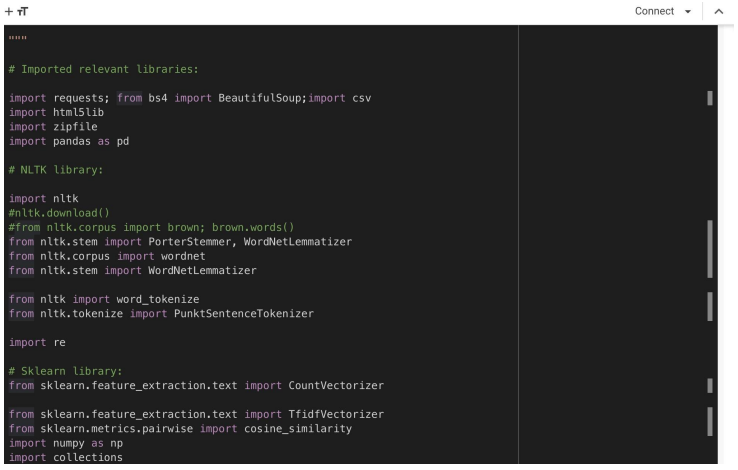
Natural Language Processing(NLP) Applied to Slang Language Over the Decades:

Project Description:

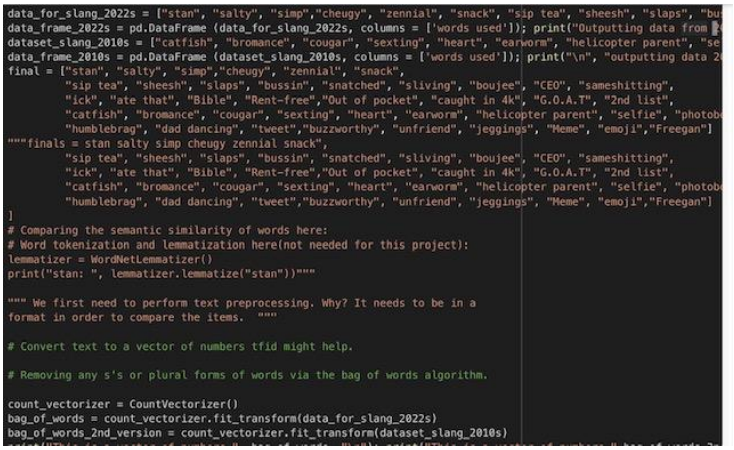
The project involved Natural Language Processing(NLP) and utilized the Python programming language and its various built-in libraries. NLP is a field that is part of the areas of Artificial Intelligence, Linguistics and Computer Science. NLP mainly focuses on how computers process language; this can involve processing large volumes of data. There are several phases of NLP. These phases are as follows: lexical analysis, parsing, semantic analysis, discourse integration, and pragmaticanalysis. Due to time and knowledge constraints, we kept it relatively simple. The project used built-in libraries.



Actual Project:

The datasets were individually coded and stored in the form of a list. During the planning portion of the project, I had trouble finding enough relevant datasets.
There was also a time constraint. This also meant that I did not have as much time to preprocess the data.

Below you can see an example of the code:



```
data_for_slang_2022s = ["stan", "salty", "simp","cheugy", "zennial", "snack", "sip tea", "sheesh", "slaps", "bu
data_frame_2022s = pd.DataFrame (data_for_slang_2022s, columns = ['words used']); print("Outputting data from
dataset_slang_2010s = ["catfish", "bromance", "cougar", "sexting", "heart", "earworm", "helicopter parent", "se
data_frame_2010s = pd.DataFrame (dataset_slang_2010s, columns = ['words used']); print("\n", "outputting data 2
final = ["stan", "salty", "simp","cheugy", "zennial", "snack",
        "sip tea", "sheesh", "slaps", "bussin", "snatched", "sliving", "boujee", "CEO", "sameshitting",
        "ick", "ate that", "Bible", "Rent-free","Out of pocket", "caught in 4k", "G.O.A.T", "2nd list",
        "catfish", "bromance", "cougar", "sexting", "heart", "earworm", "helicopter parent", "selfie", "photob
        "humblebrag", "dad dancing", "tweet","buzzworthy", "unfriend", "jeggings", "Meme", "emoji","Freegan"]
"""finals = stan salty simp cheugy zennial snack",
        "sip tea", "sheesh", "slaps", "bussin", "snatched", "sliving", "boujee", "CEO", "sameshitting",
        "ick", "ate that", "Bible", "Rent-free","Out of pocket", "caught in 4k", "G.O.A.T", "2nd list",
        "catfish", "bromance", "cougar", "sexting", "heart", "earworm", "helicopter parent", "selfie", "photob
        "humblebrag", "dad dancing", "tweet","buzzworthy", "unfriend", "jeggings", "Meme", "emoji","Freegan"]
]
# Comparing the semantic similarity of words here:
# Word tokenization and lemmatization here(not needed for this project):
lemmatizer = WordNetLemmatizer()
print("stan: ", lemmatizer.lemmatize("stan"))"""

""" We first need to perform text preprocessing. Why? It needs to be in a
format in order to compare the items.  """

# Convert text to a vector of numbers tfid might help.

# Removing any s's or plural forms of words via the bag of words algorithm.

count_vectorizer = CountVectorizer()
bag_of_words = count_vectorizer.fit_transform(data_for_slang_2022s)
bag_of_words_2nd_version = count_vectorizer.fit_transform(dataset_slang_2010s)
```

The project quickly became a solo project. Overall, after finding the right resources, the project went rather well. Eventually, it came time to present my findings at Code2gether's showcase.

Project Outcome:

This project ended up winning in the category of "Most Unique."