**AI/ML Practical Assessments Summary**

**Assessment 1: EDA and Clustering Techniques**

The primary purpose of this project is to apply Exploratory Data Analysis (EDA) and unsupervised clustering techniques to customer data to uncover meaningful patterns in consumer behavior. Using a dataset containing customer attributes like gender, annual income, and spending score, the project aims to group similar customers together—helping businesses understand their audience and make data-driven decisions in marketing, product placement, and customer engagement strategies. The first part of the project involves thorough EDA, where the distribution of income and spending is visualized using scatter plots, box plots, and violin plots. These visualizations help highlight trends, outliers, and gender-based differences in spending behavior. This step is critical for building intuition about the data and selecting appropriate features for modeling. Following EDA, the project applies KMeans and Hierarchical Clustering (Agglomerative Clustering) to group customers based on their income and spending patterns. The clusters aim to segment customers into distinct groups—such as high-income high-spending, low-income low-spending, or average shoppers—without any prior labels. This segmentation allows targeted strategies in promotions or personalized offers. To make the clustering insights interactive and accessible, a Gradio web application is built. It enables users to input their income and spending score using sliders, and instantly view which cluster they belong to under both KMeans and Hierarchical methods. This interface enhances the usability of the model and demonstrates how clustering can be applied in real-time to new customer data.

**Assessment 2: California Housing Dataset**

In the analysis part of this assessment, we explored regression techniques on both structured and unstructured data. For the California Housing dataset, we applied Linear Regression after preprocessing steps like feature selection (SelectKBest), feature scaling (StandardScaler), and data splitting. We evaluated the model using MSE, MAE, and R² score, and visualized feature importance through coefficient plots. We then used Lasso Regression with L1 regularization and GridSearchCV to tune the alpha parameter, which helped in automatically selecting the most relevant features by shrinking others to zero.

In the second part, we implemented Logistic Regression for a spam email classification task. Text data was converted into numerical features using CountVectorizer with unigrams and bigrams. After training and testing the model, we evaluated it using accuracy score and built an interactive Gradio interface to classify user-input emails as spam or not. Overall, this analysis helped us understand how regression techniques can be applied effectively to both numeric and text-based data.

**Assessment 3: Diabetes Prediction using Decision Trees in AI**

Based on the analysis part we’ve completed, the purpose of this assessment is to help beginners understand the practical implementation of regression and classification models using Python and machine learning libraries. By working through different examples—such as predicting house prices using Linear and Lasso Regression, detecting spam emails using Logistic Regression, and predicting diabetes using a Decision Tree Classifier—the assessment introduces how different models can be applied to structured data (like numerical features) and unstructured data (like text). It also covers essential steps such as data preprocessing, model training, evaluation, feature selection, regularization, and even deploying interactive prediction interfaces using Gradio. Overall, the goal is to build a solid foundational understanding of how machine learning algorithms are used to make predictions and support decision-making in real-world scenarios.

**Assessment 4: MBA Analysis**The goal of this project is to explore and demonstrate how Market Basket Analysis (MBA) can be effectively used to discover meaningful associations between items purchased together in transaction data—particularly groceries in this case. Using the Apriori algorithm, the project identifies frequent item combinations and generates association rules that help explain customer buying behavior in terms of "if-then" relationships (e.g., if a customer buys bread, they are likely to buy milk). Initially, the original dataset had structural issues that prevented proper MBA execution. To overcome this, a new sample dataset was created based on the original format, grouping items by transaction (via member number and date) and converting them into a one-hot encoded structure, which is required by the Apriori algorithm. Once this format was established, the project successfully identified frequent itemsets and generated association rules based on configurable parameters such as minimum support and minimum confidence. To enhance usability and interaction, a Gradio web application was built, allowing users to dynamically adjust these parameters and view the resulting association rules instantly. This interactive interface makes it easier to understand how different combinations of items co-occur, how strong their relationships are (confidence), and how much more likely they are to appear together than by random chance (lift). Overall, this project showcases how unsupervised learning techniques like Apriori and tools like Gradio can help retailers optimize product placement, cross-sell strategies, and personalized recommendations—ultimately enhancing both customer satisfaction and business profitability.

**Assessment 5: Anomaly Detection**   
The primary purpose of this project is to explore and apply unsupervised machine learning techniques—specifically clustering and anomaly detection—to detect potentially fraudulent credit card transactions. This project simulates a more realistic approach where we aim to identify suspicious patterns without needing prior knowledge of what constitutes fraud. The project begins by loading and preprocessing the dataset to scale the features appropriately and prepare it for analysis. We then perform exploratory data analysis (EDA) to understand the distribution of fraud and non-fraud transactions, identifying key features that may indicate fraudulent behavior. This is followed by clustering using KMeans and Hierarchical Clustering to group transactions based on similarity and examine if fraud patterns emerge naturally in any clusters. The core of the project is in Step 5, where we implement anomaly detection techniques like Isolation Forest and Z-Score. These models are applied to a sample of the data to flag transactions that deviate significantly from normal behavior. We evaluate these models using confusion matrices and classification metrics to compare their ability to balance false positives and false negatives. Finally, we use PCA-based visualizations to illustrate how these methods separate normal and potentially fraudulent transactions in a 2D space. Ultimately, the goal of the project is to build a foundational understanding of how unsupervised models can be used in fraud detection systems—where anomalies, not labels, often drive action—and to provide interpretable, visual, and measurable insights that support intelligent financial decision-making.

**Assessment 6: OpenCV preprocessing**

In this assessment, we explored fundamental image preprocessing techniques using OpenCV to understand how raw visual data can be prepared for computer vision tasks. The analysis involved converting a color image to grayscale, adjusting brightness and contrast, and applying histogram equalization for contrast enhancement. We implemented multiple thresholding methods—global, Otsu's, and adaptive—to binarize images effectively. Edge detection was performed using Sobel, Prewitt, Laplacian, and Canny operators to identify boundaries and gradients. Additionally, we applied feature detection techniques including Harris Corner Detection, Blob Detection, and SIFT to extract key points from images. The purpose of this assessment is to introduce beginners to essential preprocessing steps in computer vision, enabling them to understand how images are transformed and analyzed to support more complex tasks such as object recognition, segmentation, and tracking.

**Assessment 7: Skin Disease Text Classification**

In this assessment, we worked on a text classification problem using a custom dataset consisting of text descriptions categorized by different skin diseases. The analysis began by extracting text files from class-specific folders and organizing them into a structured DataFrame with Text and Label columns. We performed preprocessing steps such as tokenization using Keras’s Tokenizer, sequence padding for uniform input length, and label encoding to convert categorical disease names into numerical values. We split the dataset into training and testing sets and used these preprocessed inputs to train a deep learning model using PyTorch. The model architecture was based on an LSTM (Long Short-Term Memory) network, featuring an embedding layer, multiple LSTM layers, a dropout layer for regularization, and a fully connected output layer with LogSoftmax activation. The model was trained over 40 epochs using the Adam optimizer and evaluated using accuracy and loss metrics on a held-out test set. A custom Dataset class and DataLoader were used for efficient batch processing, and performance tracking revealed the model’s predictive capabilities and its tendency to favor certain dominant classes.

The purpose of this assessment is to provide hands-on experience in applying sequence-based deep learning techniques for multi-class text classification. Specifically, the task demonstrates how to handle domain-specific unstructured data (skin disease text records), prepare it for neural networks, and implement an end-to-end training and evaluation pipeline using PyTorch. Although the model used here was LSTM, a GRU (Gated Recurrent Unit) could also be used as a more efficient alternative. GRUs simplify the LSTM architecture by combining the forget and input gates into a single update gate, resulting in fewer parameters and faster training, while often achieving comparable performance—especially beneficial for smaller datasets or limited computational resources. Integrating GRU into this workflow would follow a similar structure but with a different recurrent unit, offering learners a valuable perspective on optimizing model complexity and performance in NLP applications.

**Assessment 8: Hotel Reviews**

In this assessment, we focused on natural language preprocessing for multilingual text data using Python and the Hugging Face Transformers library. The analysis began by importing and utilizing nltk resources for stopword handling and tokenization, along with unicodedata for normalizing text. A BERT-based tokenizer (bert-base-multilingual-cased) was used to tokenize Devanagari-script input, making it suitable for modern multilingual NLP tasks. We also created a custom list of Hindi stopwords to filter out common but semantically weak words. The text was normalized, tokenized, cleaned by removing stopwords, and converted to lowercase to ensure consistency in further processing.

The purpose of this assessment is to provide hands-on experience in building a robust NLP preprocessing pipeline tailored for multilingual datasets. It highlights key preprocessing steps—like Unicode normalization, stopword removal, and tokenization with BERT—which are critical for preparing raw text data for downstream tasks such as sentiment analysis, classification, or language modeling. This assessment helps learners understand the importance of linguistic and script-specific processing when dealing with diverse language inputs in real-world NLP applications.

**Assessment 9: NER**

Named Entity Recognition (NER) is a foundational task in Natural Language Processing (NLP) that focuses on identifying and classifying entities such as names, dates, medical terms, locations, and other domain-specific items within unstructured text. Although I haven’t worked on the analysis part of this assessment yet, the provided setup indicates a plan to train a custom NER model using spaCy to recognize medical-related entities like patient IDs, disease history, and procedures from clinical notes. This involves preparing annotated data, defining custom entity tags, training the model over several epochs, and eventually deploying it with an interactive interface like Gradio for real-time predictions.

Understanding NER has strong real-world relevance. In healthcare, for instance, it can automate the extraction of critical information from patient records, enabling faster documentation, decision support, and compliance monitoring. My understanding of NER—especially how models are trained, tested, and deployed—can be applied in industries like healthcare, legal, and finance where structured data extraction from text is a high-value task. Once I begin the analysis, I expect to work through data annotation handling, model pipeline customization, and evaluation of entity recognition performance, all of which will contribute to building a domain-specific, intelligent text-processing application.

**Assessment 10: Hotel Reviews – GPT**

In this assessment, we performed an end-to-end fine-tuning and deployment of a GPT-2 language model using a dataset of preprocessed hotel reviews. The analysis began with loading the dataset and converting it into the Hugging Face Dataset format, followed by an 80-20 train-test split. We then tokenized the review texts using the GPT-2 tokenizer with truncation and padding. The core of the analysis involved fine-tuning the GPT-2 model using the Trainer API, configuring training arguments such as batch size, learning rate, and number of epochs. After training the model on the custom review data, we saved the model and tokenizer and built a text generation pipeline using Hugging Face's pipeline() utility. Finally, we developed a Gradio-based web application that allows users to input prompts and generate human-like hotel reviews using the trained model, with adjustable settings like max length, creativity (temperature), and number of samples.

The purpose of this assessment is to provide practical experience in fine-tuning transformer-based language models (like GPT-2) for domain-specific text generation. It demonstrates the full lifecycle of a generative NLP project—from data preprocessing and tokenization to model training, evaluation, and deployment. This exercise helps build real-world skills in working with pre-trained language models, customizing them for specific use cases (like hotel reviews), and deploying them through user-friendly interfaces. Such expertise is valuable in industries like travel, marketing, and customer engagement, where generating personalized, context-aware content is highly beneficial.

**Assessment 11: Text Feature Extraction Using Bag of Words and Frequency Analysis**

In this assessment, we focused on implementing and analyzing the Bag of Words (BoW) model for a small collection of sample text documents related to Natural Language Processing (NLP). The analysis began with basic text preprocessing, where each document was converted to lowercase, non-alphabetic characters were removed, and stopwords were filtered out using sklearn’s built-in stopword list. These cleaned texts were then converted into a numerical format using CountVectorizer, resulting in a BoW matrix that represents the frequency of each word across the documents. This matrix was transformed into a DataFrame for better visualization and interpretation. Additionally, a word cloud was generated to visualize the most prominent terms in the corpus, and a bar chart was created to display the top 20 most frequent words.

The purpose of this assessment is to introduce the concept and implementation of Bag of Words, a foundational technique in NLP used for converting text into numerical features for machine learning models. It emphasizes the importance of text cleaning, tokenization, and frequency-based vectorization for downstream tasks such as text classification, clustering, or topic modeling. This exercise helps build a clear understanding of how raw textual data can be transformed into structured input that models can process, which is essential for anyone working with natural language in data-driven applications.

**Assessment 12: Sentiment Analysis on Hotel Reviews Using Logistic Regression and TF-IDF**

In this assessment, we conducted a complete sentiment analysis on a dataset of hotel reviews by following an end-to-end natural language processing (NLP) workflow. The analysis began with text preprocessing, where we converted reviews to lowercase, removed punctuation and numbers, and filtered out stopwords. We then mapped numerical ratings to sentiment labels—positive, neutral, and negative—and used TF-IDF vectorization to convert the cleaned text into numerical features. A Logistic Regression model was trained on these features and evaluated using a classification report and confusion matrix to assess its performance. Finally, we deployed the model using Gradio, creating an interactive web interface that allows users to input hotel reviews and receive real-time sentiment predictions. The purpose of this assessment is to develop practical skills in preprocessing text data, extracting meaningful features, building a classification model, and deploying it for real-world applications like customer feedback analysis.

**Assessment 13: Text Extraction from Images Using EasyOCR and OpenCV**

In this assessment, we performed Optical Character Recognition (OCR) using an image input to extract and visualize textual content. The analysis started by reading and rotating the image using OpenCV to correct its orientation. We then applied EasyOCR, a deep learning-based OCR library, to detect and extract text from the image. Each recognized text segment was printed alongside its confidence score, and bounding boxes were drawn around the detected text to visually confirm accuracy. The processed image was then displayed using Matplotlib, showing both the recognized text and its location on the image.

The purpose of this assessment is to understand how OCR technology can be applied to extract structured text data from unstructured visual content, such as scanned documents, signs, or handwritten notes. This kind of automation is widely used in domains like healthcare (digitizing prescriptions), finance (invoice scanning), logistics (label reading), and accessibility tools. Through this project, learners gain hands-on experience with OCR workflows, image handling, and practical deployment using Python.