House Price Prediction ML Project

Jason & Alister

Team 22

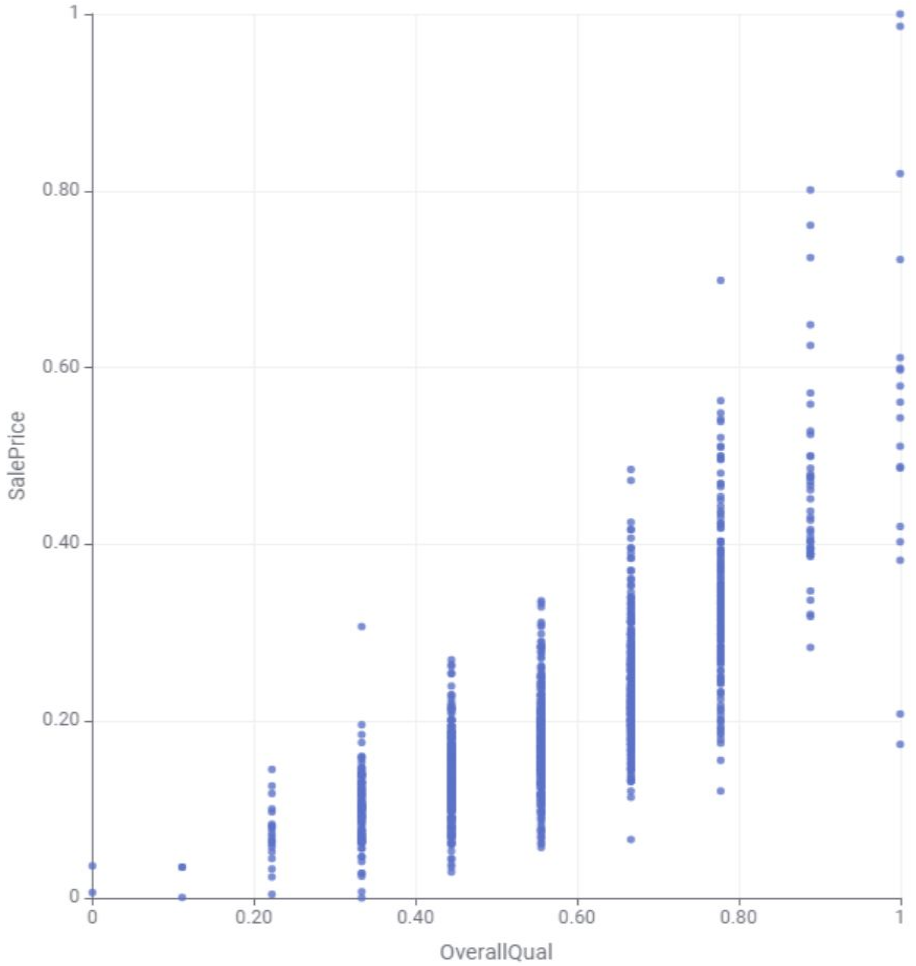
Datasets

Goal: Use machine learning techniques to make prediction on housing prices in Ames

- Train.csv: To train the Machine Learning models
- Test.csv: Use the test data to make predictions from the trained Machine Learning Models

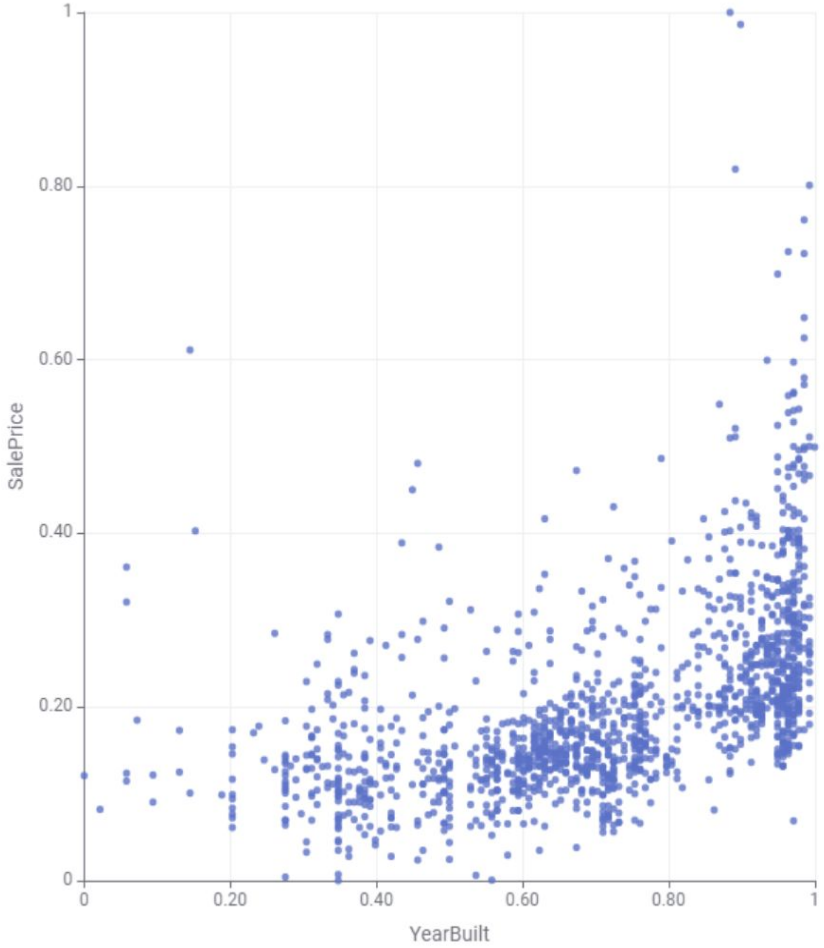
SalePrice vs OverallQuality

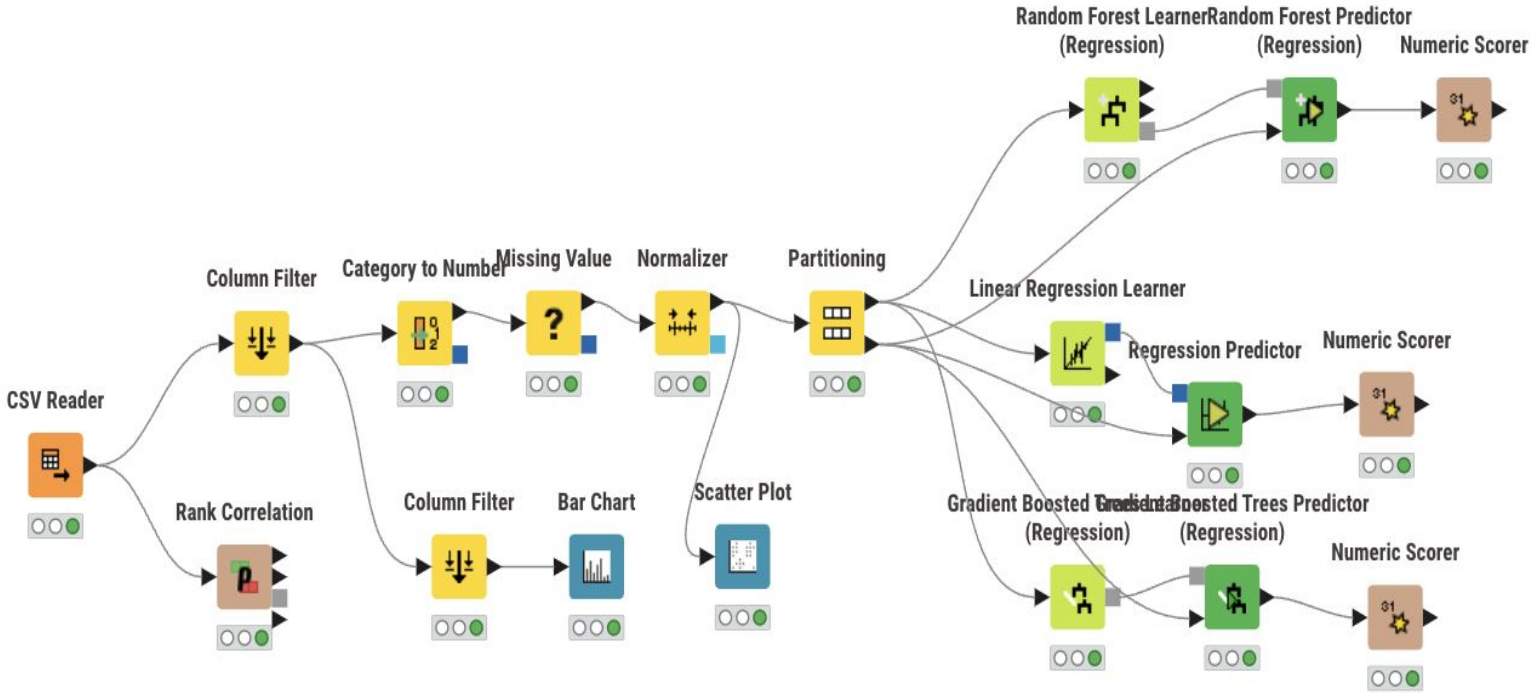
The higher the OverallQual, the higher the SalesPrice of the houses (a few potential outliers)



SalePrice vs YearBuilt

- The newer the houses, the higher the prices of the house
- Consistent in most years




CSV Reader
Column Filter
Category to Number
Missing Value
Normalizer
Partitioning
Random Forest Learner (Regression)
Random Forest Predictor (Regression)
Numeric Scorer
Linear Regression Learner
Regression Predictor
Numeric Scorer
Rank Correlation
Column Filter
Bar Chart
Scatter Plot
Gradient Boosted Trees Learner (Regression)
Gradient Boosted Trees Predictor (Regression)
Numeric Scorer

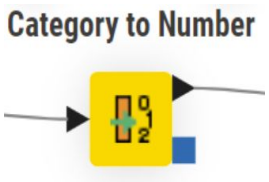
Steps by Steps

Data Loading and Preprocessing

- Read and load the training and testing datasets

Select only ‘useful’ columns and clean up any missing values

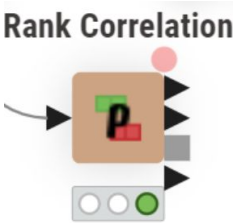
- **Column Filter:** Only includes columns (MSZoning, LotArea, OverallQual, OverallCond, YearBuilt, YearRemodAdd, SalePrice)
- **Category to Number:** Convert all the categorical variables to numerical
- **Missing Value:** Fill up all the null numerical values with the median



How to determine the right columns for training?

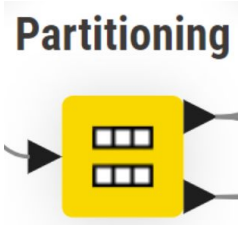
Rank Correlation

- Calculate the correlation coefficient for each pair of selected columns
- We selected LotArea, OverallQual, OverallCond, YearBuilt, MsZoning, YearRedData



#	RowID	Id Number (dou...	MSSubCl... Number (dou...	MSZoning Number (dou...	LotFronta... Number (dou...	LotArea Number (dou...	Street Number (dou...
1	Id	1	0.019	-0.023	0.006	-0.005	0.009
2	MSSubClass	0.019	1	0.128	-0.161	-0.27	-0.017
3	MSZoning	-0.023	0.128	1	-0.184	-0.22	0.036
4	LotFrontage	0.006	-0.161	-0.184	1	0.286	0.025
5	LotArea	-0.005	-0.27	-0.22	0.286	1	-0.054
6	Street	0.009	-0.017	0.036	0.025	-0.054	1
7	Alley	-0.003	0.046	-0.336	-0.011	-0.085	-0.002
8	LotShape	0.03	0.07	0.147	-0.23	-0.311	-0.011
9	LandContour	-0.016	-0.003	0.005	0.031	-0.081	0.12
10	Utilities	0.013	-0.03	-0.006	0.037	0.036	0.002
11	LotConfig	0.047	0.045	0.018	-0.118	-0.196	0.012
12	LandSlope	0.019	-0.02	-0.028	0.053	0.117	-0.176
13	Neighborhood	-0.012	-0.009	-0.206	0.056	0.093	-0.01
14	Condition1	-0.012	-0.013	-0.069	0.058	0.088	-0.037
15	Condition2	0.03	-0.03	0.016	0.034	0.047	-0
16	BldgType	0.019	0.654	0.104	-0.25	-0.426	-0.049
17	HouseStyle	0.017	0.533	-0.116	0.095	0.061	0.022
18	OverallQual	-0.029	0.108	-0.212	0.072	0.233	0.059

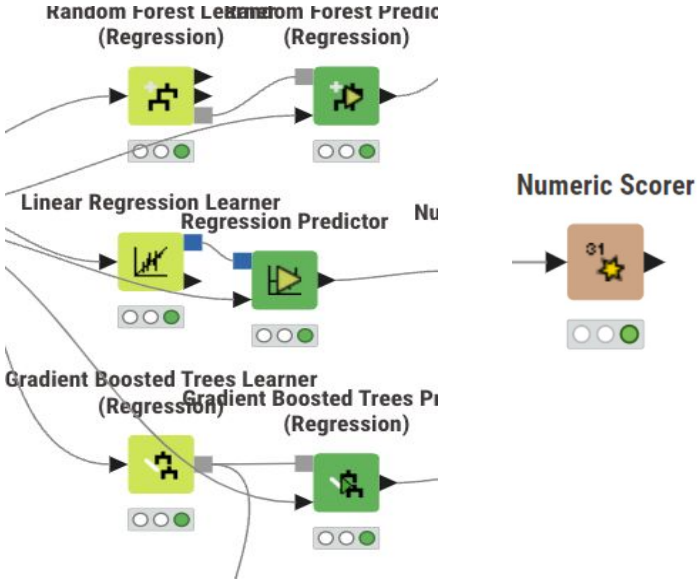
Split the training and testing



- Split the training dataset by 80/20 ratio for training and testing purposes

Model building and Model Evaluation

- Used Linear Regression, Random Forest, Gradient Boosting learner and then predictor and then scored each model (Accuracy, MSE, etc) using Numeric Scorer



Model Evaluation Results

Linear Regression

RowID	Prediction (SalePrice) *Number (double)*
R^2	0.697
mean absolute error	31,505.586
mean squared error	1,884,819,759.405
root mean squared error	43,414.511
mean signed difference	-1,707.75
mean absolute percentage error	0.181
adjusted R^2	0.697

Random Forest

owID	Prediction (SalePrice) *Number (double)*
^2	0.783
ean absolute error	25,377.376
ean squared error	1,353,254,298.268
ot mean squared error	36,786.605
ean signed difference	-366.173
ean absolute percentage error	0.14
djusted R^2	0.783

Gradient Boost

RowID	Prediction (SalePrice) *Number (double)*
R^2	0.759
mean absolute error	26,802.378
mean squared error	1,500,698,081.133
root mean squared error	38,738.845
mean signed difference	-453.361
mean absolute percentage error	0.146
adjusted R^2	0.759

Conclusion

- **Random forest** had the best performance
- We selected random forest for our kaggle submission

Future work: Use other models (SVM, Model Ensemble) and perform hyperparameter tuning and adjust the features

Thank You

Question