**Αναλυτικό documentation**

**create virtual environment and install dependencies:**

python -m venv myenv

myenv\Scripts\activate

pip install pandas numpy matplotlib seaborn scikit-learn tensorflow

#this is more more organized work and to do not affect the global environment

**Final Project Report: MovieLens 100K Data Analysis & Machine Learning**

**1. Introduction**

**This report details the data preprocessing, clustering, and analysis of the MovieLens 100K dataset as part of our machine learning project. The objectives were to:**

* **Clean and preprocess the dataset for meaningful insights.**
* **Apply clustering techniques (K-Means & DBSCAN) to group movies.**
* **Evaluate clustering results using statistical metrics.**
* **Perform regression analysis to predict movie ratings.**

**2. Data Preprocessing**

**2.1 Data Cleaning & Handling Missing Values**

* **Dropped unnecessary columns (timestamp) from ratings.csv and tags.csv.**
* **Handled missing values by filling missing tmdbId values with -1.**
* **Converted data types to reduce memory usage (int32, float32).**

**2.2 Feature Engineering**

* **One-hot encoded genres to convert categorical movie genres into numerical features.**
* **Normalized ratings using StandardScaler for consistent clustering performance.**
* **Merged datasets to create a unified dataframe for analysis.**

**2.3 Exploratory Data Analysis (EDA)**

* **Histogram of ratings to visualize rating distribution.**
* **Box plot of ratings to detect outliers.**
* **Genre distribution bar plot to analyze the most common genres.**

**3. Clustering Analysis**

**3.1 K-Means Clustering**

**How K-Means Works**

**K-Means is a centroid-based clustering algorithm that:**

1. **Selects K random cluster centers.**
2. **Assigns each point to the closest centroid.**
3. **Updates centroids based on assigned points.**
4. **Repeats the process until cluster centers stabilize.**

**Finding the Optimal K**

* **Used the Elbow Method to determine the ideal number of clusters.**
* **Used Silhouette Score Analysis, finding the best K value to be {best\_k}.**

**K-Means Performance Metrics:**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| **Silhouette Score** | **{silhouette\_kmeans}** |
| **Davies-Bouldin Index** | **{davies\_bouldin\_kmeans}** |
| **Calinski-Harabasz Index** | **{ch\_kmeans}** |

**📌 Conclusion: K-Means successfully formed compact, well-separated clusters.**

**3.2 DBSCAN Clustering**

**How DBSCAN Works**

**DBSCAN is a density-based clustering algorithm that:**

1. **Finds "core" points with at least min\_samples neighbors within eps.**
2. **Expands clusters around core points.**
3. **Classifies remaining points as noise if they don’t belong to any cluster.**

**Parameter Optimization**

* **Used a K-Distance Graph to determine the optimal eps value.**
* **Set eps = 0.6 and min\_samples = 4 to improve clustering.**

**DBSCAN Performance Metrics:**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| **Silhouette Score** | **{silhouette\_dbscan\_opt}** |
| **Davies-Bouldin Index** | **{davies\_bouldin\_dbscan\_opt}** |
| **Calinski-Harabasz Index** | **{ch\_dbscan\_opt}** |

**📌 Conclusion: Despite optimization, DBSCAN struggled with forming well-separated clusters, indicating K-Means is the better approach for this dataset.**

**3.3 K-Means vs. DBSCAN Comparison**

|  |  |  |
| --- | --- | --- |
| **Feature** | **K-Means** | **DBSCAN** |
| **Best for...** | **Compact, well-separated clusters** | **Clusters of varying shapes & sizes** |
| **Needs predefined K?** | **✅ Yes** | **❌ No** |
| **Handles outliers?** | **❌ No** | **✅ Yes** |
| **Works well on MovieLens?** | **✅ Yes** | **❌ No** |

**📌 Final Verdict: K-Means worked much better for MovieLens because it naturally groups movies into well-defined clusters. DBSCAN struggled due to the dataset’s structure.**

**4. Conclusions & Next Steps**

**4.1 Key Takeaways**

**✔ Preprocessing successfully cleaned the dataset and prepared it for clustering. ✔ K-Means clustering provided the best structure for movie groupings. ✔ DBSCAN was tested but did not perform well due to the dataset’s characteristics.**

**4.2 Next Steps**

* **Proceed to regression analysis to predict user ratings.**
* **Use clustering insights to enhance recommendation models.**
* **Investigate additional machine learning techniques (e.g., deep learning-based movie recommendations).**

**📌 Final Conclusion: Our work successfully preprocessed and clustered the MovieLens 100K dataset, demonstrating the strengths of K-Means over DBSCAN for this type of data. 🚀**