UNIVERSIDAD NACIONAL
DE SAN MARTÍN

Machine Learning - Week 2

Maestría en Ciencia con mención de Tecnología de la información

Yonatan Carlos CARRANZA ALARCÓN,
Ph. D. in Machine Learning,
ycarranza.alarcon@gmail.com

https://salmuz.github.io/

October 14, 2021

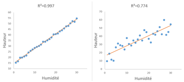
Presentación

2021 Nov. **Machine Learning Ops.**, *Warner Bros. Entertainment*, France.

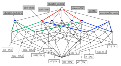
2020 – 2021 **Research And Teaching Assistant**, *Université de Technologie de Compiègne (UTC)*, France.

2018 – 2020 **PhD in Computer Science**, *University of Technology of Compiègne*, Compiègne - France.
Distributionally robust, skeptical inferences in supervised classification using imprecise probabilities

2015 – 2017 **M.S in Applied Mathematics**, *University of Montpellier*, France.
Majoring in Biostatistics,
Health-Economic modeling using Markov model and application in R



2013 – 2015 **M.S in Computer Science**, *University of Montpellier*, France.
Majoring in Data, Knowledge and Natural Language,
Modelling of the users behaviour for Crowdsourcing platform to large scale



2011 – 2012 **Diploma Course in Project Management**, *Institute San Ignacio de Loyola*, Lima, Pérou.
Project Management based on the focus of the Project Management Institute PMI

2004 – 2009 **Bachelor in Computer Science**, *National University of San Marcos*, Lima, Peru.
Computers and Systems

2007 – 2012 **Senior Developer Analyst**, Lima - Peru.

Overview

Supervised learning

Supervised classification

- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression

- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods

- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

Interpretability and explainability of ML methods

Bibliography

Overview

Supervised learning

Supervised classification
- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression
- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods
- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

Interpretability and explainability of ML methods

Bibliography

Outline of supervised learning problem.

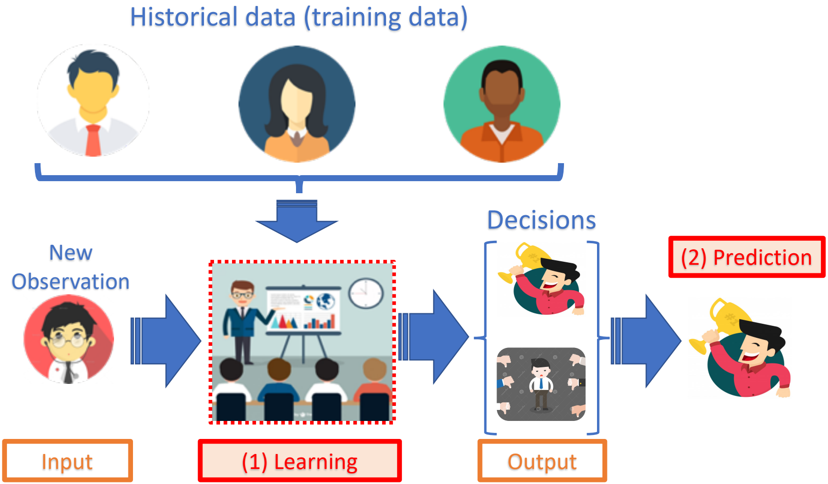


Figure: Learn a recruitment and selection process model.

Supervised learning

Any learning process is based on knowledge acquisition, be it implicit, explicit, or both. *That is how it happens in humans and not too differently in computers.*

Computers focuses on a *specific and particular task*, in which it learns to generalize repetitive and similar patterns of a well-framed and well-specific experiment, e.g. classification of images.

Objective

Learn a *model* that minimizes the risk of making a wrong decision.

Mathematical formulation

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \dots, N\} \subseteq \mathscr{X}^p \times \mathcal{K}$ be a training data set generated from an unknown joint probability distribution $\mathbb{P}$

- A response variable Y (also called output, target, outcome)
- A vector of p predictors $\boldsymbol{x}$ (also called inputs, features, attributes, explanatory variables)

The goal is to build a predictive model $\varphi : \mathscr{X} \to \mathscr{K}$ that minimizes the risk of making a wrong decision by computing

$$\mathcal{R}(\varphi) = \mathbb{E}_{X \times Y}\left[\ell(Y, \varphi))\right] = \int_{\mathscr{X} \times \mathscr{K}} \ell(y, \varphi(\boldsymbol{x}))\mathrm{d}\mathbb{P}(\boldsymbol{x}, y), \tag{1}$$

expected value of a specified loss $\ell(\cdot, \cdot) : \mathscr{K} \times \mathscr{K} \to \mathbb{R}$ penalizing every wrong decision.

✗ Equation (1) is however impossible to compute since $\mathbb{P}$ is unknown !!

Mathematical formulation

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\} \subseteq \mathscr{X}^p \times \mathcal{K}$ be a training data set generated from an unknown joint probability distribution $\mathbb{P}$

- A response variable Y (also called output, target, outcome)
- A vector of p predictors $\boldsymbol{x}$ (also called inputs, features, attributes, explanatory variables)

In practice we use the empirical risk minimization (ERM) principle as follows

$$\mathscr{R}(\varphi) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \varphi(x_i)). \tag{1}$$

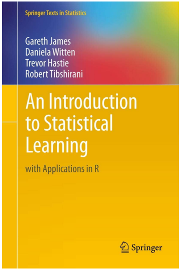
Note that if we have too much of training observations: $\mathscr{R}(\varphi) \underset{N \to \infty}{\longrightarrow} \mathcal{R}(\varphi)$

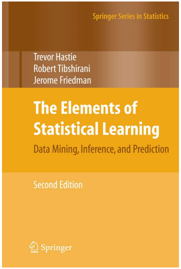
Objective

Learn an φ “optimal” model that minimizes Equation (1).

Recommended readings

- "An Introduction to Statistical Learning" (ISLR): emphasis on basic principles and application, no mathematical details. Available at https://www.statlearning.com/.
- "The Elements of Statistical Learning" (ESL): more mathematically advanced and theoretical. Available at http://statweb.stanfor edu/~tibs/ElemStatLearn





Overview

Supervised learning

Supervised classification

- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression

- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods

- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

Interpretability and explainability of ML methods

Bibliography

Binary classification

Example

Let us consider a binary classification problem, in which we need identify if the new observation is a Dog or a Cat.

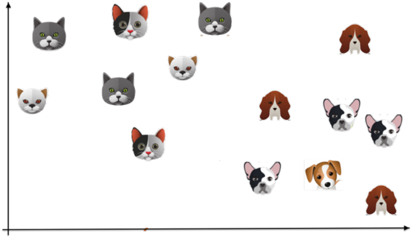


Figure: Dogs and Cats

Binary classification

Example

Let us consider a binary classification problem, in which we need identify if the new observation is a Dog or a Cat.

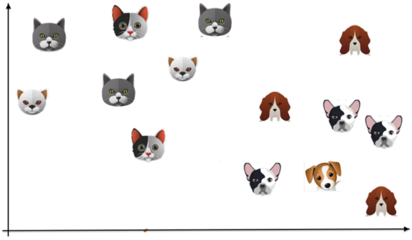


Figure: Dogs and Cats

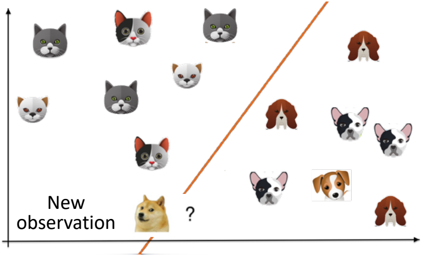


Figure: New observation

What class does the new observation belongs to?

It is the one that has *the highest probability (or score).*

Binary classification

Example

Let us consider a binary classification problem, in which we need identify if the new observation is a Dog or a Cat.

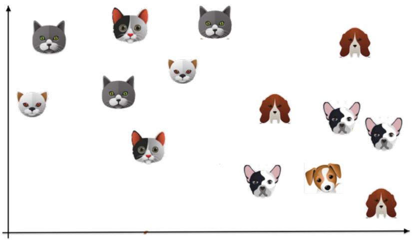


Figure: Dogs and Cats

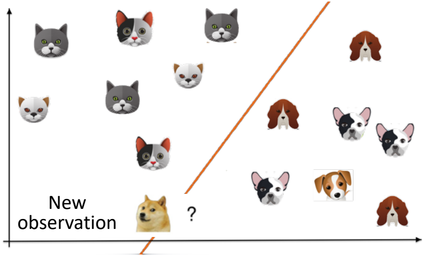


Figure: New observation

Let X be the new observation

$$P(X = Dog) = 0.9 \quad \text{and} \quad P(X = Cat) = 0.1$$

$$Dog \succ Cat \qquad \text{(Dog is preferred to Cat)}$$

Outline of classification problem

Given the training data $\mathscr{D} = \{x_i, y_i\}_{i=0}^{N} \subseteq \mathbb{R}^p \times \{m_a, \dots, m_e\}$:

Step ❶ Learning a classification rule: $\varphi : \mathcal{X} \to \mathcal{K}$.

Step ❷ Making decision on a new instance $\widehat{\varphi}(\boldsymbol{x}), \boldsymbol{x} \in \mathscr{T}$

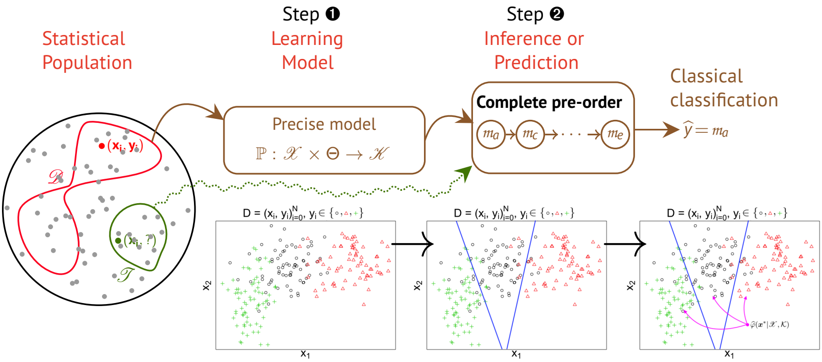


Figure: Supervised learning in a precise approach.

Mathematical formulation - Classification

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\} \subseteq \mathscr{X}^p \times \mathcal{K}$ be a training dataset

$$\mathcal{R}(\varphi) = \underset{\varphi \in \mathcal{K}}{\arg\min}\; \mathbb{E}_{X \times Y}\left[\ell(Y, \varphi))\right] \tag{2}$$

Under 1/0 loss function $\ell_{0/1}$, minimizing $\mathcal{R}$ is equivalent to

$$\phi(\boldsymbol{x}^* | \mathscr{D}) := \underset{m_k \in \mathcal{K}}{\arg\max}\; P(Y = m_k | X = \boldsymbol{x}^*), \tag{3}$$

where the last equation; (1) is also known as Bayes classifier and (2) predicts the class $\widehat{y}^* = \phi(\boldsymbol{x} | \mathscr{D})$ the most probable.

Mathematical formulation - Classification

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\} \subseteq \mathscr{X}^p \times \mathcal{K}$ be a training dataset

$$\mathcal{R}(\varphi) = \underset{\varphi \in \mathcal{K}}{\arg\min}\, \mathbb{E}_{X \times Y}\left[\ell(Y, \varphi))\right] \tag{2}$$

Under $1/0$ loss function $\ell_{0/1}$, minimizing $\mathcal{R}$ is equivalent to

$$\phi(\boldsymbol{x}^* | \mathscr{D}) := \underset{m_k \in \mathcal{K}}{\arg\max}\, P(Y = m_k | X = \boldsymbol{x}^*), \tag{3}$$

where the last equation; (1) is also known as Bayes classifier and (2) predicts the class $\widehat{y}^* = \phi(\boldsymbol{x}|\mathscr{D})$ the most probable.

In practice

Step ❶ Learning the conditional probability distribution $\mathbb{P}_{Y|\boldsymbol{x}}$.

Step ❷ Predicting the "optimal" label amongst $\mathcal{K} = \{ m_1, \ldots, m_K\}$:

$$m_{i_K} \succ m_{i_{K-1}} \succ \ldots \succ m_{i_1} \iff P(y = m_{i_K} | \boldsymbol{x}) > \ldots > P(Y = m_{i_1} | \boldsymbol{x})$$

☞ Pick out the most preferable label m_{i_K}

$\iff$ maximal probability plausible $P(y = m_{i_K} | \boldsymbol{x})$

Classical classification methods

Gaussian Discriminant Analysis

Assumptions: Conditional probability $P_{X|Y=m_k} \sim \mathcal{N}(\mu_k, \Sigma_{m_k}), \forall m_k$
Learning $\mathbb{P}$: Maximum likelihood estimation or Bayesian inference.

Discriminant analysis model	Assumptions ($\forall m_k \in \mathcal{K}$)	Parametric space ($\forall m_k \in \mathcal{K}$)
	Parametric Gaussian conditional distribution $\mathbb{P}_{X|Y=m_k}$	
Linear Discriminant [3, §4.3]	Homoscedasticity: $\Sigma_{m_k} = \Sigma$	$\Theta = \{\theta_{m_k} \| \theta_{m_k} = (\pi_{m_k}, \Sigma, \mu_{m_k})\}$
Quadratic Discriminant [3, §4.3]	Heteroscedasticity: $\Sigma_{m_k} = \Sigma_k$	$\Theta = \{\theta_{m_k} \| \theta_{m_k} = (\pi_{m_k}, \Sigma_k, \mu_{m_k})\}$
Naive Discriminant [3, §6.63]	Feature independence: $\Sigma_{m_k} = \boldsymbol{\sigma}_k^T \mathbb{I}$	$\Theta = \{\theta_{m_k} \| \theta_{m_k} = (\pi_{m_k}, \sigma_k, \mu_{m_k})\}$
Euclidean Discriminant [6]	Unit-variance feature indep.: $\Sigma_{m_k} = \mathbb{I}$	$\Theta = \{\theta_{m_k} \| \theta_{m_k} = (\pi_{m_k}, \mu_{m_k})\}$

Classical classification methods

Gaussian Discriminant Analysis

Assumptions: Conditional probability $P_{X|Y=m_k} \sim \mathcal{N}(\mu_k, \Sigma_{m_k}), \forall m_k$

Learning $\mathbb{P}$: Maximum likelihood estimation or Bayesian inference.

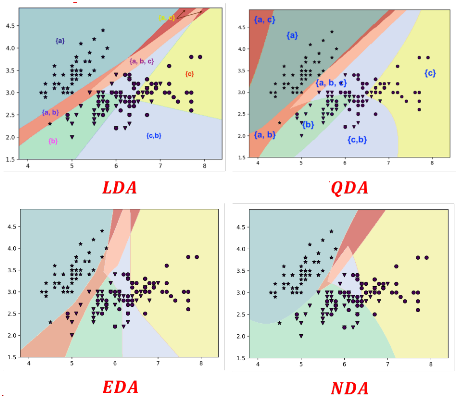


Figure: Gaussian discriminant models

Classical classification methods

Logistic regression

Assumptions:

If $Y = \{0, 1\}$ (binary classification) so $\mathbb{P}_{Y=1|x,\beta} \sim \mathcal{B}er(\psi(\beta^T x))$, where

$$P(Y = 1|X = x, \beta) := \psi(\beta^T x) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}}$$

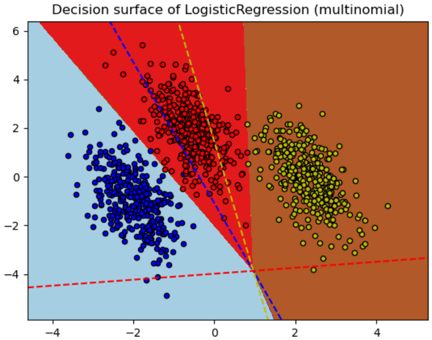
In case of multi-class classification, $\mathbb{P}_{Y|x,beta} \sim \mathcal{C}at(\beta_{m_1}, \ldots, \beta_{m_K})$.

$$P(Y = m_k|X = x, \beta_{m_k}) := \frac{e^{\beta_{m_k}^T x}}{\sum_{l=1}^{K} e^{\beta_{m_l}^T x}}$$

Learning $\mathbb{P}$: Maximum likelihood estimation or Bayesian inference (but both using approximative methods to get optimal value of parameter β_*

Classical classification methods

(Multi-class) Multinomial Logistic regression



Why it is linear?

➔ All points in the boundary must satisfy: $\{x : \psi(\beta_0^T x) = \psi(\beta_1^T x)\}$

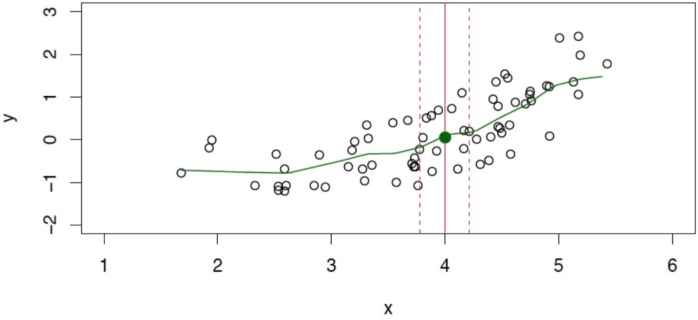
$$(\beta_0^T - \beta_1^T)x = 0$$

Classical classification methods

K-nearest neighbors algorithm

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\}$ dataset and a neighbourhood $N_k(\cdot)$ of K neighbors.

$$\psi(x) = \arg\max_{y \in \mathscr{K}} \frac{1}{K} \sum_{x_i \in N_K(x)} \mathbb{I}_{y==y_i}$$

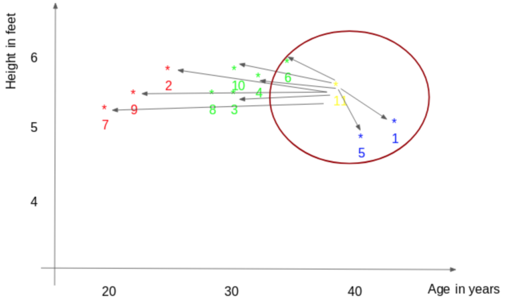


Classical classification methods

K-nearest neighbors algorithm

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\}$ dataset and a neighbourhood $N_k(\cdot)$ of K neighbors.

$$\psi(x) = \arg\max_{y \in \mathscr{K}} \frac{1}{K} \sum_{x_i \in N_K(x)} \mathbb{I}_{y == y_i}$$

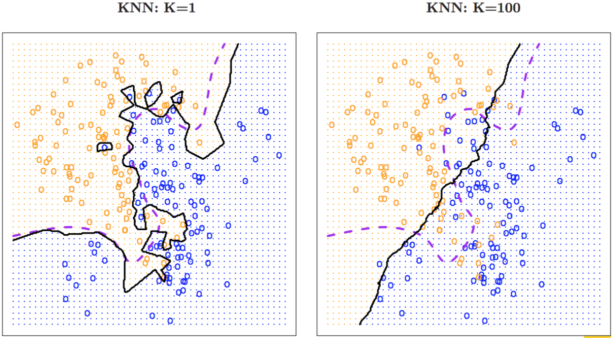


Classical classification methods

K-nearest neighbors algorithm

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\}$ dataset and a neighbourhood $N_k(\cdot)$ of K neighbors.

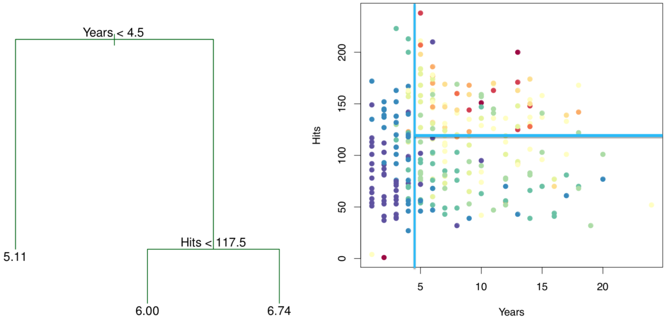
$$\psi(x) = \arg\max_{y \in \mathscr{K}} \frac{1}{K} \sum_{x_i \in N_K(x)} \mathbb{I}_{y==y_i}$$



Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

Prediction the salary in millions $= Y$

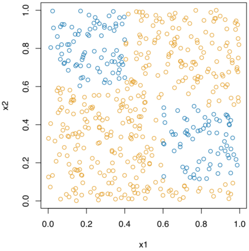


Low salary (blue, Green)
High salary (orange, red)

Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

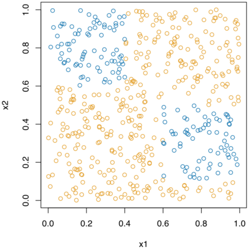
How does the algorithm work?

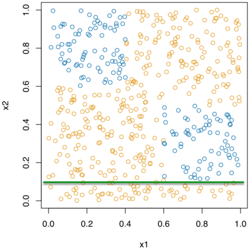


Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

How does the algorithm work?

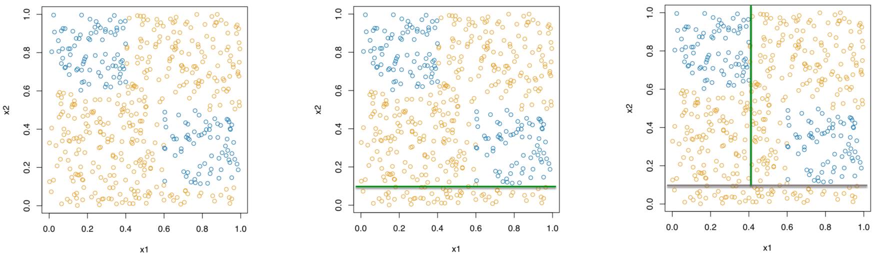




Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

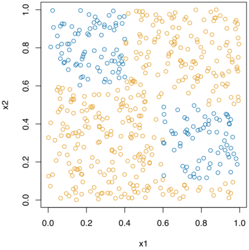
How does the algorithm work?

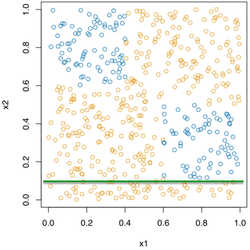


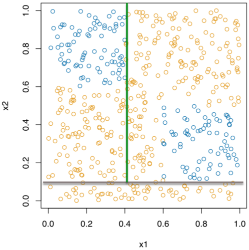
Classical classification methods

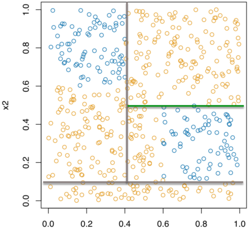
Tree-based models (Random Forest, Bagging, ...)

How does the algorithm work?





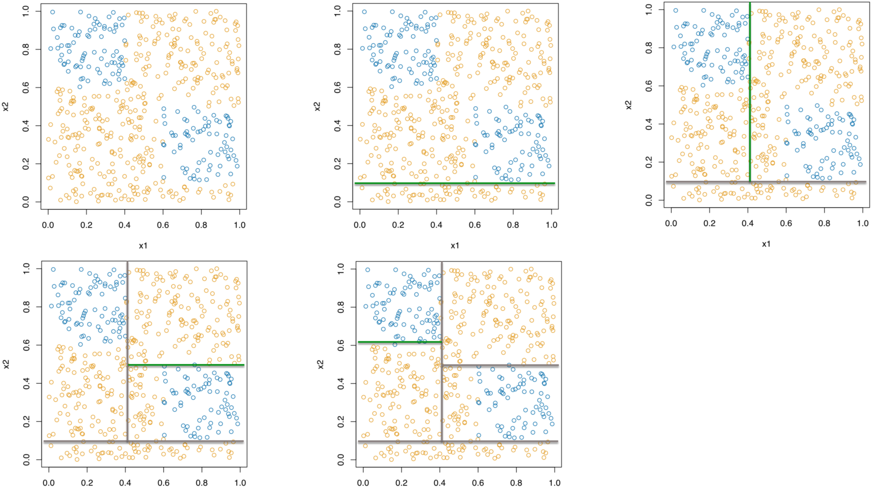




Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

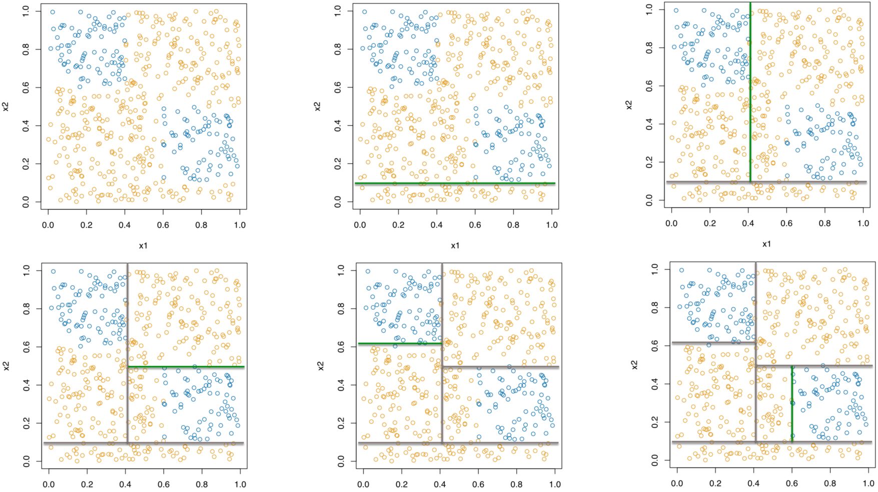
How does the algorithm work?



Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

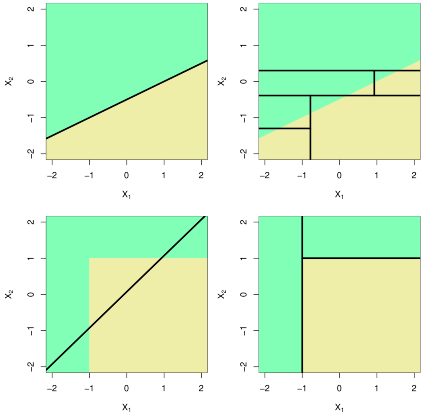
How does the algorithm work?



Classical classification methods

Tree-based models (Random Forest, Bagging, ...)

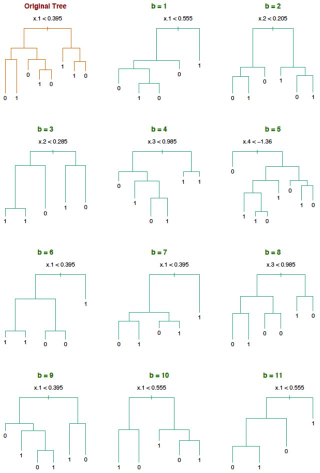
✗ Tree-based models are not always good !!



Classical classification methods

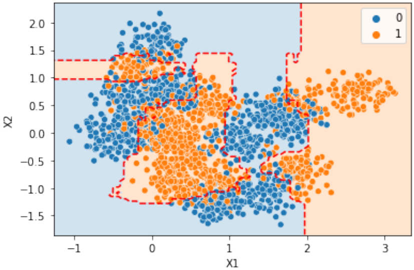
Tree-based models (Random Forest, Bagging, ...)

✓ Bagging or Random Forest method



✓ Weighted prediction

✓ Non-linear decision boundaries

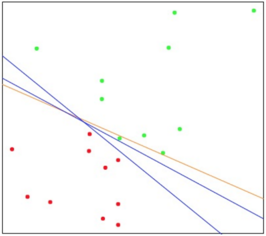


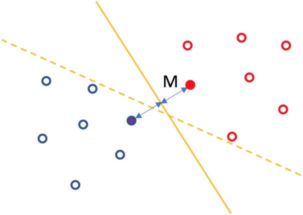
Classical classification methods

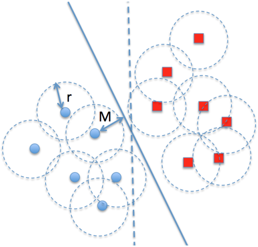
Support Vector Machine - Classification

The margin of H is the smallest distance between H and a vector x_i

$$M = \arg\min_i d(x_i, H) = \arg\min_i \frac{\beta^T(x - x_0)}{||\beta||} \tag{4}$$







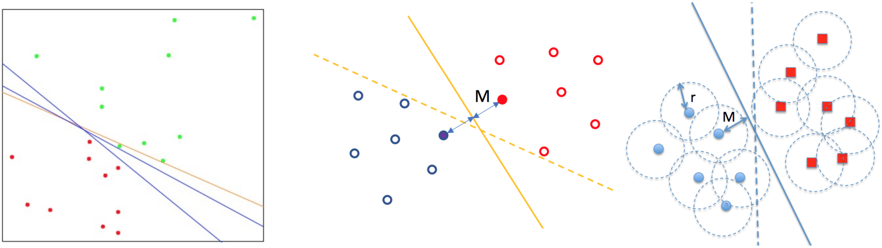
Classical classification methods

Support Vector Machine - Classification

The Optimal Separating Hyperplane is the hyperplane with the largest margin. It can be found by solving the optimization problem:

$$M \iff \arg\min_{\beta} \frac{1}{2}||\beta||^2 \tag{4}$$

$$\text{subject to} \quad y_i(\beta^T x_i + \beta_0) \geq 1 - \xi_i, i = 1, \ldots, n \tag{5}$$



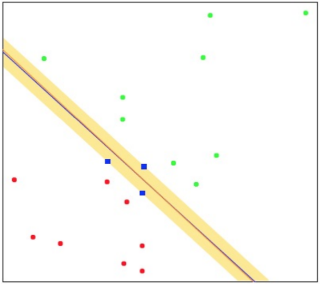
Classical classification methods

Support Vector Machine - Classification

The Optimal Separating Hyperplane is the hyperplane with the largest margin. It can be found by solving the optimization problem:

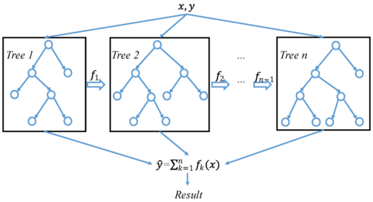
$$M \iff \arg\min_{\beta} \frac{1}{2}||\beta||^2 \tag{4}$$

$$\text{subject to} \quad y_i(\beta^T x_i + \beta_0) \geq 1 - \xi_i, i = 1, \ldots, n \tag{5}$$



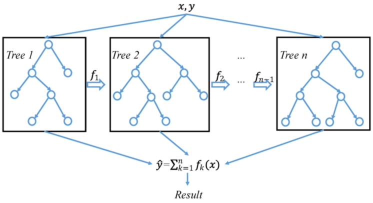
Others: XGBoost, Neural Network, Deep-Learning....

XGBoost

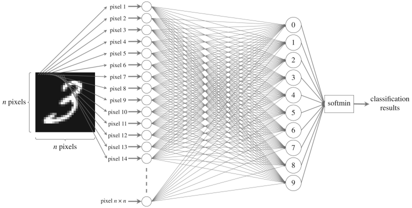


Others: XGBoost, Neural Network, Deep-Learning....

XGBoost

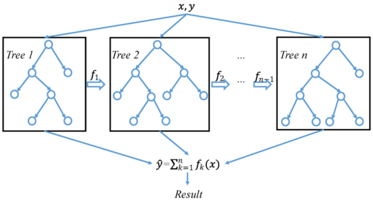


Neural Network

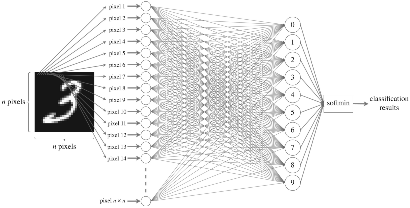


Others: XGBoost, Neural Network, Deep-Learning....

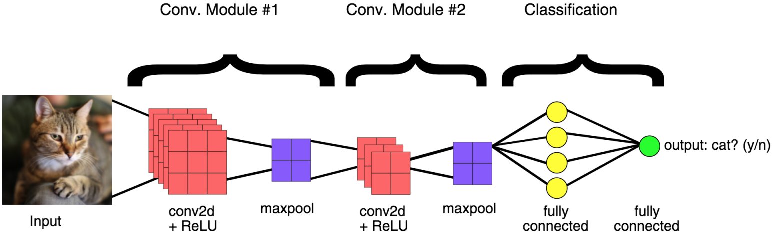
XGBoost



Neural Network



Deep-Convolution neural network



Example of classification

Let us do Machine Learning
Code source - [Link]

Overview

Supervised learning

Supervised classification
- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression
- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods
- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

Interpretability and explainability of ML methods

Bibliography

Mathematical formulation - Regression

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\} \subseteq \mathscr{X}^p \times \mathcal{K}$ be a training dataset

$$\mathcal{R}(\varphi) = \underset{\varphi \in \mathcal{F}}{\arg\min}\ \mathbb{E}_{X \times Y}\left[\ell(Y, \varphi))\right] \tag{6}$$

Under squared loss function, minimizing $\mathcal{R}$ is equivalent to

$$\varphi(\cdot) := \underset{\varphi \in \mathcal{F}}{\arg\min} \int_{\mathscr{X} \times \mathscr{Y}} (y - \varphi(\boldsymbol{x}))^2 \mathrm{d}\mathbb{P}(\boldsymbol{x}, y), \tag{7}$$

$$\varphi(\boldsymbol{x}^*) := \mathbb{E}(Y | X = \boldsymbol{x}^*), \tag{8}$$

where the last equation amounts to saying that

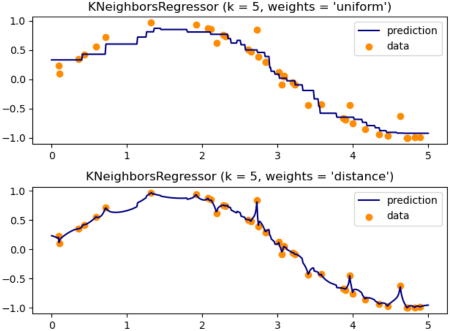
- Prediction may be interpreted as an average value.
- Again the conditional distribution $\mathbb{P}_{Y|X}$ is unknown.

Classical regression methods

K-nearest neighbors algorithm

Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\}$ dataset and a neighbourhood $N_k(\cdot)$ of K neighbors.

$$\varphi(x) = Ave\{y_i : x_i \in N_K(x)\} = \frac{1}{K} \sum_{x_i \in N_K(x)} y_i$$

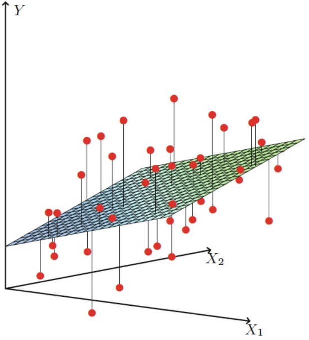


Classical regression methods

Linear regression

Assumptions: Expectation value can be written as a linear equation $\beta^T x$.

$$\varphi(\boldsymbol{x}_i^*) := Y_i := \underbrace{\beta^0 + \sum_{j=1}^{p} \beta^j x_i^j}_{\mathbb{E}(Y|X=\boldsymbol{x}^*)} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{9}$$



Let $\mathscr{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N\}$ a dataset. The most popular estimation method for β parameters is **least squares**, in which we minimize the sum of squared residuals (differences between y_i and $\varphi(\boldsymbol{x}_i^*)$). And where the optimal values of β is

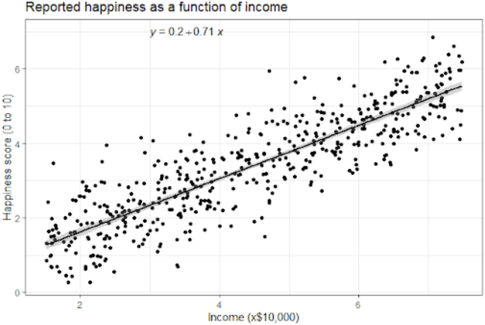
$$\beta = (X^T X)^{-1} X^T y$$

Classical regression methods

Linear regression

Assumptions: Expectation value can be written as a linear equation $\beta^T x$.

$$\varphi(\boldsymbol{x}_i^*) := Y_i := \underbrace{\beta^0 + \sum_{j=1}^{p} \beta^j x_i^j}_{\mathbb{E}(Y|X=\boldsymbol{x}^*)} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{9}$$



Classical regression methods

Regularized Linear regression

In order to avoid the overfitting and other issues, an regularized component is added:

$$\mathcal{R}^*(\varphi(x)) = \underset{\varphi \in \mathcal{F}}{\arg\min}\, \mathbb{E}\left[(Y - \varphi(x))^2 \middle| X = x\right] + \Upsilon(\varphi) \tag{10}$$

- Ridge regression
- Lasso regression
- Elastic net
- Principal component regression
- Partial least squares regression

Classical regression methods

Other regression methods

The base classification models used previously can also be adapted to the regression problem:

1. Tree-based model
2. Random Forest, Bagging, Boosting, ...
3. Support Vector Machine for regression
4. XGboost for regression
5. Deep-learning models

Example of regression

Let us do Machine Learning
Code source - [Link]

Overview

Supervised learning

Supervised classification

- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression

- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods

- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

Interpretability and explainability of ML methods

Bibliography

Multi-label classification problem

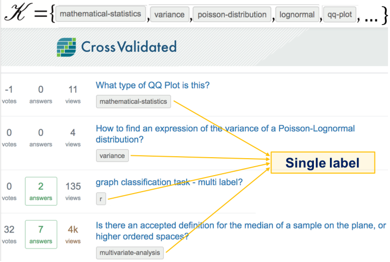
☞ The goal of multi-label problem:

Given a training data: $\mathscr{D} = \{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=0}^N \subseteq \mathbb{R}^p \times \mathscr{Y}$

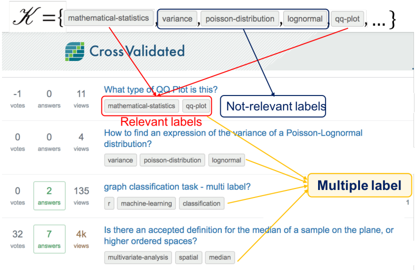
where: $\mathscr{Y} = \{0,1\}^m, \quad |\mathscr{Y}| = 2^m$

Learning a multi-label classification rule: $\varphi : \mathbb{R}^p \to \mathscr{Y}$

Classical classification



Multi-label classification



Label-wise ranking problem

☞ **The goal of label ranking problem:**

Given a training data: $\mathscr{D} = \{\boldsymbol{x}_i, Y_i\}_{i=0}^{N} \subseteq \mathbb{R}^p \times \Lambda(\mathscr{K})$

Learning a complete ranking rule: $\varphi : \mathbb{R}^p \rightarrow \Lambda(\mathscr{K})$

$\mathscr{D}$

X_1	X_3	Y
107.1	*Blue*	$m_1 \succ m_3 \succ m_2 \succ m_4$
-50	*Red*	$m_2 \succ m_3 \succ m_1 \succ m_4$
200	*Green*	$m_1 \succ m_4 \succ m_3 \succ m_2$
...	...	...

$\mathscr{D}_1$

X_1	X_3	Y
107.1	*Blue*	1
-50	*Red*	3
200	*Green*	1
...	...	...

$\mathscr{D}_2$

X_1	X_2	Y
107.1	*Blue*	3
-50	*Red*	1
200	*Green*	4
...	...	...

$\mathscr{D}_3$

X_1	X_2	Y
107.1	*Blue*	2
-50	*Red*	2
200	*Green*	3
...	...	...

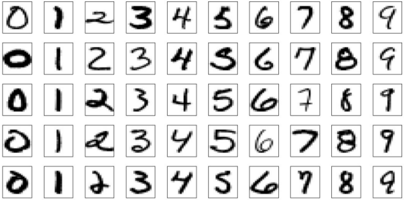
$\mathscr{D}_4$

X_1	X_2	Y
107.1	*Blue*	4
-50	*Red*	4
200	*Green*	2
...	...	...

Figure: Label-wise decomposition

Handwritten ZIP code.

Problem: Identify the numbers in a handwritten ZIP code, from a digitized image



The task is to recognize, from the matrix of pixel intensities, the digit in each image $(0, 1, \ldots, 9)$ quickly and accurately.
We can use any base classifier model

- Support Vector Machine
- Deep-learning models
- Others (logistic, ...)

Recognize the expression on a face.

Problem: Identify the expression on a face.

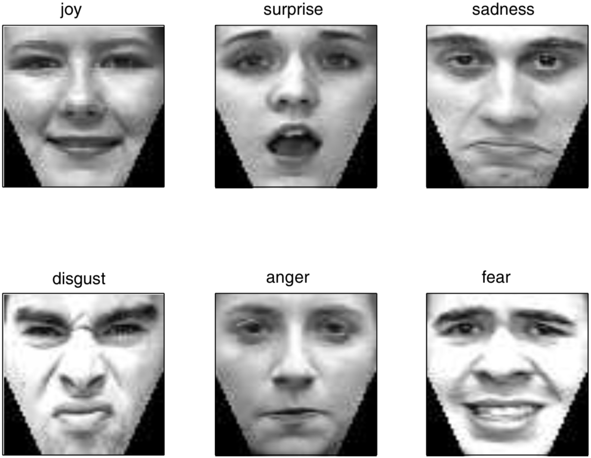
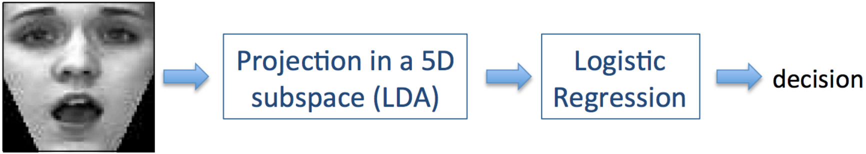


Figure: Expression recognition

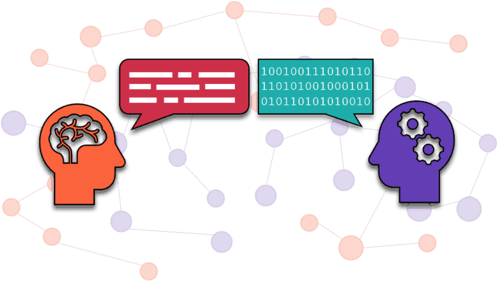
How can we solve these problems?

A simple approach may be:

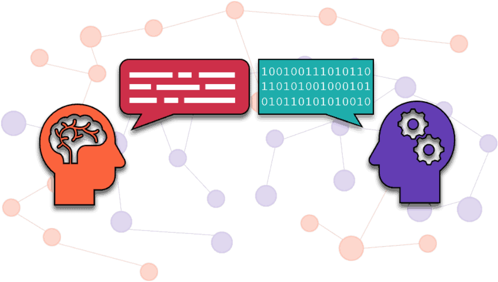
1. to save all pixel values of images in a record, in which their values is ranging in intensity from 0 to 255.
2. to use an unsupervised method to reduce the dimensionality of X input space, and then, to apply a base classifier method.

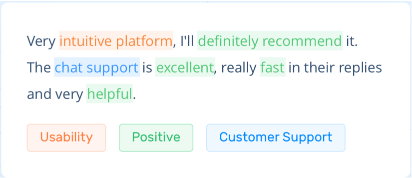


Natural Language Processing



Natural Language Processing

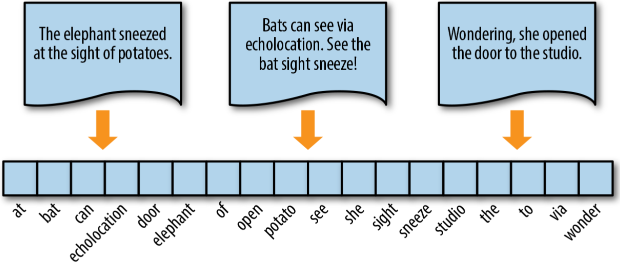




1. How can we work with unstructured data?
2. Are there mathematics tools?

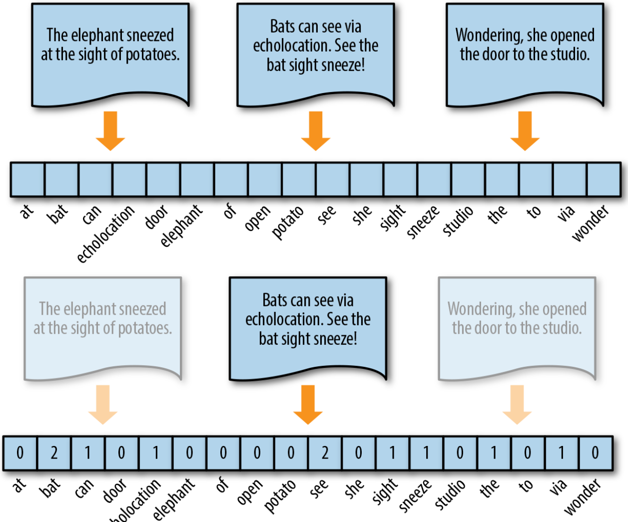
Representation as vector $\mathbb{R}$

Given three english texts



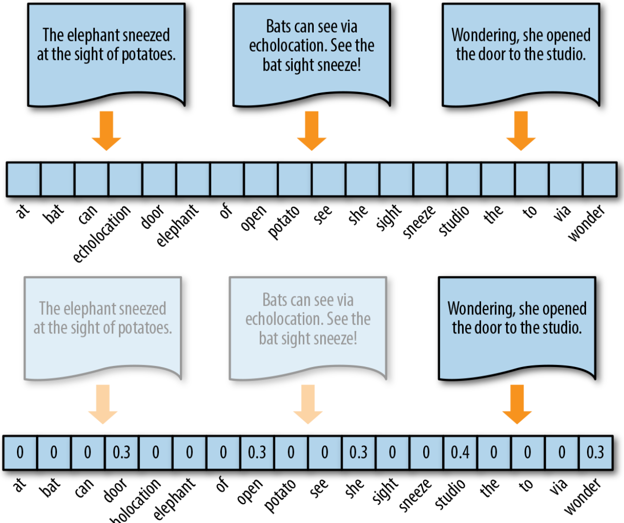
Representation as vector $\mathbb{R}$ - BagWords

Given three english texts



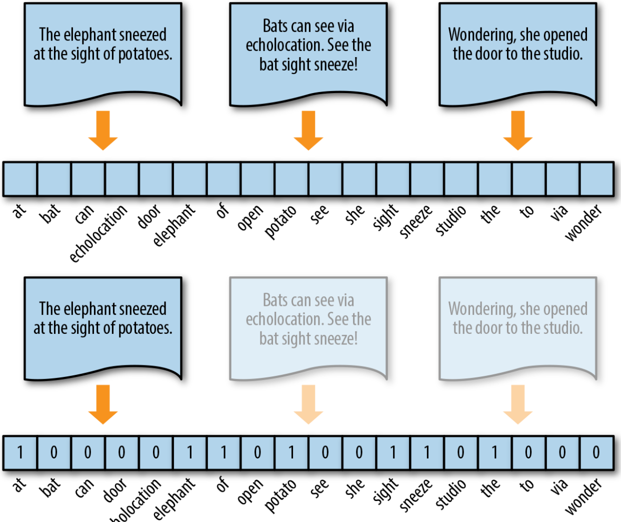
Representation as vector $\mathbb{R}$ - Term frequency

Given three english texts



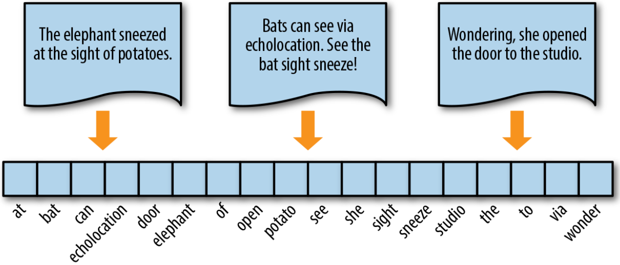
Representation as vector $\mathbb{R}$ - One-hot encoded

Given three english texts



Representation as vector $\mathbb{R}$

Given three english texts



- Word embedding is another powerful way to work with text.
- In an euclidian space, we can use any base classifier model.

Overview

Supervised learning

Supervised classification
- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression
- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods
- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

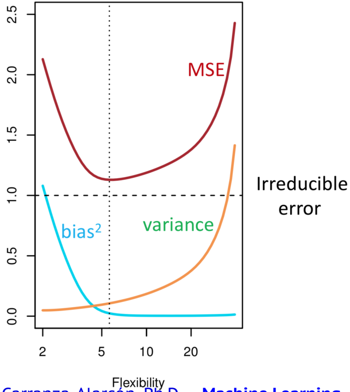
Interpretability and explainability of ML methods

Bibliography

Bias-variance tradeoff

How accurate can we be predicting?

$$\mathbb{E}\left[(y-\widehat{\varphi}(x))^2 \middle| X=\boldsymbol{x}\right] = \underbrace{\underbrace{\mathbb{V}ar[\widehat{\varphi}|X=x]}_{\text{Variance of } \widehat{f}_k} + \left(\underbrace{\mathbb{E}[\widehat{\varphi}]-\varphi}_{\text{Bias of } \widehat{\varphi}}\right)^2}_{\text{Reducible error}} + \underbrace{\sigma^2}_{\text{Irreducible error}}$$



- The more complex the model is, the more variance the model has.
- Inversely, The more simple the model is, the more bias the model has.

Overview

Supervised learning

Supervised classification
- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression
- Problem setting
- Classical methods
- Example of regression in Python

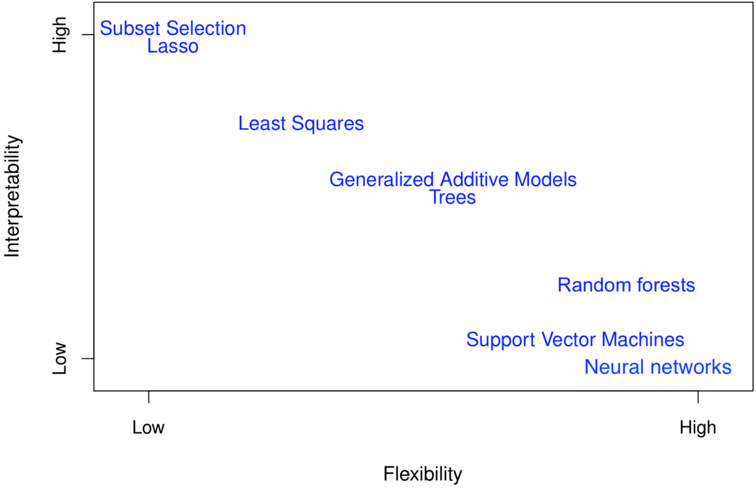
NLP and Other advanced supervised methods
- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

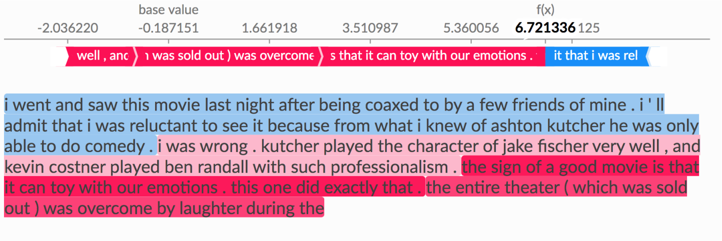
Interpretability and explainability of ML methods

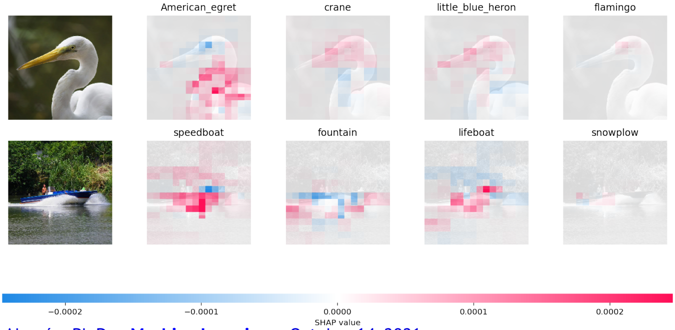
Bibliography

Interpretability/flexibility trade-off



Explainability ML using SHapley Additive exPlanations





Overview

Supervised learning

Supervised classification
- Problem setting
- Classical classification methods
- Example of classification in python

Linear regression
- Problem setting
- Classical methods
- Example of regression in Python

NLP and Other advanced supervised methods
- Multi-label and Label ranking
- Image patter recognition
- Natural Language Processing

Bias-variance tradeoff

Interpretability and explainability of ML methods

Bibliography

Bibliography I

[1] Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda. *Applied Text Analysis with Python*. url: https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html.

[2] Thierry Denoeux. *Lecture SY19 at UTC - Machine Learning*.

[3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer New York Inc., 2001.

[4] *Lecture 5 - Classification, Tree-Based Methods*. url: %7Bhttps://www.andrew.cmu.edu/user/achoulde/95791/lectures/lecture05/lecture05%5C_95791.pdf%7D.

Bibliography II

[5] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774.

[6] Virgil R Marco, Dean M Young, and Danny W Turner. "The Euclidean distance classifier: an alternative to the linear discriminant function". In: *Communications in Statistics-Simulation and Computation* 16.2 (1987), pp. 485–505.

[7] *Natural language processing (NLP): What is and it how does it work?* url: https://monkeylearn.com/natural-language-processing/.

Bibliography III

[8] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[9] *What is Natural Language Processing?* July 2021. url: https://navigate360.com/what-is-natural-language-processing/.

Thank You for Your Attention!