San Diego House Price Predicting

**Wen Yan, Mengting Wang, Salah Ahmad, Xia Song**

### Summary of Our Progress (Previous Reports)

### Report 1 Highlights:

### Motivation and Objectives:

### The business challenge that we are trying to solve in this capstone project is to evaluate and

### project housing value in San Diego area by county. Followed by a couple of years of decline

### since 2006, San Diego housing price is climbing back up again – we are curious to know that

### when the next peak hits and when the next bubble breaks, which area within San Diego holds the

### strongest housing value compare to others; in other words, which area within San Diego has the

### housing value relatively less vulnerable against bigger market.

### Since house price is one; if not; the most factor that impact people’s decision to buy a house, a

### challenge that a house hunter and realtor company faces is weather the house’s listing price is

### fair regarding to current market environment. Is it over or under priced? If so, how much? And

### this also matters on how soon the house will be sold, which is also a critical factor for short-term

### investors.

### To achieve this goal, our direction of approach will be evaluating numerous factors that

### differentiate house values between areas and quantifying appreciation or depreciation of house

### values overtime. It will involve specific periods such as housing booms and bubble breaks, risks

### of default in taxation for properties under macro-economic environment.

### The research will eventually produce insights and guidance to home owners, investors and

### government sectors, hence benefit the public. We’ll deploy various data analysis and machine

### learning algorithms to answer the questions from different perspectives. Eventually the project

### will be set up as prototype pipeline to be able to keep track of new data and identify valuable and

### interesting attributes and findings.

### 

### 

### Motivation and Objectives:

### As the general challenge and business questions laid out, we specified the bigger picture into a

### set of approachable questions, listed as follows:

### What are major factors; except global economic environment; that impact the housing prices in San Diego county under current market?

### In short-term uptrend/downtrend market, what type of houses by location, profile (attributes) will appreciate/depreciate faster than others?

### In long-term (several decades), will certain houses or areas appreciate more than others?

### Can we predict the house price based on current average price movement in its area?

### Can we predict how soon a specific house with a list price can be sold under current market?

### Can we identify the area where the house conditions being the major factor that differentiate house prices (in other words, if we flip the house with enhancement, can we make a big profit)?

### How to incorporate different data analysis skills and machine learning algorithms to address above questions?

### How to build automatic system that keep tracking on the market and identify any hotspot that are most interesting to different parties?

### 

### Data Sources:

### Main data source - San Diego County Data:

### Historical data about characteristics of each property

### Transactions, ownership, tax & mortgage history for each property

### Backup data source - Redfin scraped data:

### One snapshot (Oct. 2017) only

### Redfin page of each property for about 700K properties in San Diego County

### Team roles and responsibilities:

### Project Manager/coordinator: Mengting

### Lead Programmer: Wen

### Database Manager: Salah

### Solution Architecture: Salah

### Analytics & Machine Learning: Xia/Wen

### Visualization: Mengting

### Paper Write-up: Xia/Mengting

### 

### 

### 

### Report 2:

### Summary of Features Set:

The following is the highlight for the main attributes in our dataset:

* PIN (Parcel No.): It is the unique numerical key for each property.
* Land\_Use\_Code: Indicates the type of the property such as residential, commercial, etc…
* Year\_Built: When the property was built.
* Sqft: Total size in square feet.
* Num\_Bed: Count of bedrooms.
* Num\_Bath: Count of Baths.
* Pool: Is there a pool.
* View: What kind of view the property has.
* Sales History date and amount
* Address: Add1, Add2, Street Type, City, State, Zip
* sqft\_price: Calculated field to get the price/sqft

### Approach for Data Access:

The following are the means to access the data:

* Data dump into common format such as .csv.
* Direct query to the DW using SQL or Python.
* BI Tool such as Tableau.

### Solution Architecture:

### 

### 

### 

### Report 3:

### …

### 

### Report 4:

### Modeling pipeline

**model evaluation**

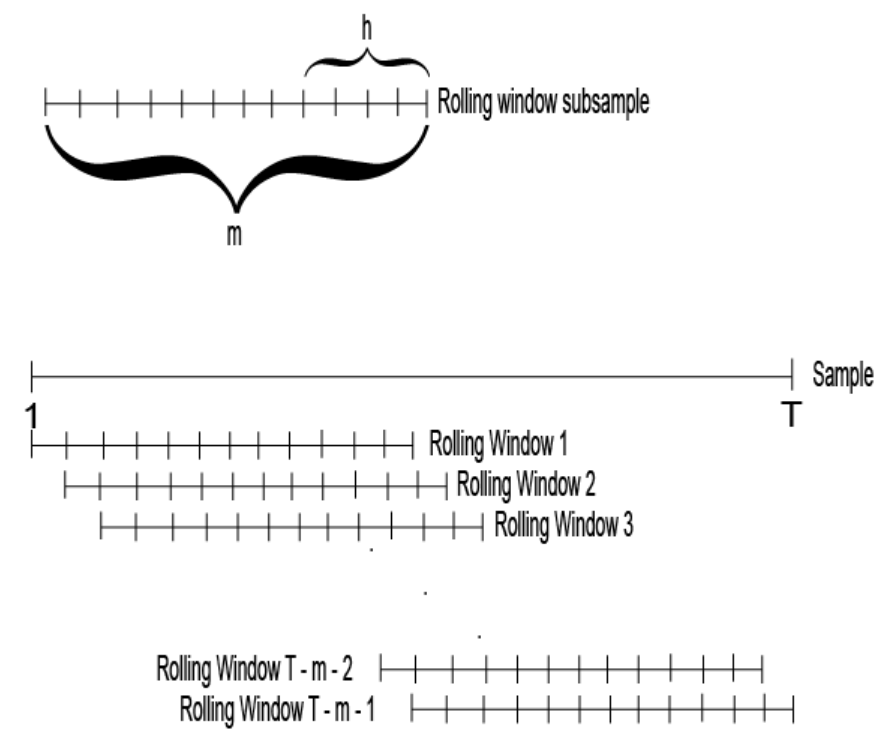
Ideally, the model evaluation should be determined at early stage which affect the model pipeline tracking facts. But since this is time-series prediction, the optimized model will likely be dynamic from time to time. It will be time consuming to train and validate the most optimized models for each time step.

Generally, we’ll use walk-forward validation method with RMSE as metrics for our model evaluation.

**Evaluation Metrics**: RMSE on target variable ‘sold\_price’

**Walk-forward validation**:

It’s good for time-series prediction model evaluation because it’s checking the stability of the model performance over time. While the traditional K-fold cross validation is not appropriate because it’s not reasonable to train the data from later time for prediction on previous time.



Time windows:

1. Forcast/Testing window (h): the lifetime that trained model can be used for prediction before retraining.
2. Sliding/Rolling window (m): the time length that data will be used for training plus the testing window
3. Tracking window (T): the whole time range of data that can be used for validation, the data before this range will be considered too old to be relevant.

Validation process:

1. With a baseline model using latest 1 year data, we lock a grid of parameter ranges for each model type. Eg RandomForestRegressor: n\_estimators = (10,20,30,40 ..), max\_depth = (4,5,6,7,8,9,10)
2. Starting from the 1st tracking window T(eg 1 year), we do a full walk-forward validation for a grid search on the parameter ranges to get the best model parameters.
3. With the selected model parameters, we train a model with data sliding window m(eg. 6 months) and use it to predict forecast window h(eg. 2 month), and roll forward for next 2 month prediction, until we predict T (1 year) new data.
4. Go back to step2 to redo walk-forward validation for next T (1 year) model selection and prediction.

Iteration needed:

Similar to K fold cross validation which need K iterations to evaluate one config in the grid of modeling hyperparameters. Walk-forward validation need (T-m)/h + 1 iterations for one config.

Suppose the model hyper parameter grid to evaluate has dimension of D, and whole data history length is A, the total iteration for validation will be ((T-m)/h+1)\*D\*A/T, plus A/h of final model training for prediction.

Eg. With 1month as timestep, T = 12, m=6, h=2, D=64, A=(2017-1983)\*12, total iteration number is 8960 + 420

**Baseline model**:

1. Dataset: sub dataset of latest year.
2. Training: months 1~6
3. Validation: months 7, 8
4. Testing: months 9, 10
5. Model method: LinearRegression, RandomForest, GradientBoostRegressor
6. Result:
7. Select the best model method, which will be used for continuous feature engineering and walk-forward validation for all time series.
8. Identify the candidate parameter ranges for the best model.

**Pending issues**:

1. Should we use fixed sliding window size m or do we also optimize it as hyper parameter through validation?

Fixed sliding window may lose the accuracy during rapid changing market, but validating it with multiple values need more computing power

1. For each model, we probably need segmentation method for better accuracy. Will a uniform segmentation be used for the whole time series or how can we dynamically changing the segmentation?

**Modeling parameters:**

Modeling pipeline is designed to track the progress of the model’s development which should at least include below parameters which affect model training and prediction results.

1. Dataset – dimension of the records. part of the dataset may be dropped due to applicability,
2. Feature set – dimension of the columns, includes both attributes from original dataset and engineered features
3. Valid criteria – the conditions identified during data preprocessing to exclude invalid data samples and outliers
4. Model profile – model method/name, hyperparams of the model, metrics score of training and testing.

**Versioning:**

We have below options

1. ModelDB from MIT: a server-client framework which include a server listening to receive the recording event, a database backend for model versioning per project and front-end web with visualization.

The project is at early stage. It only covers model profile, but none of others. The main visualization it provides includes a scatter plot of model score and histogram per model profile category.

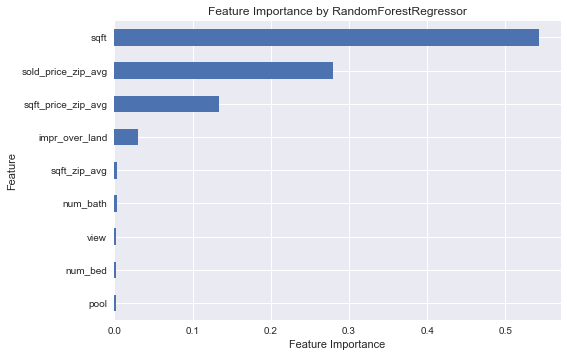
Use ModelDB with unsupported information in comment.

1. Code enhanced versioning: use well formatted python notebooks and python wrappers for all the paramters, to make each model result reproducible. With the modeling parameters recorded in config files, we can run all the configurations and dump score for each of them.
2. Create our own database to track all information, so the code can be freely updated without worrying reproduction of the result.

Right now, option C ‘Code enhanced versioning’ is more promising considering the limited time of project.

### Baseline Model result

1. The feature set used for baseline model has limited information of geolocation which is supposed to have big impact on house price. This is definitely the priority for feature engineering.

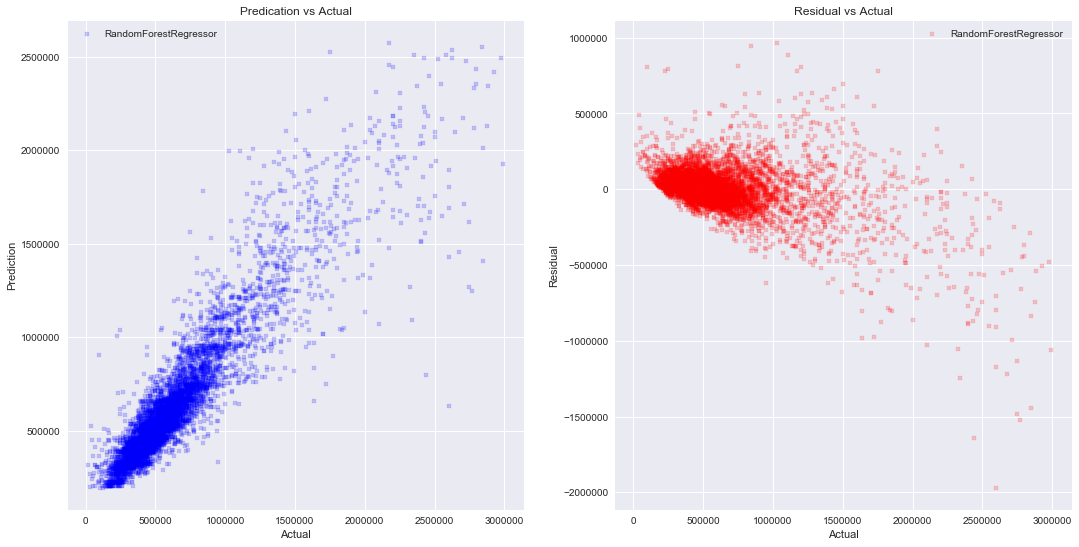


1. RMSE is 146686.

training error: 134589.63200645856

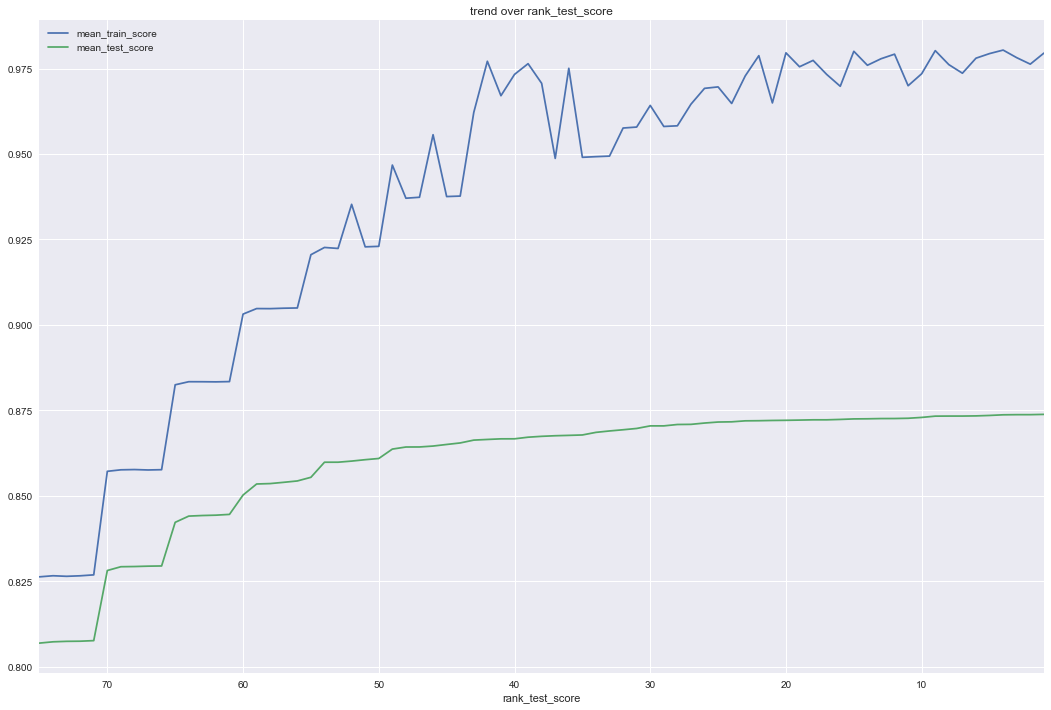
testing error: 146686.02875919602

1. Prediction vs actual and residual



From the residual plotting (right image), we can see the baseline model doing really bad on expensive houses, which suggest segmentation method with multiple modeling for each segment.

1. Learning curve of RandomForestRegressor from the validation ranking out of 75 parameter combinations from grid search. We can see after about 25 iterations, the model improves very slow. Each big bump is when we use bigger max\_depth, and each slow improvement is by using bigger n\_estimators. We probably can stop at depth 9.



### Solution Architecture:

### …

### 

### Findings:

…