# Exploring the Data Set

## Test Data

* both data sets are targeted at predicting the price of houses sold in 2017.
* data is reasonably clean

## HDB Training Data

* contains houses sold from early 1990s to 2017
* contains potential features that are categorical as well as numerical
* data appears to have some outliers
* data is reasonably clean

## Private Housing Training Data

* contains houses sold from early 1990s to 2017
* contains potential features that are categorical as well as numerical
* contains some outliers (e.g. some houses with large floor area, but low price). These records are removed from training data set manually.
* contains dirty data (e.g. tenure descriptions are incomplete)

Figure 1 HDB Training Data

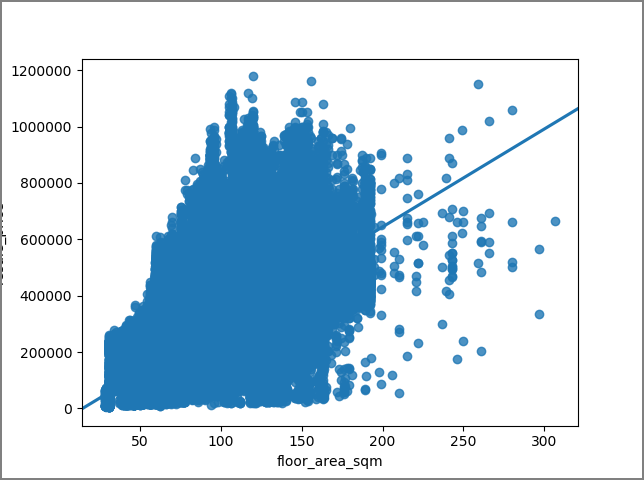
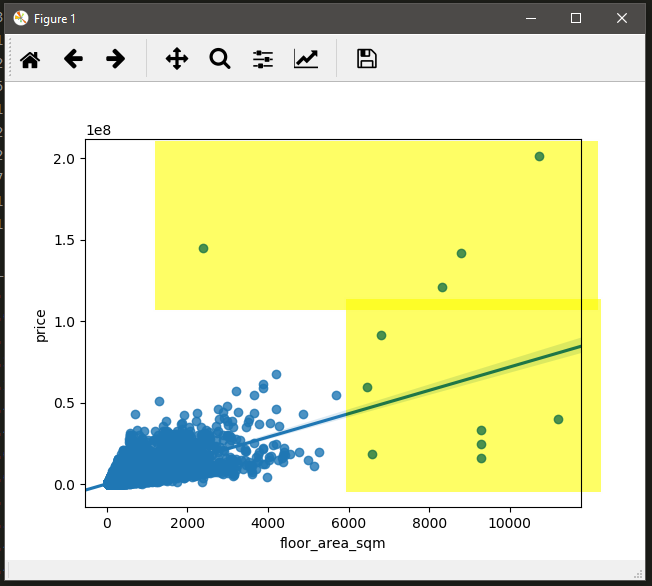


Figure 2 Private Housing Training Data with outliers highlighted



Most correlated features (in descending order)

|  |  |
| --- | --- |
| HDB | Private Housing |
| 1. resale\_price 2. flat\_type 3. floor\_area\_sqm 4. month 5. lease\_commence\_date 6. floor 7. storey\_range 8. longitude 9. latitude 10. town 11. street\_name 12. block 13. postal\_code 14. flat\_model | 1. price 2. floor\_area\_sqm 3. tenure 4. type\_of\_sale 5. property\_type 6. floor\_num 7. address\_block 8. month 9. area 10. contract\_date 11. longitude 12. address\_street (derived from address) 13. project\_name 14. unit\_num 15. completion\_date 16. postal\_district 17. postal\_sector 18. postal\_code 19. latitude 20. region |

# Data Preparation

* both Label Encoding as well as Feature Hashing are used to represent categorical features (e.g. area, storey range, property tpe)
* One Hot Encoding is not used because it generates too many feature columns that computation time takes extremely long. Instead, Feature Hashing is used.
* Since house price is too large compared to other features’ values, we apply BoxCox transformation to price to achieve normal distribution.

# Training Data Selection

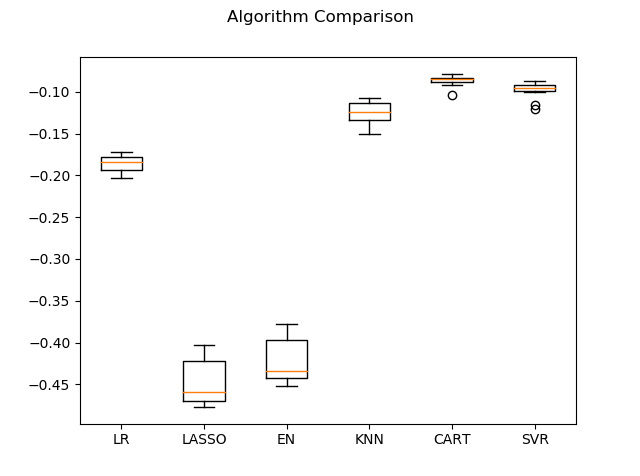
* Since test data set is aiming at housing prices in 2017, we decided to train our models with data from 2015, 2016, 2017. Our assumption is that buyers and sellers do not need to know old house price. But they need to know house prices that are based on recent house prices in current market so that they can buy or sell at competitive price range.
* For private housing dataset, we noticed that test data set is aiming at only Resale, so we decided to exclude all other sale types from our training data.

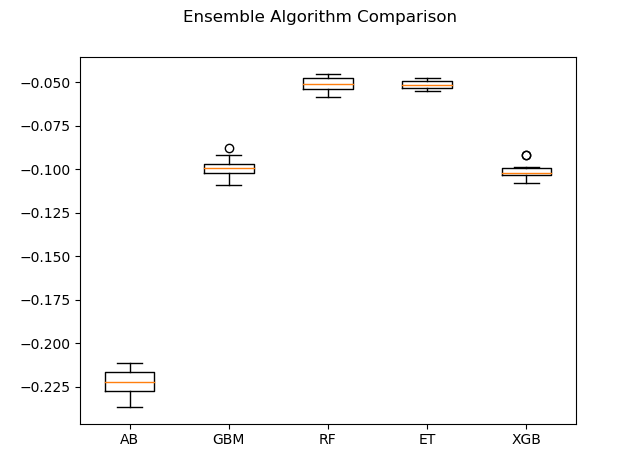
# Feature Selection

* We rely on the correlation characteristics of training data sets to identify potential features
* But we let the algorithms discover the feature importance themselves

# Evaluating Algorithms

* We evaluated a few candidate algorithms both Linear and Ensemble types
* Linear algorithms we evaluated from scikit-learn include -
  1. Linear Regression
  2. Lasso
  3. ElasticNet
  4. KneighborsRegressor
  5. DecisionTreeRegressor
  6. SVR
* Ensemble algorithms we evaluated from scikit-learn include –
  1. AdaBoostRegressor
  2. GradientBoostingRegressor
  3. RandomForestRegressor
  4. ExtraTreesRegressor
  5. XGBRegressor
* In general, Linear algorithms yield higher erorr rate than Ensemble algorithms. Diagrams below shows a sample of evaluation over 20,000 records of HDB dataset. Same holds true for Private Housing dataset.
* Among Ensemble algorithms, RandomForest and ExtraTrees seems promising, but during training with larger dataset, their performance degrades.
* Only GradientBoostingRegresor (GBM) and XGBRegressor (XGB) yield acceptable training time and comparatively good error rates.





# Tuning the models

* We used scikit-learn’s GridSearchCV and RandomizedSearchCV to find optimal parameters for the chosen two algorithms: GBM, and XGB.
* However, it takes extremely long time for GBM to complete GridSearchCV or RandomizedSearchCV. So, we started to experiment with tuning parameters manually.

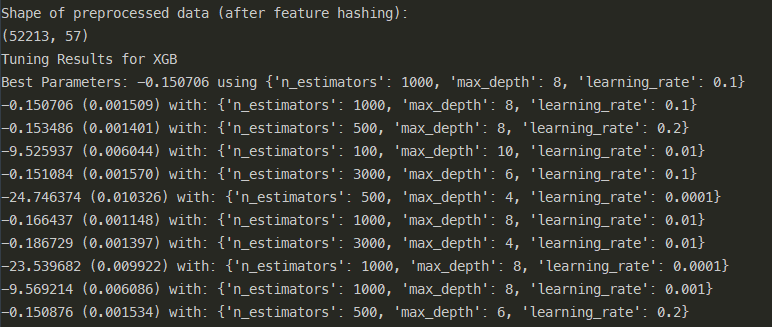


Figure XGBRegressor best parameters found via RandomizedSearchCV (took 6 hours on my Xeon workstation laptop to obtain this!)

# Training the models

* KFold cross validation scoring is used for finding optimal parameters.
* Train-test-split approach is used for training and predicting the test data.

# Appendix: Screenshots of Data Exploration

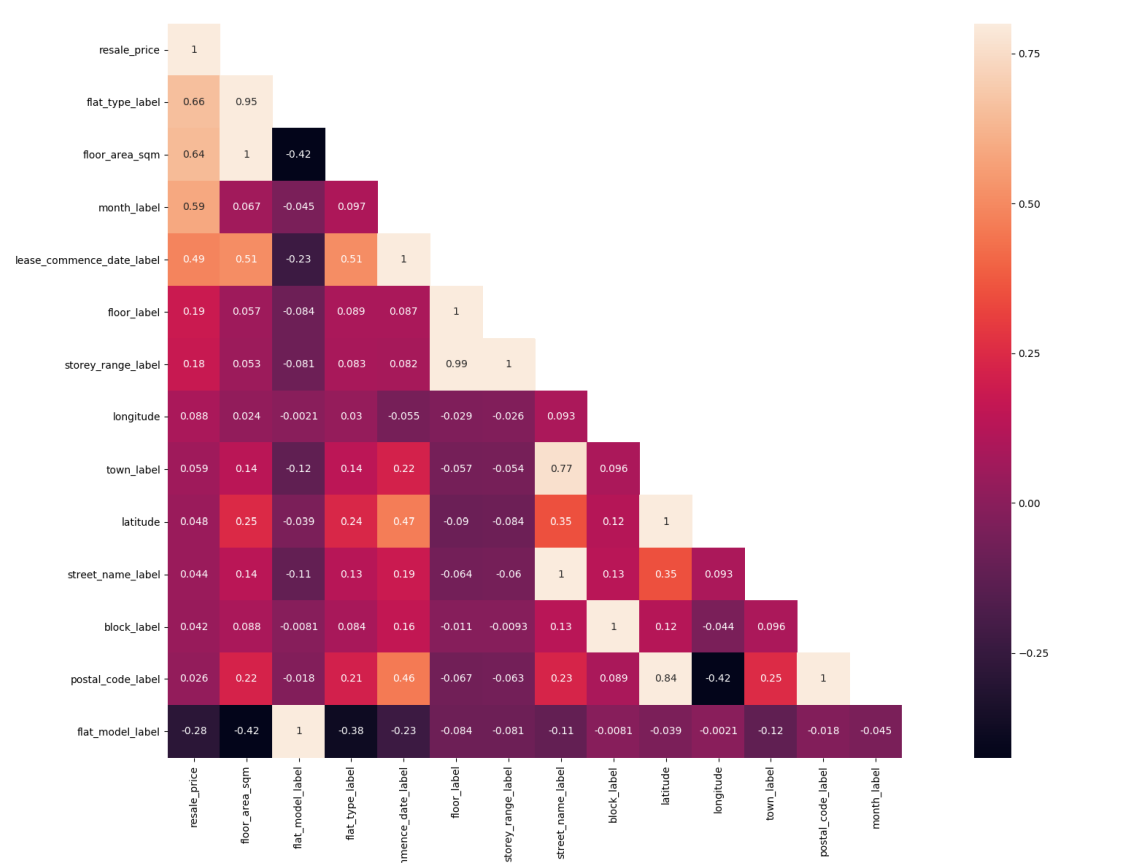


Figure HDB Training Data Correlation

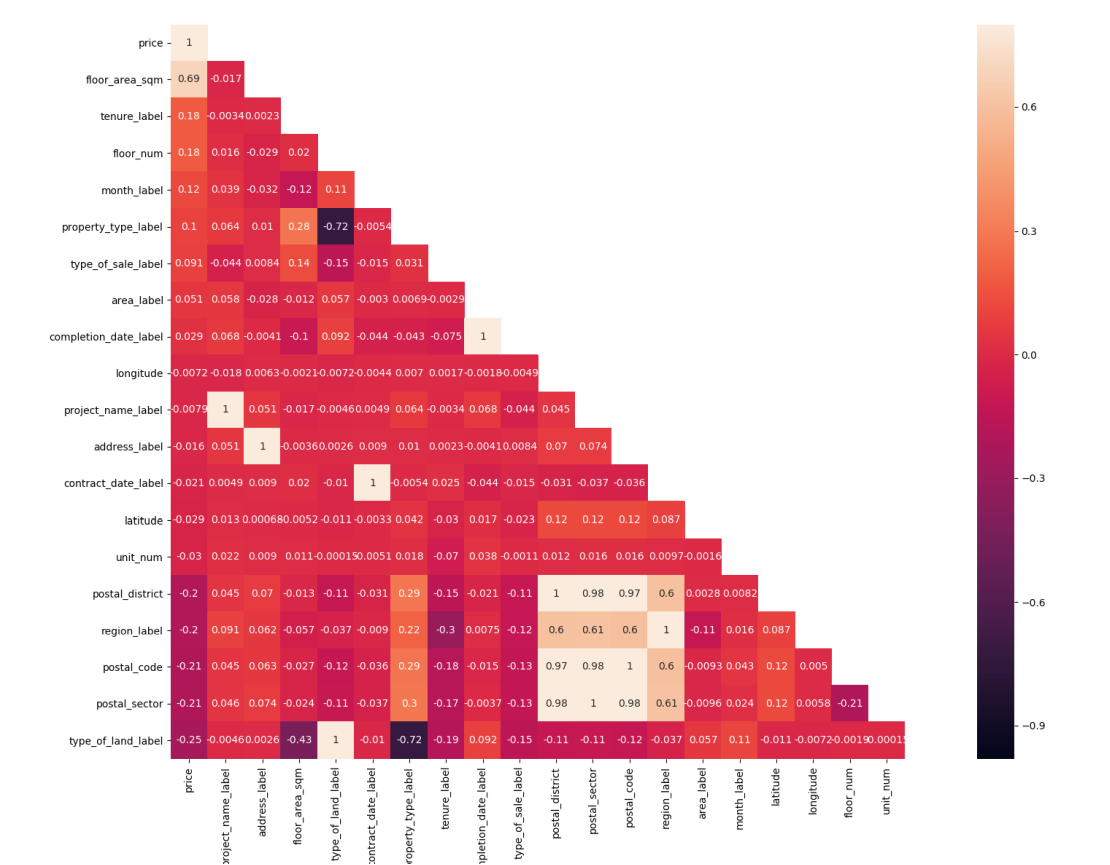


Figure Private Housing Training Data Correlation

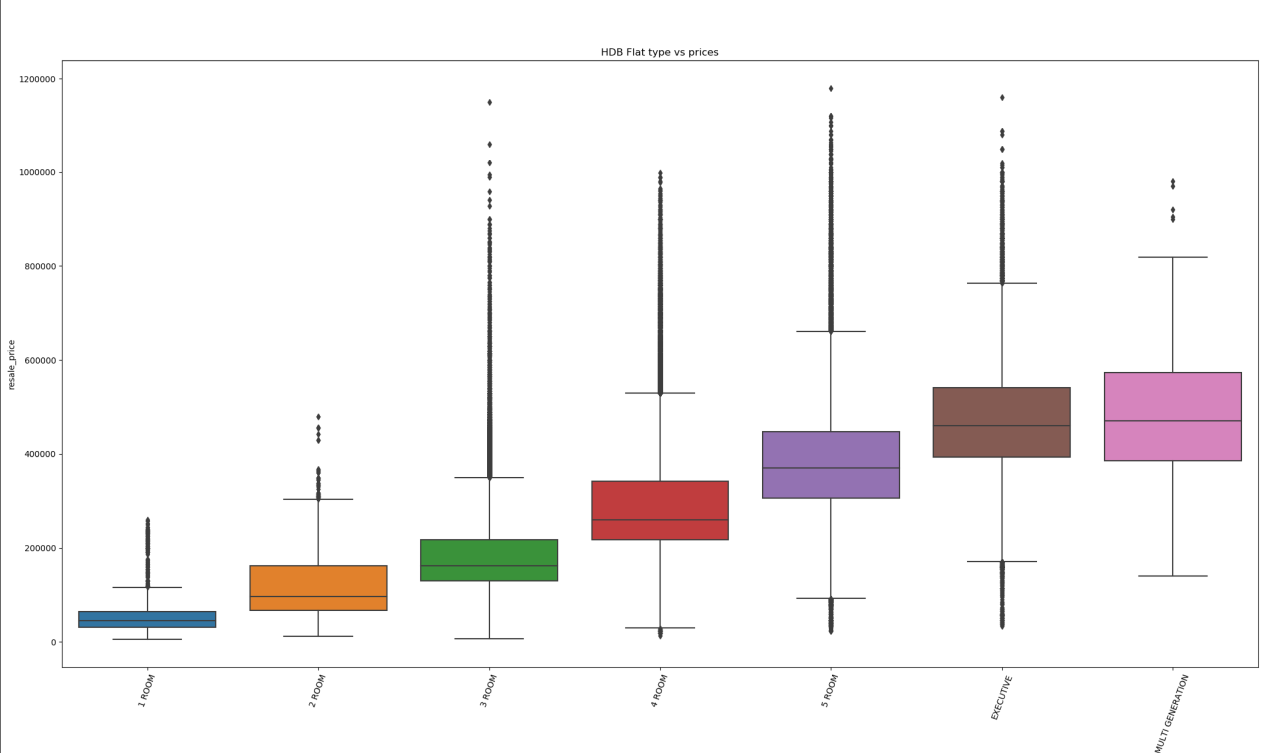


Figure HDB Flat Type vs Price (left to right: 1 room, 2 room, etc)

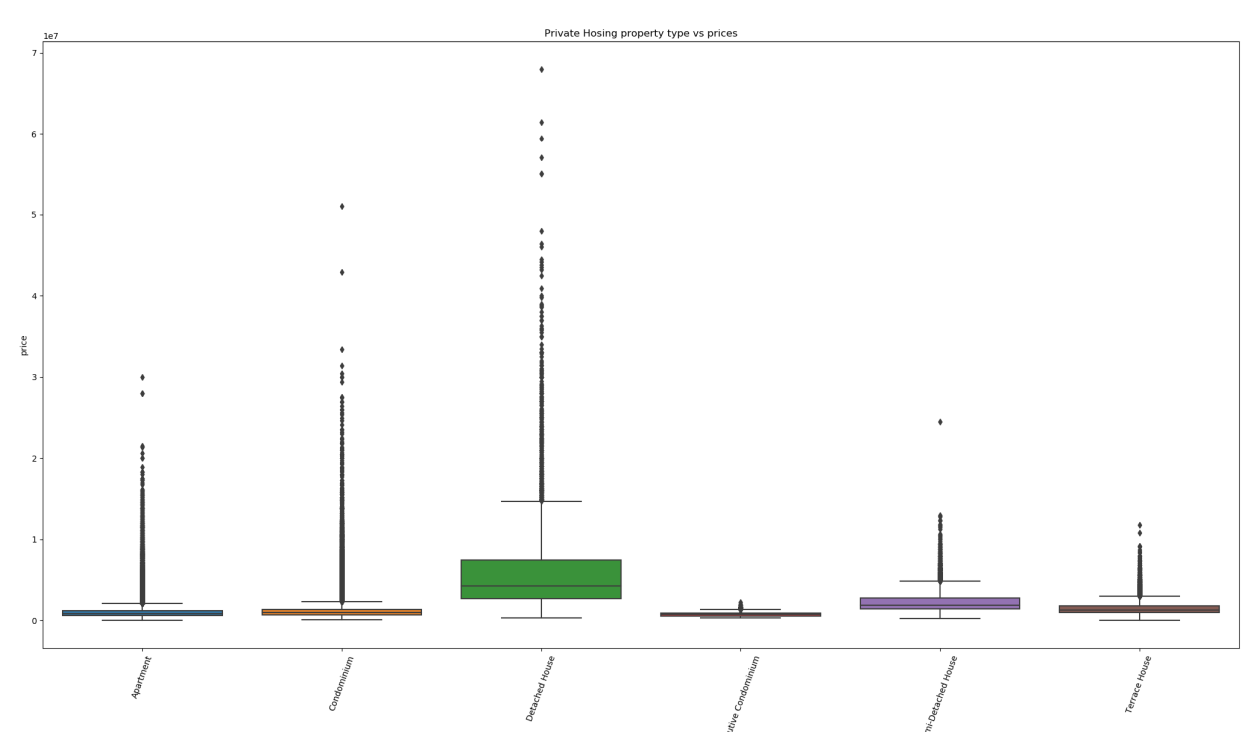


Figure Private Housing property type vs Price

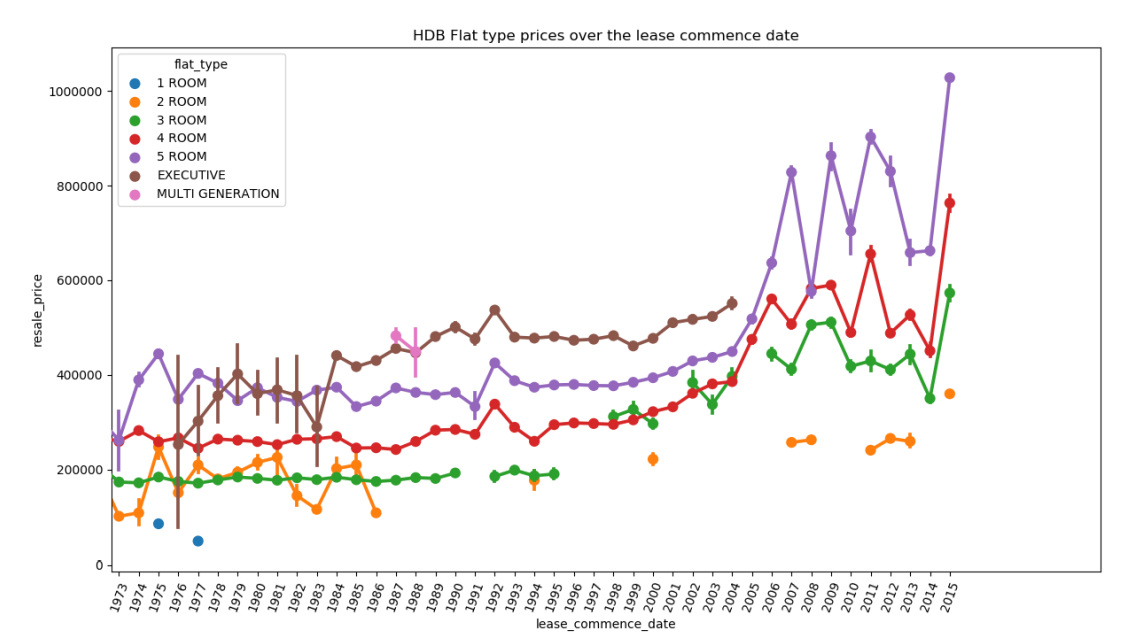


Figure HDB flat type prices over the years

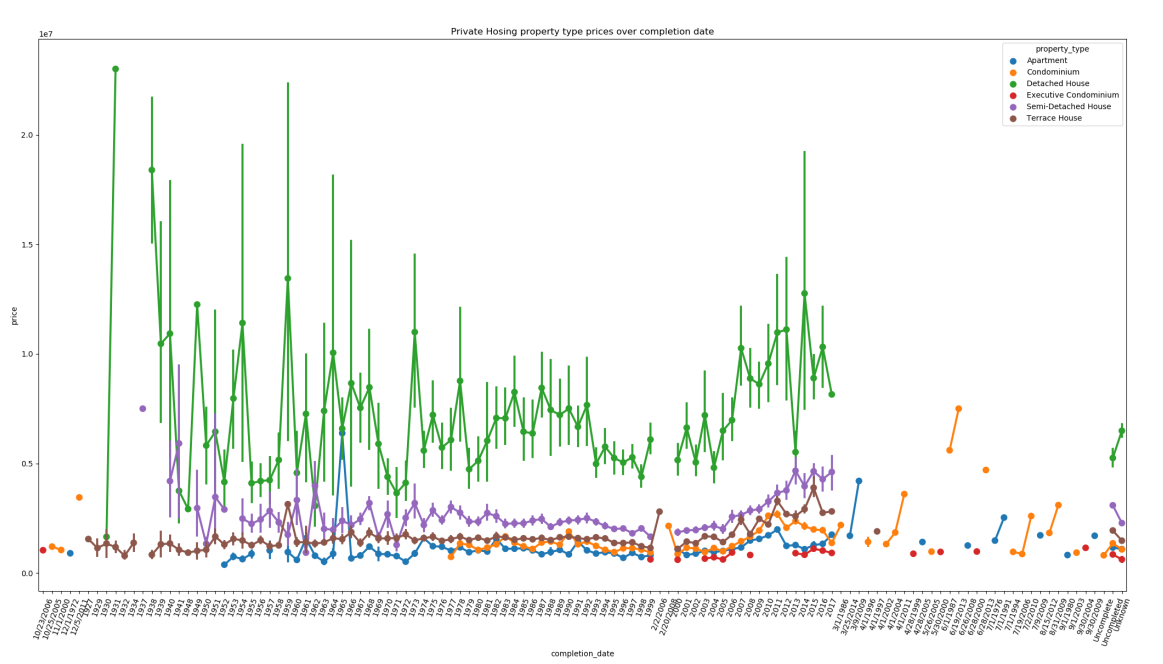


Figure Private Housing property type prices over the years (green colour shows “detached house” type)