MULTIPLE KERNELS & KERNEL COMBINATION

Instructor: Sophal CHAN

Group 5 | Presentation

CONTENT

1

Introduction to multiple kernel learning in SVM

2

Exploring methods for combining multiple kernels in SVM

3

Challenges & Benefits of using Multiple Kernel

4

Exploring advanced techniques for learning kernel weights and kernel selection

ABOUT US

BUTH KHEMRA
ID: e20201690

LIM KIMHOUNG
ID: e20200339

NANG SREYNICH
ID: e20200447

SAO SAMARTH
ID: e20200084

DORN DAWIN
ID: e20200129

YAN MANON
ID: e20200745

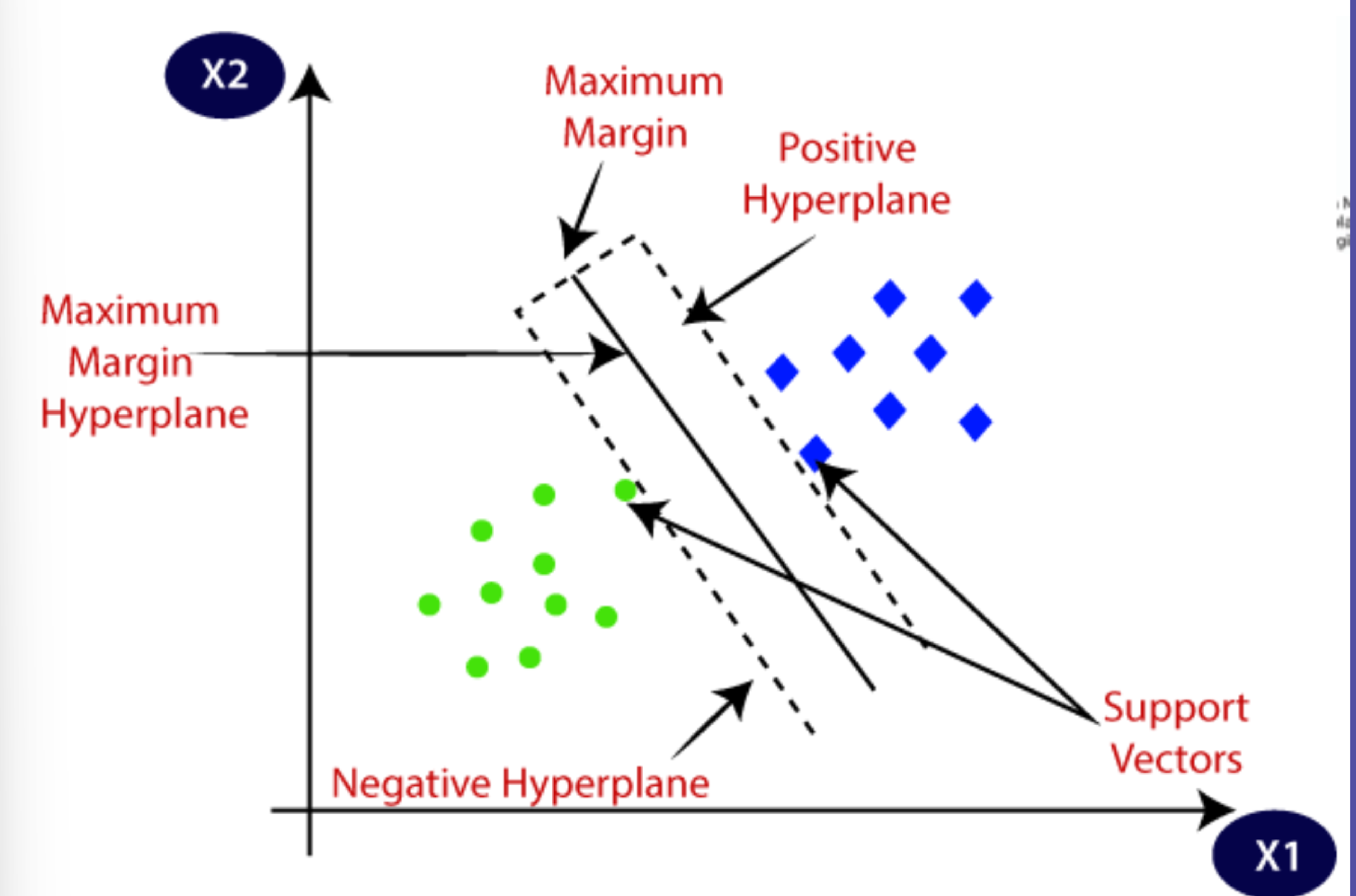
LONG CHANLEAP
ID: e20200386

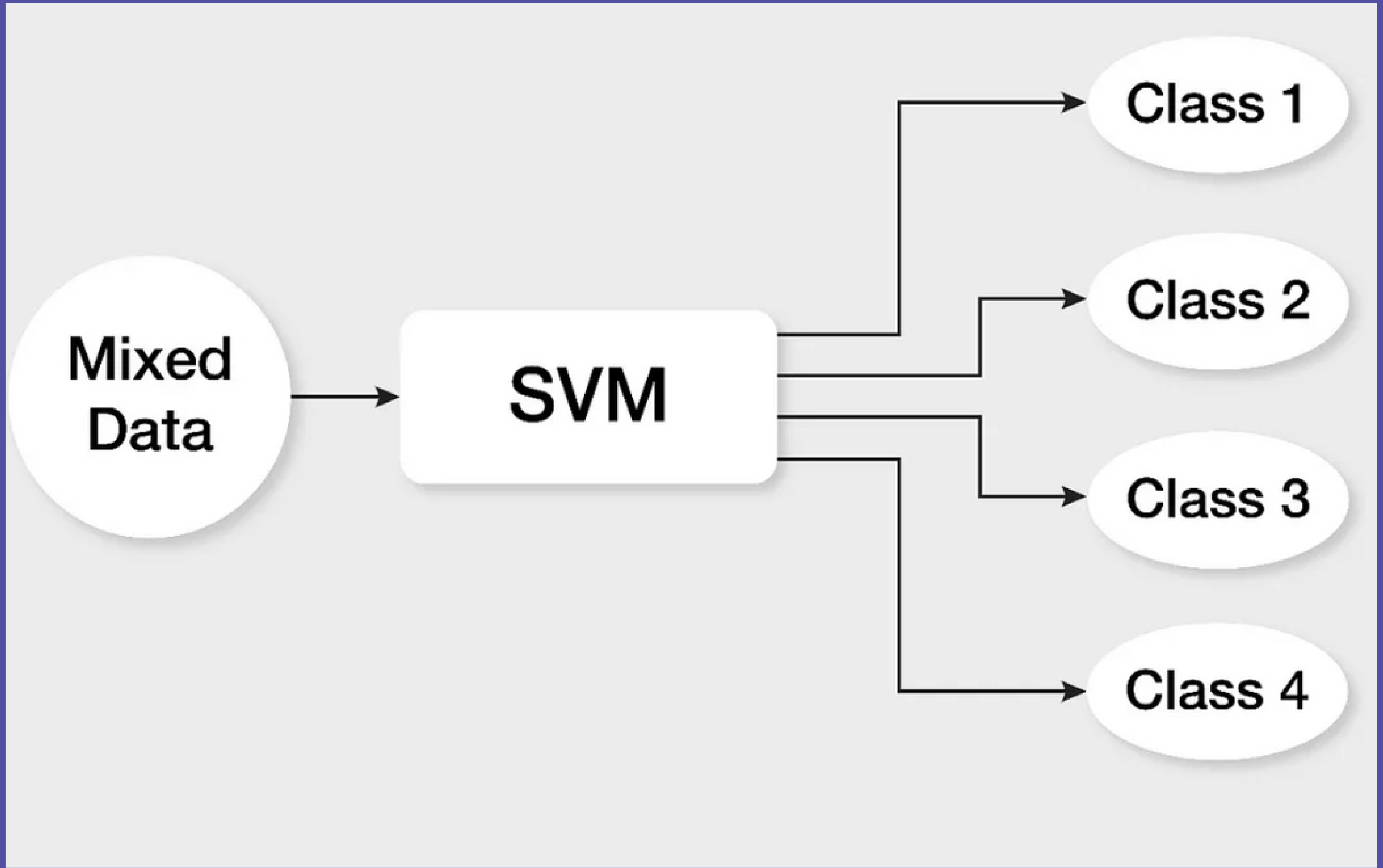
Overview about SVM

It was developed by Vladimir Vapnik and his colleagues in the 1990s.

SVMs are a type of supervised machine learning algorithm used for classification and some cases in regressions tasks.

SVM aims to find an optimal hyperplane that separates data points from different classes in a way that maximizes the margin or distance between hyperplane and the nearest data points. SVM is particularly effective for solving complex problems with high-dimensional data.





WHY WE NEED MULTIPLE KERNELS & COMBINE THEM TGT?

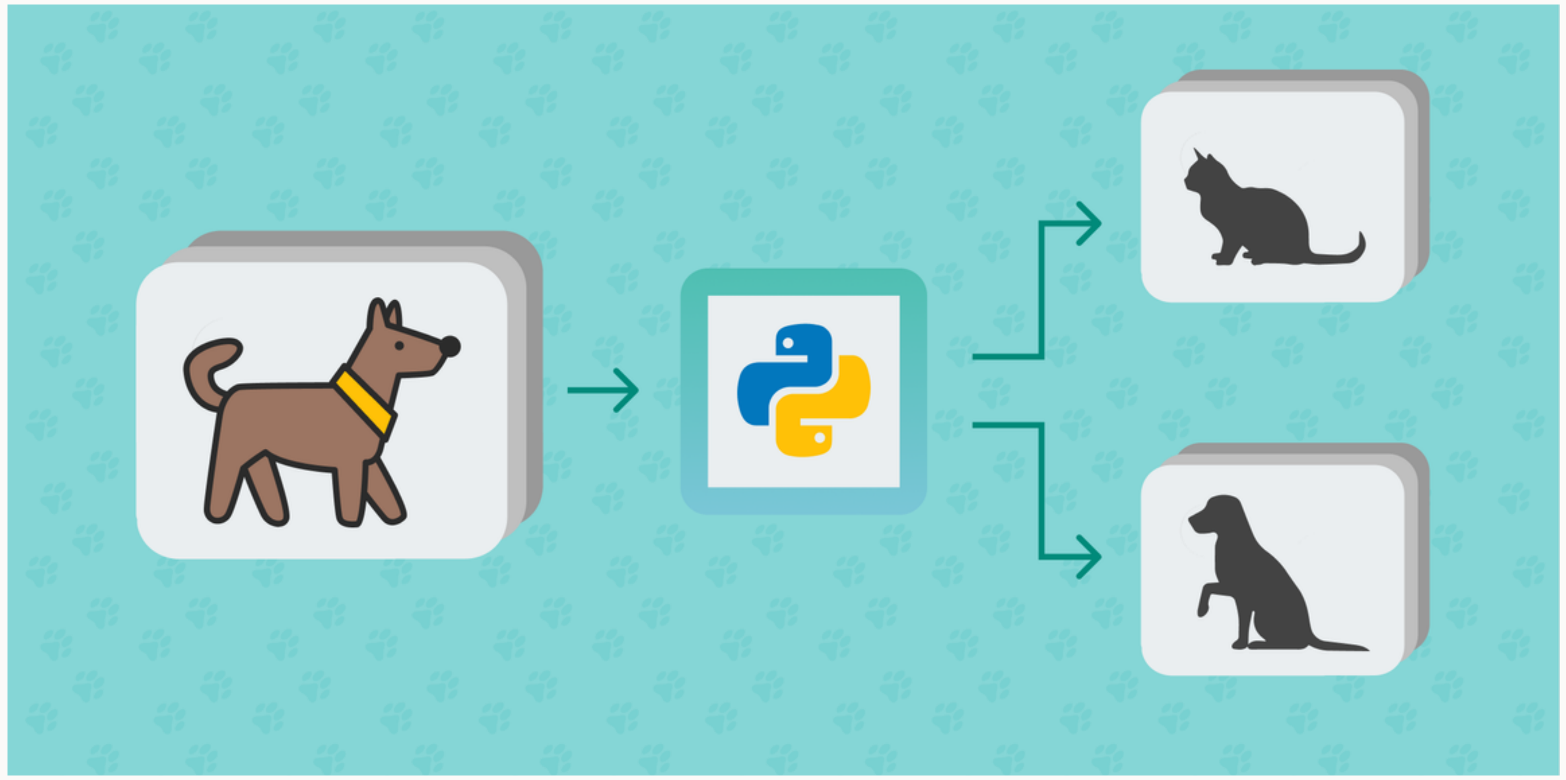
Multiple Kernel SVMs allow to use multiple *kernels together*. This can help to improve the performance of the classifier, cuz the different kernel can complement each other.

4 main types of Kernels (deference from each to another based on its func):

1. RBF(Radial Basis Function) kernel
2. Linear kernel
3. Sigmoid kernel
4. Polynomial kernel

For example: We have a dataset with various features of each image, such as color, shape, texture.....We want to classify images 'cat' or 'dog'. Out goal is to train an SVM model to accurately predict whether a new image contains a cat or dog.

- one feature need one type of kernel to identify its specialized -> the color kernel might look for specific combination of colors that characteristics of cats or dogs, while the shape kernel might seek out unique shape associated with each animal



2.Exploring methods for combining multiple kernels in SVM

1. Weighed multiple kernel method (UMK)
2. Kernel trick
3. Voting ensemble
4. Deep kernel learning

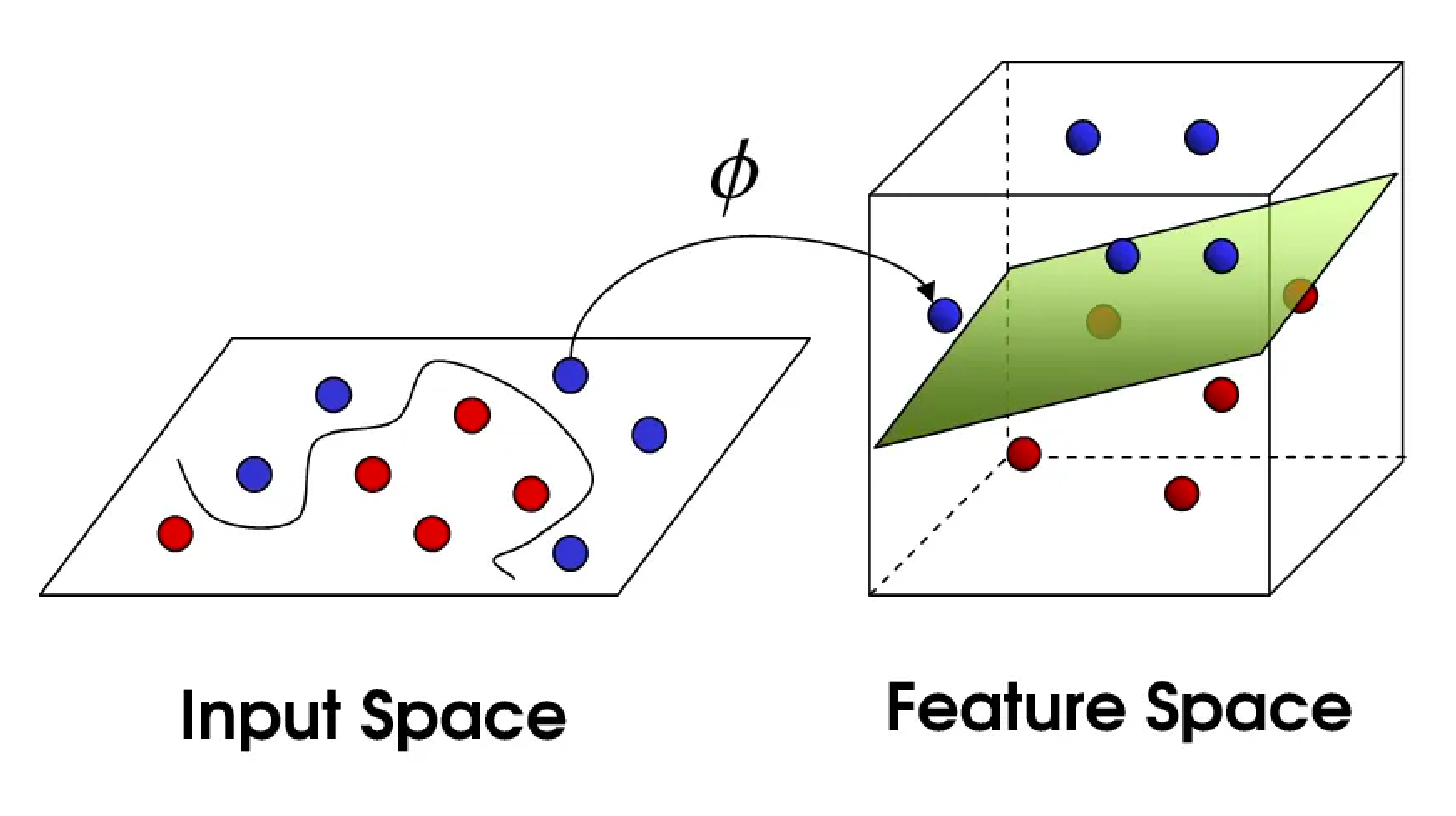
2.1. Weighed multiple kernel method (UMK)

In this method, each kernel is assigned a weight, and the decision boundary is learned by maximizing the margin between the two classes, weighted by the kernel weights. The weights of the kernels can be learned using a supervised learning algorithm, such as gradient descent.

$$K = \sum_{i=1}^{k} \mu_i K_i$$

2.2. Kernel trick

In this method the kernel function is a function that maps the data points from the original space to the higher-dimensional space. In the higher-dimensional space, the data points may be linearly separable, even if they are not linearly separable in the original space



2.3. Voting ensemble

The voting ensemble method for combining multiple kernels in SVM works by training multiple SVMs, each with a different kernel. The decision boundary is then learned by voting on the predictions of the individual SVMs.

For example, let's say we have two SVMs,

- One trained with a linear kernel and one trained with a polynomial kernel.
- We can combine their predictions by simply voting on the class that each SVM predicts.
- If the linear SVM predicts class A and the polynomial SVM predicts class B, then we can vote for class B if class B is most for voting.

2.4. Deep kernel learning

In this method, a deep neural network is used to learn how to combine the outputs of multiple kernels. This can be helpful for tasks where the data is not linearly separable and where the relationships between the data points are complex.

For example, we can train a deep neural network to learn how to combine the outputs of a linear kernel, a polynomial kernel, and a radial basis function kernel. The deep neural network can then be used to make predictions on new data points.

3. Benefits of using multiple kernels

Using multiple kernels in Support Vector Machines (SVMs) can offer several benefits and enhance the performance of the model. Here are some benefits of using multiple kernels:

- Enhanced Model Flexibility
- Improved Generalization
- Handling Heterogeneous Data
- Feature Selection and Importance
- Ensemble Modeling
- Model Adaptability

3. Challenge of using multiple kernels

Although using multiple kernel can be useful and flexible than using only one kernel but it also consist of limitation too such as:

- Multiple Kernel Combination lead to better accuracy but increase the training time
-Can only use with small data

4. Exploring advanced techniques for learning kernel weights and kernel selection

Some advanced techniques for learning kernel weights and kernel selection in Multiple Kernel Learning (MKL) models:

- Cross Validation
- Genetic Algorithms
- Multiple Kernel Learning
- Multiple Kernel Clustering

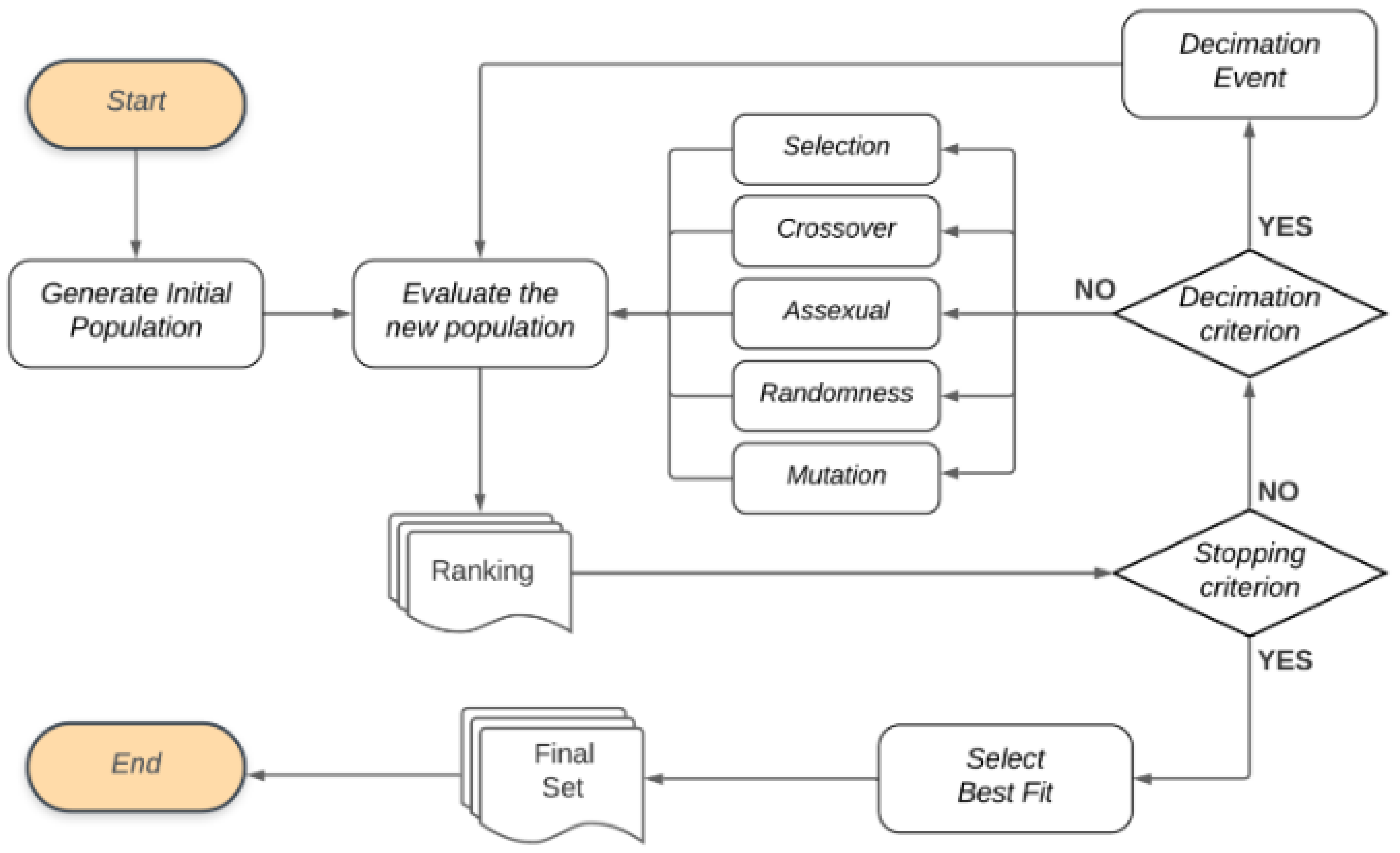
. Cross-validation

Cross-validation (CV) is a standard technique for adjusting hyperparameters of predictive models. We use Cross-validation to determine how many misclassifications and observations to allow inside of the soft magin to get the best classification.

. Genetic Algorithms

A Genetic Algorithm is an evolutionary algorithm that generates solutions of search problems using techniques inspired by the natural evolution of species. Solutions based on genetic algorithms usually generate several populations of individuals based on evolutionary strategies, such as heredity, mutation, and natural selection.

Figure 6. Genetic Algorithm flow.



. Multiple Kernel Learning

MKL : Learn and optimize the combination of multiple kernel in SVM framework. Its goal is to determine the optimal weights or contribution assigned to each kernel, so that their combination lead to improved classification performance.

. Multiple Kernel Clustering

Multiple kernel clustering (MKC) is a machine learning technique that combines multiple kernels to improve the performance of clustering algorithms. Clustering is a machine-learning task that aims to group similar data points together. The performance of clustering algorithms can be limited by the choice of kernel and we propose a multiple kernel clustering (MKC) algorithm that simultaneously finds the maximum margin hyperplane, the best cluster labeling, and the optimal kernel.

thank you