**Problem**

The premise is simple, by using the search data of a user and the hotel a user booked in the past, a ranked list of hotels is needed where the most appropriate one appears at the top of the list.

As the name of the problem suggests, this is a Learning-To-Rank(LTR) problem. And I instantly thought of using the XGBoost library.

**Plan**

There are several steps I took to solve this problem:

* EDA(Data Analysis)
* Data Preprocessing
* Tuning the Model
* Testing

**Execution**

The EDA revealed that some variables were mostly empty and that not all columns are actually useful. Along with revealing that some of the data was corrupted, for example there were negative values in the “prop\_room\_capacity” variable. Furthermore a lot of the variables were categorical.

Categorical variables needed encoding, after initially choosing to do One-Hot-Encoding, realising that I didn’t have the time to wait an hour for a model to train due to the immense number of columns made. I chose to use Label Encoding to make the dataset numerical.

Bayesian Optimisation and Grid Search methods were used to tune the Hyperparameters for the XGboost Ranking algorithm. And cross validation was used to test the model.

Then the best model is chosen to provide rankings to the test dataset from Kaggle.

Name on Kaggle Leaderboard: Srinivas Billa