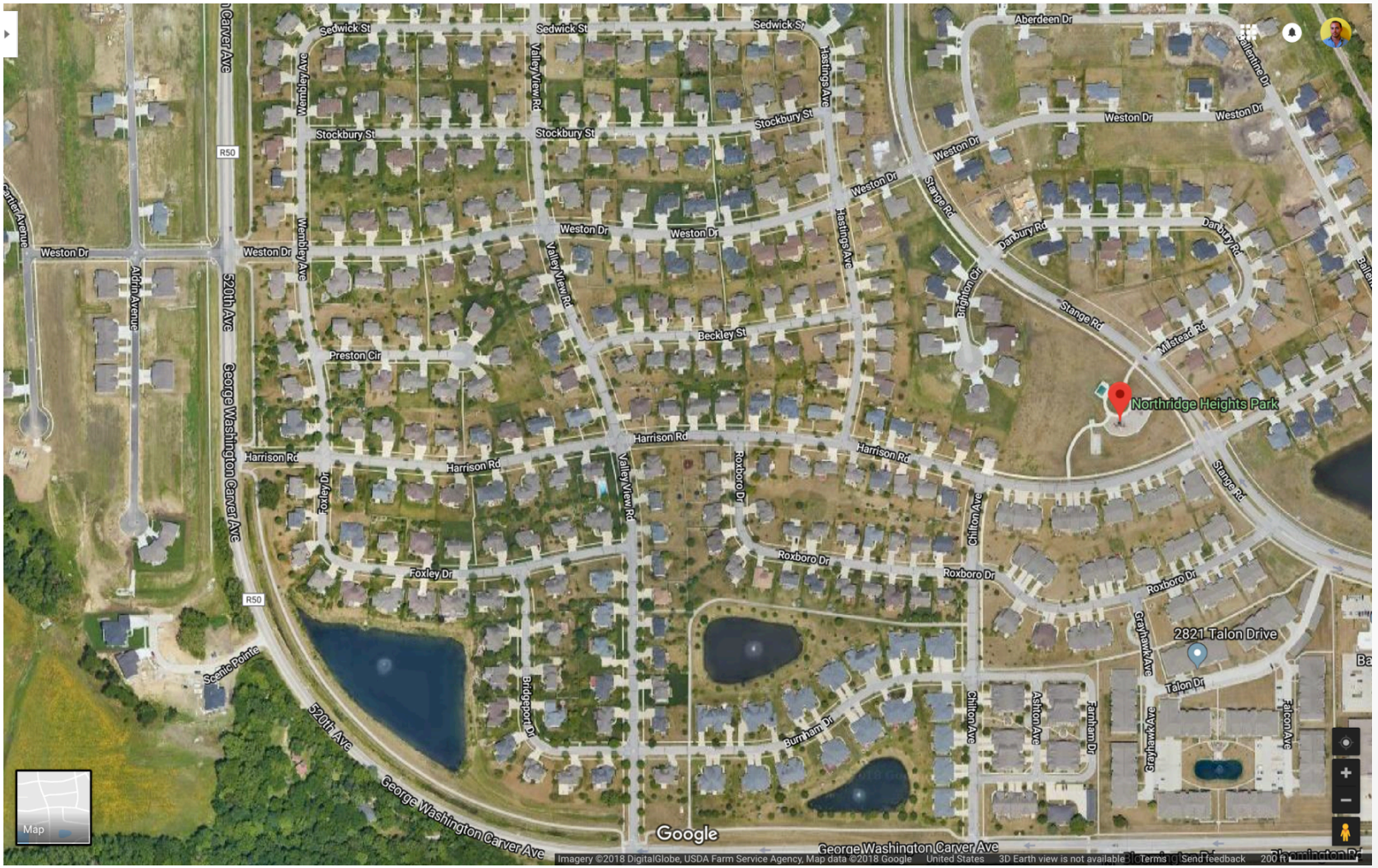
Aiming for the Right Price in Ames

Christopher Williams
DSI Project 2
December 7, 2018

Data Science Problem

- Create a regression model based on the Ames, Iowa Housing Dataset to predict the price of a house at sale from 2006 to 2010

Ames, Iowa


Sedwick St
Sedwick St
Sedwick St
Aberdeen Dr
Valentine Dr
Wembley Ave
Valley View Rd
Hastings Ave
Stockbury St
Stockbury St
Stockbury St
Weston Dr
Weston Dr
Weston Dr
Weston Dr
Weston Dr
Weston Dr
Weston Dr
Stange Rd
Darbury Rd
Darbury Rd
Brighton Cir
Stange Rd
Milstead Rd
Northridge Heights Park
Beckley St
Preston Cir
Harrison Rd
Harrison Rd
Harrison Rd
Harrison Rd
Roxboro Dr
Roxboro Dr
Roxboro Dr
Roxboro Dr
Roxboro Dr
Chilton Ave
Stange Rd
Foxley Dr
Foxley Dr
Valley View Rd
Adrin Avenue
320th Ave
George Washington Carver Ave
R50
R50
Scenic Pointe
Bridgeport Dr
320th Ave
George Washington Carver Ave
Burnham Dr
Chilton Ave
Ashton Ave
Farnham Dr
Greyhawk Ave
Greyhawk Ave
Falcon Ave
2821 Talon Drive
Talon Dr
Map
Google
George Washington Carver Ave
Imagery ©2018 DigitalGlobe, USDA Farm Service Agency, Map data ©2018 Google
United States
3D Earth view is not available
Terms
Send Feedback
200 ft

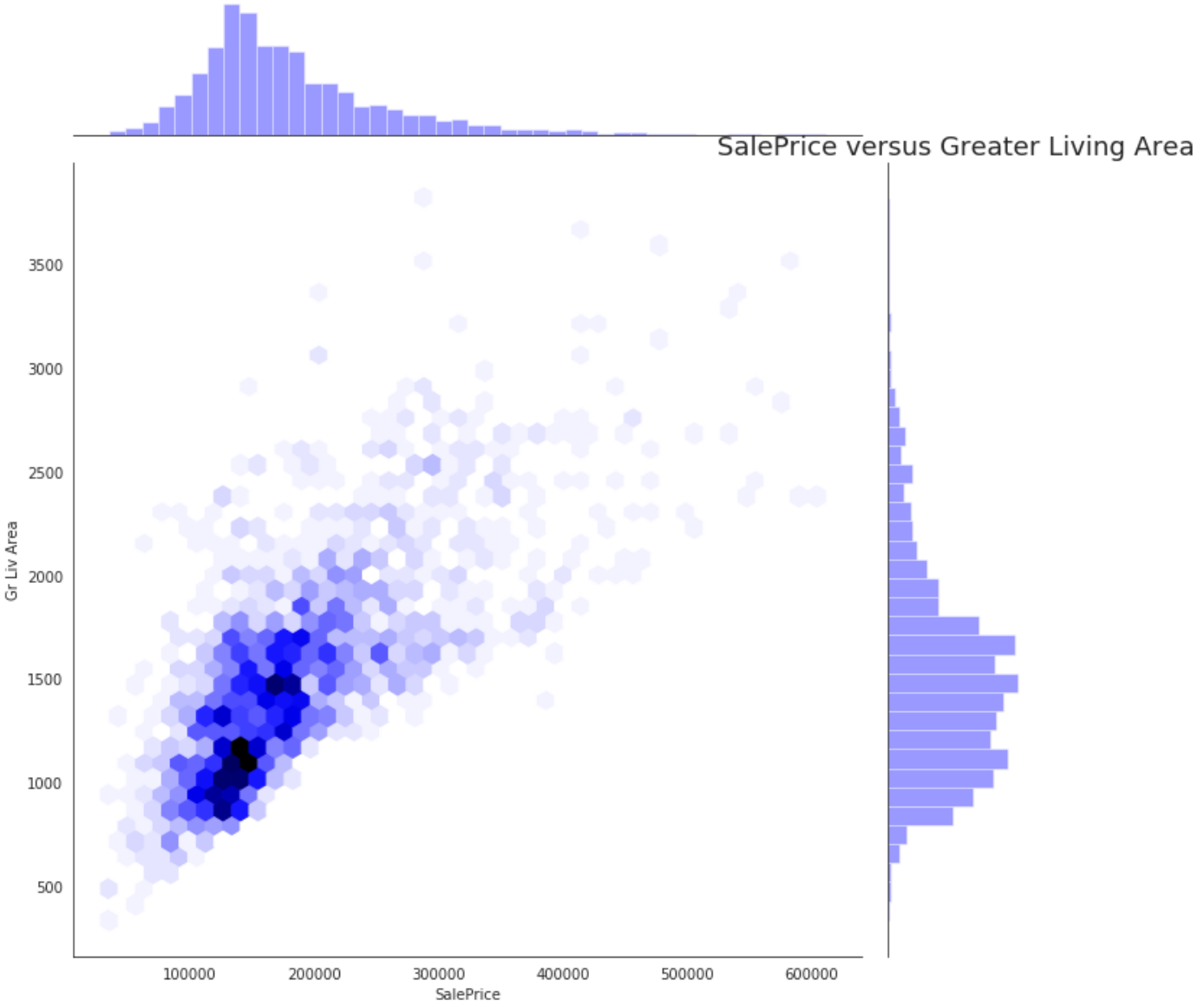
About Ames, Iowa

- Census of 2010
  - 58,965 people
  - 22,759 households
  - 9,959 families
  - 999,789,198 cats
- 23,876 housing units

The Data

- We were provided with two data files
  - Train dataset with SalePrice data
    - 2051 rows of houses
    - 81 rows columns
  - Test dataset without SalePrice data
    - 879 rows of houses
    - 80 rows columns
- What to do with the Data?
  - Clean up
  - Map out categorical & qualitative values to quantitative
  - Leave outliers out

EDA

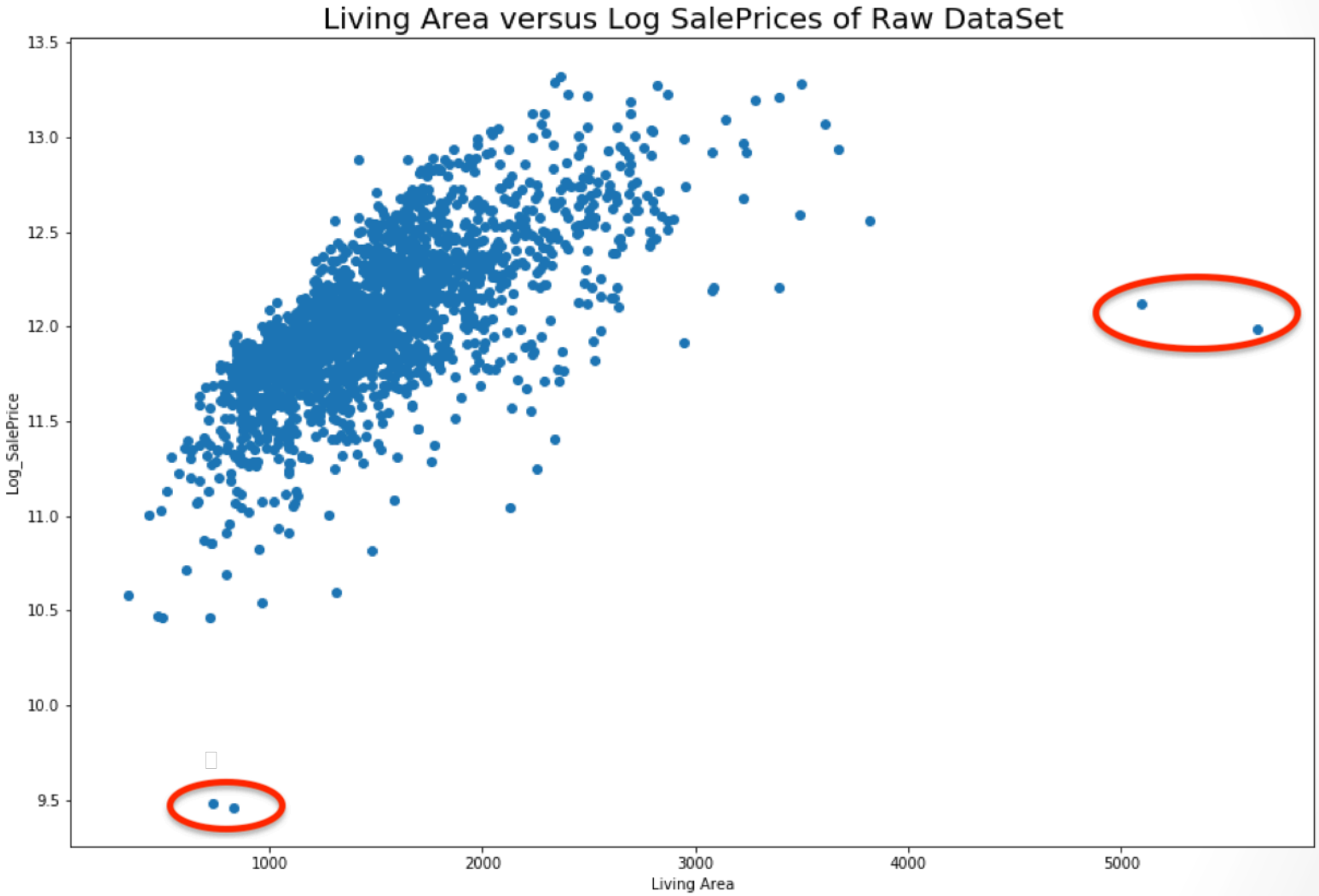

SalePrice versus Greater Living Area
Gr Liv Area
3500
3000
2500
2000
1500
1000
500
100000
200000
300000
400000
500000
600000
SalePrice

Dealing with the Data

- What changes and assumptions to make on data in model?
  - Take out four outliers (see slide)
  - Map Values for qualitative and categorical features (see slide)
    - Example mapping qualities to a numerical scale:
      - {“Ex”: 7, “Gd”: 4, “TA”: 3, “Fa”: 2, “Po”: 1}
      - Ex = excellent, Gd = good, TA = average, Fa = fair, Po = poor
  - Perform Log Transformation of SalesPrice to make normally distrubted (see slide)

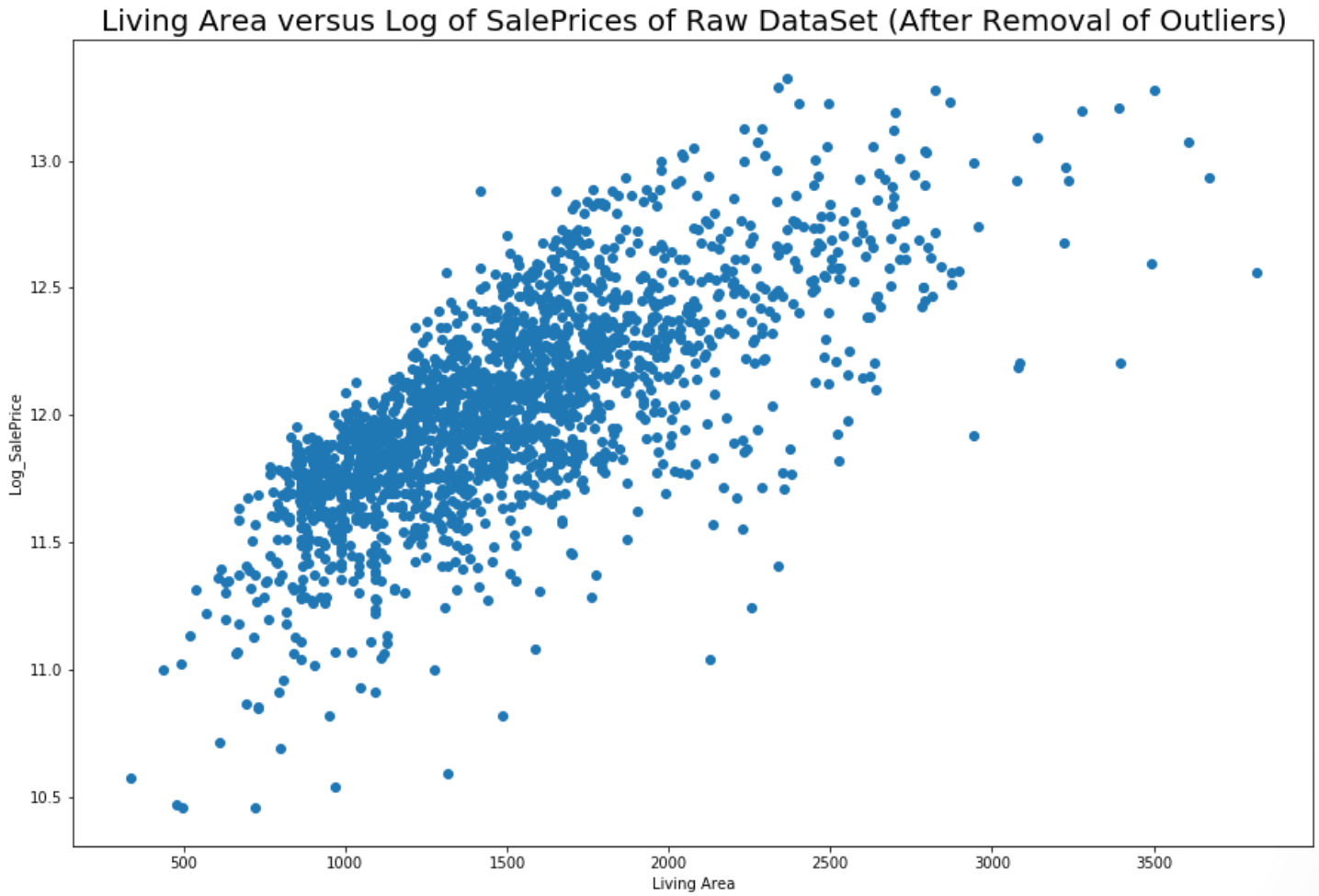
Data Cleanup

Before removing 4 outliers



Data Cleanup

After removing 4 outliers



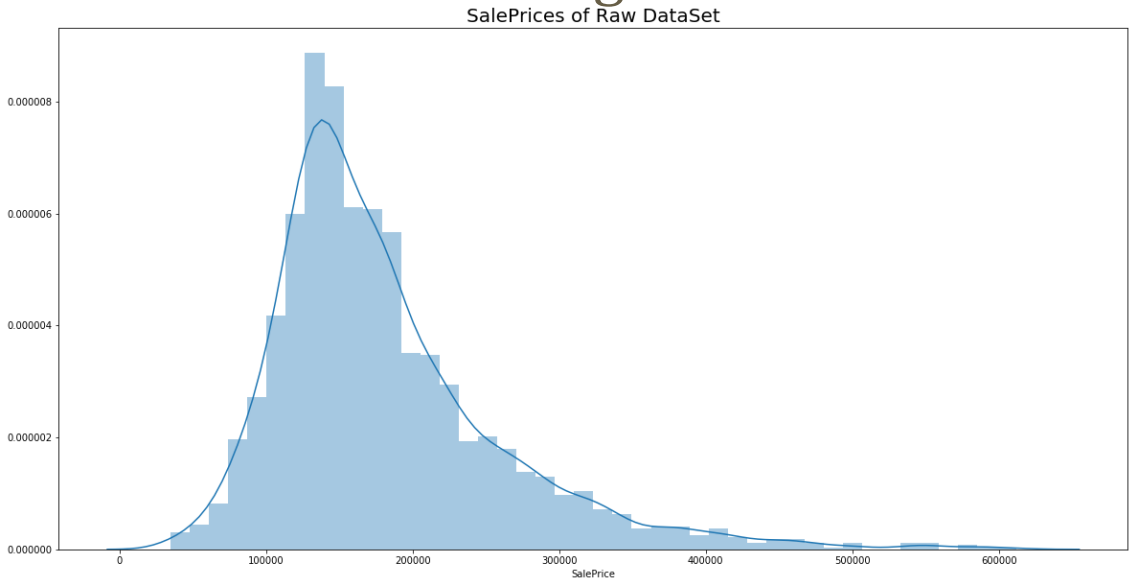
Data Mapping & Cleaning

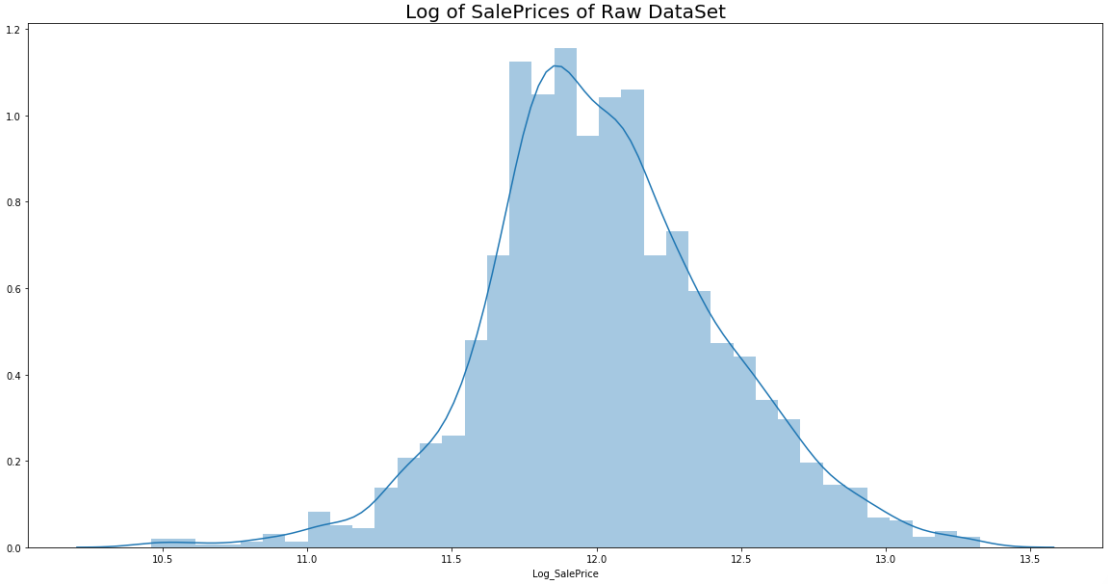
```
def Cleaning_Function(df):

    df['Lot Shape'] = df['Lot Shape'].map({'Reg':4, 'IR1':3, 'IR2':2, 'IR3':1})
    df['Utilities'] = df['Utilities'].map({'AllPub': 4, 'NoSewr':3, 'NoSeWa': 2, 'ELO':1})
    df['Land Slope'] = df['Land Slope'].map({'Gtl':3, 'Mod':2, 'Sev':1})
    df['Exter Qual'] = df['Exter Qual'].map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1})
    df['Exter Cond'] = df['Exter Cond'].map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1})
    df['Bsmt Qual'] = df['Bsmt Qual'].fillna(0).map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1, 0:0})
    df['Bsmt Cond'] = df['Bsmt Cond'].fillna(0).map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1, 0:0})
    df['Bsmt Exposure'] = df['Bsmt Exposure'].fillna(0).map({'Gd':5, 'Av':4, 'Mn':3, 'No':2, 'NA':0, 0:0})
    df['BsmtFin Type 1'] = df['BsmtFin Type 1'].fillna(0).map({'GLQ':6, 'ALQ':5, 'BLQ':3, 'Rec':4, 'LwQ':2, 'Unf':1, 0
    df['BsmtFin Type 2'] = df['BsmtFin Type 2'].fillna(0).map({'GLQ':6, 'ALQ':5, 'BLQ':3, 'Rec':4, 'LwQ':2, 'Unf':1, 0
    df['Heating QC'] = df['Heating QC'].map({'Ex':6, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1})
    df['Electrical'] = df['Electrical'].fillna(0).map({'SBrkr': 4, 'FuseA':3, 'FuseF':2, 'FuseP':1, 'Mix':2, 0:0})
    df['Kitchen Qual'] = df['Kitchen Qual'].map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1})
    df['Functional'] = df['Functional'].map({'Typ':25, 'Min1':10, 'Min2':9, 'Mod':8, 'Maj1':5, 'Maj2':4, 'Sev':1, 'Sal
    df['Fireplace Qu'] = df['Fireplace Qu'].fillna(0).map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1, 0:0})
    df['Garage Finish'] = df['Garage Finish'].fillna(0).map({'Fin': 10, 'RFn':5, 'Unf':3, 'NA':0, 0:0})
    df['Garage Qual'] = df['Garage Qual'].fillna(0).map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1, 0:0})
    df['Garage Cond'] = df['Garage Cond'].fillna(0).map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1, 0:0})
    df['Paved Drive'] = df['Paved Drive'].map({'Y': 5, 'P':2, 'N':1})
    df['Pool QC'] = df['Pool QC'].fillna(0).map({'Ex':7, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1, 0:0})
    df['Fence'] = df['Fence'].fillna(0).map({'GdPrv':4, 'MnPrv':3, 'GdWo': 4, 'MnWw':3, 'NA':1, 0:1})
    df['Garage Yr Blt'] = df['Garage Yr Blt'].fillna(df['Year Built'])
    df['Garage Cars'] = df['Garage Cars'].fillna(0)
    df['Garage Area'] = df['Garage Area'].fillna(0)
    df['Lot Area'] = df['Lot Area'].fillna(0)
    df['Lot Frontage'] = df['Lot Frontage'].fillna(0)
    df['Mas Vnr Area'] = df['Mas Vnr Area'].fillna(0)
    df['Mas Vnr Type'] = df['Mas Vnr Type'].fillna(0)
    df['BsmtFin SF 1'] = df['BsmtFin SF 1'].fillna(0)
    df['BsmtFin SF 2'] = df['BsmtFin SF 2'].fillna(0)
    df['Bsmt Unf SF'] = df['Bsmt Unf SF'].fillna(0)
    df['Total Bsmt SF'] = df['Total Bsmt SF'].fillna(0)
    df['Bsmt Full Bath'] = df['Bsmt Full Bath'].fillna(0)
    df['Bsmt Half Bath'] = df['Bsmt Half Bath'].fillna(0)
```

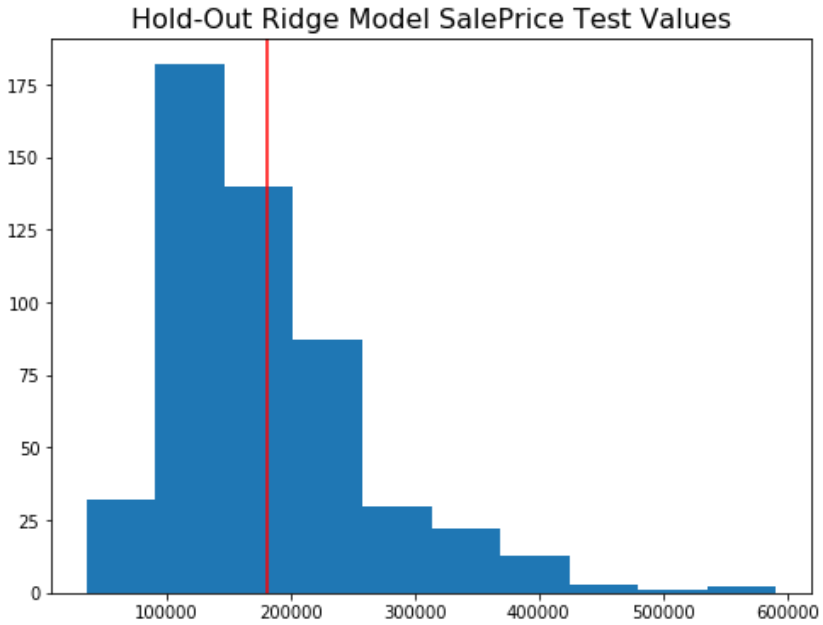
Sale Prices

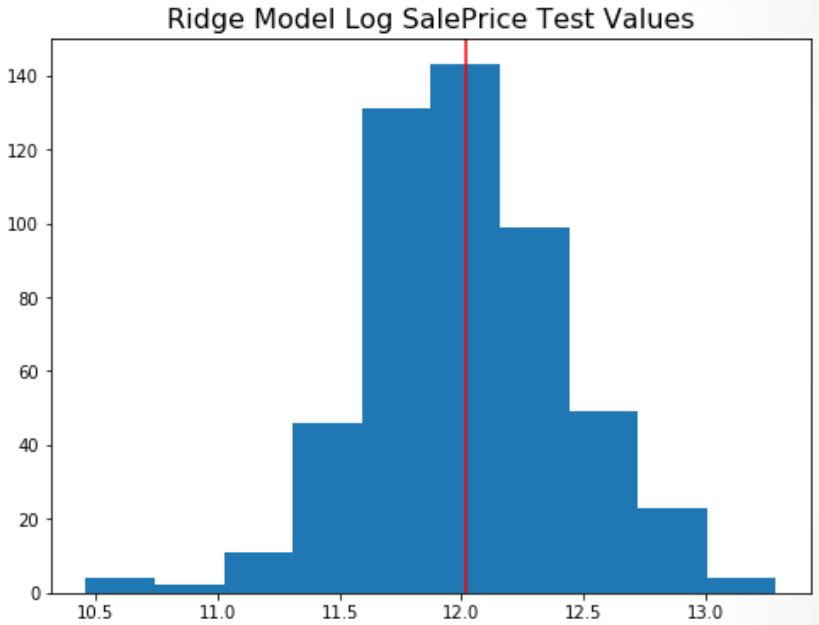
log transformation of SalesPrice





Visualization of SalePrice

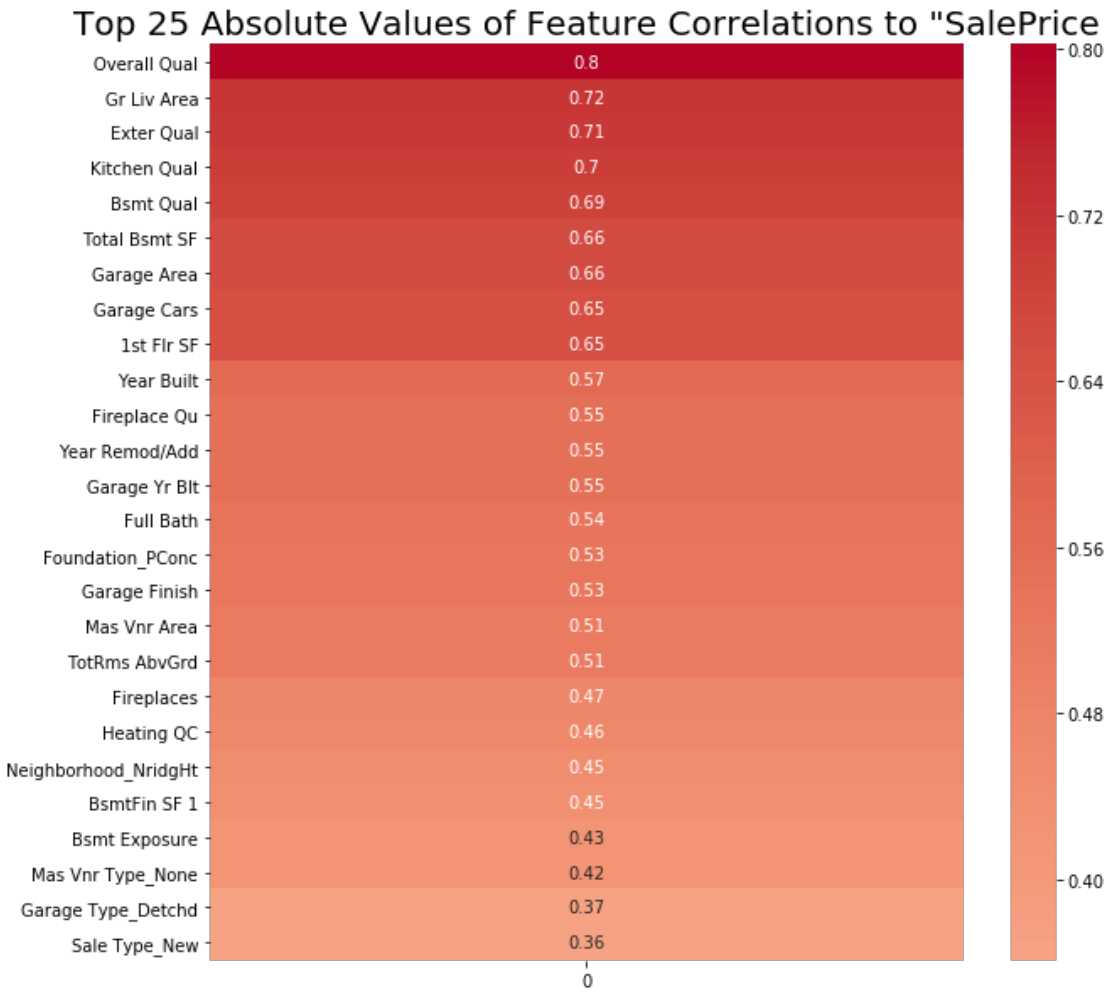




Regression Model

- Lasso vs. Ridge
  - Model using Lasso vs. Ridge regression regularization to account for feature coefficients' impact on predictions
- Feature Selection
  - Used features both positively and negatively correlated with SalePrice
- How to score and judge the models?
  - Models are predicting the target of SalePrice
  - Models are being scored by Kaggle on 30% of the test data before our presentation on Friday December 7th. I'll optimize for this for a methodical approach
  - Final models will be scored by Kaggle's remaining 70% of test data on Saturday December 8th. Which will reveal correct or incorrect assumptions and areas of improvement.

Correlation as First Factor of Feature Selection



4.2 I choose this Ridge Model to Go with for Kaggle Submission

In [50]:
```
### RIDGE MODEL
### Winner Winner Chicken Dinner

# as is now, it's unfit. so need to fit
ridge.fit(X_train_sc, y_train_log)

ridge_cv = cross_val_score(ridge, X_train_sc, y_train_log, cv=3)
ridge_cv
ridge_cv.mean()

print('Ridge CV: ' + str(ridge_cv.mean()))
print('Ridge Intercept: ' + str(ridge.intercept_))
print('Ridge Coefficients (First 3): ' + str(ridge.coef_[:3]))
print('Ridge Best Alpha: ' + str(ridge.alpha_))
print('Ridge R2 Score: ' + str(ridge.score(X_train_sc, y_train_log)))


# now have predictions for all 879 homes
ridge_predictions = np.exp(ridge.predict(X_test_sc))
print('First 3 Predictions: ' + str(ridge_predictions[:3]))


# Create submission ready info
test_master_use['SalePrice'] = ridge_predictions
submission_ridge_v1 = test_master_use[['Id', 'SalePrice']]


# Need Index=False otherwise it will create the index as another column. Only need to submit 2 columns. Id and SalePric
# Create CSV File to Submit By calling below Line
# submission_ridge_v1.to_csv('cw_ames_ridge_model_v1_100f_log.csv', index=False)

submission_ridge_v1.head()
```

```
Ridge CV: 0.9223907903895002
Ridge Intercept: 12.026866920486166
Ridge Coefficients (First 3): [0.06922495 0.07430066 0.00884748]
Ridge Best Alpha: 10.0
Ridge R2 Score: 0.9323050497410345
First 3 Predictions: [134673.93210934 159567.10061393 211797.06062443]
```

Out[50]:

	Id	SalePrice
0	2658	134673.932109
1	2718	159567.100614
2	2414	211797.060624
3	1989	105325.914538
4	625	173755.266648

Evaluating my models

- Evaluation metrics
  - MSE − mean squared error of predicted housing price and actual housing price in hold out dataset
  - *"In statistics, the **mean squared error** (MSE) or **mean squared** deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the **errors**—that is, the average **squared** difference between the estimated values and what is estimated." Wikipedia*
- Baseline score
  - Ridge R2 Baseline Score: 0.748
  - 'Overall Qual', 'Gr Liv Area' are the biggest correlating features
- How will my model will generalize to new data?
  - This will be determined by using it on the full holdout test dataset.

Results

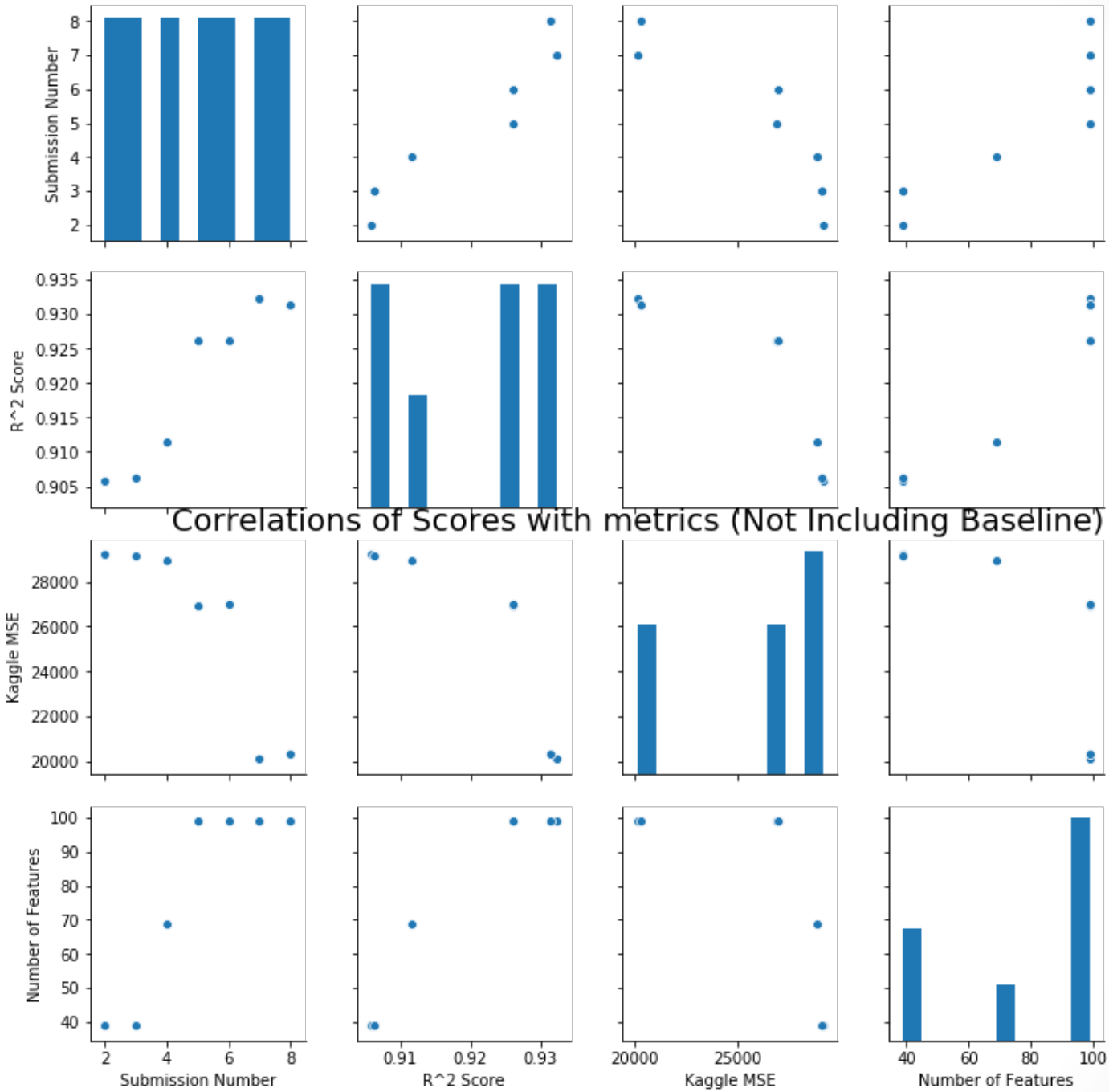
- Ridge – Winner Winner Chicken Dinner
  - 99 features, cv=3, ss, log and exp
  - R^2 of 0.932
  - Alpha of 10
  - Intercept at 12.02
  - CV of 0.922

- Lasso
  - R^2 of 0.931
  - Alpha of 0.00058
  - Intercept at 12.02
  - CV of 0.921

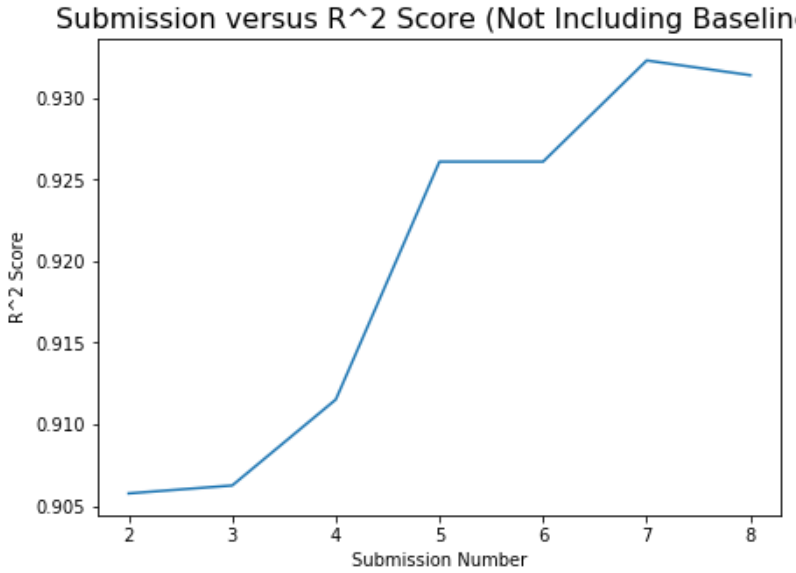
- Hold-Out model versus final submission MSE difference of 7%
  - MSE of 20,142 Kaggle versus MSE 21,647of Ridge Hold-Out

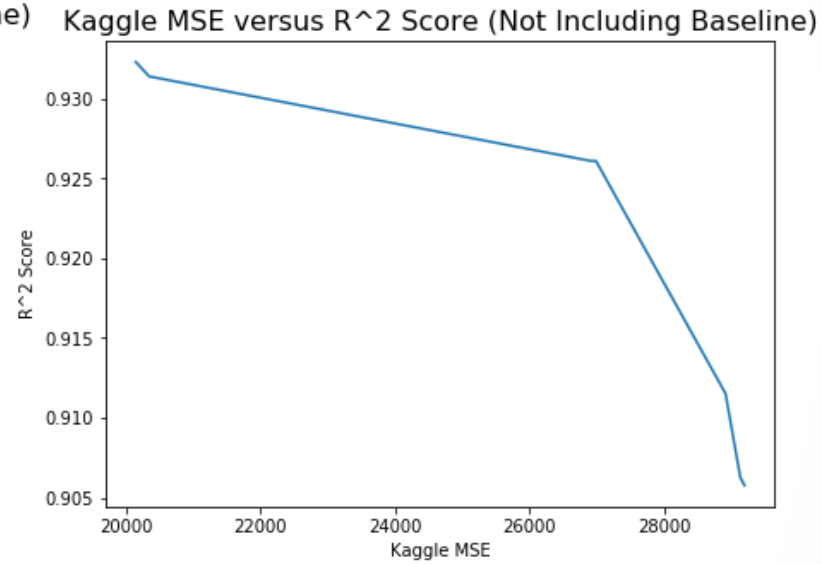
R^2 and Kaggle Scores

	Submission Number	R^2 Score	Kaggle MSE	Model Type	Number of Features
0	1	0.748192	38974.66733	Ridge	2
1	2	0.905766	29199.64552	Lasso	39
2	3	0.906252	29139.32600	Ridge	39
3	4	0.911524	28917.88712	Ridge	69
4	5	0.926102	26915.51146	Lasso	99
5	6	0.926102	26990.63567	Ridge	99
6	7	0.932305	20142.31884	Ridge	99
7	8	0.931401	20341.89054	Lasso	99


Correlations of Scores with metrics (Not Including Baseline)
Submission Number
R^2 Score
Kaggle MSE
Number of Features
Submission Number
R^2 Score
Kaggle MSE
Number of Features

R^2 and Kaggle Scores

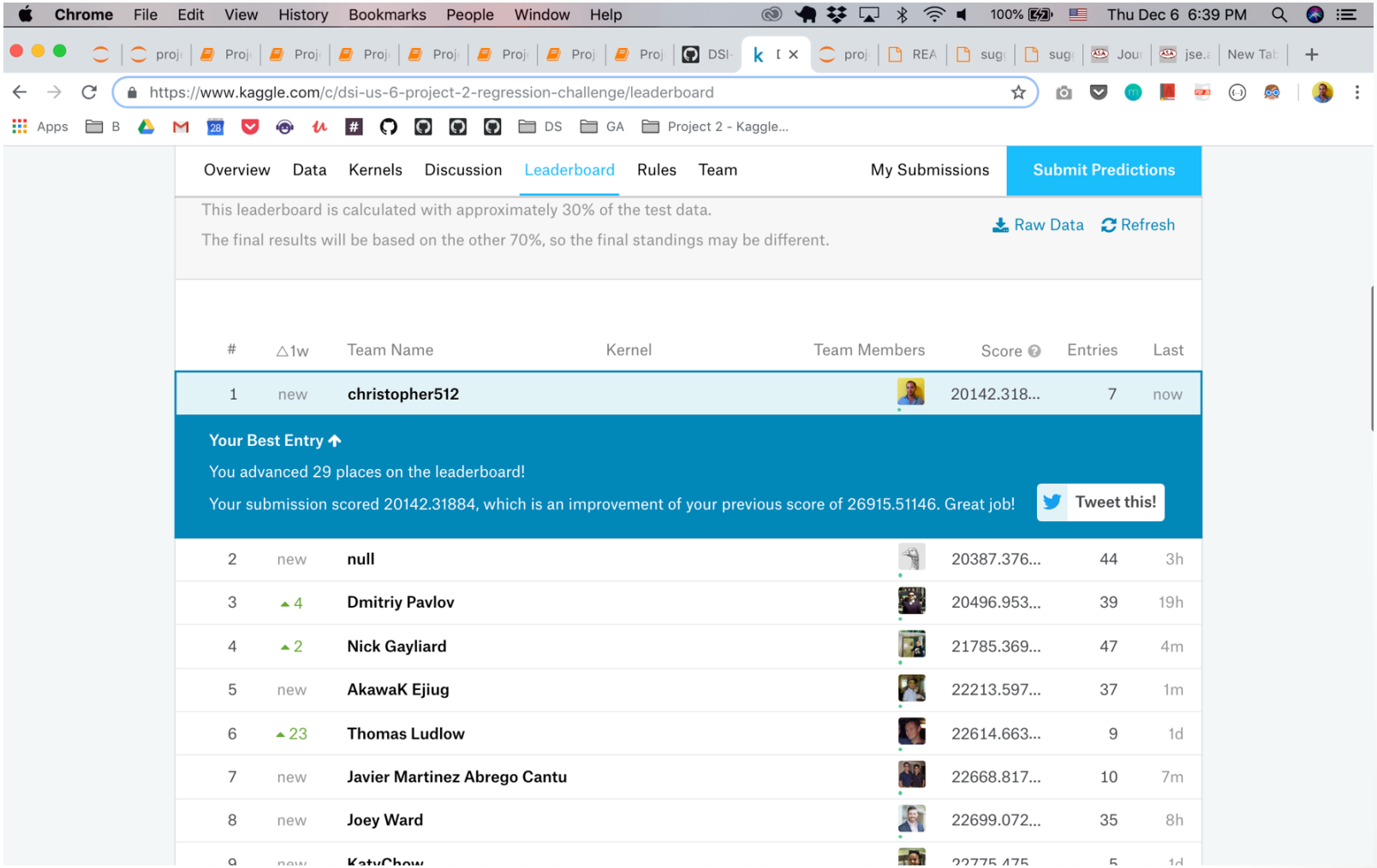




Coefficients

	Coefficients	Rank
Gr Liv Area	0.074301	1
Overall Qual	0.069225	2
Overall Cond	0.049065	3
1st Flr SF	0.046405	4
Year Built	0.044532	5
2nd Flr SF	0.042667	6
Total Bsmt SF	0.040975	7
MS Zoning_RL	0.036662	8
MS Zoning_RM	0.023598	9
Lot Area	0.023063	10
Condition 1_Norm	0.020177	11
Exterior 2nd_CmentBd	0.020038	12
MS Zoning_FV	0.019715	13
Neighborhood_Somerst	0.018006	14
BsmtFin SF 1	0.017906	15

Scoring… on 30% test data
Optimized to get first place.



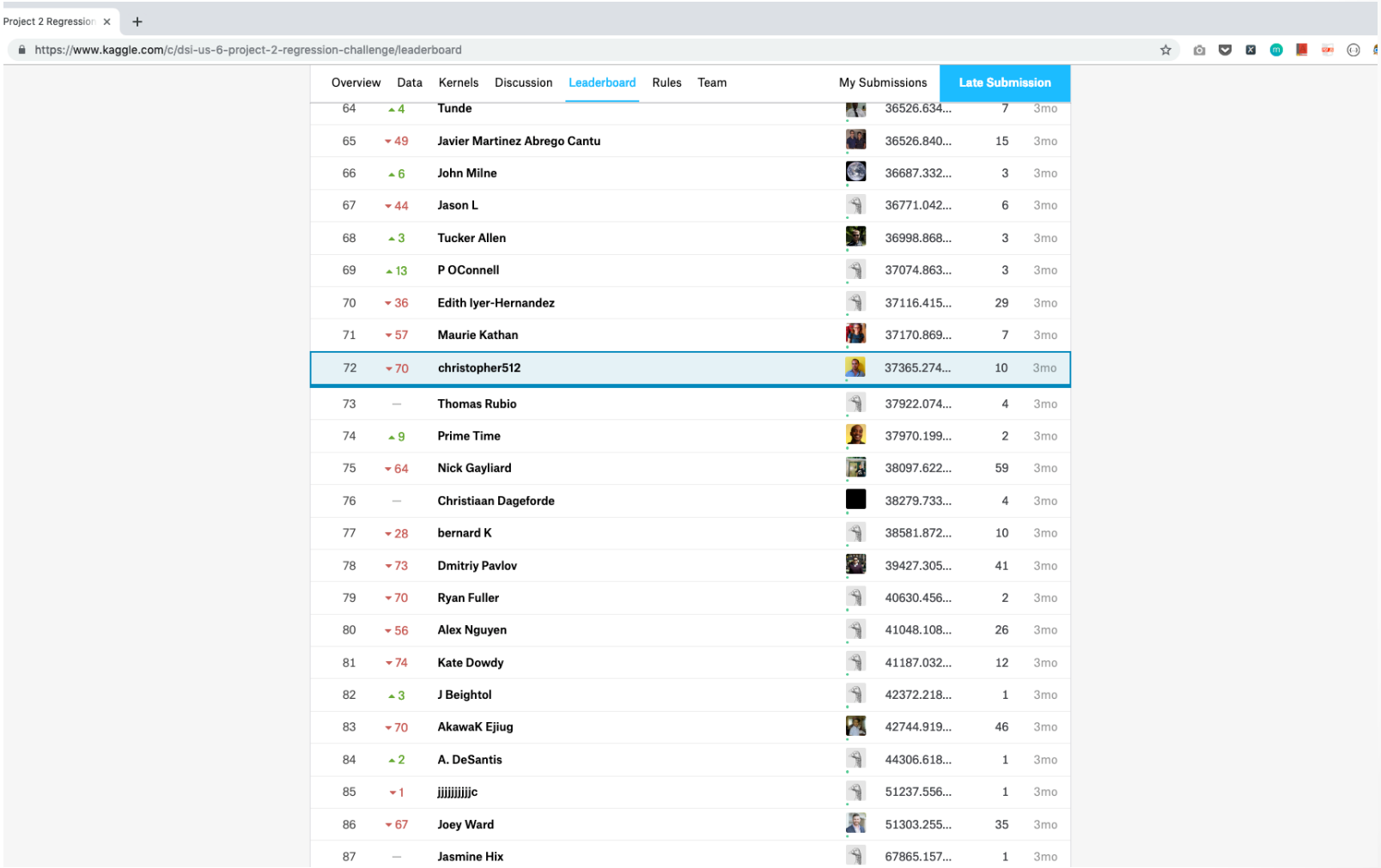
Model Performance

- For my models and the assumptions I made with the data, such as the cleaning and mapping process and leaving out the outliers, the Ridge model performed slightly better than Lasso with more features while Lasso model performed better with less features on the 30% of test scoring.

- Since the presentation was on Friday Dec 7th and we could only see Kaggle's scoring of our model on 30% of the data at that time, I optimized for that. Which I was able to in 7 attempts get to the 1st position on the leader board with a MSE of 20,142.

- The following day, on Saturday Dec 8th, the 70% of the final Kaggle Data scored our models and my models performance dropped substantially to a MSE of 37, 365. I learned that the models that performed best were the ones that left in the four outliers that I had dropped. That shows the importance of allowing the models to learn the data and make predictions with the full dataset.

Scoring… on final 70%, large drop due to dropping outliers



64	▲4	Tunde	36526.634...	7	3mo
65	▼49	Javier Martinez Abrego Cantu	36526.840...	15	3mo
66	▲6	John Milne	36687.332...	3	3mo
67	▼44	Jason L	36771.042...	6	3mo
68	▲3	Tucker Allen	36998.868...	3	3mo
69	▲13	P OConnell	37074.863...	3	3mo
70	▼36	Edith Iyer-Hernandez	37116.415...	29	3mo
71	▼57	Maurie Kathan	37170.869...	7	3mo
72	▼70	christopher512	37365.274...	10	3mo
73	—	Thomas Rubio	37922.074...	4	3mo
74	▲9	Prime Time	37970.199...	2	3mo
75	▼64	Nick Gayliard	38097.622...	59	3mo
76	—	Christiaan Dageforde	38279.733...	4	3mo
77	▼28	bernard K	38581.872...	10	3mo
78	▼73	Dmitriy Pavlov	39427.305...	41	3mo
79	▼70	Ryan Fuller	40630.456...	2	3mo
80	▼56	Alex Nguyen	41048.108...	26	3mo
81	▼74	Kate Dowdy	41187.032...	12	3mo
82	▲3	J Beightol	42372.218...	1	3mo
83	▼70	AkawaK Ejiug	42744.919...	46	3mo
84	▲2	A. DeSantis	44306.618...	1	3mo
85	▼1	jjjjjjjjjc	51237.556...	1	3mo
86	▼67	Joey Ward	51303.255...	35	3mo
87	—	Jasmine Hix	67865.157...	1	3mo

Conclusions & Takeaway

- In conclusion, if I were to do the project over with the goal to minimize mean squared error for the final 70%, it would have been best to have left in the four outliers I had taken out. In terms of how my assumptions would have helped our hindered the predictions, that is hard to infer. Let the model perform and predict to see.

- My key takeaway is that it is easiest to improve the model performance very significantly in the beginning, but that making additional changes and improving it from an already good model to a great model is difficult and is a decision to be made in light of your objectives, time, and data resource constraints.