**Logo Similarity Documentation**

**Task:**

Match and group websites by the similarity of their logos.

Context

Logos are instrumental for a company’s identity – they’re the symbol that customers use to recognize your brand. Ideally, you’ll want people to instantly connect the sight of your logo with the memory of what your company does – and, more importantly, how it makes them feel.

Guidelines

* Take the time to deeply understand the problem before writing code. Even the most sophisticated solution is ineffective if it solves the wrong problem. Misalignment in problem definition leads to incorrect conclusions and wasted effort.
* We know this is a clustering problem, you know this is a clustering problem, question is: can you do it without ML algorithms (like DBSCAN or k-means clustering)?
* Check whether the program correctly extracts the logo and matches them properly (as a human, you instantly recognize them, but this is way harder for a machine).
* Explore this from as many different angles as you can. It will generate valuable questions.
* From a tech stack perspective, you can use any programming language, toolset or libraries you’re comfortable with or find necessary, especially if you know it would be a better option or a more interesting one (we generally prefer Node, Python, Scala).
* At Veridion, we run similar algorithms on billions of records. While your solution doesn’t need to scale to that level, it would be impressive if it does. For now, however, what matters most is your approach to solving the problem—if your solution is exceptional for the given dataset, we trust that you can scale it effectively using the right tools.

Describing my approach and implementation

Short summary:My aproach of the task consists of checking the original input,converting the original input into a format which is easier to follow.It then attempts to webscrape the logos by any means necessary.After that it stores them in a database in order to classify them using different pretrained AI models.

For this project I used the Pycharm IDE due to the fact that I have received a free Jetbrains license from my university.We were also encouraged to use their IDEs in our classes so I have become used to them over time.

First step was saving the information from the logos\_list database in a format I was familiar with which I could easily visualize in my IDE.To do that I have converted it from a .parquet file to a .csv one.

I first checked the original file: A screen shot of a computer program

AI-generated content may be incorrect.

And then I did the conversion:

A black rectangle with white dots

AI-generated content may be incorrect.

The second step was webscraping the logos from each one of the domains and this is where I had to come up with an ingenious solution as I was unable to use the same web scraping aproach for each website:

My script works by first attempting to extract the logos from the company website using Selenium and BeautifulSoup

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

Many domains required preprocessing in order to be turned into accesible URLs

A screen shot of a computer code

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

The first attempt uses BeautifulSoup by looking for criteria that usually describe a logo in the websites codebase.If that doesn’t work it resorts to trying a similar aproach again but with Selenium

A computer screen shot of a program code

AI-generated content may be incorrect.

However,there are certains websites for which the way their logos are stored do not respect any general criteria,or they do not allow permission for my code to acces them,which lead me to the last resort aproach.This consists of extracting the company name from the domains(I have observed that all of the domains are written as companyname.something so I simply had to take the words before the “.”) and starting a Google query using the custom search API.It then saves the first image result it finds.Obviously this sometimes happens to not yield the most correct result which is why I have implemented it as a last resort. A black screen with colorful lines

AI-generated content may be incorrect.

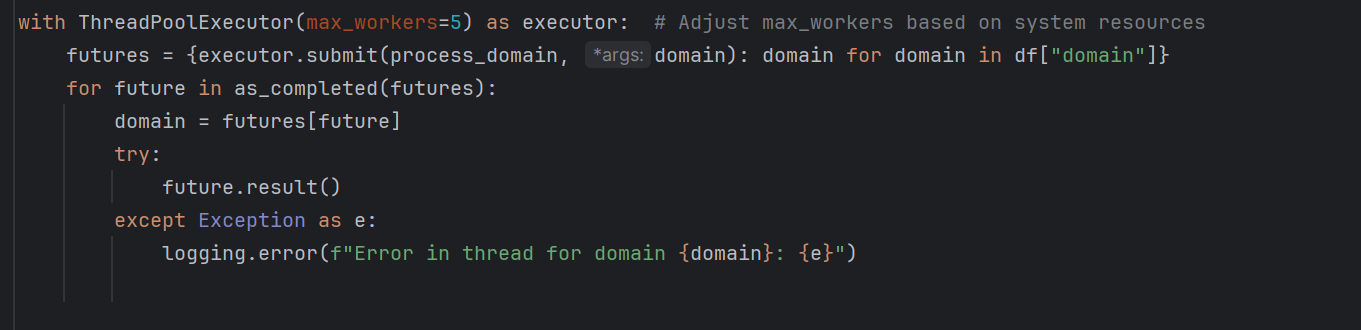
A black screen with white text

AI-generated content may be incorrect.

The combination of all these three ways of scraping logos ensures obtaining an image for every single one of the domains.However,there result is not always the correct one,from my testing I would say that my code has 98-99% percent chance of obtaining the correct result.

All of the logos are saved in a folder.I could have saved them directly in my SQL database but it was easier for me to follow how my code operates during development by seeing each image being inserted one by one in front of my eyes.

This script also uses ThreadPoolExecutor for faster execution



The third and final step was classifying logos based on their similarity as required.

First of all I connect to MySQL database and make sure it is correctly defined.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Then I insert the images:

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

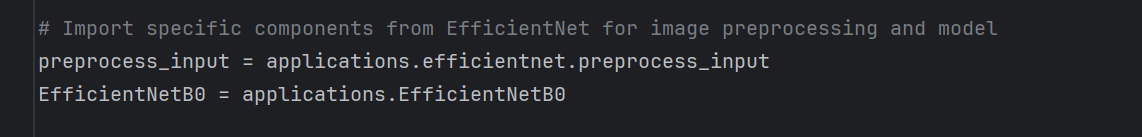
Secondly,I classify my images by using Tensorflow,PIL,Faiss and Sklearn.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.



This is the first function used in classifying the images A computer screen shot of a program code

AI-generated content may be incorrect.

It utilizes the pretrained EfficientNetB0.

A computer screen shot of a program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Then the function which uses the model listed above in order to extract the key features of each logo.

And finally clustering logos and obtaining the desired output.I have stored the results in bot a csv file and an HTML one.The latter is for easier visualization.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

Closing the connection with the database and calling the functions declared above in the main function.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Final mentions:

My script is not perfect,some issues which should be adressed in order to achieve more scalability include finding ways to bypass website restrictions,finding more efficient means of webscraping which would require less resources and finetuning an AI model in order to improve the results.