Airbnb Listing Classifier

Is your vacation booking a good deal?

Safi Khan

Background:

Have you found yourself often spending hours on Airbnb or other booking sites, wondering if this place is worth the money? What if the listing is new and does not have many reviews? You have never been to this city, so how would you know the reasonable price to pay for the outskirts? Are you willing to spend hours reading multiple descriptions and finding the prices of similar properties nearby? And when you finally think you have found a deal, you realized it's a hostel-type listing as opposed to what you were looking for - private space.

I have been guilty of spending hours on the Airbnb site trying to find the best place within our budget. Therefore, when I came across this Airbnb listings dataset, I wanted to build a tool that will tag any given listing with Good, Fair, or an Expensive deal by comparing the prices of similar size listings in a specific neighborhood.

Data Source:

For my project, I acquired the data of Airbnb listings in London, England, from the OpenDataSoft website. You can access the source data from here. The original data had over 47,000 listings from March 2017. The raw data had 89 features of various data types – text, numerical, URL and visuals.

Listing Classification Mechanism:

We classified the listings in the following steps:

Step 1: Take listing price mean and standard deviation based on the following:

- Neighborhood (Location)
- Type of Property (House, Apartment, Hostel, Cottage, etc.)
- Type of Room (Shared or Private)
- Number of Bedrooms (Size)

Step 2: I established the Fair deal's upper and lower bounds by taking half the standard deviation above and below the mean price.

Step 3: Listings above the upper bound of fair deal were classified as **Expensive** and the ones below lower bounds as **Good.**

Step 4: Then, we simply added it as a feature to our source dataset.

Data Modelling:

The raw data had 89 features. We eliminated some features to reduce it down to 45 by:

- Dropping all features missing more than 40% of the data
- Eliminating the features that had no variation, e.g., city, country as our data was only London based
- Removing personal information for the host, such as location, pictures, and activity, did not add value to models classifying the deal type based on prices.
- All visual and web data, such as pictures and links, were eliminated to keep the process less complicated.
- Any other data that I deemed does not hold predictive power for the deal classification.

To set up the data to be ready for data modeling, I used pandas **getdummies( )** functionality to One-Hot Encode all categorical variables such as Property type, Cancellation Policies, Amenities, etc. Low-frequency sub-categories were grouped into *others* to ensure they do not create any bias in our model.

Once our preprocessed data was ready, I decided to split non-categorical text data and numerical data to run classification models separately on both types. For the last exercise, I combined the data by taking text-based model predictions probabilities as additional features to my numerical dataset. For ease, I'll summarize them as such:

- **Machine Learning Modelling 1:** Non-text data
- **Machine Learning Modelling 2:** Text data using NLTK
- **Machine Learning Modelling 3:** Non-text data with probability predictions from best performing text data models

We used SMOTE methodology for Up-Sampling of our training data to address the imbalanced the target class using 'all' sampling strategy and k-neighbors set to 3.

Machine Learning Modelling:

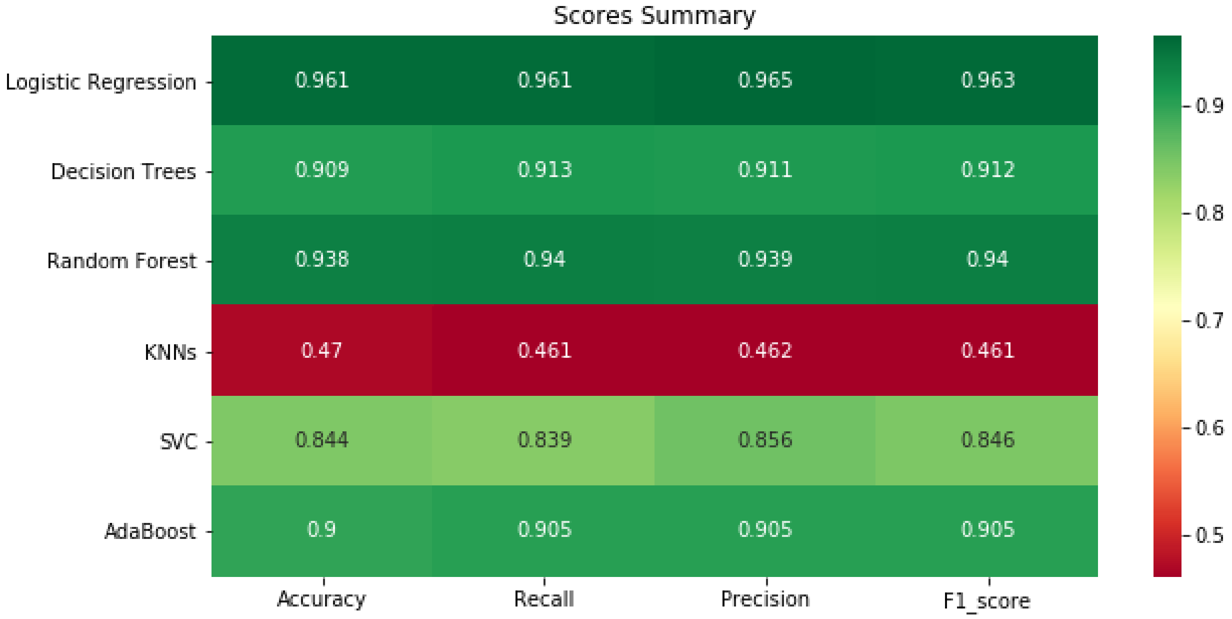
For all three versions of our data, we used a combination of the following Scikit-Learn Classification Models:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- K-Nearest Neighbors Classifier
- Support Vector Classifier
- AdaBoost Classifier

I used pipelines and Grid Search Cross-Validation with 5 folds for hyperparameter tuning to find the best models. Instead of feeding all these models in a single Pipeline, I used separate pipelines to maintain simplicity and analyze their performance separately.

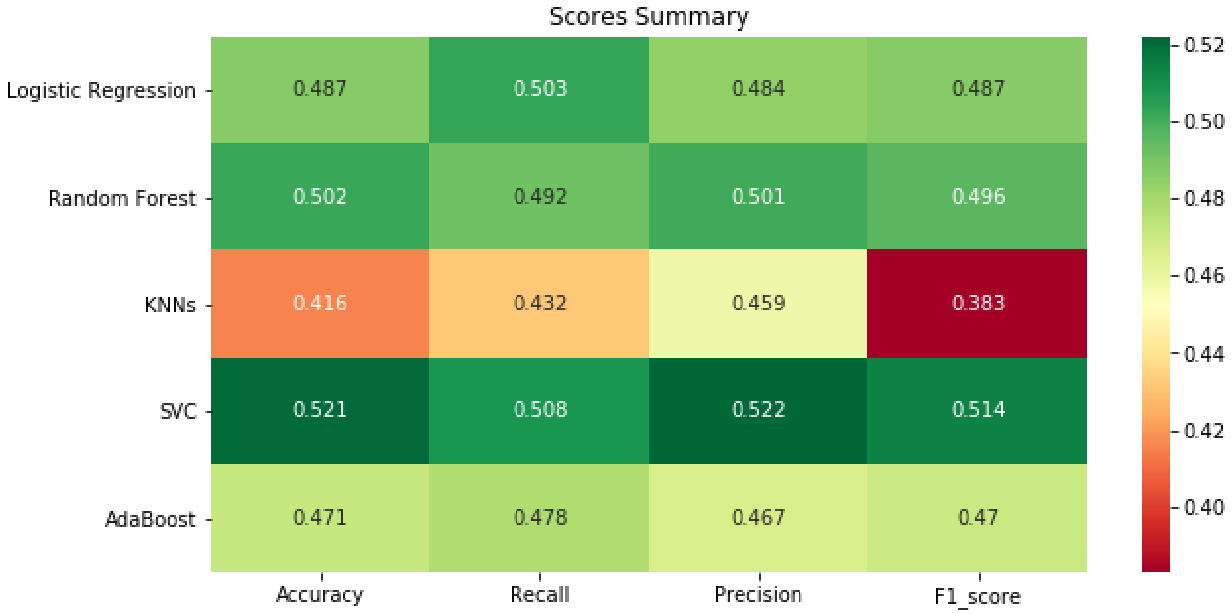
ML Model Results:

- **Machine Learning Modelling 1:** Non-text data



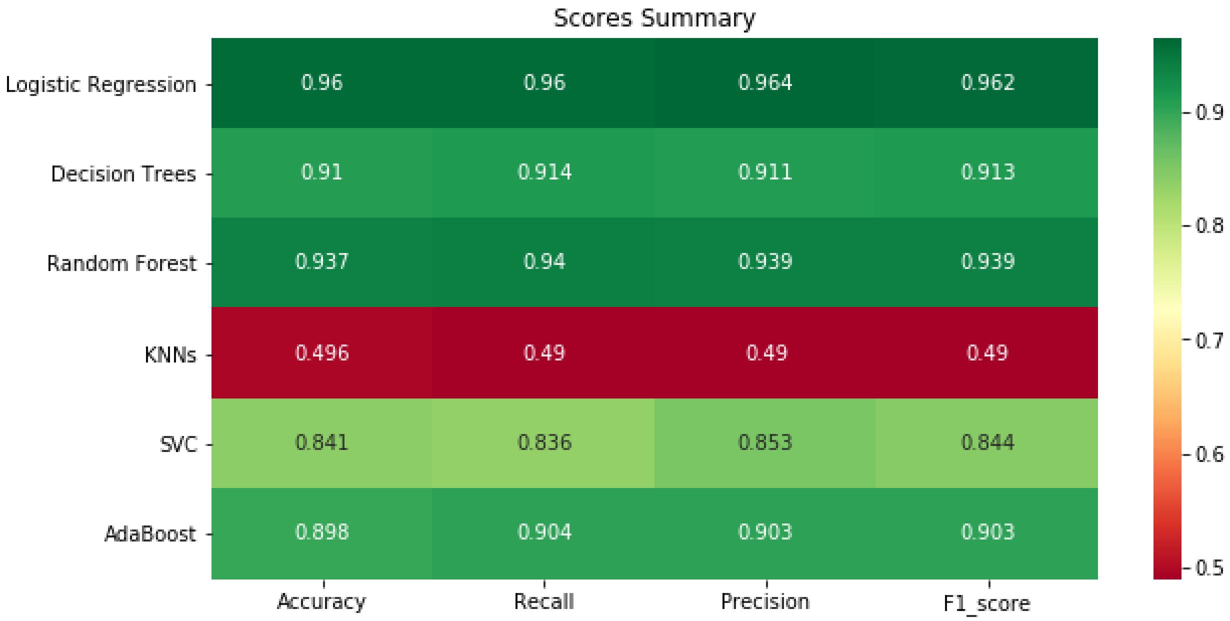
Comments: In our first version Logistic Regression slightly outperformed other models. Decision trees, Random Forest, and AdaBoost also performed well, scoring above 90% in all key metrics. SVC classifier lagged slightly with approx. 85% score in all our evaluation metrics. KNNs performed the worst with 47% accuracy, which is extremely poor compared to other models but better than random guessing.

- **Machine Learning Modelling 2:** Text data using NLTK



Comments: All our models performed moderately for the text data, with KNN's again performing the worst, but SVC did quite well in comparison performing over 52% in almost all categories. Since it was a multi-class dilemma, we had some worrisome results where models mistakenly classified good deals as expensive and vice versa. Though this provides slightly better results than random guessing, I would not rely on text-based ML models to classify the deals.

- **Machine Learning Modelling 3:** Non-text data with probability predictions from best performing text data models



Comments: The models performed similarly to version 1 with minimal fall in all evaluation metrics for all models except KNNs, which showed a slight improvement in predictions. I expected such a decrease in the evaluation metrics after analyzing the low performing text-based models; therefore, we should stick with purely numerical and categorical features-based models and ignore texts while booking your next vacation.

Future work:

This project's real-world application is only feasible if we can integrate our models with the live data to factor in demand and supply. All listings might end up classifying as expensive during high demand seasons, and that will be inaccurate. Though the prototype version, such as this, can also be made at the back end by factoring in week number when calculating mean prices and then deploying it to the live website.

The goal is to provide customers with the ease of booking in minimal time. Expensive deals tend to offer unique services such as pools, patio, or gardens, so some might be looking for such a deal.

We can also expand the deal types' categories to cater to a higher range of prices in bigger cities. We can also use a time-series analysis of comprehensive yearly data to study how people price their listings throughout the year.

For any questions or feedback, please reach out to me at safi.u.khan@outlook.com.

Note: The notebooks are very detailed and has all the required information regarding the code, data wrangling and modeling work.

Thank you for reading!

Safi Khan