PREDICTING HOUSE PRICES IN AMES, IOWA

for i in team:

Alex Tin, Deborah Leong, Tori Lowery, Jay Cohen

Agenda

Data Exploration

Data Cleaning & Feature Engineering

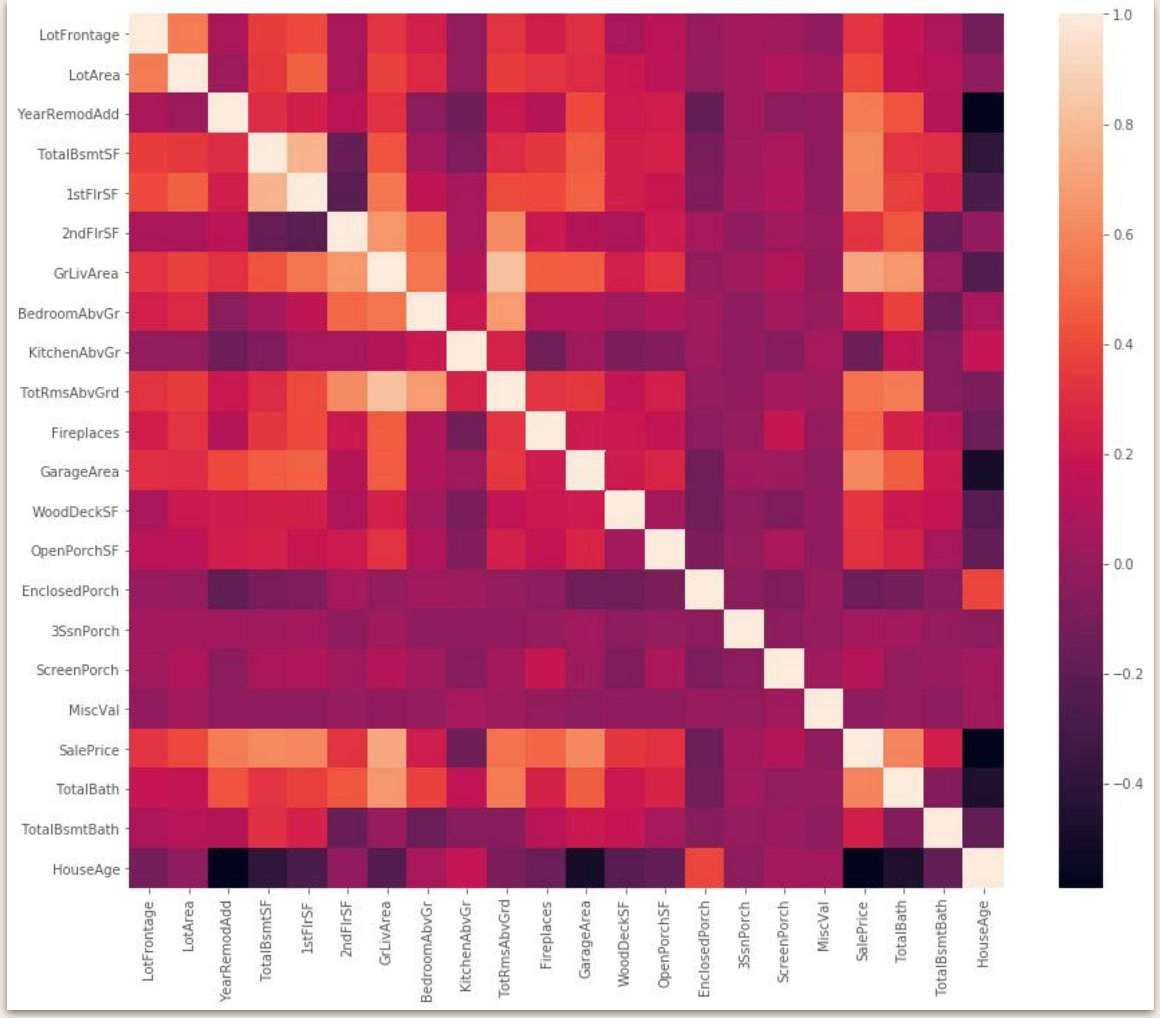
Models

Evaluating the Models

Implications

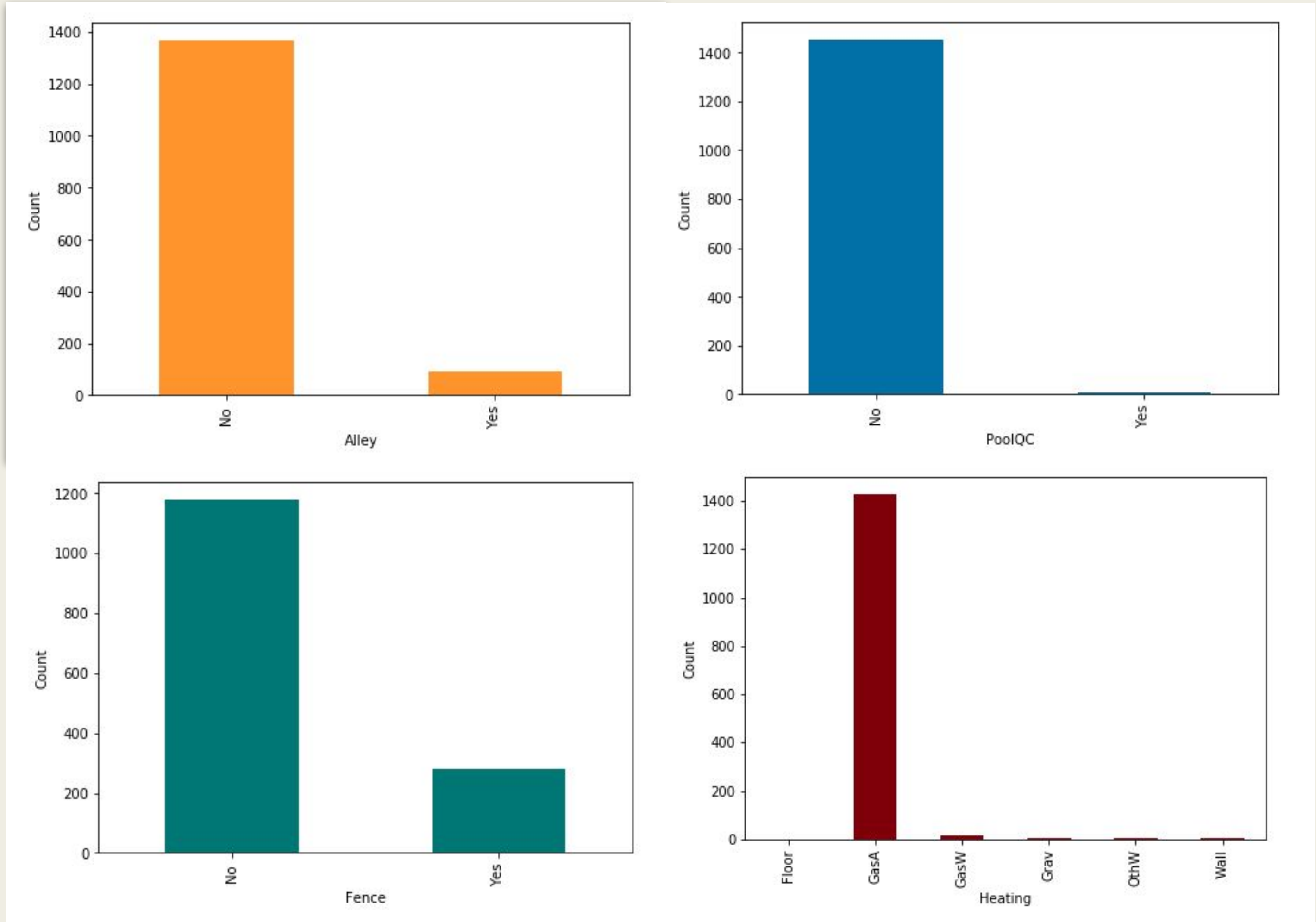
Data Exploration

- Examined features for:
  - *Low Variance*
  - *Redundancy*
  - *Missingness*



Data Exploration: Low Variance

Identifying low-variance features helps us to understand where we might find predictive power–and where we might not.

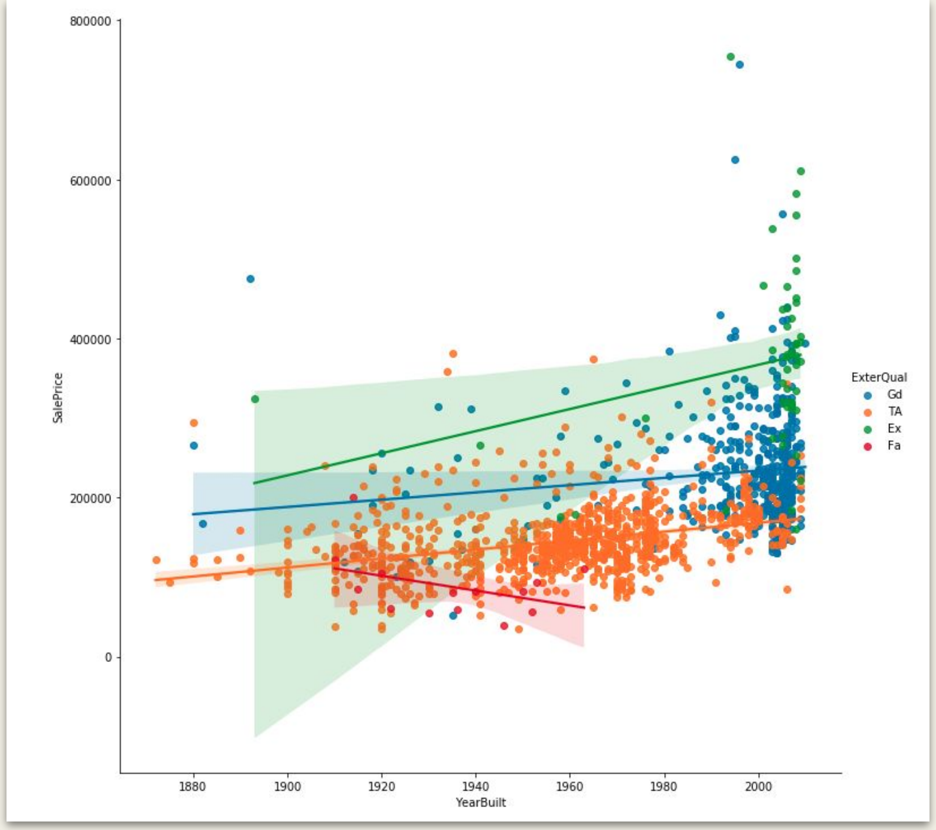


<u>Minority Columns:</u>

- Alley: 91 / 1460
- Pool: 7 / 1460
- Fence: 281 / 1460
- Heating: 32 / 1460 (non-GasA)

Data Exploration: Redundancy

Features including **External Quality** (seen below on the left) and **Foundation** (right) seemed to follow hand-in-hand with the **Year** the house was built. This creates opportunities for multicollinearity.



Data Exploration: Missingness

- **High Proportion of Missingness**
  - PoolQC (99.5%) , MiscFeature (96.3%) , Alley (93.7%), Fence (80.7%)
- **Missing at Random**
  - *Relation to other columns*
    - FireplaceQu (correspond to Fireplaces = 0)
    - LotFrontage (LotConfig)
    - Garage* (GarageType = No garage)
- **Missing Completely at Random**
  - Bsmt*
  - Electrical
- **Missing Not at Random**
  - MasVnrType
  - MasVnrArea

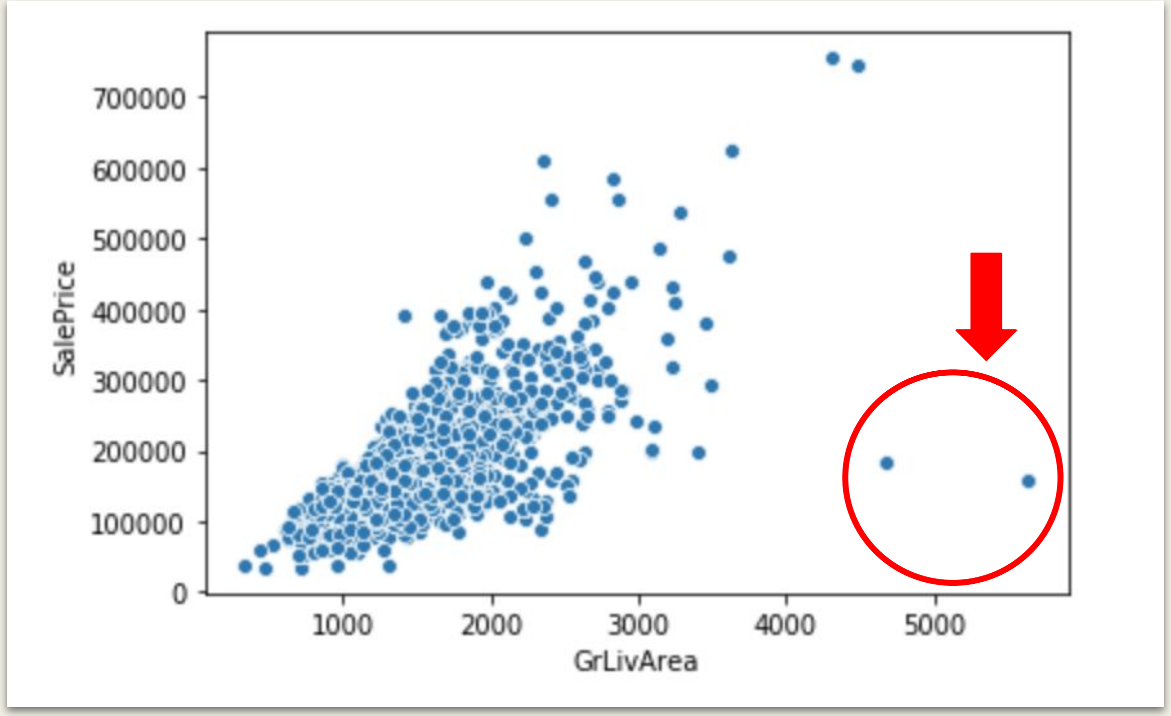
	Count	%
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageType	81	0.055479
GarageCond	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479
GarageYrBlt	81	0.055479
BsmtFinType2	38	0.026027
BsmtExposure	38	0.026027
BsmtQual	37	0.025342
BsmtCond	37	0.025342
BsmtFinType1	37	0.025342
MasVnrArea	8	0.005479
MasVnrType	8	0.005479
Electrical	1	0.000685

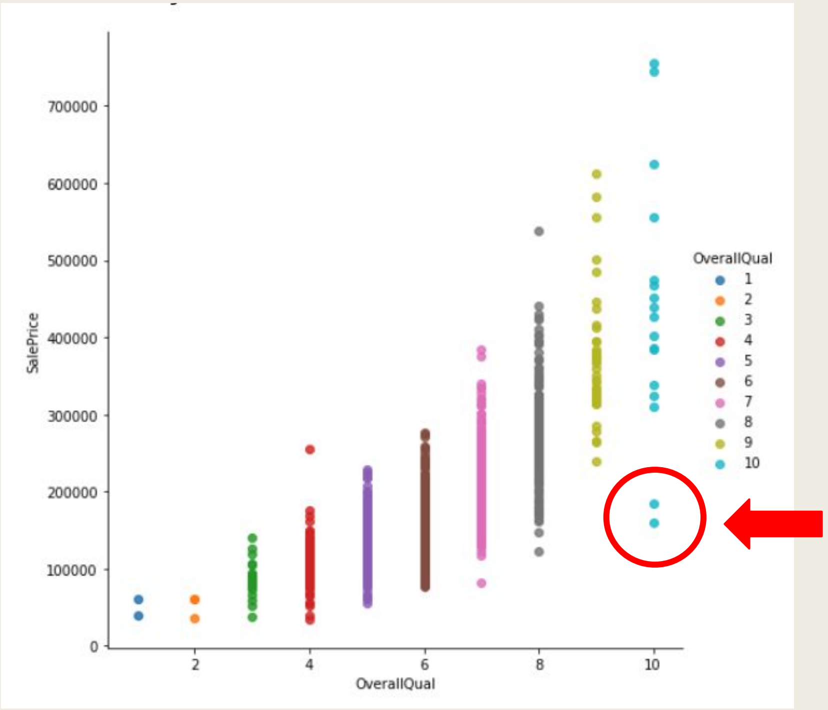
Data Cleaning and Feature Engineering

- Outliers
- Skewness
- Feature Manipulation
- Missingness/Imputation

Data Cleaning: Outliers

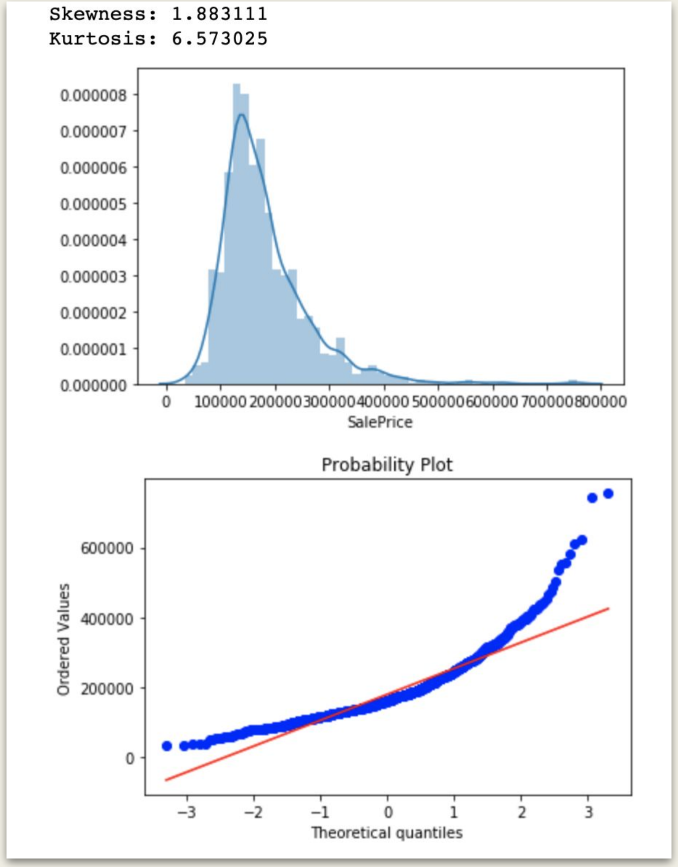
- Check for outliers to remove by plotting various features against SalePrice
- 2 main outliers: Id #s 524 and 129
- Remove those two outliers from data

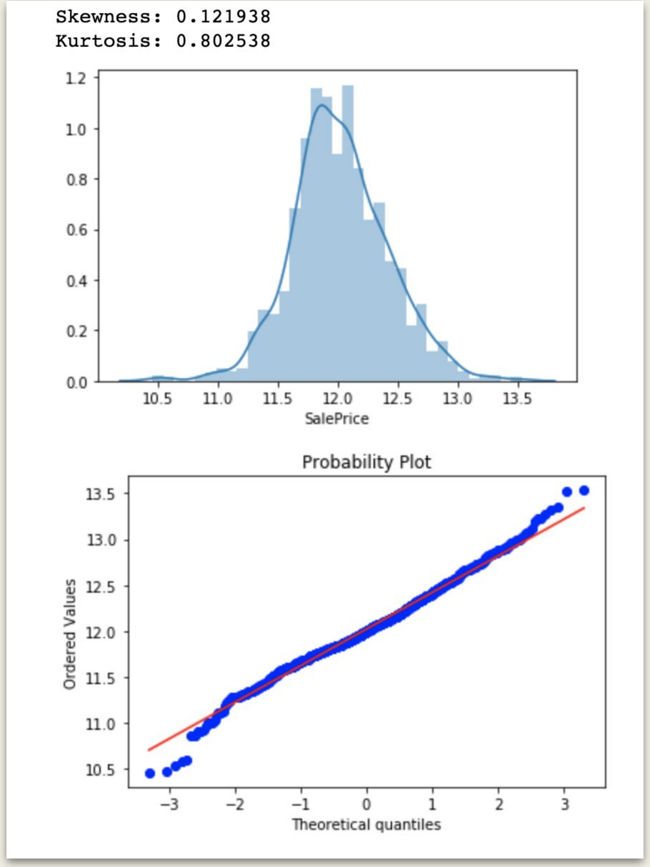




Data Cleaning: Skewness

- Because SalePrice was right skewed, we performed a log transformation on SalePrice





Data Cleaning: Skewness

- Find features that are skewed
- Perform transformation on skewed features
- Example:
  - *Find features with skew > .5*
  - *Perform Box-Cox Transformation on those features*

```
There are 18 numerical features with Skew > 0.5 :
MiscVal          21.905788
LotArea          13.146313
LowQualFinSF     12.065521
3SsnPorch        11.354131
KitchenAbvGr      4.311221
EnclosedPorch     4.006758
ScreenPorch       3.937927
OpenPorchSF       2.540619
WoodDeckSF        1.845533
MSSubClass        1.375365
1stFlrSF          1.254244
LotFrontage       1.104841
GrLivArea         1.071839
2ndFlrSF          0.863131
TotRmsAbvGrd      0.751990
Fireplaces        0.724697
HalfBath          0.701358
OverallCond       0.569427
```

Data Cleaning: Feature Manipulation

Because many of the features were a scaling metric from “Poor” to “Excellent”, e.g. the quality of the home’s exterior (“**ExterQual”**) or the basement’s condition (“**BsmtCond**”), these features were converted to a numerical value. This gives weight to the ranks since 1, “Very Poor”, has a relational distance to 10, “Very Excellent”.

Numerical data, like “**MSSubClass”**, which described a classification of the dwelling and did not have any numerical relationship, was manipulated into a string.

Alley was manipulated into a binary, “Yes” or “No”.

```
MSSubClass        int64
MSZoning         object
LotFrontage     float64
LotArea           int64
Street           object
Alley            object
LotShape         object
LandContour      object
Utilities        object
LotConfig        object
LandSlope        object
Neighborhood     object
Condition1       object
Condition2       object
BldgType         object
HouseStyle       object
OverallQual       int64
OverallCond       int64
YearBuilt         int64
YearRemodAdd      int64
RoofStyle        object
RoofMatl         object
Exterior1st      object
Exterior2nd      object
MasVnrType       object
MasVnrArea      float64
ExterQual        object
ExterCond        object
Foundation       object
BsmtQual         object
BsmtCond         object
BsmtExposure     object
BsmtFinType1     object
BsmtFinSF1        int64
BsmtFinType2     object
BsmtFinSF2        int64
BsmtUnfSF         int64
TotalBsmtSF       int64
Heating          object
HeatingQC        object
CentralAir       object
Electrical       object
1stFlrSF          int64
2ndFlrSF          int64
LowQualFinSF      int64
GrLivArea         int64
```

• • •

```
MSSubClass       object
MSZoning         object
LotFrontage     float64
LotArea           int64
Street           object
Alley             int64
LotShape         object
LandContour      object
Utilities        object
LandSlope        object
Neighborhood     object
Condition1       object
Condition2       object
BldgType         object
HouseStyle       object
OverallQual       int64
OverallCond       int64
YearBuilt         int64
YearRemodAdd      int64
RoofStyle        object
RoofMatl         object
Exterior1st      object
Exterior2nd      object
MasVnrType       object
MasVnrArea      float64
ExterQual         int64
ExterCond         int64
Foundation       object
BsmtQual          int64
BsmtCond          int64
BsmtExposure      int64
BsmtFinType1      int64
BsmtFinType2      int64
BsmtUnfSF         int64
TotalBsmtSF       int64
Heating          object
HeatingQC         int64
CentralAir       object
Electrical       object
1stFlrSF          int64
2ndFlrSF          int64
LowQualFinSF      int64
GrLivArea         int64
```

• • •

Data Cleaning: Missingness/Imputation

```
sum(pd.isna(houses["GarageYrBlt"]))
```

```
81
```

```
sum(pd.isna(houses["LotFrontage"]))
```

```
259
```

LotFrontage
65.0
80.0
68.0
60.0
84.0
85.0
75.0
NaN
51.0
50.0
70.0
85.0
NaN
91.0
NaN
51.0
NaN
72.0
66.0
70.0

LotFrontage
65.0
80.0
68.0
60.0
84.0
85.0
75.0
84.0
51.0
50.0
70.0
85.0
67.7
91.0
84.0
51.0
59.9
72.0
66.0
70.0

Missingness in features like “**GarageYrBlt**” (the year the garage was built) and the total “**Lot Frontage”** (linear feet of street connected to property) required different treatments.

For the mentioned garage feature, if the home did not have a garage, there would be no year recorded for one to be built. We imputed a value of 0 to distinguish these homes as not having garages.

Missingness in the **“Lot Frontage”** feature, however, appeared to be instances of absent data. To salvage these 259 lines, we grouped by Neighborhood then binned a similar feature, **“LotConfig”** (the configuration of the lot, e.g. a corner lot or inside lot), and imputed the respective average frontage to each missing entry.

Cleaned Data

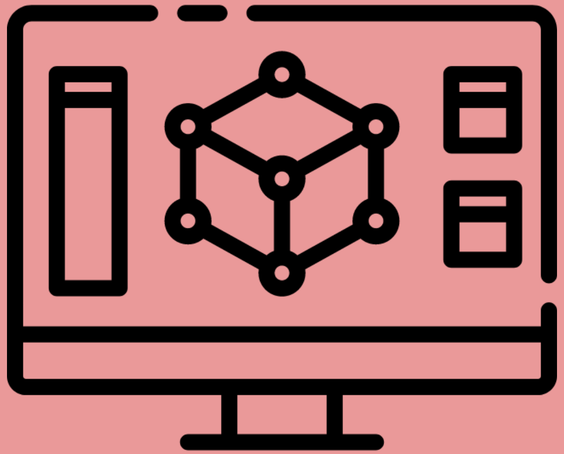
- Missingness
- Outliers
- Skewness
- Categorizing

Cleaned Dataset

```
1  iowahousingset.shape
(1451, 63)
```

41 categorical features
22 continuous features

Machine Learning Models



Models

- Linear Regressions
  - *OLS*
  - *Ridge*
  - *Lasso*
- Random Forest
- Gradient Boosting

Models: Ordinary Least Squares

- Select features that have high correlation with *log*(SalePrice)

- Try different transformations on skewed features
  - *Box-Cox*
  - *Log*

Example OLS Model
Test Score = 0.842

	Coefficient
log_GrLivArea	0.393874
log_LotArea	0.145287
log_TotalBsmtSF	0.029652
log_GarageArea	0.017314
OverallQual	0.083565
ExterQual	0.020214
KitchenQual	0.042584
YrSold	-0.000267
YearBuilt	0.002497

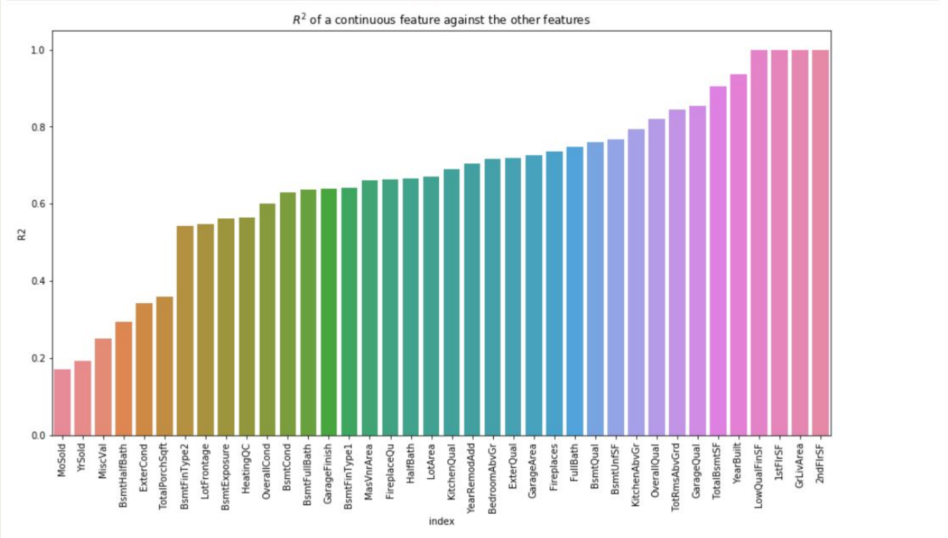
```
ols.fit(X2_train, y2_train)
ols.score(X2_train, y2_train)
```

```
0.939972827099102
```

```
ols.score(X2_test, y2_test)
```

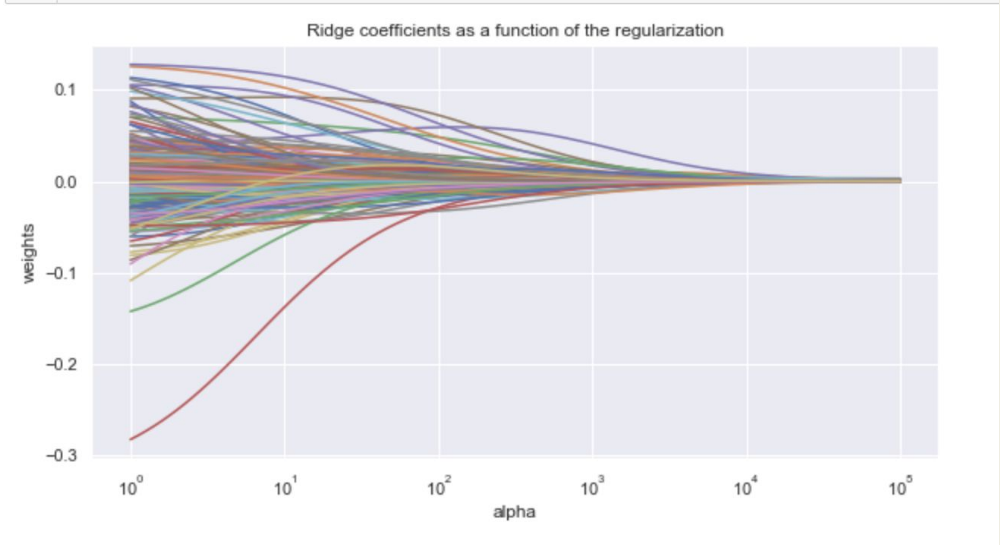
```
0.6980629831416827
```

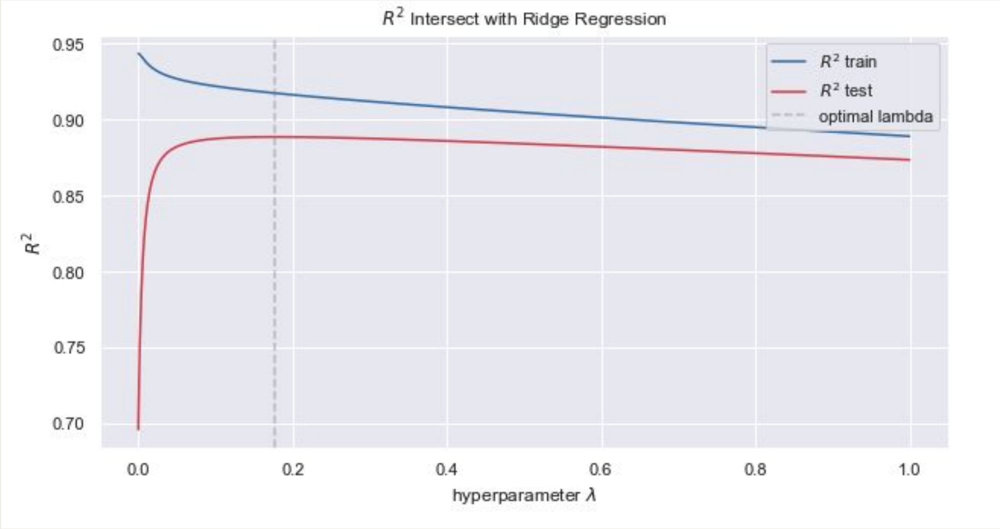
While our train score is decent, the test score shows we are overfitting. The graph to the right also shows there is high multicollinearity amongst our features. Penalization is needed.



Models: Ridge

- Finding optimal alpha
- RidgeCV(Cross Validation)
- Low RMSE



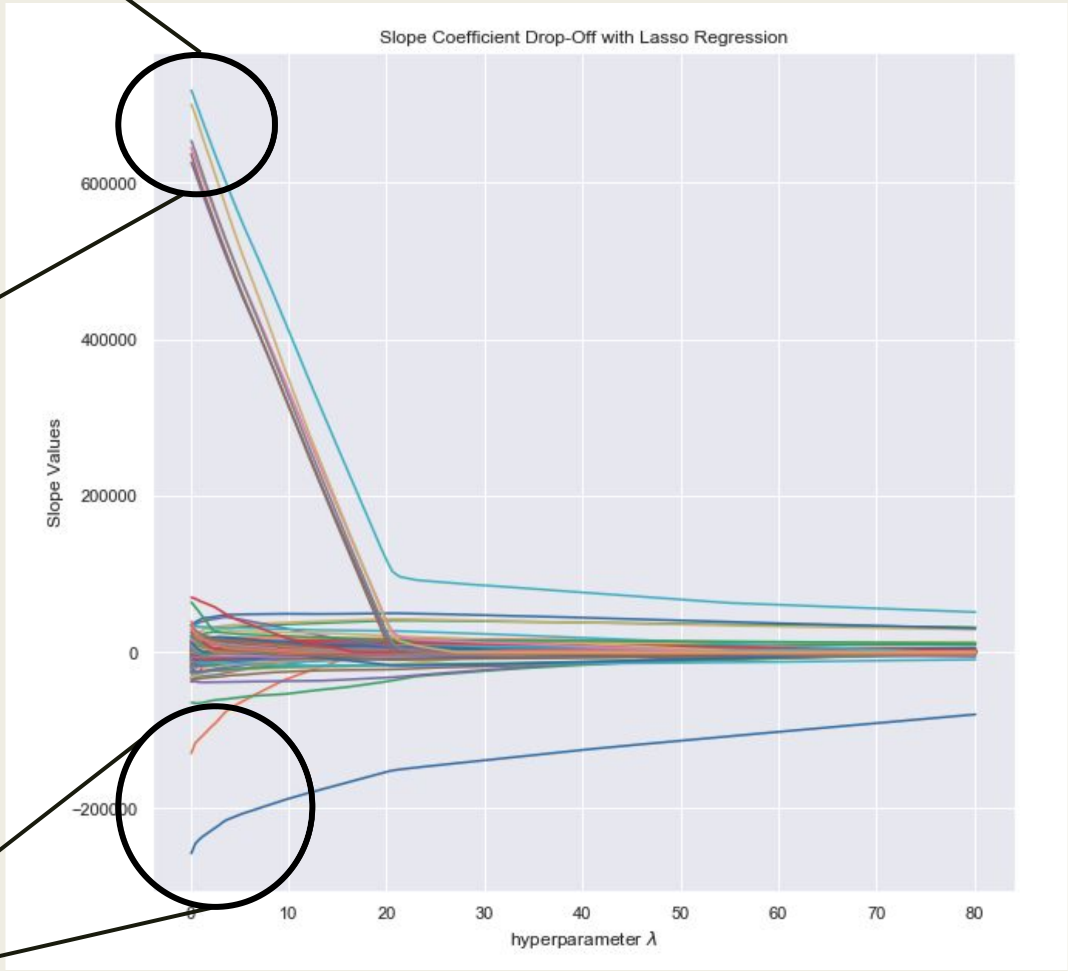


Models: Lasso

Because there were so many features, we attempted a LASSO model to guide feature trimming.

The graph to the right shows the significant drop of many features' coefficients, a sign their slope was greatly inflated from multicollinearity.

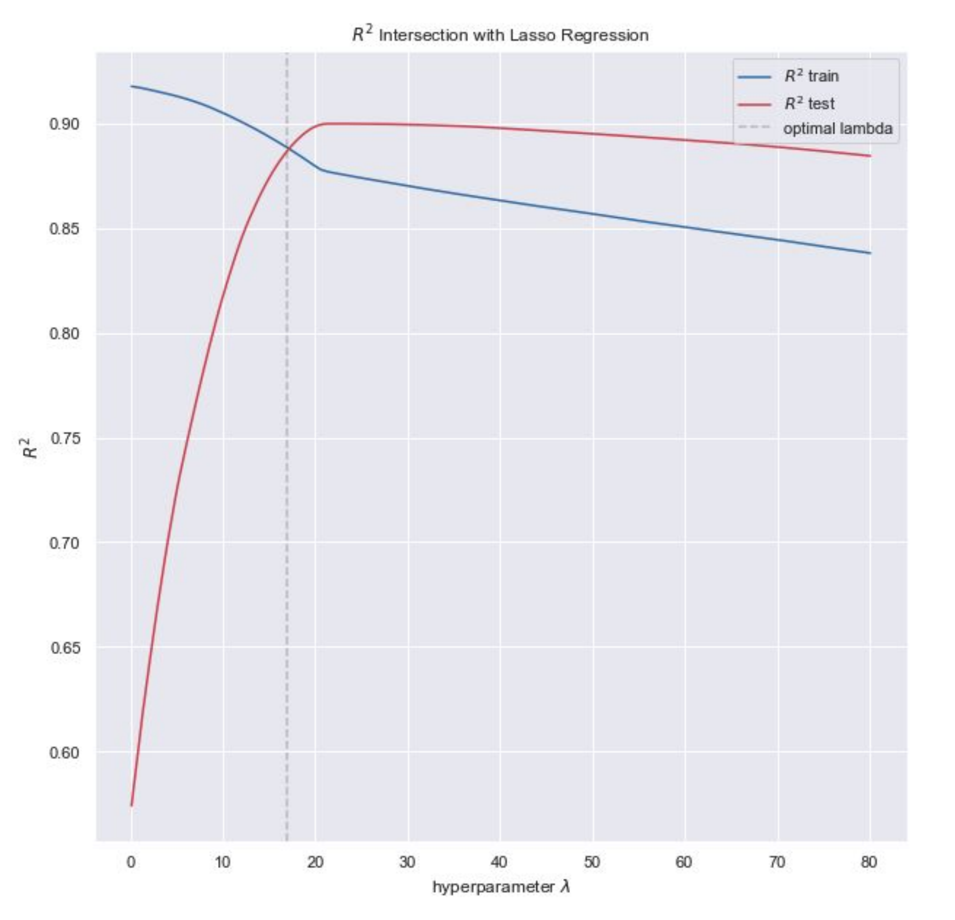
MSSubClass_45	39428.937179
MSZoning_FV	29014.299539
MSZoning_RH	32836.073765
MSZoning_RL	24439.511497
MSZoning_RM	20277.410961
Neighborhood_StoneBr	34055.277239
Condition2_PosA	24685.920748
HouseStyle_SLvl	20395.399761
RoofStyle_Shed	69710.219118
RoofMatl_CompShg	644784.831216
RoofMatl_Metal	700679.859523
RoofMatl_Roll	636969.731300
RoofMatl_Tar&Grv	653925.069645
RoofMatl_WdShake	626155.804321
RoofMatl_WdShngl	718601.103185
Exterior2nd_ImStucc	33569.063274
SaleType_Con	33529.498960
SaleType_ConLD	28760.883822
SaleType_New	63701.556078



Condition2_PosN	-257723.987128
Condition2_RRAe	-129521.928289

Models: Lasso, Hyperparameter Tuning

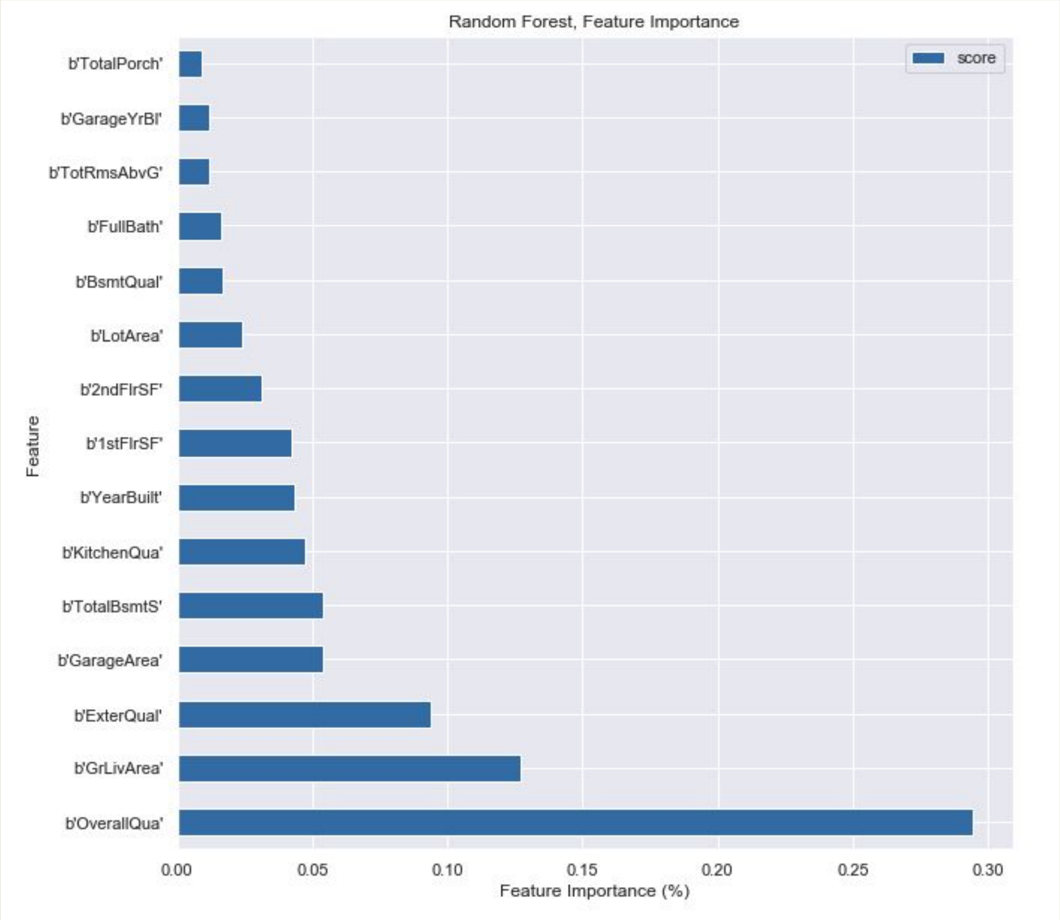
To find the *best* lambda, we found the cross where the model no longer overfits (the left side of the gray dashed line) and before the model begins underfitting (the right side of the line).



Models: Random Forest

While previously for our linear models we had taken the log of our Sale Price target, for tuning our Random Forest model we found the log transformation actually weakened the model. This may be that Random Forest is a nonlinear model and can handle far better a nonlinear relationship to the target.

Additionally, when we analyze the rankings of feature importance (see figure to the left), **“OverallQual”** greatly surpasses every other predictor.

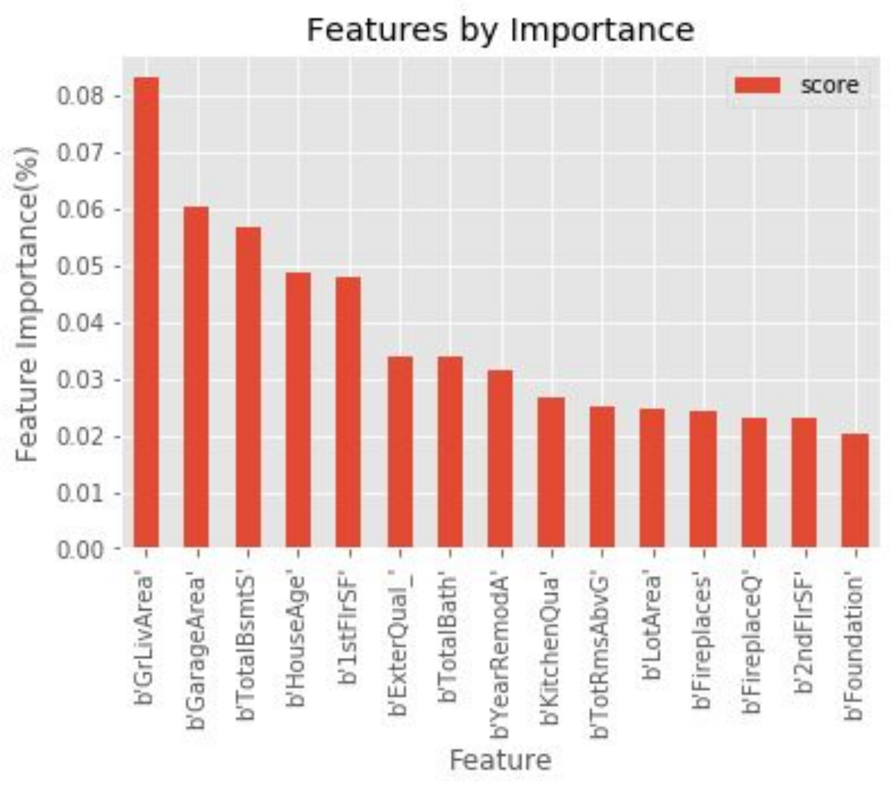


Models: Random Forest

In the RandomForest model trained where “OverallQual” was classified as categories, several other features values jumped up on the feature importance chart.

New on the leaderboard:
YearRemodA, **Fireplaces**, **FireplaceQu**, **Foundation**

Dropped off the leaderboard:
Total Porch, GarageYrBlt, YearBuilt, OverallQual



Models: Gradient Boosting

Hyperparameter tuning quickly grew to a task fit better for grid search than manual manipulation. Previously lasso and ridge required a predetermined lambda, but for our Gradient Boosting model, parameters including number of trees, learning rate, maximum depth, and maximum features to consider all needed testing.

Below is a sample grid search performed for our gradient boosting model.

```
grid_para_gbm = {
    "n_estimators": [100, 200, 300, 400, 500],
    "learning_rate": [.0001, .001, .01, 0.1],
    "max_depth": list(range(1, 10)),
    "max_features": [None, 1, 2, 3],
    "random_state": [42]}
grid_search_gbm = GridSearchCV(gbm, grid_para_gbm, scoring='r2', cv=5, n_jobs=-1)
grid_search_gbm.fit(X_train, y_train)
print("The best train score is: ", grid_search_gbm.best_estimator_.score(X_train, y_train))
print("The best test score is: ", grid_search_gbm.best_estimator_.score(X_test, y_test))
```

```
The best train score is:  0.9653301803244188
The best test score is:  0.9071519763531863
```

Evaluating the Models

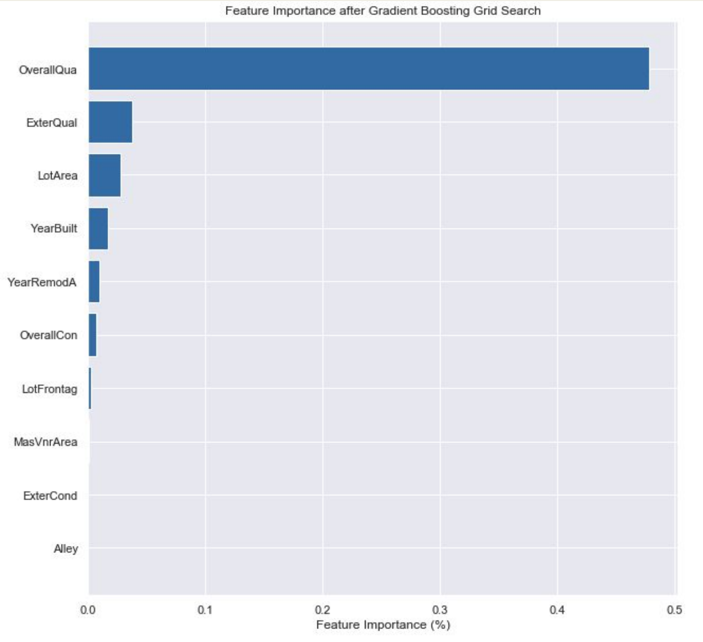
Model	Trials	Train_score	Test_score
Linear Regression	Multiple Linear Regression without dropping outliers and with Log Transformations of GrLivArea, LotArea, TotalBsmtSF, GarageArea, and SalePrice	0.846	0.842
Linear Regression	Multiple Linear Regression dropping outliers and with Log Transformations of GrLivArea, LotArea, and SalePrice	0.866	0.732
Penalized Linear Regression (Ridge)	Ridge Regression with CV with Box-Cox Transformations on all numerical features and dropping outliers	RMSE: 0.941	
Penalized Linear Regression (Lasso)	normalize=True, alpha=16.98442211	0.888	0.887
RandomForest v1	62 features, n_estimators = 1000, max_features = 20	0.974	0.877
RandomForest v2	31 features, n_estimators = 500, max_features = 100	0.979	0.906
Gradient Boosting v1	62 features, n_estimators = 1000, learning_rate = 0.1	0.998	0.907
Gradient Boosting v2	62 features, n_estimators = 500, learning_rate = 0.1	0.968	0.922
Gradient Boosting v3	62 features, n_estimators = 500, learning_rate = 0.1, max_depth = 3, max_features=None	0.994	0.927

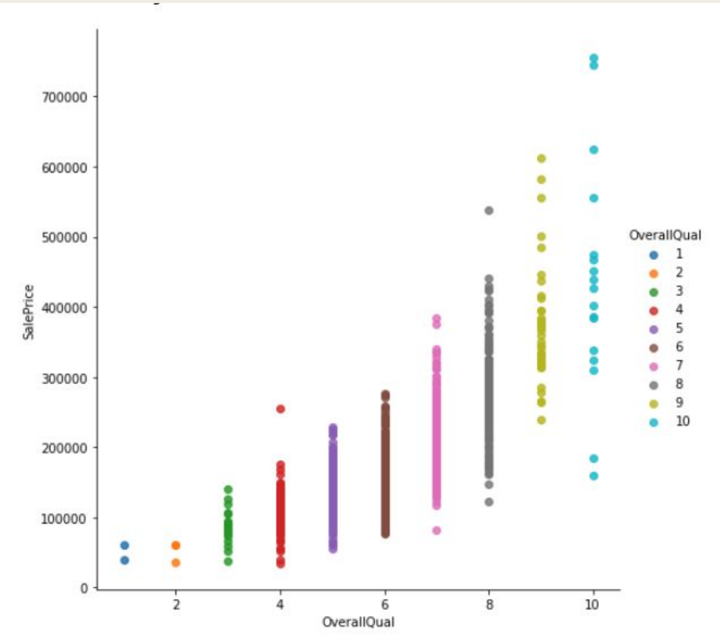
Implications

For housing prices, **human subjectivity ("OverallQual") was still the largest determinant.** Following shortly behind Overall Quality were often features measuring square footage, but quite a few other subjective categories also made it to the top depending on the model. **"ExterQual"** which judged on a scale of 1-5 from "poor" to "excellent" the quality of the material on the exterior.

Improvements

Explore ensemble techniques/stacked models to improve performance





Tips:

1. Invest in a professional photographer
2. Let in maximum light
3. Don't over-upgrade

First impressions are the only impressions. Professional photos can set the tone for house's quality.

If upgrading appliances, keep in mind the overall quality of the home.