Machine Learning Engineer Nanodegree

Capstone Proposal

Joaee Chew
August 12th, 2017

Proposal

Domain Background

The domain for this project is in the pricing of AirBnb listings in London. For new and potential Airbnb hosts, it is difficult to ascertain how much their property could earn them. While there are available commercial tools such as http://www.airsorted.uk/ that predict Airbnb value, these models are generic with minimal features used and no indication of model accuracy.

Using publicly available information, this project will build a predictive model for the yield of potential Airbnb listings. Yield is defined as the amount of revenue that a property will earn over a year. The calculation of yield uses Inside Airbnb's 'San Francisco Model' that uses a review rate of 50% to convert reviews to estimated bookings (http://insideairbnb.com/about.html).

In addition to the problem of predicting an expected yield, this project seeks to use natural language processing to extract non-traditional features in the feature engineering stage. This will demonstrate if there are certain keywords such as 'luxurious' or 'bohemian' that can improve model accuracy, and if there are correlations between those keywords and yield.

I will also go one step further and implement an interpretability layer, using the TreeInterpreter package (https://github.com/andosa/treeinterpreter) to decompose predictions into feature contribution components. This will help to identify potential "levers" that hosts can pull to increase their yield.

Problem Statement

The problem is to predict the yield on prospective AirBnb listings in London, using readily available features that are present at the point in which a potential host lists his/her new property (e.g. no. of rooms, description, location). Yield is defined as the amount of revenue that a property will earn over a year, and I will use Inside Airbnb's 'San Francisco Model' for the calculation of yield based on price, average length of stay and review rate. This is a replicable problem for all new Airbnb listings in London.

Datasets and Inputs

At Inside Airbnb (http://insideairbnb.com/get-the-data.html), publicly available information about a city's Airbnb's listings have been scraped and released for independent, non-commercial use. This includes details about the listing such as no. of rooms, guests available, description, location as well as information about the yield such as price and no. of reviews.

Specifically, I will use the detailed listings information for London listings active from 3rd October 2016 – 4th March 2017. An active listing is defined as a property that has been reviewed at least once during this time period.

The dataset has 53904 rows and 94 columns. However, there are several missing values and cleaning of data necessary. For example, I will remove data points where availability (no. of days the property is available for rent in a year) is less than 50% as this indicates part-time listings that may not be comparable. Other important features, such as the price and number of reviews might be missing. After data cleaning, the dataset has 15441 rows and 14 columns.

Importantly, the dataset contains the listings description as well as headers in text format that I will mine for additional features (see Project Design).

Solution Statement

I will build a supervised regression model and train on the existing data described in the previous section. The model will predict yield, and will be measured by its error rate (mean squared error or similar). It is replicable for every new listing on Airbnb in London.

I intend to use 3 regression algorithms:

1. Linear Regression - To establish a baseline model
2. Decision Tree – This is a more complex model that can capture nonlinear relationships in the dataset whilst still retaining interpretability. This is also robust to missing values, and has the added advantage of being able to help with feature selection based on feature importance.
3. Random Forest – This is the most complex ensemble model built from decision trees, and should be able to provide additional accuracy. While it sacrifices interpretability, sense checking can be made with the earlier decision tree model whilst added an interpretation layer using Tree Interpreter allows model users to break down individual predictions.

For feature extraction from textual data (listings descriptions), I intend to use bag-of-words to represent the data in a vector format. I will also iteratively use TF-IDF, and LDA for topic modelling, to test for better results.

Benchmark Model

To my knowledge, there are no existing Airbnb pricing models released to the public. However, I will use a hold-out set to test my model, and the error rate (e.g. MSE) will be used to measure the accuracy of the model. I will also build challenger models, starting with a linear regression model to act as the baseline benchmark.

Evaluation Metrics

I propose the use of mean squared error between the predicted yield and actual yield. This measures the average of the square of the errors (difference between the actual yield and predicted yield). It is always non-negative, and values closer to zero are better. The formula for this is as follows:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Project Design

The intended workflow for this project is as follows:

1. Read and analyse the data as described in the ‘Datasets and Inputs’ section.
2. Clean the data. This involves, but is not limited to, the following steps:
    a. Remove unnecessary features. This includes data that does not help with predicting yield, such as listing URL or scrape ID.
    b. Remove ‘leaking’ features. This includes data that should not be used to predict yield, such as ratings (which are not present at the point of listing), or the no. of other properties listed by the host (model should be built on the property only and not the host).
    c. Removing incomplete listings (e.g. listings with no price, or with availability less than 50% which indicates part-time listings that are not comparable).
    d. Impute missing values (e.g. using the most frequent or median where appropriate).
3. Build the model. The propose to use three models of increasing complexity, both to test for increase in predictive power but also to act as challenger models to the final model.
    a. I will begin with a simple linear regression to establish a benchmark accuracy.
    b. I propose the use of a decision tree as a next step.
    c. Finally, I will use a random forest ensemble model as the final model.
4. Feature engineering. This will be done iteratively with step 3, with new features tested against model accuracy. Potential ideas include:
    a. Using natural language techniques to extract features from the description. This involves the following NLP pipeline:

        i. Text data preparation (creating corpus, word stemming, creating document term matrix, removing sparse terms etc.)
        ii. Selection of features using bag-of-words and/or TF-IDF
        iii. Use topic modelling to discover any natural topics in the listing descriptions
    b. Bucketing categorical features to reduce dimensionality (eg. Whole apartment vs. non-whole apartment instead of entire class size).
5. Test the model. Here I will print final accuracy measures and produce insights such as feature importance.
6. Add interpretation layer to the final model. As a final step of the project, I will add an interpretation layer using the TreeInterpreter package (https://github.com/andosa/treeinterpreter). This decomposes each prediction into feature contribution components for tree-based models, and helps improve transparency into model predictions.
