.214
```

Figure 43: Deep Learning - Model Performance - Default Parameters - Changed the activation function from Rectifier to Maxout

The correlation has decreased from 0.928 to 0.920 and the root_mean_squared_error has increased from 31956.408 to 32817.658.

Deep Learning – Changed the activation function from Maxout to ExpRectifier

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 29077.236 +/- 0.000  
squared_error: 845485658.557 +/- 6246664144.599  
correlation: 0.941  
squared_correlation: 0.885  
prediction_average: 178462.575 +/- 83371.214
```

Figure 44: Deep Learning - Model Performance - Default Parameters - Changed the activation function from Maxout to ExpRectifier

The correlation has increased from 0.920 to 0.941 and the root_mean_squared_error has decreased from 32817.658 to 29077.236. **Thus, the Tanh activation function has the best model performance because it has the highest correlation and the lowest root_mean_squared_error and we will use that as the base model for further optimization.**

Deep Learning – Increased the epochs from 7 to 10

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 29623.893 +/- 0.000  
squared_error: 877575064.760 +/- 5308675234.722  
correlation: 0.940  
squared_correlation: 0.884  
prediction_average: 178462.575 +/- 83371.214
```

Figure 45: Deep Learning - Model Performance - Default Parameters - Increased the epochs from 7 to 10

The correlation has decreased from 0.941 to 0.940 and the root_mean_squared_error has increased from 28772.881 to 29623.893.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30769.916 +/- 0.000  
squared_error: 946787734.071 +/- 7071262480.377  
correlation: 0.944  
squared_correlation: 0.892  
prediction_average: 178462.575 +/- 83371.214
```

Figure 46: Deep Learning - Model Performance - Default Parameters - Increased the epochs from 7 to 4

The correlation has increased from 0.941 to 0.944 and the root_mean_squared_error has increased from 27904.018 to 30769.916. **We will continue to use 7 epochs as the root_mean_squared_error is lower than when epochs is 4 and there isn't any drastic increase between the correlations.**

Random Forest – Default Parameters

Parameters

Random Forest

number of trees

100

criterion

least_square

maximal depth

10

☐ apply prepruning

☐ random splits

☒ guess subset ratio

☐ use local random seed

[Hide advanced parameters](#)

[Change compatibility \(9.5.001\)](#)

Figure 47: Random Forest - Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 34192.757 +/- 0.000  
squared_error: 1169144617.927 +/- 7418768775.373  
correlation: 0.926  
squared_correlation: 0.857  
prediction_average: 178462.575 +/- 83371.214
```

Figure 48: Random Forest - Model Performance - Default Parameters

Random Forest – Changed the maximal depth from 10 to 15

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 34132.495 +/- 0.000  
squared_error: 1165027230.861 +/- 7409470914.878  
correlation: 0.926  
squared_correlation: 0.857  
prediction_average: 178462.575 +/- 83371.214
```

Figure 49: Random Forest - Model Performance - Changed the maximal depth from 10 to 15

There is no change in the correlation, but the root_mean_squared_error has decreased slightly from 34192.757 to 34132.495.

Random Forest – Changed the maximal depth from 10 to 7

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 34685.970 +/- 0.000  
squared_error: 1203116484.219 +/- 7525158491.993  
correlation: 0.925  
squared_correlation: 0.856  
prediction_average: 178462.575 +/- 83371.214
```

Figure 50: Random Forest - Model Performance - Changed the maximal depth from 10 to 7

The correlation has decreased slightly from 0.926 to 0.925 and the root_mean_squared_error has increased slightly from 34192.757 to 34685.970. **Thus, we will continue to use the maximal depth of 10 and just modify the number of trees for model optimization.**

Random Forest – Changed the number of trees from 100 to 125

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 34324.423 +/- 0.000  
squared_error: 1178166036.811 +/- 7662703905.025  
correlation: 0.926  
squared_correlation: 0.857  
prediction_average: 178462.575 +/- 83371.214
```

Figure 51: Random Forest - Model Performance - Changed the number of trees from 100 to 125

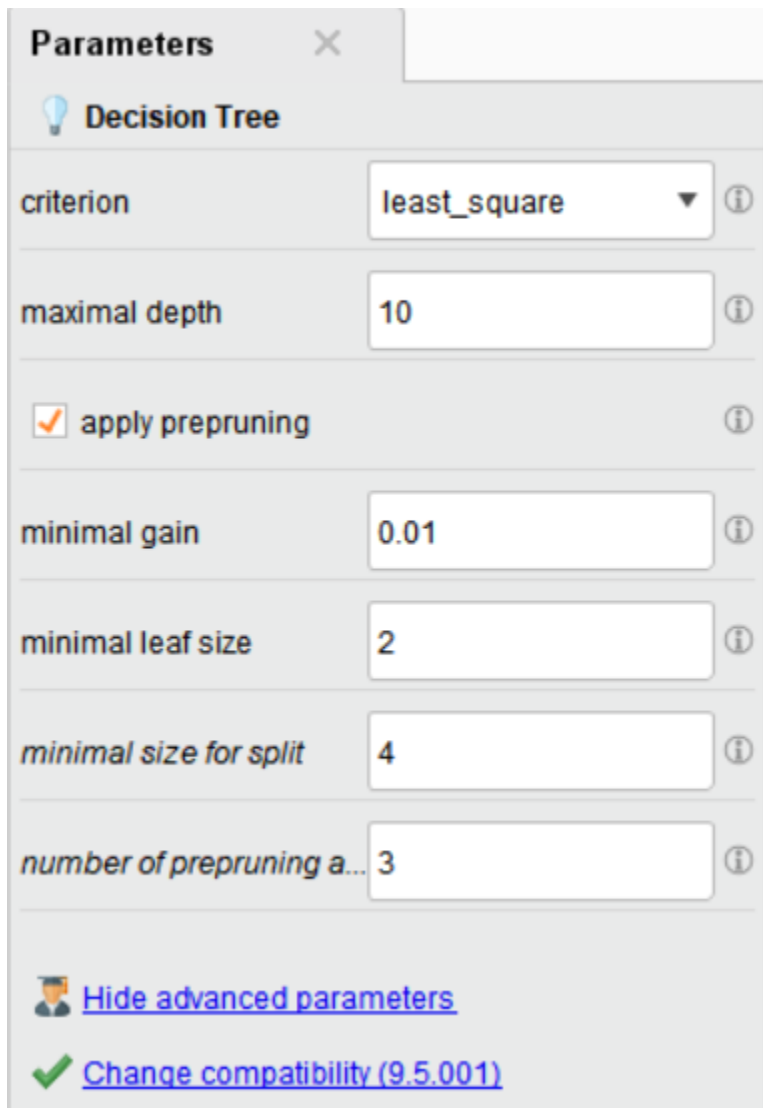
There is no change in correlation, but the root_mean_squared_error has increased slightly from 34192.757 to 34324.423.

PerformanceVector


```
PerformanceVector:  
root_mean_squared_error: 33920.910 +/- 0.000  
squared_error: 1150628152.870 +/- 7168986605.637  
correlation: 0.926  
squared_correlation: 0.858  
prediction_average: 178462.575 +/- 83371.214
```

Figure 52: Random Forest - Model Performance - Changed the number of trees from 100 to 75

There is no change in correlation, but the root_mean_squared_error has decreased from 34192.757 to 33920.910.



Parameters ✕

 **Decision Tree**

criterion ⓘ

maximal depth ⓘ


☒ apply prepruning ⓘ

minimal gain ⓘ

minimal leaf size ⓘ

minimal size for split ⓘ

number of prepruning a... ⓘ

 [Hide advanced parameters](#)


 [Change compatibility \(9.5.001\)](#)

Figure 53: Decision Tree - Default Parameters

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 53195.260 +/- 0.000  
squared_error: 2829735737.775 +/- 10652274709.339  
correlation: 0.837  
squared_correlation: 0.701  
prediction_average: 183807.384 +/- 97072.003
```

Figure 54: Decision Tree - Model Performance - Default Parameters

Decision Tree – Disabled Apply Prepruning button with everything else remaining same

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 53351.584 +/- 0.000
squared_error: 2846391545.163 +/- 10769563336.565
correlation: 0.789
squared_correlation: 0.623
prediction_average: 178462.575 +/- 83371.214
```

Figure 55: Decision Tree - Model Performance - Disabled Apply Prepruning button with everything else remaining same

The correlation has decreased from 0.837 to 0.789 and the root_mean_squared_error has increased from 53195.260 to 53351.584. **Thus, we will enable the Apply Prepruning button for further model optimization.**

Decision Tree – Changed the maximal depth from 10 to 15

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 52242.401 +/- 0.000
squared_error: 2729268481.158 +/- 13706312430.197
correlation: 0.787
squared_correlation: 0.620
prediction_average: 178462.575 +/- 83371.214
```

Figure 56: Decision Tree - Model Performance - Changed the maximal depth from 10 to 15

The correlation has decreased from 0.837 to 0.787 but the root_mean_squared_error has decreased from 53195.260 to 52242.401.

Decision Tree – Changed the maximal depth from 10 to 20

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 52242.748 +/- 0.000
squared_error: 2729304739.561 +/- 13706305855.867
correlation: 0.787
squared_correlation: 0.620
prediction_average: 178462.575 +/- 83371.214
```

Figure 57: Decision Tree - Model Performance - Changed the maximal depth from 10 to 20

The correlation has decreased from 0.837 to 0.787 but the root_mean_squared_error has decreased from 53195.260 to 52242.748. **Thus, we will continue to use maximal depth of 15 for further model optimization.**

Decision Tree – Changed the minimal gain from 0.01 to 0.02

The model performance has remained the same as shown in Figure 57.

Decision Tree – Changed the minimal gain from 0.01 to 0.005

The model performance has remained the same as shown in Figure 57. **Thus, we will continue to use minimal gain of 0.01 for further model optimization.**

Decision Tree – Changed the minimal leaf size from 2 to 5

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 36510.800 +/- 0.000  
squared_error: 1333038486.604 +/- 4110647926.770  
correlation: 0.900  
squared_correlation: 0.810  
prediction_average: 178462.575 +/- 83371.214
```

Figure 58: Decision Tree - Model Performance - Changed the minimal leaf size from 2 to 5

There is a substantial improvement in the model as the correlation has increased from 0.787 to 0.900 and the root_mean_squared_error has decreased from 52242.748 to 36510.800.

Decision Tree – Changed the minimal leaf size from 5 to 8

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 41237.062 +/- 0.000  
squared_error: 1700495268.351 +/- 7474556274.863  
correlation: 0.872  
squared_correlation: 0.760  
prediction_average: 178462.575 +/- 83371.214
```

Figure 59: Decision Tree - Model Performance - Changed the minimal leaf size from 5 to 8

The correlation has decreased from 0.900 to 0.872 and the root_mean_squared_error has increased from 36510.800 to 41237.062. **Thus, we will continue to use minimal leaf size of 5 for further model optimization.**

Decision Tree – Changed the minimal size for split from 4 to 5

The model performance is the same as shown in Figure 58.

Decision Tree – Changed the minimal size for split from 4 to 3

The model performance is the same as shown in Figure 58. Thus, we will continue to use minimal size for split of 4 for further model optimization.

Decision Tree – Changed the number of prepruning alternatives from 3 to 4

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 36499.823 +/- 0.000  
squared_error: 1332237086.194 +/- 4058966673.734  
correlation: 0.900  
squared_correlation: 0.809  
prediction_average: 178462.575 +/- 83371.214
```

Figure 60: Decision Tree - Model Performance - Changed the number of prepruning alternatives from 3 to 4

The correlation has remained the same as shown in Figure 58 but the root_mean_squared_error decreased from 36510.800 to 36499.823.

Decision Tree – Changed the number of prepruning alternatives from 4 to 2

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 37782.179 +/- 0.000  
squared_error: 1427493058.592 +/- 4549854477.115  
correlation: 0.893  
squared_correlation: 0.798  
prediction_average: 178462.575 +/- 83371.214
```

Figure 61: Decision Tree - Model Performance - Changed the number of prepruning alternatives from 4 to 2

The correlation has decreased from 0.900 to 0.893 and the root_mean_squared_error has increased from 36499.823 to 37782.179.

Decision Tree – Changed the number of prepruning alternatives from 4 to 5

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 36273.217 +/- 0.000  
squared_error: 1315746295.953 +/- 4148670291.670  
correlation: 0.901  
squared_correlation: 0.811  
prediction_average: 178462.575 +/- 83371.214
```

Figure 62: Decision Tree - Model Performance - Changed the number of prepruning alternatives from 4 to 5

The correlation has increased from 0.900 to 0.901 and the root_mean_squared_error has decreased from 36499.823 to 36273.217.

Decision Tree – Changed the number of prepruning alternatives from 5 to 6

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 36273.155 +/- 0.000  
squared_error: 1315741752.346 +/- 4132965590.436  
correlation: 0.901  
squared_correlation: 0.811  
prediction_average: 178462.575 +/- 83371.214
```

Figure 63: Decision Tree - Model Performance - Changed the number of prepruning alternatives from 5 to 6

The correlation has remained the same as shown in Figure 62 but the root_mean_squared_error has only shown a slight improvement from 36273.217 to 36273.155.

Decision Tree – Changed the number of prepruning alternatives from 6 to 7

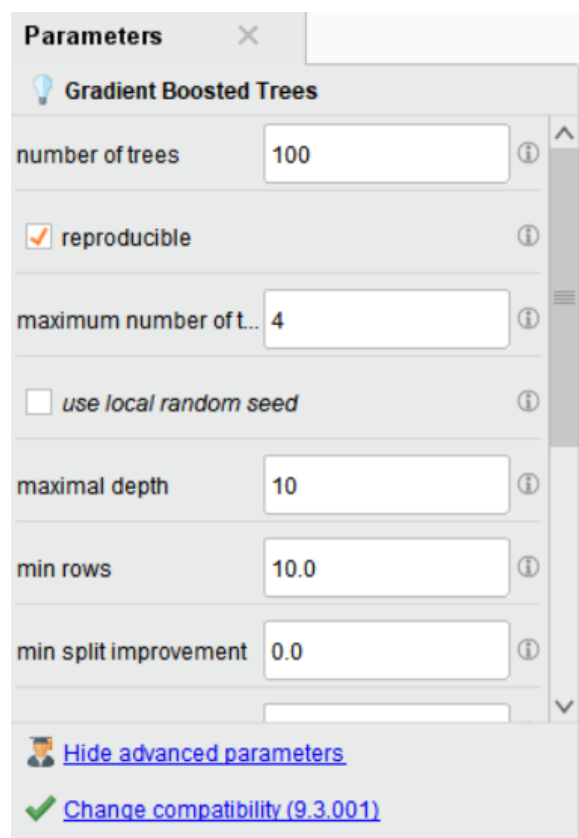
PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 36451.711 +/- 0.000  
squared_error: 1328727269.575 +/- 4190325913.343  
correlation: 0.900  
squared_correlation: 0.809  
prediction_average: 178462.575 +/- 83371.214
```

Figure 64: Decision Tree - Model Performance - Changed the number of prepruning alternatives from 6 to 7

The correlation has decreased from 0.901 to 0.900 and root_mean_squared_error has increased from 36273.155 to 36451.711.

Gradient Boosted Trees – Default Parameters



The screenshot shows a 'Parameters' window for 'Gradient Boosted Trees'. The parameters are as follows:

Parameter	Value
number of trees	100
reproducible	<input checked="" type="checkbox"/>
maximum number of t...	4
use local random seed	<input type="checkbox"/>
maximal depth	10
min rows	10.0
min split improvement	0.0

At the bottom, there is a link 'Hide advanced parameters' and a status message 'Change compatibility (9.3.001)' with a green checkmark.

number of bins: 20

learning rate: 0.01

sample rate: 1.0

distribution: AUTO

☐ early stopping

expert parameters: Edit List (0)...

[Hide advanced parameters](#)

[Change compatibility \(9.3.001\)](#)

Figure 65: Gradient Boosted Trees - Default Parameters

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 45893.305 +/- 0.000
squared_error: 2106195469.084 +/- 10429740083.333
correlation: 0.910
squared_correlation: 0.828
prediction_average: 178462.575 +/- 83371.214
```

Figure 66: Gradient Boosted Trees - Model Performance - Default Parameters

Gradient Boosted Trees – Increased the number of trees from 100 to 200

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 35516.679 +/- 0.000
squared_error: 1261434466.779 +/- 6539277269.871
correlation: 0.921
squared_correlation: 0.848
prediction_average: 178462.575 +/- 83371.214
```

Figure 67: Gradient Boosted Trees - Model Performance - Increased the number of trees from 100 to 200

The correlation has increased from 0.910 to 0.921 and the root_mean_squared_error has decreased from 45893.305 to 35516.679.

Gradient Boosted Trees – Increased the number of trees from 200 to 300

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 32008.256 +/- 0.000  
squared_error: 1024528458.897 +/- 5093938849.583  
correlation: 0.929  
squared_correlation: 0.863  
prediction_average: 178462.575 +/- 83371.214
```

Figure 68: Gradient Boosted Trees - Model Performance - Increased the number of trees from 200 to 300

The correlation has increased from 0.921 to 0.929 and the root_mean_squared_error has decreased from 35516.679 to 32008.256.

Gradient Boosted Trees – Increased the number of trees from 300 to 400

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30866.547 +/- 0.000  
squared_error: 952743706.763 +/- 4557026682.291  
correlation: 0.932  
squared_correlation: 0.868  
prediction_average: 178462.575 +/- 83371.214
```

Figure 69: Gradient Boosted Trees - Model Performance - Increased the number of trees from 300 to 400

The correlation has increased from 0.929 to 0.932 and the root_mean_squared_error has decreased from 32008.256 to 30866.547.

Gradient Boosted Trees – Increased the number of trees from 400 to 500

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30650.179 +/- 0.000  
squared_error: 939433449.740 +/- 4461867121.451  
correlation: 0.932  
squared_correlation: 0.869  
prediction_average: 178462.575 +/- 83371.214
```

Figure 70: Gradient Boosted Trees - Model Performance - Increased the number of trees from 400 to 500

The correlation has remained the same as in Figure 70, but the root_mean_squared_error has decreased from 30866.547 to 30650.179.

Gradient Boosted Trees – Increased the number of trees from 500 to 600

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30460.651 +/- 0.000  
squared_error: 927851270.187 +/- 4354081109.388  
correlation: 0.933  
squared_correlation: 0.870  
prediction_average: 178462.575 +/- 83371.214
```

Figure 71: Gradient Boosted Trees - Model Performance - Increased the number of trees from 500 to 600

The correlation has increased from 0.932 to 0.933 and the root_mean_squared_error has decreased from 30650.179 to 30460.651.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30325.758 +/- 0.000  
squared_error: 919651615.435 +/- 4298193896.929  
correlation: 0.933  
squared_correlation: 0.871  
prediction_average: 178462.575 +/- 83371.214
```

Figure 72: Gradient Boosted Trees - Model Performance - Increased the number of trees from 600 to 700

The correlation has remained the same as shown in Figure 72, but the root_mean_squared_error has decreased from 30460.651 to 30325.758.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30253.130 +/- 0.000  
squared_error: 915251903.422 +/- 4263403022.242  
correlation: 0.934  
squared_correlation: 0.871  
prediction_average: 178462.575 +/- 83371.214
```

Figure 73: Gradient Boosted Trees - Model Performance - Increased the number of trees from 700 to 800

The correlation has increased from 0.933 to 0.934 and the root_mean_squared_error has decreased from 30325.758 to 30253.130.

Gradient Boosted Trees – Increased the number of trees from 800 to 900

PerformanceVector

PerformanceVector:

```
root_mean_squared_error: 30215.662 +/- 0.000  
squared_error: 912986207.405 +/- 4230648377.460  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 74: Gradient Boosted Trees - Model Performance - Increased the number of trees from 800 to 900

The correlation has remained the same as shown in Figure 73, but the root_mean_squared_error decreased from 30253.130 to 30215.662.

Gradient Boosted Trees – Increased the number of trees from 900 to 1000

PerformanceVector

PerformanceVector:

```
root_mean_squared_error: 30199.711 +/- 0.000  
squared_error: 912022573.757 +/- 4217347602.012  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 75: Gradient Boosted Trees - Model Performance - Increased the number of trees from 900 to 1000

The correlation has remained the same as shown in Figure 73, but the root_mean_squared_error has decreased from 30215.662 to 30199.711.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30196.595 +/- 0.000  
squared_error: 911834363.887 +/- 4205820498.582  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 76: Gradient Boosted Trees - Model Performance - Increased the number of trees from 1000 to 1100

The correlation has remained the same as shown in Figure 73, but the root_mean_squared_error has decreased from 30199.711 to 30196.595.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30196.268 +/- 0.000  
squared_error: 911814624.584 +/- 4196531735.615  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 77: Gradient Boosted Trees - Model Performance - Increased the number of trees from 1100 to 1200

The correlation has remained the same as shown in Figure 73, but the root_mean_squared_error has decreased very slightly from 30196.595 to 30196.268.

Gradient Boosted Trees – Increased the number of trees from 1200 to 1300

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30193.040 +/- 0.000  
squared_error: 911619667.702 +/- 4195182173.024  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 78: Gradient Boosted Trees - Model Performance - Increased the number of trees from 1200 to 1300

The correlation has remained the same as shown in Figure 73, but the root_mean_squared_error has decreased from 30196.268 to 30193.040.

Gradient Boosted Trees – Increased the number of trees from 1300 to 1400

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30203.444 +/- 0.000  
squared_error: 912248012.796 +/- 4199159847.150  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 79: Gradient Boosted Trees - Model Performance - Increased the number of trees from 1300 to 1400

The correlation has remained the same as shown in Figure 73, but the root_mean_squared_error has increased from 30193.040 to 30203.444.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30192.611 +/- 0.000  
squared_error: 911593780.916 +/- 4195569023.397  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 80: Gradient Boosted Trees - Model Performance - Increased the number of trees from 1000 to 1250

The correlation has remained the same as in Figure 73, but the root_mean_squared_error has decreased slightly from 30193.040 to 30192.611. **Thus, we will continue to use 1250 number of trees for further model optimization.**

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30136.583 +/- 0.000  
squared_error: 908213662.753 +/- 4140973432.498  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 81: Gradient Boosted Trees - Model Performance - Increased the maximal depth from 10 to 15

The correlation has remained the same as in Figure 73, but the root_mean_squared_error has decreased from 30192.611 to 30136.040.

PerformanceVector

```
PerformanceVector:  
root_mean_squared_error: 30157.010 +/- 0.000  
squared_error: 909445264.318 +/- 4204910698.463  
correlation: 0.934  
squared_correlation: 0.872  
prediction_average: 178462.575 +/- 83371.214
```

Figure 82: Gradient Boosted Trees - Model Performance - Increased the maximal depth from 15 to 20

The correlation has remained the same as in Figure 73, but the root_mean_squared_error has increased from 30136.040 to 30157.010.