**Name: Nikhil Sanjay Shinde**

**Roll No: TYITB120**

**Assignment No. 3**

**AIM**

To apply K-Means Clustering on a real-world loan prediction dataset and analyze patterns among loan applicants.

**OBJECTIVES**

* To understand the concept and working of the K-Means clustering algorithm.
* To preprocess and analyze a real-world dataset (Loan Prediction Dataset).
* To implement K-Means clustering using Python and visualize clusters.
* To evaluate how different values of K affect clustering outcomes using the Elbow Method.
* To compare K-Means with Hierarchical Clustering in terms of scalability and interpretability.

**THEORY**

K-Means is a widely used unsupervised machine learning algorithm that divides a dataset into K distinct non-overlapping clusters based on feature similarity. It is iterative, relying on centroid initialization and Euclidean distance to form groups.

In this assignment, we apply K-Means on a Loan Prediction dataset that contains demographic, income, and loan-related details of applicants. The goal is to identify clusters of similar applicants based on features like income, loan amount, credit history, and more.

**ALGORITHM STEPS**

1. **Initialization**: Select K initial cluster centers (centroids).
2. **Assignment**: Assign each applicant to the nearest centroid.
3. **Recalculation**: Recompute centroids as the mean of assigned points.
4. **Iteration**: Repeat the assignment and update steps until convergence.

**DATASET DETAILS**

**Filename**: loan\_prediction.csv

**Important Features:**

* ApplicantIncome: The income of the applicant.
* CoapplicantIncome: The income of the co-applicant.
* LoanAmount: Loan amount applied for.
* Loan\_Amount\_Term: Loan term in months.
* Credit\_History: Credit history of the borrower (1 = good, 0 = bad/missing).
* Property\_Area: Region of the property (Urban, Semiurban, Rural).

**PREPROCESSING**

* Missing values were handled using median/mode imputation.
* Categorical columns were label-encoded.
* Numerical columns were normalized using StandardScaler to prevent dominance by higher-range features.
* Dimensionality reduction using PCA was used for 2D visualization.

**CENTROID INITIALIZATION**

We used the k-means++ initialization method provided by scikit-learn for optimal centroid placement.

**DETERMINING THE VALUE OF K**

We applied the **Elbow Method** to determine the optimal number of clusters:

* K values from 1 to 10 were tested.
* Inertia (sum of squared distances) was plotted.
* The elbow point was visually identified as **K = 3**, which provided a balance between compactness and complexity.

**K-MEANS vs. HIERARCHICAL CLUSTERING**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **KMeans** | **Hierarchical** |
| **Scalability** | Efficient for 600+ entries | Less efficient as size grows |
| **Reproducibility** | Varies with initialization | Deterministic |
| **Flexibility in K** | K must be defined beforehand | Dendrogram helps choose K visually |

**VISUALIZATION**

After applying PCA for dimensionality reduction, clusters were visualized in a 2D scatter plot using color to indicate predicted cluster labels.

**CONCLUSION**

* Applying K-Means clustering to the Loan Prediction dataset helped uncover patterns among loan applicants.
* Clusters showed groupings based on income, credit history, and loan amounts.
* The best K was found to be **5**, indicating clusters possibly representing:
  + High-income approved
  + Low-income approved
  + High-risk rejected applicants, etc.
* Compared to hierarchical clustering, K-Means was significantly faster and better for this dataset.

**REFERENCE**

* <https://www.geeksforgeeks.org/k-means-clustering-introduction/>

**GITHUB REPOSITORY**

* <https://github.com/vishwatej2711/ML_Lab_assignment73/blob/main/third_new.ipynb>