Ames Iowa Machine Learning

December 15, 2023

Nick Herman
Charlotte Wolf
Chui Pereda

Agenda

Key Objectives and Background

Exploratory Data Analysis (EDA)

Lasso and Multiple Linear Regression (MLR)

Gradient Boosting

Other Models

Key Findings and Future Work

Key Objectives

- Predict price for home buyers and sellers
- Determine home attributes impacting price
- Insights into the market in Ames, Iowa

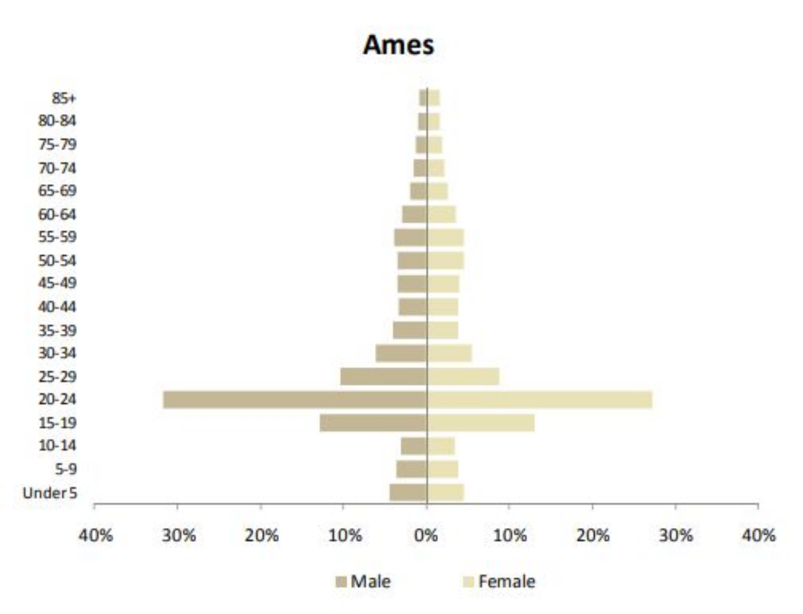
Background

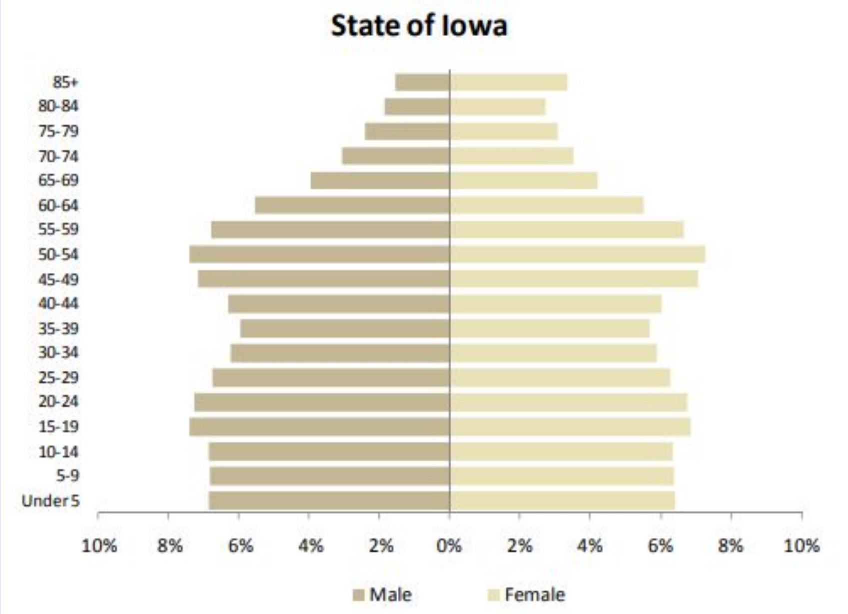
Data set covers sales from 2006 to 2010

Demographics: Students, professors

According to Ames Economic Development Commission, ISU is the largest employer with over 10,000 employees, or about ⅙ of the total population

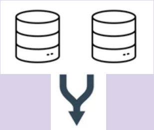
Demographics





Exploratory Data Analysis

Data Cleaning and Preprocessing



Merge

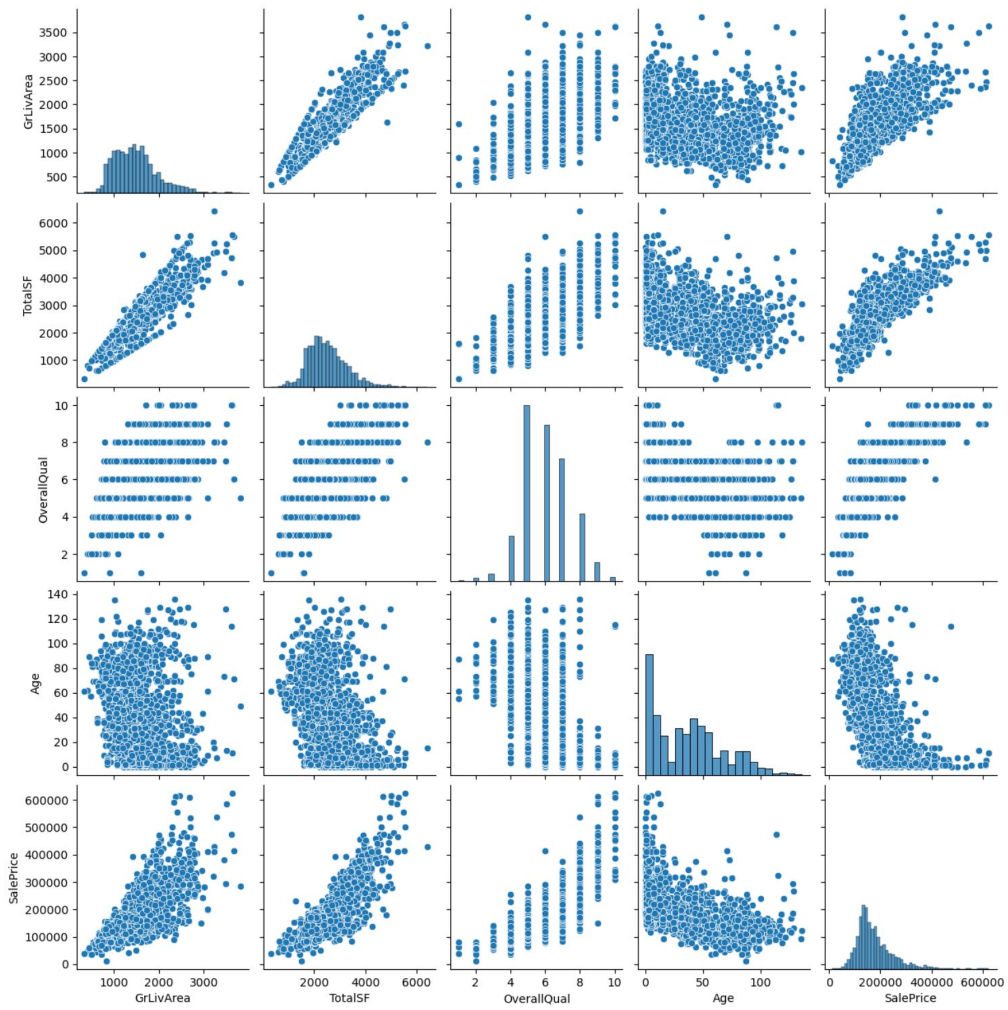
Housing with Real Estate data

Features Added

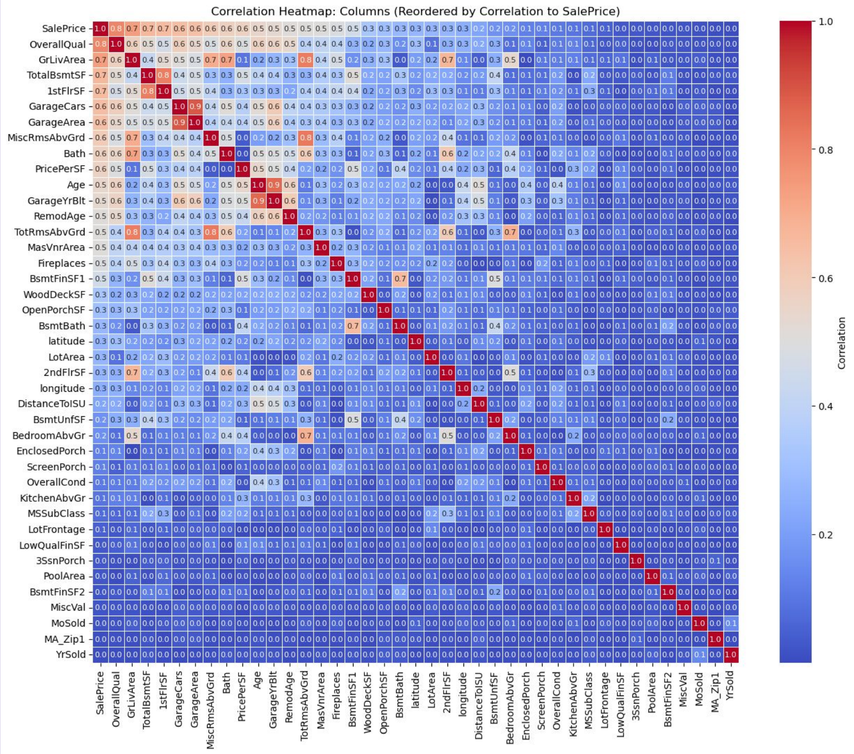
Latitude, Longitude, DistancetoISU, DistanceCategory, TotalSF

Features Dropped

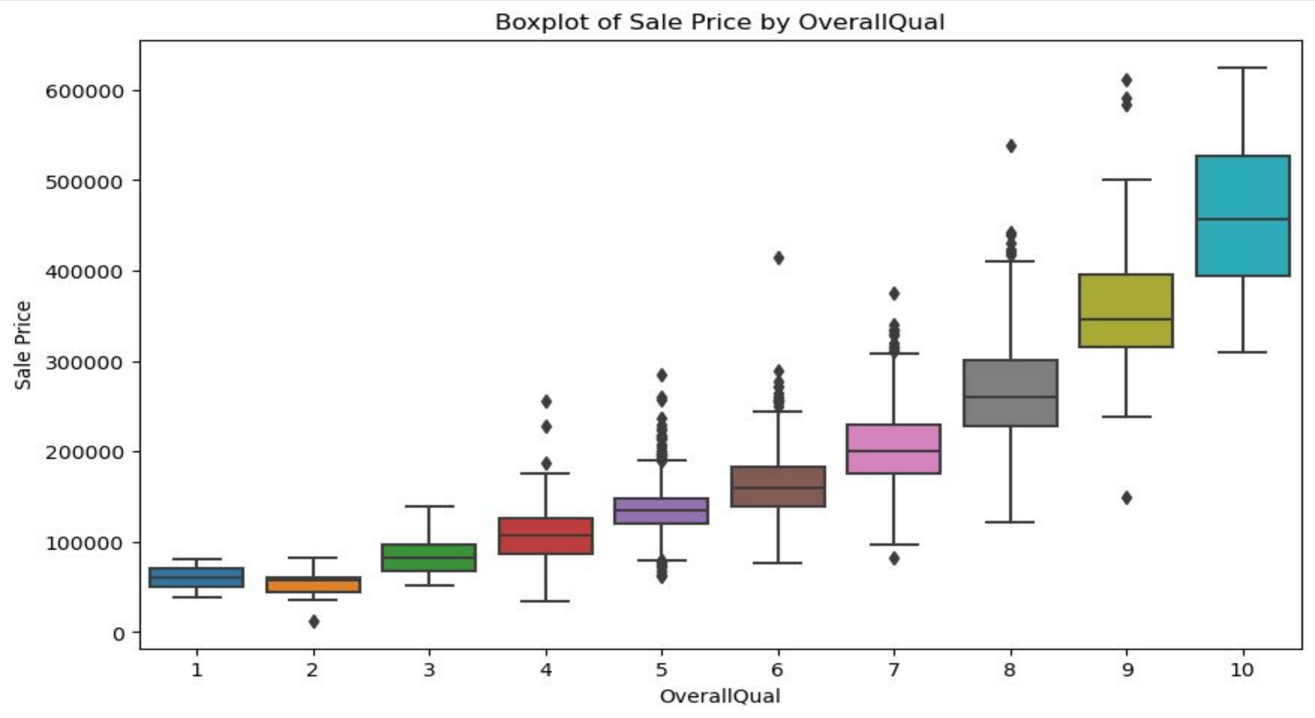
PID, GeoRefNo, Prop_Addr, Utilities


GrLivArea
TotalSF
OverallQual
Age
SalePrice

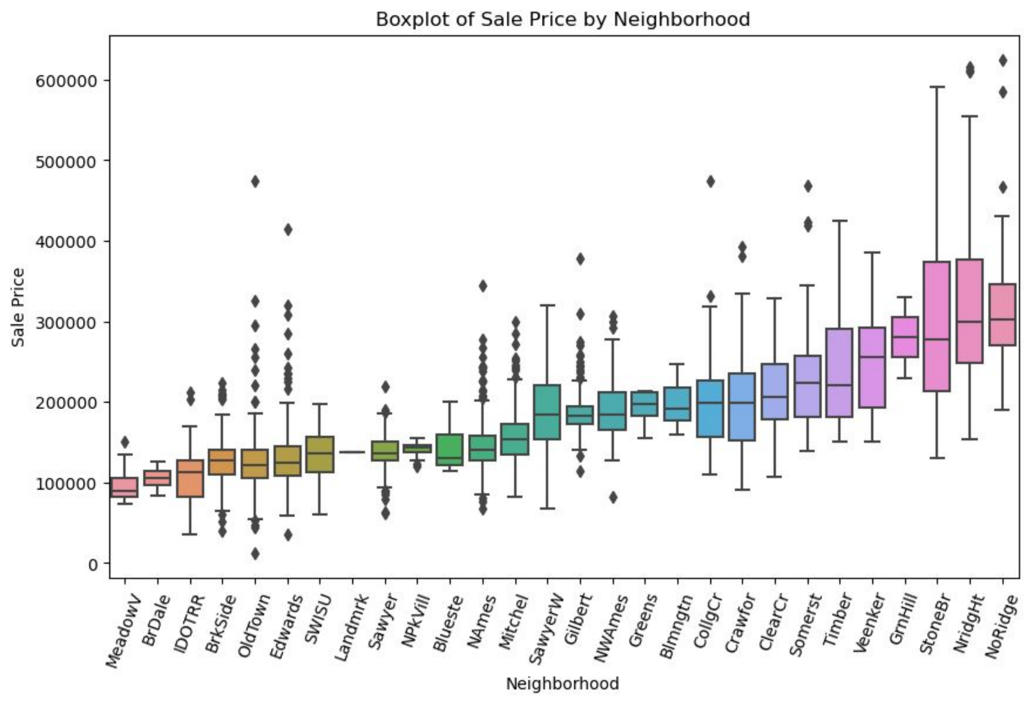
Feature Correlation



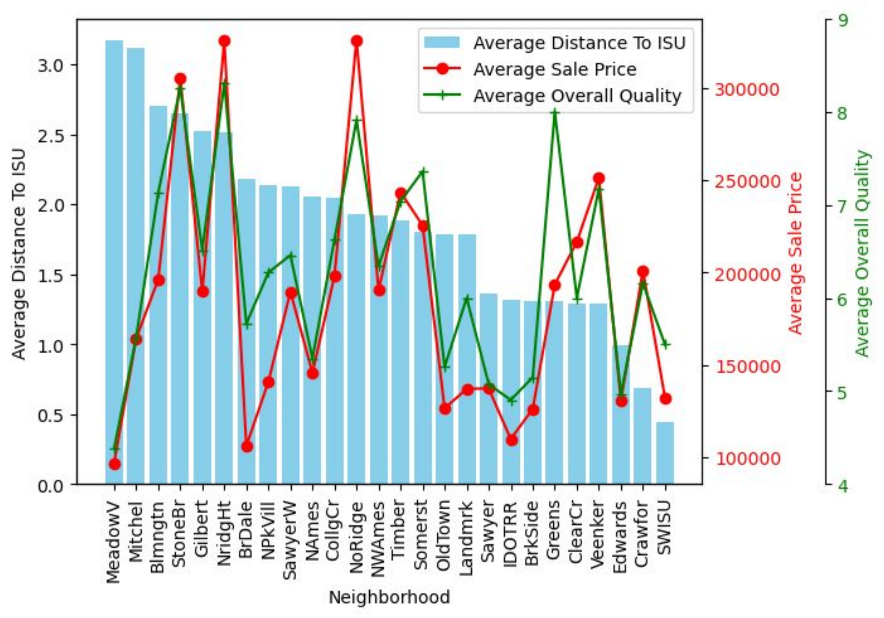
Sale Price and OverallQual

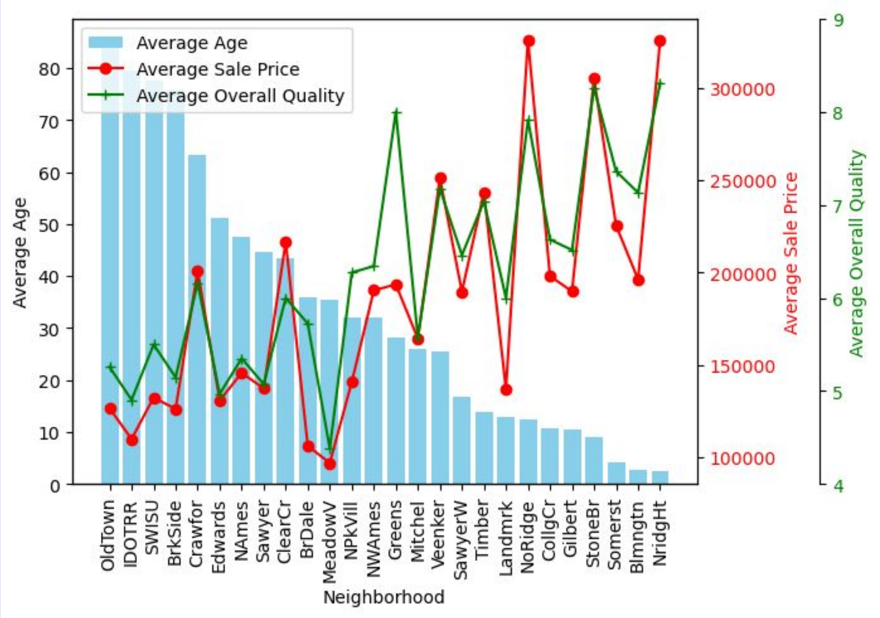


Sales Price and Neighborhood

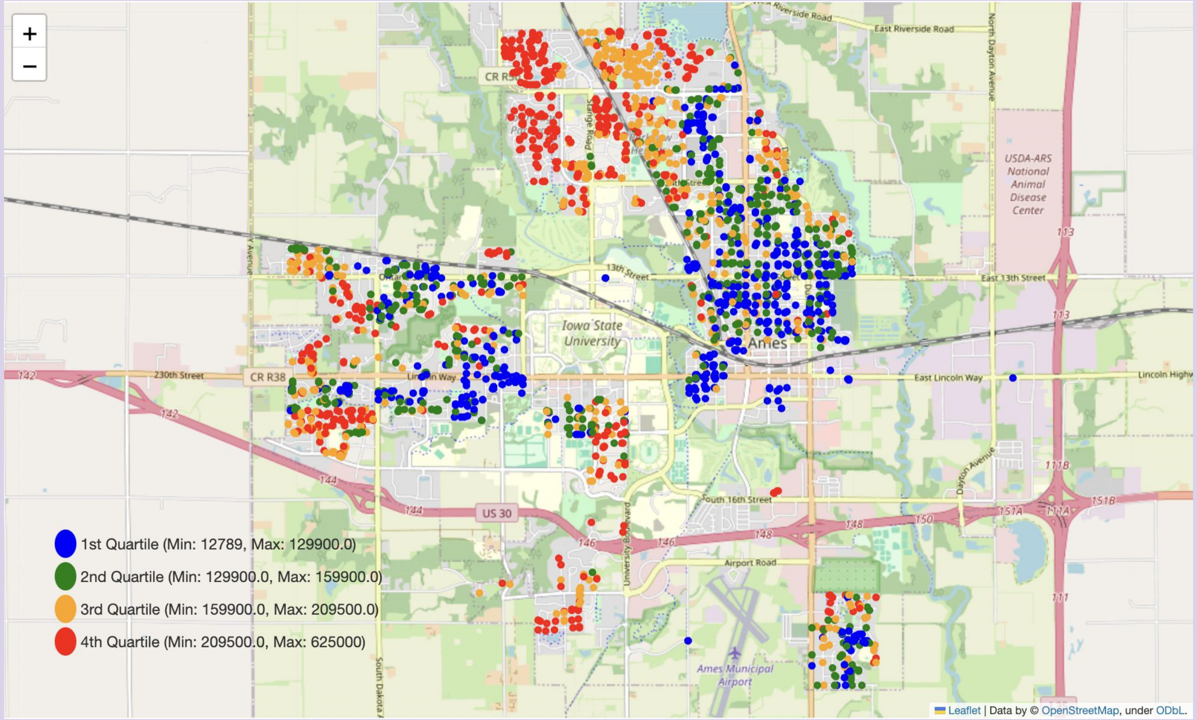


Sales Price and Neighborhood

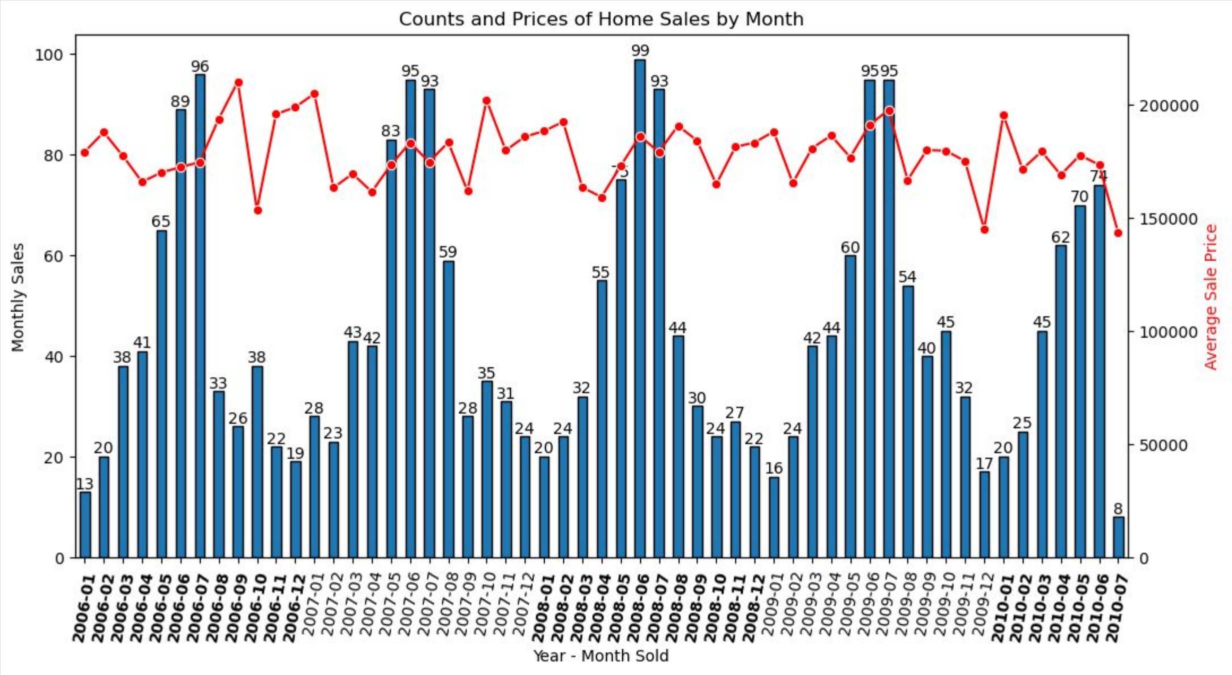




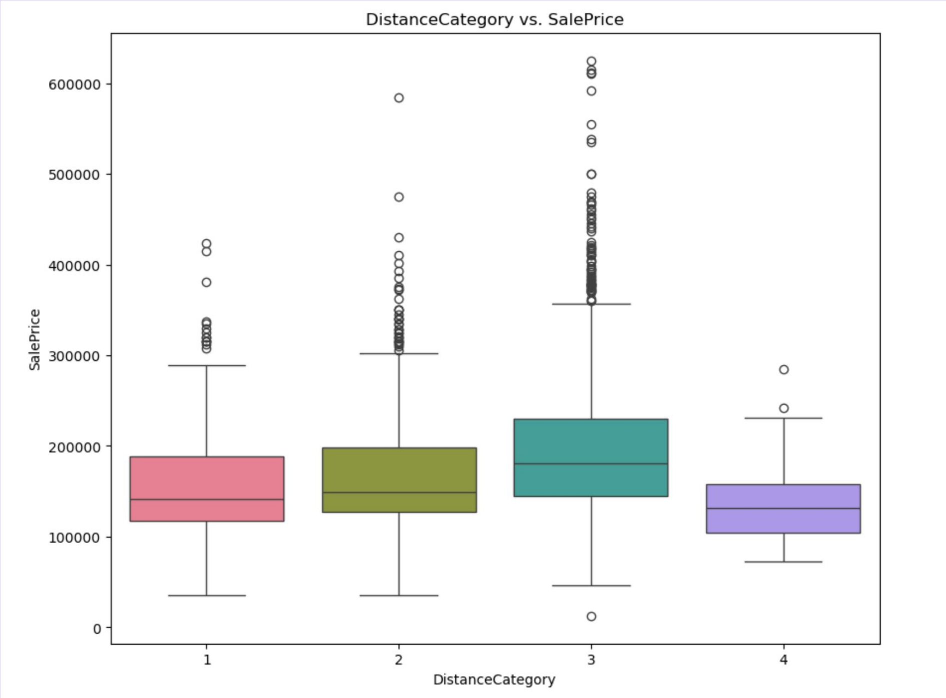
Sales Price and Location



Sales Price and Seasonality



Sales Price and Distance from ISU



Lasso and MLR

Lasso Regression - Handling nominal, ordinal, numerical features

All features needed to be an int or float type before modeling.

Nominal

- Dummified for linear models. First category was **not** dropped.
- NaN filled with “missing”

Ordinal

- mapped. Missing was 0, Poor 1, Fair 2, etc

Numerical

- NaN filled with mean/mode/median

Lasso Regression Round 1

Standard Scaler

- Ensures each feature contributes equally to computation

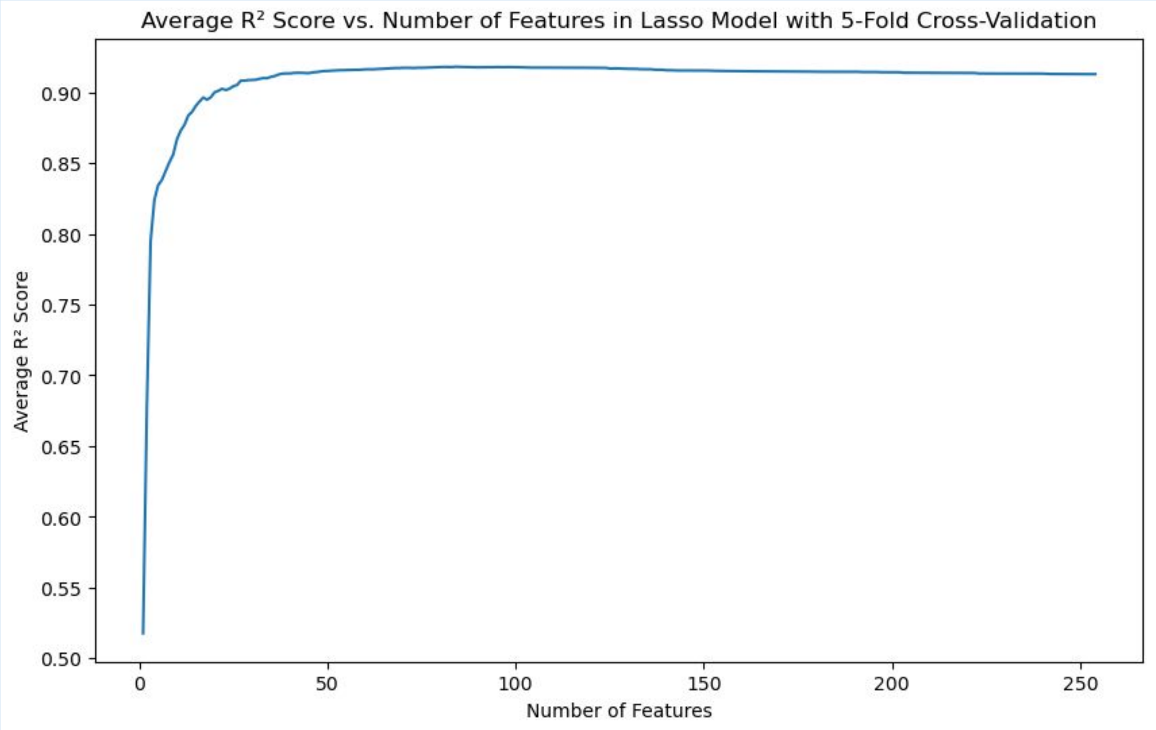
K-fold cross-validation

- reliable estimate of the model’s performance
- Train test splits have a high variance depending on how the split is made. K-fold reduces this by averaging performance over folds

Base lasso model with ALL features + no tuning

- Mean r2 = .8886, Mean RMSE = 24666.8431

R^2 Changes as More Features Are Included



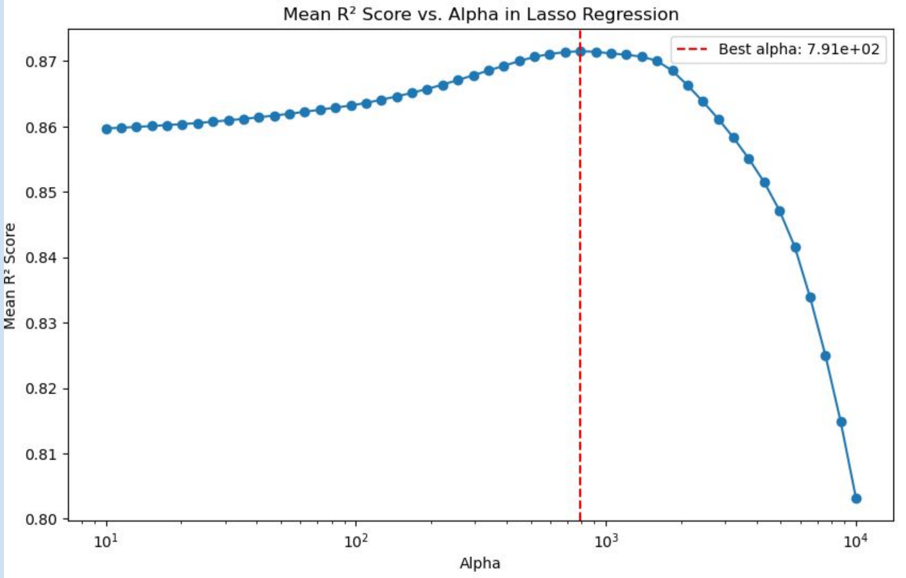
Lasso Tuning

GridSearchCV used to determine best alpha = 790

- features shrunk to 0 (192)
- non-zero coeff features (84)

GridSearchCV with non zero coeff features

- Best alpha = 271



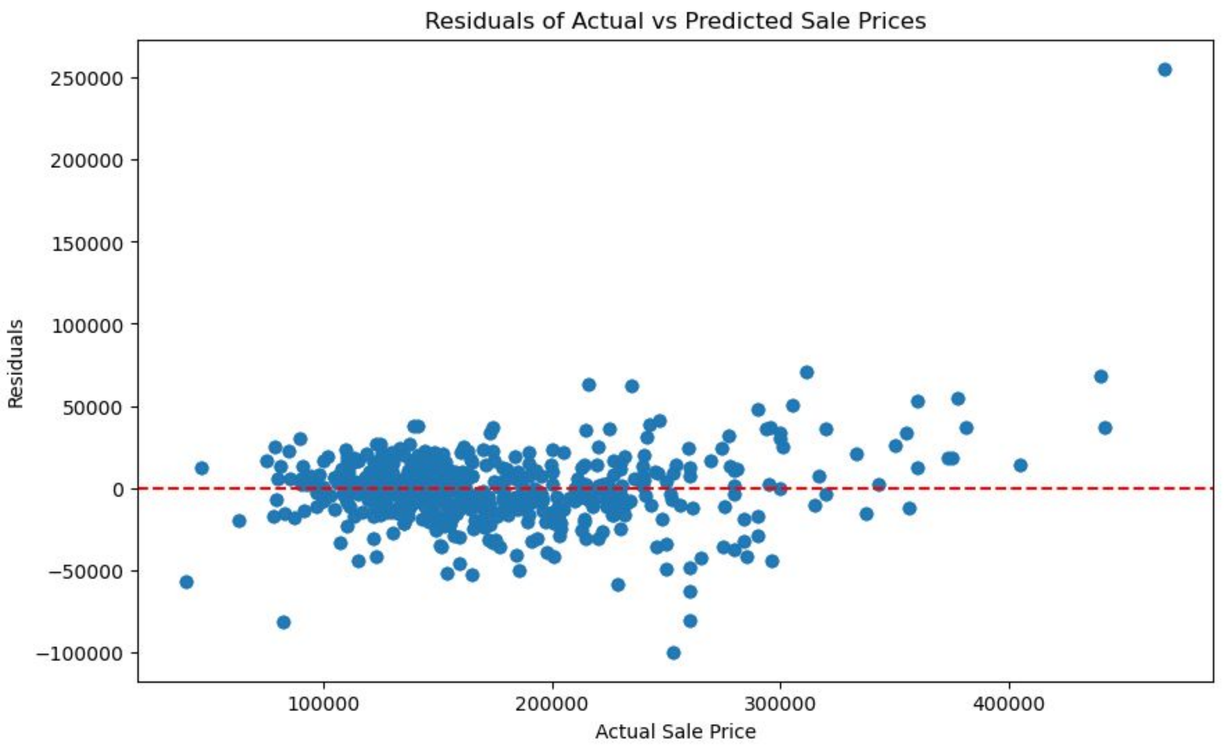
Lasso Regression Round 2

Best alpha + non-zero coeff features

- Mean R2= .8939
- Mean RMSE= 24106.9580

Outliers removed

- Mean R2=.9442
- Mean RMSE=15483



Multiple Linear Regression Round 1

Preprocessing differences:

- NO standard scaler so the coefficients are interpretable
- For dummified categorical variables, drop_first was used

Same 49 non-zero features were used.

- Mean R2=.8881

Multiple Linear Regression Round 2

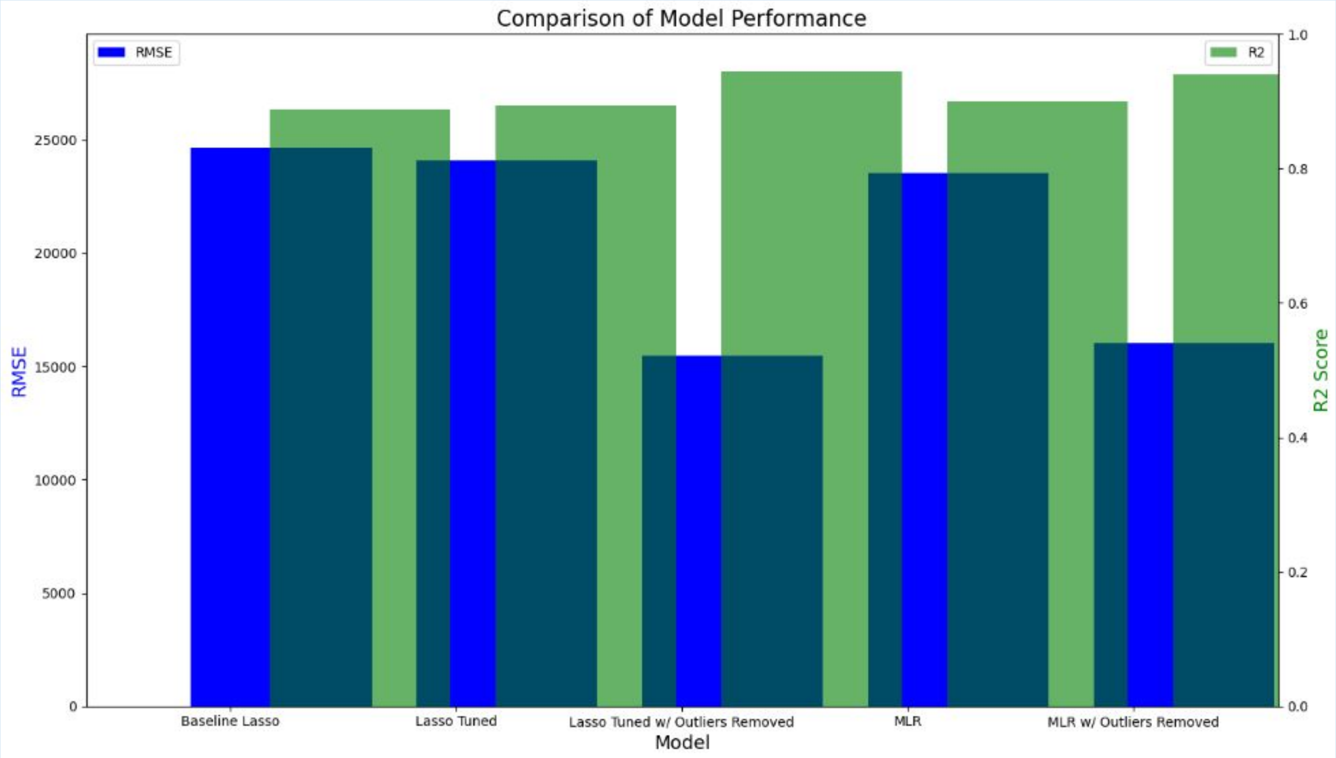
Non-zero features removed if they do not increase R2

- 28 features kept
- Mean R2 = .8998, RMSE= 23557.2429

Same outliers were removed

- Mean R2 = .9403, RMSE=16012.39.09

Comparison of Models



Linear Model Insights

Interpretable results

- Switch wood shake to wood shingles to increase property by $43k
- Switch “other” to wall heating can increase property value by $46k

Outliers

- Opportunity to buy and sell. Buy low and sell high.

Feature	Average Coefficient
GrLivArea	47.326946
OverallQual	8004.122468
YearBuilt	359.111089
MasVnrArea	21.307342
BsmtFinSF2	9.538167
BedroomAbvGr	-5003.002423
OverallCond	5450.969043
ScreenPorch	24.280998
TotalSF	11.755731
BsmtFinSF1	20.695211
GarageArea	18.654225
LotArea	0.576059
Fireplaces	2736.805119
HeatingQC	1486.802328

Gradient Boosting Models

Missing/Incorrect Values

Each feature was reviewed and the best method for imputing the missing values was determined. Some of these solutions included:

- The mean/mode value from the feature was used
- Some missing/impossible year values were filled with the year the house was built
- Some numerical features such as zip code were converted to categorical with the NA's filled with 'Other'
- When specific features related to each other a map was used to calculate the average ratio of one feature to another based on unique combinations. The ratios were used to calculate missing values
- Categorical features had their NA's replaced with "None"

Handling Nominal, Ordinal, & Numerical Features

Ames Dataset: 2577 observations, 81 features

(37) Numerical features were left as is

(17) Ordinal categorical features were mapped to numerical values starting at 0 for ‘none’ or whatever the lowest value was for that feature

(27) Nominal categorical features were also mapped starting at 0 in an arbitrary order. Changing the values that these were mapped to did not change the model scores

Model Selection

After data cleaning and linear modeling, many models were tested (with their default parameters) to see which would be the best to investigate further.

We decided to work on a model using Gradient Boosting Regressor

Model	R2	MSE	MAE
GradientBoostingRegressor	0.92703	399600557.78222	13176.81917
HistGradientBoostingRegressor	0.92454	413669045.81594	12858.33696
ExtraTreesRegressor	0.91723	454027408.12039	13649.25736
XGBRegressor	0.91551	458607222.70555	13972.52447
RandomForestRegressor	0.90901	496327398.72313	14427.60776
Lasso	0.89256	582028189.80295	16652.67575
Linear Regression	0.89253	582175274.44554	16656.33405
Ridge	0.89210	584595134.87604	16703.01480
ElasticNet	0.88496	624847494.15185	16917.58553

Grid Search

After broad experimentation with the various parameters one at a time, the ranges for each was narrowed down and a using the GridSearchCV function with the following parameter ranges the best values were found

```
param_grid = {
    'learning_rate': [0.008, 0.01, 0.012],
    'subsample': [0.2, 0.3, 0.4],
    'n_estimators': [5000, 6000, 7000],
    'max_depth': [3, 4, 5, 6],
    'min_samples_split': [3, 5, 7],
}
```

```
Best Parameters: {'learning_rate': 0.008, 'max_depth': 3, 'min_samples_split': 5,
'n_estimators': 6000, 'subsample': 0.3}
Best R-squared: 0.941400438914043
R-squared on Test Data: 0.9495
Elapsed Time: 34569.19 seconds
```

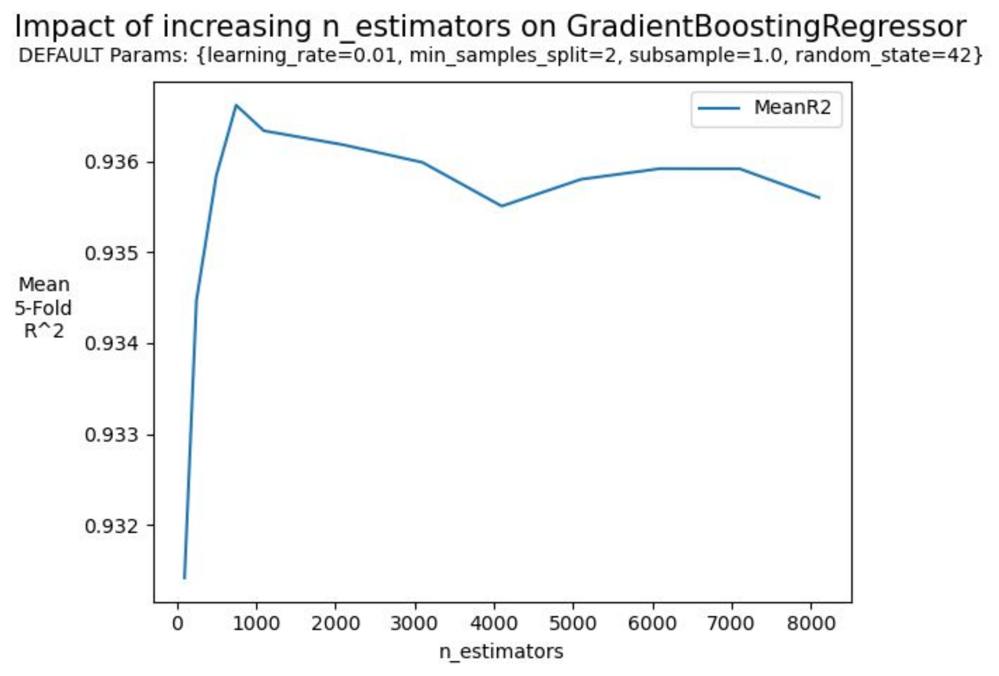
Analysis on both dummified and numerical dataframes

While dummification is not necessary for a tree based model, analysis was run on both a dataframe with the categorical columns treated as previously described (called df_numerical) as well as the version where those columns were dummified (called df_with_dummies).

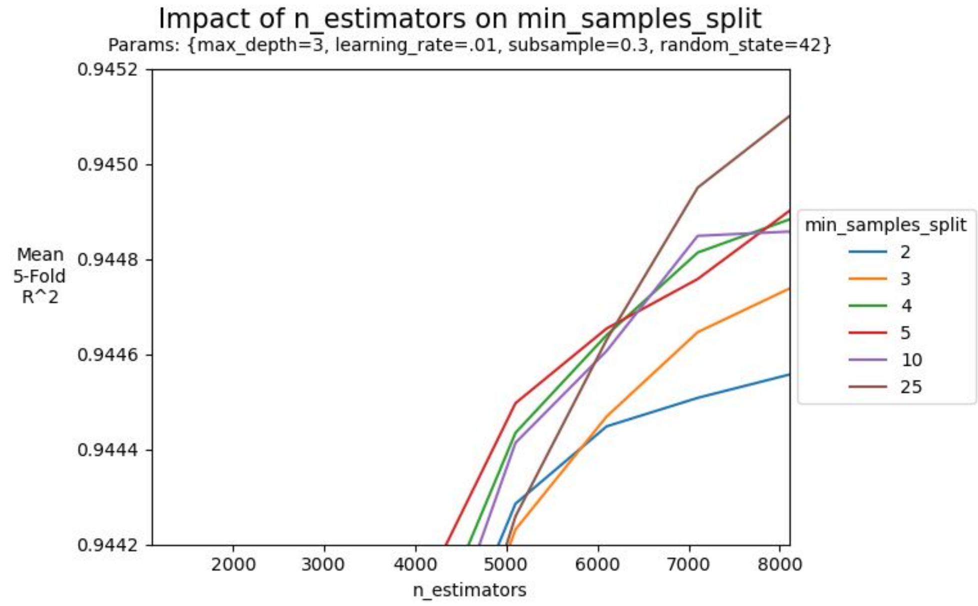
While there is minimal impact on the mean R2 score on a 5-fold cv, we decided to continue improving our model based on the df_numerical.

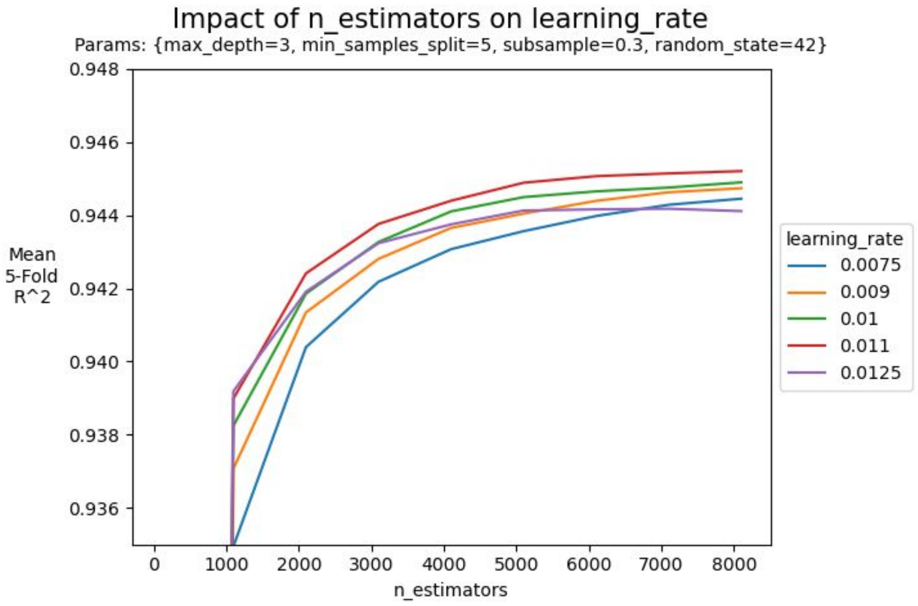
	DataFrame	Hyperparameters	Fold 1 R2	Fold 2 R2	Fold 3 R2	Fold 4 R2	Fold 5 R2	MLR Mean R2	Standard Deviation
0	df_numerical	best for numerical	0.94646	0.93912	0.95445	0.94875	0.93245	0.94424	0.00768
1	df_with_dummies	best for numerical	0.94787	0.94144	0.95584	0.94682	0.92937	0.94427	0.00875
2	df_numerical	best for dummies	0.94283	0.93457	0.94317	0.94421	0.92979	0.93892	0.00572
3	df_with_dummies	best for dummies	0.94387	0.93625	0.94926	0.94060	0.92350	0.93870	0.00871

Parameter Tuning - Null Model

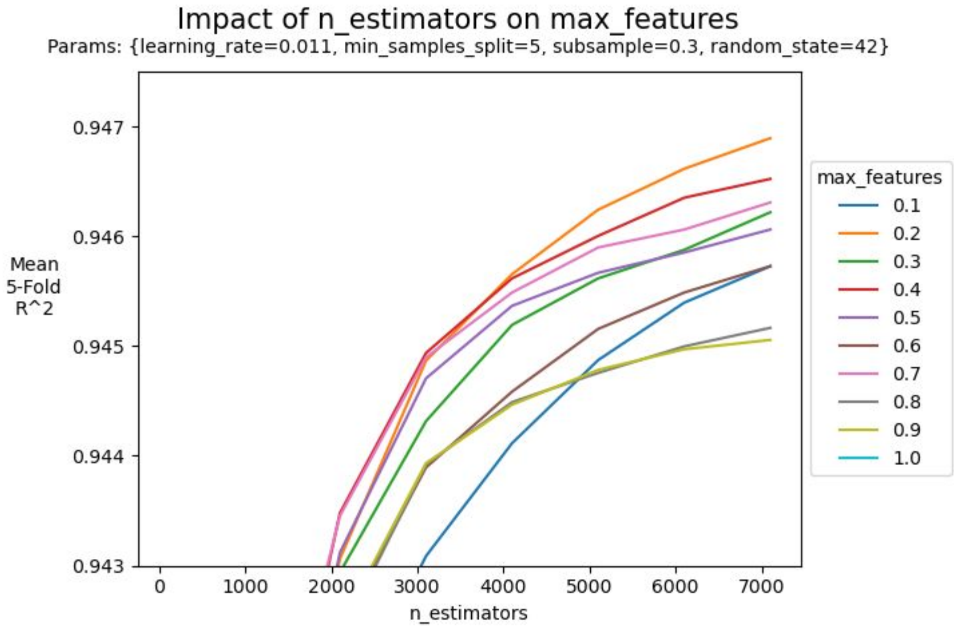


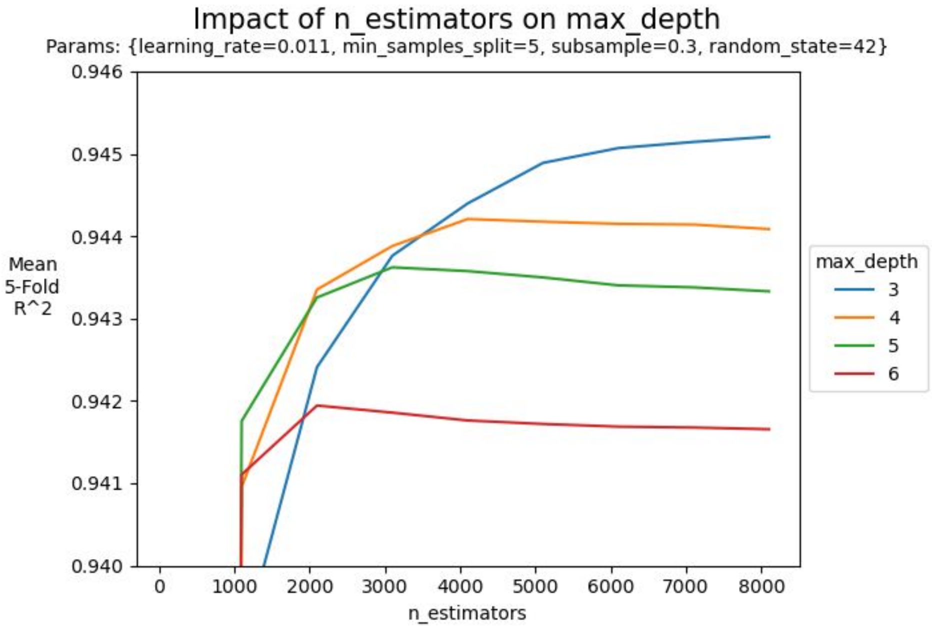
Specific parameters were tuned on a loop to see their performance changed with increasing estimators





Specific parameters were tuned on a loop to see their performance changed with increasing estimators



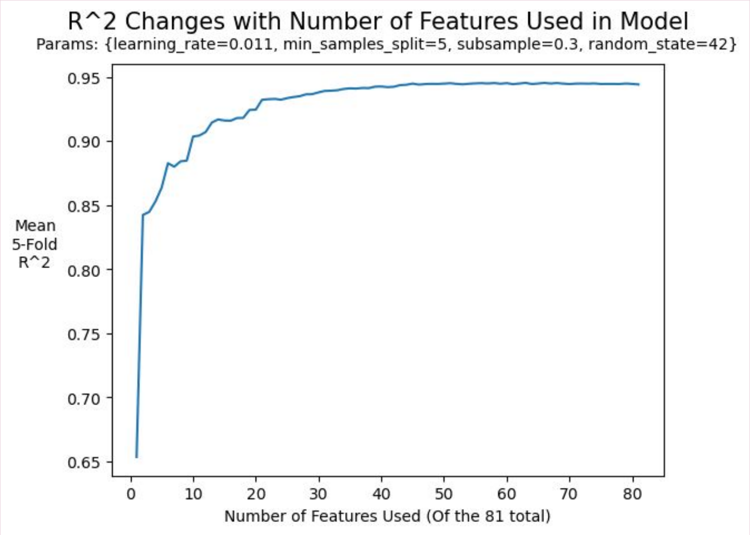


Feature Selection

Once the model had been tuned, the most important features were determined using the GradientBoostingRegressor feature_importances_ attribute. These were the top 10 features:

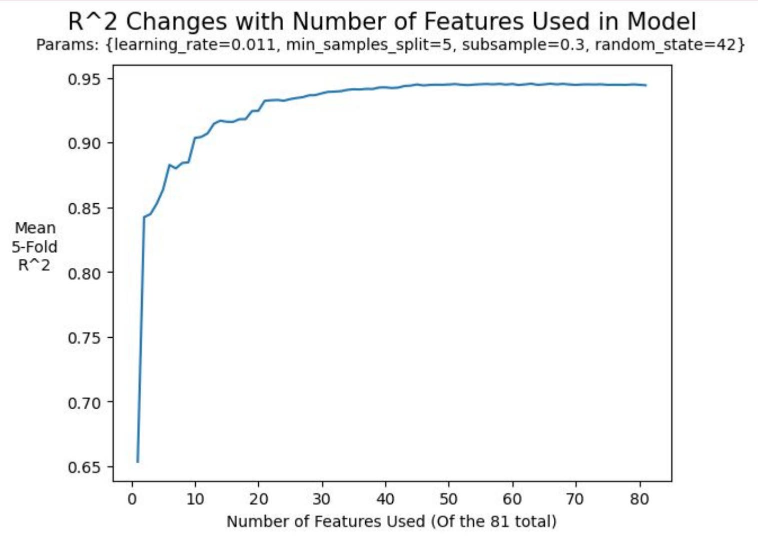
1. 'TotalSF',
2. 'OverallQual',
3. 'GrLivArea',
4. 'ExterQual_n',
5. 'GarageCars',
6. 'Age',
7. 'TotalBsmtSF',
8. 'KitchenQual_n',
9. 'GarageArea',
10. 'BsmtFinSF1'

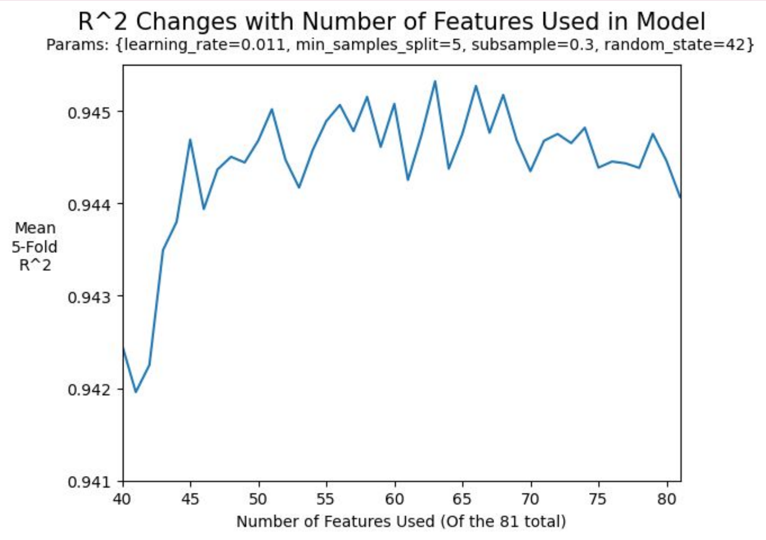
The features were added one by one to a model tuned with the best hyperparameters to track how adding features impacted R^2



Feature Selection

While this result seems to indicate that the R^2 of the model always improves with more features included, when zoomed in we can see that this is not exactly so:





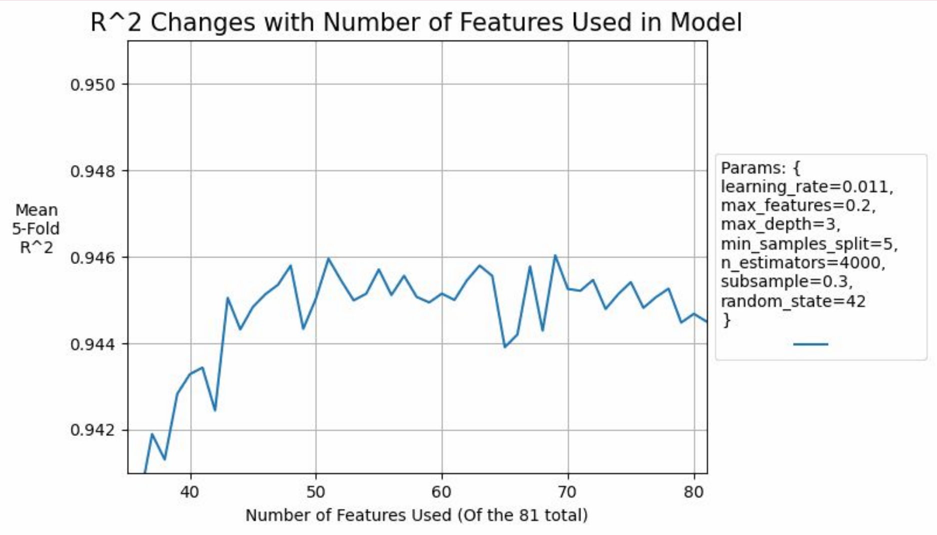
Feature Selection

Removed features:

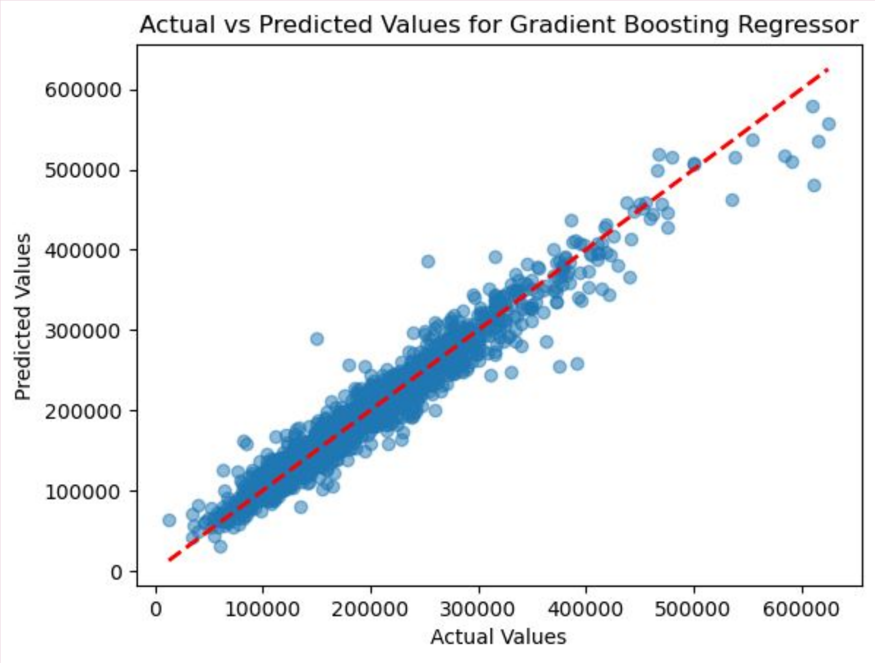
- MasVnrArea
- MiscRmsAbvGrd
- LotShape_n
- SaleType
- Foundation
- EnclosedPorch

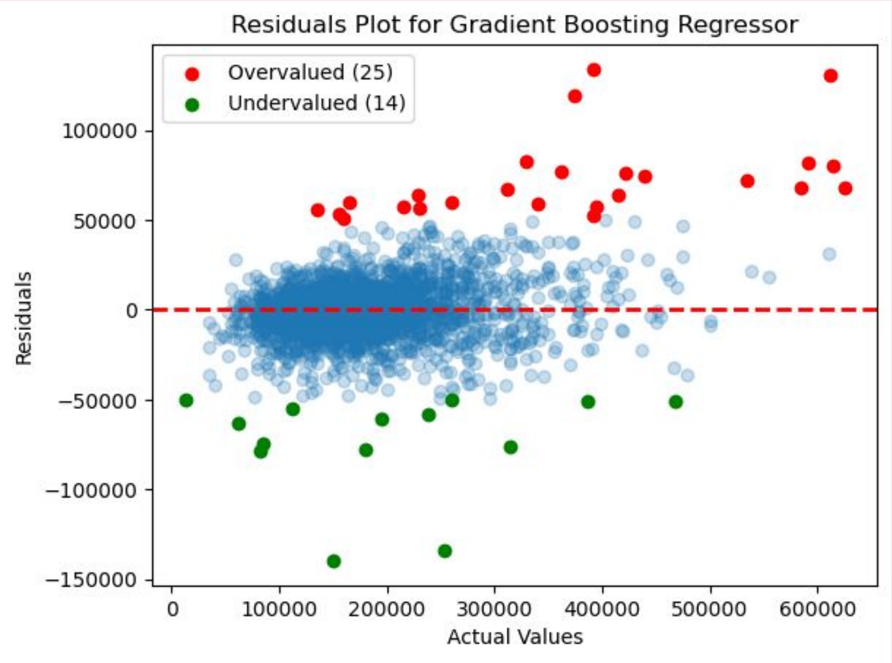
Max R^2 Score = **0.94917** at the 51 most important features included

Max R^2 Score = **0.92703** with default parameters and all features included



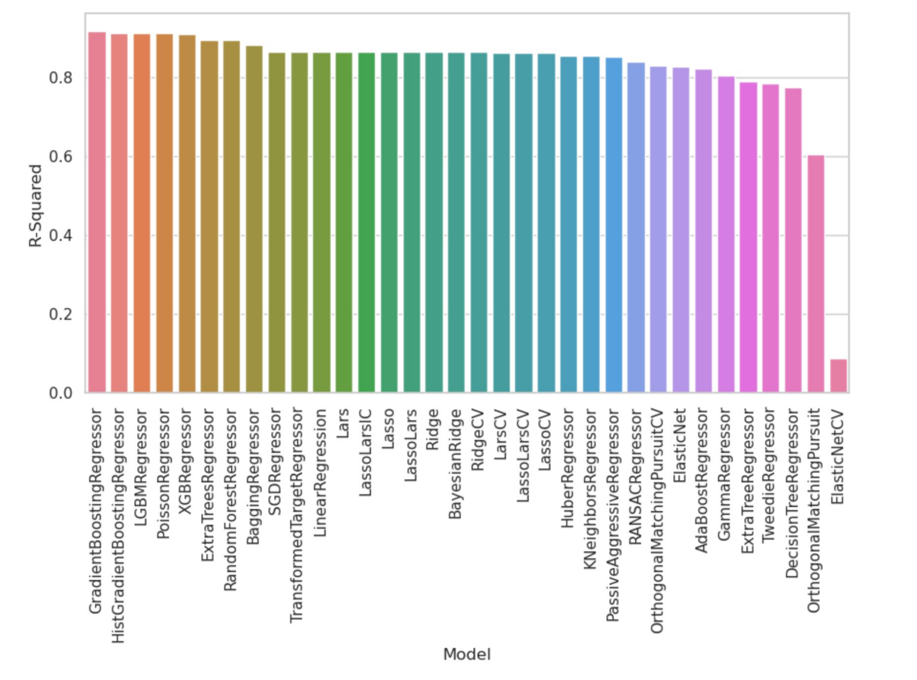
Model Accuracy





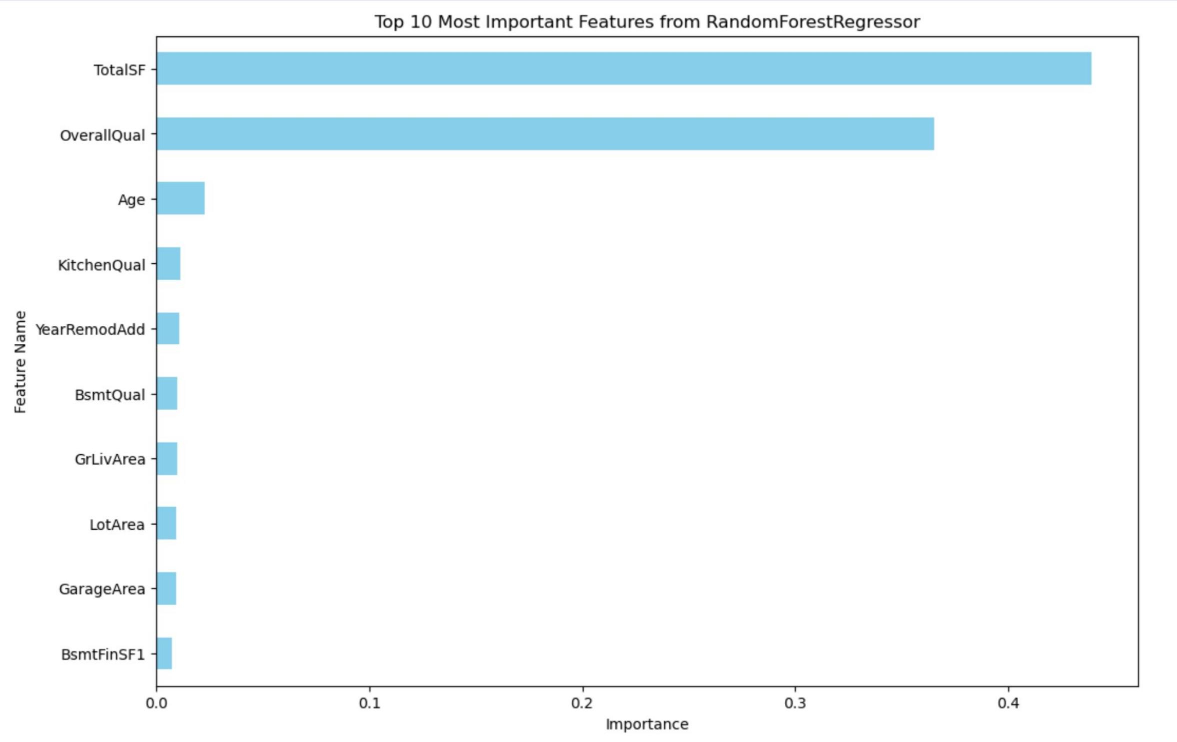
Additional Models

Lazy Predict



Random Forest

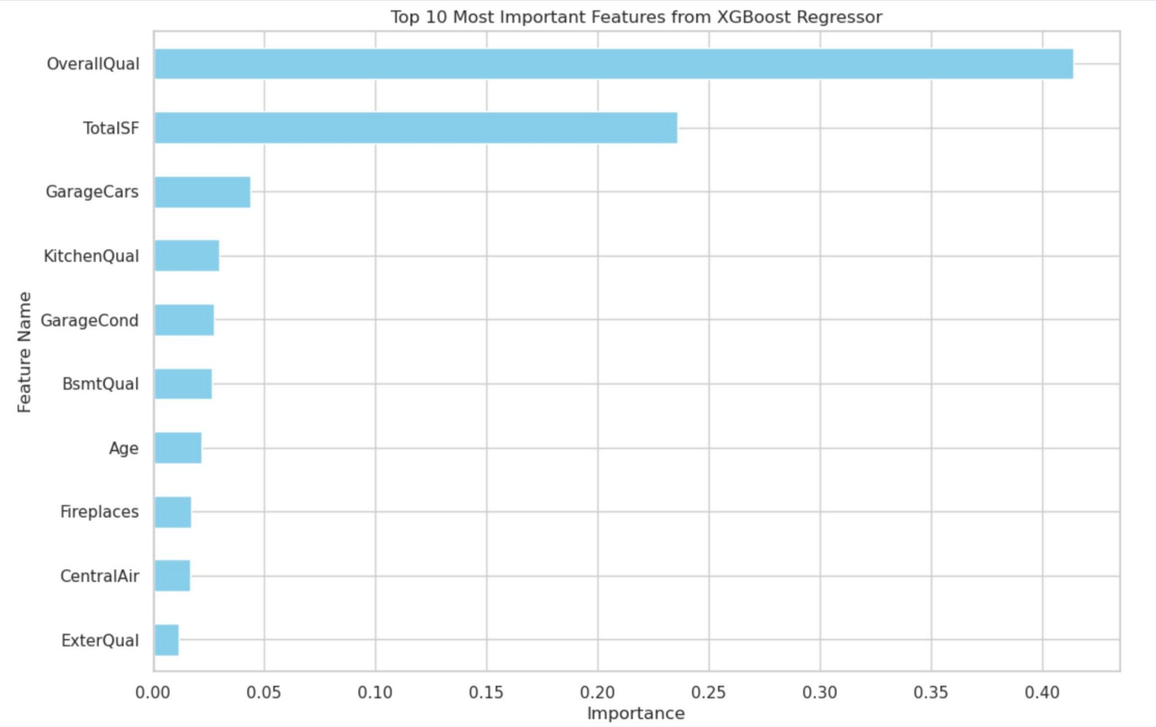
R^2 over 5-fold CV:
0.9059



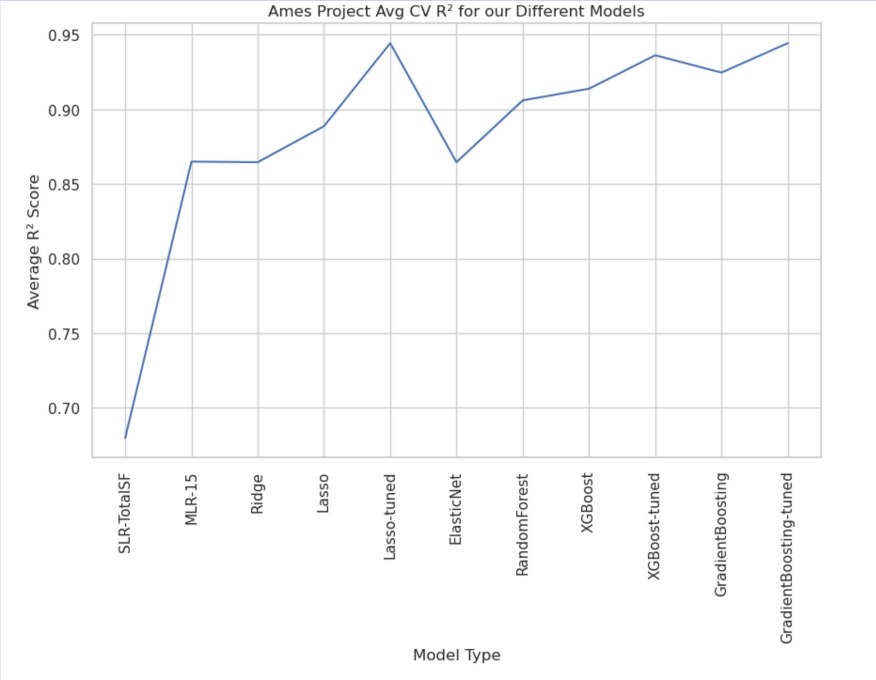
XGBoost

R^2 over 5-fold CV, no tuning:
0.9137

R^2 over 5-fold CV, with tuning:
0.9361



Model Comparison



Takeaways

Final Takeaways

- Key features
  - **Total SF + Overall Quality**
- Prediction with **~95% accuracy**
- Coefficients show what drives price per unit of measure
- A **combination of ~50 best features** accurately predicts price.
- Extrapolate to other housing models
  - **Year built, overall quality, other top 10s** are likely to predict housing in other markets
- What makes this market unique?
  - **College town**

Future Work

- Examining other models
- Further feature engineering
- Adjusting price based on price time series
- Incorporating other data sources
- Determining undervalued/overvalued homes

References

Github Repositories:
https://github.com/cpereda/Ames-Iowa-Housing-ML-Project
https://github.com/lottiewolf/NYCDSA_ML_Ames
https://github.com/nherman3/NYCDSA_ML_Project

Reference Material:
https://www.census.gov/quickfacts/fact/table/amescityiowa/POP010210#POP010210
https://www.icip.iastate.edu/sites/default/files/2010census/2010census_1901855.pdf
https://www.researchgate.net/publication/337048557_A_Comparative_Analysis_of_XGBoost
https://github.com/thismlguy/analytics_vidhya/blob/master/Articles/Parameter_Tuning_GBM_with_Example/GBM_Parameters.xlsx
https://xgboost.readthedocs.io/en/stable/parameter.html

Special Thanks:
Vinod Chugani