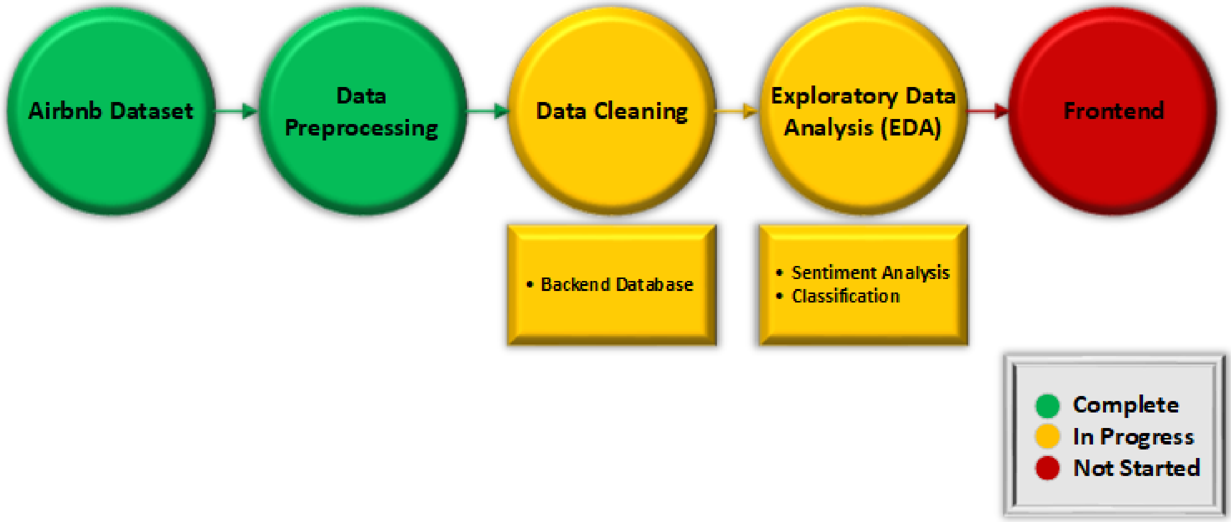
# Meep Meep Go NLP - Sentiment Analysis (Progress Report)

Problem Statement – Airbnb is the leading and rapidly growing alternative to the traditional hotel networks.  Travelers always hop on the websites to find location, neighborhood, and top stay places for their vacation.  To make their much-deserved vacation a good experience stay is a very integral part and hence the travelers spend a large amount of time going over the reviews for the listings.

| First Name | Last Last | NetID | Position |
|---|---|---|---|
| Joel | Rosales | joelar2 | Captain |
| Abhinav | Abhay | aabhay3 | |
| Syed | Peerzada | syedp3 | |
| Prasanna | Muralimanohar | pkm7 | |
| Piyush | Gambhir | piyushg2 | |



## 1. Which tasks have been completed?

We have completed enough research to progress to the end of the project but additional tool familiarity will be required when attempting to create the frontend for our project.  We performed data preprocessing and cleaning in Python, primarily using Pandas within local Jupyter Notebooks, and also OpenRefine.  These steps involved the following operations on the "comments" column of our dataset:  trimmed leading and trailing whitespace, collapsed consecutive whitespace, replaced smart quotes with ascii, changed all values to string, removed lower case, tokenized text and removed punctuation, removed '</br>' from text, removed words with numbers, removed stop words and empty tokens, and performed stemming.  While we had our dataset in OpenRefine, we also decided to accomplish clustering utilizing key collision with fingerprint, ngram, and metaphone3 algorithms to get a general idea of the words in the "comments" column that we could use for positive and negative sentiment analysis.

Concurrently working on our backend solution, we explored SQLite to load our airbnb dataset locally. The use of SQLite for our database would introduce issues hosting our cleaned dataset. Initial tests had our cleaned dataset seating over 150MB. We decided a remedy would be to provide a public link to our cleaned dataset using the University of Illinois Urbana-Champaign's Box application for a final cleaned dataset deliverable. Using Python, we then created the SQLite database and populated the database with our cleaned dataset. Lastly, we tested and readied the database for connection to a frontend.

Pivoting over to the Classification / Sentiment Analysis step of our project, we initiated Exploratory Data Analysis (EDA) by considering positive or negative sentiments based on the review rating score. While preprocessing the data using Python, we also utilized Word Cloud to explore common word distributions to feed the positive or negative word lists for training our classification models. Utilizing the NLTK library for Python, we accomplished some basic feature engineering by creating doc2vec columns and added TF-IDF columns. This was used to predict positive versus negative reviews using a Random Forest Classifier. We then evaluated our Random Forest Classifier by plotting a Receiver Operating Characteristics (ROC) curve.

We also implemented the Natural Language Processing Vader algorithm for Sentiment Analysis and plotted the output of the positive and negative sentiments. From this result, it seems our dataset is skewed by mostly positive reviews. We then also created the following classifiers, Naive Bayes, SVM, and a Confusion Matrix. These classifiers were evaluated using Precision, Recall, and an F-1 Score. We then began plotting the spread of positive and negative reviews using GeoPandas. Retrieving the shape-file(.shp) that contains the geo spatial data for a city, we used the latitude and longitude attributes of the initial data to highlight areas with high positive and high negative reviews.

**2. Which tasks are pending?**

We need to accomplish a little more data cleaning to get rid of non-english words and characters. When geo-spatial plotting, we need to restrict to within a city or remove multi-city datasets. More metrics need to be retrieved out of Sentiment Analysis such for plotting precision-recall curves. We also need to refine model parameters to get a higher ROC curve for our classifiers. We then need to begin developing the Front End so that we can build our visuals from the SQLite database. Lastly, we need to trim any additional steps we accomplished to aggregate and finalize our documentation and code for submission.

**3. Are you facing any challenges?**

The JSONs from OpenRefine were exported in an non-standard manner and it seems Pandas is not able to read them as a dictionary. Due to the amount of similarities found while clustering in OpenRefine, it has been difficult to extract recurring words in an automated manner. There are non-english words and characters in the Airbnb dataset that we still need to figure out how to remove in an efficient manner. The Airbnb dataset also seems to contain mostly positive reviews for our dataset skewing the results meaning we may need to train our models on a more evenly distributed dataset. We attempted to incorporate Hawaii since it is a highly visited destination for vacations but Airbnb mixes this dataset with all islands instead of per city which was caught early and removed. We did not catch the Jersey dataset which has similar issues to Hawaii and made it into our GeoPandas step which caused issues. We could implement data preprocessing steps to handle the multi-city datasets but we have enough data to accomplish our use case; we could explore this option if time allows.