

**School of Arts and Sciences**

**Machine Learning Course Project**

**Restaurant Recommendation System using Machine Learning**

**Submitted By:**

**Majid El Ezzi: 202206742**

**AbdulAziz Sayyed: 2022**

**CSC461: Introduction to Machine Learning**

**Submitted to Dr. Seifedine Kadry**

**Abstract:**  
  
This project focuses on building a restaurant recommendation system using clustering algorithms to identify patterns in Yelp business data. The dataset was filtered to focus on restaurants in Las Vegas, and two clustering methods, DBSCAN and K-Means, were implemented and compared. DBSCAN, a density-based algorithm, effectively identified clusters and noise points, while K-Means partitioned the data into pre-defined clusters based on the Elbow Method. Both algorithms were evaluated for their clustering performance and practicality in making recommendations based on user-provided coordinates. The system aims to enhance user experience through personalized recommendations and increase visibility for businesses.

This project focuses on building a restaurant recommendation system using clustering algorithms to identify patterns in Yelp business data. The dataset was filtered to focus on restaurants in Las Vegas, and two clustering methods, DBSCAN and K-Means, were implemented and compared. DBSCAN, a density-based algorithm, effectively identified clusters and noise points, while K-Means partitioned the data into pre-defined clusters based on the Elbow Method. Both algorithms were evaluated for their clustering performance and practicality in making recommendations based on user-provided coordinates.

**Keywords:**

* Clustering
* DBSCAN
* K-means
* Recommendation System
* Yelp Dataset

1. **Project Proposal:**
   1. **Objective:**

The goal of this project is to develop a restaurant recommendation system that leverages clustering algorithms to group similar restaurants based on location, customer reviews, and star ratings.

The project aims to solve the problem of finding restaurants that best align with a user’s preferences and geographical proximity, particularly in Las Vegas. By using machine learning techniques like DBSCAN and K-Means, the system provides personalized and efficient recommendations.

* 1. **Dataset:**

The dataset used for this project is the Yelp Academic Dataset: Business. It contains detailed information about businesses listed on Yelp, including attributes such as location, ratings, and categories. The dataset originates from Yelp's open dataset for academic and personal projects and includes **192,609 records** with the following key features:

* **Business\_id**: Unique identifier for each business.
* **Name**: The name of the business.
* **Address**: The address of the business.
* **State** and **City**: Location details.
* **Postal Code:** The postal code for the business.
* **Latitude** and **Longitude**: Geographic coordinates for mapping and clustering.
* **Stars**: Average star rating of the business.
* **Review\_count**: Number of reviews received, reflecting popularity.
* **Is\_open:** Aboolean indicating whether the business is currently operational.
* **Attributes:** Additional metadata about the business, such as whether it is kid-friendly or takes reservations.
* **Categories**: A list of business categories, such as "Restaurants" or "Plumbing."
* **Hours**: Operational hours for each day of the week.

For this project, the focus was on businesses categorized as "Restaurants" located in **Las Vegas, NV**. After filtering and cleaning the data, relevant features such as **latitude**, **longitude**, **stars**, and **review\_count** were used for clustering and recommendation tasks. The dataset is ideal for demonstrating the application of clustering algorithms due to its size, diversity, and richness in spatial and review data.

* 1. **Approach:**

The project explores clustering techniques to recommend restaurants based on their location, star ratings, and review counts. Two distinct clustering algorithms, **DBSCAN** and **K-Means**, were applied separately to the dataset for comparison and evaluation.

* **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):**
  + Chosen for its ability to detect clusters of arbitrary shapes and handle noise effectively.
  + Applied to geographic coordinates (latitude and longitude) and business-related features (review count and star rating) to identify clusters based on density.
* **K-Means:**
  + Chosen for its simplicity and efficiency in partitioning data into a predefined number of clusters.
  + Utilized the Elbow Method to determine the optimal number of clusters and then grouped restaurants accordingly.

The separate implementation of both algorithms enables a direct comparison of their strengths and weaknesses in addressing the problem. These clustering methods are integrated with visualization and recommendation systems to aid users in finding highly rated restaurants near their location.

* 1. **Impact of the Recommendation System:**  
     This recommendation system significantly enhances user experience by delivering personalized and efficient restaurant suggestions, saving time and effort. For businesses, it provides targeted visibility, potentially boosting customer engagement and revenue. By grouping similar restaurants, the system enables users to discover unique and highly rated dining options tailored to their preferences.

1. **Data Exploration and Preprocessing:**  
     
   **Data Cleaning:**  
     
   - No entries with missing geographic coordinates (latitude/longitude), as these are critical for clustering.  
   - For non-essential columns such as hours and attributes, missing values were left as-is, as they are not directly relevant to clustering.  
   - Removed columns such as hours, is\_open, attributes, and business\_id because they are not directly useful for clustering or recommendations.  
   - Retained name, latitude, longitude, stars, review\_count, and categories for clustering and visualization.  
     
   **Exploratory Data Analysis (EDA):**  
     
   Analyzed the distribution of star ratings and review counts to identify trends. Key visualizations include:  
   1. A histogram of star ratings showing most businesses have ratings between 3 and 5.  
   2. A scatter map plotting restaurants based on their latitude and longitude, color-coded by star ratings and sized by review count.  
   3. A bar plot highlighting the top 20 restaurants based on review count and star ratings.  
     
   Feature Engineering:  
     
   - Created a binary feature 'Restaurants' to indicate whether a business is a restaurant.  
   - Added a 'Cluster' feature to represent predicted cluster assignments from the clustering models.  
     
   Data Splitting:  
     
   As this is an unsupervised learning task, the traditional training-validation-test split is not directly applicable.

**Data Cleaning:**

* **Handling Missing Values**:
  + No entries with missing geographic coordinates (latitude/longitude), as these are critical for clustering.
  + For non-essential columns such as hours and attributes, missing values were left as-is, as they are not directly relevant to clustering.
* **Duplicate Entries**:
  + No Duplicate Entries were found (Checked on google colab).
* **Irrelevant Features**:
  + Removed columns such as hours, is\_open, attributes, and business\_id because they are not directly useful for clustering or recommendations.
  + Retained name, latitude, longitude, stars, review\_count, and categories for clustering and visualization.

**Exploratory Data Analysis (EDA):**

* Analyzed the distribution of star ratings and review counts to identify trends:
  + The majority of businesses have star ratings between 3 and 5.
  + A high skew in review counts was observed, with a small number of businesses having very high reviews.
  + There is no clear trend between the number of reviews and the star rating.
  + Businesses with low review counts tend to have varied ratings, while those with higher review counts seem to have a more evenly distributed range of ratings.

PS: Graphs and visuals are on the colab.

**Feature Engineering:**

Two new features were created:

* Restaurants: A binary feature indicating whether a business is a restaurant, based on the "categories" column.
* Cluster: A feature representing the predicted cluster assignment from the clustering models (K-Means and DBSCAN), helping to identify which group a business belongs to.

**Data Splitting:**

As this is an unsupervised learning task (clustering), the traditional training-validation-test split is not directly applicable.

1. **Model Selection and Training:**
   1. **Model Choice:**

For this project, we used two clustering algorithms: **K-Means** and **DBSCAN**. Both are unsupervised learning models suitable for clustering restaurants into groups based on similar features, such as location, review count, and star rating.

* **K-Means** is chosen due to its efficiency and ease of understanding for clustering data into predefined numbers of clusters. It is suitable when clusters are roughly spherical and can be defined by distances.
  + **n\_clusters:** The number of clusters was chosen based on the “Elbow Method”.
  + **init**: Used the 'k-means++' initialization to improve convergence speed.
* **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) is selected for its ability to identify arbitrarily shaped clusters and handle noise in the data. It does not require the number of clusters to be specified upfront, making it ideal for real-world data that may contain outliers.
  + **eps:** The maximum distance between two samples to be considered neighbors, adjusted based on the data.
  + **min\_samples:** The minimum number of samples required to form a dense region, typically set to 5.
  1. **Training:** Both models were trained on the entire dataset after preprocessing, including feature engineering and cleaning.
  2. **Hyperparameter Tuning:**

For K-Means, the number of clusters was determined using the Elbow Method and Silhouette Score. For DBSCAN, min\_samples was set to 5, and eps ranged from 0.1 to 1, with the Silhouette Score used to evaluate cluster quality for each eps value.

* 1. **Evaluation Metrics:**

**Silhouette Score**: Measures how similar each data point is to its own cluster versus other clusters. A higher score indicates better-defined clusters.

* 1. Computational Efficiency Discussion:

- DBSCAN:

Effective for detecting arbitrarily shaped clusters and handling noise. However, it can be computationally expensive for larger datasets, especially with high-dimensional features.

- K-Means:

Computationally efficient and scalable, making it ideal for simpler clustering tasks. However, it assumes clusters are spherical, which may limit its flexibility with complex data distributions.

1. Evaluation and Analysis:  
     
   Performance Evaluation:  
     
   For K-Means, the silhouette scores were calculated for a range of cluster numbers (2 to 10), with the highest score achieved at k = 2 (0.89). In contrast, for DBSCAN, the best silhouette score (0.9974) was achieved with eps = 0.01, showing superior performance in identifying well-defined clusters.  
     
   Error Analysis:  
     
   - K-Means showed declining silhouette scores as cluster numbers increased, suggesting some clusters may not be well-separated.  
   - DBSCAN performed well with small eps values but was sensitive to parameter tuning and dataset size.  
     
   Model Comparison:  
     
   - K-Means provided consistent clustering and scalability, with reasonable results across datasets.  
   - DBSCAN excelled at identifying noise and arbitrarily shaped clusters, achieving higher silhouette scores but requiring more computational effort.
   1. **Performance Evaluation:**

For K-Means, the silhouette scores were calculated for a range of cluster numbers (2 to 10), with the highest score achieved at k = 2 (0.89). This indicated that two clusters were optimal for this dataset. In contrast, for DBSCAN, the best silhouette score (0.9974) was achieved with eps = 0.01, showing that DBSCAN detected well-defined clusters with a minimal eps value.

* 1. **Error Analysis:**

The **K-Means** model consistently produced strong clustering results, but the silhouette scores showed a decline as the number of clusters increased, indicating that some clusters may not have been well-separated. **DBSCAN** performed exceptionally well for low eps values, but as eps increased, the clustering became less distinct, and noise points appeared, which reduced the silhouette score.

* 1. **Model Comparison:**

First, we trained the model using K-means based only on the longitude and latitude. The silhouette score was very low (0.39). then, after training the model with more features (review\_count, stars), the silhouette score went up to 0.89.

Using the elbow method, the optimal K was 5 clusters. However, after getting the silhouette score for a number of clusters, we saw that the best score was for 2 clusters with a silhouette score of 0.8914.

K-Means was more stable and provided reasonable results across different numbers of clusters, with the silhouette score decreasing as the number of clusters grew. On the other hand, DBSCAN with eps = 0.01 yielded the highest silhouette score (0.9974), but it is more sensitive to the choice of eps, making it more prone to variability, particularly in larger datasets. Overall, DBSCAN showed superior performance with a higher silhouette score at a specific eps value, while K-Means provided a more consistent clustering experience.

1. **Conclusion and Future Work:**
   1. **Summary of Findings:**

This project successfully developed a restaurant recommendation system using DBSCAN and K-Means clustering algorithms. DBSCAN demonstrated superior performance with a silhouette score of 0.9974, effectively identifying clusters and noise points. K-Means provided consistent and scalable clustering with a silhouette score of 0.89 for two clusters.

* 1. **Limitations:**  
     1. DBSCAN's performance was sensitive to the choice of parameters (eps and min\_samples), requiring careful tuning for optimal results.  
     2. K-Means assumed spherical clusters, limiting its ability to detect arbitrarily shaped groups.  
     3. The dataset focused only on restaurants in Las Vegas, limiting the generalizability of the findings to other locations.  
     4. Computational resources were a constraint for DBSCAN on larger datasets due to its complexity.
  2. **Future Work:**  
     1. Integrating user-specific features, such as cuisine preferences and budget, to enhance the personalization of recommendations.  
     2. Exploring hybrid models that combine clustering with collaborative filtering or supervised learning.  
     3. Expanding the dataset to include multiple cities, enabling broader applicability.  
     4. Incorporating text analysis from user reviews to better understand qualitative aspects like ambiance and service quality.  
     5. Optimizing DBSCAN for large-scale datasets by leveraging parallel processing or approximate nearest neighbor algorithms.  
       
     By addressing these limitations and implementing the proposed future work, the system can achieve higher accuracy, scalability, and user satisfaction.

1. **Reproducibility Instructions:**  
     
   To ensure the project is reproducible:  
   1. **Environment Setup**:  
      - Python version: 3.x  
      - Required libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, etc. (List detailed versions in requirements.txt.)
   2. **Execution Instructions:**  
      1. Download the dataset from the Yelp Academic Dataset website.  
      2. Place the dataset in the data/ folder.  
      3. Run the notebook Location\_based\_Recommendation\_with\_DBSCAN.ipynb in order.

Google colab:

* DBSCAN:  
  <https://colab.research.google.com/drive/1Y98JGCkRbdVrfNXGZomZHemvAR0nHW0c?usp=sharing>
* K-means:

<https://colab.research.google.com/drive/1YL44E0DpkpHJDcIYr7P38uuBRKeU9LWU?usp=sharing>

* K-means (original):

<https://colab.research.google.com/drive/1GpqZnV0KxHhA9fsw2vOoyV30ebdCzAB?usp=sharing>