House Price Prediction Modeling

Regression Model Demo

Ren Hwai, 2024 July

Information Links

- **Github page for the code:**

  https://github.com/Ren1990/house_price_reg_model

- **Dataset from Kaggle:**

  https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview

- **My Linkedin:**

  https://www.linkedin.com/in/renhwai-kong/

- **My Tableau:**

  https://public.tableau.com/app/profile/kyloren.kong/viz/Demo_2024InvestmentPortfolio/DBPortfolio

- **My GenAI Job Interviewee Agent :**

  https://renhwaichatbot.streamlit.app/

About Myself

Hi! This is me, Ren Hwai, chilling in Iceland. Happy family trip during my career break!

"You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future." - Steve Jobs

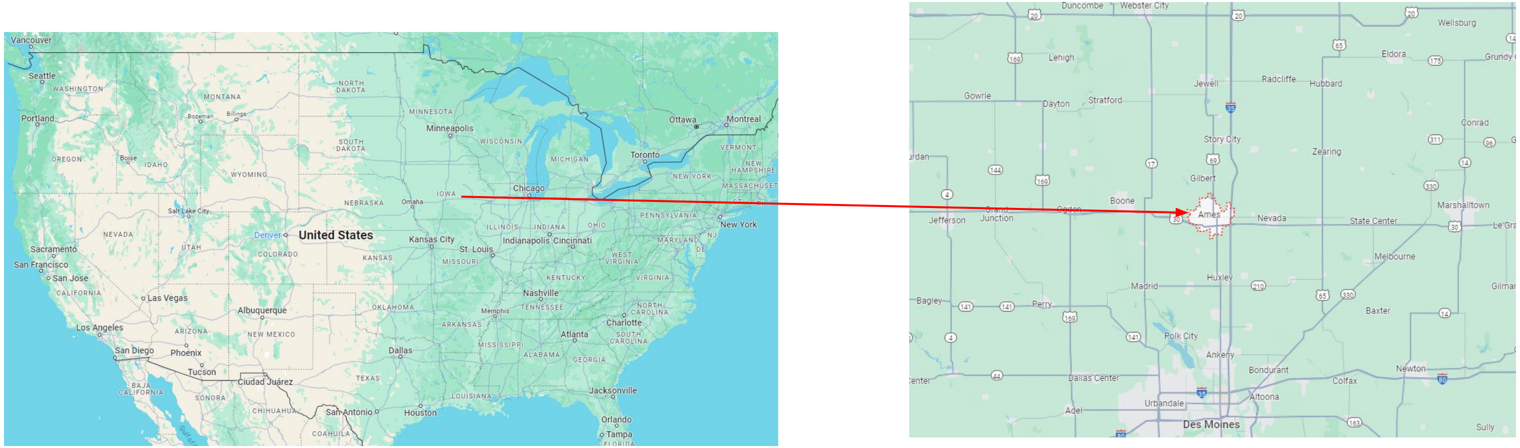
After working in top US semicond company for 8 years as Senior Technology Development Process Engineer & Smart Manufacturing Analyst (Eng. IV), I take a long break to sharpen my Python skill in data science & analysis, and study for CFA (Chartered Finance Analyst) to look for new industry exposure and work opportunity.

Executive Summary

- This project is to train regression model to predict US house price in Ames, Lowa.
- Data transformation is performed before model training. 5-fold Cross Validation is used to reduce overfit risk.
- RMSE is chosen as the deciding performance metric to prioritize in minimizing prediction error.
- Optuna Library is used to auto finetune model hyperparameter.
- The mean of data population is $180,921 and Std is $79,442; **Acceptable RMSE values is set to be less than 30% of Std ($23,833)**
- The first Gradient Boost Regression (GBR) Model created under current workflow does not meet the target RMSE ($25,365), further enhancements are explored:
  - Time-Based Train-Test Split to improve model generalization
  - Switch to Extra Gradient Boost Regression (XGBR) Model which is the 2nd model candidate during model selection
- **Final XGBR Model has achieved the target RMSE with $22,619 meeting the target RMSE $23,833**
- Most critical factors to house price predictions are:
  - ‘OverallQual’: Rates the overall material and finish of the house.
  - ‘GrLivArea’: Above grade (ground) living area square feet.
  - ‘GarageCars’: Size of garage in car capacity.

Introduction

- This dataset, titled '[House Prices - Advanced Regression Techniques]' from Kaggle, is the US Housing dataset in City **Ames, Lowa**. It contains 79 features related to house information such as 'OverallQual', 'LotArea', 'YearBuilt' etc.
- The data is from year 2006 to 2010.

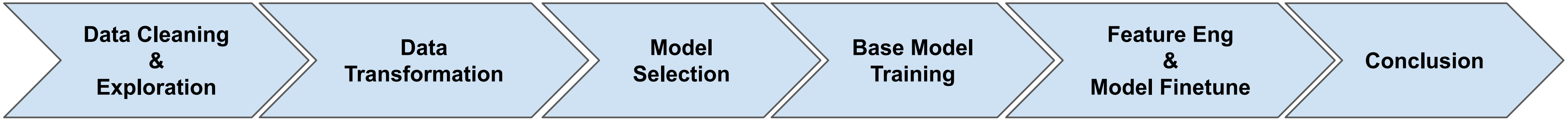


Objective

- The goal is to predict the housing price by training a regression model with the provided data and explain what are the key factors affecting the house price.

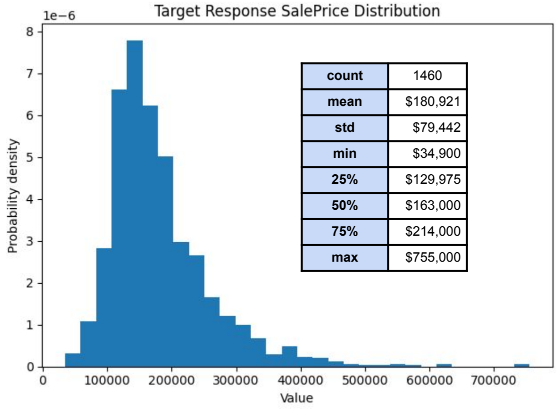
Model Training

- The model targets to predict house 'SalePrice', which is a **Regression Problem** using supervised learning.
- Below is the model training flow:

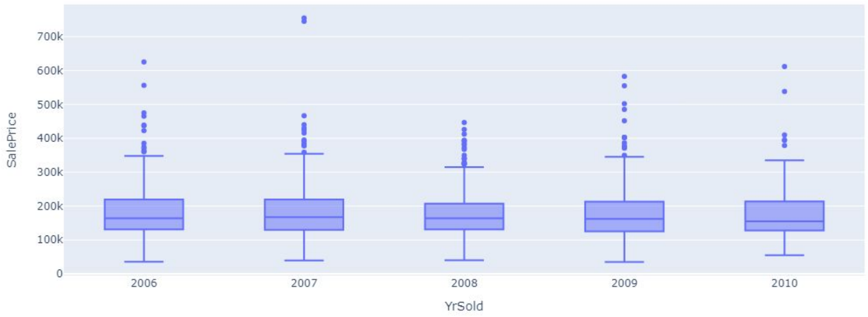


Dataset Overview

1. Data provided:
   a. Train.csv (1,460 rows x 81 columns)
2. Data quality:
   a. No observable duplications or single value data.
   b. Missing data was found when checking for ‘NaN’ value. However based on data description, those are true null. For example if the house does not have the ‘pool’, the ‘pool quality’ will be null. The null will be handled duuring data transformation.
3. ‘SalePrice’ distribution is positively skewed due to few extreme high ‘SalePrice’



count	1460
mean	$180,921
std	$79,442
min	$34,900
25%	$129,975
50%	$163,000
75%	$214,000
max	$755,000

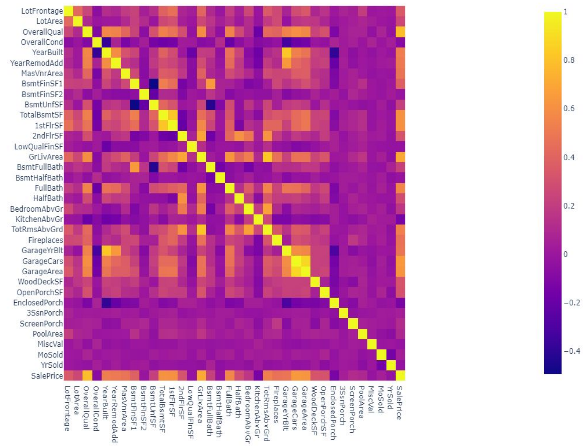


Data Exploration

1. For Numeric Features, correlation with ‘SalePrice’ is checked: ‘OverallQuality’, ‘GroundLivingArea’ . ‘GarageCars’ etc are the features with high correlation

Correlation Map

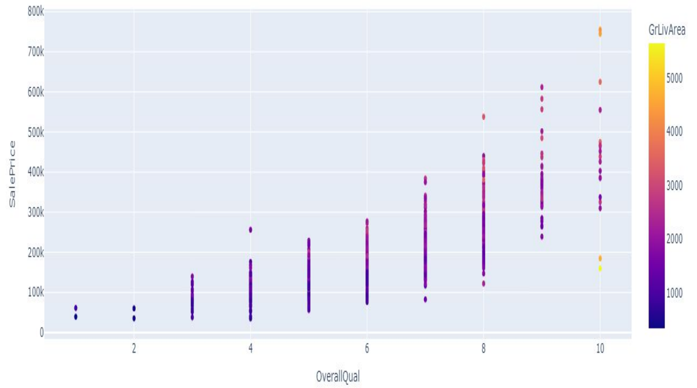
Correlation Heatmap



Correlation Score

Numeric Feature	Correlation to SalePrice
OverallQual	0.790982
GrLivArea	0.708624
GarageCars	0.640409
GarageArea	0.623431
TotalBsmtSF	0.613581
1stFlrSF	0.605852
FullBath	0.560664
TotRmsAbvGrd	0.533723
YearBuilt	0.522897
YearRemodAdd	0.507101

Scatter Plot of SalePrice vs OverallQual



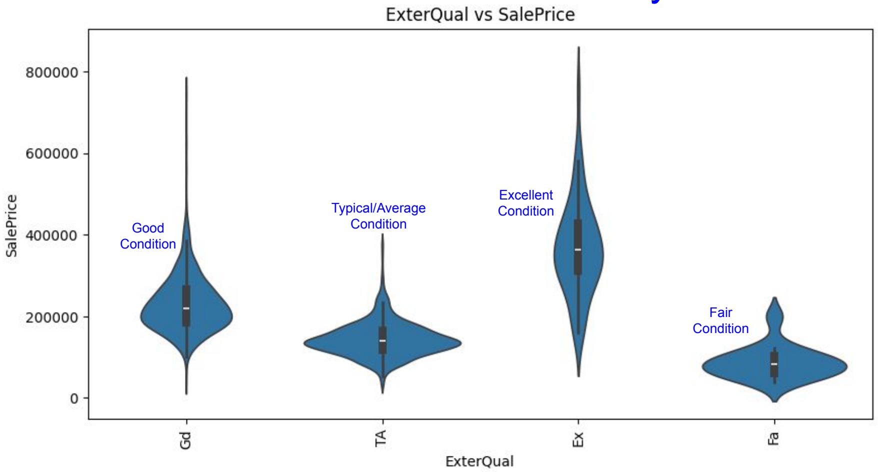
Data Exploration

1. For Categorical Features, t-test is used to check if the any pair of the subgroup data of the Categorical Feature has different mean, i.e:
   a. Null hypothesis(ho): there is no significant difference between the ‘SalePrice’ means of the two subgroups;
   b. Alternative hypothesis(ha): there is significant difference between the ‘SalePrice’ means of the two subgroups
2. Distinctions are found for various categorical features, such as ExteriorQuality, KitchenQuality etc

Examples of T-Test Result

feat	at least one pair feat_val reject h0	feat_val1	feat_val2	p value	statistic	degree of freedom
ExterQual	TRUE	TA	Ex	3.50E-153	-31.97	958
KitchenQual	TRUE	TA	Ex	4.70E-148	-32.12	835
BsmtQual	TRUE	TA	Ex	3.60E-140	-31.43	770
GarageFinish	TRUE	Unf	Fin	1.50E-81	-21.12	957
FireplaceQu	TRUE	nan	Gd	1.40E-78	-20.43	1070
Foundation	TRUE	PConc	CBlock	3.70E-72	19.16	1281
Neighborhood	TRUE	NridgHt	NAmes	6.50E-69	23.16	302
MasVnrType	TRUE	nan	Stone	1.20E-57	-17.09	1000
GarageType	TRUE	Attchd	Detchd	8.90E-56	16.54	1257
HeatingQC	TRUE	Ex	TA	1.10E-51	15.9	1169
SaleType	TRUE	WD	New	1.40E-44	-14.52	1389
SaleCondition	TRUE	Normal	Partial	1.40E-42	-14.17	1323

Distinct SalePrice Difference by ExterQual



*Initial plan is to use F-test(ANOVA) to study distinctions of SalePrice by all categorical features. During coding there is a challenge to automate f-test for all categorical features. Scipy Library f-test function requires to pass all the subgroups at once, for example f_oneway(subgrp1, subgrp2, subgrp3). The number of subgroups are not identical for the categorical features. After run into the coding bottleneck, t-test function is used, and a customized function which perform t-test on all subgroup pairs is created, for example ttest_ind(subgrp1,subgrp2), then ttest_ind(subgrp1,subgrp3), then ttest_ind(subgrp2,subgrp3) etc

Data Transformation is performed before fitting data for model training

1. Convert rating categorical feature into numeric value based on data description.

Index	ExterQual
0	NaN (N/A)
1	Po (Poor)
2	Fa (Fair)
3	Ta (Typical/ Average)
4	Gd (Good)
5	Ex (Excellent)

→

Index	ExterQual
0	0
1	1
2	2
3	3
4	4
5	5

2. Binary Feature -> Label Encoder

Index	CentralAir
0	N
1	Y
2	Y
3	N
4	N

→

Index	CentralAir
0	0
1	1
2	1
3	0
4	0

3. Multiple value categorical feature -> one-hot encoder. New columns will be created

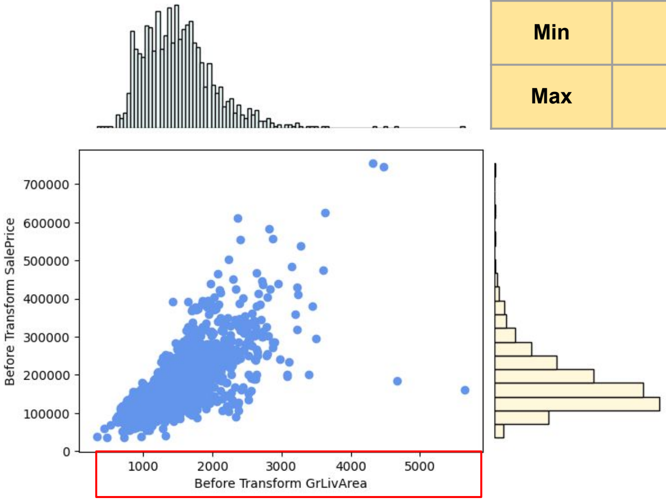
- In example below, if label encoder is used, ‘BrkTil’ will be assigned 0, ‘CBlock’ -> 1, ‘PConc’ -> 2 etc, and total columns will remain same.
- Downside of using label encoder is, model will treat ‘CBlock’ higher value than ‘BrkTil’ (1>0) etc.
- Since this is not the true relation or comparison, label encoding should be avoid to prevent machine learning to pick up this relation

Index	Foundation
0	BrkTil
1	CBlock
2	PConc
3	Slab
4	Stone
5	Wood

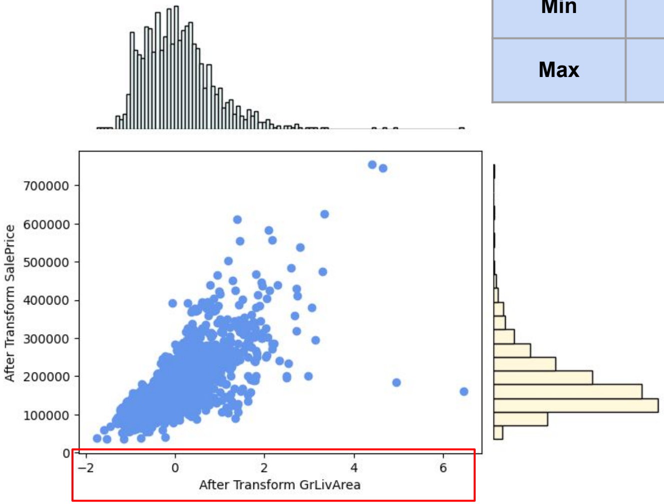
Index	Foundation_BrkTil	Foundation_CBlock	Foundation_PConc	Foundation_Slab	Foundation_Stone	Foundation_Wood
0	1	0	0	0	0	0
1	0	1	0	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1

Data Cleaning & Exploration | **Data Transformation** | Model Selection | Base Model Training | Feature Eng & Model Finetune | Conclusion

4. Numerical feature -> robust-scaler encoder.



Count	1460
Mean	1515.46
Median	1464
Std	525.48
Min	334
Max	5642



Count	1460
Mean	0.0795
Median	0.0000
Std	0.8119
Min	-1.7458
Max	6.4550

Model Selection

- There are many regression models for selection. Instead of trying all models and perform fine tuning for all, Model Selection is applied to select potential model candidate for optimization.
- Below are 10 regression models used in Model Selection:
  a. Linear Regression
  b. Gradient Boost Regression
  c. Extra Boost Regression (XGR)
  d. Ridge
  e. LASSO
  f. LARS
  g. Decision Tree Regression
  h. Support Vector Regression
  i. Random Forest Regression
  j. LASSO LARS

Random Train-Test Split and K-fold Cross Validation

- The train.csv is **split into r_train.csv and r_test.csv** using 80/20 random split. r_train.csv is used for model screening, hyperparameter tuning and feature engineering; r_test.csv is used for final model performance validation.
- 5-fold (i.e. K=5) Cross Validation (CV) is applied in model training to obtain generalized model to avoid overfitting caused by machine learning.
- The r_train.csv provided by dataset is randomly split into 5 equal sets (test.csv is not used).
  - 4 sets are used for training model and 1 set is used for validating model performance
  - 1 set is used for model result validation
  - Rotate the set so that all 5 sets have been used for validation
- Average **RMSE** of 5 folds result is used to assess model performance

1. Split train.csv randomly into 5 sets

1	2	3	4	5

train.csv

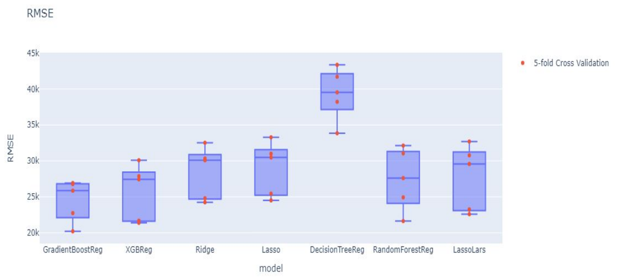
2. Split train.csv randomly into 5 sets for cross validation

Train Data

Validation Data

K=1	1	2	3	4	5
K=2	1	2	3	4	5
K=3	1	2	3	4	5
K=4	1	2	3	4	5
K=5	1	2	3	4	5

3. Assess the 5-fold model performance result

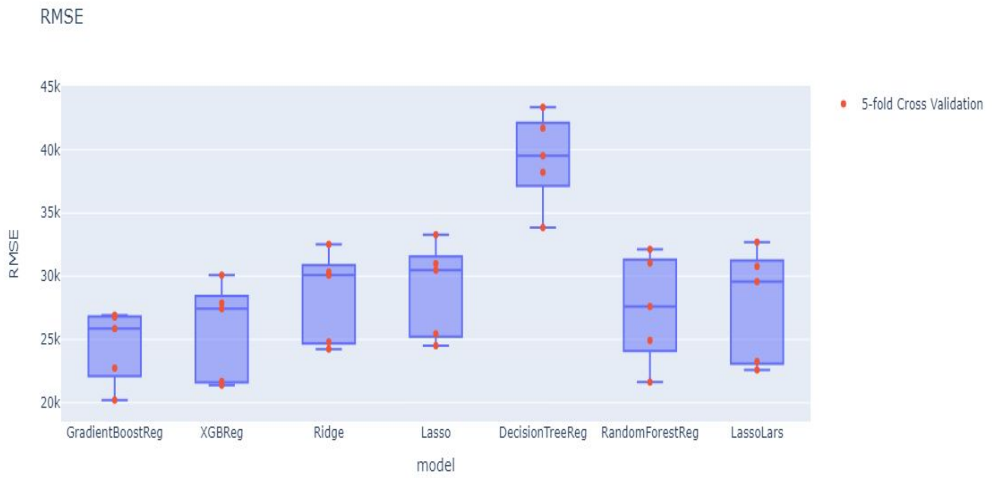


Model Selection Outcome: GBR Model is selected

1. Summary of average score: GradientBoost Regression has the lowest average RMSE in 5-fold assessment.

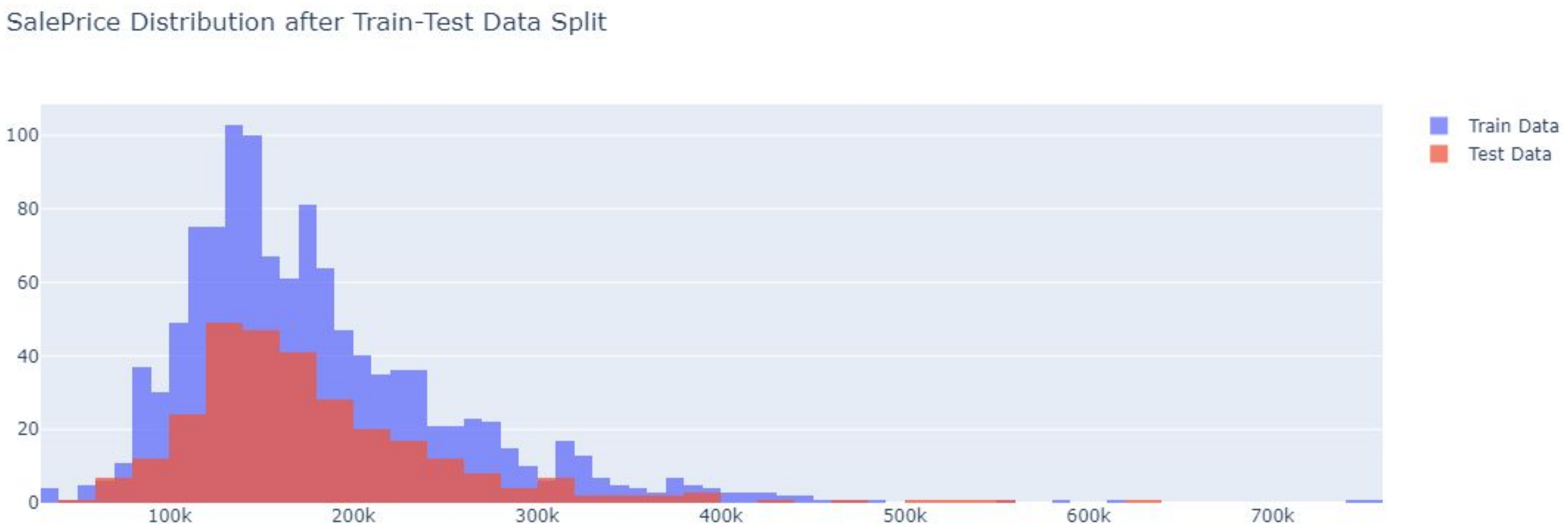
2. Box plot of 5-fold result shows that GBR is one of the models without abnormal widespread/variation in 5-fold CV.

Model	R2	RMSE
GradientBoost Reg	0.8518	$29,982
XGBReg	0.8503	$30,120
RandomForest Reg	0.8380	$31,532
LassoLars	0.7653	$36,483
Ridge	0.7473	$38,096
Lasso	0.7284	$39,127
DecisionTree Reg	0.7128	$41,853
SupportVector Reg	-0.0569	$80,990
Linear Reg	-6721579…	$8156058…
Lars	-22322870…	$5743938…



Train-Test Data Split

- Data set is split into Train and Test Data (i.e ‘unseen data) to reduce overfitting risk. Train Data is used for model training and finetune, while Test Data is the final assessment of model performance.

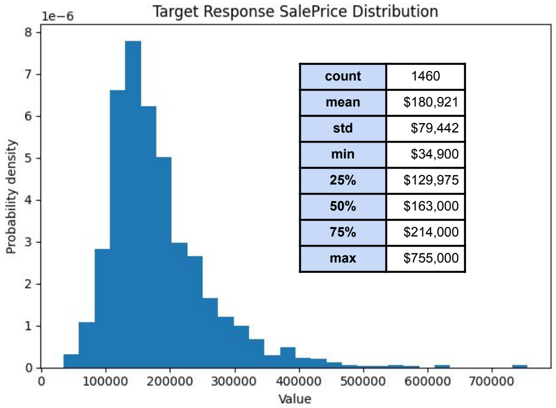


Select RMSE as The Key Performance Metric

- RMSE is selected as the deciding performance metric. The smaller the RMSE, the smaller expected prediction error.
- Higher R2 is preferred but a R2>0.7 should be sufficient as about 70% of the 'SalePrice' variations can be explained by the model:
  - High R2 is unrealistic unless the data set have all explaining factors. This housing price data set is definitely still missing important factors, such as buyer/seller finance status and interest rate
  - Data collected from historical process tends to have lower R2 than data collected from controlled experiment. Design of Experiment(DOE) will optimize the response & factor levels and data collection for model fitting.
- Adjusted R2 is used to assess whether features used are relevant for model prediction. Good Adj R2 indicate the model is generalized and less overfit.

Defining Acceptance Criteria for RMSE

- Acceptance criteria of RMSE can be specified by criticality of use cases (Is this model used for internal research or use for trading?).
- For this case, we derive the acceptance criteria based on SalePrice distribution.
- In general, model prediction error should be smaller than the spread of distribution, such as:
  - Std (**$79,442**),
  - Or Interquartile Range(IQR)=Q3-Q1=$214,000-$129,975=**$84,0245.**
  - Std will be used since it is the tighter measure
- Any Model Prediction RMSE larger than the Std should be considered bad model. RMSE criteria is depending on use case and is varied from industry to industry. For this project, the criteria of RMSE is set to be:
  - Acceptable: less than 30% of Std: RMSE<=$23,833
  - Good: less than 10% of Std: RMSE<=$7,944
  - **To certain extent, the expectation for the house price prediction is to have average error less than $23,833**



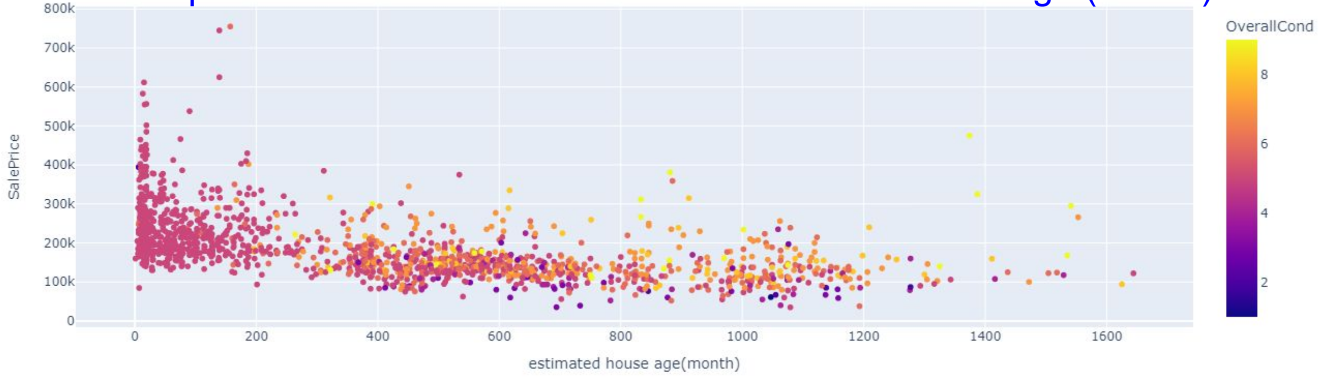
Data Cleaning & Exploration | Data Transformation | Model Selection | **Base Model Training** | Feature Eng & Model Finetune | Conclusion

New Age Related Features Are Created

Based on the year and month feature in dataset to estimate age related features:

1. estimated_house_age(month)= ‘YrSold’*12+ ‘MoSold’- ‘YearBuilt’
2. estimated_house_age(yr)= ‘YrSold’- ‘YearBuilt’
3. estimated_remodadd_age(yr)=’YrSold’- ‘YearRemodAdd’
4. estimated_remodadd_age(month)= ‘YrSold’*12+ ‘MoSold’- ‘YearRemodAdd’*12
5. estimated_garage_age(yr)= ‘YrSold’- ‘GarageYrBlt’
6. estimated_garage_age(month)= ‘YrSold’*12+ ‘MoSold’- ‘GarageYrBlt’*12

Example of ‘SalePrice’ vs new feature estimated house age (month)

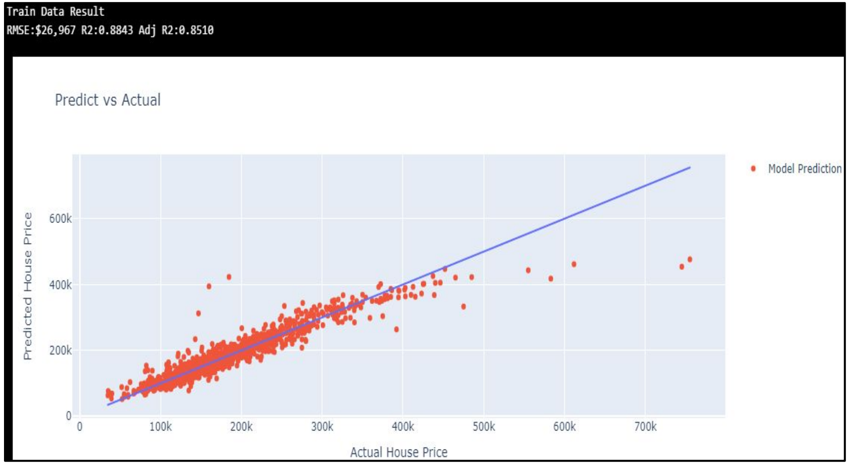


Data Cleaning & Exploration | Data Transformation | Model Selection | **Base Model Training** | Feature Eng & Model Finetune | Conclusion

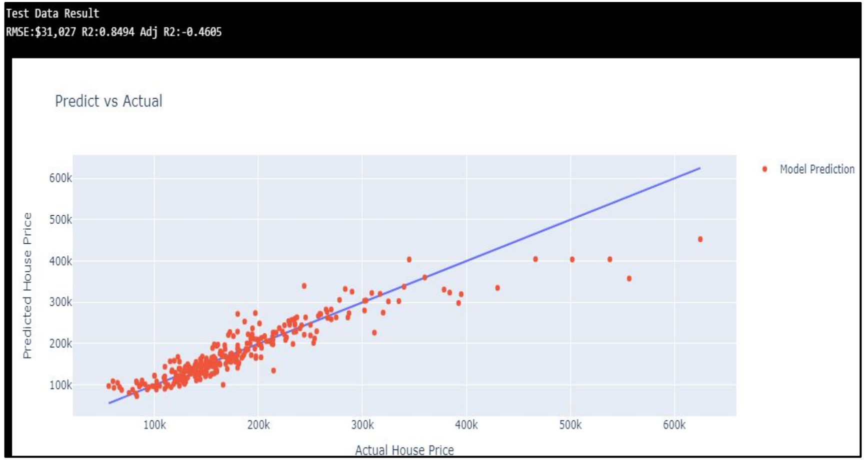
Assess Model Performance Using Train and Test Data

- Model performance is measured on Train and Test Data:
  - Test Data result is the final gauge of model performance.
  - Overfit risk can be assessed by comparing results between Train and Test Data.
- Below is default Model (Model0) results:

Train Data RMSE: $26,967



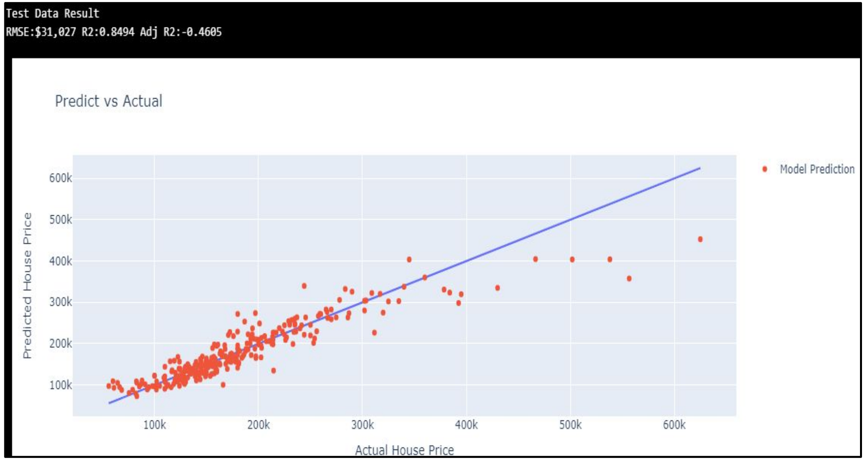
Test Data RMSE: $31,027



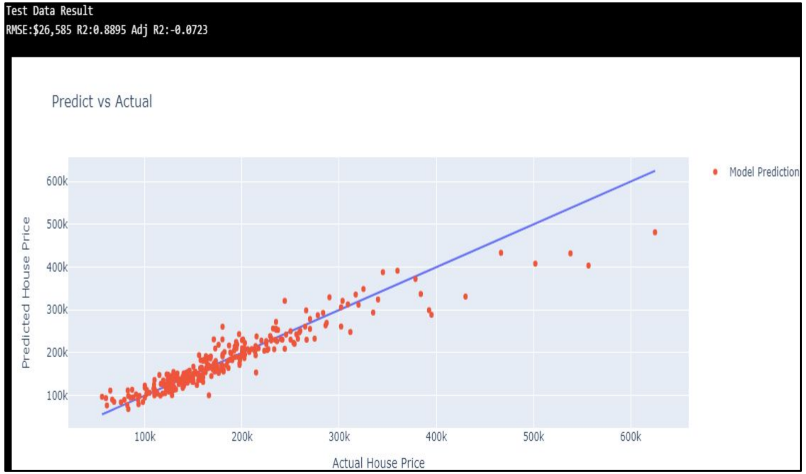
Create Base GBR Model (‘Model1’) with Optuna

- Optuna is a Python Library used for Model Hyperparameter Tuning:
  - Setup hyperparameter range for optimization search space
  - Setup Cross Objective function: maximize average RMSE calculated from 5-fold cross validation
  - Leverage Optuna Algo to search best hyperparameter
- After Optuna optimized ‘Model1’ has improved RMSE from $31,027 to $26,585 using Test Data

Before Optimization RMSE: $31,027

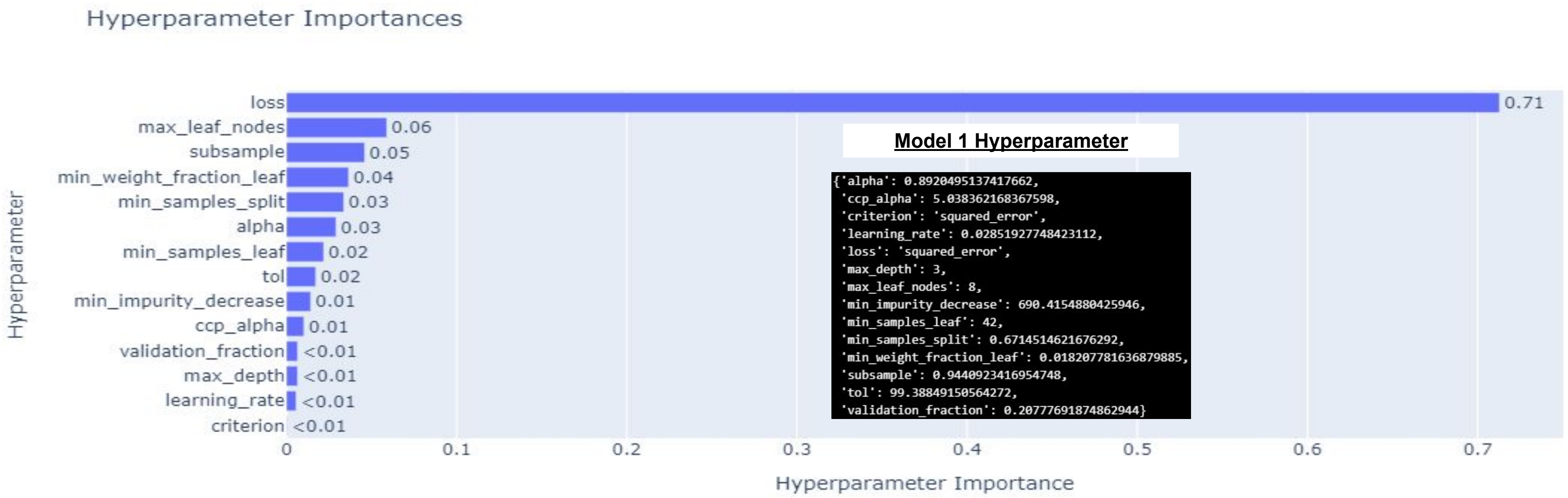


After Optimization RMSE: $26,585



GBR Hyperparameter Importance of ‘Model1’

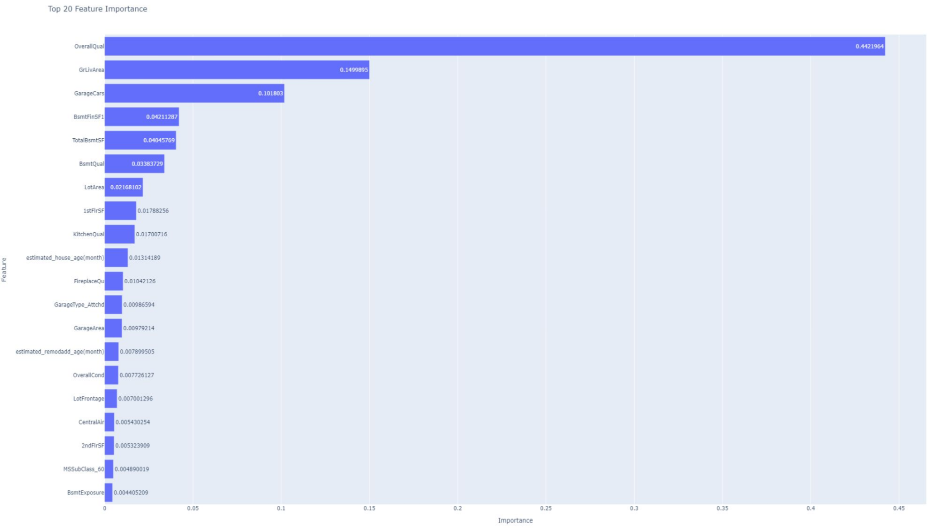
- Through Optuna hyperparameter importance feature, it is found that ‘loss’ is the most important hyperparameter
- ‘loss’:‘squared_error’ could be leading to overfitting Train Data. ‘squared_error’ will be removed in next Model Finentune



Feature Importance of ‘Model1’

- ‘Model1’ contains 261 features. Most important features are:
  - ‘OverallQual’: Rates the overall material and finish of the house.
  - ‘GrLivArea’: Above grade (ground) living area square feet.
  - ‘GarageCars’: Size of garage in car capacity.

1. Top 20 Important Features



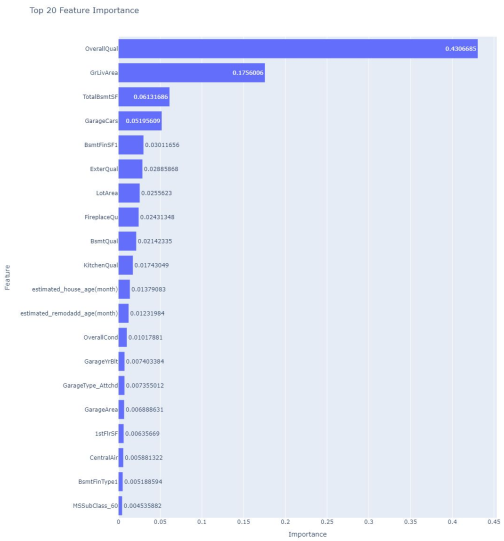
2. More than half of the Features have zero importance

Data Cleaning & Exploration | Data Transformation | Model Selection | Base Model Training | **Feature Eng & Model Finetune** | Conclusion

Optimized GBR Model

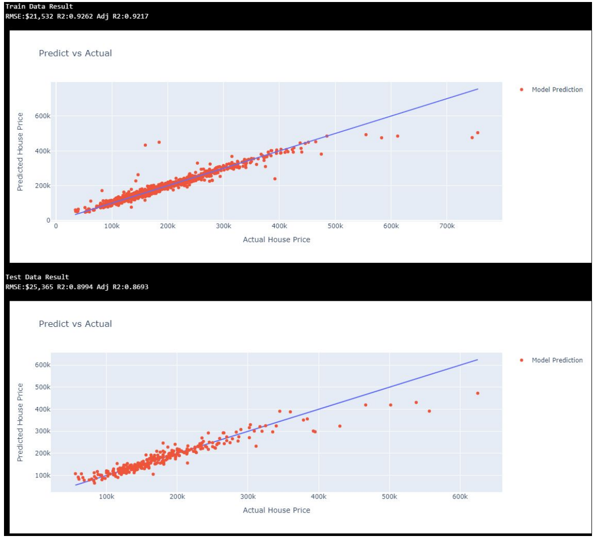
- Low importance features are removed. Final model features is 206->**105**
- Reoptimize the hyperparameter using Optuna. **Final Model has achieved Test Data RMSE: $25,365**, with R2:0.8994 and Adj R2:0.8698
- Train Data performance is comparable although minor degradation is found in Test Data

1. Top 20 Features in Final Model



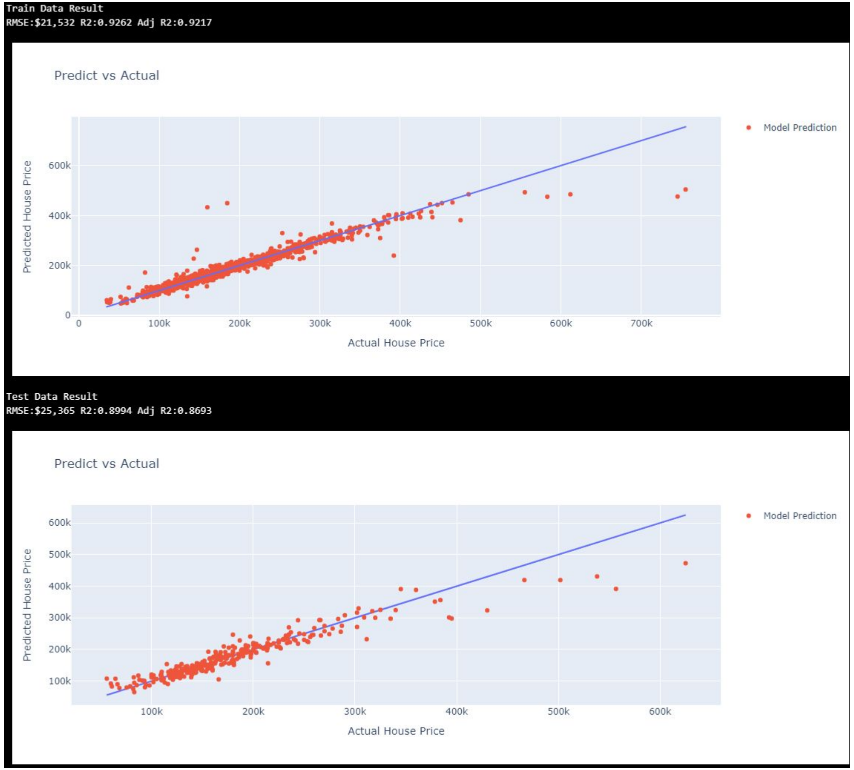
2. Final model has achieved RMSE $25,365 with optimized hyperparameter

final_hp={'alpha': 0.9255961785026856,
'ccp_alpha': 91.41840526743391,
'criterion': 'squared_error',
'learning_rate': 0.08954087410410139,
'loss': 'huber',
'max_depth': 3,
'max_leaf_nodes': 13,
'min_impurity_decrease': 696.0681036379524,
'min_samples_leaf': 35,
'min_samples_split': 0.6231724861087317,
'min_weight_fraction_leaf': 0.019118424312254844,
'subsample': 0.8744747724971967,
'tol': 93.81456919882424,
'validation_fraction': 0.24390880714068283}



Model RMSE Is Below Borderline of Acceptance

- Recall that based on criteria RMSE<30% Std of SalePrice distribution, RMSE should be lower than $23,833, but the Final Model RMSE is $25,367, or 31.9% of Std of data set.
- Need follow up with enhancement plan to meet the target RMSE



Model Enhancement Plan

Typical model enhancement is to revisit model training workflow and deep dive on:

1. Data Engineering during data cleaning. Exam the 'leads' found during data exploration and their effects during model training.
2. Change data transformation approach or investigate and remove outlier
3. More Feature Engineering based on new learnings from current model training
4. Explore new model during model selection

Above options might take longer time to develop. Based on experience, below enhancements are planned:

- **Improve model generalization by:**
  - Reduce the number of features (~50)
  - Use time-based split to replace random split during cross validation and model optimization. Time-based split should reward model setting with better stability over time.
- Explore **XGBR model** since it is the 2nd rank model during model screening

Enhancement 1: Time-Based Split Train-Test → **Enhancement 2: XGB Reg Model** → **Combining 1&2: Time-Based Split XGB Reg** → **Conclusion**

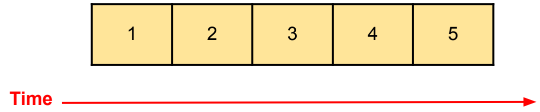
Time-Based Data Split vs Random Data Split

Time-Based Data Split can be thought as Backtesting Model which retrains with rolling time period data:

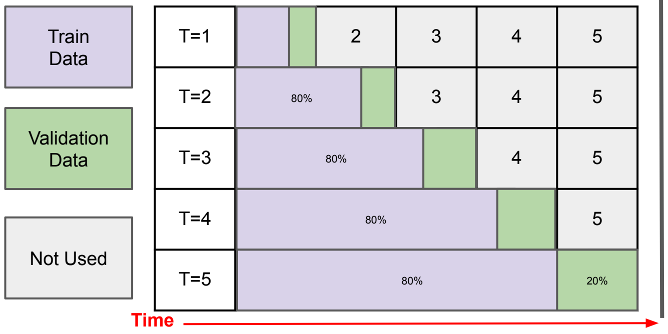
1. Training data is split into 5 sets (5-fold) according to time
2. During cross-validation, at T (period)=1,the earliest data set is split into 80:20 according to the time for model train-validation; At K=2, the earliest two data set are combined and split into 80:20 according to the time, and so on.

Time-Based Data Split

1 Split train.csv into 5 time periods.:



2 Cross validate the model with each time period (T) data

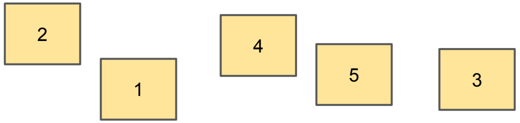


Time-Based Data Split is similar to Backtesting Model: Deploy model at T=1, and when T=2, retrain the model with new period data, and deploy again, etc.

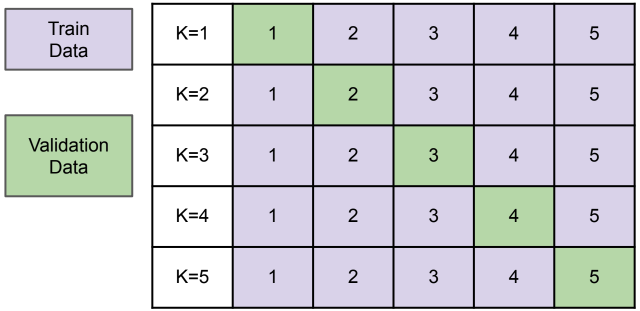
Time-Based Data Split might prioritize model stability (i.e. consistent performance across different time periods)

Random Data Split

Split train.csv randomly into 5 sets:



Split train.csv randomly into 5 sets for cross validation

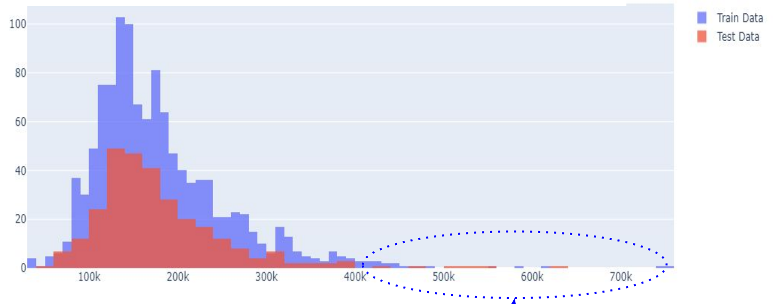


Enhancement 1: Time-Based Split Train-Test | Enhancement 2: XGB Reg Model | Combining 1&2: Time-Based Split XGB Reg | Conclusion

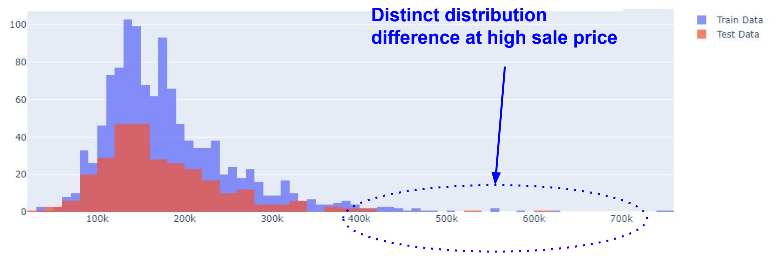
- Although the Train and Test data sets have same sample numbers, the samples are not equal. Therefore, direct model performance comparison between two split methods are meaningless
- However the relative ranking still shows that Gradient Reg is the best model with the lowest RMSE

	Random Split		Time-Based Split	
Model	R2	RMSE	R2	RMSE
GradientBoost Reg	**0.8518**	**$29,982**	**0.7776**	**$33,242**
XGBReg	0.8503	$30,120	0.7211	$37,899
RandomForest Reg	0.8380	$31,532	0.7992	$33,376
LassoLars	0.7653	$36,483	0.6364	$41,263
Ridge	0.7473	$38,096	0.6099	$44,482
Lasso	0.7284	$39,127	0.4302	$50,574
DecisionTree Reg	0.7128	$41,853	0.5649	$55,748
SupportVector Reg	-0.0569	$80,990	-0.0495	$83,509
Linear Reg	-6721579…	$8156058…	-54994157…	$314697…
Lars	-22322870…	$5743938…	-86121627…	$104067…

Random Split Distribution



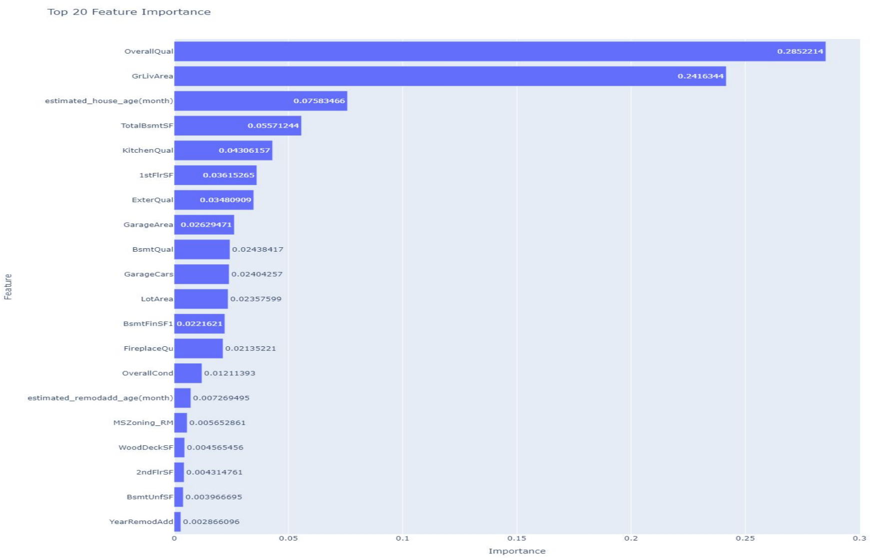
Time-Based Split Distribution



Enhancement 1: Time-Based Split Train-Test | Enhancement 2: XGB Reg Model | Combining 1&2: Time-Based Split XGB Reg | Conclusion

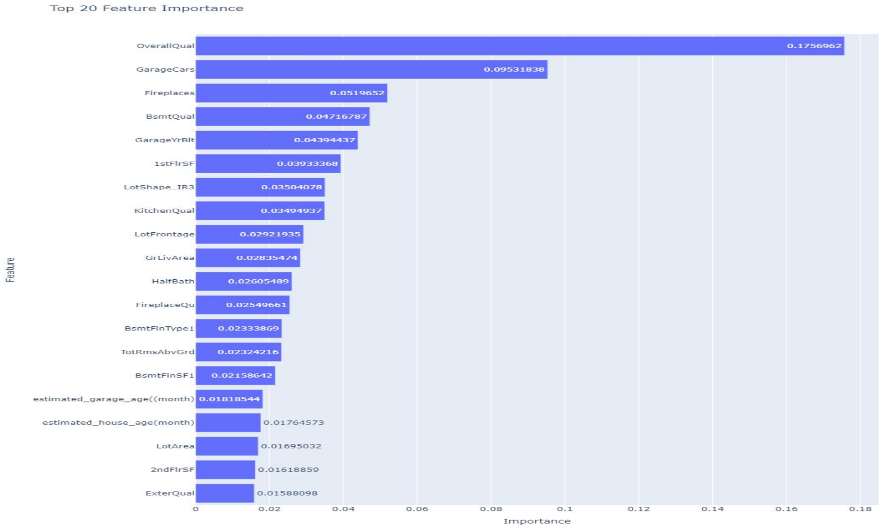
Time-Based Split GBR Model Meets The RMSE Criteria

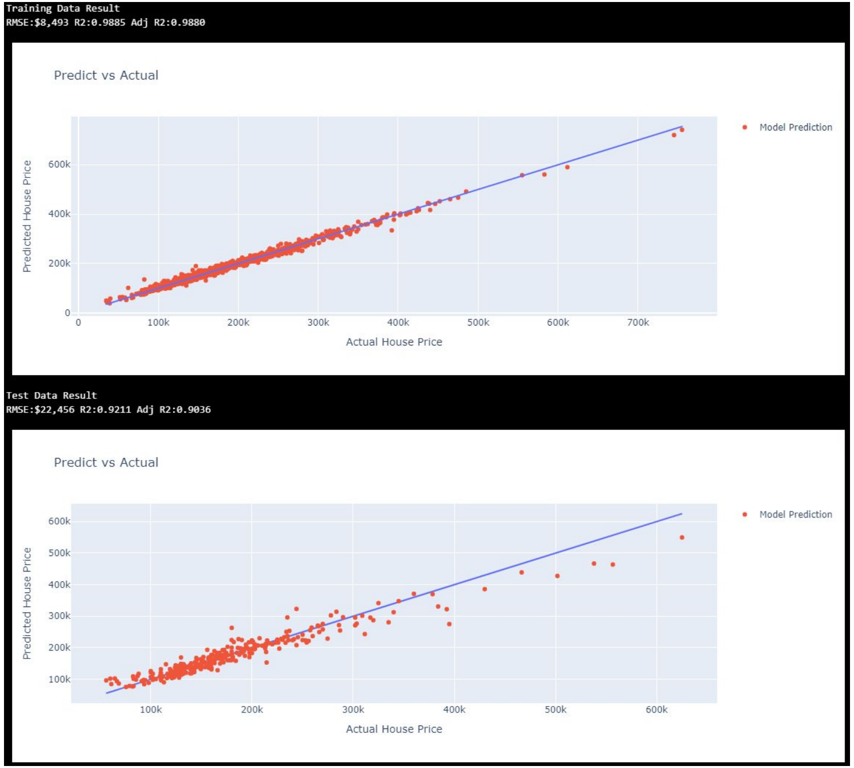
- Final optimized GBR Model has achieved RMSE $22,671 using Time-Based Split with 53 features.
- The model meets the acceptance criteria for less than 30% of the Std, $23,833.



Random Split XGR Model Meets The RMSE Criteria

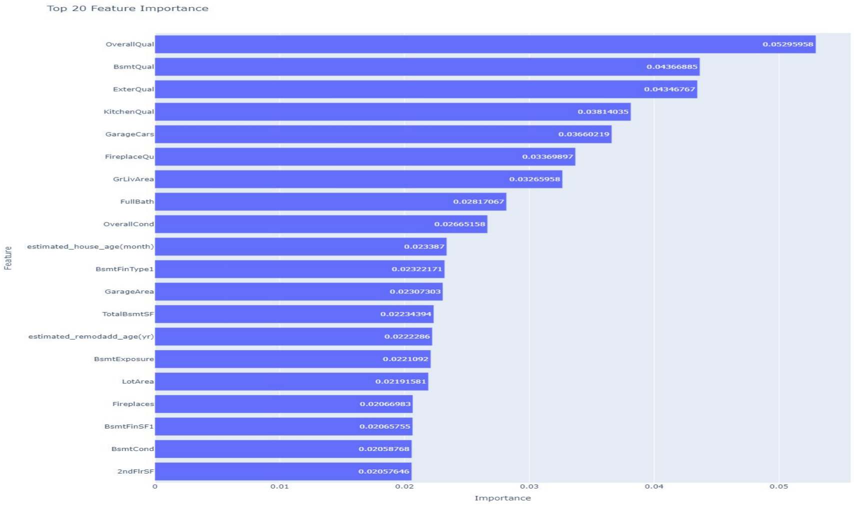
- XGB is selected as alternate model candidate since it is the 2nd rank model during model screening in Part 1
- After optimization, with 53 Features, XGB achieves RMSE $22,456 and is slightly better than Time-Based Split GBR.
- It is found that there is huge performance drop from Train Data to Test Data. for example Training RMSE is $8,492

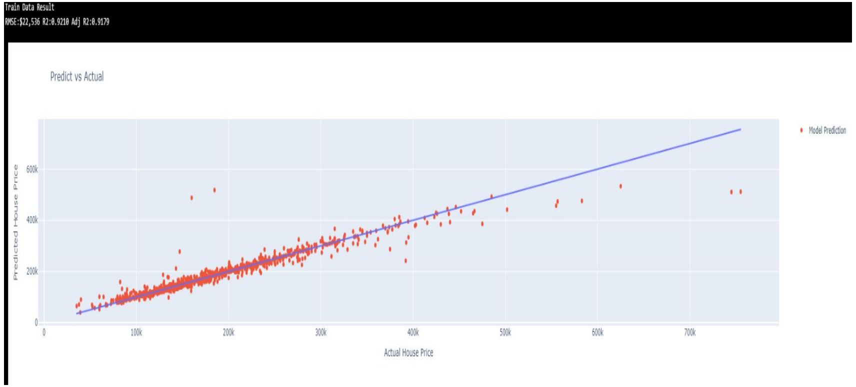


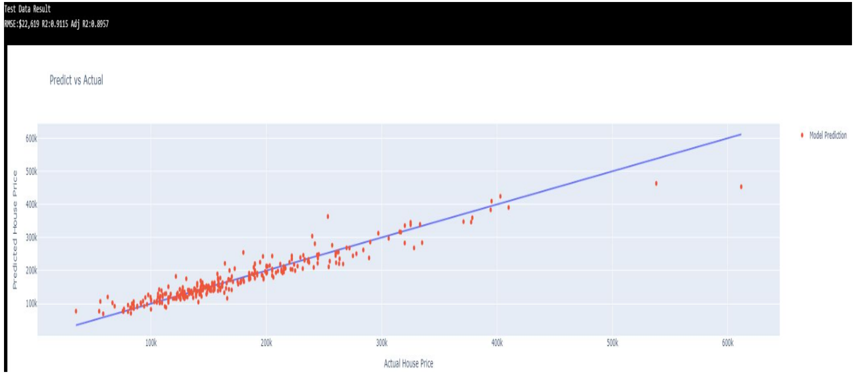


Time-Based Split XGR Model also Meets The RMSE Criteria

- After optimization, with 44 Features, XGB achieves RMSE $22,619 and is slightly worse than Time-Based Split GBR.
- However, there performance between Train and Test Data are very close, unlikely has overfitting







Conclusion

- XGB Regression Model performs better than GBR no matter in either data split methods.
- **Time-Based Split XGB Regression has comparable performance in both Train and Test Data. This might indicate that it is more generalized, recommend to use this model;**
- Random Split XGB Regression has the best performance, this can be used too but should monitor with care for overfitting risk since its Train Data are fitted much better than Test Data
- In all models, the features importance ranking are same. Based on the result, the most critical factors affecting house price are:
  - ‘OverallQual’: Rates the overall material and finish of the house.
  - ‘GrLivArea’: Above grade (ground) living area square feet.
  - ‘GarageCars’: Size of garage in car capacity.

Model	Random Split					Time-Based Split				
	Feature #	Train		Test		Feature #	Train		Test	
		RMSE	Adj. R2	RMSE	Adj. R2		RMSE	Adj. R2	RMSE	Adj. R2
GBR	105	$21,532	0.9217	$25,365	0.8693	53	$22,671	0.8913	$24,142	0.9050
XGBR	53	$8,493	0.9880	$22,456	0.9036	44	$22,536	0.9179	$22,619	0.8957