Project Documentation

Project Title
Credit Risk Segmentation Using KMeans Clustering

Author
Ritanath Malakar

Project Overview
This project focuses on segmenting credit applicants using unsupervised machine learning to identify customer clusters based on financial behavior and demographic attributes. It aids in understanding risk profiles without requiring labeled data.

Objectives
- • Identify distinct groups of credit applicants (e.g., low-risk, high-risk)
- • Apply clustering to assist in better decision-making for credit approvals
- • Visualize clusters to interpret data patterns
- • Create an easy-to-use interface using Streamlit and Gradio

Technologies Used
- • Python
- • Pandas, NumPy – data processing
- • Scikit-learn – clustering and preprocessing
- • Matplotlib, Seaborn – data visualization
- • Joblib – model serialization
- • Gradio – alternative web-based UI

Dataset
Source: German Credit Risk Dataset

Attributes Used: Age, Sex, Job, Housing, Saving accounts, Checking account, Credit amount, Duration (months), Purpose

Missing values were handled by filling with 'unknown' or median imputation.

Model Architecture

Step 1: Preprocessing

- • Categorical encoding with LabelEncoder
- • Scaling with StandardScaler
- • Null value imputation

Step 2: Clustering

- • Applied KMeans with n_clusters = 3
- • Clusters represent: 0 - Low-risk, 1 - Medium-risk, 2 - High-risk

Step 3: Dimensionality Reduction

- • Used PCA to reduce features to 2D for visual clustering

Evaluation

As this is an unsupervised model, we used:

- • Silhouette Score = ~0.40 – indicates moderately good cluster separation and internal cohesion.

Visualization

Clusters were plotted using PCA-reduced components, colored by cluster labels. This visual helped in analyzing risk-based grouping of applicants.
https://8555d6307a33f3b581.gradio.live/ (Valid for 7 days)

Web Applications

🔷 Streamlit App: A clean interface to enter applicant details and predict which credit risk cluster they belong to.

Launch:
streamlit run streamlit_app.py

🔶 Gradio App: Alternative interactive version with similar functionality, good for quick prototyping.

Launch:
python gradio_app.py

Why This Model is Optimal

- • KMeans performed best compared to KMedoids and DBSCAN due to: simpler convergence, better scalability
- • Higher Silhouette Score
- • Easy deployment with lightweight models
- • Interpretable clusters for business decisions
- • Seamless integration with web interfaces

Project Structure

credit-risk-clustering/
├── german_credit_data.csv
├── training_model.py
├── streamlit_app.py
├── gradio_app.py
├── scaler.pkl
├── kmeans_model.pkl
├── encoders.pkl
├── clustered_credit_data.csv
└── README.md

Future Enhancements

- • Replace KMeans with advanced clustering like Gaussian Mixture Models (GMM) or Agglomerative Clustering
- • Integrate historical repayment or loan default labels (if available)
- • Add cluster explanation via SHAP/LIME
- • Deploy as a web service (API + DB)

About the Author

Ritanath Malakar
A passionate developer and data science enthusiast working on real-world applications of machine learning and system design.