Linear Algebra and Its Applications
Term Project

Hung-Ta, Wang
Institute of Industrial Engineering
National Taiwan University
r13546017@ntu.edu.tw

1 Rationale

1.1 Introduction

Machine learning has become more and more popular nowadays. It can not only fit a number of nonlinear problems but also performs with great accuracy and adaptability in various situations we encounter.

Among these machine learning methods, artificial neural networks have developed rapidly in recent years, however, these advanced methods still have some limitations. I summarized some demerits of artificial neural networks:

1. Artificial neural networks are highly dependent on enormous amounts of data, and the training process costs a lot in both time and computing resources.
2. Since we apply a lot of activation functions and layers in the model, the model's explainability is poor. Therefore, it is difficult to understand the training process and improve the model.
3. When the model demonstrates high prediction accuracy, it is necessary to determine whether this performance stems from the model's effectiveness or is merely a result of overfitting.

In conclusion, though artificial neural networks are powerful, we cannot neglect the importance of other traditional methods like Support Vector Machine (SVM) that are more mathematically based, explainable, and robust.

1.2 Motivation

SVM performs excellently in many fields, including "Speech emotion recognition using support vector machine"[2] or "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review"[5].

Most modern tasks like emotion recognition or image classification are completed by Natural Language Processing (NLP) or Convolutional Neural Networks (CNN), however, SVM should not be a forgotten method.

SVM can demonstrate better results than other modern methods in some situations. Especially in small-sample or high-dimension circumstances, SVM is a more robust method[4]. Even if we cannot certify training results, we can still adjust the model parameters to optimize the results through its high explainability by the solid mathematical foundation of SVM.

To sum up, based on its practicality and structure in linear algebra. I regard SVM as suitable as a topic for this term project.

2 Problem background

2.1 Parameters

Table 1: Parameters in this term project

Parameter	Description
m	A constant number of data points
n	A constant number of features
$W_{m\times 1}$	The weight vector that is perpendicular to the decision boundary
w_i	The i^{th} weight in $W_{m\times 1}$
$X_{m\times n}$	The matrix of m data points with n features
x_{ij}	The data of the j^{th} feature of the i^{th} data point in $X_{m\times n}$
$Y_{m\times 1}$	The vector of types of all data points
y_i	The type of the i^{th} data point
b	The bias that also stands for the interception of the hyperplane
ξ_i	The distance that the i^{th} data point is classified in the wrong type
C	A constant of the tolerance of the error
l_i^{hinge}	The hinge loss of the i^{th} data point

2.2 Support Vector Machine

2.2.1 Definition

I would like to introduce SVM from its definition and geometry. The description is inspired by "Support vector machines–an introduction"[4] and "Data classification using Support Vector Machine (SVM), a simplified approach"[1].

Firstly, Figure1 below is plotted by Python, and it illustrates the fundamental concept of SVM

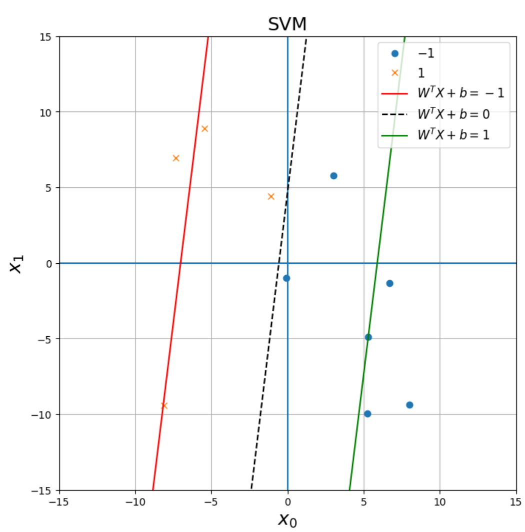

Figure 1: SVM Demostration

SVM can be divided into three parts:

- Decision boundary: $W^Tx + b = 0$
- Positive class support plane: $W^Tx + b = 1$
- Negative class support plane: $W^Tx + b = -1$

The relationship between these three hyperplanes is that the decision boundary is the classification baseline. Thus, it is in the middle of the other two hyperplanes, and the distance from the decision boundary to the support plane is called the margin. We will show that both margins have the same value of distance below.

2.2.2 Primal Form

The margin from the decision boundary to the positive class support plane is

$$\frac{|\ (W^TX+b)_{decision\ boundary} - (W^TX+b)_{positive\ class\ support\ plane}\ |}{||W||}$$
$$= \frac{|\ 0-1\ |}{||W||} = \frac{1}{||W||}$$

The margin from the decision boundary to the negative class support plane is

$$\frac{|\ (W^TX+b)_{decision\ boundary} - (W^TX+b)_{negative\ class\ support\ plane}\ |}{||W||}$$
$$= \frac{|\ 0-(-1)\ |}{||W||} = \frac{1}{||W||}$$

Our objective is to maximize the margin between the two classes

$$maximize \quad \frac{2}{||W||}$$

Since it is difficult to optimize when $||W||$ is the denominator, we transform the problem into an minimization

$$\begin{aligned} &minimize \quad \frac{||W||}{2} = \frac{1}{2}W^TW \\ &subject\ to \\ &\qquad\qquad y_i[W^TX+b] \geq 1 \end{aligned}$$

2.2.3 Dual Form

Since we define the primal form, it can be transformed into the dual form that provides additional properties to the primal form.

The primal form of the problem

$$\begin{aligned} &minimize \quad \frac{1}{2}W^TW \\ &subject\ to \\ &\qquad\qquad y_i[W^Tx_i+b] \geq 1 \end{aligned}$$

Then we can use Lagrange multipliers (α_i) to reform the primal form

$$\begin{aligned} &minimize \quad L = \frac{1}{2}W^TW - \sum_{i=1}^{m} \alpha_i(y_i[W^Tx_i+b]-1) \\ &\alpha_i \geq 0 \end{aligned}$$

Since our objective is to maximize α_i, we have to find the saddle points of W and b by KKT condition.

$$\frac{\partial L}{\partial W} = 0 \rightarrow W = \sum_{i=1}^{m} \alpha_i y_i x_i$$
$$\frac{\partial L}{\partial b} = 0 \rightarrow b = \sum_{i=1}^{m} \alpha_i y_i = 0$$

Recalculate $\frac{1}{2} W^T W$ with new W

$$\begin{aligned} \frac{1}{2} W^T W &= \frac{1}{2} (\sum_{i=1}^{m} \alpha_i y_i x_i)^T (\sum_{j=1}^{m} \alpha_j y_j x_j) \\ &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned}$$

We can reform the dual form L_α

$$\begin{aligned} &minimize \quad L_\alpha = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ &subject\ to \\ &\qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned}$$

Define new notations to reform the dual form

- $\alpha = [\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_m]^T$
- $f = [1, 1, 1, \cdots, 1]^T$
- $H = [y_i y_j x_i^T x_j]$

New dual form

$$\begin{aligned} &minimize \quad L_\alpha = -0.5 \alpha^T H \alpha + f^T \alpha \\ &subject\ to \\ &\qquad Y^T \alpha = 0 \\ &\qquad \alpha_i \geq 0 \end{aligned}$$

2.2.4 Model Generalization

To achieve the generalization of SVM, we add the concept of error tolerance. Thus the model is revised into the form (We use primal form to be example here)

$$\begin{aligned} &minimize \quad \frac{1}{2} W^T W + C \sum_{i=1}^{m} \xi_i \\ &subject\ to \\ &\qquad y_i [W^T X + b] \geq 1 - \xi_i \\ &\qquad \forall i \in [1, m],\ \xi_i \geq 0 \end{aligned}$$

To simplify the problem, we substitute ξ_i into hinge loss. The reason why I chose this concept is inspired by "On the error resistance of hinge-loss minimization"[6].

The hinge loss can be shown below

$$l_i^{hinge} = max(0,\ 1 - y_i W^T x_i)$$

The final model is

$$minimize \quad Z = \frac{1}{2} W^T W + C \sum_{i=1}^{m} l^{hinge}$$

2.2.5 Training Process of the Final Model

After we define the model, we have to find the optimal value. Since W and b are the decision variables, we compute their gradient.

For gradient of W

- When $y_i(W^T x_i + b) < 1$:

$$\frac{\partial}{\partial W}(\frac{1}{2}||W||^2 + C \max(0, 1 - y_i(W^T x_i + b))) = W - C y_i x_i$$

- When $y_i(W^T x_i + b) \geq 1$:

$$\frac{\partial}{\partial W}(\frac{1}{2}||W||^2 + C \max(0, 1 - y_i(W^T x_i + b))) = W - 0$$

- In total, we can get

$$\frac{\partial Z}{\partial W} = W - C \sum_{i=1}^{m} y_i x_i$$

For gradient of b

- When $y_i(W^T x_i + b) < 1$:

$$\frac{\partial}{\partial b}(\frac{1}{2}||W||^2 + C \max(0, 1 - y_i(W^T x_i + b))) = C y_i$$

- When $y_i(W^T x_i + b) \geq 1$:

$$\frac{\partial}{\partial b}(\frac{1}{2}||W||^2 + C \max(0, 1 - y_i(W^T x_i + b))) = 0$$

- In total, we can get

$$\frac{\partial Z}{\partial b} = -C \sum_{i=1}^{m} y_i$$

Just like the gradient descent algorithm, we apply the learning rate to avoid local optima and prevent it from converging too slowly.

$W = W -$ learning rate $\times \frac{\partial Z}{\partial W}$

$b = b -$ learning rate $\times \frac{\partial Z}{\partial b}$

2.2.6 Kernel

For the merits of the dual form, we can substitute the original $x_i^T x_j$ into the kernel to make SVM adapt to the nonlinear problem. My recognition of the kernel is referred to by "Tutorial on support vector machine (SVM)"[3].

Review the new dual form

$$
\begin{aligned}
&minimize \quad L_\alpha = -0.5\alpha^T H\alpha + f^T\alpha \\
&subject\ to \\
&\qquad\qquad Y^T\alpha = 0 \\
&\qquad\qquad \alpha_i \geq 0
\end{aligned}
$$

H is a positive definite matrix and $\alpha^T H\alpha$ can expand to $\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j$, it has the following properties

- $\alpha_i\alpha_j$: The influence of interactions between the data point i and the data point j.
- $y_i y_j$: If the data point i and the data point j are the same, $y_i y_j$ would be 1. Otherwise, it would be -1.
- $x_i^T x_j$: The inner product of the data point i and the data point j. It is equilibrium to $||x_i||\,||x_j||\,cos\theta$. We can also divide the situation into three situations
  1. $x_i^T x_j > 0$: It means $\theta < 90°$, the data point i and the data point j have a positive correlation.
  2. $x_i^T x_j = 0$: It means $\theta = 90°$, the data point i and the data point j is orthogonal. They are irrelevant.
  3. $x_i^T x_j < 0$: It means $\theta > 90°$, the data point i and the data point j have a negative correlation.

Since $x_i^T x_j$ stands for the feature space, we can substitute it into other transformations of the space named kernel. Kernels turn the decision boundary from a straight line into a curve. Here are the kernels in the model and some examples.

$$
\begin{aligned}
&\alpha^T H\alpha = \sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j = \sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \\
&where \quad K(x_i, x_j) = \Phi(x_i)\cdot\Phi(x_j)
\end{aligned}
$$

Types of kernels

- Linear: $K(x_i, x_j) = x_i^T x_j$, original form that the decision boundary is linear.
- Polynomial: $K(x_i, x_j) = (x_i^T x_j + c)^d$, it can capture the nonlinear rule of the data and make the decision boundary a curve by the polynomial.
- Gaussian Radial Basis Function: $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$, it can project the feature space onto an unlimited dimension hyperplane.

3 Solutions with linear algebra theories and techniques

3.1 Vector Space and Eigenvalue

3.1.1 Vector Space

In the beginning, we have to define the feature space $X_{m \times n}$ of data points and the weight vector $W_{m \times 1}$.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix} \qquad W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

Each row of X is properties of a data point and each column of X is the data regarding to the feature.

- $\Re(X^T) \in \mathbb{R}^m$: the linear combination of data points
- $\Re(X) \in \mathbb{R}^n$: the linear combination of features.

Since we always have more data points than features, it is supposed $m > n$. The maximum value of Rank(X) is n. Although it means there are linear dependencies among data points, it weakens the influence of white noises and make the norm of the j^{th} feature more close to the real value. It can help us to define the real $||X_{feature}||$.

3.2 Orthogonality and Projection

3.2.1 Orthogonality

We can interpret $W^T X + b$ as $W \cdot X + b$. It means the inner product of W and X plus the intercept b. Since $||X_{feature}||$, $||W||$ and $||b||$ in the trained model is a certain value, the magnificence of $W \cdot X + b = ||W||\ ||X_{feature}||\ cos\theta + b$ will only be affected by θ.

The positivity of $W^T X + b$ stands for the correlation between W and X. The value $W^T X + b$ will become larger if W and X are more correlated. If $W^T X + b$ is close to 0, it means this hyperplane can hardly classify the type of this data point. In the worst case, $W^T X + b = 0$ means this hyperplane is not effective in classification that is the decision boundary.

3.2.2 Projection

If we want to get the projection matrix directly, the function is too complicated to calculate. Let Φ be X after transformation.

$$P_\Phi = \Phi(\Phi^T \Phi)^{-1} \Phi^T$$

Thus, we transform X and then calculate the projection matrix. The kernel is the tool that projects the feature space onto the hyperplane with a higher dimension.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) = \begin{bmatrix} K(x_i, x_i) & K(x_i, x_j) \\ K(x_j, x_i) & K(x_j, x_j) \end{bmatrix}$$

$$\rightarrow \quad P_\Phi = K(x_i, x_j)[K(x_i, x_j)^T K(x_i, x_j)]^{-1} K(x_i, x_j)^T$$

Where K is a semi-positive definite matrix, its eigenvalues must be nonnegative.

3.3 Quadratic Function in Matrix Form

In the dual form of SVM

$$minimize \quad L_\alpha = -0.5\alpha^T H\alpha + f^T\alpha$$

Where $H = y_i y_j K(x_i, x_j)$ is a semi-positive definite matrix because of $K(x_i, x_j)$. We can see $y_i y_j$ as a weight of the kernel. Since eigenvalues of $K(x_i, x_j)$ must be nonnegative, we can suppose that the determinant and the trace of H is also nonnegative. It is almost impossible to have eigenvalues that are all zeros.

Moreover, we can know that H must be convex for the nonnegative determinant. Thus, it has the minimum.

4 Examples / Applications

We choose two examples to elaborate on the properties of SVM.

4.1 Example 1: Titanic from Kaggle

4.1.1 Introduction

The Titanic dataset is a popular dataset on the Kaggle. I chose it because it is complete and simple. I expect to observe the performance of SVM in a linearly separable dataset. We standardized numerical features and one-hot encoded categorical features to preprocess data. 2

	Survived	Age	SibSp	Parch	Fare	Sex_female	Sex_male	Pclass_1	Pclass_2	Pclass_3	Embarked_C	Embarked_Q	Embarked_S
0	0.0	34.5	0.0	0.0	7.8292	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
1	1.0	47.0	1.0	0.0	7.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0
2	0.0	62.0	0.0	0.0	9.6875	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0
3	0.0	27.0	0.0	0.0	8.6625	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
4	1.0	22.0	1.0	1.0	12.2875	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
413	0.0	27.0	0.0	0.0	8.05	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
414	1.0	39.0	0.0	0.0	108.900002	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
415	0.0	38.5	0.0	0.0	7.25	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
416	0.0	27.0	0.0	0.0	8.05	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
417	0.0	27.0	1.0	1.0	22.358299	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0

418 rows × 13 columns

Figure 2: Preprocessed Data

4.1.2 Model Evaluation

Figure 3 is implemented by sklearn package with Gaussian RBF kernel, and Figure 4 is self-made with linear kernel. We also need a version with the package to certify our outcome is correct. In this simple and linear separatable dataset, we can see that the linear kernel may be slightly better than the RBF kernel by their f1 scores.

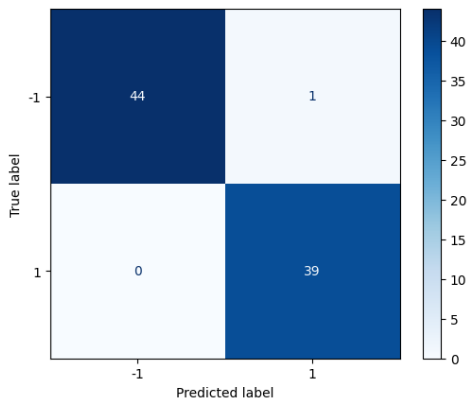

Figure 3: Confusion Matrix (RBF Kernel)

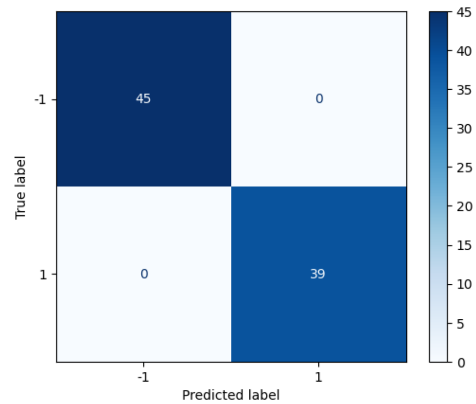

Figure 4: Confusion Matrix (Linear Kernel)

4.2 Example 2: Generated Moon Data

4.2.1 Introduction

This dataset is used to compare the Titanic with its nonlinear properties. The distribution of one class is a curve like a moon in Figure 5. SVM with the RBF kernel is also implemented by the package, while SVM with the linear kernel is self-made.

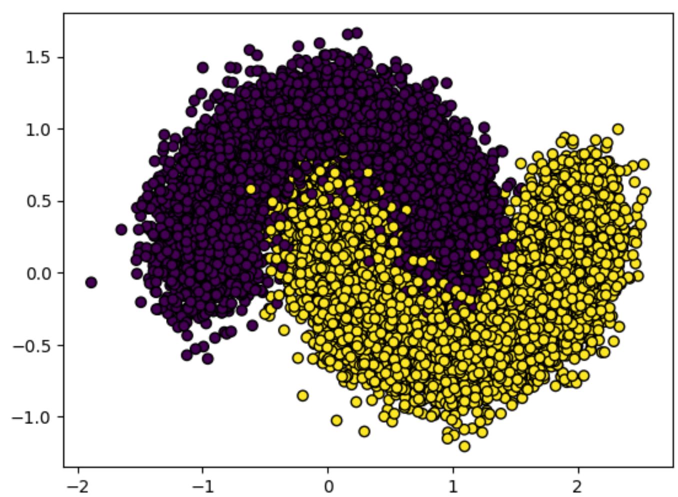

Figure 5: Generated Moon Data

4.2.2 Model Evaluation

In a nonlinear dataset, SVM with the RFB kernel has a great performance like Figure 6, while the accuracy of SVM with the linear kernel is shown by Figure 7, which is worse than the former one.

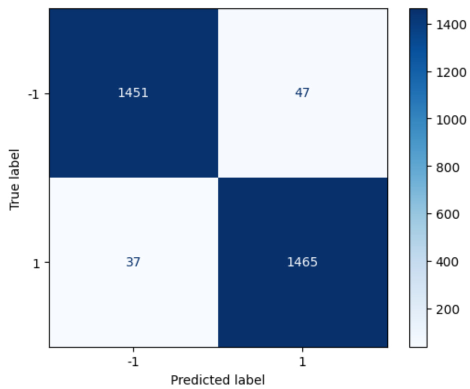

Figure 6: Confusion Matrix (RBF Kernel)

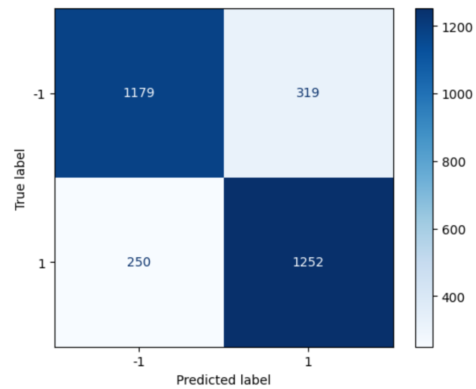

Figure 7: Confusion Matrix (Linear Kernel)

To achieve data visualization, we plot the decision boundary. From Figure 8, it can be observed that the decision boundary is a curve while it is linear in Figure 9. This difference make SVM with the RBF kernel adapt the moon data better than SVM with the linear kernel and predict more correctly.

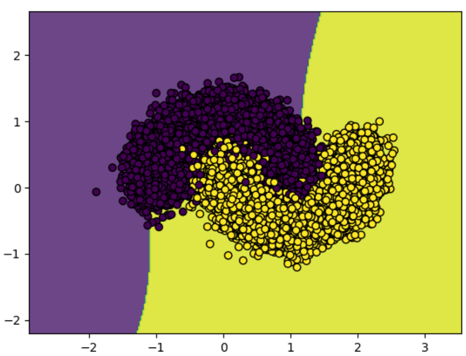

Figure 8: Confusion Matrix (RBF Kernel)

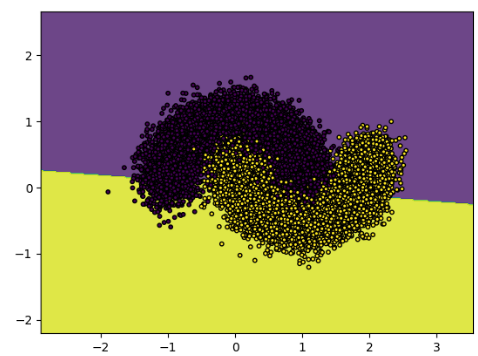

Figure 9: Confusion Matrix (Linear Kernel)

5 Discussions

5.1 Definition of SVM

From the definition of the SVM, I regard it as a traditional but powerful machine learning model. We can know how data is transformed by its kernel, and which kernel we should choose. Moreover, the feature space spanned by $x_i^T x_j$ is also a specific tool to see the correlation between data points.

We can use the feature space and the kernel to project our data on a different hyperplane. It makes SVM have decent performance on the high-dimension dataset.

5.2 Summary of Examples

The choice of the kernel is also crucial for optimizing predictions. It is shown that some kernels like Gaussian Radial Basis Function can turn the decision boundary from a line into a curve. However, it does not mean the RBF kernel is always better than the original linear kernel.

"All models are wrong, but some are useful." is the best quote to conclude the result. What is the space of features? Where may be a proper place for the decision boundary? How do we project it? Why? We have to always keep these questions in mind.

References

[1] S. Amarappa and S. V. Sathyanarayana. Data classification using support vector machine (svm), a simplified approach. *International Journal of Electronics and Computer Science Engineering*, 3:435–445, 2014.

[2] M. Jain, S. Narayan, P. Balaji, A. Bhowmick, and R. K. Muthu. Speech emotion recognition using support vector machine. *arXiv preprint arXiv:2002.07590*, 2020.

[3] Vikram Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.

[4] Vojislav Kecman. Support vector machines–an introduction. In *Support Vector Machines: Theory and Applications*, pages 1–47. Springer, Berlin, Heidelberg, 2005.

[5] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni. Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:6308–6325, 2020.

[6] Kunal Talwar. On the error resistance of hinge-loss minimization. *Advances in Neural Information Processing Systems*, 33:4223–4234, 2020.