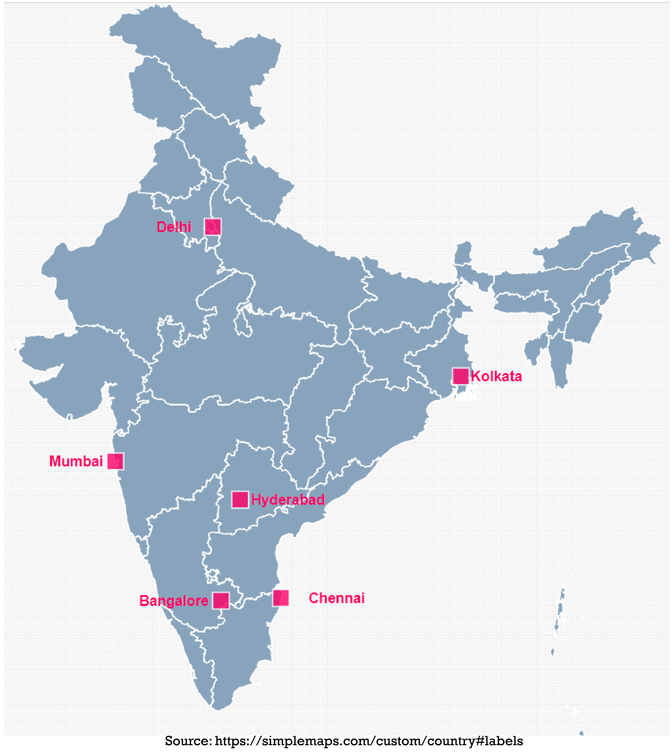
Prediction of Housing Prices of Indian Cities with AdaBoost Regression and Neural Network

Andrew Rose, Hiranyeshwar Vuppala
Faculty of Engineering, University of Victoria

Summary

- Like most sectors, the real estate sector has been transformed by technology in recent years. Nowadays, realtors in North America employ multiple listing services (MLSes) to efficiently sift through the property market and match buyers to sellers.
- Potential applications of a machine learning tool capable of predicting housing prices from vital statistics about housing properties include:
  - Use by housing sellers to obtain reasonable estimates of how much money they can expect to sell their property for.
  - Use by housing buyers to determine roughly how much money they should offer for a property.
  - Use by builders and developers of housing projects to gain insight into what features of housing properties contribute most to its value, and thus how to design the most desirable housing possible.
- We used Indian house listings from major cities like Bangalore, Chennai, Delhi, Hyderabad, Kolkata and Mumbai because North American property was only available for licensed real estate agent



Source: https://simplemaps.com/custom/country#labels

Data Munging

- In the Kaggle data set we obtained, every sample had Price, Location, Number of bedrooms, area of the house and amenities such as gym and elevator.
- We had two data sets for each city
- One data set had all the missing data in amenities converted into simple imputation called NaNs Imputed.
- We removed all the missing data for other data set called NaNs Dropped.
- Location for each was converted into coordinates system to make entire data set into numerical
- We removed all the samples whose location was unknown in coordinates system
- We normalized data to get even results in both adaboost regression and neural network.

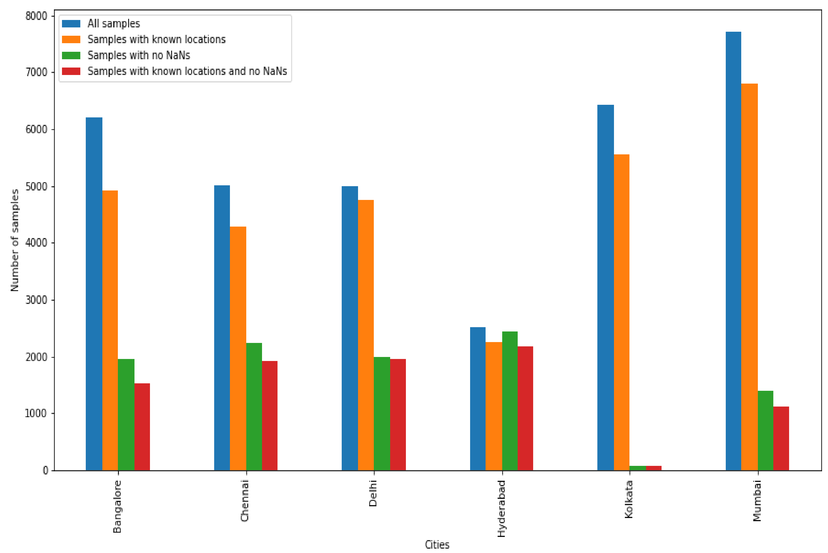
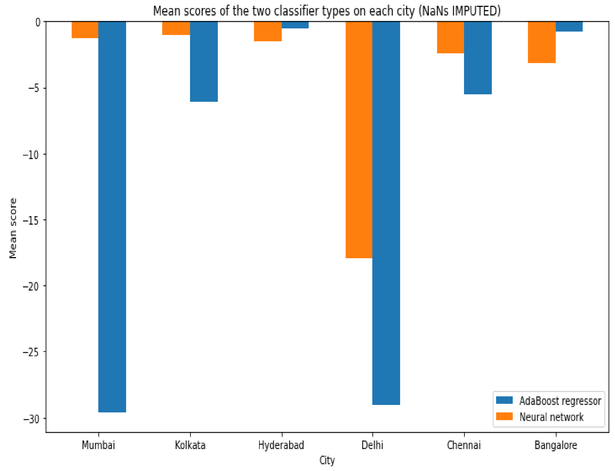


Figure 1: Shows number of sample left after each data munging process

Results



- When the imputed values are binary data, inevitably results in the incorrect evaluation of many samples that have been grouped into the wrong branch in AdaBoost Regreesor.
- In contrast, a neural network that is supplied mean values allows those values to propagate through the network without causing nearly as much distortion of the output.
- AdaBoost Regressor can yield vastly superior results when performing one-dimensional regression, its accuracy can be ruined by mean imputation, which neural networks are not so totally derailed by.
- The speed of training decision trees makes it more feasible to finely tune their performance and input data formatting in practice, which is no small consideration

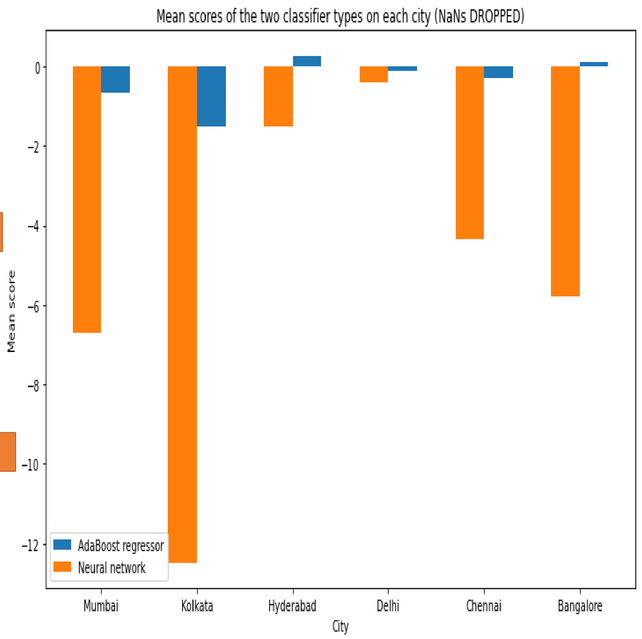
Conclusion

- The team's most important conclusions are that AdaBoost regression with decision trees works well for this application and that neural networks are unsuited to one-dimensional regression.
- Mean imputation appears to be highly damaging to decision.

Future Work

- Comparison between decision trees and other base estimators within the AdaBoostRegressor setup used for this problem.
- Investigation of the effects of different imputation strategies on both decision-tree-based learners and densely-connected neural networks.
- Investigation of principal component analysis for the purpose of determining which major factors and combinations of factors determine a property's value.

- Research into the scikit-learn API revealed that categorical data must be converted to one-hot encoding.
- The AdaBoost Regressor performed far worse with one-hot location data than it did with no location data at all.
- 10-fold cross-validation was used on both the AdaBoostRegressor and the MLPRegressor as a means of eliminating variability in the regressors' estimated real-world performance.
- Both regressors were optimized using squared loss, simply because that is the only available loss function in the MLPRegressor class and the two regressors were intended to be directly comparable in as many ways as possible.
- Both regressors' overall accuracy was determined by taking the mean of their R2 scores over the ten folds for each city. This was repeated for both variants of the dataset, NaN dropped and NaN Imputation. To read a graph comparing R2 scores, remember that R2 scores range from negative infinity (arbitrarily bad) to 1 (perfect), and that higher values are better.



References:
[1] https://www.kaggle.com/ruchi798/housing-prices-in-metropolitan-areas-of-india