# PASS TASK (WEEK 4)

## About this task

**Step-1**

At the completion of week 3 and 4 modules, you are required to complete a lesson review to tell us what you learnt and how you learnt it by submitting evidence requested at the end of this file.

**Step-2**

Your tutor will then review your submission and will give you feedback. If your submission is incomplete the tutor will ask you to include missing parts. Tutor can also ask follow-up questions, either to clarify something that you have submitted or to assess your understanding of certain topics.

## Feedback and submission deadlines

**Feedback deadline:** Monday 21 April (No submission before this date means no feedback!)
**Submission deadline:** Before creating and submitting portfolio.

## Evidence of Learning

1. Submit a summary report (pdf format) in Ontrack (https://ontrack.deakin.edu.au)
    1.1. Summarise the main points that is covered in the week 3 and 4.
    1.2. Provide summary of your reading list – external resources, websites, book chapters, code libraries, etc.
    1.3. Reflect on the knowledge that you have gained by reading contents of the week 3 and 4 with respect to machine learning.
    1.4. Attempt the quiz given in weekly content (3.13 and 4.18) and add screenshot of your score (>85% is considered completion of this task) in this report.
2. Complete the following problem-solving task given in weekly content, and submit your code file (.ipynb) separately to OnTrack (https://ontrack.deakin.edu.au).

# Problem Solving Task

**Instructions:**
Ensure you provide well-structured code for each question. Clearly explain your reasoning where required and submit all necessary files, including the modified dataset after performing the required transformations.

---

1. **Data Preprocessing and Exploratory Analysis:**
   - Load the dataset ("Dataset.csv") and verify its integrity.
   - Confirm that there are no missing values.
   - Identify and analyze outliers using visualizations such as boxplots.
   - Visualize feature distributions with histograms and KDE plots to understand the overall distribution of each feature.
   - Review feature statistics (e.g., mean, standard deviation) to get insights into the data.
   - Normalize or standardize the dataset so that all features contribute equally in distance calculations, which is crucial for clustering.
2. **Impact of the Number of Clusters on KMeans Clustering with Euclidean Distance**
   - Apply KMeans clustering (using Euclidean distance) on the standardized dataset.
   - For a range of cluster numbers (e.g., from 1 to 10), compute the inertia (SSE) and plot these values to identify the “elbow” point.
3. **Evaluating the Stability of KMeans and KMeans++ Initialization**
   - Run KMeans clustering 50 times using two initialization methods:
     a) Standard random initialization.
     b) KMeans++ initialization.
   - Compute and compare the average inertia (SSE) and the Silhouette Score for each method over these iterations.
4. **Clustering Evaluation Using Purity and Mutual Information**
   - Use KMeans (with the optimal k from Question 2) to cluster the data. Assume the dataset contains a ground-truth label column (e.g., "label"). For each cluster, assign a label based on the majority class.
   - Evaluation Metrics: Compute and report the following:
     a) Purity Score: Measures how homogeneous each cluster is relative to the true labels.
     b) Mutual Information Score: Quantifies the mutual dependence between the clustering results and the true labels.
     c) Silhouette Score: Evaluates the clustering quality without reference to the ground truth by comparing intra-cluster cohesion versus inter-cluster separation.
5. **Principal Component Analysis (PCA) for Dimensionality Reduction**
   - Apply PCA to reduce the dataset to 4 principal components.
   - Plot the cumulative variance explained by the principal components and determine how many components are needed to retain 90% of the total variance.
   - Create a 3D scatter plot of the first three principal components.
6. **(Only for SIT720) Density-Based Clustering Using DBSCAN with Different Distance Metrics**

- Apply DBSCAN to the dataset twice:
    a) Once using Euclidean distance.
    b) Once using Mahalanobis distance.
- Determine the optimal values for eps (ε) and min_samples for each distance metric.
- Compare the clustering results from both distance metrics.

**7. Clustering Performance on PCA-Reduced vs. Full Dataset**

- Apply KMeans clustering to:
    a) The original standardized dataset.
    b) The PCA-transformed dataset (using the principal components from Question 5).
- Evaluate the clustering quality using the Silhouette Score.
- Compare whether the PCA-transformed dataset results in better-separated and more compact clusters relative to the full dataset.

**8. (Only for SIT720) Clustering Using t-SNE**

- Apply t-SNE (using the exact method) to reduce the dataset to 4 components.
- Create a 3D scatter plot of the first three t-SNE components.
- Apply KMeans clustering on the t-SNE–reduced data using an appropriate number of clusters (e.g., based on prior optimal k or an elbow method on the t-SNE output).
- Evaluate the clustering performance on the t-SNE–reduced data using metrics such as the Silhouette Score and compare these results to clustering on the original and PCA–transformed dataset.
- Discuss whether the clusters formed on the t-SNE–reduced data are more distinct and how well they correspond to the known data structure.