**DATA 240: DATA MINING & ANALYTICS**

**SAN JOSE STATE UNIVERSITY**

**“Carbon Footprint Estimation: Leveraging Data Mining Techniques”**

**Project By Team 2**

**Sachin Kumar Srinivasa Murthy (016594773)**

**Sangamithra Murugesan (015357563)**

**Aayush Srivastava (016652727)**

**Motivation:**

The global movement towards carbon neutrality serves as a fundamental source of inspiration for this project. The Sustainable Development Goals of the United Nations places a high premium on countries and industries to reduce their carbon emissions.The announcement made by CEO Tim Cook at Apple's 2023 product launch event to make every product carbon-neutral by 2030 acted as a major impetus for this global trend.The market for carbon-neutral products is expanding, but buyers are not well-equipped to determine how much of a carbon footprint their purchases will actually cause. The project recognizes the importance of developing a resource that is both accurate and efficient in informing customers about the CO2 emissions associated with each dollar spent on products , hence we aim to enhance the efficiency of the existing work [1] by improving accuracy and reducing time complexity.

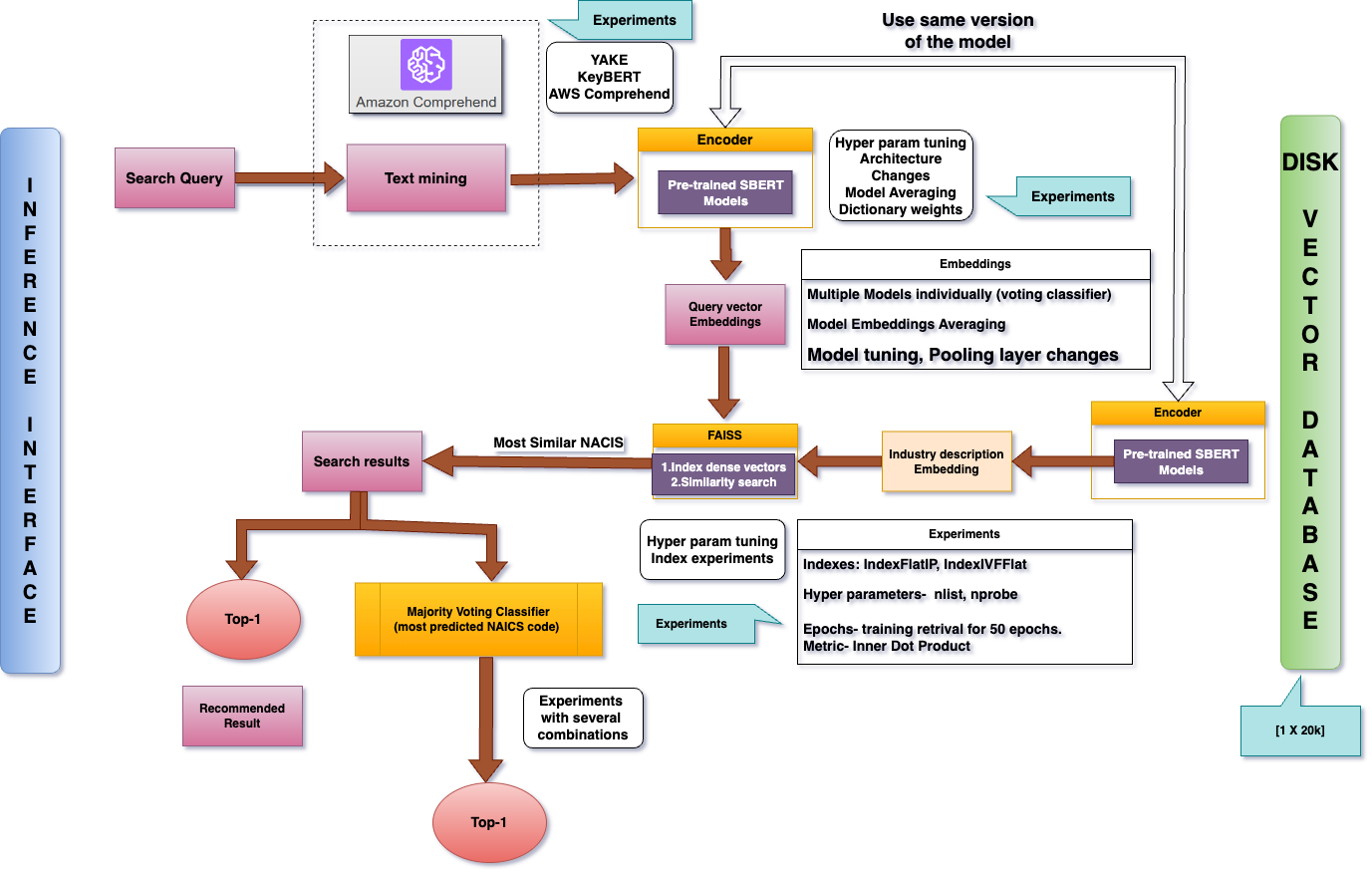
**Background:**

The North American Industry Classification System (NAICS) is a standard used by Federal statistical agencies to classify industries,it is crucial to Life Cycle Assessment (LCA).The two primary methodologies that comprise LCA are process-based and Economic Input-Output (EIO) analysis-based. Process-based life cycle assessment (LCA) reveals environmental hotspots along a product's supply chain by monitoring all material and energy inputs and outputs. However, it takes a lot of time and labor to complete. EIO-LCA, on the other hand, takes a macroeconomic approach, estimating the financial effects of goods and services on the environment using economic data provided by the government. This approach is faster and makes use of already-existing financial data, which makes it perfect for preliminary evaluations of a product's environmental impact when comprehensive data is hard to come by.

**Literature Survey:**

In the evolving field of automated environmental assessment and industry classification, significant contributions have been made by Balaji et al. in 2023 with two studies: "CaML: Carbon Footprinting of Household Products with Zero-Shot Semantic Text Similarity"[1] and "Flamingo: Environmental Impact Factor Matching for Life Cycle Assessment with Zero-Shot Machine Learning."[2] The CaML algorithm applies semantic text similarity for automated Economic Input-Output Life Cycle Assessment (EIO-LCA) using a dataset of 46,646 products from Amazon, achieving a 61% accuracy in its zero-shot implementation and 52% when fine-tuned, tested on 6,308 products.While Flamingo automates the selection of Environmental Impact Factors (EIF) for product carbon footprinting using SBERT and industry sector codes, applied to a smaller dataset of 967 products from Amazon, including ground truth HS6 codes and EIFs, but it recorded a modest accuracy of 11.7%. Earlier, in 2018, Wood et al. presented "Automated Industry Classification with Deep Learning,"[3] using EverString's API and a deep neural network to exceed the precision of gold-standard databases in industry classification, with an overall accuracy of 47.9% on a substantial dataset of 18 million companies. These studies collectively mark significant strides in the use of machine learning for environmental impact assessments and industry classification in terms of progress and challenges.  
 Our proposed model uses FAISS-IVF for search and retrieval reducing the time complexity of the task by six folds and the fine tuned all-mpnet-base-v2 increased the accuracy to 64.55.

**Methodology:**

****

*Figure 1.The above diagram presents a comprehensive overview of the project's methodology and illustrates the flow of data throughout the system.*

**Dataset:**

The CaML study utilized data from three primary sources to achieve its objectives. Firstly, it included an extensive compilation of NAICS descriptions, encompassing over 20,000 rows across 1,016 codes, which exhibited a varied distribution of definitions per code. Secondly, the study incorporated USEEIO CO2 emissions data for each of these NAICS codes, providing a crucial measure of environmental impact. Lastly, it involved product industry annotations for about 6,000 grocery items, which were categorized under 210 different NAICS codes.

**Data preprocessing:**

The process begins with lower casing, which ensures uniformity in text by converting all letters to lowercase. This is followed by the removal of punctuation and numbers.

**Removing Stop words:**

Since the product descriptions are in English, we systematically remove words that are frequently used but have little semantic significance for the purposes of our analysis. These consist of terms like "a," "the," and "and."

**Lemmatization:**

This process transformed words to their root forms, leading to greater consistency across the dataset and enabling more efficient analysis in subsequent stages. Here the data in the column (descriptions) were lemetized.

**Key Phrase Extraction:**

The process of automatically identifying significant terms that best capture the topic or primary ideas of a document or text is known as "key phrase extraction". Here we experimented with three methods:

**YAKE:** It is an unsupervised, lightweight method.It focuses on automatic keyword extraction from single documents utilizing statistical text features for keyword extraction.When experimented with YAKE. Too many keywords were extracted which actually created noise in the document. Hence this method was dropped

**KeyBERT:** KeyBERT generates document embeddings using sentence-transformers. It compares pretrained multilingual models to unsupervised models for keyword extraction. When tried with keyBERT very few keywords were extracted which was not sufficient enough. Hence this method was also dropped.

**Amazon Comprehend**: After reaching out to the author of the paper and based on his suggestion Amazon Comprehend was tried. It is used for semantic analysis of patents.

It features capabilities such as clustering patents for topic modeling and identifying key terms/phrases.This method gave us better results.

**Removing duplicates, least common and most common words:** Duplicate words are removed to avoid redundancy. The process also involves eliminating the most common generic words and the least common non-dictionary words to refine the dataset's relevance for example “manufacture”, “ product”. Products with less than five words and those with fewer than two consensus votes are dropped to maintain data quality.

Additionally, there is a mapping of CO2e/$ for each NAICS code’s description, providing an environmental impact assessment. Finally, the merged file is filtered to contain only descriptions of NAICS codes in the product annotations.

**Modeling and Model Details and Training:**

Our project uses FAISS (Facebook AI Similarity Search) which is a library designed to facilitate efficient similarity search and clustering of dense vectors. It works by indexing and compressing vectors to enable fast and large-scale retrieval. FAISS with flat indexing and IVF indexing were performed to observe the results. For flat indexing, we've chosen parameters including a batch size of 64, k=20 and metrics as METRIC\_INNER\_PRODUCT. Similarly for IVF, nlist (110), nprobe (20), quantizer (flat), train (True), metric (METRIC\_INNER\_PRODUCT) were used.

As mentioned in the methodology diagram, a Voting Classifier was used to find the top-1 NAICS code to match the product. The regular VotingClassifier and Weighted Voting Classifier with voting='hard',weights= [.6, .4, .4] were used to compare and find the best approach.

**The proposed model combinations are:**

*all-mpnet-base-v2 model*: A sentence-transformer library which maps sentences and paragraphs into a 768-dimensional vector space. It is intended for use in tasks such as semantic search and clustering.

On top of this Base paper method Zero-shot CaML, a series of experiments which includes the usage of FAISS and voting classifiers with the corresponding hyperparameters have been performed to improve the accuracy and reduce the time complexity of mapping the Product to its appropriate NAICS code which are explained in the next section.

**Experiments**

Metrics like accuracy and speed were the main focus of our evaluation of the model's performance across different configurations. To determine the best solution for our use case, we experimented with a variety of machine model models and search and retrieval techniques.These are the main tests carried out for the investigation.

**Approach 1:Base Paper's Zero-shot CaML Method (all-mpnet-base-v2) with pytorch.cos\_sim**:

The first experiment used the all-mpnet-base-v2 model with PyTorch cosine similarity to replicate the zero-shot CaML method described in the base paper. This method used the same preprocessing and default NLP model parameters as the base paper. Its accuracy on NAICS is 61.199% Support for the model was registered at 5719 instances, completed in 1 hour and 3 minutes, indicating the requirement for improvement.

**Approach 2: Base Paper's Zero-shot CaML Method (all-mpnet-base-v2) with faiss\_search**

The second experiment integrated the FAISS-FlatIndex for faster similarity computations, improving the zero-shot CaML method of the base paper. The preprocessing and default model parameters from the base paper were used in this adaptation, along with the all-mpnet-base-v2 model. Compared to the first experiment, it achieved a slight decrease in accuracy: 60.258% on NAICS and 70.320% on BEA. But processing speed was the main improvement, cutting the time down to a mere 51 seconds. This experiment illustrated the efficiency gains that can be achieved by using sophisticated indexing methods such as FAISS.

**Approach 3:Input Tuned with all-mpnet-base-v2 and faiss\_search:**

Two configurations of the all-mpnet-base-v2 model integrated with FAISS search were evaluated for product classification into NAICS codes as part of our project's evaluation. While the second configuration used a hyperparameter-tuned IVFIndex in FAISS to slightly improve accuracy to 63.839% and reduce the runtime to 49 seconds, the first configuration applied FAISS with FlatIndex and achieved an NAICS accuracy of 63.822% in 51 seconds. The two models yielded identical accuracies of 74.331% for BEA and supported 5719 instances.

**Approach 4: Input-tuned, all-mpnet-base-v2 with FAISS Search (IVFIndex) trained: (Proposed Model)**

All-mpnet-base-v2 that has been input-tuned, optimized for parameters, and FAISS search This advanced configuration made use of weighted word embeddings targeted at NAICS words, a modified pooling layer (pooling\_mode\_mean\_sqrt\_len\_tokens), and KeyPhase extraction output. FAISS-IVFIndex with hyperparameter tuning and retrieval training was also utilized. After a noteworthy performance improvement, it was determined to be the recommended model. It completed the BEA with 75.10% accuracy and the NAICS with 64.55% accuracy in a manageable 2 minutes and 2 seconds.This combination was chosen as our "proposed model" because it produced the best accuracy with a marginally longer running time than the input-tuned combination of models with weighted averaging.

**Approach 5: Input-tuned Combination of Models with Weighted Averaging**:

The input-tuned models all-mpnet-base-v2, gtr-t5-large, and multi-qa-mpnet-base-cos-v1 were combined in this experiment. For effective text similarity computations, it trained the FAISS-IVFIndex with weights of 0.7, 0.1, and 0.2. It then applied weighted averaging to the model embeddings. This novel method processed the data in 59 seconds and produced accuracy of 60.657% on NAICS and 71.935% on BEA. This experiment demonstrated how combining multiple models with weighted embeddings can improve efficiency and accuracy.

**Approach 6: Input-tuned Combination with Weighted Majority Voting**:

This approach focused on a weighted majority vote utilizing a combined, input-tuned model configuration. The ensemble included the all-mpnet-base-v2, gtr-t5-large, and multi-qa-mpnet-base-cos-v1 models. It implemented a weighted majority voting system, assigning weights of 0.6, 0.4, and 0.4 to the respective models. This strategy aimed to leverage the collective strengths of these models to enhance overall accuracy. The process, completed in a duration of two minutes and eleven seconds, yielded impressive results: an accuracy of 62.755% on the NAICS and 73.509% on the BEA. This experiment successfully illustrates the efficacy of ensemble methods in machine learning, achieving an optimal balance between speed and accuracy.

| **MODEL(Approach 1 to 6)** | **Accuracy (NAICS)** | **Accuracy (BEA)** | **Support** | **Speed** |
| --- | --- | --- | --- | --- |
| Base paper‘s Zero-shot CaML Method  (all-mpnet-base-v2) pytorch.cos\_sim | 61.199 | 72.075 | 5719 | 1 hour 3 min |
| Base paper‘s Zero-shot CaML Method  (all-mpnet-base-v2) faiss\_search | 60.258 | 70.32 | 5719 | 51 secs |
| input tuned, all-mpnet-base-v2/ FAISS- FlatIndex.  input tuned, all-mpnet-base-v2/FAISS-IVFIndex (HP-tuned, retrieval trained) | 63.822  63.839 | 74.331  74.331 | 5719  5719 | 51 secs    49 secs |
| **input-tuned, all-mpnet-base-v2**  **( HP-optimized, Weighted embeddings Pooling layer changed) faiss\_search- trained** | **64.55** | **75.1** | **5719** | **2 min 2 secs** |
| Input-tuned, Averaging embeddings of (gtr-t5-large, all-mpnet-base-v2, Multi-qa-mpnet-base-cos-v)  faiss\_search-trained | 60.657 | 71.935 | 5719 | 59 sec |
| input-tuned Weighted Voting classifier to above 3 NLP models, faiss\_search-trained | 62.755 | 73.509 | 5719 | 2 min 11 sec |

*Table 1 - Comparative Analysis of methods for Industry Classification.*

**Other Experiments :**

Apart from experimenting with the CaML model, the project also employed different Sentence Transformer Models,Universal Sentence Encoder and GPT-3, all of which produced different degrees of accuracy. The Sentence Transformer Model-grt-t5-base obtained an accuracy of 0.32, while the Model all-miniML-L12-v2 obtained an accuracy of 0.37.The Universal Sentence Encoder obtained an accuracy of 0.27 and the GPT-3, which is capable of sophisticated language processing, recorded an accuracy of 0.38.

**Discussion and Future improvements.**

As demonstrated by a noteworthy 60-fold reduction in time complexity and a 3% increase in accuracy for sentence similarity models, our project represented a major advancement in the application of CaML. This development is especially helpful for upcoming studies that test the performance of models in end-user computing environments. Our investigation of Data Mining Techniques, such as Model Combinations, FAISS, and Keyphrase Extraction, was essential to improving the process of using text descriptions to map Products to Industries. In order to increase text similarity accuracy, our results demonstrated the value of pre-processing and the tactical significance of weighting words in target classes. Notably, the author of the original work has acknowledged our valuable contributions and is poised to incorporate them into the forthcoming version of their publication.

**Future improvements**

Our work provides a wealth of new directions for future investigation. When our reduced time complexity approach is used, it becomes possible to evaluate more than 2700 available sentence-similarity models, and our FAISS approach can test about 60 models at once. This allows for extensive model testing. The mismatch in the search space for sentence similarity between product and industry descriptions, however, is a serious problem that we have found. Pre-training models on specialized datasets such as SEC-EDGAR, Amazon Products, and NAICS may be a viable approach to improve model accuracies. By laying the foundation for more sophisticated and effective text mapping approaches, our work encourages further investigation into our discoveries and methods.

**References**

[1] Balaji, B., Vunnava, V. S. G., Guest, G., & Kramer, J. (2023). CaML: Carbon Footprinting of Household Products with Zero-Shot Semantic Text Similarity. In Proceedings of the ACM Web Conference 2023. WWW ’23: The ACM Web Conference 2023. ACM. https://doi.org/10.1145/3543507.3583882

[2] Balaji, B., Vunnava, V. S. G., Domingo, N., Gupta, S., Gupta, H., Guest, G., & Srinivasan, A. (2023). Flamingo: Environmental Impact Factor Matching for Life Cycle Assessment with Zero-shot Machine Learning. In ACM Journal on Computing and Sustainable Societies (Vol. 1, Issue 2, pp. 1–23). Association for Computing Machinery (ACM). https://doi.org/10.1145/3616385

[3] Wood, S., Muthyala, R., Jin, Y., Qin, Y., Rukadikar, N., Gao, H., & Rai, A. (2018). Automated Industry Classification with Deep Learning. In 2018 IEEE 12th International Conference on Semantic Computing (ICSC). 2018 IEEE 12th International Conference on Semantic Computing (ICSC). IEEE. https://doi.org/10.1109/icsc.2018.00018