Housing Price Predictions

W207 Final Project

Lee Moore and Paul Petit

Background

Objective: Predict housing prices in Ames, Iowa

Dataset: The sales of individual residential property in Ames, Iowa

- 2930 observations collected from 2006 to 2010
- 79 feature variables to assess home values
- (23 nominal, 23 ordinal, 14 discrete, and 19 continuous)

Exploratory Analysis

- Training set reduced to 1460 records
- Further categorized 79 features into sets to view them more intuitively
  - Eg. Location features (ie. neighborhood, zoning), lot features (ie. area, alley), etc.
  - Eight features sets altogether
- Data were largely clean; removed 4 records of partial, odd sales and replaced missing categorical data with “NA” and numeric data with 0s
  - Year of garage build tricky → replaced with median year
- Ran correlation matrix of features with sale price to inform feature engineering

Feature Engineering

Goal: improve key model predictions by manipulating train data set

Three categories of manipulation:

1. Additional features derived from existing ones (eg. livable square feet per lot)
2. Transformed features (eg. categorical to ordinal, binarizing, log)
3. Removing feature sets

Testing method:

1. Test impact of individual feature transformation on key model score
2. Chain features that improve score together

Feature Engineering Learnings

1. Greatest model performance improvements result of augmenting training data set with derived features
2. Transforming applicable features from categorical to ordinal did have an impact on model performance
3. Removing variables that were strongly correlated with each other did impact the score, but removing variables weakly correlated with outcome did not
4. Removing feature sets almost categorically worsened results; the only exception was removing “sale features” (eg. year sold, sale type, etc.)
5. Chaining individual manipulations that improved model performance didn’t result in iteratively improved performance

Modelling Approach

Metric: Root Mean Squared Error (RMSE)

Testing approach: 10-Fold Cross-Validation

- Rationale:
    - Only 1460 observations with labels (another 1459 without labels)
    - With 300+ features, better to use K-fold cross-validation instead of a static train & dev set.
    - Concern: Is this time series data? If so, can't use K-fold CV...checked test data and covered same 2006-2010 time period so a non-issue

Baseline model: Median House Price Prediction: RMSE : 0.40 (0.03)

All studied regression model types evaluated
(with exception of Neural Nets)

1. KNN Regressor
2. Linear Regression
    a. L2 Regression applied using Ridge Regression
3. Decision Trees (with Stacking/Boosting etc)
    a. Decision Tree Regressor
    b. Random Forest Regressor
    c. XGB Regressor
4. Support Vector Machine Regressors
    a. SVR
    b. LinearSVR (once confirmed best SVR kernel = 'linear')

Modeling testing procedure

1. Fit using two different training sets:
   a. Fully feature-engineered training data
   b. Reduced dataset to 30 features using PCA for dimensionality reduction (0.99999991 explained)
2. GridSearchCV over researched range of potential values
   a. Hyperparameter tuning only possible with reduced dataset for more complex models (i.e. SVM and ensemble methods)
3. Blended model predictions (70% ridge + 30% XGB)

	Model	Mean Score	Std Dev
0	Baseline	0.3957	0.0263
1	KNN	0.1985	0.0118
2	Linear Reg	0.1317	0.0109
3	Ridge Reg	0.1139	0.0137
4	Decision Tree	0.2272	0.0152
5	Random Forest	0.1344	0.0143
6	XGB Regressor	0.1162	0.0140
7	SVM Reg	0.1267	0.0152

Best Kaggle Result (out of 4,234 teams)

Top 17%!

716	zhaomoing	0.11634	13	16d
717	dasc1x	0.11635	23	5d
718	Alex Tan	0.11636	5	16d
719	Lee Moore	0.11636	6	now

Your Best Entry ↑

You advanced 76 places on the leaderboard!

Your submission scored 0.11636, which is an improvement of your previous score of 0.11706. Great job!

Tweet this!

720	Casey Zhang	0.11636	25	23d
721	iamsaurabhc	0.11636	5	23d
722	Andrei Brasoveanu	0.11637	38	10d

Modelling lessons learned

1. The hyperparameter space is enormous, difficult to optimize.
    a. Sequencing of feature engineering v. choice of model + fit
    b. Even scalar best option (StandardScaler, RobustScaler, MaxMinScaler) differed by model
2. Even ML on a ‘small’ dataset can suffer from speed/performance issues
    a. Our 1500 x 300 dataset struggled with some model choices
    b. Finding optimised versions (e.g. LinearSVR v. SVR, LightGBM v. XGB Regressor, and using Gridsearch on PCA reduced feature sets) are helpful to get things moving
3. Tree-based models not always necessarily the best
    a. linear regression was more accurate and considerably faster to fit and predict
4. Blending will improve generalizability almost without fail
5. Kaggle competitions make it fun!