Student Name: Muhammad Wahaj

Brief Description of Machine Learning Tasks (ELEVVO)

Task 1: Student Score Prediction

In this task, the objective was to build a regression model to predict students' exam scores based on the number of hours they studied. The dataset contained two columns: Hours_Studied and Exam_Score. The approach involved reading the dataset using Pandas, performing basic exploration, visualizing the relationship between study hours and scores using Matplotlib, and then training a simple Linear Regression model using Scikit-learn.

1. Data Reading: Loaded only the necessary columns (Hours_Studied, Exam_Score) from the dataset using Pandas.
2. Data Exploration: Checked for missing values, verified data types, calculated correlation, and summarized statistics.
3. Visualization: Created a scatter plot to visualize the relationship between hours studied and exam scores.
4. Model Training: Used a Simple Linear Regression model from Scikit-learn to learn the relationship between study hours and scores.
5. Evaluation: Calculated evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score.

Results: The correlation between Hours_Studied and Exam_Score was found to be approximately 0.445, indicating a moderate positive relationship. The scatter plot confirmed a general upward trend. Both Linear and Polynomial Regression models produced almost similar results, with an R^2 score of 0.23 and MAE of 2.45. Since Polynomial Regression did not improve performance, Linear Regression is preferred for its simplicity and comparable accuracy.
These results suggest that study hours alone moderately explain exam scores, and incorporating additional features could improve the model's predictive accuracy.

Task 2: Customer Segmentation

This task aimed to segment mall customers into distinct groups based on their annual income and spending score. The dataset(Mall Customer) was preprocessed to select only the relevant features. Two clustering methods were implemented:
K-Means Clustering: The optimal number of clusters was determined using the Elbow Method, and customer groups were visualized in a scatter plot.
DBSCAN: A density-based clustering method that identifies core clusters and outliers without requiring a predefined number of clusters.

Results: K-Means successfully segmented the customers into distinct groups based on Income and Spending Score, identifying clear behavioral patterns. However, DBSCAN treated high-income customers with unusual spending behaviors as outliers (noise), failing to form

meaningful clusters due to the dataset's uniform density. This shows K-Means is more suitable for segmenting general customer groups, while DBSCAN was able to detect outliers and handle non-linear cluster shapes.

Task 3: Forest Cover Type Classification :

This task aimed to the Forest Cover Type Classification problem using two machine learning algorithms: Random Forest and XGBoost. The dataset(Covertype UCL) was split into training and testing sets, and both models were initially trained with default parameters to establish baseline performance. Random Forest achieved better accuracy and confusion matrix results compared to XGBoost in the initial run. Next, we applied hyperparameter tuning using RandomizedSearchCV, adjusting parameters such as n_estimators, max_depth, and learning_rate to optimize performance. Post-tuning, both models showed slight improvements in prediction accuracy and class-wise distribution, with Random Forest still outperforming XGBoost. Starting with default models and then applying hyperparameter tuning via RandomizedSearchCV. Initially, Random Forest outperformed XGBoost with higher accuracy and a better confusion matrix. After tuning parameters like n_estimators, max_depth, and learning_rate, both models showed slight improvements, but Random Forest remained superior in accuracy and class-wise predictions. The overall impact of tuning was marginal, indicating that Random Forest's ensemble nature is better suited for this dataset's binary and numerical features. So, Random Forest is the recommended model for this task.

Task 4: Music Genre Classification

This task involved building two different approaches to classify music genres from audio dataset(GTZAN). The first approach used tabular data containing pre-extracted audio features (e.g., spectral centroid, zero-crossing rate, chroma frequencies). The data was cleaned, scaled, and passed to a Random Forest classifier for training and evaluation. This method provided good accuracy with minimal computational cost.

The second approach used image data in the form of spectrograms generated from audio files. Pre-trained CNN models (VGG16 and EfficientNetB0) were applied using transfer learning. The models were trained, fine-tuned, and evaluated on a validation set. EfficientNetB0, after fine-tuning, achieved the highest accuracy.

Results: The image-based CNN approach outperformed the tabular Random Forest model in accuracy but required more computational resources and time. The tabular approach (using extracted features) outperformed the image-based spectrogram approach, achieving a higher accuracy of 68.5% compared to 55.3% from image-based validation. The tabular model also showed strong class-wise performance, especially in genres like classical, metal, and pop, with f1-scores above 0.75. This suggests that numerical features like tempo, spectral bandwidth, and chroma can effectively capture genre-specific patterns. In contrast, the image-based model struggled likely due to data size, or insufficient spectrogram resolution. Tabular models are

computationally lighter and offer better interpretability, making them advantageous for smaller or feature-rich datasets.

Task 5: Traffic Sign Classification using Custom CNN and MobileNetV2

In this task, the German Traffic Sign Recognition Benchmark (GTSRB) dataset was used to classify traffic signs. Two different models were implemented and compared: a Custom Convolutional Neural Network (CNN) and MobileNetV2 with transfer learning. Data preprocessing involved resizing images, normalizing pixel values, and one-hot encoding labels. The dataset was split into training and validation sets. The Custom CNN was built from scratch with convolutional, pooling, and fully connected layers, while MobileNetV2 leveraged pre-trained ImageNet weights with a fine-tuned classifier head. Both models were trained and evaluated on the same dataset split.

Results: Showed that the Custom CNN achieved a test accuracy of 98%, outperforming MobileNetV2, which achieved 94% accuracy. The confusion matrix and classification report provided detailed performance metrics for each class. This indicates that the Custom CNN was better suited for this dataset, possibly due to its architecture being optimized for the specific characteristics of traffic sign images, while MobileNetV2, being a general-purpose model, performed slightly lower.