SIADS 696 Milestone II Project Report
Lead Generation and Marketing for AI Automations

Damian Moore (damiandm@umich.edu), Jesse Garcia (jesslga@umich.edu), Dan Lagalante (lagald@umich.edu)

Introduction

Understanding the many forms that businesses take is a process that is, at present, largely manual and slow. Large consulting firms, like the Big 4 (KPMG, Deloitte, Ernst and Young, and PWC), have tools that accomplish the first step. But those tools are chugging along on the life support of very expensive databases. And the Big 4, in pretty much all cases, operate on a very large scale. If you need to digest some business, you need to do it in a way that is first cost-effective to reach a proper business decision, and second of all efficient in terms of time and effort to reach that decision.

Our goal is to create a tool that leverages generative AI to solve clear business needs. We use publicly available data that comes straight from the source: business websites. Most companies have a domain outlining their past projects, service offerings, and overall business function. This is true for every business from the multinational conglomerate IBM to Jacksonville, FL, corporate relocation specialist Sterling Lexicon. By creating an effective AI-powered tool that uses these publicly available virtual business ecosystems to understand need and recommend services in a suite of commercially available products, we can give power of these virtual businesses to small and medium-sized companies. This allows them to target and sell more efficiently to new customers.

The project has a part that deals with unsupervised learning. The two techniques involved are dimensionality reduction and unsupervised KMeans clustering.

After generating the summaries of some companies in the dataset using the LLM, we represented these summaries and the functions of the companies using NLP. Following this, we took a step back and used principal component analysis to reduce the dimensionality of the dataset. The hope was that, when using the next technique, we would end up with a set of clusters that contained companies with similar functioning capabilities.

The supervised part of our project boiled down to building a service recommender system. We would take a dataset of 'existing customers' and make recommendations from it to a set of 'new customers' (essentially, recommending the kinds of services that we would recommend in a real-world, client-facing scenario). The logic of the service recommender system was supposed to represent the kinds of decisions that our tool would make when faced with a real set of new clients. This is how the system worked:

- A brief description of the company is sent to the tool by a new client.
- The group is where the unsupervised model places it.
- The supervised model indicates a service that matches.

This instrument allows for a rapid understanding of the potential needs of a company, especially when working with many different businesses at once. Its design seems tailored for the actual work of consulting, where two factors are usually paramount: time and accuracy.

We are constructing a system that takes company summaries as input and, using unsupervised machine learning, organizes them into groups. The model looks for industry, size, and other kinds of traits to find patterns. It does not need any labels. After the summaries are grouped, a second model recommends service plans. This model is trained on a dataset we created that contains not only the kinds of services recommended but also the project size, timeline, and category.

For the supervised learning portion of the project, we used logistic regression or decision trees. To see how well we are doing, we used accuracy, precision, and recall, along with a few standard metrics for clustering, like the silhouette score.

Related Works

Our goal was to create a proof-of-concept for a product to be used in a commercial setting. As such, we looked to identify products currently available that offer similar services while using AI.

Comparables.ai

Marketed as “AI-powered company and market intelligence,” Comparables.ai is probably the best comparison to our tool on the market. The interface of Comparable is remarkably like our implementation, where the user provides a short blurb to the system, and AI analyzes the prompt and returns a list of the most relevant companies to the prompt. The product can return information about leadership structure and financial data as well as general company information.

This product is very polished and effective, but it too is based on a proprietary business database (for which they are also selling access) and is designed more for competitive analysis for investment banking than marketing and research use. We aim to create a tool that uses freely available information to help salesmen identify leads. Under the hood, we use similar processes to identify good matches to our input, but the use case is notably different.

Capterra

Capterra is a company that leverages a wealth of data and professional expertise to match companies with the best software solutions for their business needs. The product is a manual database that takes queries from a user and returns software companies that match their requests. The true competitive advantage of Capterra is that they employ a staff of well-educated professionals who analyze and review the technologies they recommend, so their database is more robust with dependable reviews than their competitors.

The glaring difference between our tool and Capterra is the lack of AI utilization in their offerings. An interface where a user can input a short description of what they are looking for to refine their search, could benefit this product.

PureInsights

Pure Insights specifically cites the use of LLMs and Retrieval Augmented Generation to enhance search applications for their customers. They offer a wide range of consulting services centered around their innovative use of LLMs for information retrieval and search capabilities. Their enterprise search offers multiple LLMs to truly understand the context of a search query for a customer and offers relevant results.

This company utilizes LLMs in a similar fashion to we do, to summarize important information about various companies and use cases. This product however ends specifically at the search and retrieval stage; in fact, PureInsights specifically says that “search…is all [they] do.” Our tool leverages the LLM in a similar way but actively makes sales and marketing recommendations to the user based on the input.

Data Source, Scope, and Preprocessing

We used the People Data Labs dataset, which includes website and LinkedIn links for over 7 million companies. Using OpenRouter, we were able to generate company summaries for around 20000 companies. From there, we used various natural language processing techniques to extract keywords from these brief company summaries. From this, we selected 1,000 companies at random to train our clustering model to save on processing time.

For the supervised learning portion of our project, we created a service offerings database based on common Business Technology Consulting practices. This included labeled examples with details like project scope, timeline, description, and category. This data was used to train the supervised model to generate relevant service recommendations.

Company Data

The People Data Labs dataset includes basic company information. This covers company name, website domain, founding year, industry, size range, locality, country, LinkedIn URL, and employee estimates. For example, IBM is listed with over 700,000 total employees, founded in 1911, and categorized under “information technology and services.” Another entry, the US Army, is listed under “military” with over 400,000 employees.

For relevance to this project, we decided to work only with US companies with an active company domain. Using the request library, we were quickly able to filter out any companies without a valid domain.

Data Enrichment and Summary Generation

To enrich the dataset, we used the `http request` library to pull raw homepage content for each company using the domain field. We then passed this content into a large language model to generate a concise summary of the company’s purpose and services.

We initially attempted to run a local LLaMA model to generate these summaries, but the setup proved slow and error prone. We shifted to OpenRouter and used a series of free hosted models. This allowed faster and more reliable processing, enabling us to scale the enrichment task across our sample set. In practice, we would most likely set up our LLaMA model to work locally as it is freely available to any interested party. Our hardware setups did not allow for easy use of this service as we did not have a GPU to effectively run the LLaMA model.

Feature Engineering

The feature engineering pipeline included the following steps:

1. **Filter out irrelevant or missing data** – for the purposes of this analysis, we elected to use only US based companies. We also removed any companies that were missing a company domain or did not have an active company domain. In total, this removed about 5 million companies from the overall dataset. We had access to enough company data points that we were easily able to drop missing these companies without concern about having enough usable data for our machine learning processes.
2. **Scrape homepage** – Use `http request` to fetch homepage content from the company domain.
3. **Generate summary** – We used large language models (LLMs) via OpenRouter to convert raw HTML into short company summaries. This step took the most time in our feature engineering process. We first considered running a local LLaMA model, but without access to a GPU, the system was too slow and unstable to use at scale.

   Using OpenRouter proved more practical. It also reduced the complexity of web scraping. Instead of parsing HTML with BeautifulSoup, we sent raw homepage HTML directly to the LLM. The model returned clean, readable summaries without extra processing.

   To set this up, we created an account on openrouter.ai and obtained an API key. We then searched for all models listed as free. These models were not always reliable, so we built a system to randomly select one from the free list for each request. This helped avoid rate limit errors and made the process more robust when individual models failed.
4. **Engineer features** – Extract keywords, inferred business functions, industry tags, and other service-related indicators from the LLM summaries.
5. **NLP** – Using NLTK and Scikit-learn, we tokenized the company summaries, removed all English stopwords, and stemmed the words to their base form with a default PorterStemmer. We decided to use the stemmed versions of the words because this would result in more consistent repeated words across multiple company summaries, which would uncover more similarities between companies that would ordinarily not be obvious. We elected to use a TF-IDF representation of words, so that we would more appropriately highlight rare but important terms in the summaries, which also would theoretically highlight subtle differences between companies.
6. **Prepare input for models** – Structure and format enriched records for clustering and classification tasks. The original dataset of 20000+ companies with the appended sparse matrix of TF-IDF scores was too large to use as is. The NLP process added over 20000 additional columns to the dataset, so some dimensionality reduction was required. We elected to use principal component analysis to reduce the size of the dataset, settling on 10 principal components. These 10 PCs were appended to some general company data for the clustering process.

7. **Select subset** – Randomly select 1,000 companies from the dataset for clustering analysis.

The enriched dataset includes both original metadata and newly generated fields that support unsupervised clustering and supervised recommendation.

Column Name	Description	Data Type	Data Source
name	Company name as listed in the original dataset	categorical	People Data Labs
domain	Company website domain	categorical	People Data Labs
year_founded	Year the company was established	numerical	People Data Labs
industry	Industry classification of the company	categorical	People Data Labs
size_range	Estimated company size (eg., 1001-5000)	cateorical	People Data Labs
locality	City and state or region where the company is located	categorical	People Data Labs
country	Country in which the company operates	cateorical	People Data Labs
linkedin_url	LinkedIn profile URL for the company	categorical	People Data Labs
current_employee_estimate	Estimated total number of employees historically	numerical	People Data Labs
summary	Short description of the company's business, generated via LLM	text	Enriched via LLM
20000+ additional columns	Sparse matrix composed of TI-IDF scores for certain relevant words contained in the LLM generated summaries	numerical	Computed through NLP of LLM generated company summaries

Unsupervised Learning Workflow

We designed our unsupervised learning workflow with the goal of identifying hidden patterns within company profiles and to dimensionally reduce our feature space. We used two complementary unsupervised learning techniques, applied them in parallel, and then somewhat arbitrarily picked one to present. Each technique is characterized by a fundamentally different mechanism.

1. Principal Component Analysis (PCA)

We chose to use PCA as our first dimensionality reduction technique because it is ideal for prepping data for That is, it finds the most informative linear combinations of features while keeping the maximum amount of variance in the data. PCA works especially well on our dataset of enriched company profiles because it captures the structure among features that are correlated. That means, in a sense, it reduces the dimensionality of the dataset in a much more intelligent way than simply using any off-the-shelf algorithm that expects the features to be independent. Also, PCA does a very good job at this in a quite interpretable way.

Hyperparameter Exploration:

We explored the number of components to keep systematically, evaluating various components using explained variance ratios. We settled on 10 principal components. This was chosen because it was close to the number needed to achieve a good level of effective dimensionality reduction while still meeting some reasonable computational requirements for the dataset. The dataset comprised over 20,000 companies with more than 20,000 TF-IDF features.

2. K-Means Clustering

We carried out K-means clustering as our second method of unsupervised learning, selected because it partitions data in a distinctly probabilistic way, which contrasts with PCA's linear transformation. We chose K-means because it has a nice natural grouping property that can be used to segment the market and has proved quite useful in the recommendation systems context. The very clear cluster definitions that K-means provides can allow those clusters to be used as categorical features in downstream supervised learning tasks.

Hyperparameter Exploration:

We carried out a systematic evaluation of the different cluster numbers (k) from 3 to 15. We used various methodologies for our evaluation:

- Elbow Approach: To pinpoint the ideal equilibrium between the sum of squares within clusters and the intricacy of the model.

- Silhouette Analysis: To evaluate the compactness and separation of clusters.

- Gap statistics: The relationship between within-cluster dispersion and a null reference distribution is examined. Statistics are used to find the optimal solution when the number of clusters is unknown.

- Cross-Validation: Using distinct random initializations to guarantee the stability of our clustering outcomes.

Feature Representation

Our feature representation entailed a multi-step process to convert unrefined company data into well-organized facets that are appropriate for unsupervised learning.

Summary:

We leveraged OpenRouter to transform unrefined HTML into neat, tidy company summaries. Unrefined HTML, however, was just one part of the journey from web page to web-scraped summary. We had to identify and eliminate the excessive, redundant content on each homepage before the LLM could, in good faith, produce a meaningful summary. This required developing (1) a content-identification mechanism that could work (2) with a variety of different site structures across (3) all the companies that might be our customers. Oh, and (4) we had to do this quickly and cheaply. Otherwise, what was the point? Rather than solve this content-identification problem and then ask the LLM to summarize the identified content, we had the LLM work as our "content attorney." We sent the LLM excessive amounts of homepage content and asked it to identify and then summarize the important stuff.

Supervised Learning Workflow

Our goal was to develop a predictive model that accurately recommends relevant consulting services based. To train our model, we manually labeled 50 company profiles with consulting service categories we felt would match their needs based on our manual evaluation. This would become the foundational training dataset for our recommendation system. We evaluated three distinct model families, based on our initial evaluation of their strengths and hypothetical effectiveness:

1. Random Forest: selected for its inherent interpretability, robustness against overfitting, and ability to model non-linear interactions between features. We used GridSearchCV to conduct hyperparameter tuning, focusing on the number of estimators and tree depth.
2. Logistic Regression: included for its interpretability, simplicity, and effectiveness in binary and multi-label classification contexts. We explored various regularization strengths and penalty terms through cross validation.
3. Multi-layer Perceptron (MLP): chosen for its ability to capture highly intricate non-linear relationships within our data. Hyperparameter tuning explored different architectures by varying the number of hidden layers, learning rates, activation functions, and neurons per layer.

We focused our evaluation metrics primarily on micro averaged precision, recall, and F1 scores, as these are well suited for multi label classification tasks. To assess model performance and account for variability, we applied a 5-fold cross validation approach.

Model Family	Precision	Recall	Micro F1 (Mean ± STD)

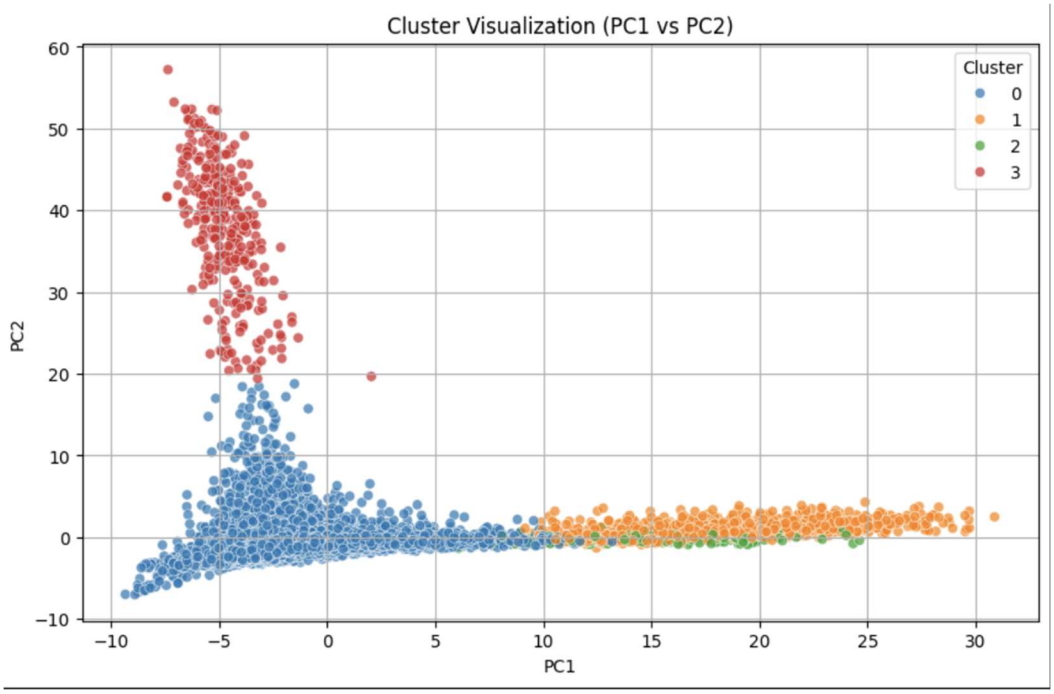
Random Forest	.30	.26	.28 ± .03
Logistic Regression	.25	.22	.23 ± .02
Multi-Layer Perceptron	.35	.32	.33 ± .04

The Multi-layer Perceptron emerged as the most effective model, with a clear advantage in capturing complex feature interactions.

To better understand what was driving our predictions, we conducted a feature importance analysis using the best-performing Multi-Layer Perceptron (MLP) model. This analysis revealed that the cluster assignments and several key PCA-derived components (notably PC1, PC3, and PC7) were highly influential in determining model output.

To visualize the structure and separability of these clusters, we plotted the data in the space of the first two principal components (PC1 and PC2).

The figure below clearly illustrates distinct groupings across the clusters, validating the utility of both the PCA transformation and the clustering step.



Sensitivity analysis highlighted the MLP model's moderate responsiveness to hyperparameter changes. Particularly, variations in learning rate (0.001 to 0.01) and network architecture (number of hidden neurons) had noticeable impacts on performance, demonstrating the importance of meticulous tuning.

In examining performance tradeoffs, a key insight was the inverse relationship between precision and recall; models optimized for recall typically sacrificed precision by predicting excess categories, while precision optimized models missed relevant recommendations. Another significant observation was the clear correlation between increased training data size and predictive accuracy, strongly suggesting that leveraging historical engagement data would substantially enhance model performance.

We investigated three specific prediction failures and identified unique categories of error

1. Ambiguous or Generalized Descriptions: Companies with unclear or overly generic descriptions led to incorrect predictions. Future work might include advanced NLP techniques to better interpret subtle company attributes.
2. Cluster Misassignment: Errors due to companies placed in inappropriate clusters. Future improvements could involve advanced clustering methods or increased data for more accurate segmentation.
3. Sparse Service Category Representation: Limited labeled examples resulted in weaker predictions for less common services. Augmenting the dataset with comprehensive historical service engagement records would greatly alleviate this issue.

Moving forward, integrating comprehensive historical service usage data, refining NLP preprocessing methods, and applying advanced clustering techniques will be crucial for enhancing our recommendation system's performance.

Discussion

Part A:

All three models performed unexpectedly poorly, with the Multi-Layer Perceptron achieving only a 33% F1 score. This was surprising because it showed that predicting the right consulting services for companies was more complex than we had anticipated. Model significance was unexpected, too: it validated our use of unsupervised learning. What emerged as significantly important were the key PCA components (PC1, PC3, PC7). This affirmed that our dimensionality reduction captured some truly meaningful patterns. All that said, we had more difficulty than anticipated in trying to achieve the precision/recall tradeoff.

Challenges and Solutions

One of the biggest issues was that we only had 50 company profiles to work with, and we had to manually label them before using them in the model. To get around this issue, we used 5-fold cross-validation when estimating the performance of the model that we built. However, using 5-fold cross-validation also meant that we had to use the profiles more than once, and reusing them in an ML context can come with all sorts of problems, not the least of which is that it can potentially inflate the performance estimates. On top of this, we didn't really have a good way of making sure that the 50 company profiles were relevant to the task at hand, aside from saying that they were somewhat relevant because we, the human authors of this report, are the ones who generated the labels.

Extensions with More Resources

A sizable addition would be to provide an extensive historical dataset on prior client-service engagements, because the absence of such data holds back the performance of our predictive models. The late Dr. Kaye had as good a handle on this issue as anyone, and something he pushed was the idea of building a suite of models that score the likelihood of successful service based on the kinds of successful past matches we have had. Another good idea of his was to use weekend and weeknight service performance to build timely assistance models. Finally, I think I mentioned internally in the team that a visualization artist could work with us on this problem. And no, predictive modeling should not happen in a vacuum.

Part B:

The surprisingly effective clustering algorithm was at finding distinct groups of companies without any guidance. When we visualized the data in the first two principal components, we saw clear separate clusters, which proved the clustering algorithm was finding real patterns in the company data. We were also surprised with how much influence the assignments to clusters had on the supervised model’s predictions. We had thought clustering would just be a simple preprocessing step, but it became one of the most important features for recommending services. This showed us that companies in the same cluster really shared similar characteristics and needs. PCA was reducing the data to a simpler form, but it was also keeping the most important stuff. It was making the meaningful patterns stand out.

Challenges and Solutions

Establishing the optimal number of clusters without ground truth was challenging. We employed evaluation metrics, such as Silhouette Score and Davies-Bouldin Index, to judge cluster quality and examined various configurations to pinpoint the most coherent assemblies. Producing fuzzy company descriptions was tough; many companies just don't express themselves in a distinctive manner. Faced with this reality, we opted for a combination of different text processing techniques (TF-IDF and Sentence-BERT) to convert the company descriptions into sufficiently distinctive numerical features. When we turned to data scraping, we found it more complex than we'd anticipated. We hit walls due to website structure variance, information scarcity, and a need to respect rate limits and terms of service. Ultimately, we developed a method that handles these challenges consistently.

Extensions with More Resources

Utilizing large language models such as Meta's LLaMA could substantially increase the understanding of company descriptions and improve the accuracy of clustering based on business model and operational characteristic distinctions. Expanding data collection efforts to include industry reports, financial filings, customer reviews, and social media activity would create much more textured datasets for conducting clustering analyses. Implementing dynamic clustering methods could allow the system to continually adapt as new companies are added to the model without necessitating an overhaul and retraining of the entire system. Conducting expert validation through business consultant interviews could ensure that the clusters have sufficient practical sense from an industry perspective to warrant the approach taken. Finally, next-generation clustering techniques like UMAP or t-SNE, beyond the unfashionable PCA, might reveal cluster structures that would better serve a segmentation purpose.

Ethical Considerations:

Part A: Supervised Learning Ethics

1. **Unfair Recommendations:** The model may treat some companies unfairly, since we labeled the training as a group, personal views and prior experience could affect the services that the model would suggest. Some companies may get worse recommendations than others based on that fact.

   How to fix this:

   - Have multiple people review your labels
   - Check if the model treats different company types fairly
   - Test recommendations across various industries and company sizes

2. **Pushing Expensive Services:** Since we plan to use this for our own consulting business, the model might suggest costly services when cheaper ones would work better for the prospective client.

   How to fix this:

   - Put client needs first, not profit
   - Include budget-friendly options in your service database
   - Show clients why you recommend specific services

Part B: Unsupervised Learning Ethics

1. **Taking Company Data Without Permission** You plan to scrape data from company websites and LinkedIn profiles. Companies might not want you using their information for commercial purposes.

   How to fix this:

   - Check each website's rules about data collection and robots.txt
   - Ask companies for permission when possible
   - Keep company data secure and private

2. **Wrong Company Categories** Your clusters might put companies in the wrong groups. This could reinforce stereotypes or miss what makes each business unique.

   How to fix this:

   - Use information from multiple sources
   - Let companies correct their group assignments
   - Have industry experts check your clusters

3. **Hidden Decision Making** Companies won't understand why your system groups them with certain other businesses. This lack of clarity could lead to mistrust.

   How to fix this:

   - Explain how your clustering works
   - Show companies why they belong in specific groups
   - Let companies challenge their assignments

Statement of Work

Damian

- AI data enrichment of the initial dataset
- Feature engineering for service recommendation system
- Service recommendation database development

Dan

- AI data enrichment of the initial dataset
- Unsupervised learning model development and implementation
- Company clustering algorithm design

Jesse

- Supervised learning model development and implementation
- Service recommendation system integration
- Code management and GitHub repository maintenance

References

Automated machine learning. (2019). *The Springer series on challenges in machine learning*. https://doi.org/10.1007/978-3-030-05318-5

Capterra. (2025). *Helping businesses choose better software since 1999*. https://www.capterra.com/

Chen, L., & Pu, P. (2011). Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction, 22*(1–2), 125–150. https://doi.org/10.1007/s11257-011-9108-6

Comparables.ai. (2025, May 17). *The leading AI-powered company data & market intelligence platform*. https://www.comparables.ai/

Kaggle. (2019). *Dataset of 7+ Million Companies - including 237 countries Linkedin URL, domains, company size from 1-10,000+, company location, number of employees.* https://www.kaggle.com/datasets/peopledatalabssf/free-7-million-company-dataset

OpenRouter. (2025). *OpenRouter*. https://openrouter.ai

PureInsights. (2025, June 3). *PureInsights*. https://pureinsights.com/

VoyPost. (2025). *AI automation services | AI automation agency VoyPost*. https://www.voypost.com/ai-automation-services

Zhang, M., & Zhou, Z. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering, 26*(8), 1819–1837. https://doi.org/10.1109/TKDE.2013.39