Assignment: Algorithm Comparison

Objective:

The objective of this assignment is to help students understand the situations where specific machine learning algorithms—Logistic Regression, KNN, Decision Tree, and SVM—are most suitable. Students will explore the strengths, weaknesses, and suitability of each algorithm for different datasets.

Instructions:

Complete the following tasks by comparing the algorithms based on their characteristics, performance, and application scenarios. The assignment consists of 2 parts.

Part 1: Algorithm Overview

For each of the algorithms below, write a brief overview that includes:

- How the algorithm works.
- Two key strengths and two limitations.

Algorithms:

1. Logistic Regression
2. K-Nearest Neighbors (KNN)
3. Decision Tree
4. Support Vector Machine (SVM)

Expected Output:

- 50-75 words per algorithm.

Part 1: Algorithm Overview

1. Logistic Regression

Logistic Regression is used for classification (like yes/no, spam/not spam). It works by finding a line (or curve) that best separates the classes using probabilities.

- **How it works:** Fits a straight line to predict binary outcomes (e.g., yes/no).

Strengths:

- It is easy to understand and use, and it operates quickly and efficiently.
- Works well with linearly separable data.

Limitations:

- Struggles with non-linear patterns.
- Not great with complex relationships.

2. K-Nearest Neighbors (KNN)

KNN looks at the 'K' closest data points to predict the class of a new data point.

- **How it works:** Classifies data based on majority vote of nearby points.

Strengths:

- No training needed.
- Works for non-linear data.
- Simple and easy to implement.

Limitations:

- Slow with large datasets.
- Sensitive to irrelevant features and noise.

3. Decision Tree

Decision trees split data into branches based on features, making them suitable for complex, for non-linear tasks.

- **How it works:** Decision trees split data based on conditions, not straight lines or formulas.

Strengths:

- Handles both numeric and categorical data.
- Handles non-linear data well.

Limitations:

- Can overfit if not controlled.
- Small changes affect structure.

4. Support Vector Machine (SVM)

SVM tries to find the best boundary (or margin) that separates two classes. It can also use kernels to handle complex shapes.

- **How it works:** Finds the best boundary (line/curve) to separate classes.

Strengths:

- Good for high-dimensional data.
- Works well with clear margins between classes.
- Robust to overfitting.

Limitations:

- Hard to tune and slower with big data.
- Doesn't work well when classes overlap too much.
- SVM can be very accurate, but it's slower and uses more resources when working with big or complex data.

Part 2: Application Scenarios

For each of the following dataset scenarios, recommend the most suitable algorithm

(Logistic Regression, KNN, Decision Tree, or SVM). Provide a brief explanation for your

choice.

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

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

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

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

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

Part 2: Application Scenarios

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

Algorithm: SVM

Why: SVM (Support Vector Machine) often performs well in high-dimensional spaces, especially when the number of features is larger than the number of samples. Its ability to find an optimal hyperplane with a maximal margin can be effective in distinguishing between classes even with many features. While Logistic Regression can also handle high dimensions, SVM's focus on margin maximization can lead to better

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

Algorithm: Decision Tree

Why: Decision Tree are often preferred for imbalanced datasets. They can naturally handle skewed class distributions by creating splits that focus on the minority class. While SVM can be adapted with class weights, and Logistic Regression might struggle with predicting the minority class, tree-based methods can learn complex decision boundaries that capture the nuances of the less frequent class.

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

Algorithm: SVM

Why: SVM shines here as well. Its regularization capabilities help prevent overfitting, which is a significant risk when we have many features and few samples. The ability of SVM with different kernels (like RBF) to find complex decision boundaries in a high-dimensional space can be advantageous. Logistic Regression might also work with proper regularization, but KNN is prone to overfitting, and Decision Trees might create overly complex trees that don't generalize well.

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

Algorithm: SVM(with kernel)

Why: SVM with a non-linear kernel (like Radial Basis Function - RBF) is highly suitable for this scenario. The kernel trick allows SVM to implicitly map the data into a higher-dimensional space where a linear hyperplane can separate the non-linear data. KNN can also handle non-linear boundaries but might be sensitive to the choice of 'k'. Logistic Regression, being a linear model, will struggle significantly. Decision Trees can create non-linear boundaries through a series of axis-parallel splits, but SVM with a kernel can often find more elegant and effective separations.

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

Algorithm: SVM

Why: SVM can be relatively robust to noise due to its focus on the margin and the support vectors, which are the data points closest to the decision boundary. Noisy points that are far from the boundary have less impact on the model. Decision Trees are prone to overfitting noisy data by creating branches to accommodate outliers. Logistic Regression might be affected by noisy features pulling the decision boundary. KNN can also be sensitive to noise, as noisy neighbors can influence the classification of a point.