507 Final Project Document

Xinran Wang

Project Code: https://github.com/Xinranwxrtzxy/507-Final-Project/blob/main/final%20project.py

Required packages:

```
from cgitb import small
import pandas as pd
from bs4 import BeautifulSoup
import wikipedia
import lxml
import re
import os
from googleapiclient.discovery import build
from copy import deepcopy
import webbrowser
```

Data Sources:

1. Wikipedia: use API to collect summaries and URLs of 27 dog breeds and use web scraping to collect six attributes including breed *name, summary, origin, height, weight, life span, color* from Wikipedia data. API key is not needed to use this API: https://pypi.org/project/Wikipedia-API/; use beautifulsoup library to scrape web content: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
2. YouTube Data API v3: use API to collect the URLs and other information of top 25 popular videos of each dog breeds. Attributes including *urls, title, description* are collected and stored. Following instructions from the below website to obtain API key and change the *api_key* variable value in the program: https://developers.google.com/youtube/v3/getting-started?hl=en_US
3. The data are collected using web-scraping, Wikipedia API and YouTube Data API v3, and the I used cache to store the data collected using YouTube API by writing data to local JSON file. The cleaned combined data is of JSON format, named : https://github.com/Xinranwxrtzxy/507-Final-Project/blob/main/dogsInfo.json

4. Data Summary: my data collected from Wiki include breed name, summary, origin, height, weight, life span, there are 27 breeds in total for now, users can manually add names and ids of other dog's breed to the program to include more dogs. Data from YouTube include URLs, titles, and descriptions of 25 top view-counted videos with breed name as the key words.

Data Structure:

1. README: The README.md file store the information about how to apply for YouTube API key, data structure, and how to use the program. The data are stored in tree structure. The first node divides 27 dog breeds into 12 big dogs (left leaf) and 15 small dogs (right leaf). The big dog's node has 5 dogs from Europe as left leaf and 7 dogs from other places as right leaf. The small dogs' node has 7 dogs from Europe as left leaf and 5 dogs from other places as right leaf.
2. Final_project.py is the full program including access to data using API, storing data using caching, and running the main program. User can access and store the data from YouTube and Wikipedia, combine data into a dictionary. Then create a class structure named *Dogs* to store the data, which has 12 attributes. Each class object has 25 elements for *urls, title, description* and each one is information about one video of that class object. Another attribute *question* is used to store question that will ask the user to choose between two leaves of a node in next step. Then divide the 27 class objects into six groups according to their size and origin, to prepare for creating tree structure. Then construct a tree structure to store the data stored using classes. Finally, users can run the main program to take the quiz to search for a dog and look at related information. The screenshots of the class structure and tree structure are as below:

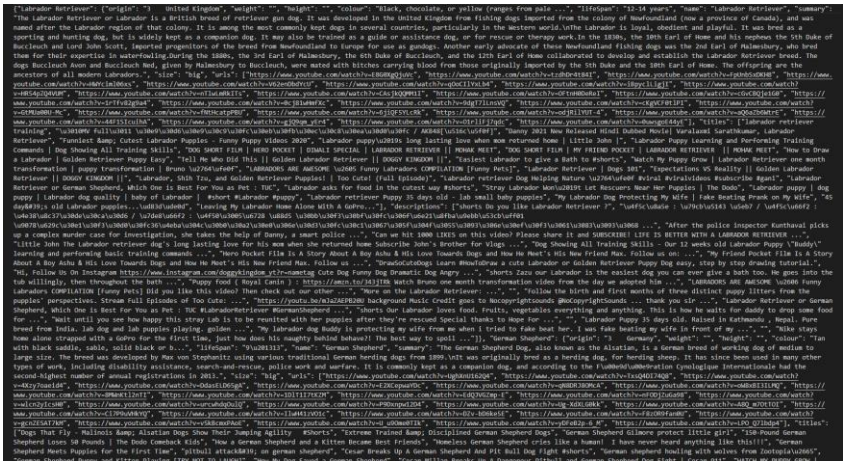
```
# The first node and it's left and right leaf
root = Node(dogsClass)
BigDogs=Node(bigDogs)
SmallDogs=Node(smallDogs)

# The left leaf also has left and right leaf
root.insert_left(BigDogs)
BigDogs.insert_left(Node(EUbigDogs))
BigDogs.insert_right(Node(NonEUbigDogs))

# The right leaf also has left and right leaf
root.insert_right(SmallDogs)
SmallDogs.insert_left(Node(EUsmallDogs))
SmallDogs.insert_right(Node(NonEUsmallDogs))
```

```
class Dogs:
    """
    Create class structure
    """
    def __init__(self, json):
        self.name=json["name"]
        self.summary=json["summary"]
        self.origin=json["origin"]
        self.weight=json["weight"]
        self.height=json["height"]
        self.colour=json["colour"]
        self.lifeSpan=json["lifeSpan"]
        self.size=json["size"]
        self.urls=json["urls"]
        self.titles=json["titles"]
        self.descriptions=json["descriptions"]
        self.question=None
```

3. The readJSON.py file reads the json file of my trees and run the program. Firstly, read json file to json object, store data in class objects, divide the objects to different groups, and create a tree structure to store the data. Users can also run the main program using this file. But this program should be used with tree.json. The screenshot of the tree.json file is as follows:



Interaction and Presentation Options:

1. The program helps user to find out what breed of dog fit them well by answering several questions. There are two questions in sequence: “Do you prefer big dogs than small dogs?” and “Do you prefer dogs originally from Europe or other places?” User can type yes or no to answer the question.

2. After typing the answer to both questions, the user will be provided with names of dog breeds recommended for them.
3. They will be able to explore more detailed information of a specific dog breed among these dogs. They can choose a dog and view the summary, body size statistics, origin, colour, and life span.
4. Then the system asks if they want to see videos of this dog on YouTube. If yes, they will be provided with a list of titles of popular dog videos of this breed at first. Then they can type the index of video to see descriptions and urls. They are able to further choose to open the url in browser.
5. After that, the users are asked if they want to choose a new dog by answering the two questions again. If no, the program will end.

Demo Video Link: https://youtu.be/TOo-ll4ITec