# Algorithm Comparison Assignment

## Part 1: Algorithm Overview

### 1. Logistic Regression

Logistic Regression models the probability of a binary outcome. It uses a sigmoid function to map a linear combination of input features to a probability between 0 and 1. It is primarily used for classification tasks.

- Strengths: Simple to implement and interpret; computationally efficient for large datasets.
- Limitations: Assumes a linear relationship between features and the log-odds of the outcome; struggles with non-linear data.

### 2. K-Nearest Neighbours (KNN)

KNN classifies a data point based on the majority class among its k-nearest neighbours in the feature space. It relies on a distance metric to determine proximity.

- Strengths: Simple to understand and implement; no explicit training phase.
- Limitations: Computationally expensive for large datasets; sensitive to irrelevant features and the scale of features.

### 3. Decision Tree

A Decision Tree creates a tree-like structure of decisions based on feature values. It recursively partitions the data into subsets based on feature thresholds to achieve homogenous groups regarding the target variable.

- Strengths:
    - Easy to visualize and interpret;
    - can handle both categorical and numerical data.
- Limitations:
    - Prone to overfitting, especially with complex trees
    - can be sensitive to small changes in the data.

### 4. Support Vector Machine (SVM)

SVM finds an optimal hyperplane that maximally separates data points of different classes in a high-dimensional space. It uses kernel functions to handle non-linear data by mapping it to a higher-dimensional space.

- Strengths: Effective in high-dimensional spaces; versatile due to different kernel functions for various data types.
- Limitations: Computationally intensive for large datasets; sensitive to the choice of kernel and hyperparameters.

## Part 2: Application Scenarios

### 1. High-Dimensional Data (e.g., text or gene expression data)

SVM is most suitable for high-dimensional data. SVMs are effective in spaces with many features and can use kernel tricks to efficiently handle complex relationships in these high-dimensional spaces without explicitly calculating the coordinates in that space. While other algorithms might struggle with the curse of dimensionality, SVMs are designed to handle it effectively.

### 2. Imbalanced Dataset (e.g., fraud detection, rare disease prediction)

Decision Tree is a good choice for imbalanced datasets. Decision trees focus on creating homogenous groups, and their performance is less affected by the skewed class distribution compared to algorithms like Logistic Regression, which can be biased towards the majority class. Techniques like cost-sensitive learning or ensemble methods with decision trees (e.g., Random Forest) can further improve performance.

### 3. Small Dataset with Many Features (e.g., medical or genetic data)

SVM is again suitable for small datasets with many features. SVMs are effective even when the number of features exceeds the number of samples. Their ability to find an optimal separating hyperplane in high-dimensional space makes them robust in these situations. Regularization techniques in SVM can also prevent overfitting, which is a concern with small datasets.

### 4. Non-linear Data Separation (e.g., complex shapes like spirals or circles)

SVM with a non-linear kernel (e.g., RBF kernel) is the best choice for non-linear data separation. The kernel trick allows SVM to map the data to a higher-dimensional space where it becomes linearly separable, effectively creating complex decision boundaries in the original space. While Decision Trees can also handle non-linearity, SVMs with appropriate kernels often provide better generalization.

### 5. Dataset with Noise (e.g., data with many irrelevant or misleading features)

Logistic Regression can be a reasonable starting point for datasets with noise, especially if feature selection/engineering is applied beforehand. While not inherently robust to noise, its simplicity and interpretability can be beneficial for understanding the impact of features. Decision Trees can be susceptible to overfitting on noisy features. SVM, while powerful, can also be negatively impacted if the noise significantly distorts the optimal separating hyperplane. Feature selection or dimensionality reduction techniques are generally recommended before applying any algorithm to noisy datasets.