Project 3 - Small Search Engine

## Objectives:

* Task 1: Load Data and Index Values based on embedding
* Task 2: Get User Query and Find Embedding for Query
* Task 3: Find Closest Indexed Values to the Query using Cosine Similarity

## Implementation:

### Libraries

* transformers
* torch
* numpy
* scikit-learn

### Algorithm and Code

Functions:

embed\_text(text: str) -> np.ndarray:Generate embeddings for the given text using BERT.

index\_documents(path: str):Index text files in the given directory and store embeddings

search\_documents(query: str) -> List[Tuple[str, float]]: find top 5 most similar document

main(): Console interface for the search engine

## Results

\*All of the files were indexed

Query: “Porsche”

Results:

56-car.txt: 0.9289

57-cars.txt: 0.8724

19-cars.txt: 0.8235

14-cars.txt: 0.8171

24-cars.txt: 0.8132

Explanation: 56 and 57 cars have information about Porches. 57 is more specific to the 911

Query: “911”

Results:

57-cars.txt: 0.8349

56-car.txt: 0.7960

50-tech.txt: 0.7688

28-health.txt: 0.7636

21-cars.txt: 0.7628

Explanation: Since the 57-cars is more specific to the 911 model it’s ranked higher correctly

Query: "Vehicle Engine”

Results:

8-cars.txt: 0.8716

19-cars.txt: 0.8695

14-cars.txt: 0.8670

9-cars.txt: 0.8622

47-cars.txt: 0.8614

Explanation: Cars semantically appear

Query: “Relationship”

Result:

51-love.txt: 0.8480

34-health.txt: 0.8190

45-health.txt: 0.8046

36-health.txt: 0.8045

44-sports.txt: 0.7966

Explanation: The file with the description of love has the highest score

Query: “how to make my home cooler”

Result:

52-ac.txt: 0.8012

55-weather.txt: 0.7659

43-tech.txt: 0.7460

42-health.txt: 0.7452

40-travel.txt: 0.7449

Explanation: The text talking about fixing the AC and weather description came on top