Customer Segmentation Project Report

1. Introduction & Problem Statement

Customer segmentation is a key strategy in marketing that helps businesses understand their customers better. It involves grouping customers into different categories based on shared characteristics, such as age, income, and spending habits. This allows companies to:

- Personalize marketing campaigns
- Improve customer satisfaction
- Enhance product and service delivery
- The goal of this project is to build a **customer segmentation model** that groups customers based on **Age, Annual Income, Spending Score**

We are using the **'Mall Customers' dataset**, which contains customer information collected by a mall.

2. Data Exploration & Pre-processing

Before building the model, we need to prepare the dataset for analysis. The dataset contains the following key features:

- **Age**: The age of the customer (integer).
- **Annual Income (k$)**: The customer's income per year, measured in thousands of dollars.
- **Spending Score (1-100)**: A rating given to the customer based on their shopping behaviour, where a higher score means they spend more frequently.

To ensure our model works effectively, we applied several pre-processing steps:

1. **Data Cleaning**
    - Checked for missing or null values.
    - Removed any inconsistencies to keep the data reliable.
2. **Feature Selection**
    - Only selected the relevant columns (**Age, Income, Spending Score**) to ensure meaningful clustering.
3. **Data Scaling**
    - Used **StandardScaler**, which standardizes the data so that all features have a mean of 0 and a standard deviation of 1.
    - This is important because clustering algorithms like **KMeans** perform better when all features are on a similar scale.

These pre-processing steps help improve the accuracy and efficiency of the clustering model.

3. Model Implementation & Evaluation

To achieve effective customer segmentation, we implemented a combination of **Autoencoders** and **KMeans Clustering**. Below are the details of each model:

- **Autoencoder Model:**
  - An autoencoder is a type of neural network used for data compression and feature extraction. It learns a lower-dimensional representation of the input data through encoding and attempts to reconstruct the original data through decoding.
  - The architecture used includes an input layer, two encoding layers, and two decoding layers.
  - The encoded data (compressed representation) is extracted and passed to the clustering model for further processing.
  - The model was trained using **Mean Squared Error (MSE)** as the loss function and **Adam Optimizer** for faster convergence.
  - Training was performed over 100 epochs with a batch size of 8.

- **KMeans Clustering:**
  - The KMeans algorithm is applied to the encoded data generated by the Autoencoder.
  - The model attempts to group customers into clusters based on similarity, with the number of clusters being adjustable by the user.
  - Two important metrics used for evaluation:
    - **Silhouette Score:** Measures the quality of clustering. Higher values indicate well-separated clusters.
    - **Inertia (WCSS):** Measures the compactness of clusters. Lower values indicate better-defined clusters.
  - The trained KMeans model is saved using the `pickle` library for future use.

---

4. Results & Insights

The Customer Segmentation App provides the following insights:

- **Visualization of Clusters:** Scatter plots are generated to visualize customer groups based on their Annual Income and Spending Score. Each cluster is color-coded to provide clarity in differentiation.
- **Cluster Summary:** A summary table is displayed with the number of customers in each cluster, along with average Age, Annual Income, and Spending Score for each group.
- **Metrics Displayed:**
  - **Silhouette Score:** A higher score indicates well-defined clusters.
  - **Inertia:** Lower values signify better clustering performance.

5. Challenges & Future Improvements

- **Challenges Faced:**
  - The Autoencoder model may suffer from overfitting if not properly tuned.
  - Difficulty in handling overlapping clusters where boundaries are not well defined.
  - Limited interpretability of deep learning models compared to simpler algorithms.
- **Future Improvements:**
  - Enhancing the Autoencoder architecture by experimenting with different activation functions, layer sizes, and training methods.
  - Implementing other clustering techniques like **DBSCAN or Agglomerative Clustering** for comparison.
  - Improving the user interface to allow better data visualization and analysis.
  - Adding functionality to allow users to upload their own datasets and visualize their clusters effectively.
  - Allow users to upload datasets with different features (not just Age, Income, etc.) and still achieve good segmentation results.

This project successfully demonstrates how deep learning (Autoencoders) and unsupervised learning (KMeans Clustering) can be combined to achieve effective customer segmentation. The application provides a user-friendly interface developed using Streamlit, allowing interactive exploration of clustering results.