A Quality Score Based Algorithm for House Price Prediction

Yi Wang

Mar. 12. 2021

Code: https://github.com/ywang110/HouseAssessment

Outline

- Motivation
- Challenging
- Problem Formulation
  - Quality score estimation
  - Non-quality score estimation
- Data Preprocessing
- Feature Engineering
- Evaluation Data & Metrics
- Performance
- More Applications
  - Lasso Coefficients
  - School Ranks
  - Recommend a Market deal
- Code Structure

Motivation

- Problem Statement
  - Predict the market price of a house given historical sold prices and features.

- Expected deliverables
  - Suggested listing price to guide the online house seller
  - *suggested, low and high* range of a listing price
  - a *real time* online pricing algorithm

- Business impact
  - Improve customer retention rate, indirectly or indirectly boost business revenue
  - Able to use it for either individual e-market seller (opendoor, zillow, redfin) or a business real estate investment

Challenging

- House price is sensitive to the real life market value (time)
- Historical data can not fully represent the current market
- Most data are categorically based, features spaces could be large
- Model complexity & interpretation

Problem Formulation

- A house price is formulated as

$$Price^{(style)}(t) = \beta^T features + \gamma_{zipcode}(t) + \gamma_{schoolcode}(t) + \alpha_{intercept}(t) + \epsilon$$

Quality score

Non-quality score

Flutuation

  - $Price^{style}(t)$: train one model **per style** (e.g. townhouse, single family, condo, etc)

  - $\beta^T features$: **quality score** of a house
    - Not sensitive to the market change, e.g.
      - #bedrooms, #bathrooms, size, lot area, house condition, stories, years built
    - Include the seasonality feature, e.g. sell month, sell quarter

Problem Formulation

- A house price is formulated as

$$Price^{(style)}(t) = \beta^T features + \gamma_{zipcode}(t) + \gamma_{schoolcode}(t) + \alpha_{intercept}(t) + \epsilon$$

Quality score

Non-quality score

Flutuation

  - $\gamma_{zipcode}(t) + \gamma_{schoolcode}(t) + \alpha_{intercept}(t)$: **Non-quality score** (**market score**) of a house
    - Sensitive to the market change t, e.g.
      - market in a particular zip code, market in a particular school zone
    - Include the intercept $\alpha_{intercept}(t)$
  - ϵ: individual fluktuation
    - Follows a Gaussian $N(\mu, \sigma^2)$ distribution
    - Indicate a house is either overpriced or underpriced comparing to the average

Quality score estimation

$$Price^{(style)}(t) = \beta^T features + \gamma_{zipcode}(t) + \gamma_{schoolcode}(t) + \alpha_{intercept}(t) + \epsilon$$

- $\beta^T features$:
  - quality score coefficients β are **trained offline**
    - The model assumes the quality of a house does not change much in training period
  - *features* used in model
    - House features:

      #bedrooms, #bathrooms, size, lot area, house condition (good, bad, …), stories, years built, roof materials, grade (A, B, C, …), condition (good, bad, …), CDU, basement, garage, fireplace, years to sell
    - Seasonality features: sale month, sale quarter

Non-quality score estimation

- Let non-quality score $\gamma(t) = \gamma_{zipcode}(t) + \gamma_{schoolcode}(t) + \alpha_{intercept}(t)$, we **online estimate** non-quality score coefficients $\hat{\gamma}(t)$ using ***KNN***

1. Choose the k most recent sold houses on the market (k=10) for the same ***style, zipcode, schoolcode***.
2. For each sold house i=*1 to* k, compute its non-quality score by

$$\gamma_i(t) = PriceSold_i^{(style)}(t) - \beta^T features_i$$

3. Rank and removal the first and last 10% percentile of $\gamma(t)_i$, so we have k-2 left.
4. Calculate the estimated non-quality scores as

$$\hat{\gamma}_{suggest}(t) = median(\gamma(t)_i), i = 1,2,..,k-2$$
$$\hat{\gamma}_{low}(t) = max(\gamma(t)_i), i = 1,2,..,k-2$$
$$\hat{\gamma}_{high}(t) = min(\gamma(t)_i), i = 1,2,..,k-2$$

5. Use three scores $\hat{\gamma}_{suggest}(t), \hat{\gamma}_{lower}(t), \hat{\gamma}_{high}(t)$ to compute **suggested**, **low** and **high** price range.

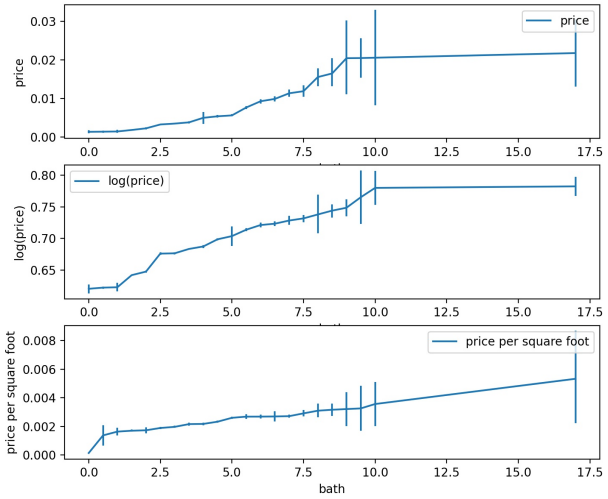
Data Preprocessing

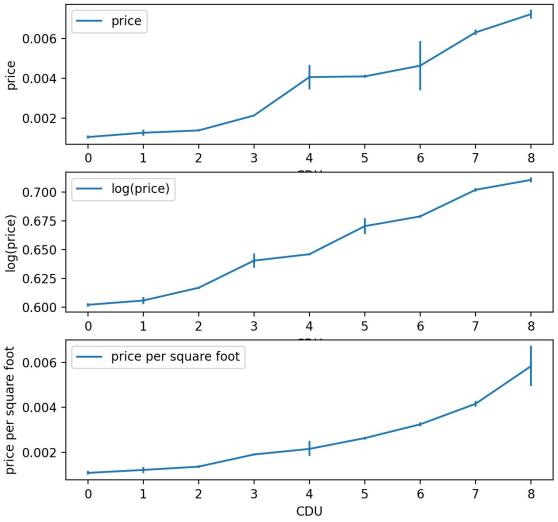
- Categorical data level reduction
  - Example Categorical data levels:
    - “Grade”: 22 categories
    - “CDU”: 9 categories
    - “Roof”: 10 categories
    - “Sale Month”: 12 categories
  - Covert some categorical data into numeric data (redo encoding)
    - Reduce the feature dimension, model variance (overfitting)

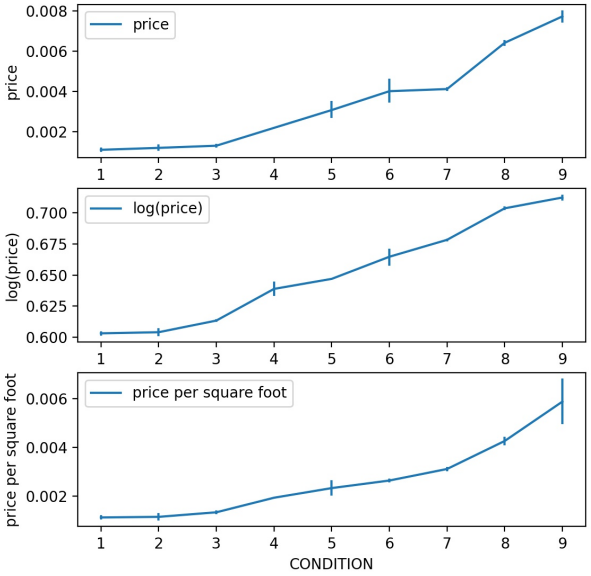
- Noise removal, e.g.
  - Remove the first and last 5% percentile of price for each style
  - Remove houses with sold price < $20k
  - Fill missing data

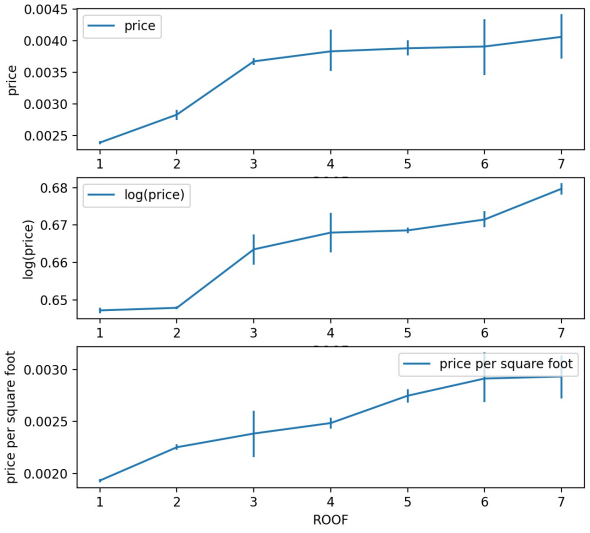
- Log transformation
  - An unit increase of some features result in proportional increase of price

- Response choose: model price per square feet, instead of price or log(price)

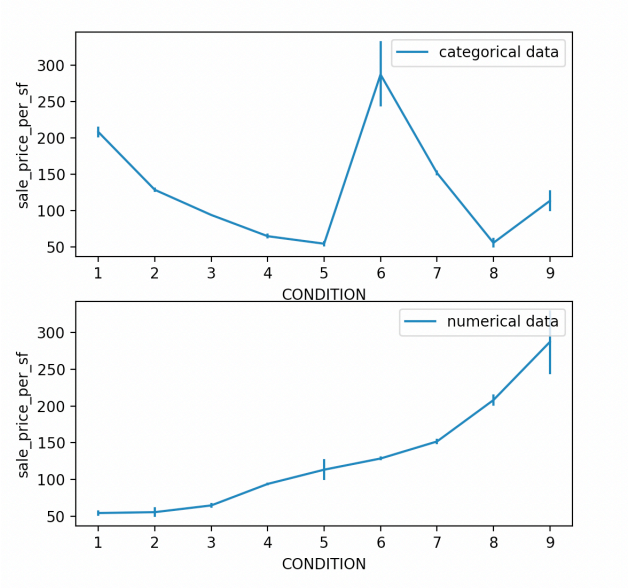


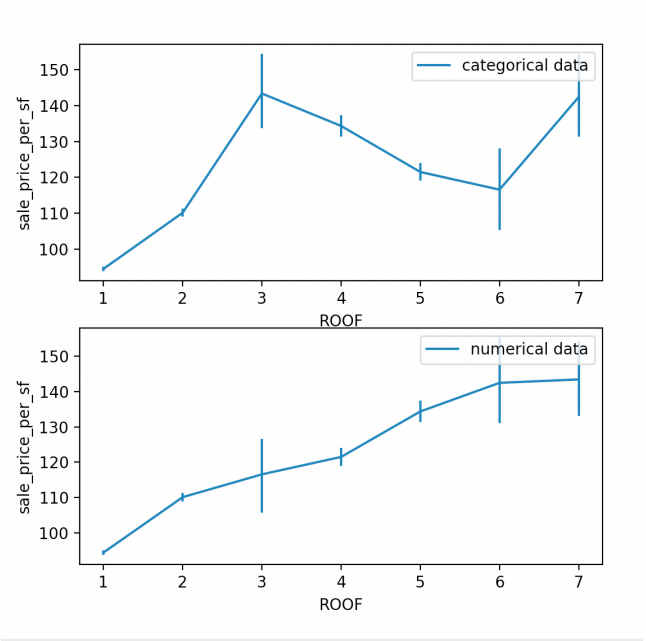


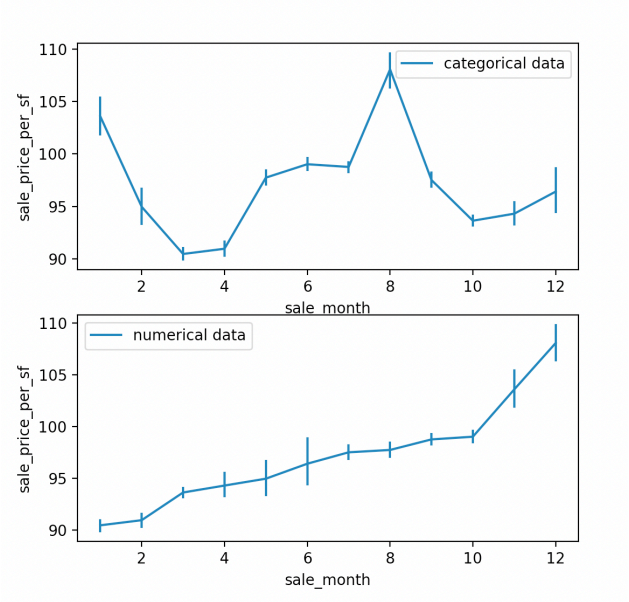


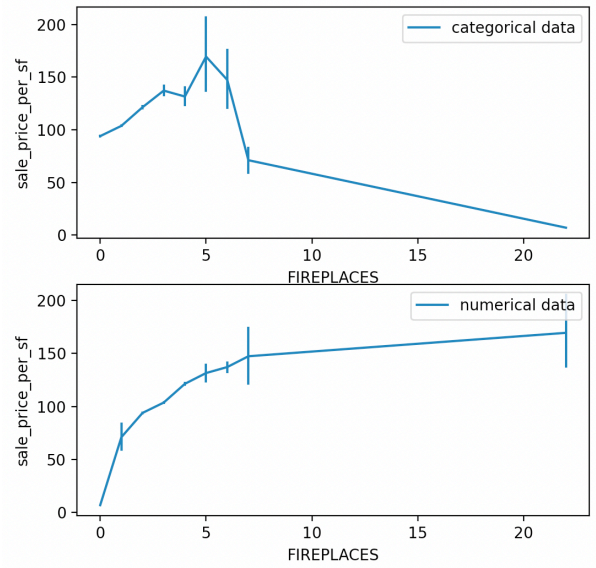


- Convert categorical data to numerical data and redo encoding after sorting









Feature Engineering

- Feature selection is built in our 3 regression models
  - Lasso regression
  - Random Forest regression
  - Xgbooting tree regression

- Dummy variable are created for our non-quality score features
  - $\gamma_{zipcode}(t)$
  - $\gamma_{schoolcode}(t)$

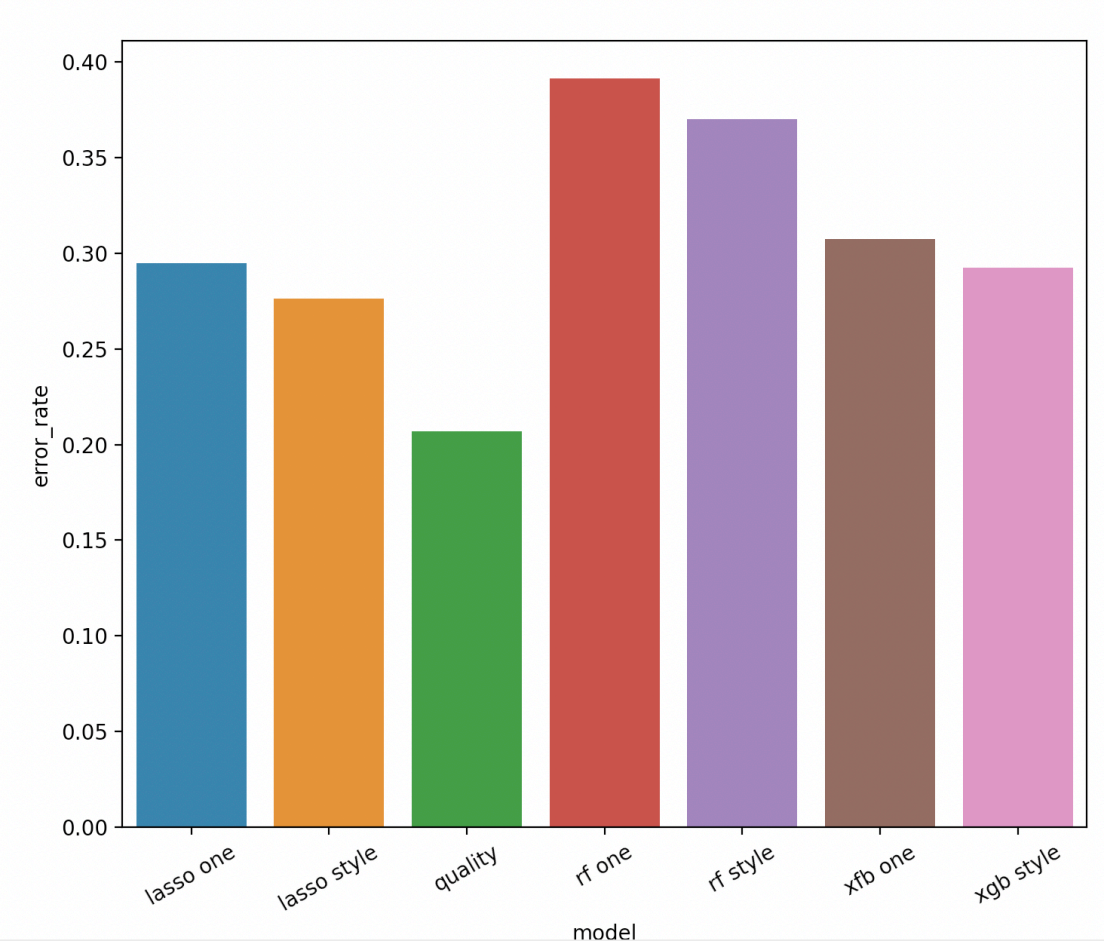
Evaluation Data & Metrics

- Evaluation Data
  - 20 years data, 340532 properties, 29 styles, 127 zip codes, 46 school codes
  - Training year: 2000-2015
    - 234365 properties
  - Testing year: 2016 - 2021 (recent 5 years)
    - 106167 properties

- Evaluation Metrics
  - median absolute error rate

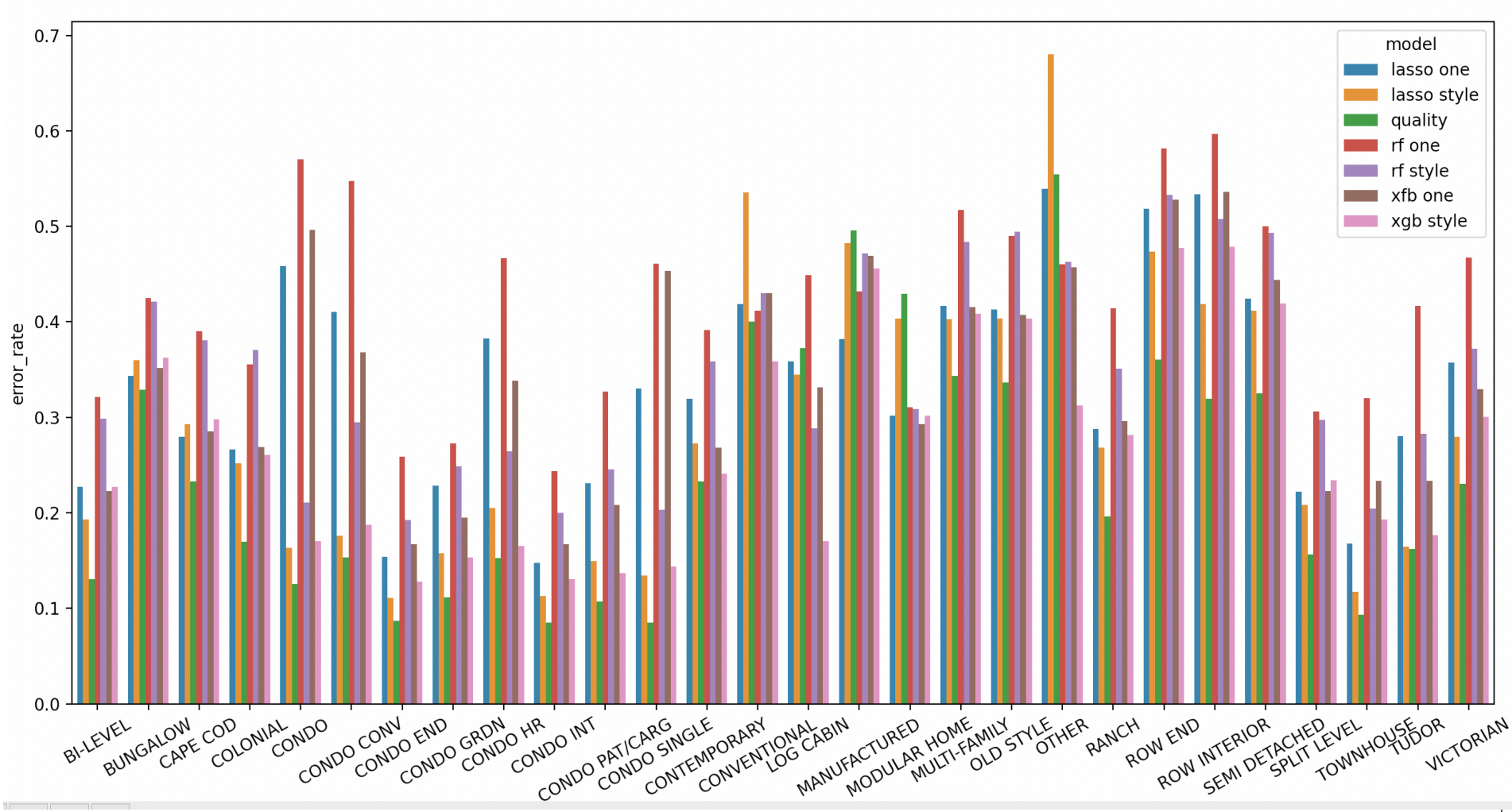
$$median(\frac{|Price^{(style)}(t) - \hat{Price}^{(style)}(t)|}{Price^{(style)}(t)})$$

Performance



- "quality": quality model
- "lasso one": a single lasso model
- "lasso style": one lasso model per style
- "rf one": a single random forest model
- "rf style": one random forest model per style
- "xgb one": a single xgb model

Performance



Lasso Model's Interpretation

- Positively contribute to price per square feet:
  - Grade, Condition, CDU, Fireplaces, Garage, Sale month, Sale quarter
- Negatively contribute to price per square feet:
  - Stories, Living Area, Years built, Years to sell
- Features not selected:
  - Roof material, Basement, Bedrooms, Lot Area, Bath

Selected Feature	Lasso Coeff
Stories	-1.304
Grade	3.232
Condition	0.000659
CDU	4.095
Fireplaces	5.695
Garage	3.829
Living Area	-0.01067
Years built	-0.07906
Sale month	0.08816
Years to sell	-1.2994
Sale quarter	3.57002

School Ranks

- Use the model school quality score $\hat{\gamma}_{schoolcode}(t)$ to buy a house?

rank	School ID by Model Score	School ID by Price per Sqft	Model Score	Price per Sqft
0	3	26	16.901005	112.27125
1	26	17	10.705427	111.20941
2	32	27	6.849829	110.05766
3	27	28	6.565823	109.65551
4	17	8	4.1878069	109.19439
5	42	32	3.9063721	107.5653
6	2	20	3.3590302	103.5131
7	20	5	2.9442477	101.81237
8	5	24	2.7791844	95.510712
9	24	3	2.5594571	95.204031
10	8	43	0.4851836	95.044302
11	25	34	0.2660076	94.745909

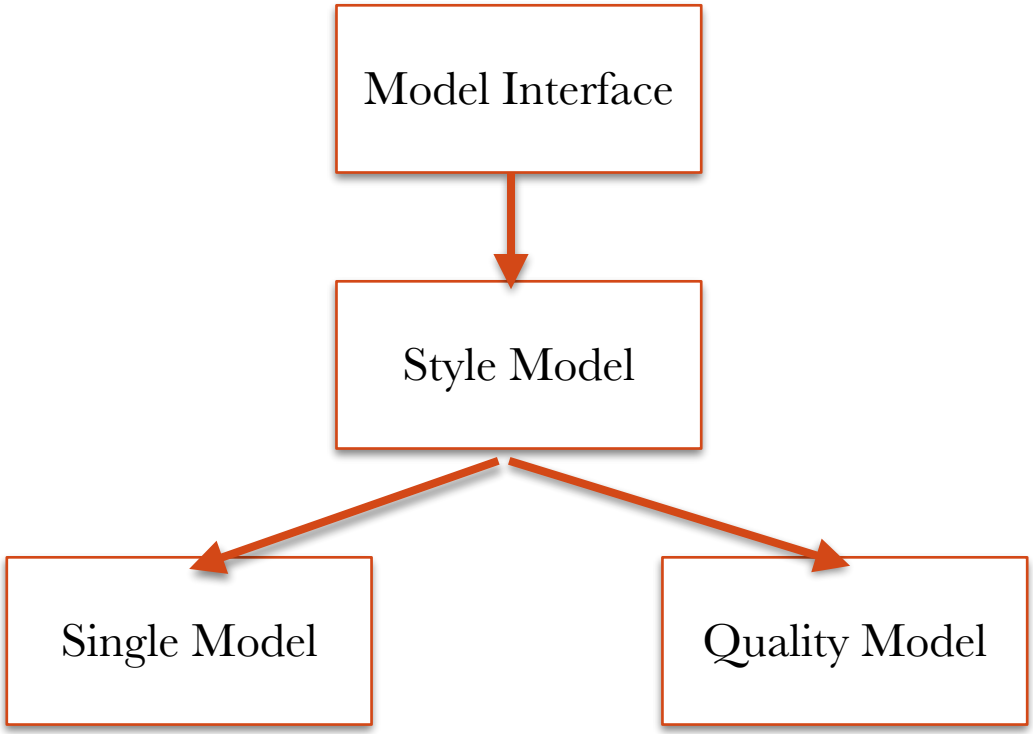
Recommend a market deal

- A deal is the ranked list of properties whose listing price is relative low comparing to the predicted price.

$$deal = Min(\frac{ListingPrice^{style}(t)}{\hat{Price}^{style}(t)})$$

Code Structure

- Python + OOP



Code: https://github.com/ywang110/HouseAssessment