Progress Report II: Are Petfinder's Breed Classifications Visually Subjective? Analyzing the Results of a Purebred-Trained Dog Breed Image Classification Model on a Mixed-Breed Petfinder Dataset

Allison Aprile, April 2019

Abstract

After overcoming several setbacks in the first week, the past few days were dedicated to regaining stability on the project. However, the initial steps of data-processing revealed a large issue in my custom dataset. This required the repetition of data collection and cleaning, as well as patience and perseverance. This document reports on the progress from the period of **April 23rd-29th, 2019**.

Work Proposed

The beginning of the week was to be used creating visuals, practicing, and addressing my progress to the class during the Lighting Talks. In the proposal, I had projected to spend the rest of work period modifying and running the CNN model on the testing data. After last week's obstacles, I had revised the plan to complete the data annotation process and beginning to focus on selecting and modifying a model. This was planned to regain control of my project and begin preparations for the final analysis.

Complications

In the previous progress report, I discussed how I implemented cURL commands to call the Petfinder API, specifying the animal type as 'dog' and setting the return limit to 100 elements to maximize the requests. This allowed me to collect 1,600 data points in about 20 minutes. Unfortunately, I had wrongly assumed that each request was dependent on the previous return—that is, that this method would yield 1,600 unique elements.

My first step in cleaning the data was to delete all of the rows with overflown description cells, as the text strings overwrote the necessary image links. This affected approximately 20 per 100 elements (20%). Because all of the animals were assumedly unique, I was alarmed when I began reading verbatim descriptions after the first 100 rows were cleaned. I revisited the DataFrame and ran the line: `dogs['id'].unique().size,` which, after cleaning, only returned 202.

Since this quantity could still build a sufficient testing dataset, I created a new DataFrame of the unique elements by calling `drop_duplicates()` and wrote it to a CSV file. I then began the annotation process (discussed in detail in **Work Complete**) under the impression that the new dataset would still have a variety of breeds. However, after distinguishing 64 Labrador Retrievers, I realized that I needed to establish a new dataset.

Because the Petfinder API efficiently returned clean, favorable data, I decided to use it once more for the raw collection. This time, I made sure to do adequate reading about cURL commands and RESTful APIs. My research confirmed that I had been very uninformed about the dependency of cURL requests. In hindsight, even without research, it was not sensible to believe that each request, which is made with individual calls on individual command lines, would be unique if I did not further specify unique parameters. It was simply careless to use identical code for each request and additionally to not compare results when merging the JSON files.

Before starting the new collection, I revisited the Petfinder API documentation to determine the best method to ensure unique data. I ran trials on different lines of code, manipulating combinations of the `sort, page,` and `limit` Query parameters. After about an hour of testing, I determined that I would maintain a limit of 100 returns (the maximum) but also include a page specification alongside such in each call. I initially went page-by-page, starting with one, then two, etcetera, yet still faced some minor duplication as well as limited breed variation.

Because I do not have extensive knowledge (or access to the knowledge) of the Petfinder Database architecture, I decided that randomly selecting 20 page numbers would be the least-biased method of specification. This would not only prevent duplication but ensure a wide collection of dogs as well.

Work Complete

Data Collection: Once more I implemented the Petfinder API to collect the raw data. I altered the command by specifying the page number in each request, as displayed below:

```
curl -H "Authorization: Bearer eyJ0eXA… "
https://api.petfinder.com/v2/animals\?type\=dog\&limit\=100\&page\=(page#) > dogs(call#).json
```

At the end of each request during the testing period (discussed in **Complications**), the output specified the database parameters, specifically the current page number out of the total. Therefore, for the actual data collection, I used Python to randomly generate 20 unique numbers in the range 1 to 26,422 (the total number of pages at the time of the requests). For each call, I replaced the (page#) portion of the above template with one of the numbers.

I collected 2,000 elements to ensure a unique selection of dog breeds as well as to account for any missing information or broken image links. The cURL collection took approximately 30 minutes and, after carefully examining the data, yielded excellent results.

Data Cleaning: I once more converted all of the JSON files to CSV using an online program. Before manually cleaning the data, I read the master CSV into a DataFrame and removed any duplicates. I was rejoiced to retrieve 1,992 unique elements. After performing some preliminary EDA, I learned that my dataset consisted of 123 different primary breeds, which fit perfectly with the label quantity of the purebred dataset. I wrote the improved DataFrame back to a CSV file and, after spending approximately two hours deleting empty and overflowing lines, remained with 1,898 elements.

As with last week's data collection, I read the cleaned dataset into another DataFrame and narrowed it down to four columns: ID, Primary Breed, Mixed Breed (Boolean), and Medium Photo (URL). When annotating last week, I realized that many breeds (including Dachshund and Australian Shepherd) did not belong to the Purebred labels list. Therefore, before selecting a subset for my testing dataset, I removed any elements that had labels without a match in the training dataset. Then, I randomly selected 150 random elements with the Mixed Breed = True filter and saved it into a separate DataFrame.

Model: Despite the issues with the original training dataset, I successfully selected a prebuilt Dog Breed Image Classification model, which can be found here. I chose this model because it was clearly described, seemed easy to manipulate and, as a plus, included a link to a cleaned, easier-to-download Stanford Dog Dataset. Although I have not yet tested it, I have downloaded and previewed the data.

Data Annotation: With the refined dataset in place, I began annotating. I created an CSV file that mimicked that provided in the model and entered the ID, Primary Breed, and Medium Photo URL columns for reference. Currently, I have translated all of the Primary Breed classifications into the corresponding labels, completing about half of the total annotation process.

Other: In addition to working on the project, this week I was responsible for creating a presentation to inform the class on my topic. I spent about two hours compiling a PowerPoint and another hour writing a script and practicing my presentation. On **Tuesday, April 23rd** I successfully addressed the class.

Time Approximation: 12 hours

Work Incomplete

Due to unexpected complications, I once more fell behind in my progress. This week, it is necessary that I finish the data annotation process by visiting each individual image link and saving them by their unique IDs to a folder. Next, I will need to modify the CNN model, training and testing it on the purebred dataset until I attain a desirable accuracy. Then, I will perform several tests on the model with the mixed-breed testing dataset. I intend to have this complete by **May 4th**.

The final project will summarize my findings regarding the visual subjectivity of the Petfinder breed classifications as well as the possible addition of a general Petfinder Exploratory Data Analysis. This is to be completed by **May 7th**, followed by an individual project discussion on **May 8th**.

Reflection

This past work period was filled with more unexpected setbacks and disappointments. My intended goals were compromised by my previous lack of attention, causing me to be more cautious going forward. In my experience, I was forced to learn efficient methods for cleaning the data and become more informed with cURL commands, which is ultimately beneficial to my skillset. Overall, recollecting the data was a large obstacle to this work period's progress but a crucial decision. Now, I feel more confident in the diversity of tests that my new dataset will provide and am excited to continue on the project.