Report on Machine Learning Projects

1. Customer Segmentation Using Clustering (K-Means & Agglomerative Clustering)

Approach:

In this task, the goal was to segment customers into distinct groups based on their purchasing behavior. The dataset provided included columns like Age, Annual Income, and Spending Score, and we performed the following steps:

Data Preprocessing:

We removed the CustomerID column, which was irrelevant for clustering.

Categorical data such as Gender was converted into numerical values (Male = 0, Female = 1).

The data was scaled using StandardScaler to ensure the features were on the same scale, which is critical for distance-based clustering algorithms like K-Means.

Clustering:

K-Means Clustering: We used the Elbow Method to determine the optimal number of clusters. After plotting the Within-Cluster Sum of Squares (WCSS), we decided on 5 clusters.

Agglomerative Clustering: We applied hierarchical clustering as a comparison. This method was implemented using the Ward linkage and Euclidean distance as the metric.

Visualization:

We applied PCA (Principal Component Analysis) for dimensionality reduction and visualized the clusters in a 2D plot.

Challenges:

Data Cleaning: Handling missing values and ensuring all features were numeric was an initial challenge.

Optimal Number of Clusters: Determining the right number of clusters was not straightforward. The Elbow Method provided some insight, but it was still a judgment call to select 5 clusters.

Model Performance & Improvements:

K-Means performed well, as the clusters were clearly distinguishable in the PCA plot. However, more advanced techniques like Silhouette Analysis could have been used to validate the clusters further.

Improvements: A different scaling technique or feature engineering (e.g., adding interaction terms) might have improved clustering results. Also, using DBSCAN could provide more flexibility by not requiring the number of clusters to be set.

2. Fake News Detection Using Machine Learning

Approach:

The goal was to create a model that could classify news articles as real or fake. The dataset provided included labeled articles, and we followed these steps:

Data Preprocessing:

Text cleaning: Removal of stopwords, special characters, and tokenization.

Vectorization: We used TF-IDF to convert text data into numerical vectors for the model.

Model Selection:

We experimented with several models: Naïve Bayes, Random Forest, and LSTM (Long Short-Term Memory networks).

After evaluating model performance, we settled on using Naïve Bayes due to its simplicity and strong performance with text classification tasks.

Evaluation:

We measured model performance using metrics like accuracy, precision, recall, and F1-score.

Challenges:
Text Preprocessing: Handling text data posed challenges, such as dealing with noisy data and ensuring consistent tokenization.

Model Tuning: Tuning hyperparameters like the number of features for TF-IDF and adjusting parameters for models like Naïve Bayes required several iterations.

Model Performance & Improvements:
Naïve Bayes achieved decent accuracy (around 80-85%), but it could be improved by fine-tuning the hyperparameters or using more sophisticated models like LSTMs.

Improvements: Using deep learning models like BERT (Bidirectional Encoder Representations from Transformers) could improve performance significantly. Additionally, ensemble methods could be explored for combining the strengths of different models.

and robustness in each task.