IBM - Capstone Project

The Battle of Neighborhoods

Aug 10, 2020

House Price Prediction - Ames, Iowa

Contents

1 Introduction

When attempting to explain house prices, one often considers square footage, number of bedrooms and bathrooms. But not many variables other than a categorical neighborhood variable is introduced to capture price fluctuation explained by location. A data set popularized by Kaggle is the Ames Housing data providing 79 features to predict the price of any given house sold in Ames, Iowa. This data set contains vast information about the house itself, and also a few pieces of information about proximity to main road or railroads.

In particular, this study will break down the neighborhood variable by introducing binary indicators of various venue in the neighborhood to the prediction model. It seeks to answer:

- What is the price tag on proximity to relevant venues in the neighborhood?
- What kind of surrounding venues matter the most when explaining house prices?

2 Data Acquisition and Cleaning

2.1 Collection

The data utilized in this study will be based on Ames Housing data from Kaggle. The Ames Housing dataset was compiled by Dean De Cock for use in data science education. The Ames housing data will be merged with neighborhood data extracted from Foursquare's API. From this API, we will extract information about adjacency to e.g. schools, parks, restaurants, grocery shopping, fitness center, public transport and airport - venues that are potentially affecting the price of the residence. Only the available training data from the Ames Housing dataset is used for this study. The dataset contains the target variable, sales price, for 1460 houses, as well as features such as building year, lot size, utilties, condition of the house, quality of the house, roof style, exterior materials, size of basement, size of garage, and pool size.

- Ames Housing data is derived from *Kaggle*.
- Coordinates for neighborhoods in Ames is gathered using a *Geocoder* library.
- For each neighborhood cordinate, we call the *Foursquare* developer API to get data on

surrounding venue.

The output from this process is a two-dimensional dataframe. Each row represents a sold residence with its 79 features and 12 additional binary indicators of various venues in every neighborhood.

By combining the two data sets, we will be able to answer the research question by building a prediction model that includes these neighborhood features. In doing so, we will put a pricetag on all introduced neighborhood features, which can be used by prospective buyers in Ames during price negotiation. Moreover, using regression techniques, we will obtain the significance levels for each venue and thereby answer which surrounding venues matter in pricing a residence. The results are also relevant for project manager seeking to maximize the value of a given construction project by placing e.g. an apartment building or house in a location with attractive venues nearby.

2.2 Cleaning

For categorical features, e.g. Pool, Alley, Fence, Fireplace, Garage, Basement, missing values are replaced with 'None' to obtain a new category of houses sold without the feature. For numerical features, e.g. *LotFrontage*, we assigned the mean value for the missing values. For the *Electrical* feature, we assign the most frequent category to the missing value. *PoolQC* is mainly *NA*, this feature was converted to a binary class to indicate whether a pool is part of the house or not, since the true cases are so few. For *MasVnrType* and *MasVnrArea* missing values are replaced with zero.

As we need numerical inputs to our machine learning models, we use one-hot-encoding of all categorical features. We drop a category from each feature to avoid perfect multicollinearity. Moreover, we found that *1stFlrSF* + *2ndFlrSF* = *GrLivArea*, we therefore dropped the *1stFlrSF* and *2ndFlrSF* features. Also, *MSSubClass* seems to be a collection of variables (*YearBuilt*, *HouseStyle*, *BldgTyp*), so we dropped that feature as well.

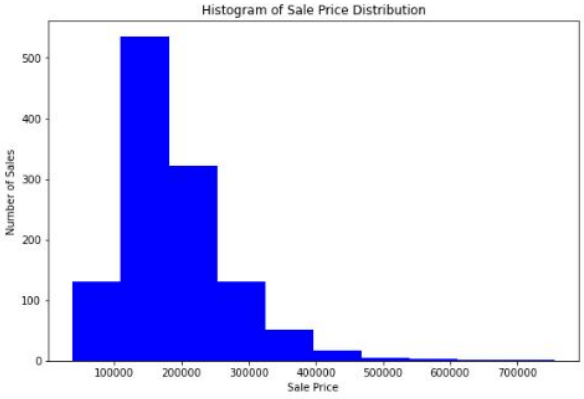
As *Exterior2nd* and *Condition2* are almost identical to *Exterior1st* and *Condition1*. We dropped the second features.

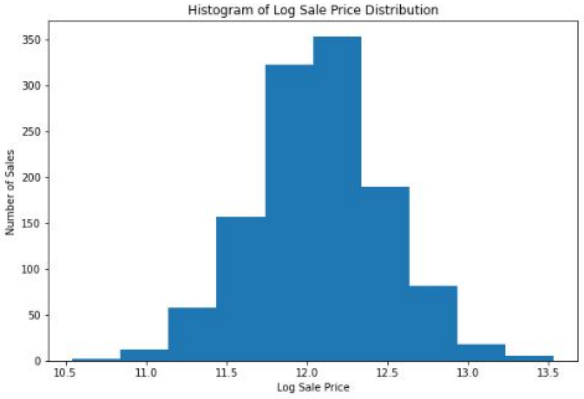
Finally, to avoid other redundant variables, we use the VIF factor, removing the features one by one starting with the largest. As a rule of thumb a VIF factor above ten is alarming. In the final feature set, only *GrLivArea*, *age* and *school* have VIF factors of around 10-12. Variables removed were: *PoolArea*, *Exterior1st_CBlock*, *Exterior1st_VinylSd*, *SaleType_New*, *SaleCondition_Partial*, *GarageType_BuiltIn*, *Electrical_FuseA*, *Heating_GasA*, *ExterQual_TA*, *ExterCond_TA*, *Condition1_Norm*, *RoofStyle_Gable*, *RoofMatl_CompShg*, *Functional_Typ*, *HouseStyle_2Story*, *Exterior1st_ImStucc*, *Exterior1st_AsphShn*, *ExterCond_Po*, *Exterior1st_BrkComm*.

2.3 Feature engineering

2.3.1 Target variable

As the sales price is highly right skewed, we take the natural logarithm to obtain a more normally distributed target.





2.3.2 Categorical variables

All features with a ranking system were converted to a numerical feature. e.g. ranking variables ('Ex', 'Gd', 'TA', 'Fa', 'None', 'Po'). This transformation was applied to e.g. *ExterQual*, *ExterCond*, *BsmtQual*, *HeatingQC*, *KitchenQual*, *FireplaceQu*, *GarageQual*.

Variables such as *GarageType* with categories as *2Types*, *Attchd*, *Basment* etc. were converted to dummy variables using pandas get_dummies() function. As this function introduces perfect multicollinearity, we remove one dummy feature from each catogorical variable.

2.3.3 House age

The age of the house could certainly be argued to have a great impact on the prices. *YrSold* and *YrBuilt* was be used to generate an *age* feature.

$$YrSold - YrBuilt = age$$

3 Methodology

3.1 Foursquare API features

From the Foursquare API we got information about venues in the proximity of each neighborhood. From the latitude and longitude of each neighborhood, we find venues within a radius of one kilometer. This method is the best approximation, as the address of each house is not given. One should note that this method is liable to the fact that some houses may be sold far away from the neighborhood lat and lng coordinates, and the venue proximity will therefore not be completely accurate.

These venues are categorized based on buzzwords in the venue title:

- **stores** Buzzwords: stores, boutiques and shop.
- **nightlife** Buzzwords: lounge, nightclub, pub, bar, wine, beer, brewery, and bodega.
- **grocery** Buzzwords: grocery, farmers market, convenience, bakery.
- **service** Buzzwords: salon, pharma, bank, atm, service, agency and landscaping.
- **restaurants** Buzzwords: taco, restaurant, food, diner, coffee, café, bistro, breakfast spot, burrito, and sandwich.
- **transport** Buzzwords: bus and ferry.
- **school** Buzzwords: college.
- **golf** Buzzwords: golf.
- **park** Buzzwords: park, trail, and tree.

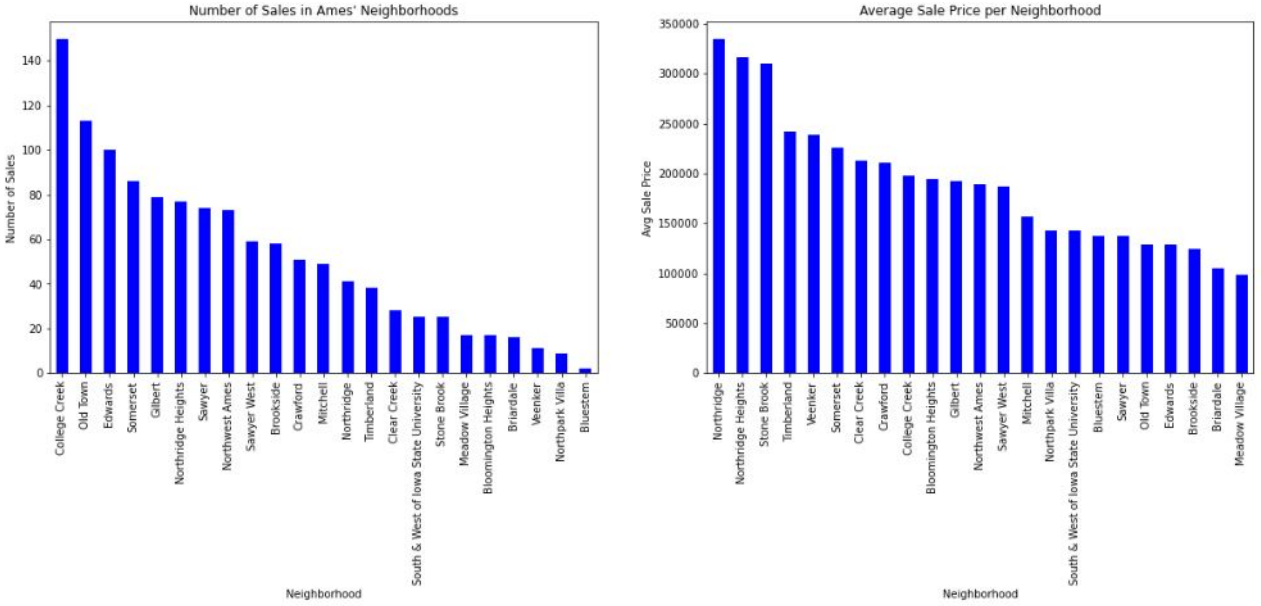
- **recreational** Buzzwords: gym, ice skating, yoga, disc golf, basketball court, baseball field, and arcade.
- **cultural** Buzzwords: Historic site, museum, music venue, rock club, and stadium.
- **leisure** Buzzwords: spa, bed breakfast, and hotel.

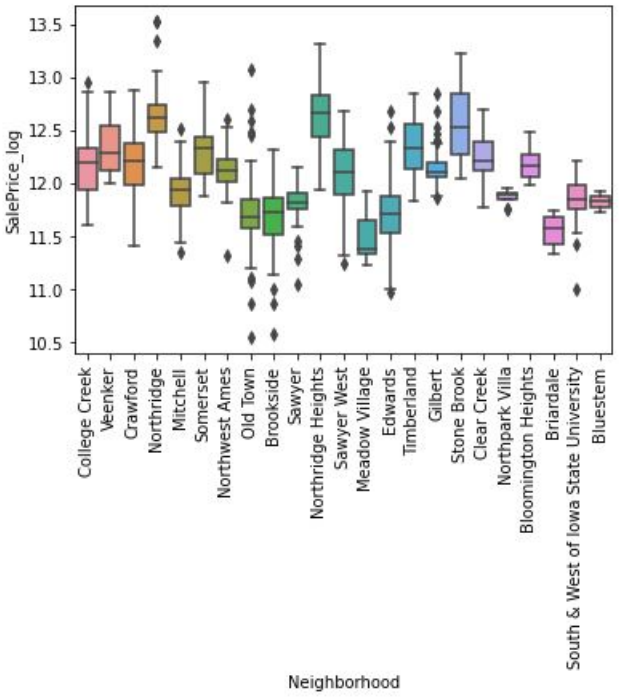
For each category we rank each neighborhood depending on the amount of venues in the proximity of the neighborhood. A rank of 1 means that the neighborhood is the neighborhood with the fewest amount of venues. If some neighborhoods have the same amount of venues, they both get the same rank. Ranks are incremented by one. One could argue that the relationship between sales and venue ranks is not linear. Just because a neighborhood has 18 restaurants rather than 17, it does not necessarily constitute an increment in house price premium or discount. Another approach could be to simply construct binary variables for each venue, but this approach does not capture a difference between a neighborhood with one restaurant vs a neighborhood with 20.

	Neighborhood	leisure	cultural	recreational	service	grocery	park	golf	nightlife	school	transport	stores	restaurants
0	Bloomington Heights	1	1	3	2	1	3	1	2	1	1	8	6
1	Bluestem	1	2	2	1	1	2	2	2	3	2	1	1
2	Briardale	1	1	3	1	1	4	2	1	1	1	1	1
3	Brookside	2	3	1	4	5	4	1	7	1	2	9	9
4	Clear Creek	1	1	2	1	1	2	2	4	2	1	1	3
5	College Creek	1	3	2	2	2	1	1	8	5	2	6	8
6	Crawford	2	3	2	3	4	4	1	5	1	1	7	7
7	Edwards	2	1	4	2	1	2	1	1	1	1	5	4
8	Gilbert	1	3	2	1	1	2	1	2	1	1	2	1
9	Meadow Village	1	2	3	2	2	1	2	6	4	2	6	8
10	Mitchell	1	1	1	2	1	1	1	1	1	1	3	1
11	North Ames	2	3	1	4	5	4	1	7	1	2	9	9
12	Northpark Villa	1	1	1	1	1	1	1	3	1	1	1	1
13	Northridge	2	1	3	1	2	2	1	1	1	1	2	2
14	Northridge Heights	2	1	3	1	2	1	1	1	1	1	2	2
15	Northwest Ames	2	3	1	4	5	4	1	7	1	2	9	9
16	Old Town	1	4	1	2	3	2	1	3	1	1	4	4
17	Sawyer	1	1	1	1	1	4	1	2	1	2	1	2
18	Sawyer West	1	1	1	1	1	4	1	2	1	2	1	2
19	Somerset	1	1	3	2	2	2	1	2	1	1	5	5
20	South & West of Iowa State University	1	1	1	1	1	1	1	1	1	1	1	1
21	Stone Brook	1	1	1	2	1	4	1	1	1	1	1	2
22	Timberland	3	1	1	1	1	1	1	1	1	1	1	2
23	Veenker	1	1	2	1	1	2	3	1	1	1	1	2

3.2 Neighborhood pricing

In the barplot below we observe that College Creek has the most number of sales (> 140), while Bluestem has almost none (< 5). We also observe that three neighborhoods stands out on average pricing. Northridge, Northridge Hights and Stone Brook all have an average sales price of more than \$60,000 higher than the other neighborhoods. Meadow village is found to be the least expensive neighborhood to live in on average. This also stands out in the box plot.

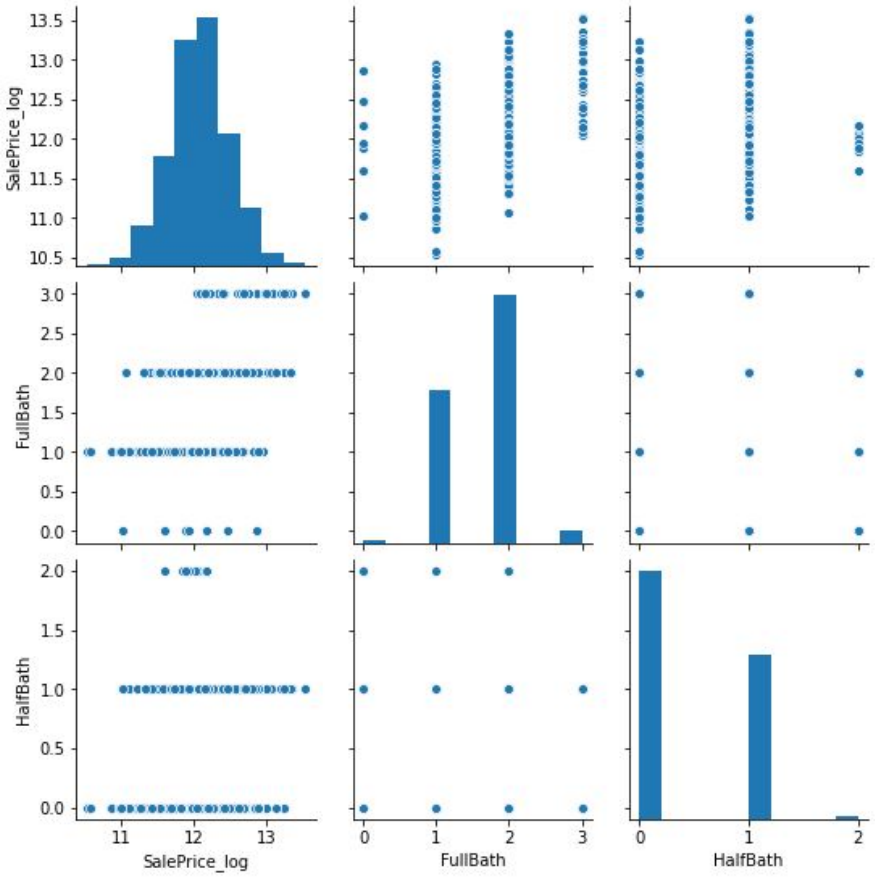




3.3 Feature selection

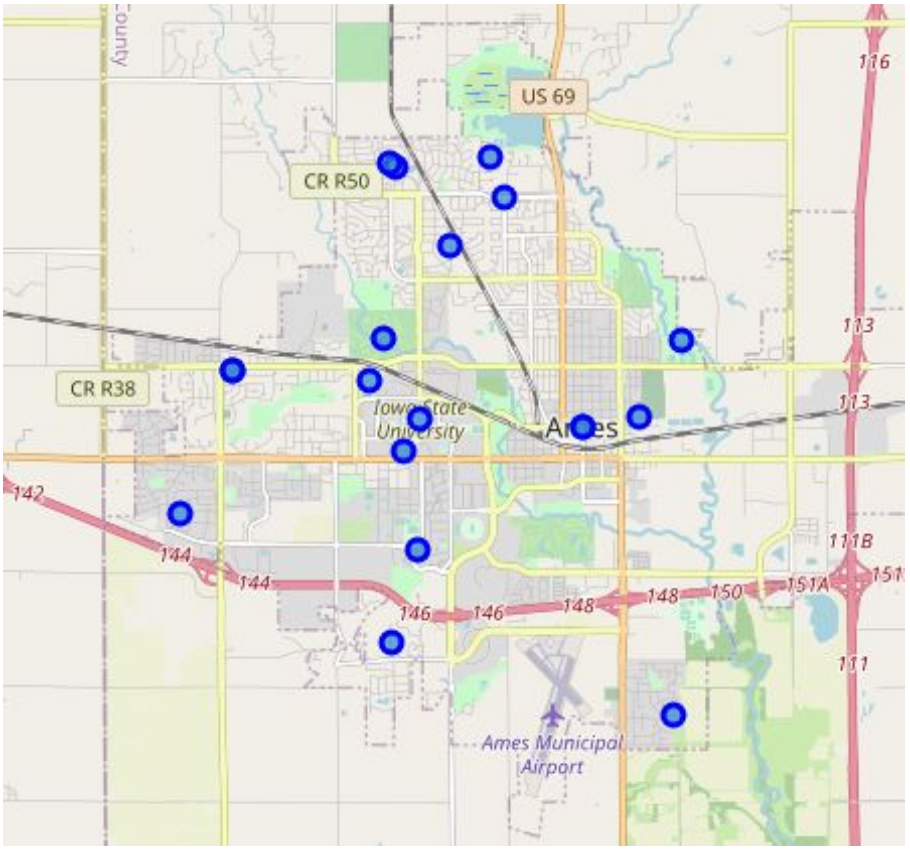
We will leave the walkthrough of the full feature selection process to the Jupyter Notebook called *CapstoneProject - Feature selection and EDA*. But to give an example of the method. We split our feature analysis into categories depending on what part of the house the features tell us about. E.g. the Bath variables. From the correlation heatmap, we observe that *FullBath* has the greatest correlation with *SalePrice_log*. From the scatter plot below it is evident that *FullBath* display some positive linear relationship with the sales price, while the HalfBath does not show any clear pattern with limited observations in the *2* category. We therefore keep the *FullBath* feature while dropping the *HalfBath* feature.

	SalePrice_log	FullBath	HalfBath
SalePrice_log	1.000000	0.591848	0.290486
FullBath	0.591848	1.000000	0.116346
HalfBath	0.290486	0.116346	1.000000



3.4 Folium map

From the folium map, we find that Northridge and Northridge Highs are located next to the CR R50 label. Stonebrook is the marker just below US 69. Meadow Village in the middle of Iowa State University with College Creek just below. Michell in the bottom right next to the airport. Timberland on the left side of the airport. Edwards furthest to the left. The expensive areas are therefore located in the northern part of Ames.



3.5 Model selection

As we are working with a continuous target variable, regression models are appropriate.

Linear regression is naturally included. But as we have a large amount of potentially still redundant features, Lasso and Elastic net regressions are also introduced.

We also include Random Forest regression. Given the many binary features in the data set, a tree-based model makes a lot of sense. As it uses a random sample of the data, we are

less likely to overfit on the training data.

Moreover, a Support Vector Regression (SVR) is introduced. The benefits of this model is that SVR acknowledges the presence of non-linearity in the data set. This model could therefore provide a more proficient prediction model.

Finally, a Gradient Boosting model is selected for testing. As it build trees one at a time and correct errors made on previously trained trees, this model could potentially provide a similar and potentially better model than the random forest.

First we standardize the feature set using SkLearn's preprocessing library's StandardScaler function. From a spot check using cross validation on the train set using each of the six algorithms, we obtain the following negative mean absolute errors (Neg MAE)

Model	Negative mean absolute error	Stdev
Linear Regression	-3.626.332.352	10.878.997.056
Lasso Regression	-0.318790	0.0176
Elastic Net	-0.318790	0.0176
Random Forest	-0.105817	0.0064
Support Vector	-0.127016	0.0148
Gradient Boosting	-0.096784	0.0053

The linear regression errors are most likely a result of a large amount of features compared to limited data, or simply breaches of GM-assumptions (e.g. correlated features).

Random Forest, Support Vectors and Gradient Boosting performs better, and model tuning will be performed on those.

The model tuning is performed using SkLearns GridSearch function.

- **Random Forest:** Number of estimators tuned to 280. Max depth: tuned to 25.
- **Support Vector Regression (SVR):**. The regularization parameter, C, is tuned to 0.5. The kernel to 'linear'.
- **Gradient Boosting:** Learning rate tuned to 0.1. Number of estimators tuned to 180. Loss function tuned to 'huber'.

4 Results

4.1 Linear Regression Price Tags

From a linear regression of the full model, we find the following coefficients and test statistics subset for the Foursquare API features:

	coef	std err	t	P>|t|	[0.025	0.975]
restaurants	-0.0125	0.011	-1.094	0.274	-0.035	0.010
transport	-0.0462	0.011	-4.022	0.000	-0.069	-0.024
school	0.0567	0.014	4.104	0.000	0.030	0.084
golf	-0.0002	0.006	-0.027	0.979	-0.012	0.012
park	0.0444	0.010	4.491	0.000	0.025	0.064
recreational	-0.0289	0.008	-3.828	0.000	-0.044	-0.014
cultural	-0.0137	0.010	-1.408	0.159	-0.033	0.005
leisure	0.0288	0.008	3.702	0.000	0.014	0.044

Ames house buyers are willing to pay extra to live near universities (This is most likely an effect of College Creek being the 7th most expensive neighborhood and the location of Iowa State University). Recreational venues are concentrated in cheap city center neighborhoods (Meadow Vlg and Edwards), which explain the negative sign. Transportation options are also greater in the cheaper city center neighborhoods. In summary, it would seem to indicate that Ames housing is priced with a higher price tag on houses in the outskirts of the city center, and buyers put a larger price on houses in the vicinity of more schools, parks and leisure venues.

4.2 Test ensample

After tuning the prediction models we test the prediction performance on the test set. We note from the Jupyter notebook, that the tuned models all perform slightly better than the un-tuned models in terms of cross validation scores.

We obtain the following train and test Neg MAE, after taking the exponential of the

predictions and SalePrice_log to get an interpretable result:

	Train Neg MAE	Test Neg MAE
Random Forest	-7581.4	-17262.531536
Support Vector	-17709.8	-15930.752618
Gradient Boosting	-9391.56	-15504.138400
Combined		-14335.539862

From the output above, we observe that the random forest. The gradient boosting model performs better than random forest and SVR.

In this scenario we have a case where $1 + 1 = 3$ as an average of all three models combined outperforms the individual prediction model. Taking the average results in an average error of $14.335 on house prices in Ames.

We observe some degree of overfitting for Random Forest and Gradient Boosting, as the test error is significantly higher than the train error. Surprisingly, for SVR, we see the train error > generalization error. This could indicate that it is an unknown fit, or that the test set contained easy cases to price, and the train set contained harder cases.

4.3 Feature importance

From a random forest feature importance output, we observe a clear indication that overall quality and above ground living area were key in explaining house prices.

The first Foursquare feature was restaurants with a VI of 0.00689 in the 25th rank.

From these results, it is evident that the neighborhood venues features does not add much in explaining the house prices. They only serve as indicators of a (small) positive or negative impact by proximity to the specific venues.

5 Discussion

As mentioned earlier, the Foursquare API takes lat and lng coordinates to provide venues for each neighborhood. As we do not have any information about specific locations of each observation, the neighborhood coordinates may be far off. Therefore, the venues associated with each observation may in reality be different.

Moreover, the ranking system is, as stated earlier, flawed. As one could argue that the relationship between sales and venue ranks is not linear. At this moment, the only other approach attempted was a binary variable for each venue. But it did not improve the fit, or change the conclusions.

While one should expect to perform better on the train set than on the testing set, one could attempt to alleviate this issue of overfitting on the gradient boosting and random forest models. An average error of $14,335$ is not a catastrophic error for houses selling at an average of $190,003$, but some recommendations could be to improve tuning, introduce other prediction models, and take a deeper dive into the feature engineering section of the project.

6 Conclusion

We sought to answer: What is the price tag on each relevant neighborhood feature? and: What kind of surrounding venues matter the most?

Based on a linear regression we found that house buyers are paying a statistically significant premium on proximity to parks, leisure and school venues, while significantly discounting proximity to transportation and recreational venues. The cheapest neighborhood on average, Meadow Village, is also a neighborhood in central Ames, meaning lots of transportation options and recreational venues – as well as restaurants, which also appear with a negative (insignificant) coefficient.

The model does not improve the fit from a model based on a simple neighborhood category feature. This could mainly be due to the degrees of freedom being different. However, the model does tell a story about how certain venues make the neighborhoods differ in price. The results

are therefore still useful to e.g. potential house buyers in Ames, as the model can give a fair estimate of the price of any house given a list of the feature inputs with an average error of 14 thousand dollars.