Beating the Bubble: Predicting House Prices in Ames, IA

Alex Kim, Jesus Cervantes, Alexandre Bucquet

Stanford University

Motivation

1. **Predict housing prices** via machine learning
2. Explore machine learning **beyond standard techniques**
3. Design and compare automated models that parse data about houses and relate it to their sale price
4. While our project is specific to Ames, IA, we aim to develop techniques that are **generalizable** to other regions as well

Data

- **1,460 house sales** in Ames, Iowa, with detailed data on each house (Kaggle)
- Each entry contains **79 house and sale features** as well as the final sale price for the house (Kaggle)
- **Spatial data** on Ames, IA: includes welfare and demographic figures, as well as information about school proximity and bus lines access for every house in our dataset

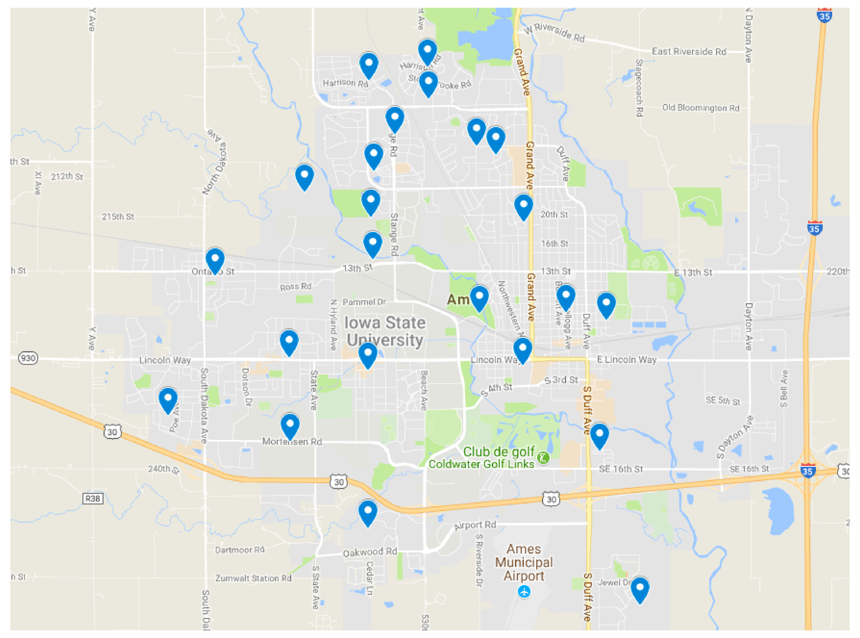

Figure 3: Neighborhoods of Ames, IA

Linear Regression with Gradient Boosting

Linear Regression

- Use **stochastic gradient descent** on the squared loss to find the weight vector w
- Convert non-numeric features into **indicator features**

Gradient Boosting

- **Boosted trees:** secondary linear regression models trained on the error of prior models
- Predict the **error of past prediction(s)**

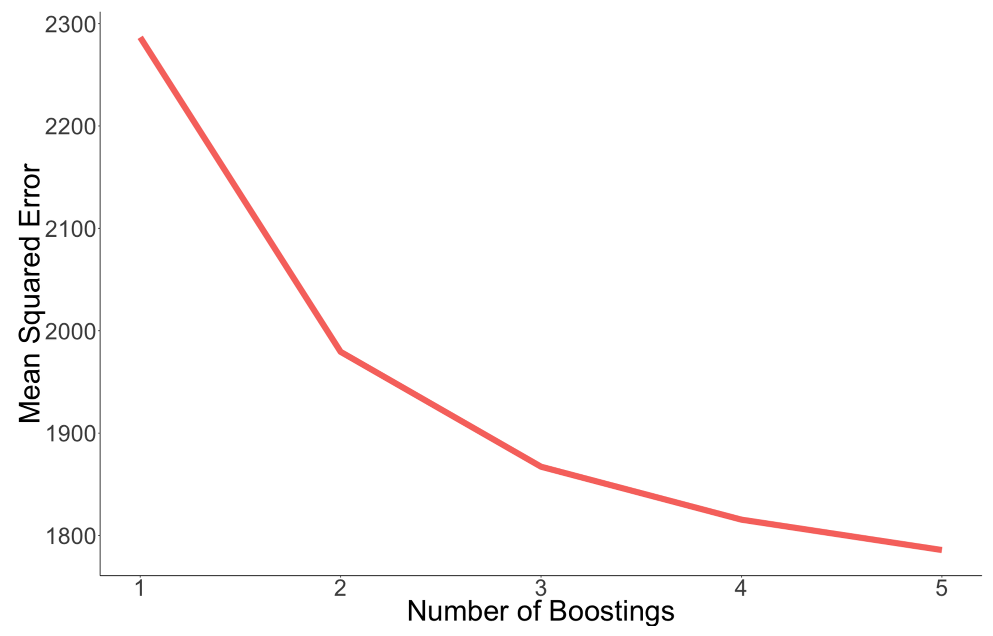

Figure 1: Gradient Boosting Reduces the Test Error

Clustering

Key idea: Group the houses in clusters based on similarity, and run a linear predictor on each cluster of data points.

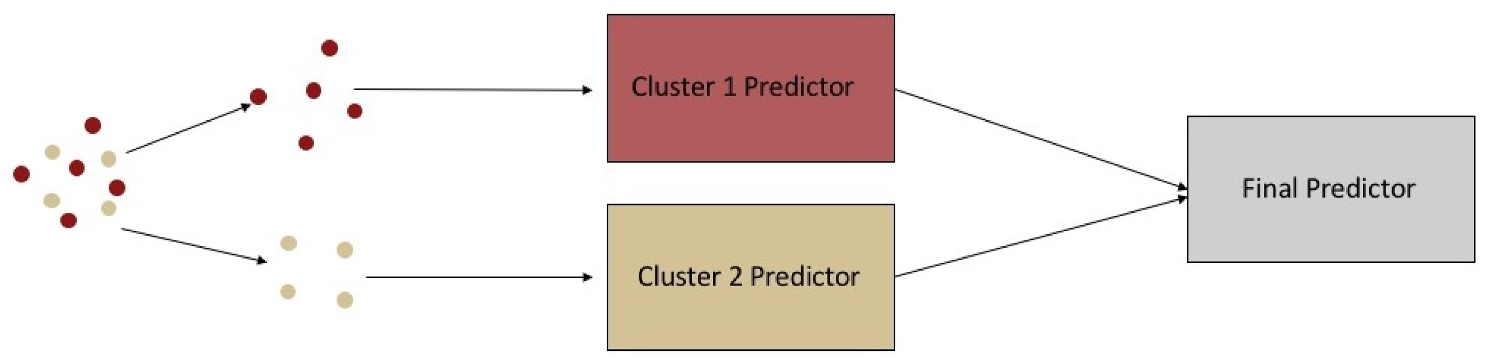

Figure 4: Clustering Method

- **House Clustering:** on each cluster, train a linear predictor using gradient boosting
- **Boosted k-means:** on each cluster, use the centroid as the first layer of predictions then use gradient boosting
- **Neighborhood Clustering:** group the neighborhoods in K clusters, sort the houses in buckets based on neighborhood, and run a predictor on each bucket

House clustering (4)	Boosted k-means (15)	Neighborhood clustering (2)	NC (6)
6758.89	6009.944	2147.58	3661.836

Table 2: Test Error for Clustering Methods

Lasso & Regularization

Key idea: Add penalties to our loss function to reduce overfitting. Two methods:

- **Regularization:** add the L_2 norm of w: $\sum_{i=1}^{n} w_i^2$
- **Lasso:** add the L_1 norm of w: $\sum_{i=1}^{n} |w_i|$

Lasso	Regularization
1498.02	1690.66

Table 1: Test Error After Reducing Overfitting

Discussion

- **Gradient boosting** provides a sharp increase in accuracy compared to the linear regression ($r^2 \approx 0.63$ vs 0.71)
- **Clustering**, although the fastest converging methods, is more prone to overfitting ($r^2 \approx 0.5$)
- **Neighborhood clustering** with 2 clusters gives the best speed/accuracy ratio ($r^2 \approx 0.659$)
- **Feature selection** (with either **lasso** or **regularization**) can further improve predictions by up to 20% ($r^2 \approx 0.75$)

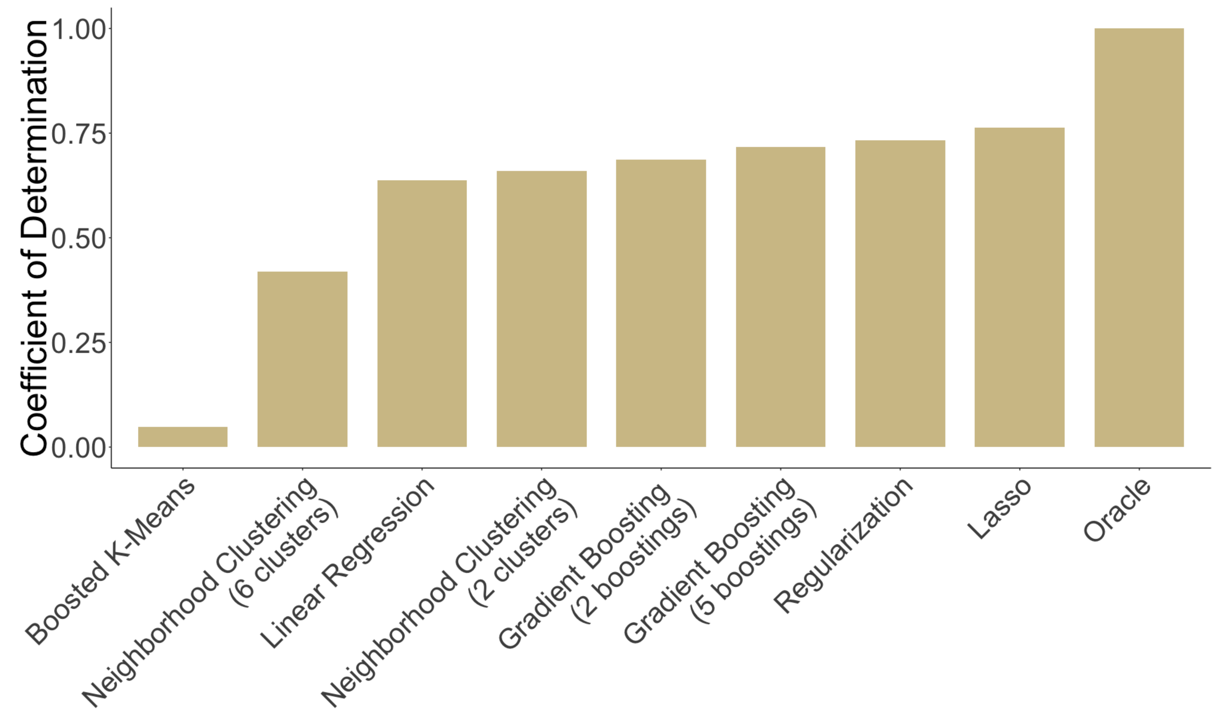

Figure 2: Comparison of r^2 values

Future Research

With more computing resources, we could consider the following next steps:

- Cross-validated **best subset selection** based on the coefficient of determination (r^2)
- Investigate the benefits of adding **more than five trees** on runtime, precision and overfitting
- Account for the relationship between variables by adding **non-linear features** to our model
- **Batch** some of the gradient descent computation to reduce runtime and conduct more experiments
- Incorporate **unique neighborhood features** beyond linear regression

References

[1] Trevor Hastie Gareth James, Daniela Witten and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.

[2] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[3] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 02 2002.

Acknowledgements

We would like to thank the CS221 course staff for their support throughout the quarter, especially our mentor Anna Wang for her help in this project.