**Aim and Objectives**

* The first aim of the assignment is to observe how the price of an AirBnb listing in the US differs with location. Location here is not used only in the narrow term of a country or a city, but also includes the specific neighbourhood from which the listing was posted. While it is quite clear that rents in general differ even by neighbourhood, this assignment specifically looks at AirBnb listings only, since these are usually less expensive when compared with a traditional rented apartment. This is to be observed through the correlation matrix and visualizations.
* By this analysis, it is to be determined how much the location of a listing affects its price if it does at all, and most importantly the role of other features quantified - such as the extent to which people can be accommodated, the type of property, the number of beds and so on in determining the price. Through the model, we also examine how best the price can be determined with the least amount of information needed. This is to be done by applying principal component analysis to the data, as well as observing the correlation matrix. The best regression model is also to be determined in each case - regular and the feature-reduced data.
* Further, observations are to be made on the type of property that the average user/customer would be interested in, and the city in which these prospective users would look for an AirBnb the most. While this may or may not affect the price directly, it would certainly be of use for hosts with listings in multiple cities. This would allow hosts to tailor their future listings accordingly. This observation is to be made solely through visualizations.

**Literature Review**

The literature available on this particular problem statement was mostly restricted to finding the factors that affected the price of a listing, and not specifically tailored around its location. Of particular interest was Scalable Prediction Models for Airbnb Listing in Spark Big Data Cluster3, and redictive Price Modeling for Airbnb listings1. Scalable Price Prediction Models of Hosting Business Levaraging Big Data2 was quite similar to Scalable Price Prediction Models of Hosting Business Levaraging Big Data with GPU, but lacked enough analysis, and was focussed more on accuracy. Both papers (references [1] and [3]) looked at the data from a similar standpoint - i.e how multiple factors affected the price of a listing, but Karkala [1] was an approach aimed at deployment.

Both these papers focussed on the RMSE as the major parameter for evaluating the model. Both of them split the data into 0.8 training, but Karkala [1] also had a validation set defined, with 0.1. The RMSE on the Random Forest Classifier showed 37.4 with Muralidharan et al. [3], while it was slightly better with Karkala with the median at around 30. While Muralidharan et al. implemented XGBoost with both CPU and GPU, only the CPU metric will be considered for consistency. They concluded that XGBoost was most optimal for the price prediction model, also considering that it took the least amount of time to run (60s) when compared with the other regression models. The RMSE for Muralidharan et al. was around 36, while the median RMSE for XGBoost stood at around 30 for Karkala. While Karkala has specified and demonstrated the features removed, this was not properly mentioned by Muralidharan et al. Considering that the latter’s work is mainly aimed at academia, it must be presumed that more features were left untouched with them, when compared with Karkala’s work, which was aimed at production. Generally, it is hence followable that Karkala’s RMSE values were in general lesser than Muralidharan et al. 's due to the fewer number of dimensions in the data. This trend is followed with the Decision Tree Regressor too.

Karakala’s encoding and visualizations showed that there was very little correlation between the city/neighbourhood of a listing, and its price. Muralidharan et al. ‘s work does not touch upon these. Karkala demonstrated that the number of beds and bathrooms, amenities available, type of property and so on were much more important factors in determining the price of a listing. Karakala’s work is more applicable in a real-life environment, since it was done with deployment in view. It hence offers more to the eye than just the accuracy of different models, which have only resource-based restrictions and not analytical ones.

**Methodology**

A variety of data preprocessing steps were used in the preparation of the dataset prior to feature engineering. There were originally 100 features in the dataset, and 134545 rows. Since a lot of the features involved descriptions of the listing and URLs, these were apt to be removed. The dataset was manually examined to see which of the categorical features were to be retained based intuitively on how they could affect the price of the listing. The other features with datatype `object` were removed. The data was then examined manually again to see if any numerical features were left which wouldn’t be of use, and they were also dropped. Next was the handling of null values. While there were some columns with a major part of the entries being NaN, there were some that had around a quarter of entried being null. It was hence decided that any feature with a threshold of null values beyond 25% of the dataset would be dropped. This removed most features from the dataset, and dropping the null rows left us with around 85,000 rows and 35 columns to work with.

The categorical features were then ordinally encoded. While this potentially has a chance of introducing bias into the dataset, it is seen later during visualization that no bias had crept in with regard to the location and the price of the listing. OneHotEncoding was not performed to avoid the dimensionality of the data getting too big.

The data was split into 70:30 training and testing, and standardization and PCA was then performed on the dataset, and the 20 most important features were extracted. The encoded data without PCA was also kept for comparison. The data was then checked for bias with respect to the location with the price of the listing. Reproducibility was maintained throughout.

Linear regression, regularization (L1, L2, L3), Linear support vector regressor, DT, RF and GB were the regression models used on the encoded dataset. These were applied on all 3 datasets - the regular encoded data, standardized data, and PCA reduced data. The model’s residual errors and scatterplots were also made in the case of the linear regression models. Kernel SVM could have also been performed, but the task was computationally too intensive given the dataset.

**Conclusion**

The major conclusions of the model were in line with Karkala [1]’s analysis. The city and the neighbourhood area of a listing scarcely affects the price of it, while the specifics of the accommodation surrounding the availability of basic needs affected it much more. With the possibility of bias having crept in due to the ordinal encoding of categorical data, this was checked with a boxplot of the 15 most occurring cities on the y-axis, with `price` on the x-axis. The trend showed no specific bias - in line with its correlation with the price. However, although the city and the neighbourhood per se did not affect the price of a listing much, it was observed that the review score of the location of the listing affected the price by around 11%. This means that the quality of the location is more important to a prospective customer than the exact latitude and longitude of it, and they are willing to travel distances to their places of commitment, as long as the previous users rate the location of the listing highly.

During visualization, boxplots were made of `beds` (one of the highly correlated features with `price`) and `city` with `price`. While `beds` showed a clear positive and high correlation as expected, `city` showed almost none at all at around 0.04. The features that most affected the `price` were `accommodates`, `bedrooms`, `bathrooms`, and `guests\_included`. The nature and flexibility of the cancellation policy also played a major role in determining the `price`. As mentioned before, the review score of the location too slightly affected the price, almost as much as the extent to which extra people can be accommodated in the listing. As far as the regression models are concerned, the gradient boosting regressor performed the best among the others, while random forest follows next. Gradient boosting gave an R2 score of around 0.66 for both the regular and the standardized data, while it gave around 0.57 for the PCA-reduced data. The random forest regressor gave around 0.57 for the regular and standardized data, while it gave ~0.51 for the PCA-reduced data. This shows that the dropping of features, although barely correlated with the price, did affect the R2 score by around 0.06-0.9. If this is a compromise that is willing to be made in order to allow fewer pieces of information to be needed in determining the price of a listing (and a slightly faster model runtime), then these features can be done away with, and the PCA-reduced model can be used.

From the visualizations made, it can be observed that the number of listings for `Entire home/apt` is much higher than a private room. This shows that hosts expect families or groups of people to be more likely candidates to look at listings on AirBnb, rather than a person or two looking for slightly less expensive accommodations. It is also observed that Los Angeles, Nashville and New Orleans are the cities having the most number of listings on AirBnb. It can be concluded from these observations, that hosts might benefit more in these cities, and especially so if they have listings where the entire home or apartment is available for accommodation. From the correlation matrix, it is also seen that the availability of the listing all 365 days a year slightly favours the hosts more.

Given more computational resources with regards to memory, kernel SVM could have also been implemented. Although polynomial regression has usually never worked well in a real-life scenario, it could be interesting to see how a polynomial curve of higher degrees (say 4 or 5) could fit the data without overfitting it. A more interesting study for the future would be to use the same dataset, but with price ranges instead (Putting the prices into bins). This would make it a multi-class classification problem, but it would mean slightly greater accuracy since the user may have a lesser chance of an outlier prediction, and might have a better idea about the price of a listing, given its range (rather than an exact value in the case of regression, which can sometimes be well off).

**References**

[1] Predictive Price Modeling for Airbnb listings. (n.d.). www.deepakkarkala.com. https://www.deepakkarkala.com/docs/articles/machine\_learning/airbnb\_price\_modeling/about/index.html

[2] Scalable Price Prediction Models of Hosting Business Levaraging Big Data with GPU. www.calstatela.edu.

<https://www.calstatela.edu/sites/default/files/airbnbraysamyukthaapic-ist_2022.pdf>

[3] Muralidharan, S., Yadav, S., Huh, J., Lee, S. and Woo, J. (2022). Scalable Prediction Models for Airbnb Listing in Spark Big Data Cluster using GPU-accelerated RAPIDS. www.jicce.org, [online] 20(2), pp.96–102. doi:https://doi.org/10.6109/jicce.2022.20.2.96.

‌[4] Coursehero.com. (2023). Available at: https://www.coursehero.com/file/162070317/Project-Report-Airbnb-Listingsdocx/ [Accessed 25 Oct. 2023].