

# Housing Price Prediction

Kaggle Competition

The Avengers:

Andrew Dodd



Michael Goldman



Peter Yang



Roger Ren



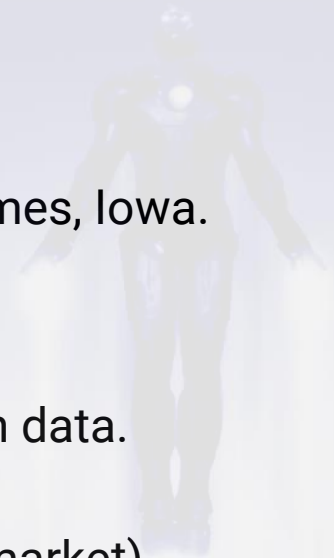
# Housing Price Prediction

## Main objective:

Develop accurate models for estimating housing price in Ames, Iowa.

## Sub-objectives:

- a. Explore dataset, variable distributions, correlations, clean data.
- b. Perform feature engineering.
- c. Look for outside effects (inflation, seasonality, housing market).
- d. Cluster dataset into sub-datasets.
- e. Identify which variables significantly impacted sales price.
- f. Select and tune models.
- g. Create train, validate, and prediction pipeline.



# Presentation Overview

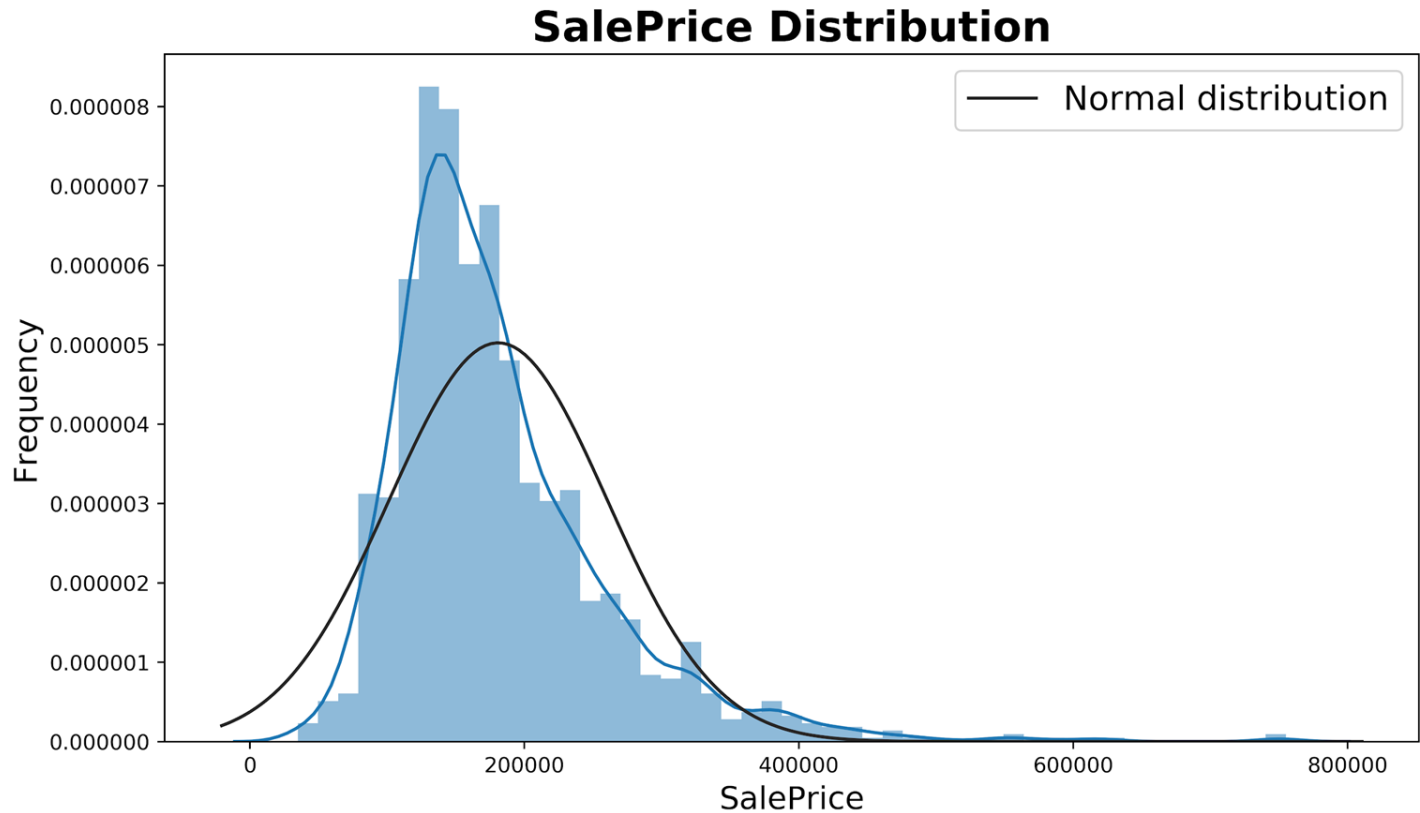
1. THE DATA
2. THE MODELS
3. THE PIPELINE
4. THE RESULTS
5. THE FUTURE



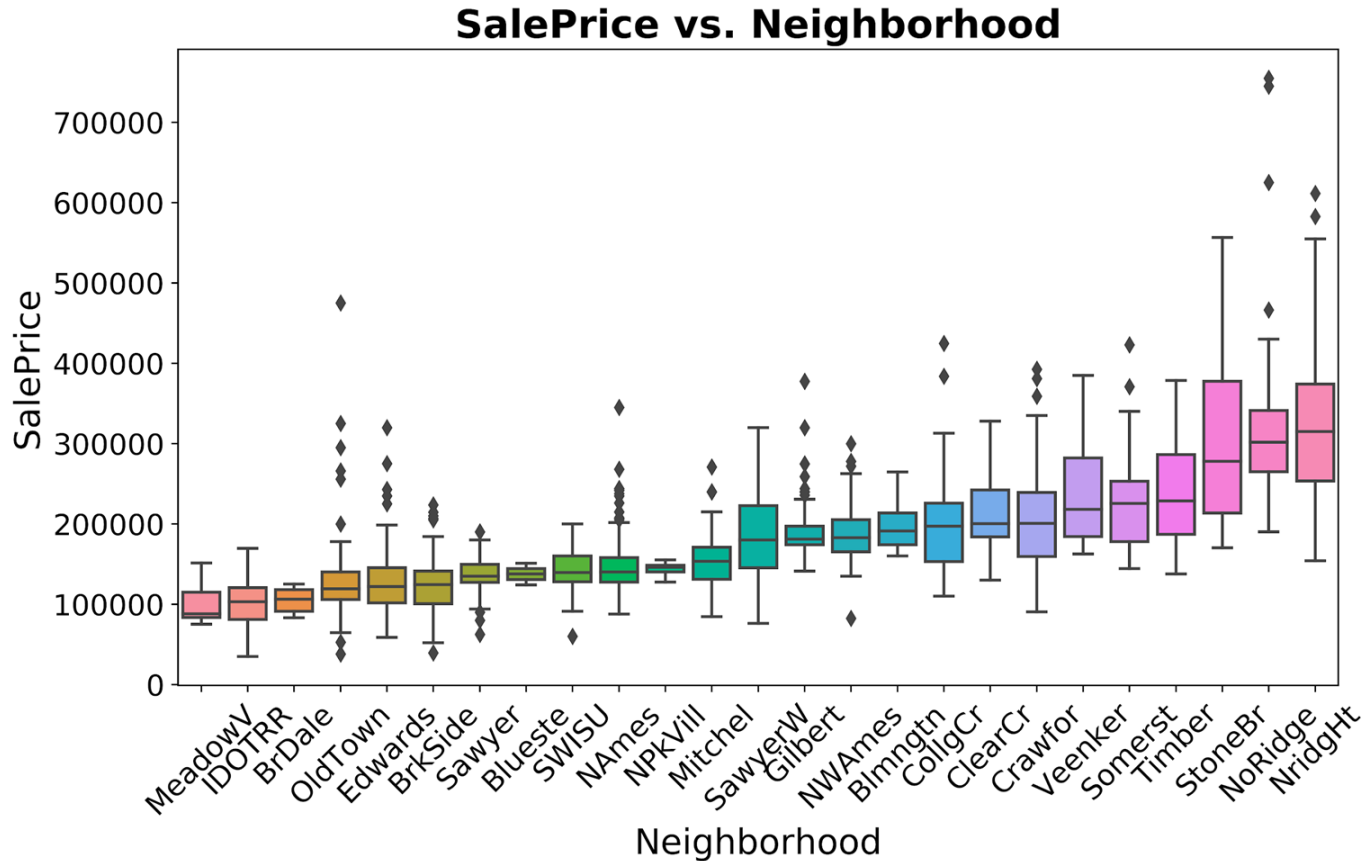
# 1. THE DATA



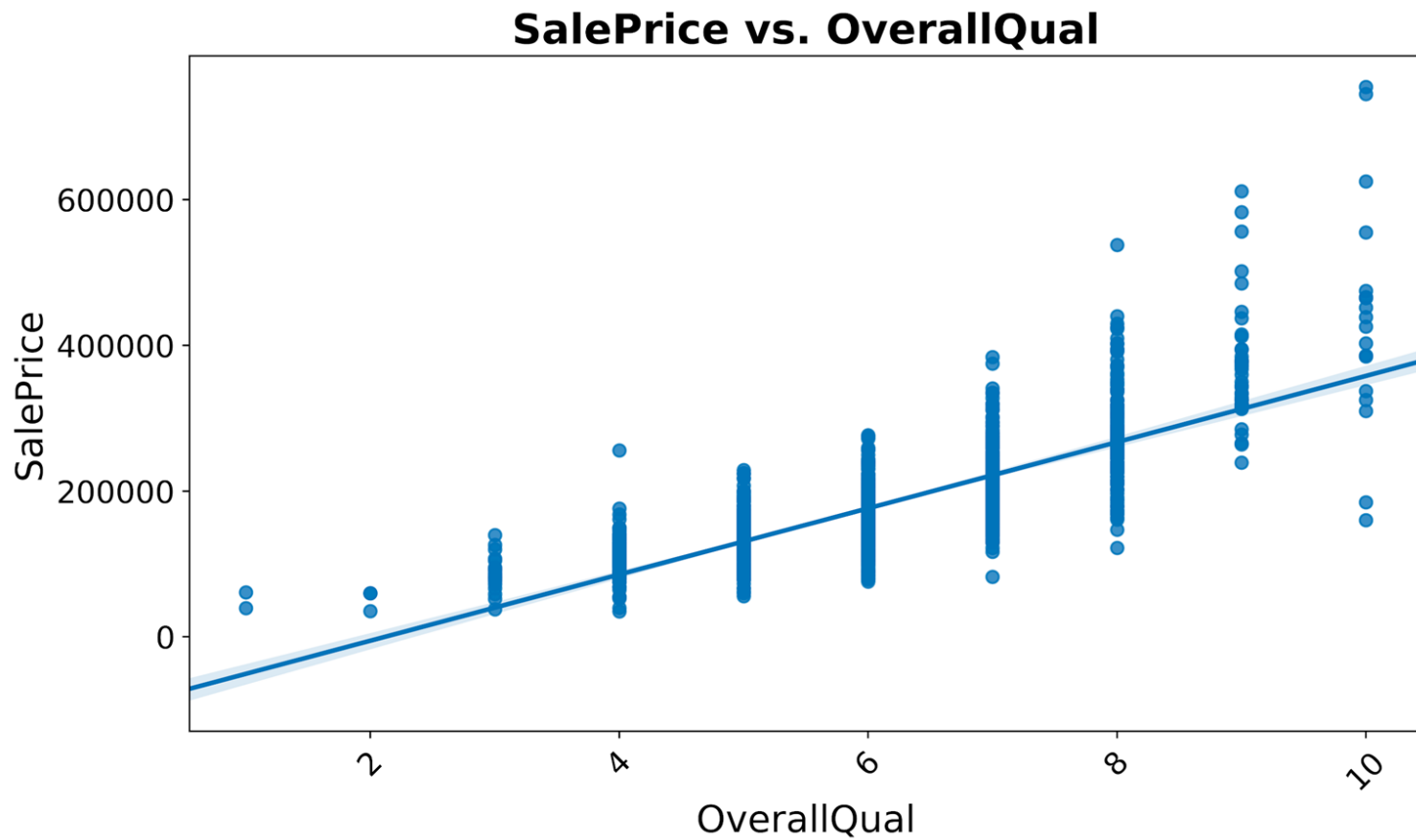
# Dataset Exploration



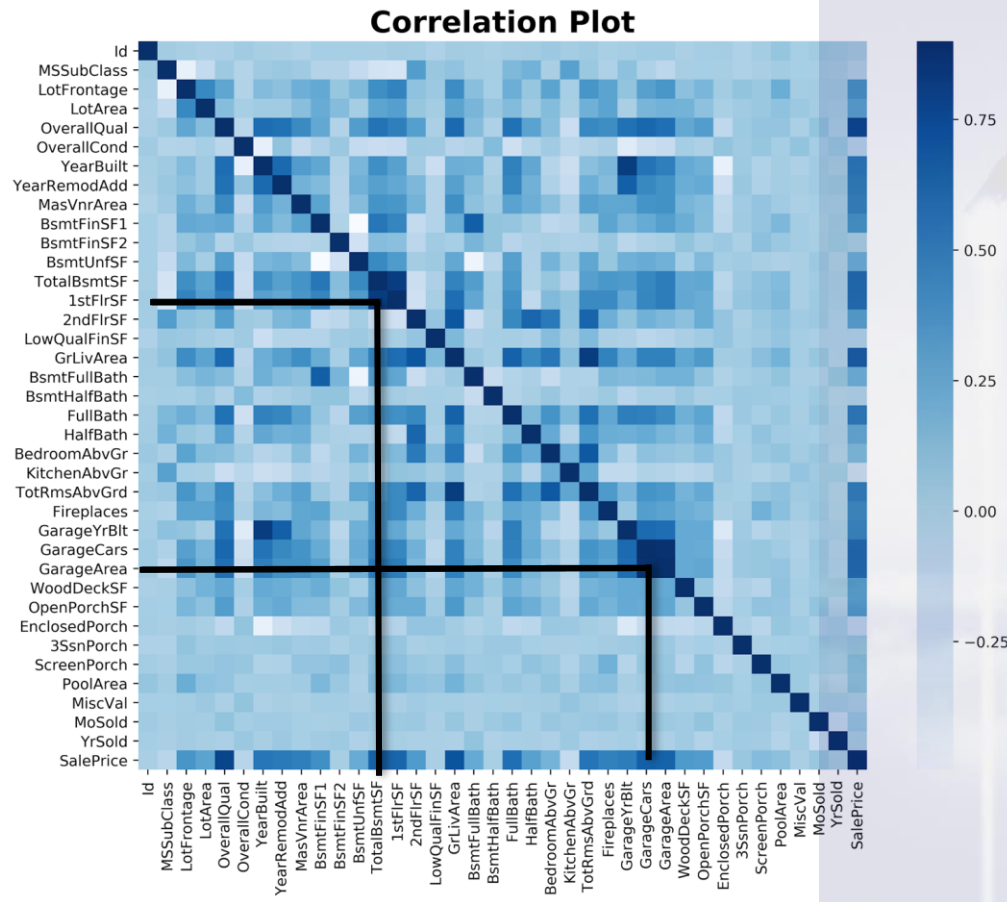
# Dataset Exploration



# Dataset Exploration



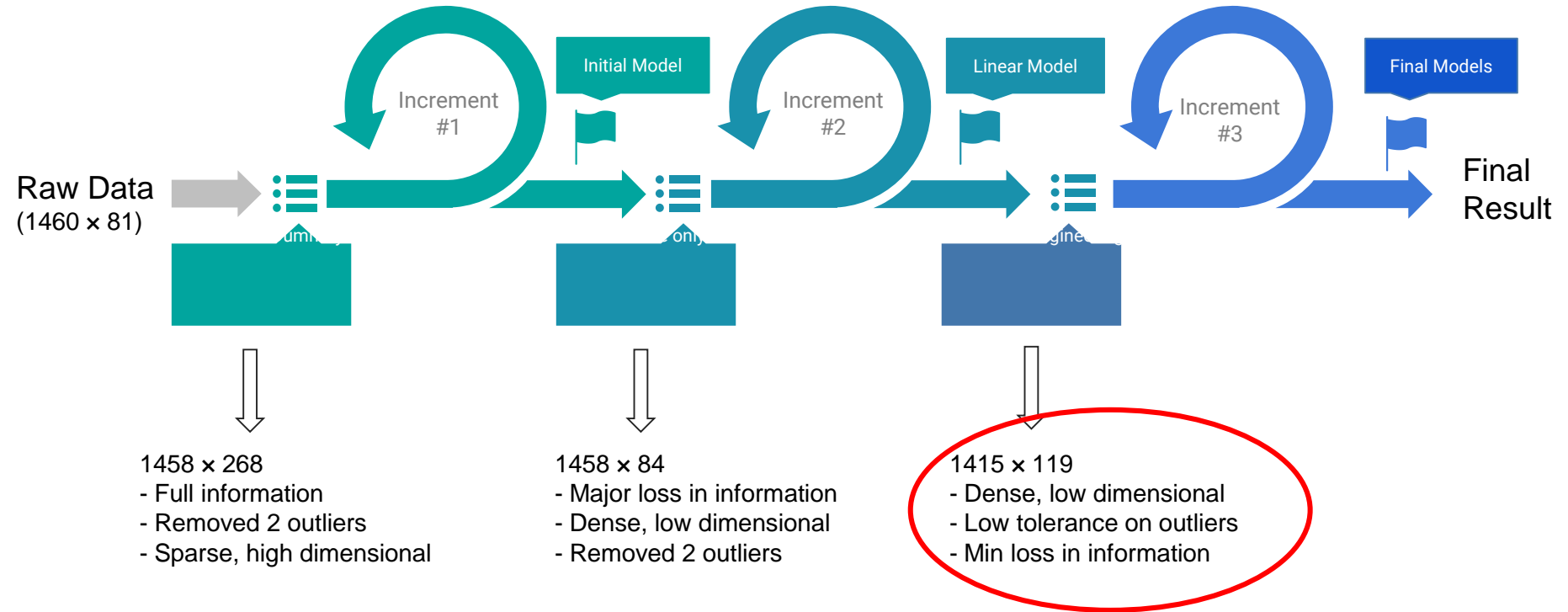
# Dataset Exploration



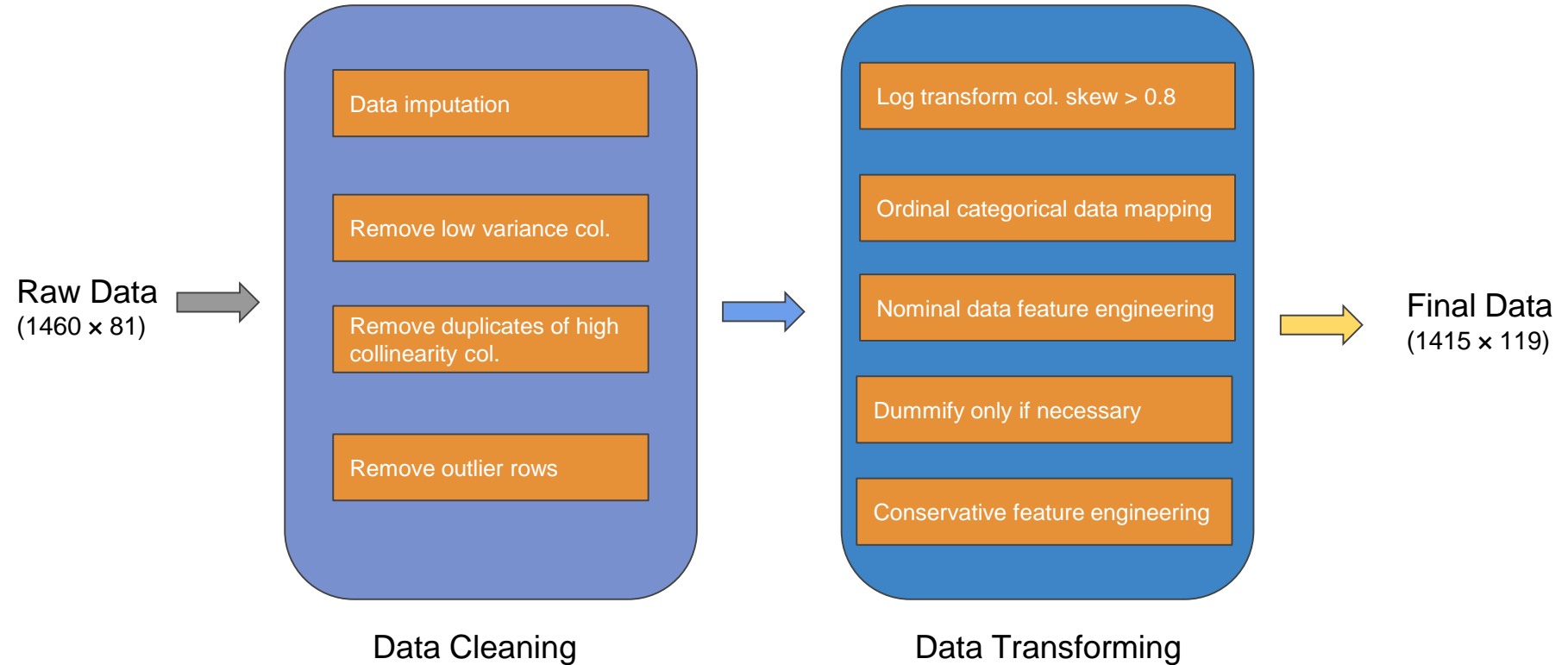


# Data Preparation: Overview

## Agile Approach

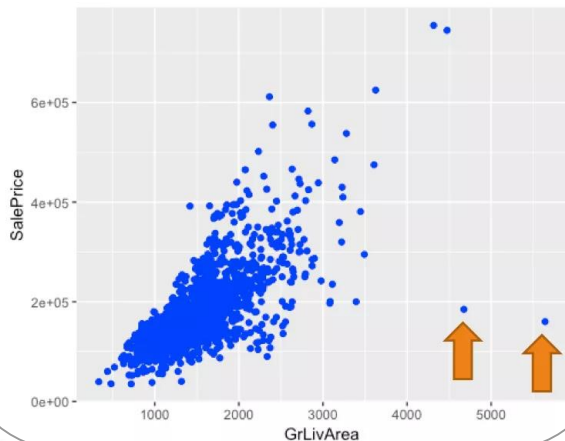


# Data Preparation: Pipeline

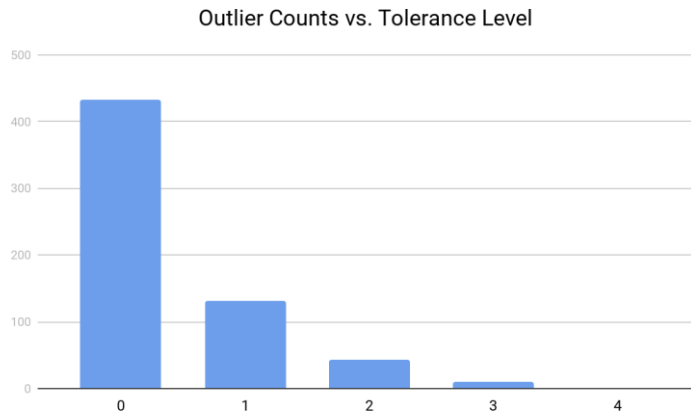


# Data Preparation: Outliers Tolerance

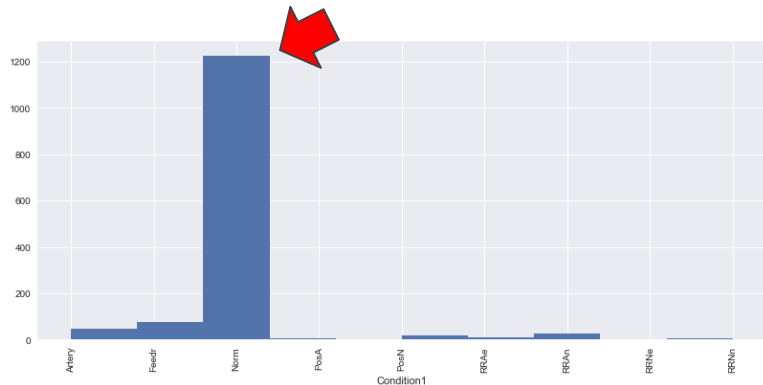
- Remove outliers by plotting
- Some obvious ones
- Lack of systemic approach



- Remove outliers by  $\pm 3$  std or 1.5 IQR
- Can set 'tolerance level'



# Data Preparation: Feature Engineering



## Simple dummify:

- 1 col → 9 col
- Preserve all information
- Data set becomes sparse

## Label by frequency:

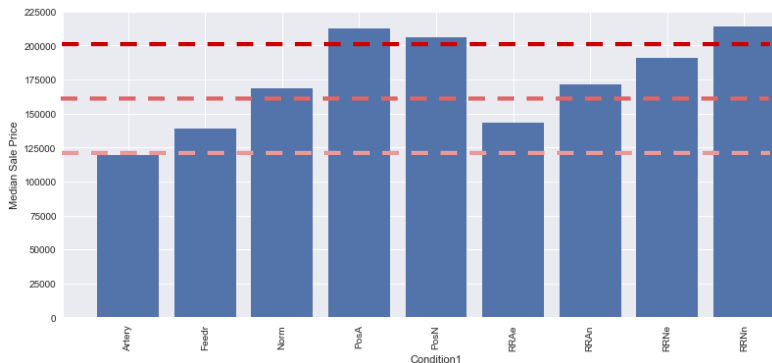
- Lost information
- Arbitrary assignment

## Significant factor:

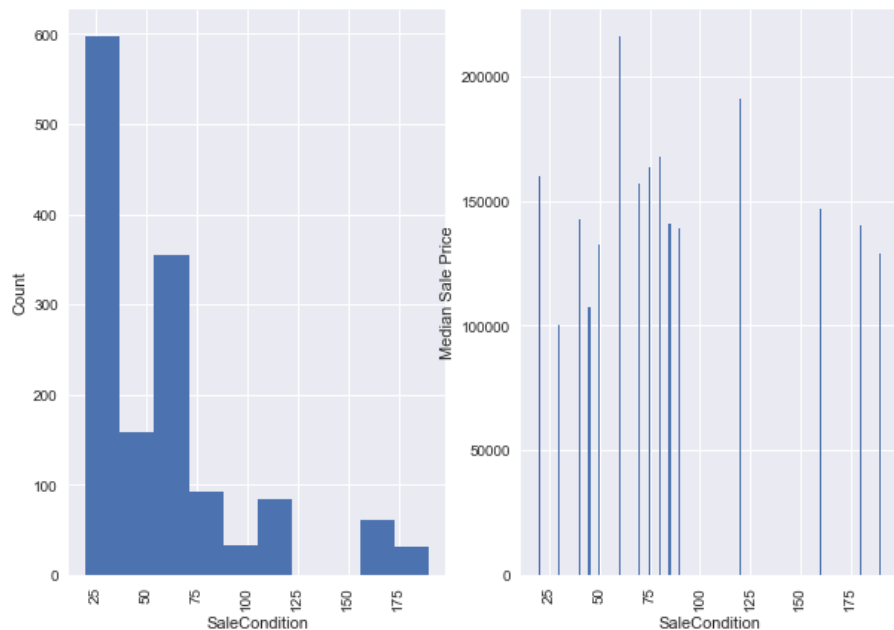
- Preserve the information of a single / a few dominant factor(s)

## Group and label:

- Keeps majority of information between 'Factor' and 'SalePrice'
- Avoid unnecessary cols count increase
- Keeps dataset dense



# Data Preparation: Only 1 Dummy



**'MSSubClass'** column:

- Appears to be numerical at first
- It is actually categorical data
- Do not exhibit 'dominant' factor

# Time Trends

## a. Inflation

- The US inflation rate is approximately 2%
- This might lead to a ~10% increase in houses over a 5 year span

## b. Seasonality

- Quantity of sales is higher in middle of year

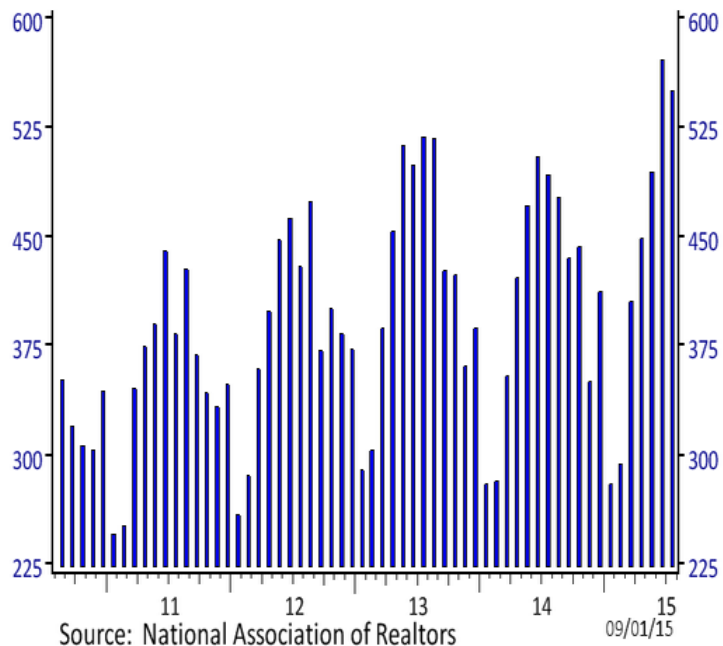
## c. Housing Market

- Do the changes in the housing market (i.e. housing crash) affect the home prices in Ames, Iowa?



### Existing Home Sales

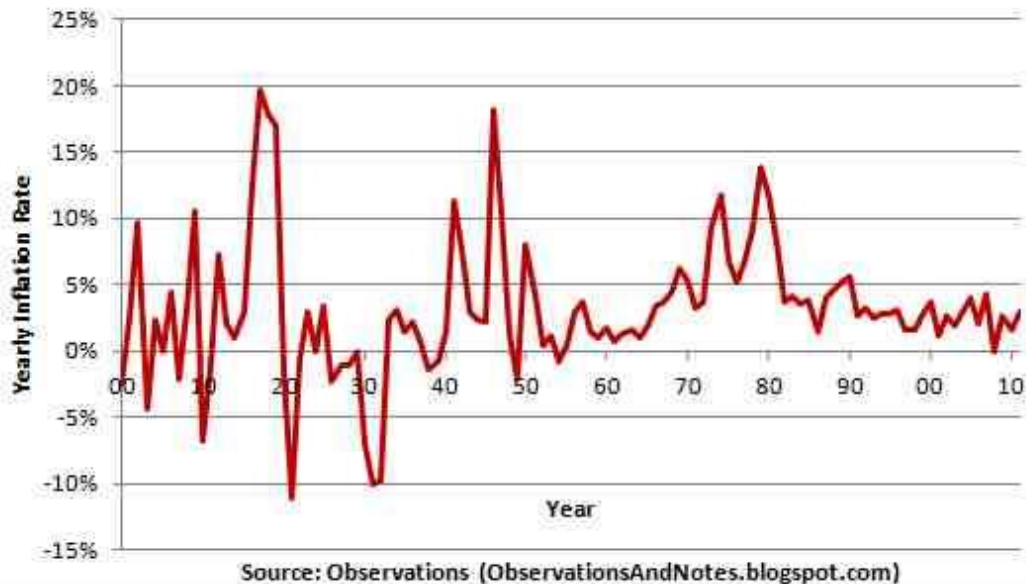
Raw Count (not seasonally adjusted), Thous



**Seasonality changes in number of homes sold**

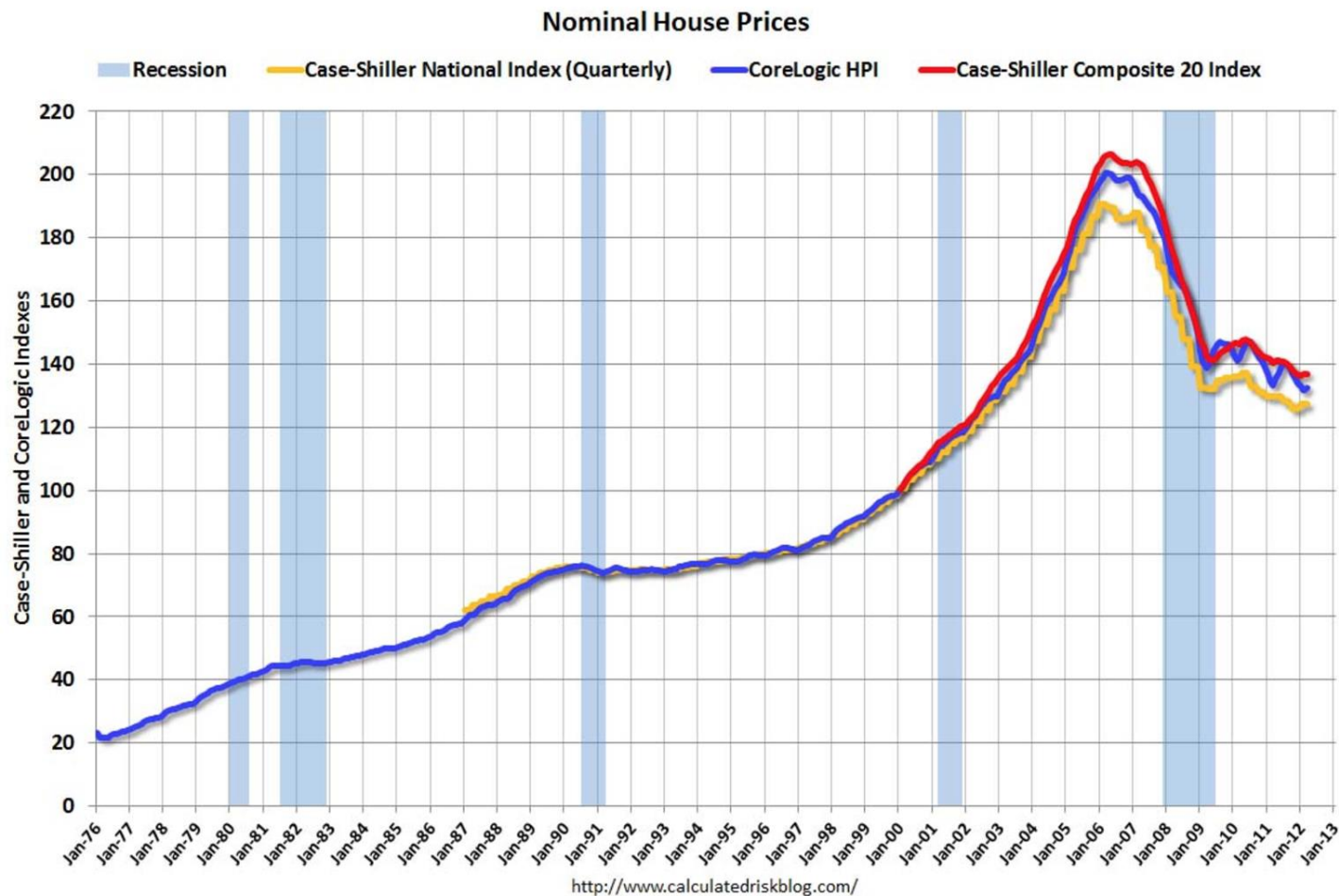
### U.S. Yearly Inflation since 1900

1900-2011 (CPI-U)



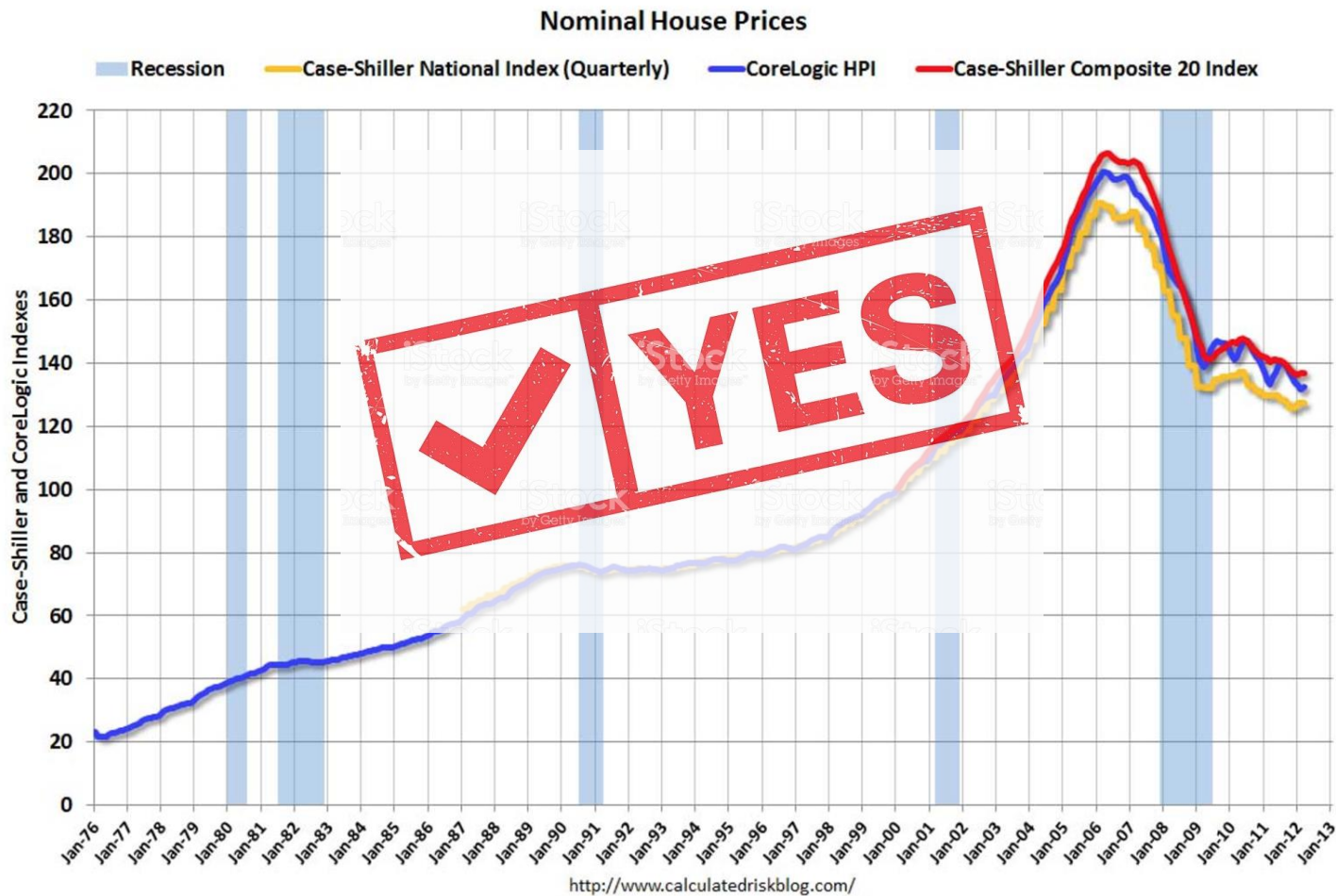
**Inflation rate versus year**

# Do home prices trend strongly with time in the United States?



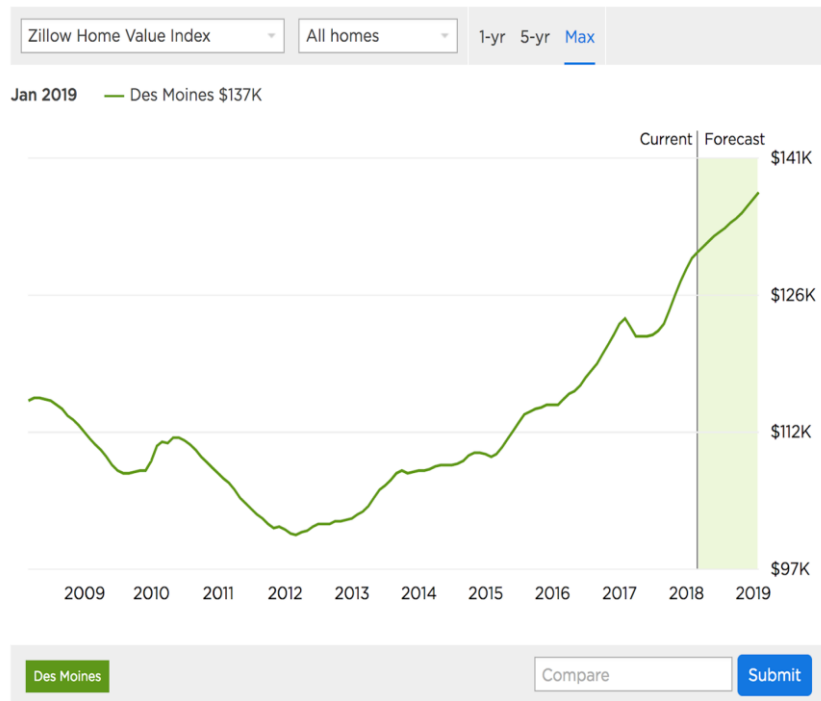


# Do home prices trend strongly with time in the United States?



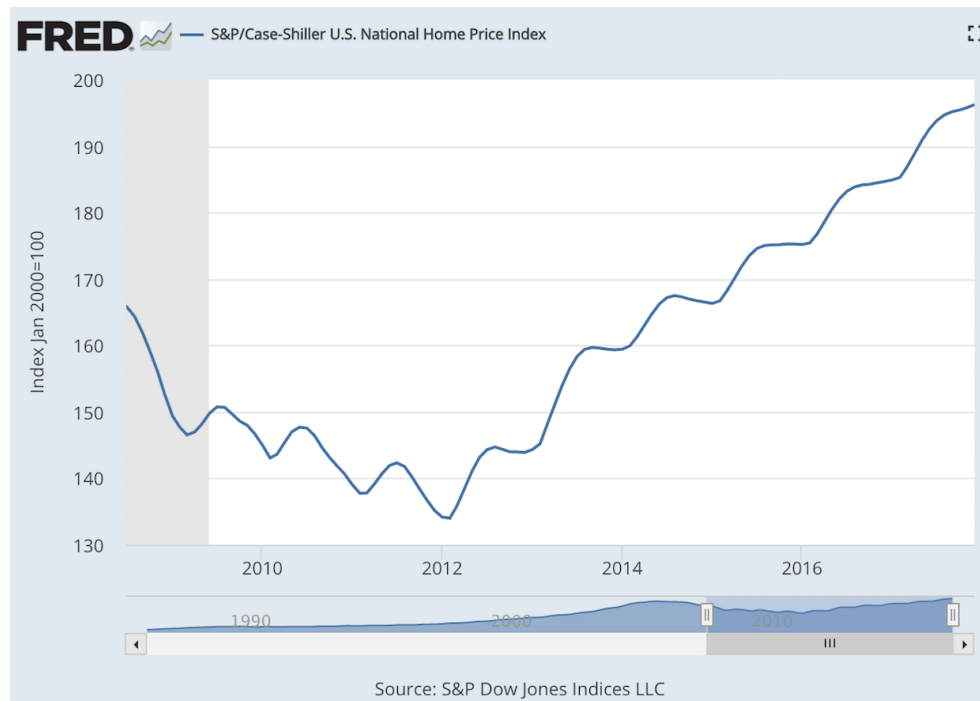
# Does there seem to be a reasonable level of price correlation between Des Moines and US National Home Price Index?

**Below left: Sales price in Des Moines, Iowa, 37 miles south of Ames.**



[View Data Table](#)

**Below right: US National Home Price index.**



# Does there seem to be a reasonable level of price correlation between Des Moines and US National Home Price Index?

Below left: Sales price in Des Moines, Iowa, 37 miles south of Ames.

Below right: US National Home Price index.



[View Data Table](#)



# Modeling the effect of time...

- Transform SalePrice column based on scaled inverse of US home index
- Steps:
  1. Create train/test split
  2. Normalize SalePrice column w.r.t. time before fitting model.
  3. Train/fit model using these normalized prices as `y_train`
  4. Predict with this model
  5. Reverse the transform
  6. Try a variety of time trends and see which trends (if any) decrease the rmsle.

Did this work?



# Modeling the effect of time...

- Create transform using home price index to normalize SalesPrice column based on average US sales price
- Steps:
  1. Create train/test split
  2. Normalize SalePrice column w.r.t. time before fitting model.
  3. Train/fit model using these normalized prices as `y_train`
  4. Predict with this model
  5. Transform predictions back to original time varying world
  6. Try a variety of time trends and see which trends (if any) decrease the rmsle



Did this work?

## 2. THE MODELS



# Models

- a. Linear Models
  - Ridge
  - Lasso
  - Elastic Net
- b. Random Forest
- c. Boosting Models
  - Gradient Boosting
  - XGBoost
  - Light Gradient Boosting

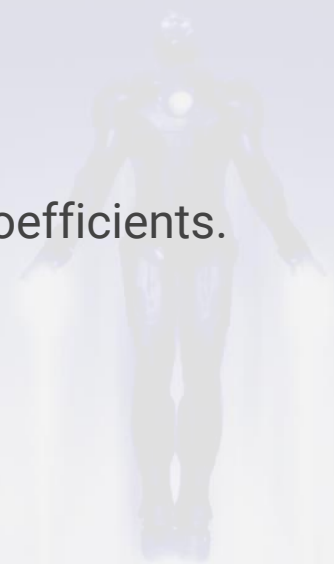


# Linear Models

- a. Ridge
  - Want to minimize the impact of weak prediction coefficients.
- b. Lasso
  - Want to drop unnecessary coefficients.
- c. Elastic Net
  - A combination of both Lasso and Ridge.

Pros: They provide a good baseline and are relatively easy to implement.

Cons: Assume a linear relationship between variables and sale price. Suffer from multicollinearity.



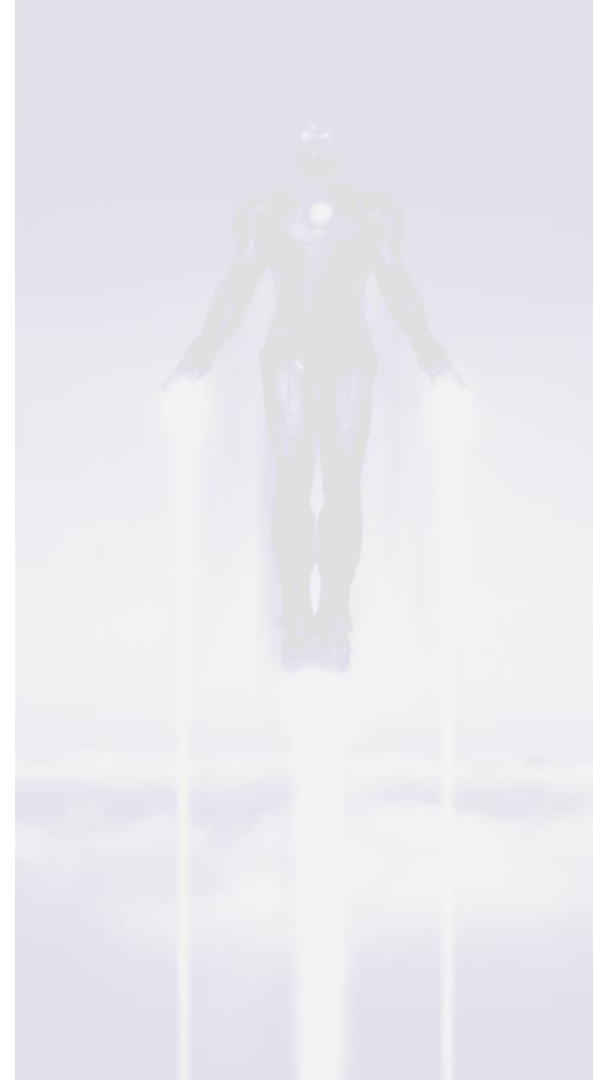
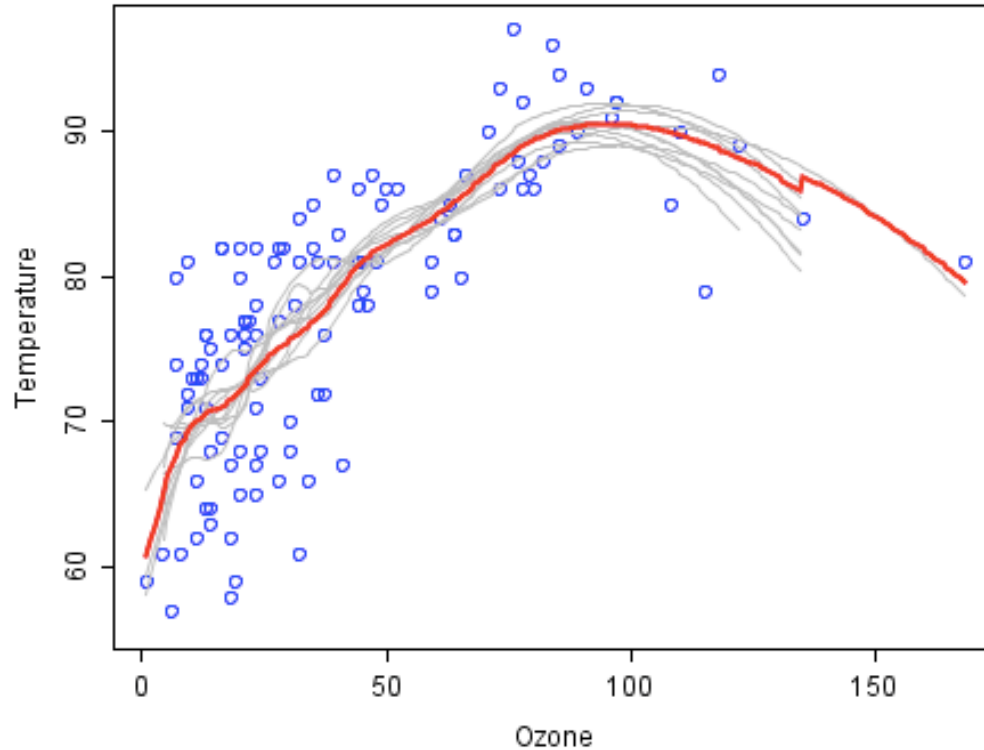


# Random Forest

- It is a large number of decision trees randomly selecting a limited number of factors at each split.
  - a. Pros:
    - Tends to not overfit when creating a large enough number of trees.
    - This works because all the individual trees are uncorrelated. Weaker predictive trees will average out to a more predictive stronger learner.
  - b. Cons:
    - Although it can handle multicollinearity for predictions and feature selection, we cannot be certain that the feature importance for multicollinearity factors is completely accurate.
      - With enough trees, this should be addressed, but can never be certain we have enough trees to address this.
    - Accuracy can suffer when predicting on data it has not seen before.
      - Since there are a maximum number of leaf nodes, there can only be that many different prices predicted.

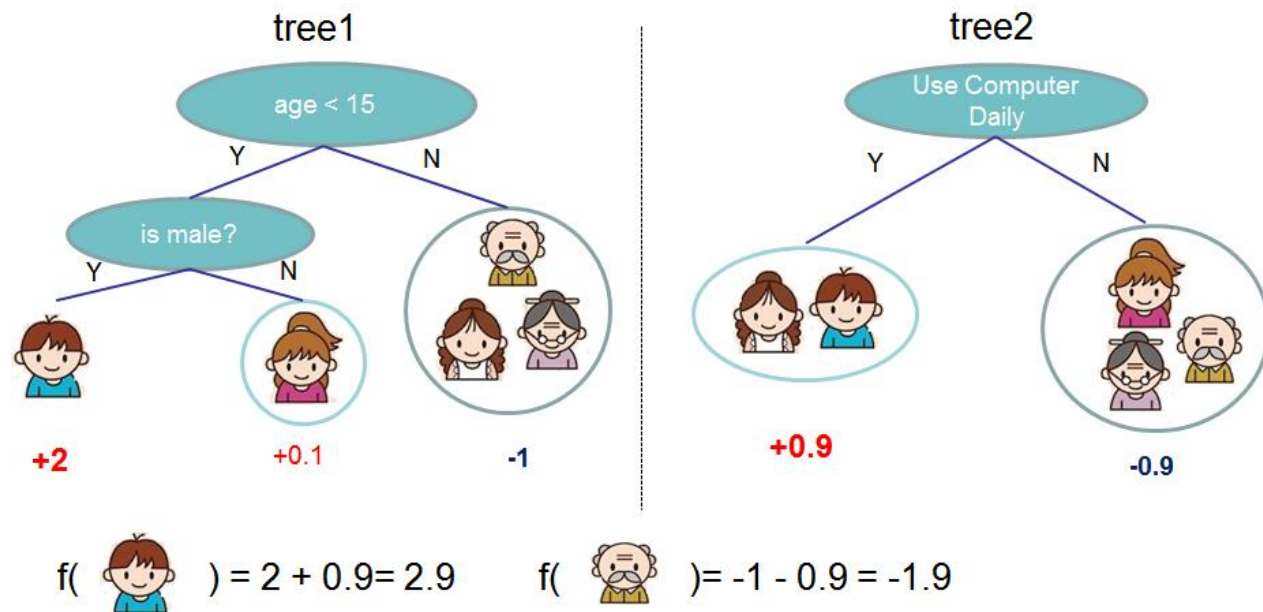


# Random Forest - an example



# Boosting Methods

- a. We used Gradient Boosting, XGBoost, Light Gradient Boosting
- Predicting on the residual of a decision tree



# Boosting Methods

## a. Pros:

- Very accurate. As the number of trees increase, the closer our prediction will be to the true value.
- A bit faster than random forest.

## b. Cons:

- Susceptible to overfitting the data, since it is predicting on the residual of a pervious tree.



# Optimizers Used

- a. Grid Search
  - Brute force method.
  - Put in a list of parameters and it generates the model with every combination of the parameters listed.
- b. Bayesian Optimization
  - Initialize the regression based on values put into the regression.
  - Explores the model to find the global maximum based on the greatest widths of the confidence interval.
  - Exploits the the max iteratively.

Ran the optimizers twice. First time with a large scope search (eg: .001-10). Second time with a more accurate scope within the more 'correct area'.



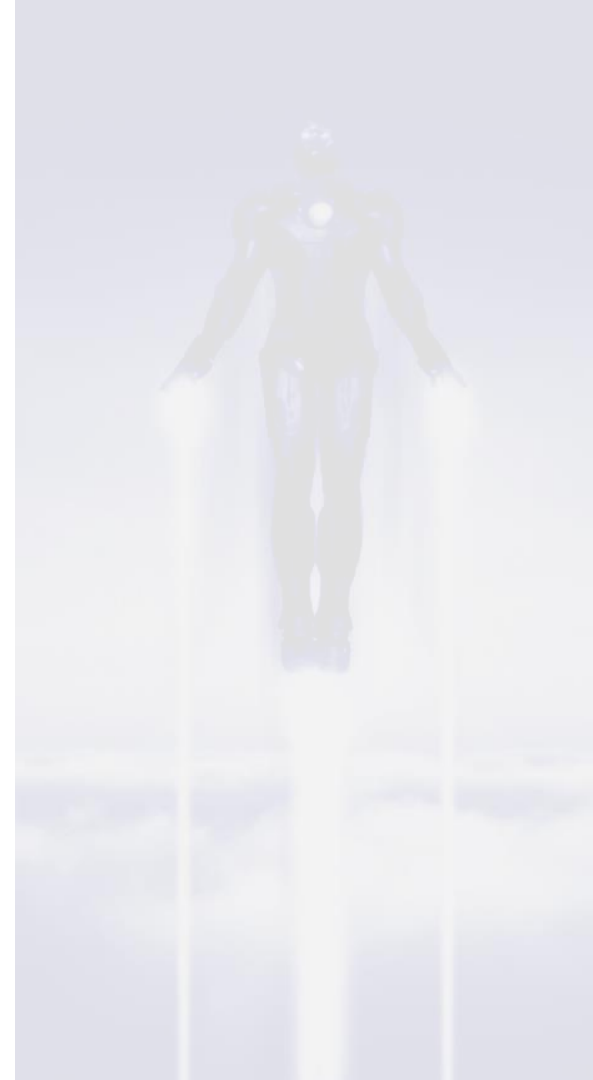
**~Now Some Graphs~**



# TABLE OF CVs

Model	CV Score (approx.)
Lasso regression	.120
Ridge regression	.120
Elastic net regression	.120
Random Forest	.125
XGBoost	.118
GBoost	.118
LightGBM	.118

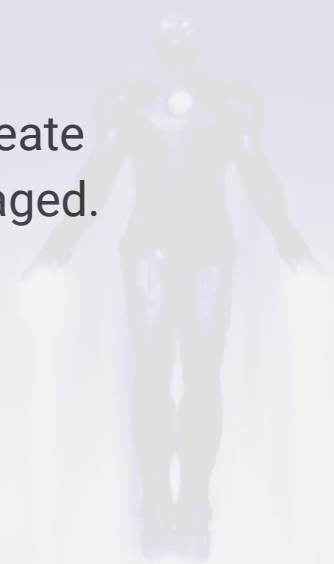
### 3. THE PIPELINE





# Object oriented modeling

- **Goal:** Create a way to quickly implement, train, validate and create predictions from multiple models that can be stacked or averaged.
- Model class
  - Functions:
    - `init(model_type)`
    - `train_validate(data)`
    - `train_predict(data)`
- Stacking class
  - Functions:
    - `init(metamodel_type)`
    - `train_validate_predict(base_model_predictions)`



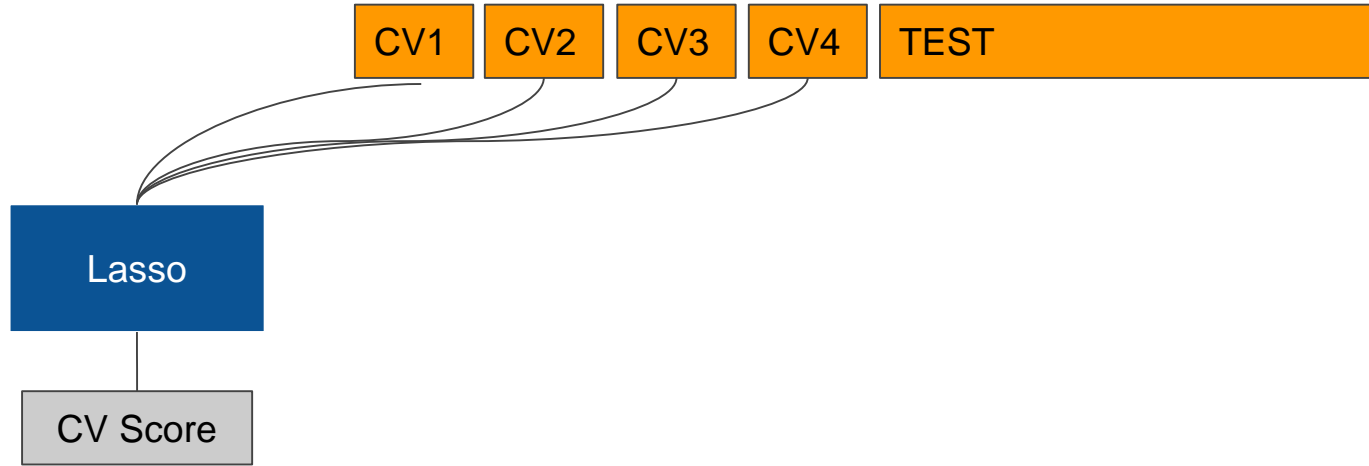
# Stacking Approach

TRAIN

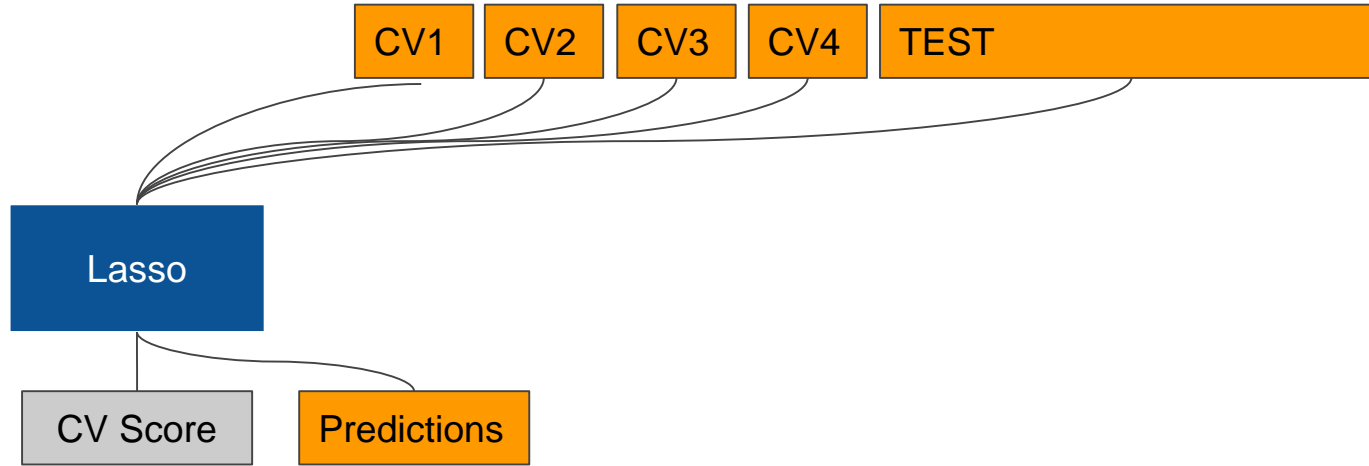
TEST

Lasso

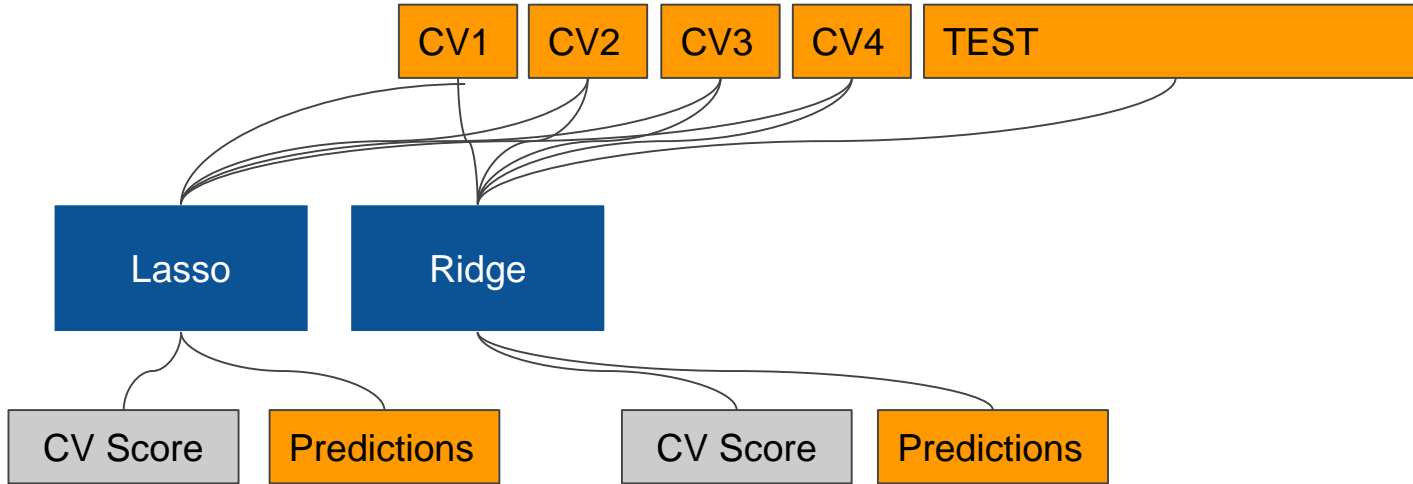
# Stacking Approach



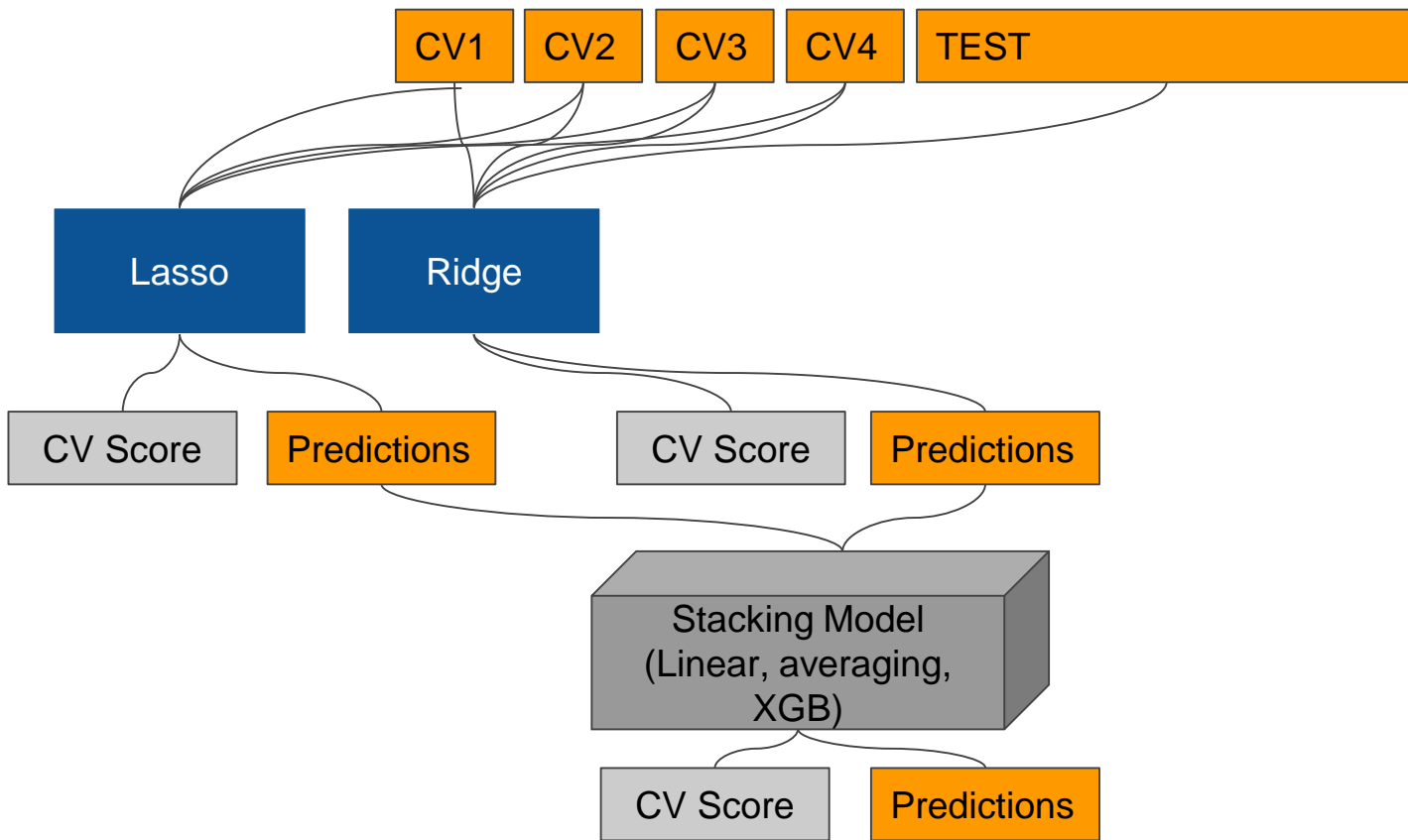
# Stacking Approach



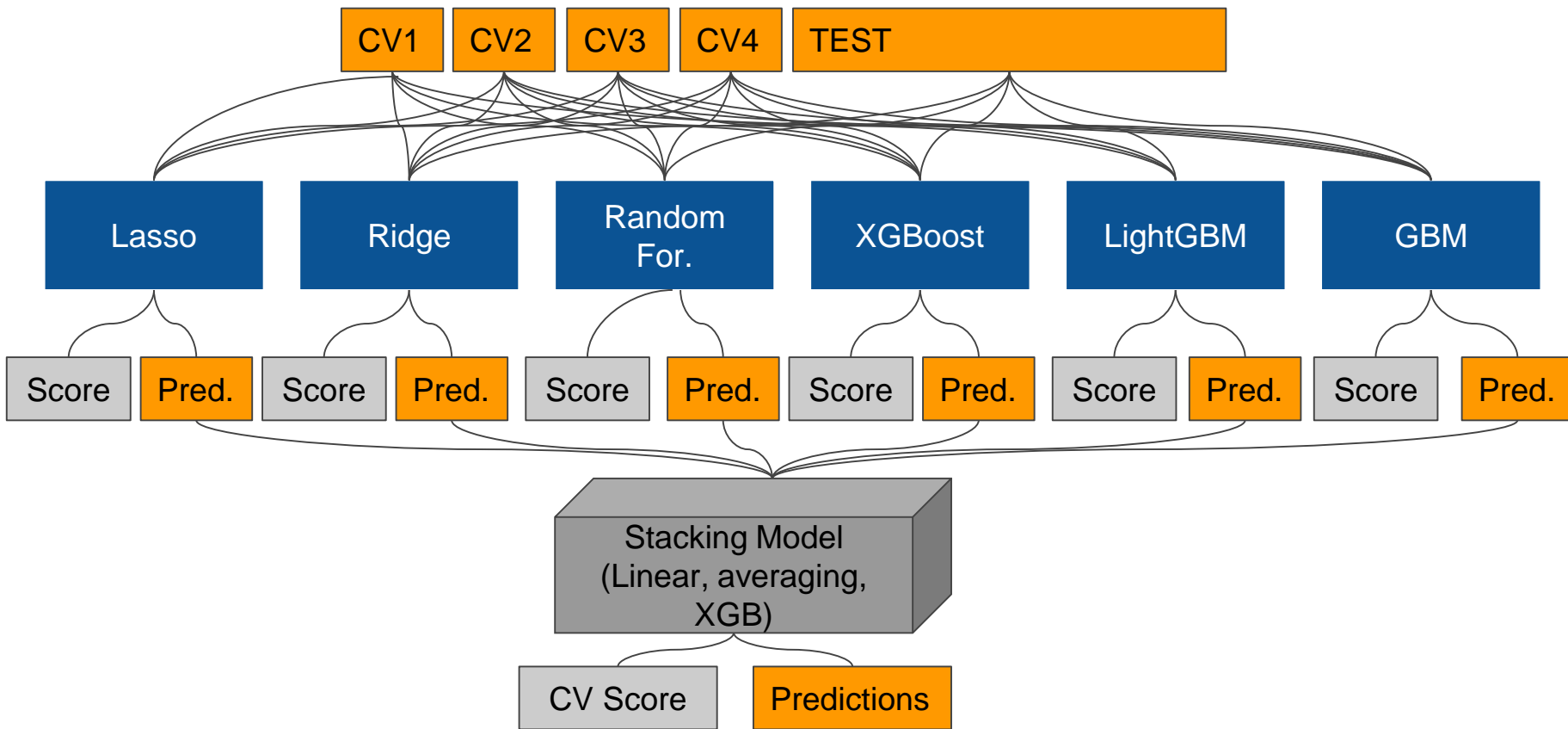
# Stacking Approach



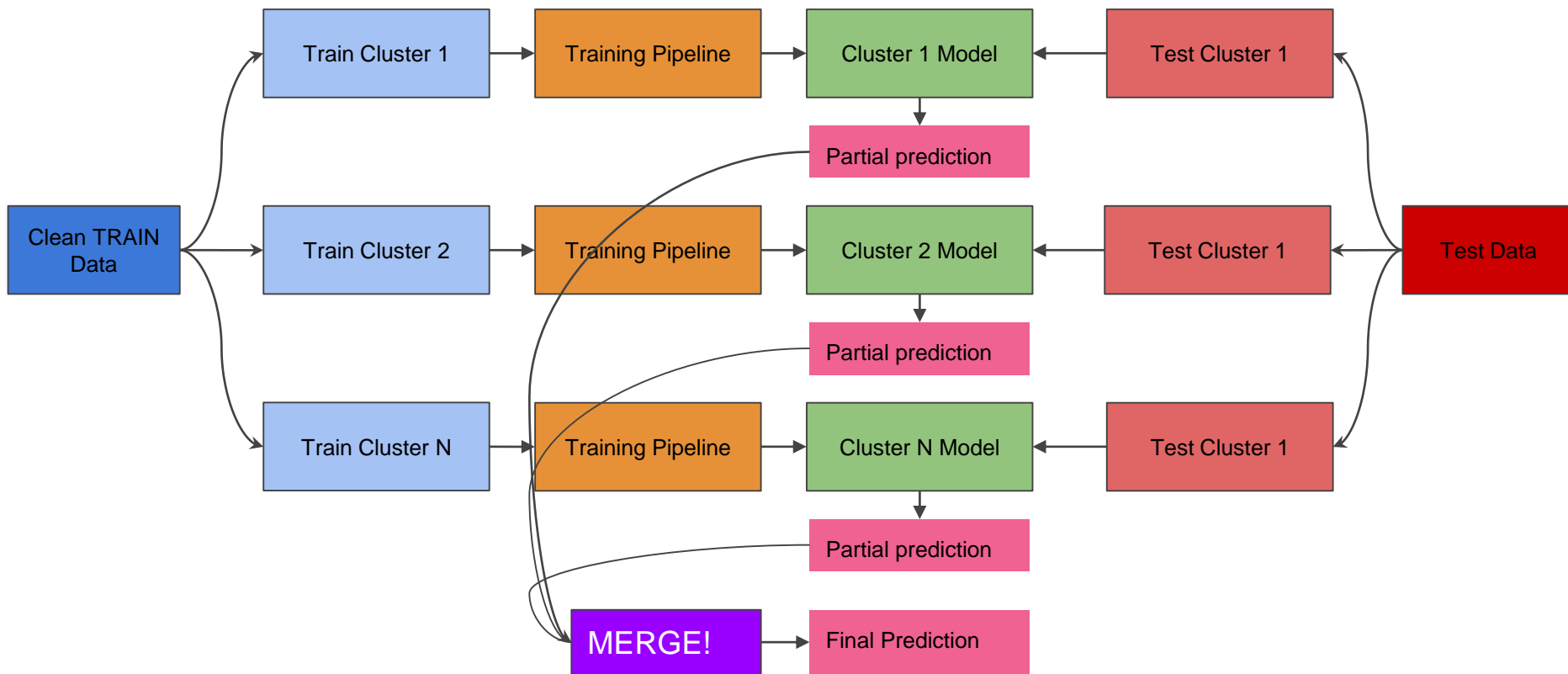
# Stacking Approach



# Stacking Approach



# Clustering Exploration: Overview





# Clustering Exploration: Pipeline

- **Trade off:**

1. Cluster sample size
2. Homogeneous cluster

- **High dimension cluster:**

1. Reduce dimension
2. Keep AMAP info

- **Cluster algorithms:**

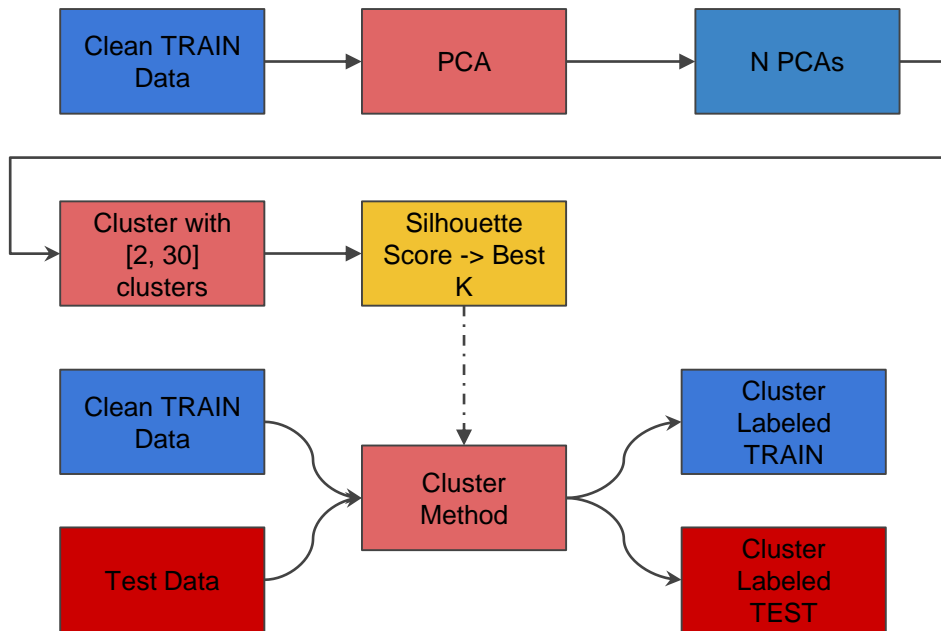
1. KMeans, Agglomerative, Gaussian Mixture...

2. Distance metrics:

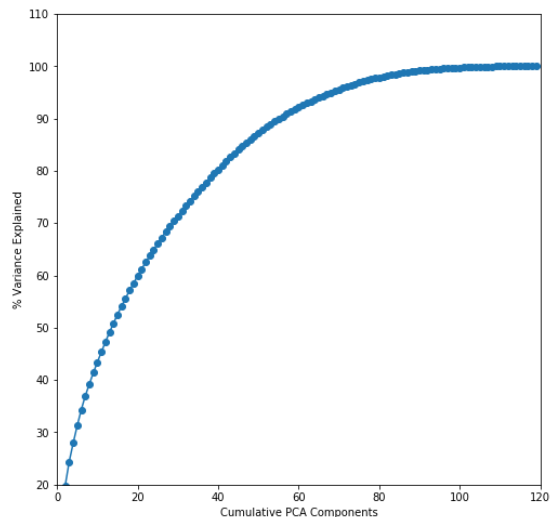
['cityblock', 'cosine', 'euclidean', 'l1', 'l2', 'manhattan']

- **Feature selection:**

1. Filtering
2. Wrapping



# Clustering Exploration: Pipeline



54 PCAs explains 90%



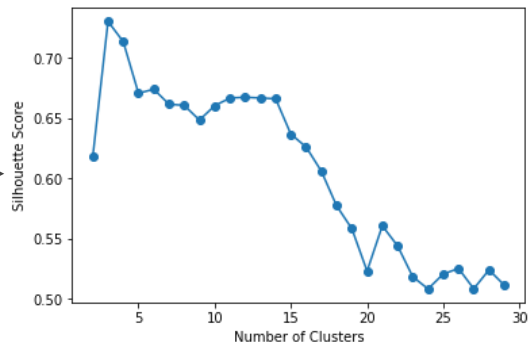
Silhouette  
Score -> Best  
K

Cluster  
Labeled  
TRAIN

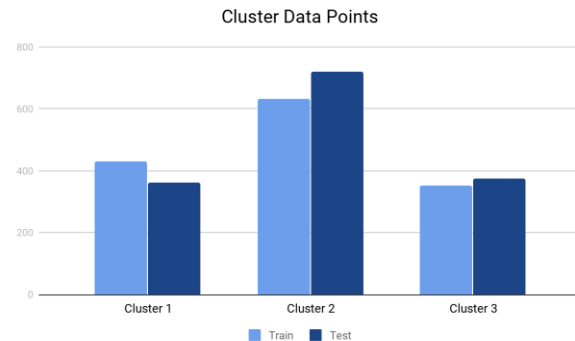
Cluster  
Labeled  
TEST

# Clustering Exploration: Method and Metric

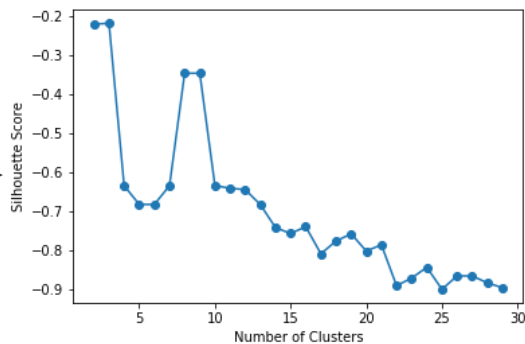
KMeans



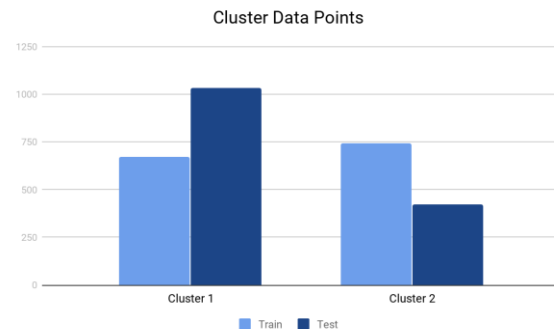
3 Clusters



Agglomerative



2~3 Clusters



## 4. THE RESULTS



# Results

- Placed 898/4376, with a leaderboard score of .12186
- This means our CV error was not a great match to our real test error

898

new

**The\_Avengers**



0.12186

4

3d

**Your Best Entry** ↑

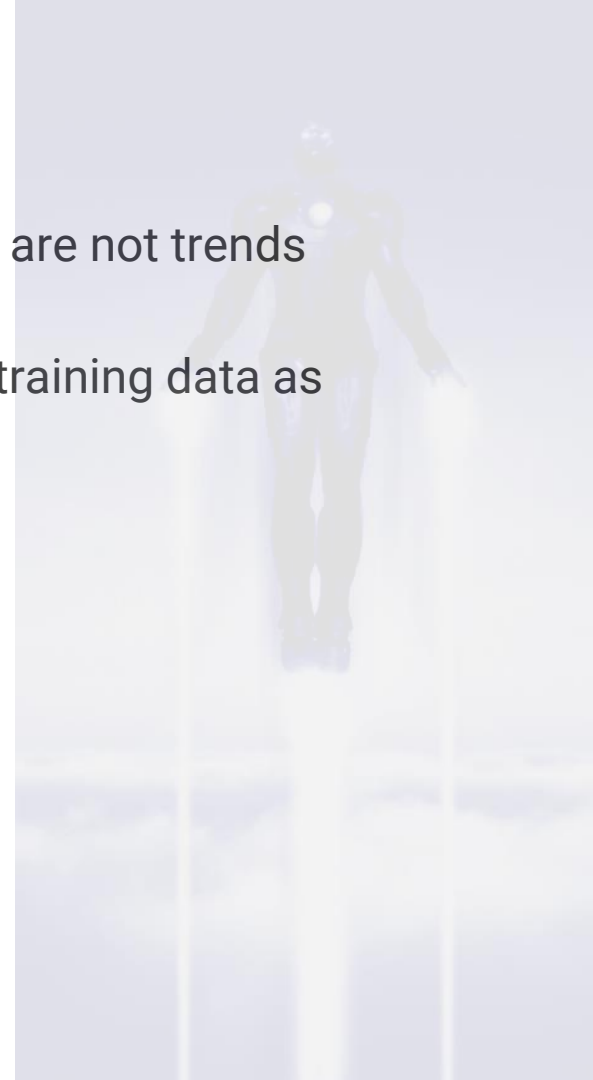
Your submission scored 0.70045, which is not an improvement of your best score. Keep trying!

## 5. THE FUTURE



# Future considerations

- Investigate more into effects of time (are we sure there are not trends in housing price with time? Is this possible?)
- Include an OOB error method (maybe put aside 20% of training data as a test set)
- Focus more on an iterative feature engineering process
- Using GitHub more effectively
- Better team workflow (redundant work)



**Questions?**