# K-Nearest Neighbors (KNN) and Support Vector Machines (SVM)

## 1. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm used for classification and regression. It operates on the principle that similar instances exist in close proximity.

### Key Features:

• Non-parametric: Makes no assumptions about the data distribution.  
• Instance-based: Stores all available instances and predicts based on a similarity measure.

### Hyperparameters:

• \*\*Number of Neighbors (k):\*\* The number of nearest neighbors to consider for making predictions.  
• \*\*Distance Metric:\*\* The distance measure used to find the nearest neighbors (e.g., Euclidean, Manhattan).  
• \*\*Weights:\*\* Determines whether all neighbors are weighted equally or closer neighbors have greater influence.

### Steps:

1. \*\*Store Training Instances:\*\* Store all the training data points.  
2. \*\*Calculate Distance:\*\* For a new data point, calculate the distance to all training data points.  
3. \*\*Find Neighbors:\*\* Identify the k-nearest neighbors based on the calculated distances.  
4. \*\*Predict:\*\* For classification, use the majority class of the neighbors. For regression, use the average value of the neighbors.

### Advantages:

• Simple and easy to implement.  
• No training phase required.  
• Can handle multi-class classification problems.

### Disadvantages:

• Computationally expensive during prediction.  
• Sensitive to the choice of k and distance metric.  
• Memory-intensive as it stores all training instances.

## 2. Support Vector Machines (SVM)

Support Vector Machines (SVM) are powerful supervised learning models used for classification and regression tasks. They work by finding the hyperplane that best separates the classes in the feature space.

### Key Features:

• Effective in high-dimensional spaces.  
• Robust to overfitting, especially in high-dimensional space.  
• Can use different kernel functions to handle non-linear data.

### Hyperparameters:

• \*\*C (Regularization Parameter):\*\* Controls the trade-off between achieving a low training error and a low testing error.  
• \*\*Kernel:\*\* Determines the type of hyperplane used to separate the data (e.g., linear, polynomial, radial basis function).  
• \*\*Gamma:\*\* Defines how far the influence of a single training example reaches (used in non-linear kernels).  
• \*\*Degree:\*\* Degree of the polynomial kernel function (if used).

### Steps:

1. \*\*Select Hyperplane:\*\* Choose the hyperplane that maximizes the margin between the classes.  
2. \*\*Kernel Trick:\*\* If the data is not linearly separable, use a kernel function to transform the data into a higher-dimensional space where it is separable.  
3. \*\*Optimize:\*\* Solve the optimization problem to find the optimal hyperplane.  
4. \*\*Classify:\*\* For new data points, determine on which side of the hyperplane they fall.

### Advantages:

• Effective in high-dimensional spaces.  
• Works well for both linear and non-linear data.  
• Robust to overfitting if properly tuned.

### Disadvantages:

• Can be memory-intensive and computationally expensive.  
• Performance depends heavily on the choice of hyperparameters and kernel.  
• Less interpretable compared to other models.

## Conclusion

Both K-Nearest Neighbors and Support Vector Machines are valuable tools in a data scientist’s arsenal. KNN is straightforward and easy to understand, making it suitable for simple tasks, while SVM is powerful and versatile, capable of handling complex and high-dimensional data. Proper tuning of hyperparameters is crucial for both algorithms to achieve optimal performance.