DATA 621 – FINAL PROJECT PRESENTATION

Group # 4 (Juan Falck,Al Haque,Euclides Rodriguez, Melvin Matanos)

OUR MAIN RESEARCH QUESTION

Can you accurately predict house prices?

DATASET TO BE USED

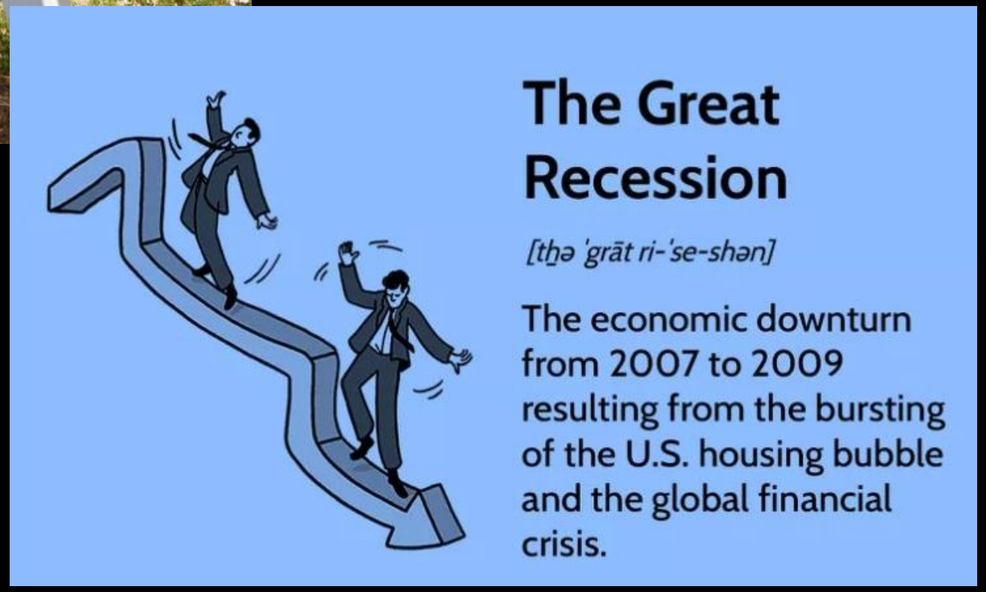
We will use for this analysis the Kings County dataset, which has over 21,000 rows of house sales

Is predicting house prices closer to a science or a gamble?

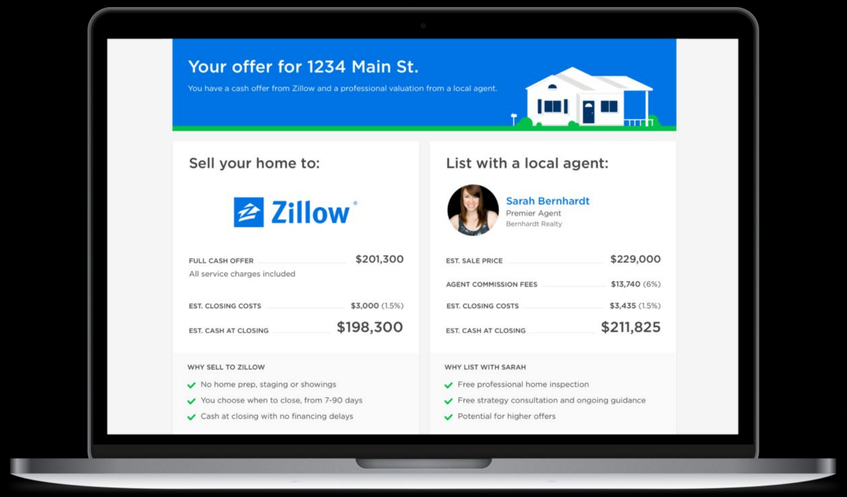
What history has taught about predicting house prices

The 2008 financial crisis taught us many things, one of them was that predicting house prices is a risky gamble. But is it?

Is the issue that predicting house prices is inherently risky and difficult? Or that people make risky decisions with the information derived from prediction models?



More recent case of Zillow offers (2021)



Bloomberg Businessweek

Businessweek | Real Estate

Zillow Wants to Flip Your House

A new breed of high-tech real estate flippers is using algorithms (and a healthy dose of Silicon Valley venture capital) to buy at massive scale.

By Patrick Clark
February 14, 2019, 5:00 AM EST

The New York Times

Daily Business Briefing ›

Zillow, facing big losses, quits flipping houses and will lay off a quarter of its staff.

The real estate website had been relying on its algorithm that estimates home values to buy and resell homes. That part of its business lost about $420 million in three months.

Roughly 10 years after the financial crisis of 2008 Zillow the largest Real State company in the US, tried also to get into the business of predicting House prices and failed.

Again the question is if price prediction is in inherently volatile, imprecise, therefore risky ?

With so much information available on house sales, how come it has been notoriously difficult to get it right?

HOW ACCURATE CAN WE GET PREDICTING

Using the 21 features in the dataset we want to see how good price predictions can we get. Using a test dataset can we get accurate enough?

MODEL COMPARISON

Do we see a significant performance difference (in terms of prediction power) between models, or all basic regression produce similar results with no model being outlier (positive negative

VARIABLE ANALYSIS

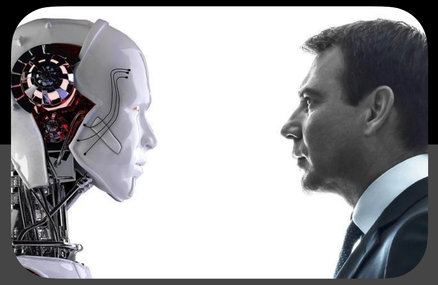
What variables (if any) are the most important ones for our regression model?
Can we engineer variables that improve the results of our predictions?

Specific questions to be researched

Methodology we will follow

COMPARE SEVERAL REGRESSION MODELS

We will run several models on the exact same training and test data and we will compare them using metrics like RMSE and R-squared



FEATURE ENGINEERING

Although dataset is fairly rich in number of features, we will attempt to engineer new features and test if we get better performance out of them

ANALYZE VOLATILITY

Not only we will look at how good our model is prediction, but how sensitive it is with fluctuations of some of the variables. If we change one variable by let's say 10%, how much our model fluctuates

VARIABLE IMPORTANCE

Which variables are more important for our prediction model? Do the make sense from what we know as humans who live in a house? What happens to our model if that important variable is no longer available?

Glimpse of the Dataset

- The dataset contains 21,613 observations with 21 features
- Each observation contains information about each individual house (price,id,latitude,longitude, yr_built,zip_code,condition,etc..)
- Each column is an integer or numeric column, some columns contain discrete values while others contain continuous values
- Dataset doesn’t contain any empty data
- Target variable Price is right skewed

Data Cleaning:

Before we begin modeling we will first clean the data, we will remove unnecessary columns and convert the appropriate columns into factors some of these columns are floors,waterfront,view,waterfront, condition, and etc. we will categorize the year_built values of the houses to help reduce the number of variable levels. We will create a new column called “yr_built_bin”. Houses built during the 2000s will be considered as “newly built”, houses built after 1950 are considered “medium built” and houses built before 1950 are “old built”. We will create another new column called “yr_renovated_bin” where we will categorize houses renovated into two bins, houses renovated before 2000 will be considered as “recent renovation” and house renovated after 2000 will be “not recent renovation”.Finally, we conclude our data-cleaning efforts by combining the relevant columns into a new dataset

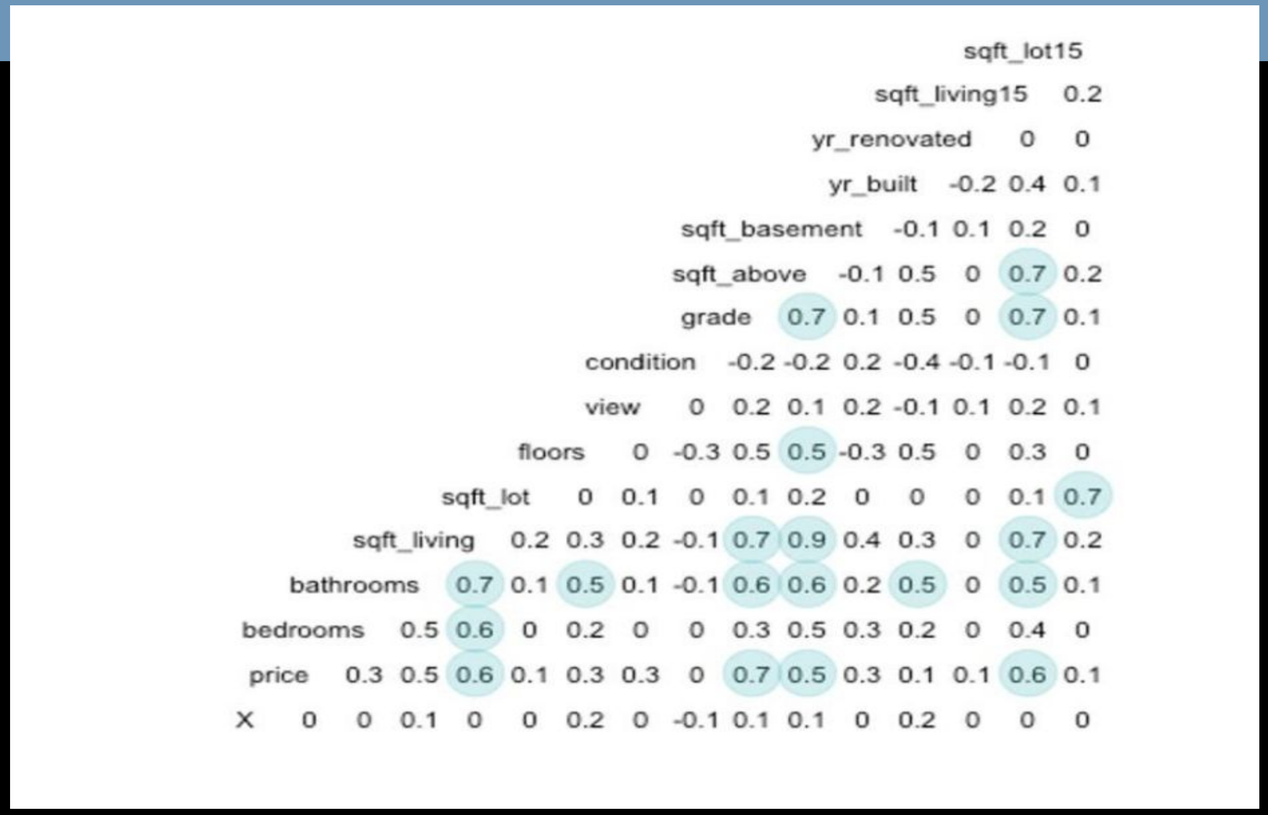
```
df$floors <- as.factor(df$floors)
df$waterfront <- as.factor(df$waterfront)
df$view <- as.factor(df$view)
df$condition <- as.factor(df$condition)
df$grade <- as.factor(df$grade)
df$yr_built_bin <- as.factor(df$yr_built_bin)
df$yr_renovated_bin <- as.factor(df$yr_renovated_bin)
```

```
df <- df %>% mutate(yr_built_bin = case_when(
  yr_built >= 2000 ~ "new build",
  yr_built >=1950 & yr_built <2000 ~ "medium build",
  yr_built < 1950 ~ "old build"
  ))

df <- df %>% mutate(yr_renovated_bin = case_when(
  yr_renovated >= 2000 ~ "recent renovation",
  yr_renovated < 2000 ~ "not recent renovation"
  ))
```

Correlation Matrix

A quick glimpse of the correlation matrix between variables, the correlation matrix allows us to see pairwise correlations between each variable. We can see some correlations with prices and bedrooms.



													sqft_lot15	
												sqft_living15	0.2	
											yr_renovated	0	0	
										yr_built	-0.2	0.4	0.1	
									sqft_basement	-0.1	0.1	0.2	0	
								sqft_above	-0.1	0.5	0	0.7	0.2	
							grade	0.7	0.1	0.5	0	0.7	0.1	
						condition	-0.2	-0.2	0.2	-0.4	-0.1	-0.1	0	
					view	0	0.2	0.1	0.2	-0.1	0.1	0.2	0.1	
				floors	0	-0.3	0.5	0.5	-0.3	0.5	0	0.3	0	
			sqft_lot	0	0.1	0	0.1	0.2	0	0	0	0.1	0.7	
		sqft_living	0.2	0.3	0.2	-0.1	0.7	0.9	0.4	0.3	0	0.7	0.2	
	bathrooms	0.7	0.1	0.5	0.1	-0.1	0.6	0.6	0.2	0.5	0	0.5	0.1	
	bedrooms	0.5	0.6	0	0.2	0	0	0.3	0.5	0.3	0.2	0	0.4	0
	price	0.3	0.5	0.6	0.1	0.3	0.3	0	0.7	0.5	0.3	0.1	0.1	0.6	0.1
X	0	0	0.1	0	0	0.2	0	-0.1	0.1	0.1	0	0.2	0	0	0

Approach # 1 Multiple Linear Regression

We will create three linear regression model, we will evaluate the model based on the Root mean squared value and R-squared metric since they are easy to understand.The first model will be predicting the price by using all the predictors we will call this model, ***model_all,*** our second model predicts housing prices based on correlated predictors which are (sqft_living,grade,sqft_above,sqft_living15..) . We also tried a backward stepwise regression which yielded a model with almost all predictors with an adjusted r-squared value of 0.6502 and an RMSE of 163102.7. Overall our multiple linear regression approach struggles to make accurate predictions, but the high r-squared value explains approximately 60% of the variation in the data.

```
model_all <- lm(price ~ ., data = house)
## Residual standard error: 217200 on 17330 degrees of freedom
## Multiple R-squared:  0.6505, Adjusted R-squared:  0.6502
## F-statistic:  2151 on 15 and 17330 DF,  p-value: < 2.2e-16
```

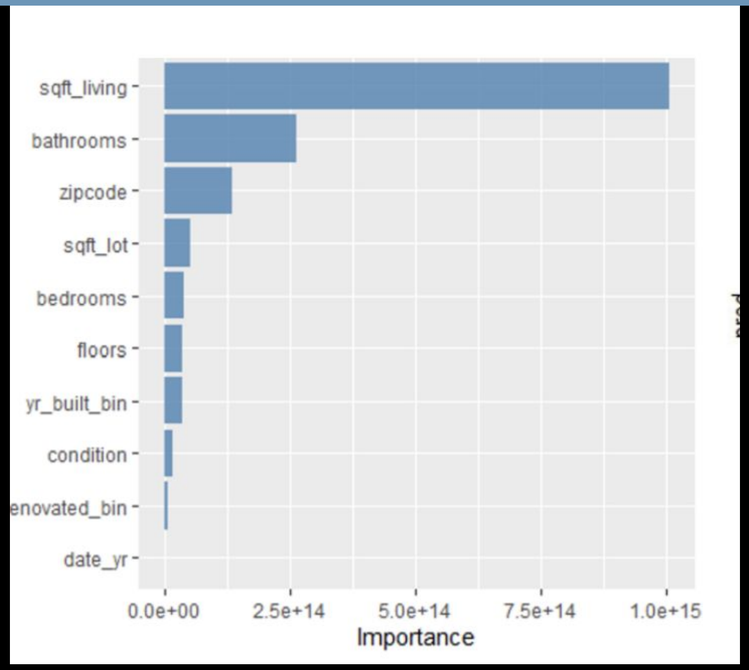
```
summary(lm(price ~ sqft_living + grade + sqft_above + sqft_living15,
           data = house))
## Residual standard error: 249400 on 17341 degrees of freedom
## Multiple R-squared:  0.5389, Adjusted R-squared:  0.5388
## F-statistic:  5068 on 4 and 17341 DF,  p-value: < 2.2e-16
```

Approach # 2 Decision Trees

```
collect_metrics(final_rs)

## # A tibble: 2 x 4
##   .metric .estimator  .estimate .config
##   <chr>   <chr>           <dbl> <chr>
## 1 rmse    standard   228944.    Preprocessor1_Model1
## 2 rsq     standard        0.586 Preprocessor1_Model1
```

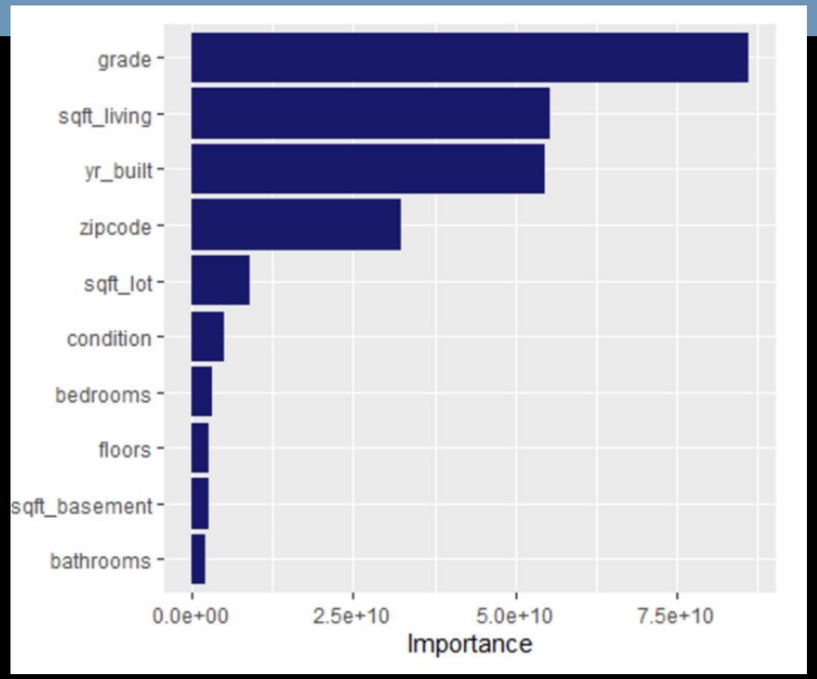
We will next use decision tree to predict house prices. We will use tidy models in R to select predictors and tune parameters to achieve the best R-squared and RMSE values. We will split the dataset into training and testing sets, and stratify the sample by price to ensure equal variation of prices in both sets. Next, we'll create 10-fold cross-validations to resample the training set, into 10 folds to prevent resampling with an equal proportion of prices. Then, We let the cost_complexity, tree_depth and min_n parameters to be tuned by the Tidy Models package to optimize predictions. After that, we set the engine to rpart, the mode to regression. and specified a tree grid with 4 levels. We got an R-squared value of 0.586 and an RMSE of 228944 (top-right) which is not a good fit as indicated by the actual vs. predicted(bottom-right.). We can also see the important predictors that arose in the models which are **(sqft_living,bathrooms and zipcodes.)**

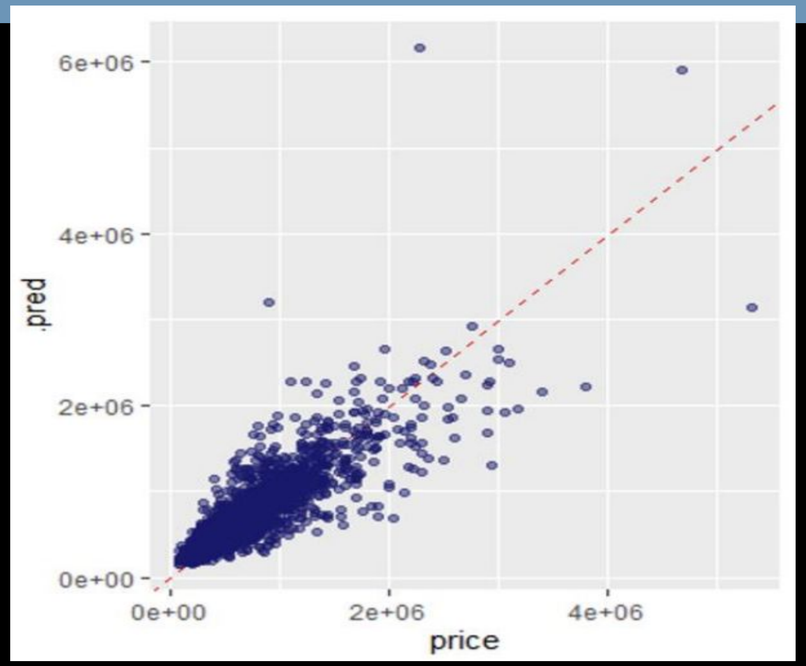


Approach # 3 Random Forest

```
## # A tibble: 2 × 4
##   .metric .estimator  .estimate .config
##   <chr>   <chr>           <dbl> <chr>
## 1 rmse    standard   173029.    Preprocessor1_Model1
## 2 rsq     standard        0.775 Preprocessor1_Model1
```

Next, we will try the random forest engine, We use and set the ranger package for the engine, set the mode to regression, the number of trees to 1000, and set the number of candidate points to 11. Selecting the metric for the best model we get an RMSE of 173029 and an R-squared value of 0,775 a much better improvement than the decision tree model. Though the RMSE is still high, it is lower than the one in the decision tree and with a higher R Squared. The important predictors used in Random Forest were **grade,sqft_living,yr_built,and zipcode.**



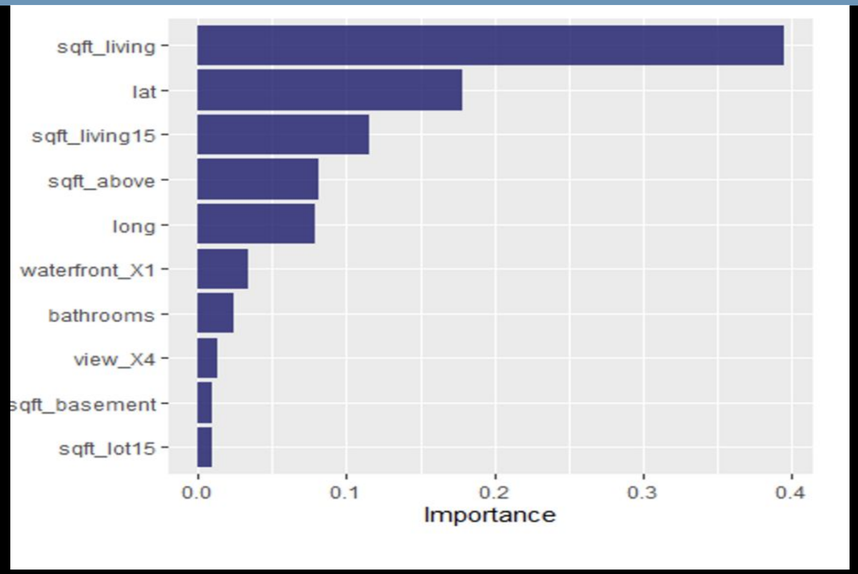


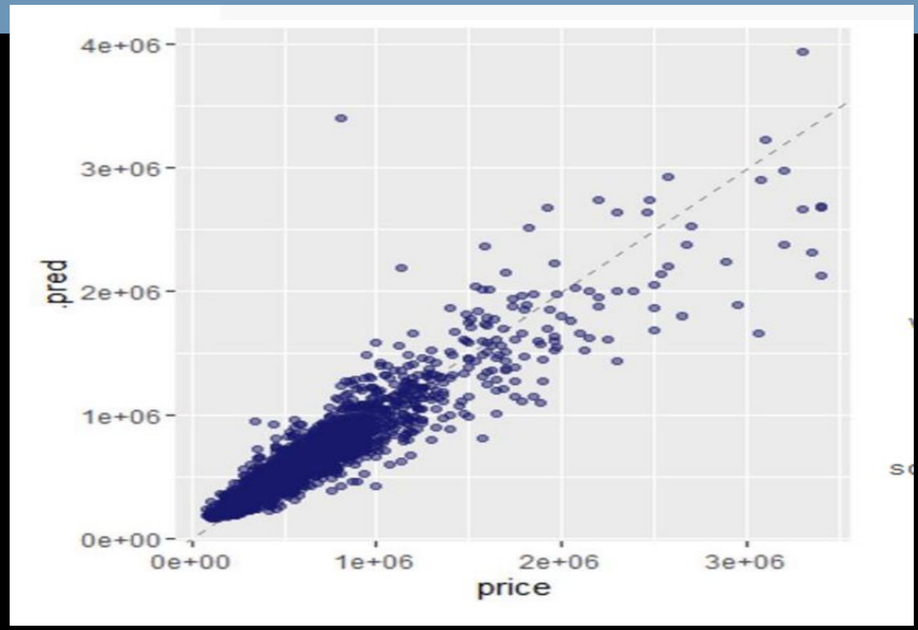
Approach # 4 xGBoost

```
collect_metrics(house_fit)

## # A tibble: 2 x 4
##   .metric .estimator  .estimate .config
##   <chr>   <chr>           <dbl> <chr>
## 1 rmse    standard   124007.    Preprocessor1_Model1
## 2 rsq     standard        0.874 Preprocessor1_Model1
```

Finally, the xGboost algorithm We will let Tidy models to tune the tree_depth,min_n, sample_size,mtry, and learn_rate while we set the tre to 1000. We set the engine to xgboost, the mode to regression, and the xgb_grid size to 30. After running the model and gaining the met we get the R-squared value is 0.87 and the RMSe is 124007 which was an improvement from both the decision tree and random forests model. The most important predictors in the xgboost models were **sqft_living,lat, and sqft_living15.**





Conclusion:

In conclusion, predicting house prices is difficult, for our initial effort using multiple linear regression we see that the models we've created are not a good fit for the data with high RMSE and low R-squared value. However, once we started implementing the decision tree and other ensemble method was the fit getting more accurate, however, once we implemented random forest and xgboost we were able to reduce the RMSE and get a high R-squared even though the diagnostics were overplotted **(decision tree top left, random forest bottom left, xgboost middle right..).** For now using decision tree and ensemble methods are a good start to accurate price prediction

```
collect_metrics(final_rs)

## # A tibble: 2 x 4
##   .metric .estimator  .estimate .config
##   <chr>   <chr>           <dbl> <chr>
## 1 rmse    standard   228944.    Preprocessor1_Model1
## 2 rsq     standard        0.586 Preprocessor1_Model1
```

```
## # A tibble: 2 x 4
##   .metric .estimator  .estimate .config
##   <chr>   <chr>           <dbl> <chr>
## 1 rmse    standard   173029.    Preprocessor1_Model1
## 2 rsq     standard        0.775 Preprocessor1_Model1
```

```
collect_metrics(house_fit)

## # A tibble: 2 x 4
##   .metric .estimator  .estimate .config
##   <chr>   <chr>           <dbl> <chr>
## 1 rmse    standard   124007.    Preprocessor1_Model1
## 2 rsq     standard        0.874 Preprocessor1_Model1
```

Thank you!