ENSF 611 – Winter 2023 - Final Project Proposal

Name: Michael Le

UCID: 10104883

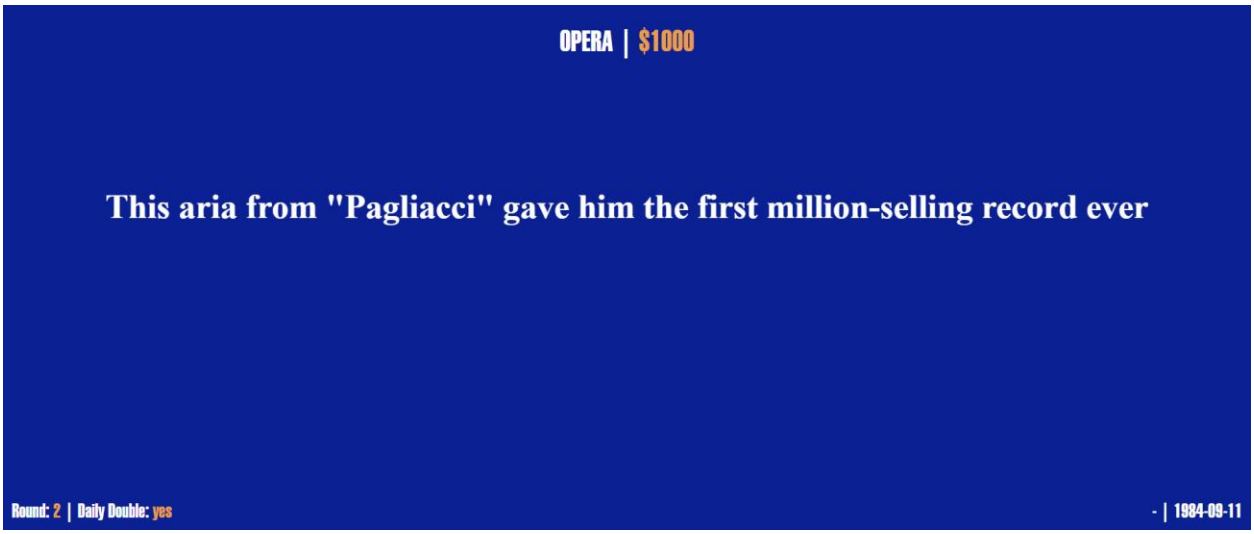
Project Motivation (The Why?): Question and Topic Investigated

Question: Using the clustering models learned in ENSF 611, experiment and explore how effective each of them are in clustering clues from the game show Jeopardy! based on the keywords used in each clue. If the models learned in class are ineffective, see if there are any models that do work.

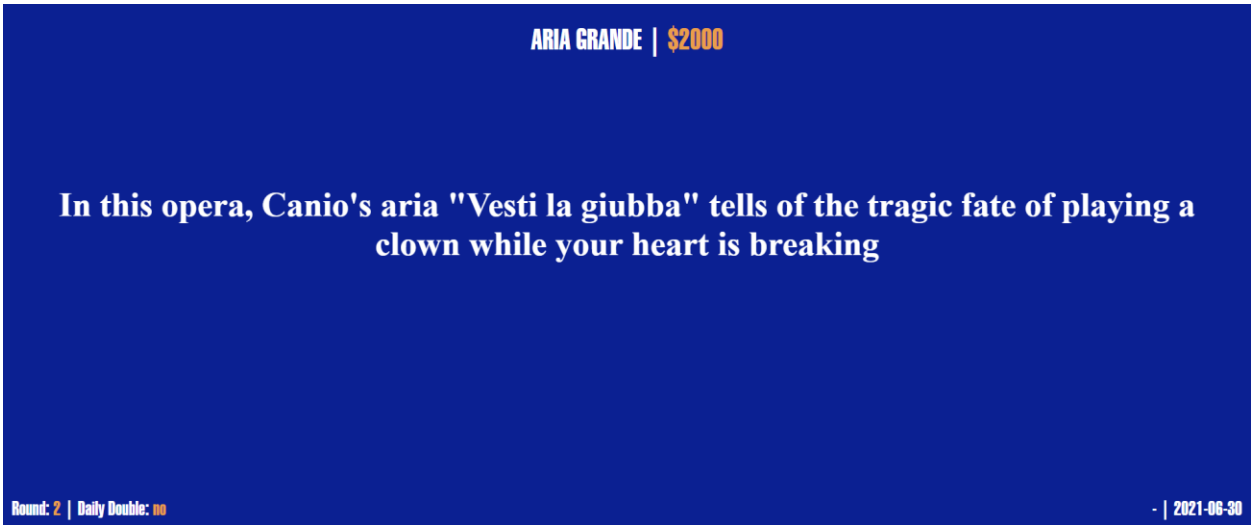
Motivation: As a viewer of the game show Jeopardy!, I have always been curious as to which categories and topics appear the most frequently in clues that appear on the show.

Looking at the fan-created archive website J! Archive (https://j-archive.com/), which contains a database of all the clues aired since the debut of the show in 1984, I noticed a definite shift in the writing style of the clues and category names from then to now:

Example: Clue from September 1984



Example: Clue from June 2021



Both of these clues are opera clues. In the 1984 clue example, this is obvious since the category at the top is shown to be "Opera". However, in the 2021 clue example this is not as clear, since the category is shown as "Aria Grande" – this is a play on the name of the pop singer Ariana Grande and also references the opera term "Aria", which is self-contained opera piece for one voice. The category name hints at indicates multiple possibilities for what the clue may be about. The clue content in the 2021 example is also less straightforward and contains trickier writing, making the answer harder to find.

The main point is that the category and content of the clues that appear on the current episodes of Jeopardy! are less obvious and more subtle. Therefore, choosing to separate clues based solely on category name may not be the most reliable way to find out which topics appear the most frequently in clues.

Instead, what if we clustered clues based on keywords used in the clue instead? Would this improve the chances of discovering distinct topics for the clues?

Expectations: As someone who is new to the topic of Machine Learning, I definitely do not expect the clustering models I choose to produce worthwhile results, especially since the clue language is so complex and tricky. This is more of an experimental project for my own curiosity to see first-hand how effective models can be with real-world data.

Project Plan (The How?): Research and Workflow Plan

1. Download a dataset of all Jeopardy! clues from September 1984 – June 2021 on J! Archive (https://j-archive.com/). No web scraping will be needed since lots of people have already scraped the clue data and converted them CSV files.
2. Upload three versions of the clues CSV file into a Jupyter Notebook:
    a. A CSV file that contains all clues from 1984-2021
    b. A CSV file that contains just clues from 1984-1995
    c. A CSV file that contains just clues 2010-2021

    The logic behind using these three datasets is to see if the clustering is different for each one. My hypothesis is that the clustering of clues from 1984-1995 will be more effective than the other two datasets since the clue writing is more simplistic.
3. For each dataset, convert them to a pandas DataFrame and perform:
    a. Exploratory Data Analysis
    b. Data Cleaning
    c. Pre-Processing: Eliminate HTML tags and special characters in the clues (if there are any), convert all words to lower case, and create a new column for these cleaned clues (which will be used for the models)
    d. Vectorize questions and assign a weight to each word in the cleaned clue using either TF-IDF (Term Frequency-Inverse Document Frequency) or CountVectorizer. Do more research on these two methods: use CountVectorizer if I think the

common words in each clue matters. Use TF-IDF to add more weight to unique words if I think the common words are less important.

e. Cluster clues using K-Means, Gaussian Mixture Model, Agglomerative Clustering, and DBSCAN. Experiment and find an ideal number of clusters for each (if any).
f. Analyze, interpret and visualize results (if possible). If none of the above clustering models work, see if there are models that may work better. Suggestion: Doc2Vec (https://medium.com/@ermolushka/text-clusterization-using-python-and-doc2vec-8c499668fa61)

Project Resources (The What?): Dataset, Model, Tools/Framework, Components

1. Clue Datasets: Download datasets that other people have scraped from J! Archive (https://j-archive.com/)
2. Tools: Anaconda, Jupyter Notebook, Python, GitHub, pandas, matplotlib, Seaborn
3. Scikit-Learn Models: KMeans, GaussianMixture, AgglomerativeClustering, DBSCAN
4. Optional Models (if time allows): Doc2Vec from genism library
5. Resources on how to use models:
    a. https://medium.com/@ermolushka/text-clusterization-using-python-and-doc2vec-8c499668fa61
    b. https://scikit-learn.org/
    c. Links to any tutorials or similar projects I come across Google that help explain how to cluster the data.