Group Name: CATS (Categorical Analytics Squad)
Member Name(s): Abhimanyu Choudhary, Ezra Max, Hao Zhu, Shiyu Tian

Descriptions and Goals:

Idea: an interactive program that asks the user questions about what sort of vacation they would like, and then proceeds to the next best question (the one that narrows the options down the most at each step) until it returns a list of vacation ideas and possible itineraries. For example, the program might begin by asking the user if he/she wanted to travel to a warm destination. The user would then have five choices:
Definitely Not
Probably Not
Not Sure/Indifferent
Probably
Definitely
Based on their answer to this question, we would then ask them another question, and continue to a fixed depth, or until we reach a node in our decision tree. We would then point users to the resources we accessed about the location, so that they might learn more on their own.

Data types and sources:

-Safety (CIA world factbook/travel advisories)->tranche by advisory level, exclude all locations that have level 3 or level 4 advisories unless explictly disabled
-Weather (https://openweathermap.org/api)->tranche into warm, tropical, cold, etc.
-Language (country-level)->tranche by % english speakers
-Nearest airport (Lonely planet) and flight times/costs (Skyskanner API)
-Cost of accomodations (Hotel types, prices)
-Activities and key words (maybe keywords from Tripadvisor) For example, things like: nightlife, sports activities (snowboarding, skiing, surfing, swimming), natural attractions etc.
BIG QUESTION: HOW TO GET KEYWORDS -
For example, if someone says they want to see places with history, and online descriptions state that a place is "historic", how might we get that place to show up?
-Lots of people considering? (From google trends)->tranche by popularity

What project entails:
Setting up data structure (nodes etc.) and order of questions to go through nodes
Scraping the data and adding to consistent data structures (nodes etc.)
Cleaning the data
Building a user interface
Possibly implementing certain machine learning techniques (for example, decision trees).
We are not sure how difficult this would be to implement in the time allotted.

Timeline: (Done by)
-Plan presentation (1/31)
-Look at all types of data that we get from sources above, including a) detailed info on scraping for relevant variable in everything but keywords and b) type of variable returned by APIs (2/1)
-Decide on format of data structure to store these data (what levels correspond to what in our nodes? Implicitly, what is order of questions to be asked?) (2/3)
-Find plan to clean, import data, record linkage etc. so that it can be efficiently and consistently stored (most likely into CSVs) (2/3)
-Import data on 20 cities to start (2/8)
-Set up interface (non-graphic part) (2/14)
-Cleaning data, and importing data on 500 cities and testing speed of interface. Make corrections to improve efficiency (2/30)
-Try to set up graphic interface (3/5)
-Finalize presentation write-up (3/10)

Extensions/extra features that would be nice
-Food and drink data (Yelp Open Dataset?)
-Machine Learning process to train for better questions and understanding answers. Specifically, implement some sort of decision tree, and "information-type" criterion to try and determine the best way to proceed down the tree.

Extra Resources:
References for Machine Learning Aspect:
Creating the self-learning expert system for solving problems with fuzzy logic:
https://iopscience.iop.org/article/10.1088/1742-6596/1368/5/052015/pdf

On playing 20 questions with a liar:
https://dl.acm.org/doi/10.5555/139404.139409

Decision-Tree Construction
http://fastml.com/how-a-russian-mathematician-constructed-a-decision-tree-by-hand-to-solve-a-medical-problem/
find an expert and imitate his/ her questions