

MACHINE LEARNING IN BIOINFORMATICS

FROM LOGISTIC REGRESSION TO SVMs

Philipp Benner
philipp.benner@bam.de

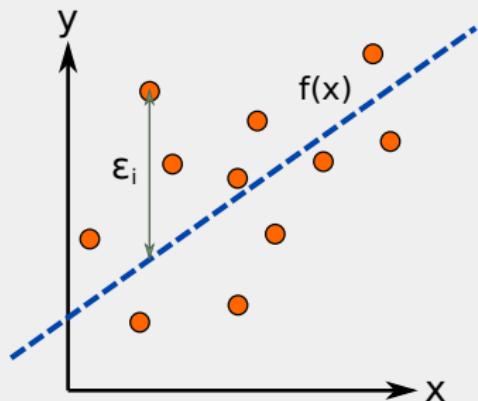
VP.1 - eScience
Federal Institute of Materials Research and Testing (BAM)

February 8, 2026

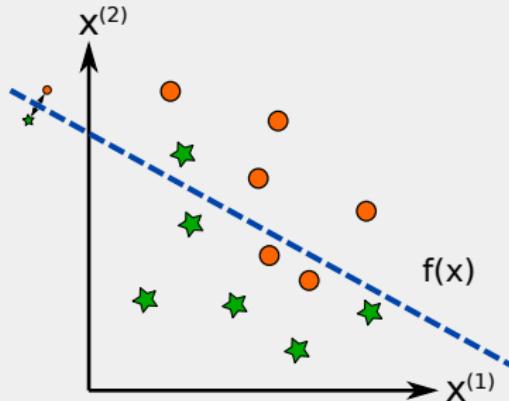
LOGISTIC REGRESSION (CLASSIFICATION)

LINEAR REGRESSION AND CLASSIFICATION

Linear regression



Logistic regression



$$y = X\theta + \epsilon, \quad y \in \mathbb{R}$$

$$y \stackrel{?}{=} \sigma(X\theta) + \epsilon, \quad y \in \{0, 1\}$$

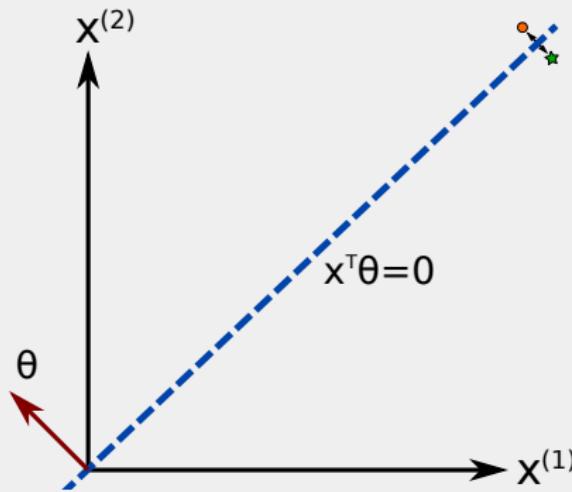
How is the hyperplane defined? What is σ ?

DEFINING HYPERPLANES

- We use the properties of the dot product to define the separating hyperplane:

$$x^\top \theta = \|x\| \|\theta\| \cos \angle$$

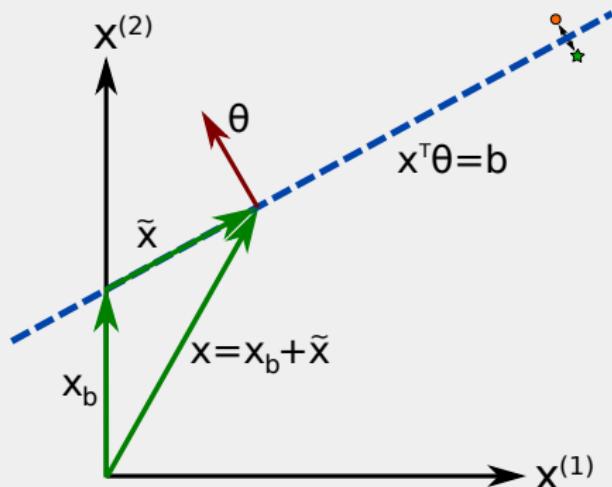
- For vectors x perpendicular to θ we have $\cos \angle = 0$



DEFINING HYPERPLANES

- For hyperplanes with bias b we use $x^\top \theta = b$

$$\begin{aligned}x^\top \theta &= (x_b + \tilde{x})^\top \theta \\&= \underbrace{x_b^\top \theta}_\text{=} + \underbrace{\tilde{x}^\top \theta}_\text{=0}\end{aligned}$$



DEFINING HYPERPLANES

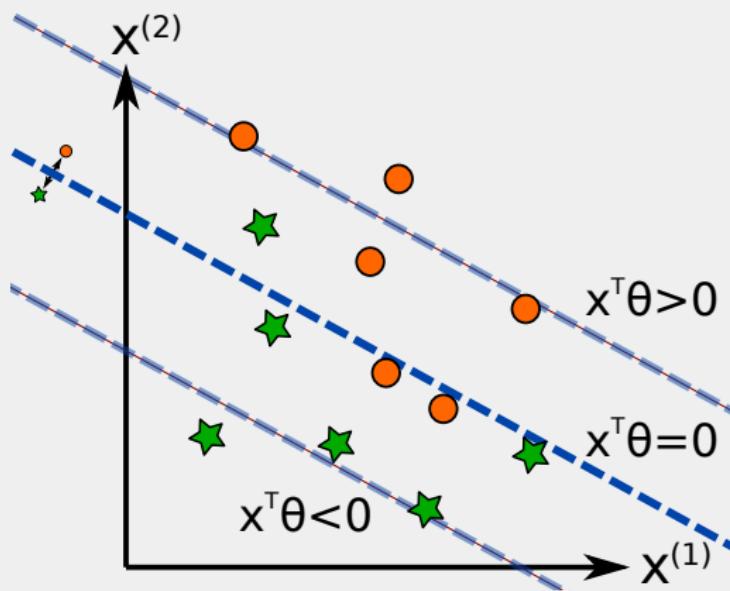
- Remember our convention:

$$x = \begin{bmatrix} 1 \\ x^{(2)} \\ \vdots \\ x^{(p)} \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}$$

- Hence, instead of $x^\top \theta = b$ we can write $x^\top \theta = o$, because $\theta_1 = -b$

SEPARATING HYPERPLANE

- $x^\top \theta > 0$: predicting positive class
- $x^\top \theta < 0$: predicting negative class



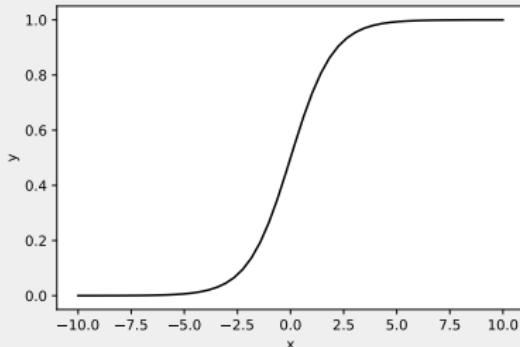
LOGISTIC REGRESSION

- We convert $x^\top \theta$ to probabilities

$$\text{pr}(Y = 1 | x) = \sigma(x^\top \theta)$$

- The function σ denotes the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



LOGISTIC REGRESSION

- Given a training set (X, y) how do we estimate θ ?
- Option 1: Minimizing squared error (similar to OLS)

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^n \left[y_i - \sigma(x_i^\top \theta) \right]$$

Problem: Not convex!

- Remember how we justified OLS for linear models?
- Option 2: Maximum likelihood

$$\hat{\theta} = \arg \max_{\theta} \text{pr}(y | X, \theta)$$

LOGISTIC REGRESSION

- What is the probability of (X, y) ?
- Remember a Bernoulli experiment (coin flip) with outcomes H (head) and T (tail)
- H is observed with probability p
- T is observed with probability $1 - p$
- The sequence HHTHT has probability

$$\text{pr}(\text{HHTHT}) = pp(1-p)p(1-p)$$

- Remember the following rule of thumb:

\times = "and"

$+$ = "or"

LOGISTIC REGRESSION

- For logistic regression, assume $y = (1, 1, 0, 1)$, hence

$$\text{pr}(1, 1, 0, 1 | X, \theta) = \sigma(x_1^\top \theta) \sigma(x_2^\top \theta) (1 - \sigma(x_3^\top \theta)) \sigma(x_4^\top \theta)$$

- Write it nicely in general form:

$$\text{pr}(y | X, \theta) = \prod_{i=1}^n \sigma(x_i^\top \theta)^{y_i} (1 - \sigma(x_i^\top \theta))^{1-y_i}$$

- Maximum likelihood

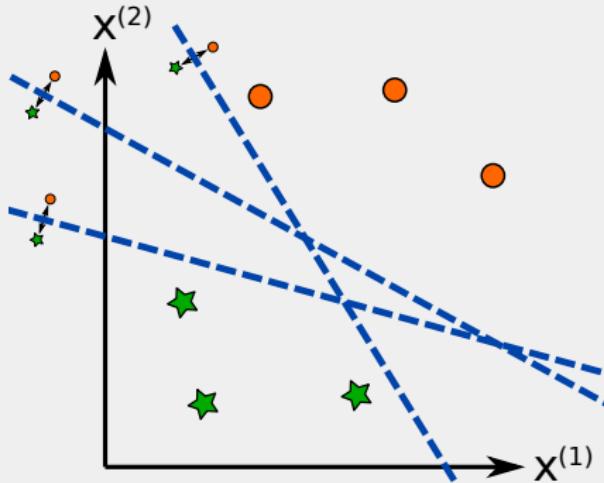
$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} \prod_{i=1}^n \sigma(x_i^\top \theta)^{y_i} (1 - \sigma(x_i^\top \theta))^{1-y_i} \\ &= \arg \max_{\theta} \sum_{i=1}^n y_i \log \sigma(x_i^\top \theta) + (1 - y_i) \log(1 - \sigma(x_i^\top \theta))\end{aligned}$$

- Convex optimization problem, but must be solved numerically

SUPPORT VECTOR MACHINES (SVMs)

SUPPORT VECTOR MACHINES

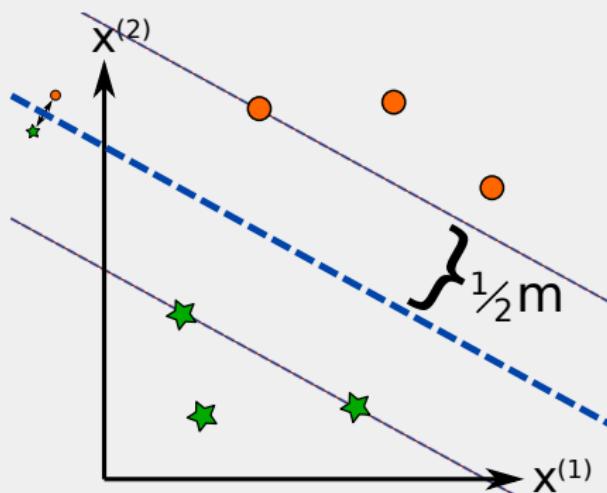
- Support vector machines (SVMs) are similar to logistic regression, however, their learning algorithm is geometrically motivated:



- What is the *best* separating hyperplane?

SUPPORT VECTOR MACHINES

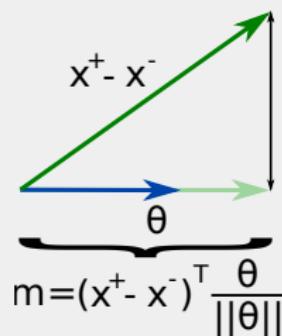
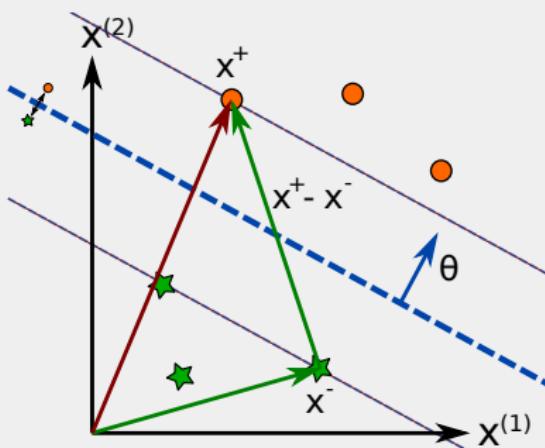
- SVMs take the hyperplane with maximum margin m :



- Data points touching the margin are called *support vectors*
- What is m and how can we maximize it?

SUPPORT VECTOR MACHINES

- Computing the margin m given a fixed hyperplane:



- Hence, the margin is determined by the scalar projection of $x^+ - x^-$ onto $\theta / \|\theta\|$:

$$m = (x^+ - x^-)^\top \frac{\theta}{\|\theta\|_2}$$

SUPPORT VECTOR MACHINES

- So far we did not enforce any constraints on θ
- There are infinitely many θ for the same separating hyperplane
- We apply the constraint

$$(x^+)^T \theta = 1, \quad (x^-)^T \theta = -1$$

for positive x^+ and negative x^- support vectors

- This definition leads to a simplified margin:

$$\begin{aligned} m &= (x^+ - x^-)^T \frac{\theta}{\|\theta\|_2} \\ &= \frac{2}{\|\theta\|_2} \end{aligned}$$

SUPPORT VECTOR MACHINES

SVM optimization problem

Let $(x_i, y_i)_i$ denote a training set such that $y_i \in \{-1, 1\}$. The parameters of the SVM are estimated as follows:

$$\begin{aligned}\hat{\theta} = \arg \min_{\theta} & \|\theta\|_2 \\ \text{s.t. } & x_i^\top \theta y_i \geq 1\end{aligned}$$

- Note that minimizing $\|\theta\|_2$ is equivalent to maximizing the margin $2/\|\theta\|_2$
- The solution can be computed using the Lagrangian

$$L(\theta, \lambda) = \frac{1}{2} \|\theta\|_2^2 - \sum_{i=1}^n \lambda_i (x_i^\top \theta y_i - 1)$$

- The solution is a *saddle point* of the Lagrangian $L(\theta, \lambda)$

SUPPORT VECTOR MACHINES

SVM dual problem

The solution of the SVM is obtained by maximizing the dual problem

$$\begin{aligned} Q(\lambda) &= \min_{\theta} L(\theta, \lambda) \\ &= \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j x_i^\top x_j \end{aligned}$$

subject to $\lambda_i \geq 0$.

- We have a Lagrange multiplier λ_i for each data point
- λ_i is zero except for support vectors
- The dual problem is solved using the *Sequential minimal optimization (SMO)* algorithm [Cristianini et al., 2000]
- **The dual representation depends on $x_i^\top x_j$**

SUPPORT VECTOR MACHINES

- What if the data is not linearly separable?

Option 1: Slack variables

SVM optimization problem for non-linearly separable data

Let $(x_i, y_i)_i$ denote a training set such that $y_i \in \{-1, 1\}$. The parameters of the SVM are estimated as follows:

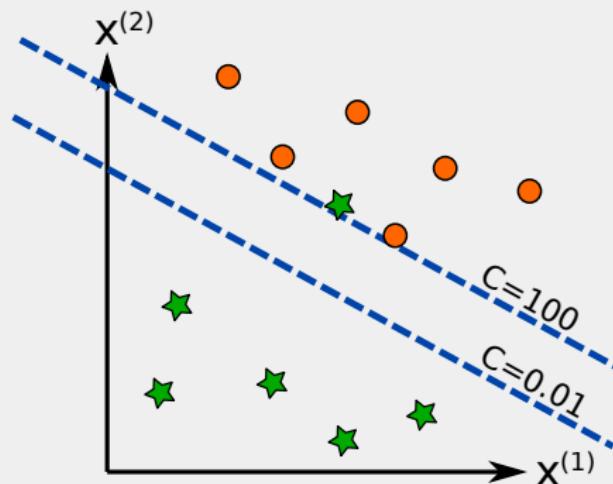
$$\hat{\theta} = \arg \min_{\theta} \|\theta\|_2 + C \sum_{i=1}^n \xi_i$$
$$\text{s.t. } x_i^\top \theta y_i \geq 1 - \xi_i$$

where C is the slack penalty.

- $C = \infty$: Data must be linearly separated. $C = 0$: Ignore data.
- The dual problem is almost identical to the case of linearly separable data

SUPPORT VECTOR MACHINES - SLACK VARIABLES

- The effect of the slack penalty:

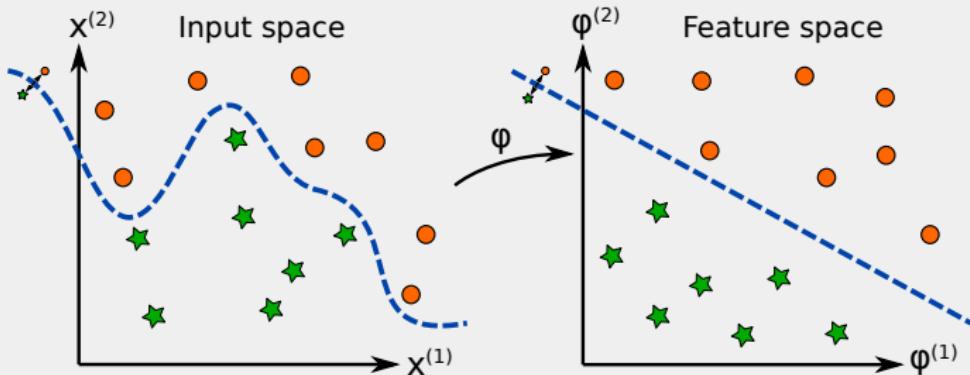


- $C = 0.001$: Some misclassified points are almost ignored
- $C = 100.0$: Get as close as possible to misclassified points

SUPPORT VECTOR MACHINES - FEATURE SPACE

- What if the data is not linearly separable?

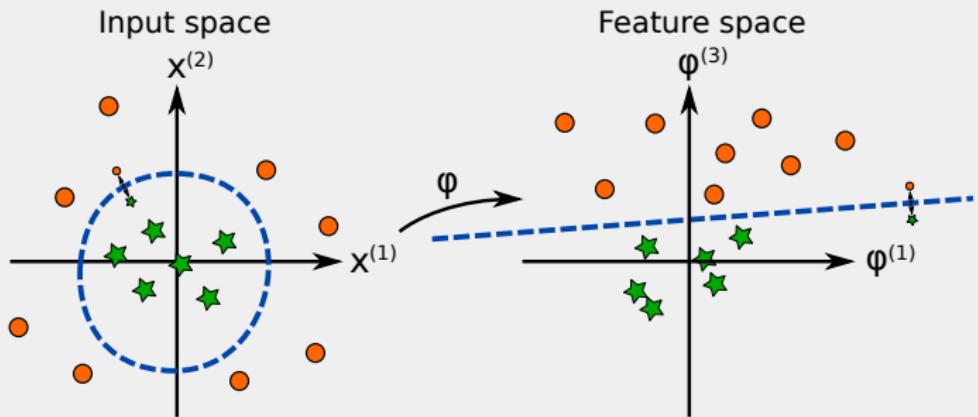
Option 2: Feature space



- $x_i^T x_j$ measures similarity in input space
- $\phi(x_i)^\top \phi(x_j)$ measures similarity in feature space
- Dimension of feature space is typically much larger
- Data often becomes linearly separable

SUPPORT VECTOR MACHINES - FEATURE SPACE

- Example: $\phi(x^{(1)}, x^{(2)}) = (x^{(1)}, x^{(2)}, x^{(1)}x^{(1)} + x^{(2)}x^{(2)})$



- Projection of the hyperplane back to input space will result in a non-linear decision boundary

SUPPORT VECTOR MACHINES - KERNELS

Definition: Kernel function

A function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is called a *kernel* if there exists a feature map $\phi : \mathcal{X} \rightarrow \mathcal{F}$ such that

$$\kappa(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

$K = (\kappa(x_i, x_j))_{x_i \in \mathcal{X}, x_j \in \mathcal{X}}$ is called the kernel matrix.

- \mathcal{X} can be an arbitrary space, for instance DNA sequences
- $\kappa(x_i, x_j)$ is interpreted as a similarity measure in feature space
- Evaluating $\kappa(x_i, x_j)$ does not always require to explicitly compute $\phi(x)$
- Not having to map data into feature space is called the **kernel trick**

SUPPORT VECTOR MACHINES - RBF KERNEL

- The Gaussian or radial basis function (RBF) kernel:

$$k(x_i, x_j) = \exp \left\{ -\frac{\|x_i - x_j\|_2^2}{2\sigma^2} \right\}$$

- Instead of the dot product $x_i^\top x_j$ we use the difference $x_i - x_j$ as the similarity measure
- What is the corresponding feature map ϕ ?
- Taylor expansion of the kernel leads to

$$\phi(x) = \exp \left(-\frac{x^2}{2\sigma^2} \right) \left[1, \sqrt{\frac{1}{1!\sigma^2}}x, \sqrt{\frac{1}{2!\sigma^4}}x^2, \sqrt{\frac{1}{3!\sigma^6}}x^3, \dots \right]$$

- Feature space of the RBF kernel has infinite dimensions

SUPPORT VECTOR MACHINES - κ -SPECTRUM KERNEL

- Suppose our input data is DNA sequences or any other type of strings
- How would we measure the similarity of two strings x_i and x_j ?
- Feature map ϕ of the κ -spectrum kernel counts the number of occurrences of substrings of length κ
- Example:

$x_1 = \text{"statistics"}$

$x_2 = \text{"computation"}$

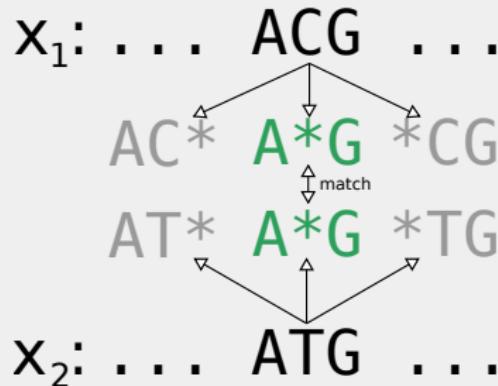
- For $\kappa = 3$ we get:

$$\phi(x_1) = \begin{bmatrix} \text{aaa} & \text{aab} & \dots & \text{sta} & \dots & \text{tat} & \dots \\ 0 & 0 & \dots & 1 & \dots & 1 & \dots \end{bmatrix}$$

- $k(x_1, x_2) = \phi(x_1)^\top \phi(x_2) = 1 \cdot 1 + 1 \cdot 1 = 2$
- We don't have to compute ϕ explicitly, only count the common substrings

SUPPORT VECTOR MACHINES - GAPPED κ -MERS

- l : word or substring length
- k : number of non-gaps



- Number of gapped κ -mers:

$$\binom{l}{\kappa} 4^\kappa$$

- Requires very efficient implementation (e.g. gkmSVM [Ghandi et al., 2014])

SUPPORT VECTOR MACHINES - SUMMARY

- Support vector machines and logistic regression have different learning objectives
- SVMs maximize the margin between positive and negative samples
- Two approaches to deal with non-linearly separable data:
 - ▶ Slack variables to weaken the separability objectives
 - ▶ Implicit mapping into high-dimensional feature space with Kernels
- SVMs and logistic regression have different number of parameters:
 - ▶ SVMs: One parameter for each training point
 - ▶ Logistic regression: One parameter for each feature
- Evaluation of the kernel matrix takes $\mathcal{O}(n^2)$ steps

SUPPORT VECTOR MACHINES - READING

- Reading: Chapter 12 [Hastie et al., 2009], Section 6.1 [Cristianini et al., 2000]
- Advanced reading: *Representer Theorem*

REFERENCES

-  CRISTIANINI, N., SHawe-Taylor, J., ET AL. (2000).
AN INTRODUCTION TO SUPPORT VECTOR MACHINES AND OTHER KERNEL-BASED LEARNING METHODS.
Cambridge university press.
-  GHANDI, M., LEE, D., MOHAMMAD-NOORI, M., AND BEER, M. A. (2014).
ENHANCED REGULATORY SEQUENCE PREDICTION USING GAPPED K-MER FEATURES.
PLoS computational biology, 10(7):e1003711.
-  HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. (2009).
THE ELEMENTS OF STATISTICAL LEARNING: DATA MINING, INFERENCE, AND PREDICTION.
Springer Science & Business Media.