Project: AirBnB Pricing Prediction

**Team:**

James R Gearheart  
Danny Zhuang  
Bob Saludo  
Ryan, Wallace

**Milestone #2:** Literature Review

The online article titled “Predicting Airbnb Listing Prices with Scikit-Learn and Apache Spark” by Nick Amato was published on April 20th, 2016 on the Mapr blog site. This article details the construction of a predictive model using python’s Scikit-Learn package in combination with Apache Spark for performance enhancement. This model leverages the publically available dataset titled “Inside Airbnb” published by Airbnb and focuses on the San Francisco area.

The article walks the reader through the step-by-step process of building and evaluating a predictive model using publically available data. In the first section, the author explains the available data sources and provides python code for accessing and evaluating some core features of the data. This initial analysis demonstrates the raw nature of the data and the need to clean the data. In order to create a concise model, the author limits the data to Airbnb listings in the San Francisco area with one bedroom. This filter captures the majority of the available listings and limits the possible complex interactions involved in a broad study of all listing types.

Once the author limits the data to a select number of features, he cleans the data by replacing missing values for the ‘reviews\_per\_month’ field from NaN to 0. He also cleans the data by filtering out any listings where there are missing values for price, bedrooms and beds. The next step is to recode the categorical variables into dummy variables (values of 1 or 0) by using the ‘get\_dummies’ feature of Scikit-Learn. This technique allows the model to incorporate the categorical data into the model as unique numeric features. An analysis of the scatter plot matrix shows that there is not an issue with correlation that would need to be addressed.

The author uses several predictive models from the Scikit-Learn package to demonstrate the differences in predictive performance by using the median squared error metric to evaluate model fit. The models that were initially developed were: linear regression, ridge and lasso regressions, ElasticNet, Bayesian ridge and Orthogonal Matching Pursuit. A comparison of the model outputs shows that the Bayesian ridge model provides the lowest error from the six candidate models. The error is translated into dollars in order to be consistent with the project goal of predicting prices. The Bayesian ridge model results in an error of $30 - $35 dollars.

The next step is to implement a model tuning algorithm. For this process, the author uses the GridSearchCV function from the Scikit-Learn package. The GridSearchCV function is an ensemble model that explores a large variety of hyperparameters through an exhaustive search process. This process tries all the supplied parameter combinations and uses cross-validation to find the best one. The concern with this approach is that it requires a lot of CPU processing power and time. In order to process the exhaustive grid search efficiently, the author uses Apache Spark to distribute the process across several machines and cores. The author uses the GradientBosstingRegressor predictive model in combination with the GridSearchCV cross-validation technique on the Apache Spark system to achieve an error rate of $21.43.

Conclusion:

This article provided an excellent step-by-step look at developing several predictive models and assessing the accuracy of those models. The author also demonstrated how Apache Spark can be leveraged to increase processing efficiency and allow the full range of exhaustive search methods to be employed in model development.