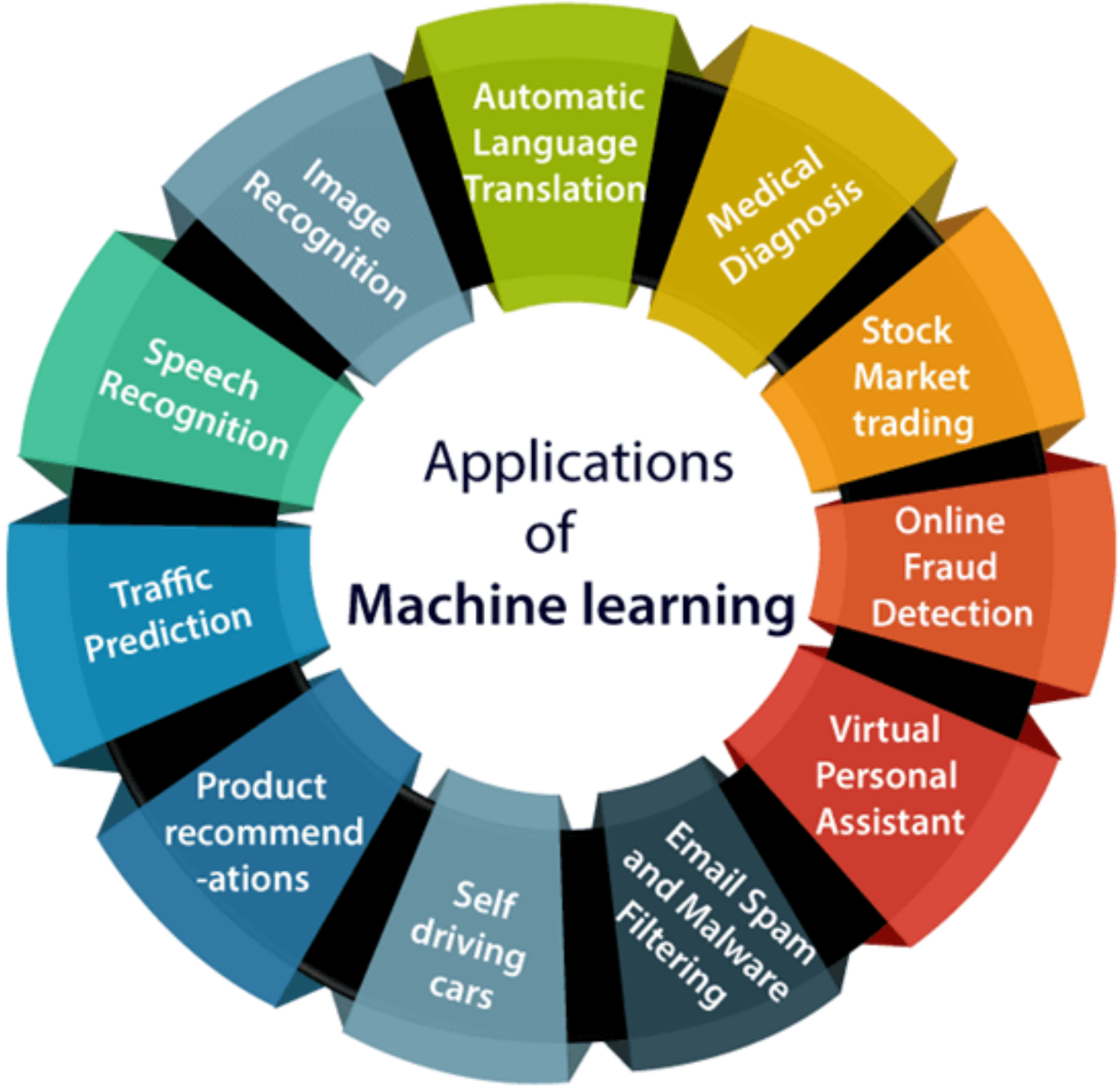
Midterm Review

ICS 435/635

DATA 435

Spring 2023

Any questions on HW3? All HW3 material will be on the exam.


Applications
of
Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition


AI In Our Everyday Lives
Telecom, GPS, Imagery
Airliners,
TripIt
Machine Learning in the Cloud
Fighter Aircraft
Unmanned Combat
Aerial Vehicle (UCAV)
Anticipatory/Predictive AI
(e.g. Amazon)
ADDTOCART
Smart Grid
Spam filtering
Optical Character
Recognition (OCR)
Search engines
Computer vision
Domotics
Building Industry
Personal Assistants
- Siri
- Google Now
- Cortana
- Viv
- Allo
- X.ai...
Security &
Surveillance networks
Self-Driving Cars
/ Smart vision
Delivery Drones
Robotic/Smart
Parking
Finance & Banking
Legal Litigations
High Frequency
Trading
Fraud Detection
Insurance
Predictive
Healthcare,
Watson
Shop Floor
Logistics
Service Robots
DuLight (Baidu)
Accessibility (Facebook)
© The AI Initiative, 2016

Logistic Regression: Applications

Applications [edit]

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by Boyd *et al.* using logistic regression.[6] Many other medical scales used to assess severity of a patient have been developed using logistic regression.[7][8][9][10] Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.).[11][12] Another example might be to predict whether a Nepalese voter will vote Nepali Congress or Communist Party of Nepal or Any Other Party, based on age, income, sex, race, state of residence, votes in previous elections, etc.[13] The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product.[14][15] It is also used in marketing applications such as prediction of a customer's propensity to purchase a product or halt a subscription, etc.[16] In economics, it can be used to predict the likelihood of a person ending up in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

Linear Regression: Applications

Epidemiology [edit]

Early evidence relating tobacco smoking to mortality and morbidity came from observational studies employing regression analysis. In order to reduce spurious correlations when analyzing observational data, researchers usually include several variables in their regression models in addition to the variable of primary interest. For example, in a regression model in which cigarette smoking is the independent variable of primary interest and the dependent variable is lifespan measured in years, researchers might include education and income as additional independent variables, to ensure that any observed effect of smoking on lifespan is not due to those other socio-economic factors. However, it is never possible to include all possible confounding variables in an empirical analysis. For example, a hypothetical gene might increase mortality and also cause people to smoke more. For this reason, randomized controlled trials are often able to generate more compelling evidence of causal relationships than can be obtained using regression analyses of observational data. When controlled experiments are not feasible, variants of regression analysis such as instrumental variables regression may be used to attempt to estimate causal relationships from observational data.

Finance [edit]

The capital asset pricing model uses linear regression as well as the concept of beta for analyzing and quantifying the systematic risk of an investment. This comes directly from the beta coefficient of the linear regression model that relates the return on the investment to the return on all risky assets.

Economics [edit]

Main article: Econometrics

Linear regression is the predominant empirical tool in economics. For example, it is used to predict consumption spending,[22] fixed investment spending, inventory investment, purchases of a country's exports,[23] spending on imports,[23] the demand to hold liquid assets,[24] labor demand,[25] and labor supply.[25]

Environmental science [edit]

> This section **needs expansion**. You can help by adding to it. *(January 2010)*

Linear regression finds application in a wide range of environmental science applications. In Canada, the Environmental Effects Monitoring Program uses statistical analyses on fish and benthic surveys to measure the effects of pulp mill or metal mine effluent on the aquatic ecosystem.[26]

K-Nearest Neighbors: Applications

A meta-analysis and review of the literature on the k-Nearest Neighbors technique for forestry applications that use remotely sensed data
G Chirici, M Mura, D McInerney, N Py... - Remote Sensing of ..., 2016 - Elsevier
...) to document development and **application** of **nearest neighbors** techniques with respect to ... characteristics, geographical regions of **applications**, accuracy and uncertainty measures, ...
☆ Save 99 Cite Cited by 141 Related articles All 9 versions

[PDF] Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background
SB Imandoust, M Bolandraftar - ... research and **applications**, 2013 - researchgate.net
... the **K nearest neighbors** when making predictions, ie, let the closest points among the **K nearest neighbors** ... W, one for each **nearest neighbor**, defined by the relative closeness of each ...
☆ Save 99 Cite Cited by 541 Related articles All 6 versions »

Application of K-nearest neighbors algorithm on breast cancer diagnosis problem.
M Sarkar, TY Leong - Proceedings of the AMIA Symposium, 2000 - ncbi.nlm.nih.gov
... number previous **nearest neighbor**) of classes. Delete the farthest of the **K-nearest neighbors**. One ... Include xi in the set of **K-nearest** contribution of each of the **K neighbors** based on ...
☆ Save 99 Cite Cited by 121 Related articles All 6 versions »

A Sensor Data Fusion System Based on *k*-Nearest Neighbor Pattern Classification for Structural Health Monitoring Applications
J Vitola, F Pozo, DA Tibaduiza, M Anaya - Sensors, 2017 - mdpi.com
... The **nearest neighbor** (NN) is a simple nonparametric and highly efficient technique [23] that ... in machine learning **applications** is the **k**-NN, also known as **k-nearest neighbors**. **k**-NN is ...
☆ Save 99 Cite Cited by 127 Related articles All 15 versions »

Application of data mining and feature extraction on intelligent fault diagnosis by artificial neural network and k-nearest neighbor
B Bagheri, H Ahmadi, R Labbafi - The XIX International ..., 2010 - ieeexplore.ieee.org
In this paper the frequency domain vibration signals of the gearbox of MF285 tractor is used for fault classification in three class: Healthy gear, Worn tooth face and broken gear. The ...
☆ Save 99 Cite Cited by 68 Related articles All 2 versions »

Application of the weighted k-nearest neighbor algorithm for short-term load forecasting
GF Fan, YH Guo, JM Zheng, WC Hong - Energies, 2019 - mdpi.com
In this paper, the historical power load data from the National Electricity Market (Australia) is used to analyze the characteristics and regulations of electricity (the average value of every ...
☆ Save 99 Cite Cited by 102 Related articles All 9 versions »

An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification
Y Li, B Cheng - 2009 17th International Conference on ..., 2009 - ieeexplore.ieee.org
... 1 ≠ **K** , NN algorithm can be extended to KNN algorithm. KNN tries to look for **K nearest neighbors** of x . Among these **K nearest neighbors**, if samples belonging to class i have the ...
☆ Save 99 Cite Cited by 63 Related articles All 3 versions

A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction
CH Cheng, CP Chan, YJ Sheu - ... **Applications** of Artificial Intelligence, 2019 - Elsevier
... approaches restrict on the **application** and performance of the ... a purity-based **k nearest neighbor** algorithm to improve the ... proposed purity-based **k nearest neighbor** algorithm to build a ...
☆ Save 99 Cite Cited by 52 Related articles All 2 versions »

Naïve Bayes: Applications

Deep feature weighting for naive Bayes and its application to text classification
L Jiang, C Li, S Wang, L Zhang - … Applications of Artificial Intelligence, 2016 - Elsevier
… naive Bayes and, in many cases, improves it dramatically. Besides, we apply the proposed deep feature weighting to some state-of-the-art naive Bayes … naive Bayes (NB) to naive Bayes …
☆ Save 99 Cite Cited by 313 Related articles All 3 versions »

The application of naive Bayes model averaging to predict Alzheimer's disease from genome-wide data
W Wei, S Visweswaran… - Journal of the American …, 2011 - academic.oup.com
… number of naive Bayes (NB) models. Design This model-averaged naive Bayes (MANB) … Its performance was compared to that of a naive Bayes algorithm without feature selection (…
☆ Save 99 Cite Cited by 102 Related articles All 7 versions

[PDF] Naive Bayesian classification approach in healthcare applications
R Bhuvaneswari, K Kalaiselvi - International Journal of Computer Science …, 2012 - ijcst.org
… This paper predicts the use of Naive Baye's classifier in medical applications. A major challenge facing healthcare organizations (hospitals, medical centers) is the provision of quality …
☆ Save 99 Cite Cited by 46 Related articles All 4 versions »

Related searches

naive bayes **text classifiers**
naive bayesian **learning algorithm**
naive bayes **classification**
bayesian **network for prediction** naïve bayes
weighted naïve bayes
multinomial naive bayes
naive bayes **model**
medical data classification naïve bayes **approach**

A novel application of naive bayes classifier in photovoltaic energy prediction
R Bayindir, M Yesilbudak, M Colak… - … and applications …, 2017 - ieeexplore.ieee.org
… of an installed photovoltaic system using the Naïve Bayes classifier. In the prediction process, … By means of the Naïve Bayes application, the sensitivity and the accuracy measures are …
☆ Save 99 Cite Cited by 41 Related articles All 5 versions

Discriminatively weighted naive Bayes and its application in text classification
L Jiang, D Wang, Z Cai - International Journal on Artificial …, 2012 - World Scientific
… Our motivation is to scale up the classification accuracy of naive Bayes without incurring the high time complexity. In IWNB, each training instance is firstly weighted according to the …
☆ Save 99 Cite Cited by 59 Related articles »

A feature dependent Naive Bayes approach and its application to the software defect prediction problem
ÖF Arar, K Ayan - Applied Soft Computing, 2017 - Elsevier
… Naive Bayes is one of the most widely used algorithms in … Naive Bayes is a probabilistic approach based on … steps, a Feature Dependent Naive Bayes (FDNB) classification method is …
☆ Save 99 Cite Cited by 143 Related articles All 4 versions »

Application of the Naïve Bayesian Classifier to optimize treatment decisions
J Kazmierska, J Malicki - Radiotherapy and Oncology, 2008 - Elsevier
… In this paper we will focus on the Naïve Bayesian Classifier, which in our opinion offers the … All calculations in this study were made with Naïve Bayesian Classifier implemented in WEKA …
☆ Save 99 Cite Cited by 97 Related articles All 9 versions

An application of naive bayes classification for credit scoring in e-lending platform
R Vedala, BR Kumar - … on Data Science & Engineering (ICDSE …, 2012 - ieeexplore.ieee.org
… the accuracy of prediction, we applied Naive Bayes classification on single loan table which … We used Multi Relational Naive Bayes classification for classifying the borrowers. A Bayes …
☆ Save 99 Cite Cited by 46 Related articles All 2 versions

Decision Trees: Applications

Data mining for meteorological applications: Decision trees for modeling rainfall prediction
A Geetha, GM Nasira - 2014 IEEE international conference on …, 2014 - ieeexplore.ieee.org
… **decision tree** evaluation can be quantified. This paper highlights a model using **decision tree** … walks of life in making wise and intelligent **decisions**. This model may be used in machine …
☆ Save 99 Cite Cited by 67 Related articles All 2 versions

Decision tree and ensemble learning algorithms with their applications in bioinformatics
D Che, Q Liu, K Rasheed, X Tao - Software tools and algorithms for …, 2011 - Springer
… Machine learning approaches have wide **applications** in bioinformatics, and **decision tree** is … review **decision tree** and related ensemble algorithms and show the successful **applications** …
☆ Save 99 Cite Cited by 151 Related articles All 9 versions »

An effective application of decision tree to stock trading
MC Wu, SY Lin, CH Lin - Expert Systems with **applications**, 2006 - Elsevier
… use various H values rather than a single one in the **application** of the **decision tree**. Empirical tests show that the two distinctions indeed improve the performance of the **decision tree**. …
☆ Save 99 Cite Cited by 154 Related articles All 4 versions

[PDF] **Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging**
H Schmid, F Laws - … of the 22nd International Conference on …, 2008 - aclanthology.org
… **Decision Trees** are turned into probability estimation **trees** by … The motivation was that a **tree** which predicts a single value (… Furthermore, we observed that such two-class **decision trees** …
☆ Save 99 Cite Cited by 242 Related articles All 11 versions »

Data mining for providing a personalized learning path in creativity: An application of decision trees
CF Lin, Y Yeh, YH Hung, RI Chang - Computers & Education, 2013 - Elsevier
… Data mining, especially with **decision tree** techniques, has been suggested to be an … system (PCLS) in which the **decision tree** technique is employed to provide adaptive learning paths …
☆ Save 99 Cite Cited by 247 Related articles All 6 versions

Application of decision trees to the analysis of soil radon data for earthquake prediction
B Zmazek, L Todorovski, S Džeroski, J Vaupotič… - Applied radiation and …, 2003 - Elsevier
… The main contribution of our work is that we have applied a new method (**decision trees**) to relate radon data to seismic activity. The results are encouraging. The learned **decision trees** …
☆ Save 99 Cite Cited by 117 Related articles All 14 versions

[BOOK] **Text mining with decision rules and decision trees**
C Apte, F Damerau, S Weiss - 1998 - Citeseer
… **applications**, such as automatic text categorizers and routers. **Decision** rules and **decision tree** based approaches to learning from text are particularly appealing, since rules and **trees** …
☆ Save 99 Cite Cited by 185 Related articles All 3 versions »

R-C4. 5 Decision tree model and its applications to health care dataset
Z Yao, P Liu, L Lei, J Yin - Proceedings of ICSSSM'05. 2005 …, 2005 - ieeexplore.ieee.org
… paper show that the predictive accuracy of **decision trees** constructed by R-C4.5s is not lower but sometimes higher than the predictive accuracy of **decision trees** constructed by C4.5. …
☆ Save 99 Cite Cited by 70 Related articles

[PDF] **Comparing decision fusion paradigms using k-NN based classifiers, decision trees and logistic regression in a multi-modal identity verification application**
P Verlinde, G Cholet - Proc. Int. Conf. Audio and Video-Based Biometric …, 1999 - Citeseer
… In our specific case (use of a **decision tree** as a fusion … **decision tree**, which is a specific kind of **decision tree** where each node has exactly two descending branches. In our **application**…
☆ Save 99 Cite Cited by 142 Related articles All 4 versions »

Random Forests: Applications

[HTML] Functional **random forest** with **applications** in dose-response predictions
R Rahman, SR Dhruba, S Ghosh, R Pal - Scientific reports, 2019 - nature.com
… **Random Forest** … **Random Forest** using functional data as compared to existing approaches have been shown using the HMS-LINCS dataset. In summary, Functional **Random Forest** …
☆ Save 99 Cite Cited by 48 Related articles All 9 versions

Multi-objective differential evolution based **random forest** for e-health **applications**
M Kaur, HK Gianey, D Singh… - Modern Physics Letters …, 2019 - World Scientific
… few decades for various medical **applications**. However, these techniques … **random forest** technique is proposed. The proposed technique is able to tune the parameters of **random forest** …
☆ Save 99 Cite Cited by 75 Related articles All 2 versions

A survey of **random forest** based methods for intrusion detection systems
PAA Resende, AC Drummond - ACM Computing Surveys (CSUR), 2018 - dl.acm.org
… , **Random Forest** models have been providing a notable performance on their **applications** in … In this work, we survey 35 **Random Forest** based methods for IDSs focused on network intru…
☆ Save 99 Cite Cited by 225 Related articles All 2 versions

Image quality transfer via **random forest** regression: **applications** in diffusion MRI
DC Alexander, D Zikic, J Zhang, H Zhang… - … Image Computing and …, 2014 - Springer
… We propose a framework for solving this problem using **random forest** regression to relate patches in the low-quality data set to voxel values in the high quality data set. Two examples …
☆ Save 99 Cite Cited by 102 Related articles All 6 versions »

Automatic selection of molecular descriptors using **random forest**: **Application** to drug discovery
G Cano, J Garcia-Rodriguez, A Garcia-Garcia… - … with **Applications**, 2017 - Elsevier
… In this paper we applied **Random Forest** as a feature selector but also as a classifier. We used public datasets to test the classification performance of the method. The main contribution …
☆ Save 99 Cite Cited by 108 Related articles All 8 versions

Random forest for bioinformatics
Y Qi - Ensemble machine learning: Methods and **applications**, 2012 - Springer
… **Application** of the **random forest** classification method to … Statistical **Applications** in Genetics and Molecular Biology 7(2… **Application** of the **random forest** classification algorithm to a seldi-…
☆ Save 99 Cite Cited by 611 Related articles All 13 versions »

Application of support vector machine, **random forest**, and genetic algorithm optimized **random forest** models in groundwater potential mapping
SA Naghibi, K Ahmadi, A Daneshi - Water Resources Management, 2017 - Springer
… study plans to apply support vector machine (SVM), **random forest** (RF), and genetic algorithm optimized **random forest** (RFGA) methods to assess groundwater potential by spring …
☆ Save 99 Cite Cited by 285 Related articles All 9 versions »

[HTML] **Application** of **random forest** for modelling of surface water salinity
MA Khan, MI Shah, MF Javed, MI Khan… - Ain Shams Engineering …, 2022 - Elsevier
… for inputs optimization and **Random Forest** are proposed to solve … , and the efficient **Random Forest** model structure able to … The best **Random Forest** structure that can yield the lowest …
☆ Save 99 Cite Cited by 26 Related articles All 3 versions

Applications of **random forest** in multivariable response surface for short-term load forecasting
GF Fan, LZ Zhang, M Yu, WC Hong, SQ Dong - International Journal of …, 2022 - Elsevier
… suggest that **random forests** may be the best classifiers. However, the **random forest** orthogonal … [18] were the first to study the tilted **random forest** in the context of time series forecasting. …
☆ Save 99 Cite Cited by 34 Related articles »

Analysis of dam behavior by statistical models: **application** of the **random forest** approach
A Belmokre, MK Mihoubi, D Santillán - KSCE Journal of Civil Engineering, 2019 - Springer
… Here, we develop an approach based on **random forest** regression for dam displacement prediction. **Random forest** regression is a non-parametric statistical technique that can deal …
☆ Save 99 Cite Cited by 29 Related articles All 4 versions

Support Vector Machines: Applications

Towards improving fuzzy clustering using support vector machine: Application to gene expression data
A Mukhopadhyay, U Maulik - Pattern Recognition, 2009 - Elsevier
… In this article, an attempt has been made in order to improve the performance of fuzzy clustering by combining it with **support vector machine** (SVM) classifier. A recently proposed real-…
☆ Save 99 Cite Cited by 61 Related articles All 5 versions

Applications of support vector machines to speech recognition
A Ganapathiraju, JE Hamaker… - IEEE transactions on …, 2004 - ieeexplore.ieee.org
… This paper addresses the use of a **support vector machine** as a classifier in a continuous … Recent related research [20] based on relevance **vector machines** (RVMs) directly addresses …
☆ Save 99 Cite Cited by 374 Related articles All 14 versions

Support vector machines in engineering: an overview
S Salcedo-Sanz, JL Rojo-Álvarez… - … : Data Mining and …, 2014 - Wiley Online Library
… This paper provides an overview of the **support vector machine** (… Kernel theory, SVMs, **support vector** regression (SVR), and … **support vector machine** (SVM) techniques and **applications** …
☆ Save 99 Cite Cited by 156 Related articles All 3 versions

Training support vector machines: an application to face detection
E Osuna, R Freund, F Girosit - Proceedings of IEEE computer …, 1997 - ieeexplore.ieee.org
… 3 SVM **Application**: Face Detection This section introduces a **Support Vector Machine** application for detecting vertically oriented and unoccluded frontal views of human faces in grey …
☆ Save 99 Cite Cited by 4015 Related articles All 18 versions

Support vector machine for regression and applications to financial forecasting
TB Trafalis, H Ince - Proceedings of the IEEE-INNS-ENNS …, 2000 - ieeexplore.ieee.org
… is to compare the **support vector machine** (SVM) developed …) Networks for financial forecasting **applications**. The theory of … The objective of this paper is to use **support vector machines** …
☆ Save 99 Cite Cited by 401 Related articles All 13 versions »

Applications of support vector machines in chemistry
O Ivanciuc - Reviews in computational chemistry, 2007 - Wiley Online Library
… the basis of **support vector machines**, followed by a section on linear **support vector machines** in … The OSH computation with a linear **support vector machine** is presented in this section. …
☆ Save 99 Cite Cited by 540 Related articles All 6 versions

Support vector machine classifier for diagnosis in electrical machines: Application to broken bar
D Matić, F Kulić, M Pineda-Sánchez… - … Systems with **Applications**, 2012 - Elsevier
… a **support vector machine** classifier for broken bar detection in electrical induction **machine**. It is a … For classification task **support vector machine** is used due to its good robustness and …
☆ Save 99 Cite Cited by 89 Related articles All 4 versions

Biological applications of support vector machines
ZR Yang - Briefings in bioinformatics, 2004 - academic.oup.com
… that give the best prediction performance are **support vector machines** (SVMs). This is … This paper will discuss the principles of SVMs and the **applications** of SVMs to the analysis …
☆ Save 99 Cite Cited by 235 Related articles All 11 versions

Support vector machine applications in computational biology
WS Noble - Kernel methods in computational biology, 2004 - books.google.com
During the past 3 years, the **support vector machine** (SVM) learning algorithm has been extensively applied within the field of computational biology. The algorithm has been used to …
☆ Save 99 Cite Cited by 445 Related articles All 14 versions

Support vector machine applications in the field of hydrology: a review
PC Deka - Applied soft computing, 2014 - Elsevier
… of **Support vector machines** (SVMs) as well as algorithmic strategies for implementing them, and **applications** of … SVMs introduced by Vapnik and others in the early 1990s are **machine** …
☆ Save 99 Cite Cited by 533 Related articles All 6 versions »

K-Means Clustering: Applications

Social media analysis using optimized K-Means clustering
A Alsayat, H El-Sayed - … , Management and **Applications** (SERA …, 2016 - ieeexplore.ieee.org
… The process continues as we apply optimized **K-Means clustering** algorithm to find different
… Optimized **Cluster** Distance (OCD) to improve the performance of ordinary **K-Means** algorithm…
☆ Save 99 Cite Cited by 40 Related articles All 2 versions

An improved overlapping k-means clustering method for medical applications
S Khanmohammadi, N Adibeig… - … Systems with **Applications**, 2017 - Elsevier
… one **cluster**. One of the simplest and most efficient overlapping **clustering** methods is known
as overlapping **k-means** (OKM), which is an extension of the traditional **k-means** algorithm. …
☆ Save 99 Cite Cited by 184 Related articles All 3 versions

Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)
P Govender, V Sivakumar - Atmospheric pollution research, 2020 - Elsevier
… overview of two commonly used **clustering** techniques ie **k-means** and hierarchical, that have
… The aim of this paper was to provide a review of the **clustering applications**, in particular by …
☆ Save 99 Cite Cited by 224 Related articles

K-means clustering algorithm for multimedia applications with flexible HW/SW co-design
F An, HJ Mattausch - Journal of Systems Architecture, 2013 - Elsevier
… **K-means clustering** algorithm with high flexibility and high performance for machine learning,
pattern recognition and multimedia **applications**… cost of a **K-means clustering** algorithm. The …
☆ Save 99 Cite Cited by 32 Related articles All 3 versions »

Clustering transformed compositional data using *K*-means, with applications in gene expression and bicycle sharing system data
A Godichon-Baggioni… - Journal of Applied …, 2019 - Taylor & Francis
… Many algorithms have been proposed to implement **K-means clustering**, and we consider
the well-known one introduced by MacQueen [27]. Note that minimizing the SSE is known to …
☆ Save 99 Cite Cited by 40 Related articles All 19 versions

Network traffic classification using k-means clustering
Y Liu, W Li, Y Li - Second international multi-symposiums on …, 2007 - ieeexplore.ieee.org
… because the unsupervised **K-means** didn't require a training set to be hand-classified in
advance. Thus, new **applications** can be identified by grouping a separate **cluster**. Considering …
☆ Save 99 Cite Cited by 137 Related articles All 5 versions »

[PDF] K-means cluster analysis for image segmentation
SMA Burney, H Tariq - … Journal of Computer **Applications**, 2014 - academia.edu
… This paper provides an overview of **K-Means clustering** method for color image segmentation
along with custom employment of color space that has been appeared in the past and …
☆ Save 99 Cite Cited by 107 Related articles All 4 versions »

The application on intrusion detection based on k-means cluster algorithm
M Jianliang, S Haikun, B Ling - … Technology and **Applications**, 2009 - ieeexplore.ieee.org
… We use the **K-means** algorithm to **cluster** and analyze the data in this paper. Computer
simulations show that this method can detect unknown intrusions efficiently in the real network …
☆ Save 99 Cite Cited by 191 Related articles All 5 versions

[PDF] Face extraction from image based on K-means clustering algorithms
Y Farhang - … Computer Science and **Applications**, 2017 - pdfs.semanticscholar.org
… and FE-RER-**clustering** algorithms. This study showed that the **K-means clustering** algorithm
… , reduced the number of iterations, intra **cluster** distance, and the related processing time. …
☆ Save 99 Cite Cited by 15 Related articles All 2 versions »

A scalable system for executing and scoring K-means clustering techniques and its impact on applications in agriculture
N Golubovic, C Krintz, R Wolski… - … Journal of Big …, 2019 - inderscienceonline.com
… In this section we focus on **K-means clustering** for multivariate correlated data. We also
discuss the application and need for such systems in the context of farm analytics when …
☆ Save 99 Cite Cited by 8 Related articles All 7 versions »

[PDF] Crime analysis using k-means clustering
J Agarwal, R Nagpal, R Sehgal - … Journal of Computer **Applications**, 2013 - academia.edu
… constitutes a **cluster** and how to efficiently find them. In this paper **k means clustering** technique
… In this paper k mean **clustering** is implemented using open source data mining tool which …
☆ Save 99 Cite Cited by 129 Related articles All 7 versions »

K-means clustering in wireless sensor networks
P Sasikumar, S Khara - 2012 Fourth international conference …, 2012 - ieeexplore.ieee.org
… **Clustering** algorithms are often useful in **applications** in various fields such as visualization,
… Practical **applications** [12] of **clustering** include pattern classification under unsupervised …
☆ Save 99 Cite Cited by 202 Related articles All 5 versions

You should now be able to understand a vast array of machine learning-related research papers and technical reports

Example paper 1

www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

Machine Learning analysis of high-grade serous ovarian cancer proteomic dataset reveals novel candidate biomarkers

Federica Farinella[1,8], Mario Merone[2,8]✉, Luca Bacco[2,3,7], Adriano Capirchio[4,5], Massimo Ciccozzi[6] & Daniele Caligiore[4,5]

Figure 1. Machine Learning pipeline.

Materials and methods

Database. For this study, we used the publicly available database generated by the Clinical Proteomic Tumor Analysis Consortium (CPTAC) [24]. The Decision Support System (DSS) was trained, tested, and validated using the CPTAC Ovarian Cancer Confirmatory Study Proteomic Dataset, which includes the analysis form Ovarian tissue sample from a cohort of 100 individuals with HGSOC and 25 Non-Tumor ovarian samples, performed by the Johns Hopkins University (JHU) and Pacific Northwest National Laboratory (PNNL) using isobaric Tags for Relative and Absolute Quantification (iTRAQ) protein quantification method[25]. Clinical features were present only for Tumor patients. The Tumor cohort was composed of women ranging from 36 to 85 years, with an average age of 59. The 7% of the participants had an history of other malignancies. The anatomic site of origin of tumor specimens are: ovary 52%, omentum 41%, peritoneum 3%, pelvic mass 3% and unknown origin 1%. All samples are classified as "Serous Adenocarcinoma". FIGO staging ranges from IIB to IV (not specified whether A or B), with the majority of the samples classified as stage IIIC (63.8%), followed by IV (15.2%), IIIB (7.6%), IIIA (2.9%), IC (1.9%), IIB (1%) and a remaining 7.6% of specimens having uncertain classification. The 80.8% of the samples are classified as Grade 3, 5.8% as Grade 2, 0.9% as Grade 1, while for 12.5% of the samples grading was not reported. The efficacy of the DSS was further tested on the dataset generated from the CPTAC and TCGA Cancer Proteome Study of Ovarian Tissue, including the analysis of samples from 174 Ovarian tumors, of which 169 from HGSOC, also performed by JHU and PNNL using iTRAQ[26]. Cohort is composed of women ranging from 35 years to 87, with an average age of 60.5. Tumor tissue site is Ovary for 98% of the samples, Omentum in 1% of the samples and Peritoneum ovary in 1%. All samples are classified as "Serous Cystadenocarcinoma". FIGO staging of the samples goes from stage IC to IV (not specified whether A or B), where stage IIIC accounts for 69.9% of the samples, IV for 17%, IIIB and IIC accounting each one for 4.4%, IC for 1.5%, and IIA, IIB and IIA accounting each one for 1%. The 81.5% of the samples are Grade 3, 16.5% are Grade 2, 1% are Grade 1, while grading is unknown for 1% of the samples. Datasets were subsequently processed in Python (distribution 3.9.1) using NumPy and pandas libraries to merge JHU and PNNL datasets and remove protein columns containing more than 10% of missing values. After that, the data were processed and analyzed using a software tool coded in MATLAB2020b (Mathworks Inc., MA).

Feature selection based on relief method. All the features selected from the Correlation Analysis are then examined with a second feature selection step based on the ReliefF algorithm[27]. Such an algorithm ranks the importance of the features with respect to the target value. The importance of a feature is represented by the weight of that feature. The values of those weights can range from -1 to 1, with the largest positive weights assigned to the most important features. The algorithm penalizes the features that provide different values to k neighbors of the same class while rewarding the ones that provide different values to k neighbors of different classes.

Decision tree. The features (i.e. the proteins) selected by the reliefF method are used to train the CART[28] algorithm for the binary (Tumor/Non-Tumor) classification task. We chose to use a decision tree classifier for its high interpretability and explainability, unlike other methods of machine and deep learning. The CART tree is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly, beginning from the root node that contains the whole learning sample. The basic idea of the tree growth is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". The purity metric defines a node as 100% impure when its samples evenly belong (50:50) to both the classes while defining a node as 100% pure when all of its data belongs to a single class. In this algorithm, only univariate splits are considered. That is, each split depends on the value of just one feature. At node t, the best split s is chosen to maximize a splitting criterion $\Delta i(s, t)$. When the impurity measure for a node can be defined, the splitting criterion corresponds to a decrease in impurity. In our case, we used a Gini criterion as the impurity measure. During the training, we chose not to impose a control on the tree's depth, fixing the maximum number of splits as the size of the training set -1 and the minimum leaf size (the minimum number of samples in the leafs) as 1. Furthermore, we fixed the cost of classifying a sample into class j if its true class is i equal to:

- $C_{i,j} = 1$, if $i \neq j$
- $C_{i,j} = 0$, if $i = j$

We decided also not to implement a pruning strategy.

Performance evaluation. To evaluate the performance of our system we computed the confusion matrix. A confusion matrix is an N × N matrix used for evaluating the performance of a classification model, where N is the number of target classes. In our case, the task performed by the model is a binary classification task, thus N is equal to 2. From the confusion matrix we calculated the classification accuracy $\left(Acc = \frac{TP+TN}{P+N}\right)$, the precision per class ($P_{Tumor} = \frac{TP}{TP+FP}$ and $P_{NonTumor} = \frac{TN}{TN+FN}$), sensitivity and specificity $\left(Sensitivity = \frac{TP}{P}, Specificity = \frac{TN}{N}\right)$. Furthermore for each class we compute the F1 score, a relevant metric in case of unbalanced dataset, $F1_{Tumor} = 2 * \left(\frac{P_{Tumor} * Sensitivity}{P_{Tumor} + Sensitivity}\right)$ and $F1_{NonTumor} = 2 * \left(\frac{P_{NonTumor} * Specificity}{P_{NonTumor} + Specificity}\right)$.

As usual, P and N denote the number of positive patients (with Tumor) and negative patients (Non-Tumor) records, whereas TP, TN, FP and FN stands respectively for true positive, true negative, false positive and false negative classifications. A true positive classification implies that the patients are correctly detected by the system as patients without tumor, whereas a true negative classification indicates that the system correctly recognizes the patients with HGSOC. We developed two main performance test:

- *Test 1* This test is developed to evaluate the performance of the system only on CPTAC dataset using a 5-fold cross-validation procedure as follows. First, we randomly shuffled the dataset and split it into 5 groups. For each group, a single group is taken as a hold out or test data set and the remaining groups as a training data set. After training and test, the evaluation score is retained and the model is discarded. This operation is then repeated for each group. Importantly, each sample in the data set is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set once and used to train the model 4 times. This procedure results in a less biased or less optimistic estimate of the system performance than other methods, such as a simple train/test split.
- *Test 2* This test is developed to evaluate the robustness of our system. We trained the system on CPTAC Dataset and tested it on a different dataset called Cancer Proteome Study of Ovarian Tissue (TCGA). This latter dataset is composed of 216 tumor patients.

	Tumor
Positive correlation	20
Negative correlation	117
Noncorrelation	6086

Table 1. Here are summarized the results of the correlation between proteomics data and tumor phenotype. It appears that a vast portion of the proteins displayed no evident correlation, and the majority of the proteins were negatively correlated.

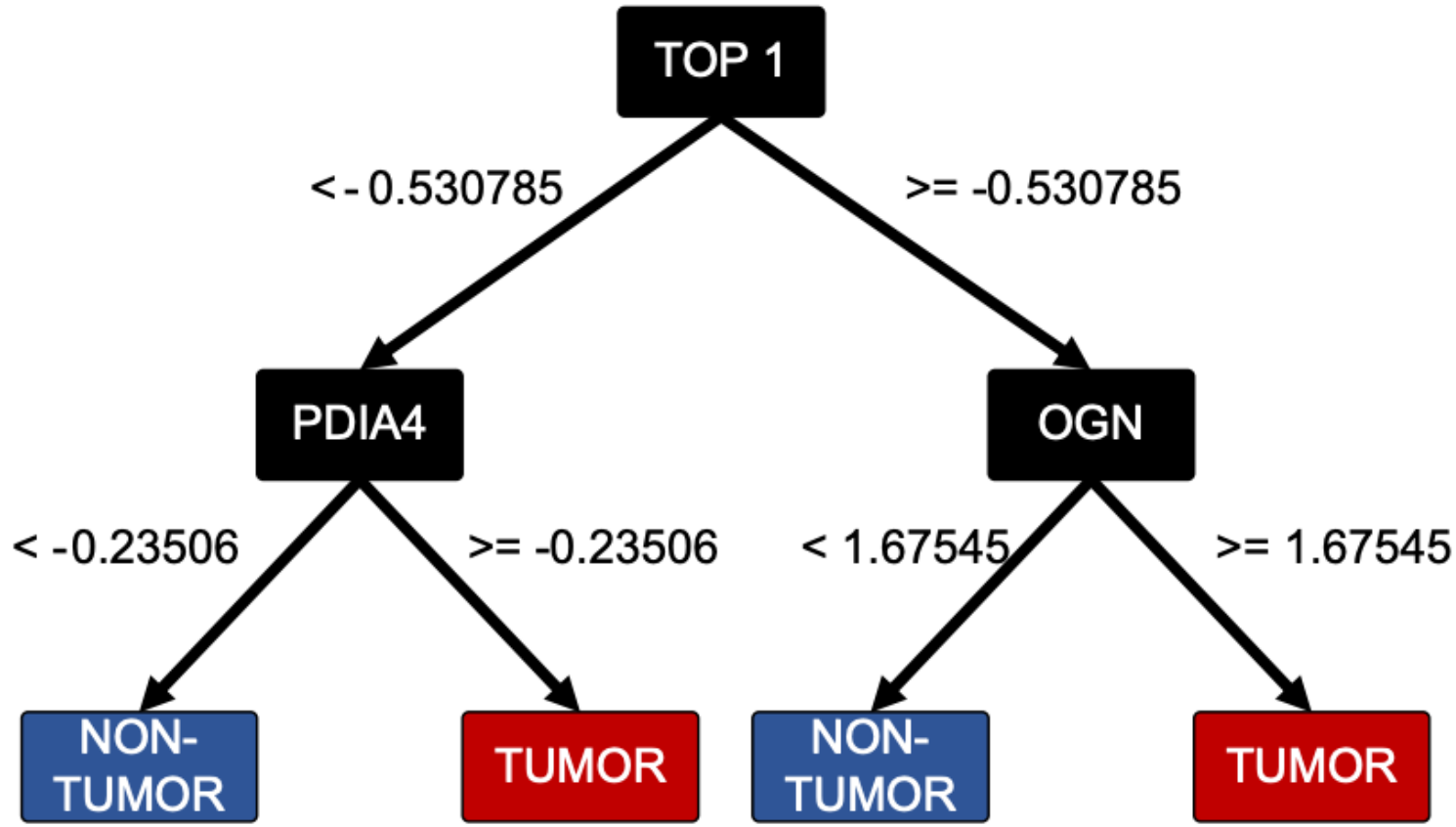


Figure 2. Final decision tree, with focus on the biomarkers.

Pred.	Truth	
	Non-tumor	Tumor
Non-tumor	40	3
Tumor	1	165

Table 2. This Confusion Matrix is achieved in fivefold-cross-validation on CPTAC Ovarian Cancer Confirmatory Study Proteomic Dataset (209 samples). The matrix compares the actual target values (Truth) with those predicted (Pred.) by our model. On first diagonal are reported the samples correctly classified, whereas on second diagonal are reported the misclassified samples.

Pred.	Truth	
	Non-tumor	Tumor
Non-tumor	0	6
Tumor	0	210

Table 3. This Confusion Matrix reports the performance of our system trained on CPTAC Ovarian Cancer Confirmatory Study Proteomic Dataset and tested on TCGA Cancer Proteome Study of Ovarian Tissue (216 samples). The matrix compares the actual target values (Truth) with those predicted (Pred.) by our model. On first diagonal are reported the samples correctly classified, whereas on second diagonal are reported the misclassified samples. The TCGA dataset only presents samples from the Tumor class.

Example paper 2

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-020-18684-2 OPEN

Machine learning based early warning system enables accurate mortality risk prediction for COVID-19

Yue Gao[1,2,10], Guang-Yao Cai[1,2,10], Wei Fang[3,10], Hua-Yi Li[1,2,10], Si-Yuan Wang[1,2,10], Lingxi Chen[4,10], Yang Yu[1,2], Dan Liu[1,2], Sen Xu[1,2], Peng-Fei Cui[1,2], Shao-Qing Zeng[1,2], Xin-Xia Feng[5], Rui-Di Yu[1,2], Ya Wang[1,2], Yuan Yuan[1,2], Xiao-Fei Jiao[1,2], Jian-Hua Chi[1,2], Jia-Hao Liu[1,2], Ru-Yuan Li[1,2], Xu Zheng[1,2], Chun-Yan Song[1,2], Ning Jin[1,2], Wen-Jian Gong[1,2], Xing-Yu Liu[1,2], Lei Huang[6], Xun Tian[6], Lin Li[7], Hui Xing[7], Ding Ma[1,2], Chun-Rui Li[8], Fei Ye[9]✉ & Qing-Lei Gao[1,2]✉

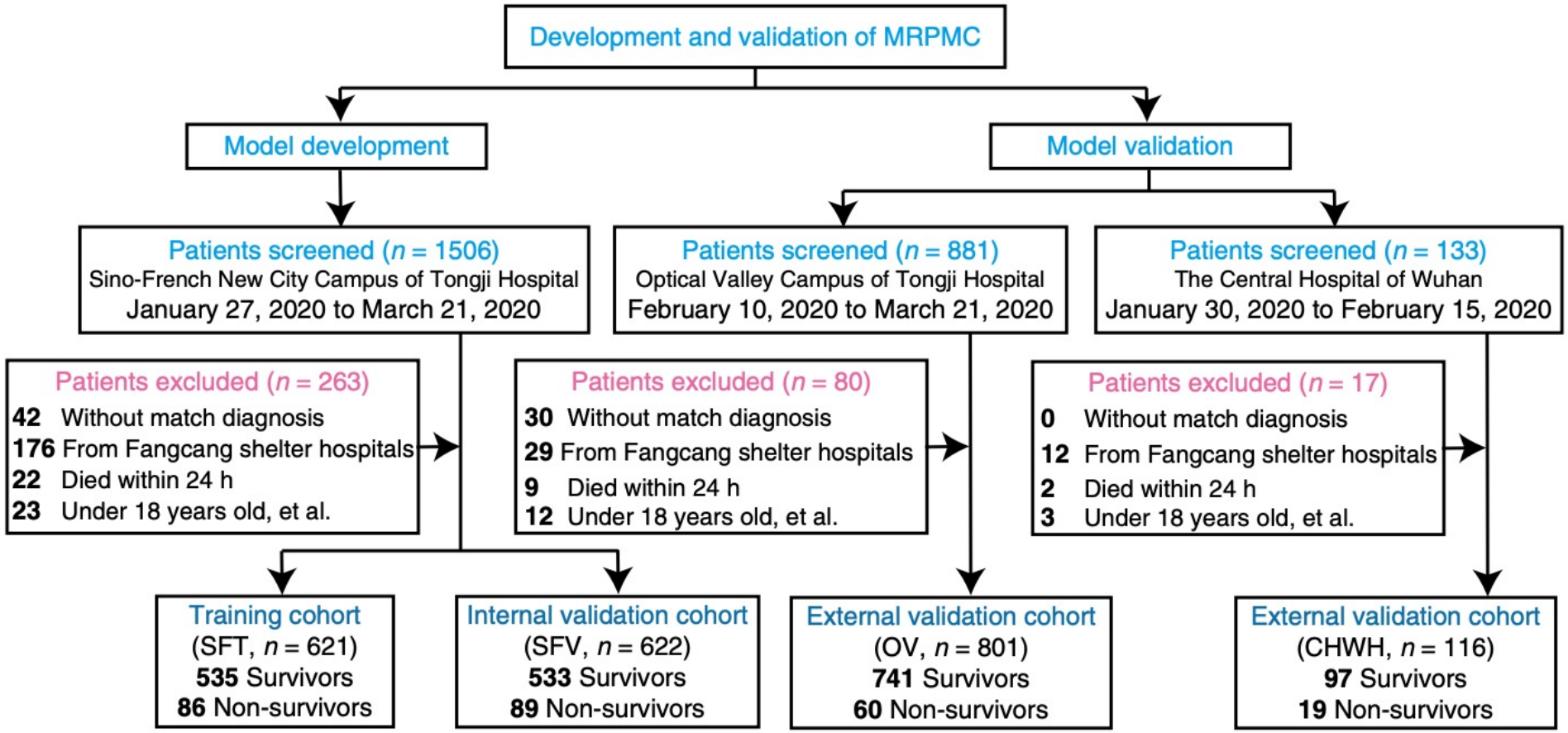


Fig. 1 Study design. MRPMC mortality risk prediction model for COVID-19, SFT training cohort of Sino-French New City Campus of Tongji Hospital, SFV internal validation cohort of Sino-French New City Campus of Tongji Hospital, OV Optical Valley Campus of Tongji Hospital, CHWH The Central Hospital of Wuhan.

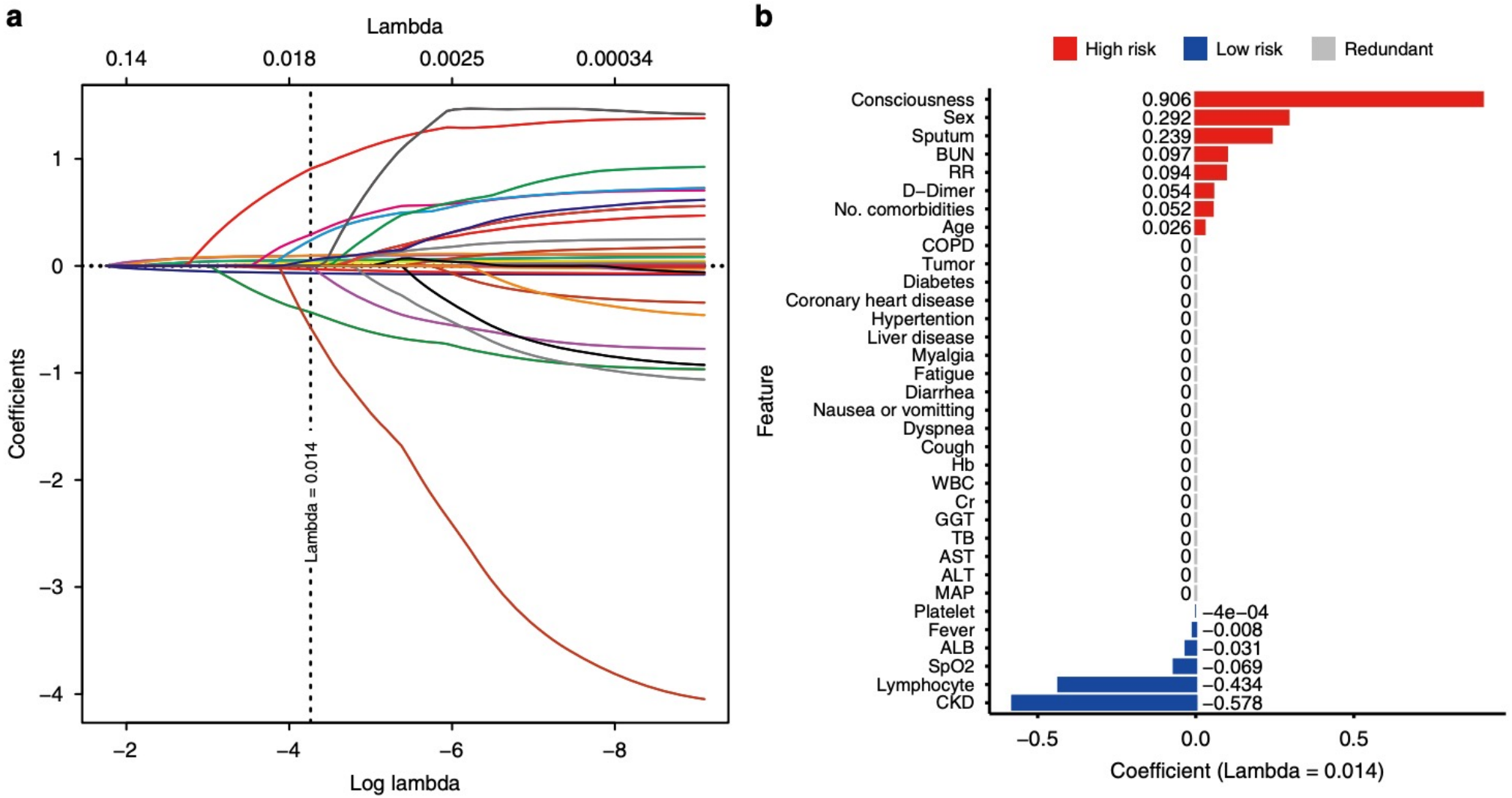


Fig. 2 Feature selection by LASSO. a LASSO variable trace profiles of the 34 features whose intracohort missing rates were less than 5%. The vertical dashed line shows the best lambda value 0.014 chosen by tenfold cross validation. **b** Feature coefficient of LASSO with best lambda value 0.014. High-risk (positive coefficient) and low-risk (negative coefficient) features are colored in red and blue, respectively. Gray features with coefficient 0 were considered redundant and removed, resulting in 14 features left for downstream prognosis modeling. LASSO least absolute shrinkage and selection operator, BUN blood urea nitrogen, RR respiratory rate, COPD chronic obstructive pulmonary disease, Hb hemoglobin, WB, white blood cell count, Cr creatinine, GGT gamma-glutamyl transferase, TB total bilirubin, AST aspartate aminotransferase, ALT alanine transaminase, MAP mean arterial pressure, ALB albumin, SpO_2 oxygen saturation, CKD chronic kidney disease.

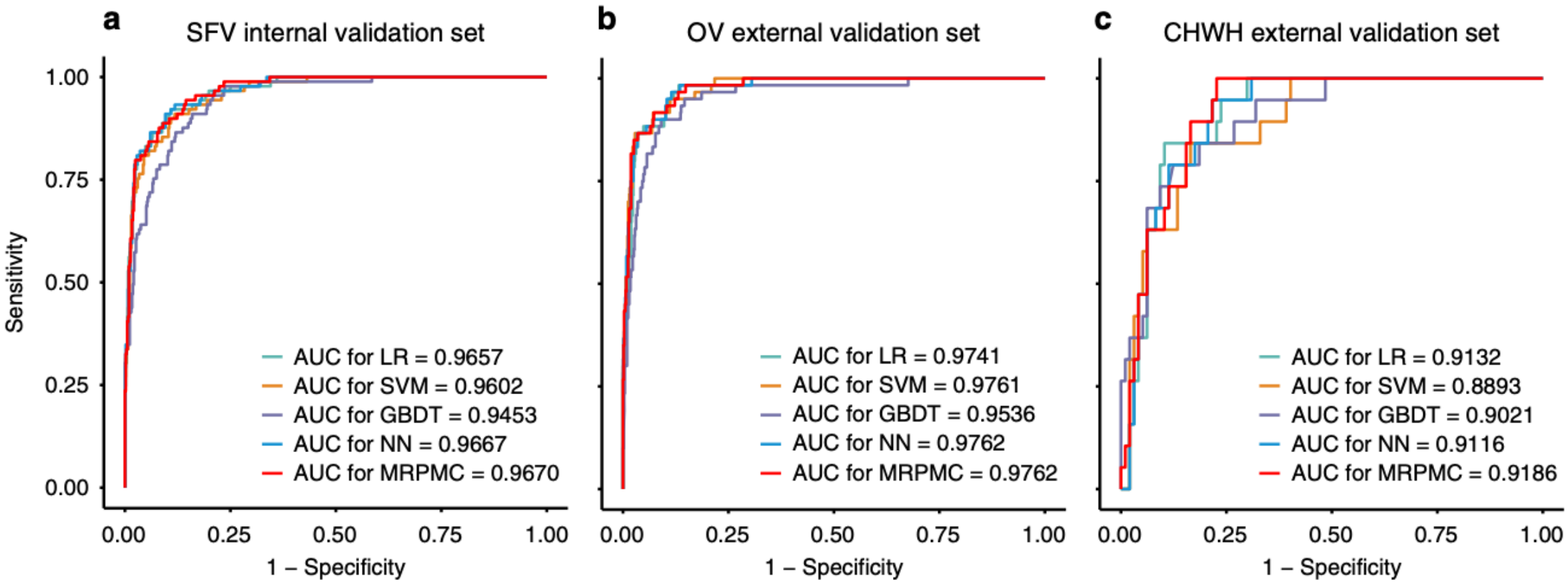

a
SFV internal validation set
Sensitivity
1.00
0.75
0.50
0.25
0.00
AUC for LR = 0.9657
AUC for SVM = 0.9602
AUC for GBDT = 0.9453
AUC for NN = 0.9667
AUC for MRPMC = 0.9670
0.00
0.25
0.50
0.75
1.00
1 – Specificity
b
OV external validation set
AUC for LR = 0.9741
AUC for SVM = 0.9761
AUC for GBDT = 0.9536
AUC for NN = 0.9762
AUC for MRPMC = 0.9762
0.00
0.25
0.50
0.75
1.00
1 – Specificity
c
CHWH external validation set
AUC for LR = 0.9132
AUC for SVM = 0.8893
AUC for GBDT = 0.9021
AUC for NN = 0.9116
AUC for MRPMC = 0.9186
0.00
0.25
0.50
0.75
1.00
1 – Specificity

Table 2 Performance for mortality risk prediction of models in validation cohorts.

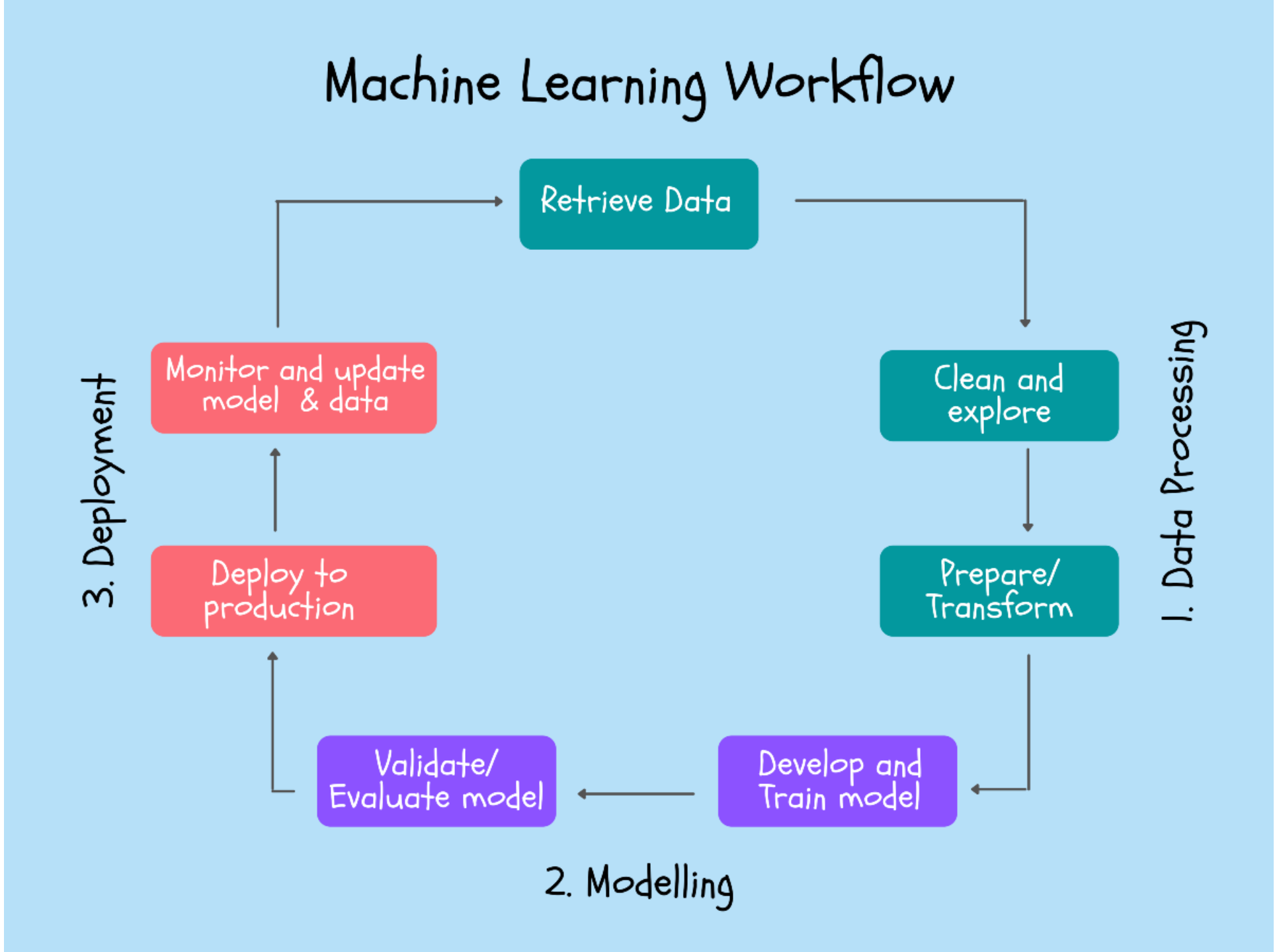
	AUC (95% CI)	Accuracy (95% CI)	Sensitivity (95% CI)	Specificity (95% CI)	PPV (95% CI)	NPV (95% CI)
Internal validation cohort (SFV)						
MRPMC	0.9621 (0.9464-0.9778)	92.4% (90.1-94.4%)	57.3% (46.4-67.7%)	98.3% (96.8-99.2%)	85.0% (73.4-92.9%)	93.2% (90.8-95.2%)
SVM	0.9594 (0.9424-0.9764)	92.4% (90.1-94.4%)	60.7% (49.8-70.9%)	97.8% (96.1-98.8%)	81.8% (70.4-90.2%)	93.7% (91.4-95.6%)
GBDT	0.9454 (0.9246-0.9662)	91.5% (89.0-93.6%)	60.7% (49.8-70.9%)	96.6% (94.7-98.0%)	75.0% (63.4-84.5%)	93.6% (91.3-95.5%)
LR	0.9614 (0.9456-0.9772)	92.1% (89.7-94.1%)	56.2% (45.3-66.7%)	98.1% (96.6-99.1%)	83.3% (71.5-91.7%)	93.1% (90.6-95.0%)
NN	0.9615 (0.9456-0.9774)	92.1% (89.7-94.1%)	51.7% (40.8-62.4%)	98.9% (97.6-99.6%)	88.5% (76.6-95.7%)	92.5% (90.0-94.5%)
External validation cohort (OV)						
MRPMC	0.9760 (0.9613-0.9906)	95.5% (93.8-96.8%)	45.0% (32.1-58.4%)	99.6% (98.8-99.9%)	90.0% (73.5-97.9%)	95.7% (94.0-97.0%)
SVM	0.9774 (0.9640-0.9908)	95.8% (94.1-97.0%)	50.0% (36.8-63.2%)	99.5% (98.6-99.9%)	88.2% (72.6-96.7%)	96.1% (94.5-97.4%)
GBDT	0.9536 (0.9279-0.9793)	94.8% (93.0-96.2%)	48.3% (35.2-61.6%)	98.5% (97.4-99.3%)	72.5% (56.1-85.4%)	95.9% (94.3-97.2%)
LR	0.9721 (0.9568-0.9875)	95.4% (93.7-96.7%)	45.0% (32.1-58.4%)	99.5% (98.6-99.9%)	87.1% (70.2-96.4%)	95.7% (94.0-97.0%)
NN	0.9754 (0.9602-0.9906)	95.6% (94.0-96.9%)	46.7% (33.7-60.0%)	99.6% (98.8-99.9%)	90.3% (74.3-98.0%)	95.8% (94.2-97.1%)
External validation cohort (CHWH)						
MRPMC	0.9246 (0.8763-0.9729)	87.9% (80.6-93.2%)	42.1% (20.3-66.5%)	96.9% (91.2-99.4%)	72.7% (39.0-94.0%)	89.5% (82.0-94.7%)
SVM	0.9067 (0.8482-0.9652)	88.8% (81.6-93.9%)	57.9% (33.5-79.8%)	94.6% (88.4-98.3%)	68.8% (41.3-89.0%)	92.0% (84.8-96.5%)
GBDT	0.9021 (0.8347-0.9694)	87.9% (80.6-93.2%)	31.6% (12.6-56.6%)	99.0% (94.4-100.0%)	85.7% (42.1-99.6%)	88.1% (80.5-93.5%)
LR	0.9213 (0.8710-0.9717)	87.1% (79.6-92.6%)	36.8% (16.3-61.6%)	96.9% (91.2-99.4%)	70.0% (34.8-93.3%)	88.7% (81.1-94.0%)
NN	0.9202 (0.8700-0.9705)	88.8% (81.6-93.9%)	47.4% (24.5-71.1%)	96.9% (91.2-99.4%)	75.0% (42.8-94.5%)	90.4% (83.0-95.3%)

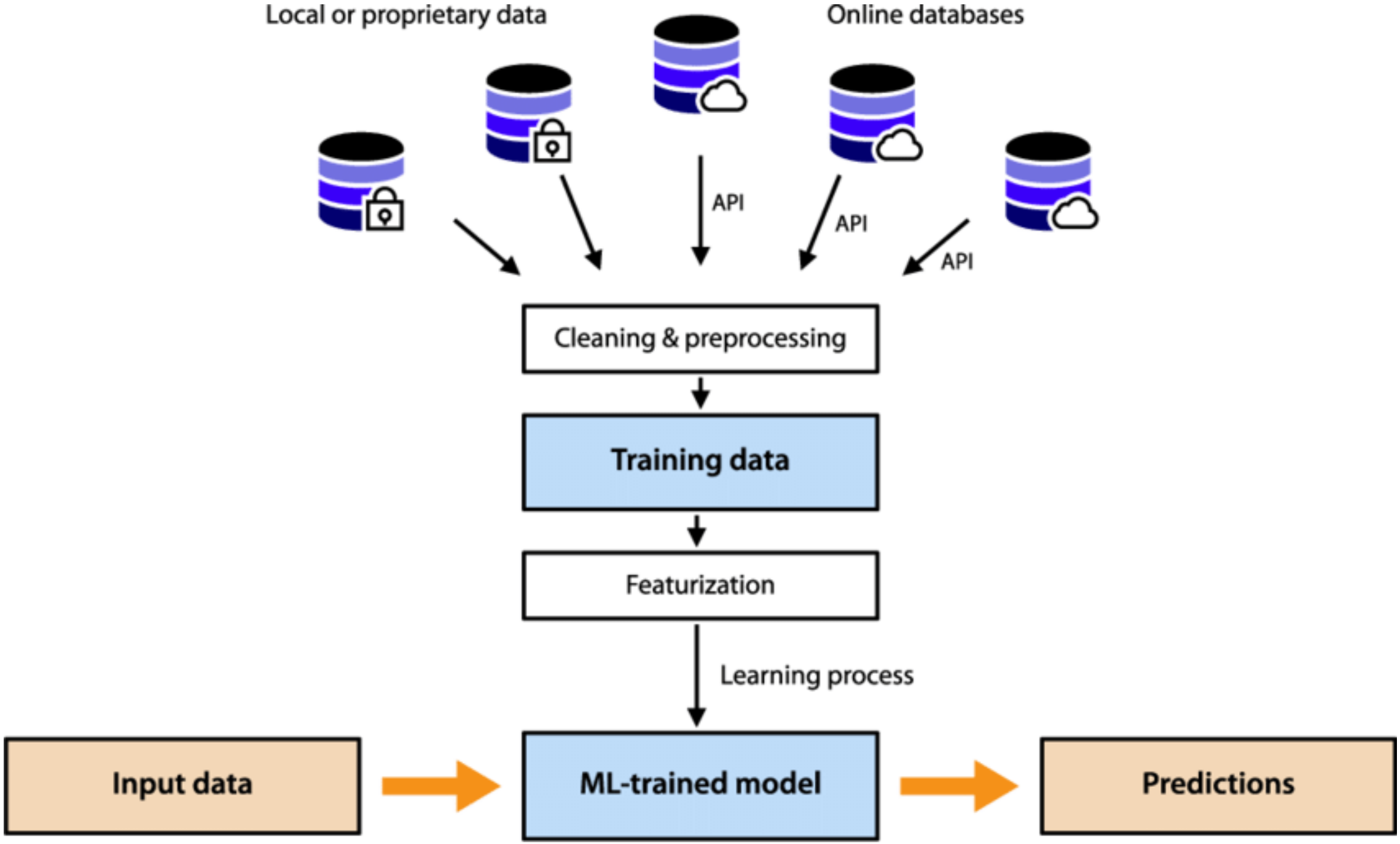
SFV internal validation cohort of Sino-French New City Campus of Tongji Hospital, *OV* Optical Valley Campus of Tongji Hospital, *CHWH* The Central Hospital of Wuhan, *MRPMC* mortality risk prediction model for C
boosted decision tree, *LR* logistic regression, *NN* neural network, *AUC* area under the receiver operating characteristics curve, *PPV* positive predictive value, *NPV* negative predictive value, *95% CI* 95% confidence int

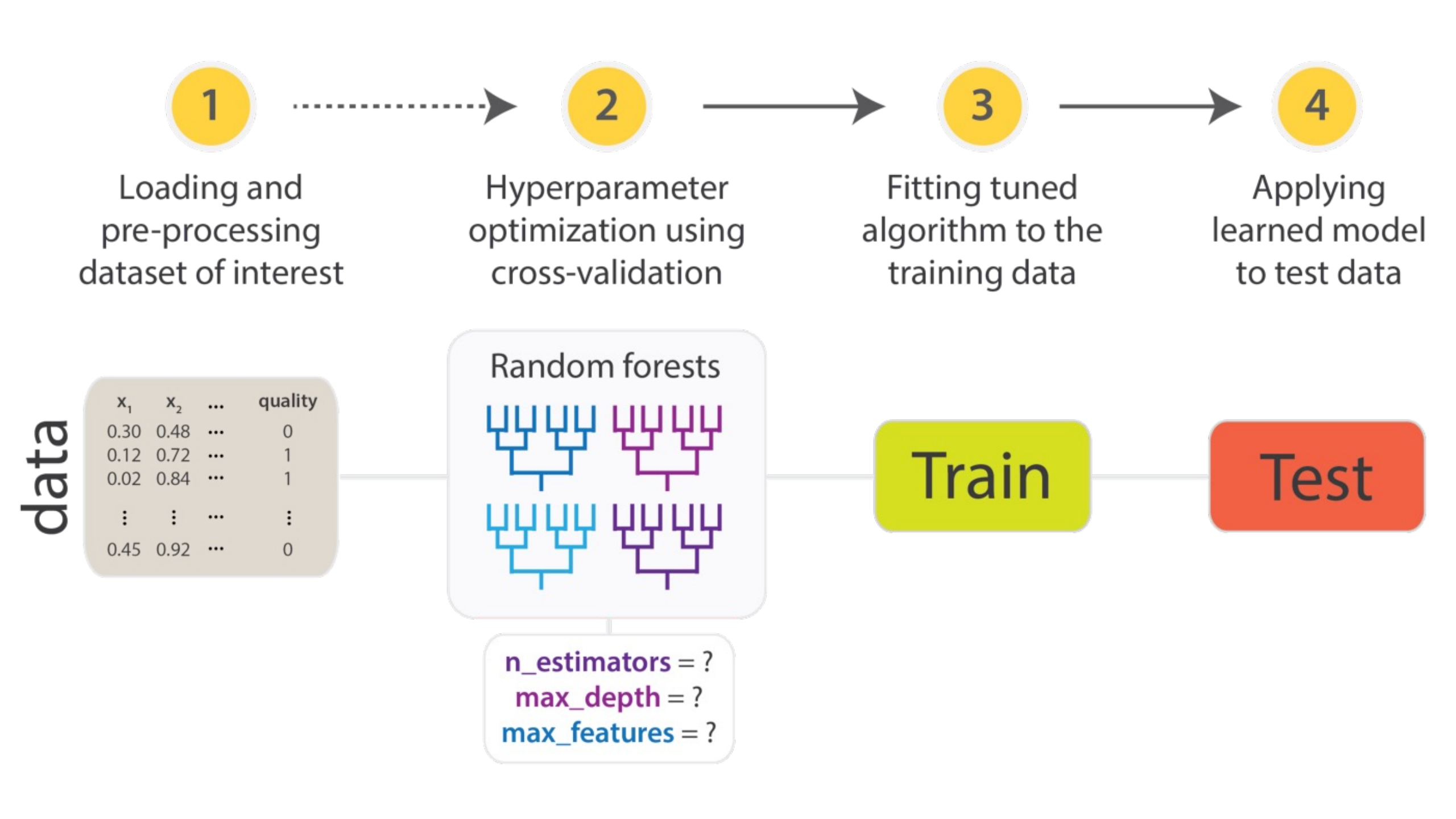
What to know for this exam

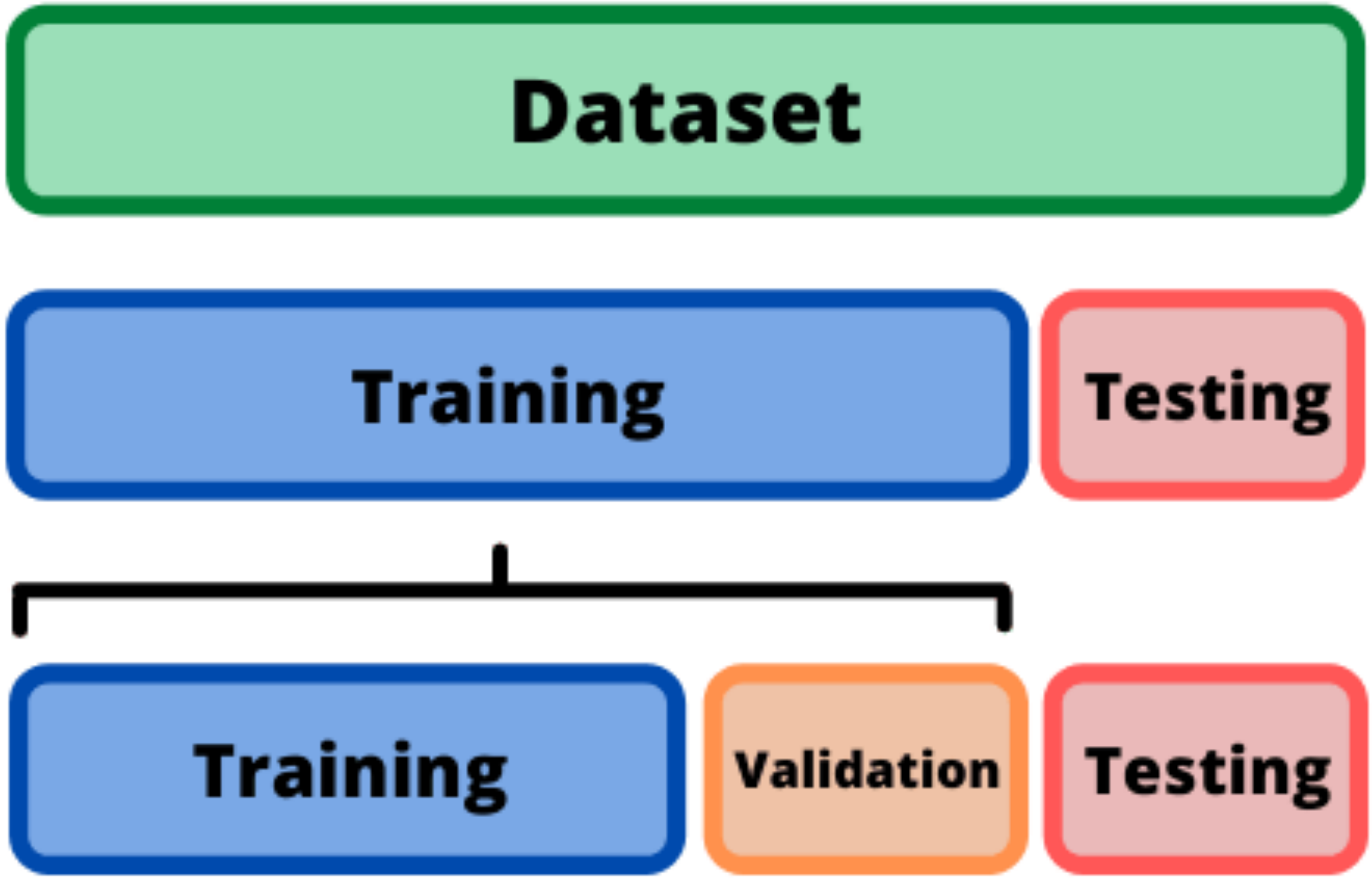
- The ML process, including evaluation of ML models
- High level of how the fundamental ML models work
- Comparison of fundamental ML models
- How (parametric) models learn: gradient descent

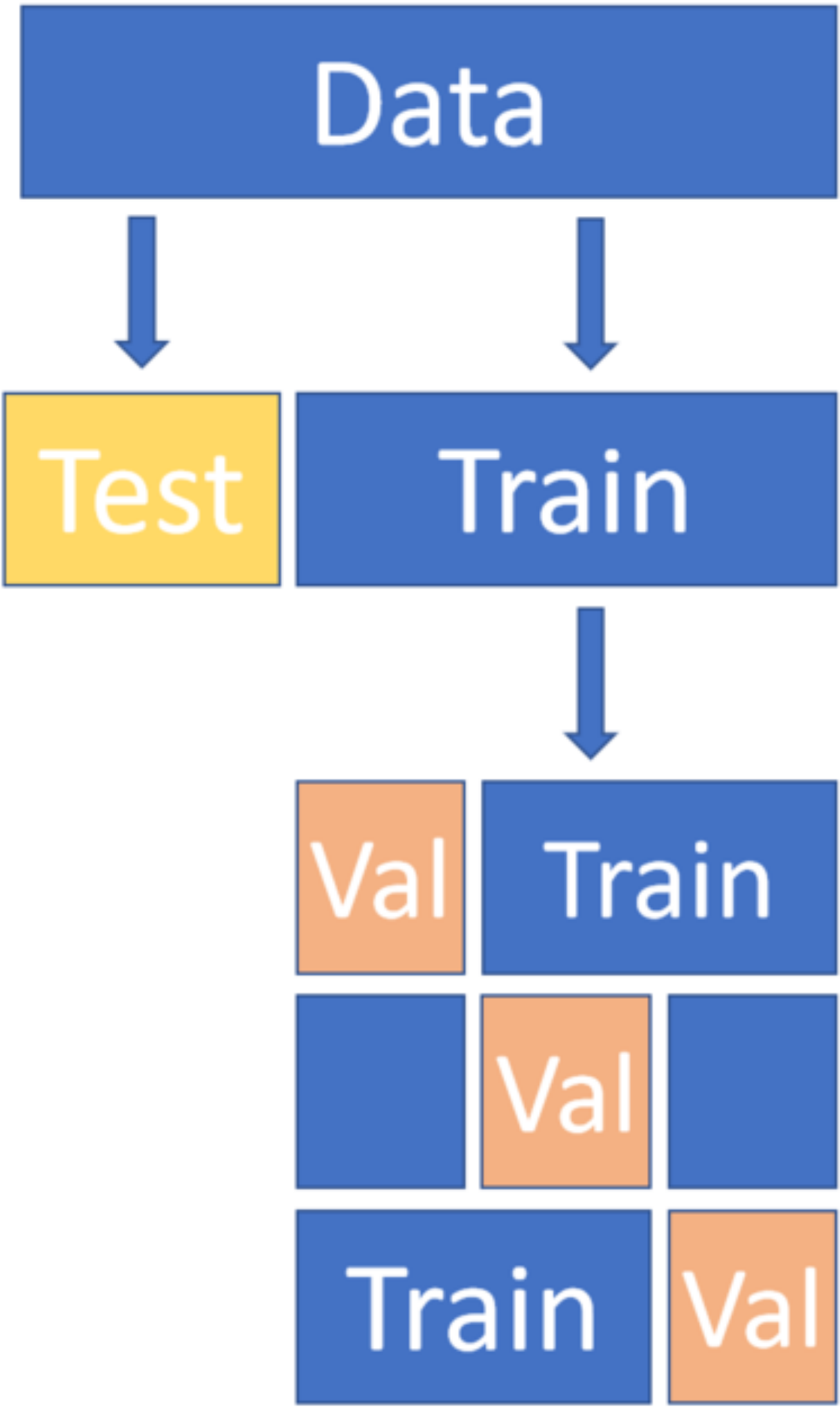
The ML Process


Machine Learning Workflow
Retrieve Data
Clean and explore
Prepare/ Transform
Develop and Train model
Validate/ Evaluate model
Deploy to production
Monitor and update model & data
1. Data Processing
2. Modelling
3. Deployment

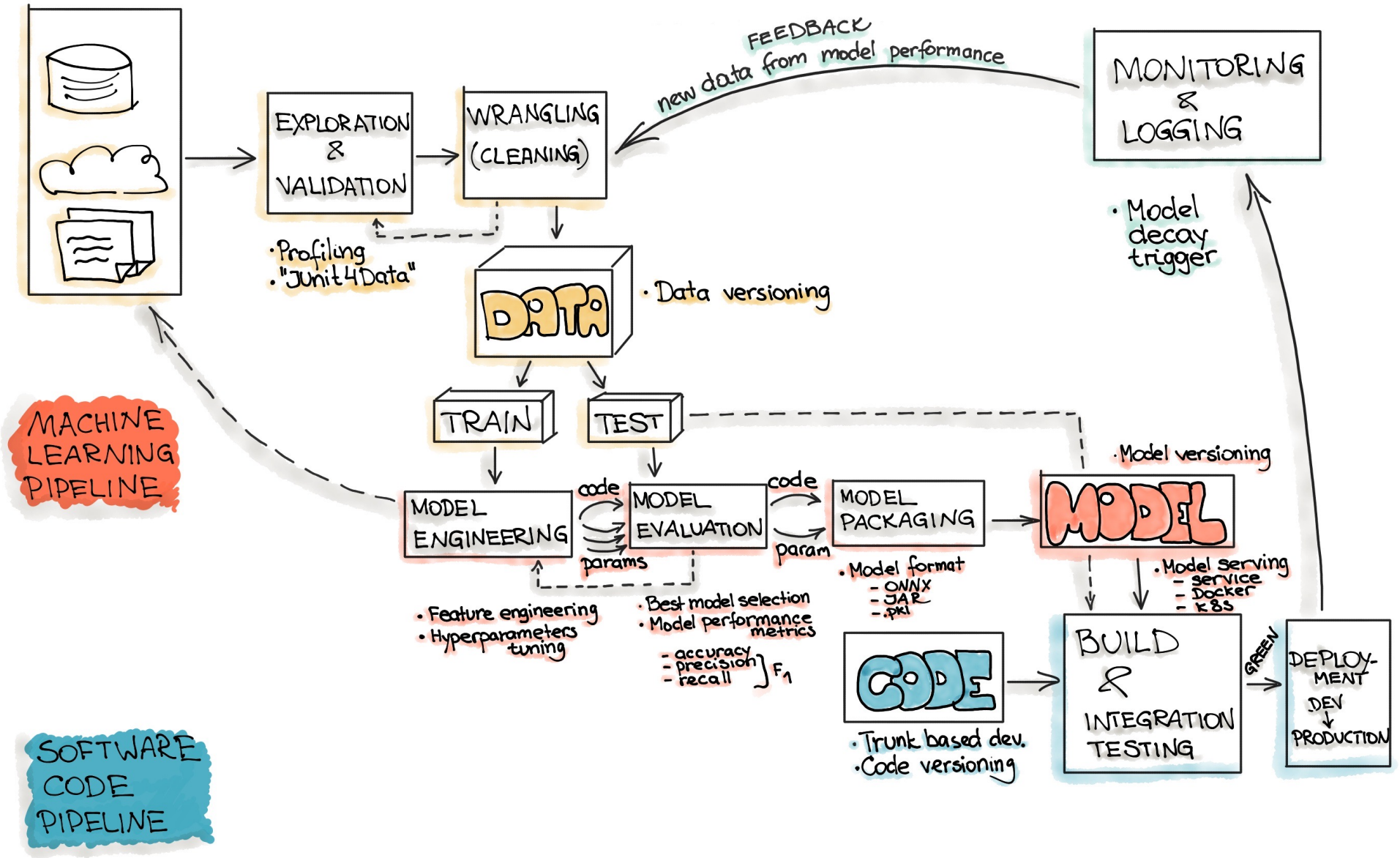

Local or proprietary data
Online databases
API
API
API
Cleaning & preprocessing
Training data
Featurization
Learning process
Input data
ML-trained model
Predictions


1
2
3
4
Loading and pre-processing dataset of interest
Hyperparameter optimization using cross-validation
Fitting tuned algorithm to the training data
Applying learned model to test data
data
x1 x2 ... quality
0.30 0.48 ... 0
0.12 0.72 ... 1
0.02 0.84 ... 1
0.45 0.92 ... 0
Random forests
n_estimators = ?
max_depth = ?
max_features = ?
Train
Test


Dataset
Training
Testing
Training
Validation
Testing


Data
Test
Train
Val
Train
Val
Train
Val

MACHINE LEARNING ENGINEERING



Comparison of Fundamental ML Models

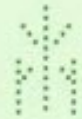
Most/all applications could use any of these models. Which one works better depends on the dataset.

Classifier	Medical record source	# features	N_{ASD}	$N_{non\text{-}ASD}$	Mean age (SD)	% Male (*N*)	Test sensitivity	Test specificity	Test accuracy
ADTree8 [30]	ADOS Module 1	8	Train: 612 Test [30]: 446 Test [25]: 2,333 Test [32]: 931	Train:15 Test [30]: 0 Test [25]: 238 Test [32]: 102	6.16 (4.16)	76.8% (*N* = 2,009)	100%	100%	100%
ADTree7 [29]	ADI-R	7	Train: 891 Test [24]: 222 Test [32]: 462	Train: 75 Test [24]: 0 Test [32]: 218	8.5 (3.3)	65% (*N* = 628)	100%	1.13%	99.9%
SVM with L1 norm (SVM5) [27]	ADOS Module 2	5	Train/test: 1,319	Train/test: 70	6.92 (2.83)	80% (*N* = 1,101)	98%	58%	98%
LR with L2 norm (LR5) [27]	ADOS Module 2	5	Train/test: 1,319	Train/test: 70	6.92 (2.83)	80% (*N* = 1,101)	93%	67%	95%
LR with L1 norm (LR9) [26]	ADOS Module 2	9	Train: 362 Test: 1,089	Train: 282 Test: 66	11.75 (10)	76.4% (*N* = 1,375)	98.81%	89.39%	98.27%
Radial kernel SVM (SVM12) [26]	ADOS Module 3	12	Train: 510 Test: 1,924	Train: 93 Test: 214	16.25 (11.58)	76.4% (*N* = 2,094)	97.71%	97.2%	97.66%
Linear SVM (SVM10) [27]	ADOS Module 3	10	Train/test: 2,870	Train/test: 273	9.08 (3.08)	81% (*N* = 2,557)	95%	87%	97%
LR (LR10) [27]	ADOS Module 3	10	Train/test: 2,870	Train/test: 273	9.08 (3.08)	81% (*N* = 2,557)	90%	89%	94%

Abbreviations: ADI-R, Autism Diagnostic Interview-Revised; ADOS, Autism Diagnostic Observation Schedule; ADTree7, 7-feature alternating decision tree; ADTree8, 8-feature alternating decision tree; LR, logistic regression; LR5, 5-feature LR classifier; LR10, 10-feature LR classifier; SVM, support vector machine; SVM5, 5-feature SVM; SVM10, 10-feature SVM; SVM12, 12-feature SVM.

https://doi.org/10.1371/journal.pmed.1002705.t001

Algorithms	Advantages	Disadvantages
The *k-means* method [22,23]	• Relatively efficient • Can process large data sets.	• Often terminates at a local optimum. • Applicable only when mean is defined. • Not applicable for categorical data. • Unable to handle noisy data. • Not suitable to discover clusters with non-convex shapes.
k-nearest neighbor (*k*-NN) classifier [7,15]	• Nonparametric • Zero cost in the learning process • Classifying any data whenever finding similarity measures of any given instances • Intuitive approach • Robust to outliers on the predictors	• Expensive computation for a large dataset • Hard to interpret the result • The performance relies on the number of dimensions • Lack of explicit model training • Susceptible to correlated inputs and irrelevant features • Very difficult in handling data of mixed types.
Support vector machine (SVM) [5,15,22]	• Can utilize predictive power of linear combinations of inputs • Good prediction in a variety of situations • Low generalization error • Easy to interpret results	• Weak in natural handling of mixed data types and computational scalability • Very black box • Sensitive to tuning parameters and kernel choice • Training an SVM on a large data set can be slow • Testing data should be near the training data
Decision Trees [7,15]	• Some tolerance to correlated inputs. • A single tree is highly interpretable, • Can handle missing values. • Able to handle both numerical and categorical data. • Performs well with large datasets.	• Cannot work on (linear) combinations of features. • Relatively less predictive in many situations. • Practical decision-tree learning algorithms cannot guarantee to return the globally-optimal decision tree. • Decision-tree can lead to overfitting.
Logistic regression [7]	• Provides model logistic probability • Easy to interpret • Provides confidence interval • Quickly update the classification model to incorporate new data	• Does not handle the missing value of continuous variables • Suffers multicollinearity • Sensitive to extreme values of continuous variables
Naïve Bayes [5,7]	• Suitable for relative small training set • Can easily obtain the probability for a prediction • Relatively simple and straightforward to use • Can deal with some noisy and missing data • Can handles multiple classes	• Prone to bias when increasing the number of training sets • Assumes all features are independent and equally important, which is unlikely in real-world cases. • Sensitive to how the input data is prepared.
Neural networks [15]	• Good prediction generally • Some tolerance to correlated inputs • Incorporating the predictive power of different combinations of inputs	• Not robust to outliers • Susceptible to irrelevant features • Difficult in dealing with big data with complex model

TYPE	NAME	DESCRIPTION	ADVANTAGES	DISADVANTAGES
Linear	 Linear regression	**The "best fit" line** through all data points. Predictions are numerical.	Easy to understand -- you clearly see what the biggest drivers of the model are.	X **Sometimes too simple to capture complex relationships between variables.** X **Tendency for the model to "overfit".**
	 Logistic regression	The adaptation of **linear regression** to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	X **Sometimes too simple to capture complex relationships between variables.** X **Tendency for the model to "overfit".**
Tree-based	 Decision tree	A graph that uses a **branching method** to match all possible outcomes of a decision.	Easy to understand and implement.	X **Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.**
	 Random Forest	Takes the average of many decision trees, each of which is made with a sample of the data. Each tree is weaker than a full decision tree, but **by combining them we get better overall performance.**	A sort of "wisdom of the crowd". Tends to result in very high quality models. Fast to train.	X **Can be slow to output predictions relative to other algorithms.** X **Not easy to understand predictions.**
	 Gradient Boosting	Uses even weaker decision trees, that are increasingly **focused on "hard" examples.**	High-performing.	X **A small change in the feature set or training set can create radical changes in the model.** X **Not easy to understand predictions.**
Neural networks	 Neural networks	Mimics the behavior of the brain. Neural networks are interconnected neurons that pass messages to each other. Deep learning uses several **layers of neural networks put one after the other.**	Can handle extremely complex tasks - no other algorithm comes close in image recognition.	X **Very, very slow to train, because they have so many layers. Require a lot of power.** X **Almost impossible to understand predictions.**

Good way to study: look at each hyperparameter in scikit-learn for the models we have learned in class. Do you understand what each hyperparameter means? Do you understand the tradeoffs between the different values of the hyperparameters?

sklearn.tree.DecisionTreeClassifier

class `sklearn.tree.`**`DecisionTreeClassifier`**(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0*) ¶ [source]

A decision tree classifier.

Read more in the User Guide.

Parameters:

criterion : *{"gini", "entropy", "log_loss"}, default="gini"*
The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation.

splitter : *{"best", "random"}, default="best"*
The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : *int, default=None*
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split : *int or float, default=2*
The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

Changed in version 0.18: Added float values for fractions.

min_samples_leaf : *int or float, default=1*

The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least `min_samples_leaf` training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

- If int, then consider `min_samples_leaf` as the minimum number.
- If float, then `min_samples_leaf` is a fraction and `ceil(min_samples_leaf * n_samples)` are the minimum number of samples for each node.

Changed in version 0.18: Added float values for fractions.

min_weight_fraction_leaf : *float, default=0.0*

The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample_weight is not provided.

max_features : *int, float or {"auto", "sqrt", "log2"}, default=None*

The number of features to consider when looking for the best split:

- If int, then consider `max_features` features at each split.
- If float, then `max_features` is a fraction and `max(1, int(max_features * n_features_in_))` features are considered at each split.
- If "auto", then `max_features=sqrt(n_features)`.
- If "sqrt", then `max_features=sqrt(n_features)`.
- If "log2", then `max_features=log2(n_features)`.
- If None, then `max_features=n_features`.

Deprecated since version 1.1: The `"auto"` option was deprecated in 1.1 and will be removed in 1.3.

Note: the search for a split does not stop until at least one valid partition of the node samples is found, even if it requires to effectively inspect more than `max_features` features.

max_leaf_nodes : *int, default=None*

Grow a tree with `max_leaf_nodes` in best-first fashion. Best nodes are defined as relative reduction in impurity. If None then unlimited number of leaf nodes.

min_impurity_decrease : *float, default=0.0*

A node will be split if this split induces a decrease of the impurity greater than or equal to this value.

The weighted impurity decrease equation is the following:

```
N_t / N * (impurity - N_t_R / N_t * right_impurity
                    - N_t_L / N_t * left_impurity)
```

where `N` is the total number of samples, `N_t` is the number of samples at the current node, `N_t_L` is the number of samples in the left child, and `N_t_R` is the number of samples in the right child.

`N`, `N_t`, `N_t_R` and `N_t_L` all refer to the weighted sum, if `sample_weight` is passed.

New in version 0.19.

class_weight : *dict, list of dict or "balanced", default=None*

Weights associated with classes in the form `{class_label: weight}`. If None, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of y.

Note that for multioutput (including multilabel) weights should be defined for each class of every column in its own dict. For example, for four-class multilabel classification weights should be [{0: 1, 1: 1}, {0: 1, 1: 5}, {0: 1, 1: 1}, {0: 1, 1: 1}] instead of [{1:1}, {2:5}, {3:1}, {4:1}].

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`

For multi-output, the weights of each column of y will be multiplied.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

sklearn.linear_model.LogisticRegression

class `sklearn.linear_model.`**`LogisticRegression`**(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)

This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. **Note that regularization is applied by default**. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Read more in the User Guide.

Parameters:

penalty : *{'l1', 'l2', 'elasticnet', None}, default='l2'*

Specify the norm of the penalty:

- `None`: no penalty is added;
- `'l2'`: add a L2 penalty term and it is the default choice;
- `'l1'`: add a L1 penalty term;
- `'elasticnet'`: both L1 and L2 penalty terms are added.

> **Warning:** Some penalties may not work with some solvers. See the parameter `solver` below, to know the compatibility between the penalty and solver.

New in version 0.19: l1 penalty with SAGA solver (allowing 'multinomial' + L1)

dual : *bool, default=False*

Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Prefer dual=False when n_samples > n_features.

tol : *float, default=1e-4*

Tolerance for stopping criteria.

C : *float, default=1.0*

Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.

fit_intercept : *bool, default=True*

Specifies if a constant (a.k.a. bias or intercept) should be added to the decision function.

intercept_scaling : *float, default=1*

Useful only when the solver 'liblinear' is used and self.fit_intercept is set to True. In this case, x becomes [x, self.intercept_scaling], i.e. a "synthetic" feature with constant value equal to intercept_scaling is appended to the instance vector. The intercept becomes `intercept_scaling * synthetic_feature_weight`.

Note! the synthetic feature weight is subject to l1/l2 regularization as all other features. To lessen the effect of regularization on synthetic feature weight (and therefore on the intercept) intercept_scaling has to be increased.

class_weight : *dict or 'balanced', default=None*

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one.

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

New in version 0.17: class_weight='balanced'

max_iter : *int, default=100*

Maximum number of iterations taken for the solvers to converge.

multi_class : *{'auto', 'ovr', 'multinomial'}, default='auto'*

If the option chosen is 'ovr', then a binary problem is fit for each label. For 'multinomial' the loss minimised is the multinomial loss fit across the entire probability distribution, *even when the data is binary*. 'multinomial' is unavailable when solver='liblinear'. 'auto' selects 'ovr' if the data is binary, or if solver='liblinear', and otherwise selects 'multinomial'.

New in version 0.18: Stochastic Average Gradient descent solver for 'multinomial' case.

Changed in version 0.22: Default changed from 'ovr' to 'auto' in 0.22.

verbose : *int, default=0*

For the liblinear and lbfgs solvers set verbose to any positive number for verbosity.

warm_start : *bool, default=False*

When set to True, reuse the solution of the previous call to fit as initialization, otherwise, just erase the previous solution. Useless for liblinear solver. See the Glossary.

New in version 0.17: warm_start to support *lbfgs*, *newton-cg*, *sag*, *saga* solvers.

n_jobs : *int, default=None*

Number of CPU cores used when parallelizing over classes if multi_class='ovr'". This parameter is ignored when the `solver` is set to 'liblinear' regardless of whether 'multi_class' is specified or not. `None` means 1 unless in a `joblib.parallel_backend` context. `-1` means using all processors. See Glossary for more details.

l1_ratio : *float, default=None*

The Elastic-Net mixing parameter, with `0 <= l1_ratio <= 1`. Only used if `penalty='elasticnet'`. Setting `l1_ratio=0` is equivalent to using `penalty='l2'`, while setting `l1_ratio=1` is equivalent to using `penalty='l1'`. For `0 < l1_ratio <1`, the penalty is a combination of L1 and L2.

sklearn.svm.SVC

class `sklearn.svm.`**SVC**(**, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None*) [source]

C-Support Vector Classification.

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using `LinearSVC` or `SGDClassifier` instead, possibly after a `Nystroem` transformer or other Kernel Approximation.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how `gamma`, `coef0` and `degree` affect each other, see the corresponding section in the narrative documentation: Kernel functions.

Read more in the User Guide.

Parameters:

C : *float, default=1.0*
Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.

kernel : *{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'} or callable, default='rbf'*
Specifies the kernel type to be used in the algorithm. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape `(n_samples, n_samples)`.

degree : *int, default=3*
Degree of the polynomial kernel function ('poly'). Must be non-negative. Ignored by all other kernels.

gamma : *{'scale', 'auto'} or float, default='scale'*
Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

- if `gamma='scale'` (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma,
- if 'auto', uses 1 / n_features
- if float, must be non-negative.

Changed in version 0.22: The default value of `gamma` changed from 'auto' to 'scale'.

coef0 : *float, default=0.0*

Independent term in kernel function. It is only significant in 'poly' and 'sigmoid'.

shrinking : *bool, default=True*

Whether to use the shrinking heuristic. See the User Guide.

probability : *bool, default=False*

Whether to enable probability estimates. This must be enabled prior to calling `fit`, will slow down that method as it internally uses 5-fold cross-validation, and `predict_proba` may be inconsistent with `predict`. Read more in the User Guide.

tol : *float, default=1e-3*

Tolerance for stopping criterion.

cache_size : *float, default=200*

Specify the size of the kernel cache (in MB).

class_weight : *dict or 'balanced', default=None*

Set the parameter C of class i to class_weight[i]*C for SVC. If not given, all classes are supposed to have weight one. The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

max_iter : *int, default=-1*

Hard limit on iterations within solver, or -1 for no limit.

decision_function_shape : *{'ovo', 'ovr'}, default='ovr'*

Whether to return a one-vs-rest ('ovr') decision function of shape (n_samples, n_classes) as all other classifiers, or the original one-vs-one ('ovo') decision function of libsvm which has shape (n_samples, n_classes * (n_classes - 1) / 2). However, note that internally, one-vs-one ('ovo') is always used as a multi-class strategy to train models; an ovr matrix is only constructed from the ovo matrix. The parameter is ignored for binary classification.

> *Changed in version 0.19:* decision_function_shape is 'ovr' by default.

New in version 0.17: decision_function_shape='ovr' is recommended.

> *Changed in version 0.17:* Deprecated *decision_function_shape='ovo' and None*.

break_ties : *bool, default=False*

If true, `decision_function_shape='ovr'`, and number of classes > 2, predict will break ties according to the confidence values of decision_function; otherwise the first class among the tied classes is returned. Please note that breaking ties comes at a relatively high computational cost compared to a simple predict.

Do this for all the other models we learned in class as well.

- RandomForestClassifier
- RandomForestRegressor
- LinearRegression
- KNeighborsClassifier
- BernoulliNB
- …

Understand tradeoffs between models as well.

- Types of predictions that can be made (classification vs. regression, binary vs. multi-class, supervised vs. unsupervised, etc).
- Types of decision boundaries that can be learned.
- Learning speed and memory.
- Prediction speed and memory.

Models are Engineered, Not Laws of Nature!

- We want to minimize loss (error), but we can define loss in many ways
- We want models that can separate our data well. There are many ways to separate the outcome (y values) based on the input data (X values).
- In general, the answer to “Could we change this method by ___” is “yes”
  - These are algorithms which can be modified and extended
  - Again, not laws of nature
  - But you should understand how then basic methods work so that you understand what the effect of changing them will be (“know the rules before you break them”)

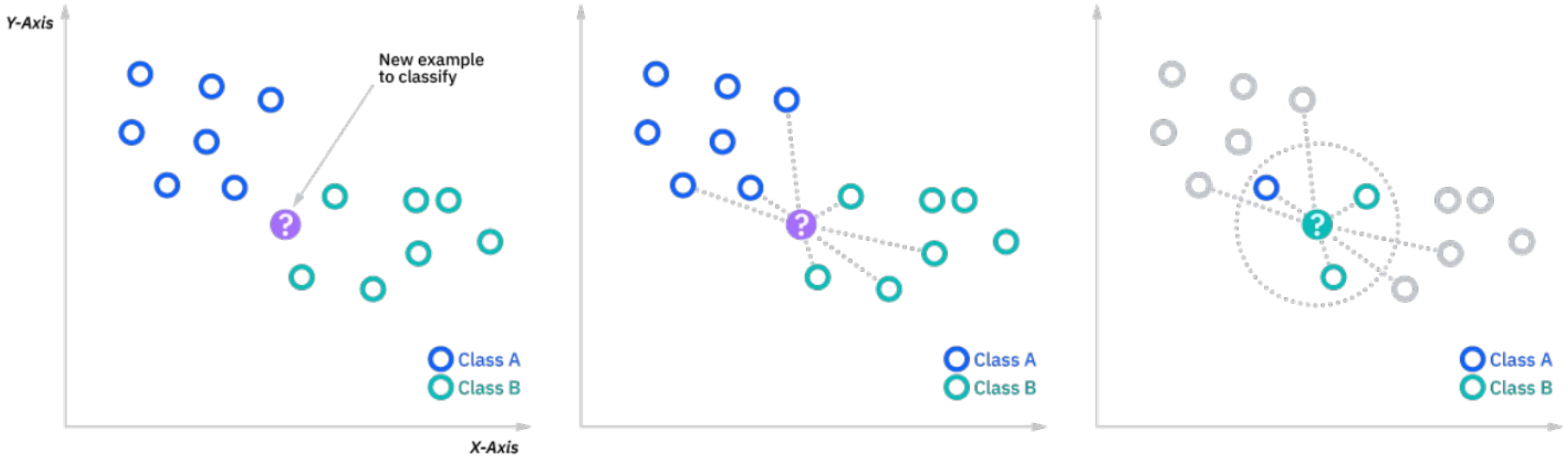
True or False? Training a k-nearest-neighbors classifier takes more computational time than applying it.

True or False? Training a k-nearest-neighbors classifier takes more computational time than applying it.

False.

True or False? Training a k-nearest-neighbors classifier takes more computational time than applying it.

False.



True or False? The more training examples, the more accurate the prediction of a k-nearest-neighbors.

True or False? The more training examples, the more accurate the prediction of a k-nearest-neighbors.

True.

True or False? k-nearest-neighbors cannot be used or modified for regression.

True or False? k-nearest-neighbors cannot be used or modified for regression.

False.

True or False? k-nearest-neighbors is sensitive to outliers.

True or False? k-nearest-neighbors is sensitive to outliers.

True.

Which of these models can separate the following data?

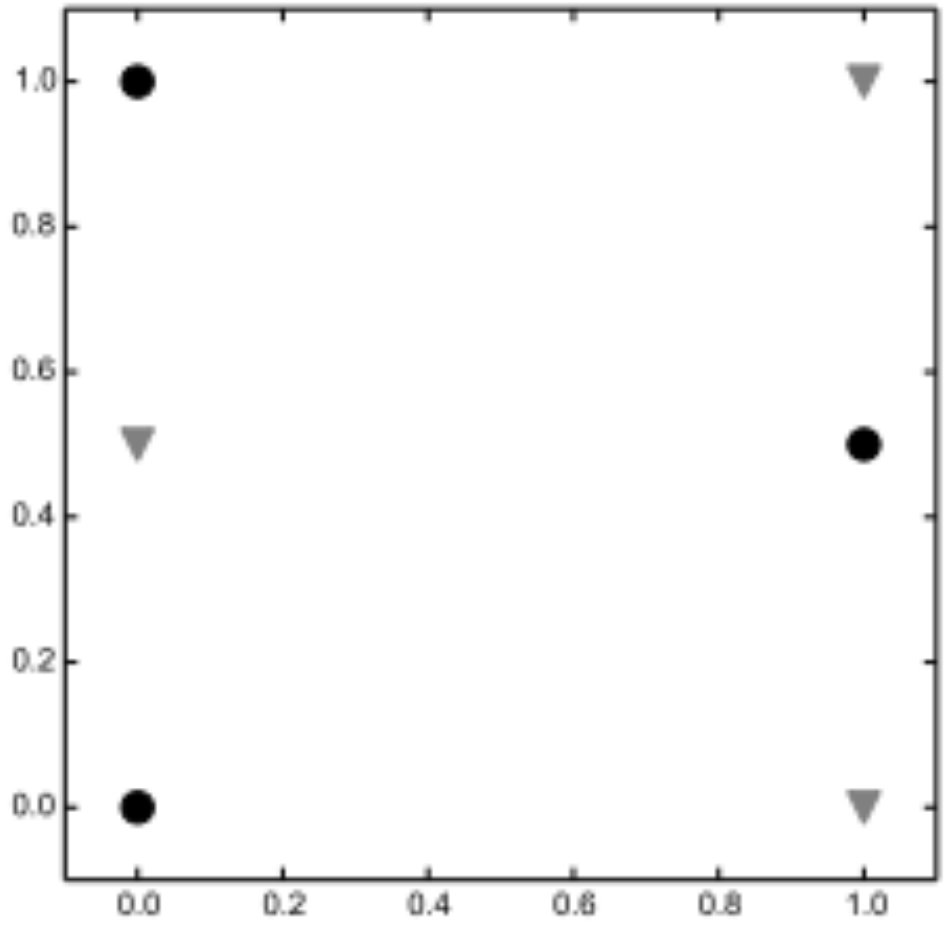
Logistic regression

Hard-margin SVM without increasing the dimensionality (linear kernel)

SVM with polynomial kernel

Decision Tree

3-Nearest Neighbors using Euclidean distance



Which of these models can separate the following data?

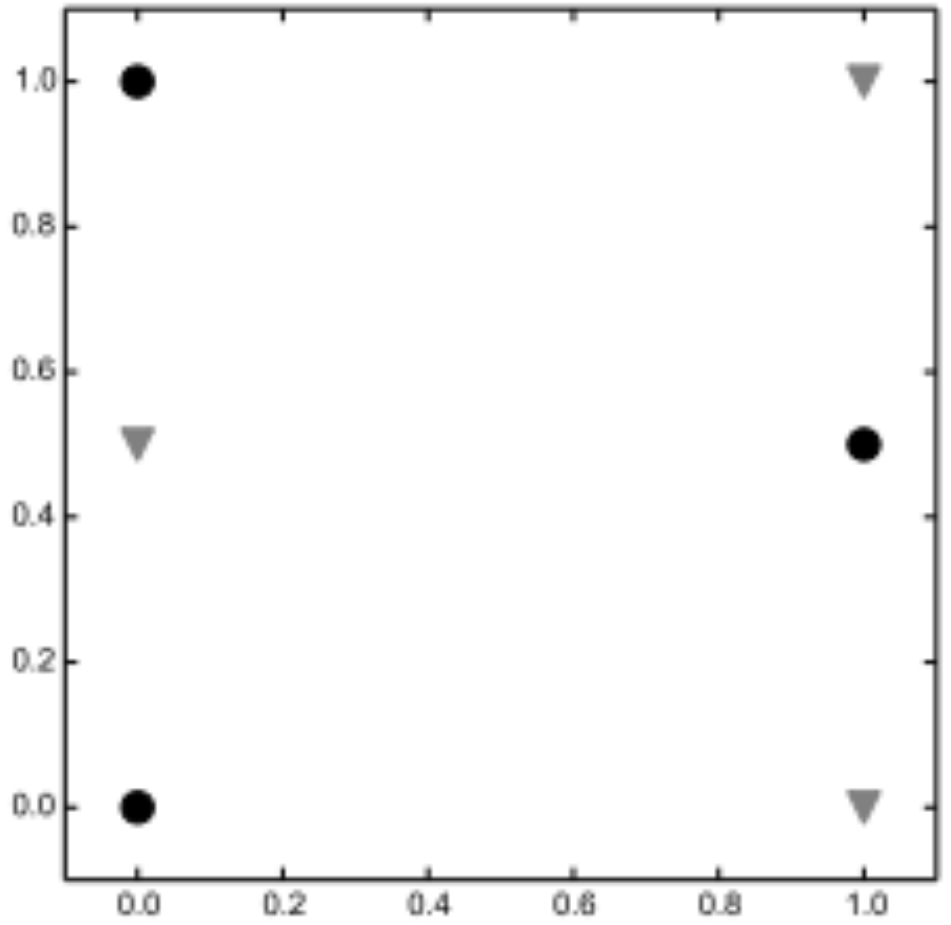
Logistic regression

Hard-margin SVM without increasing the dimensionality (linear kernel)

SVM with polynomial kernel

Decision Tree

3-Nearest Neighbors using Euclidean distance



[2 points] Suppose we have a regularized linear regression model: $argmin_w \|Y - Xw\|_2^2 + \lambda \|w\|_1$. What is the effect of increasing λ on bias and variance?

(a) Increases bias, increases variance

(b) Increases bias, decreases variance

(c) Decreases bias, increases variance

(d) Decreases bias, decreases variance

(e) Not enough information to tell

[2 points] Suppose we have a regularized linear regression model: $argmin_w \|Y - Xw\|_2^2 + \lambda \|w\|_1$. What is the effect of increasing λ on bias and variance?

(a) Increases bias, increases variance

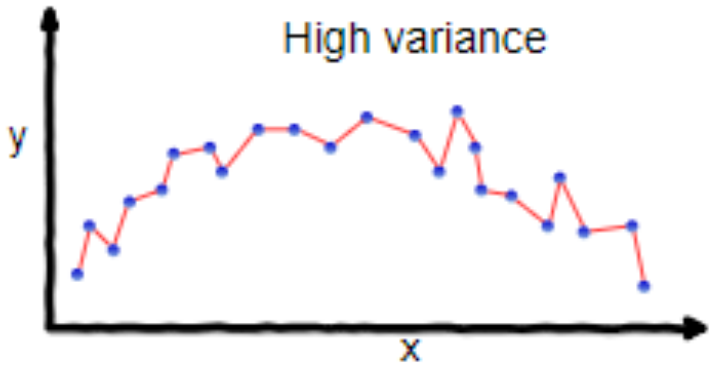
(b) Increases bias, decreases variance

(c) Decreases bias, increases variance

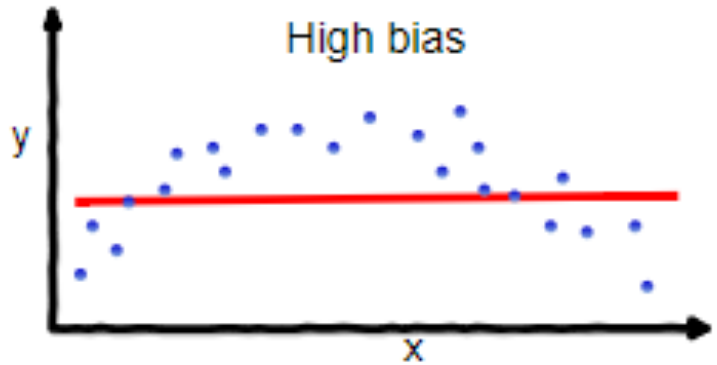
(d) Decreases bias, decreases variance

(e) Not enough information to tell

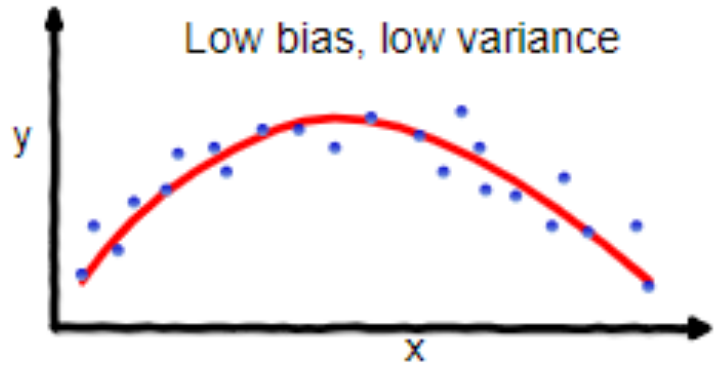
Answer: B



overfitting



underfitting



Good balance

True or False: 5-nearest neighbors is more robust to outliers than 1-nearest neighbors

True or False: 5-nearest neighbors is more robust to outliers than 1-nearest neighbors

True.

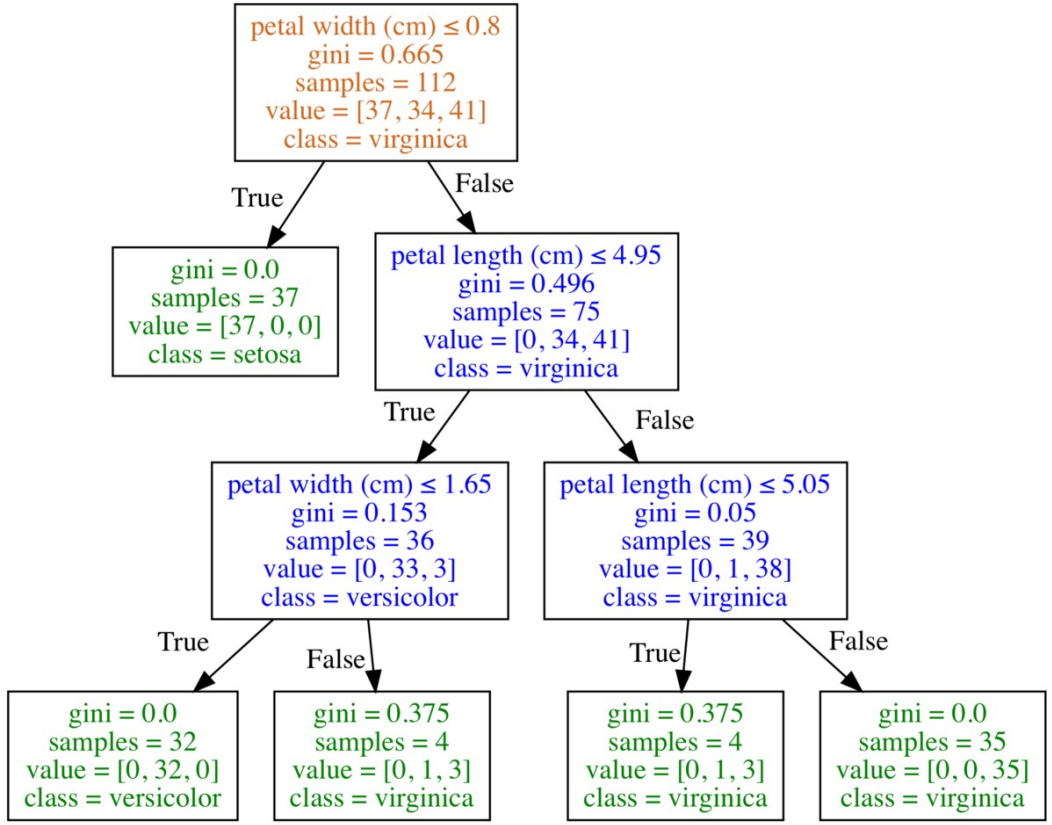
True or False? A tree with depth of 3 has higher variance than a tree with depth of 1.

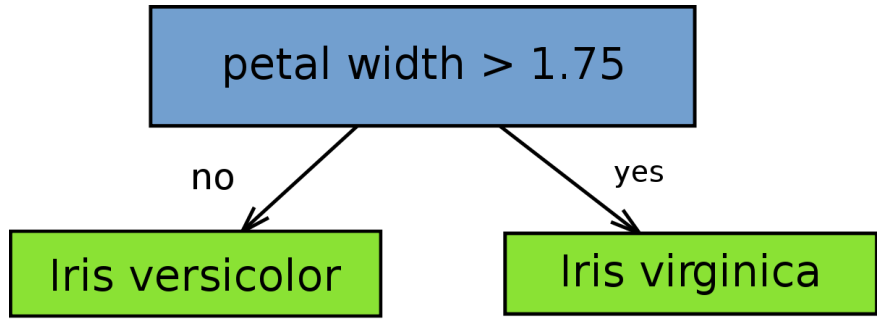
True or False? A tree with depth of 3 has higher variance than a tree with depth of 1.

True.

True or False? A tree with depth of 3 has higher variance than a tree with depth of 1.

True.





True or False? A tree with depth of 3 has higher bias than a tree with depth of 1.

True or False? A tree with depth of 3 has higher bias than a tree with depth of 1.

False.

True or False? A tree with depth of 3 never has higher training error (error on the training set) than a tree with depth 1.

True or False? A tree with depth of 3 never has higher training error (error on the training set) than a tree with depth 1.

True.

True or False? A tree with depth of 3 never has higher test error (error on the training set) than a tree with depth 1.

True or False? A tree with depth of 3 never has higher test error (error on the training set) than a tree with depth 1.

False.

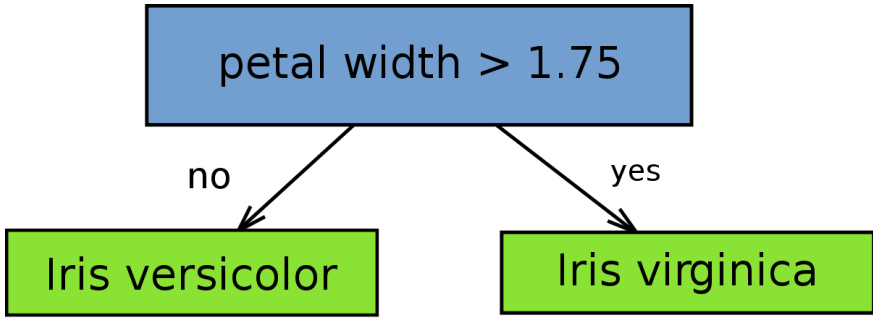
True or False? Decision trees with depth one will always give a linear decision boundary.

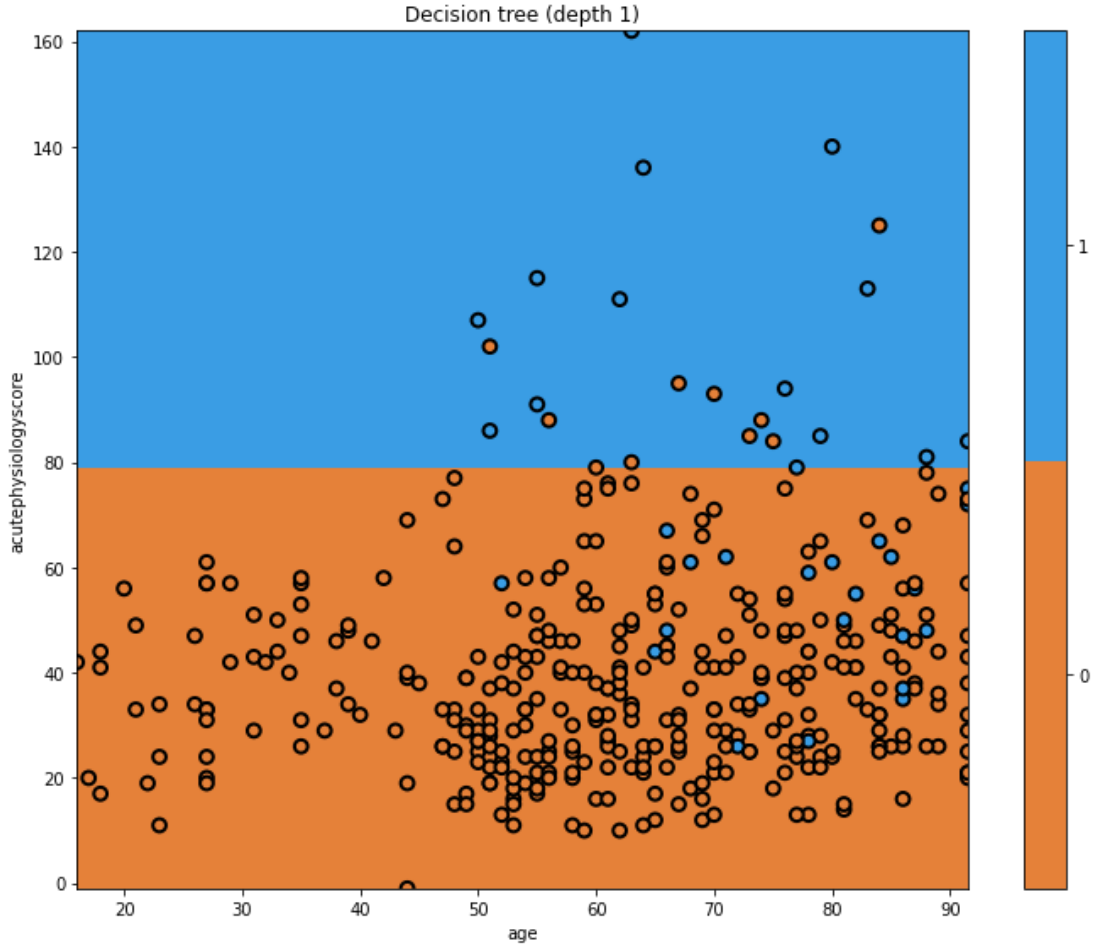
True or False? Decision trees with depth one will always give a linear decision boundary.

True.

True or False? Decision trees with depth one will always give a linear decision boundary.

True.





True or False? Logistic Regression will always give a linear decision boundary.

True or False? Logistic Regression will always give a linear decision boundary.

True.

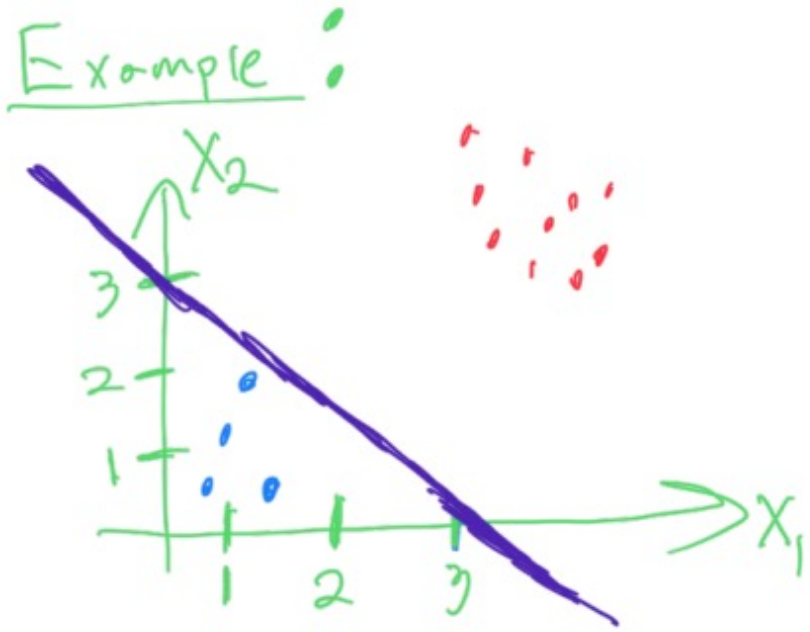
True or False? Logistic Regression will always give a linear decision boundary.

True.

$\theta_0 + \theta_1 x_1 + \theta_2 x_2$ (X-axis in sigmoid plot above)

Predict "y = 1" is $-3 + 1 \cdot x_1 + 1 \cdot x_2 \geq 0$

$\rightarrow$ $x_1 + x_2 = 3$ is the decision boundary



[2 points] Suppose we want to compute 10-Fold Cross-Validation error on 100 training examples. We need to compute error N_1 times, and the Cross-Validation error is the average of the errors. To compute each error, we need to build a model with data of size N_2, and test the model on the data of size N_3.

What are the appropriate numbers for N_1, N_2, N_3?

(a) $N_1 = 10, N_2 = 90, N_3 = 10$

(b) $N_1 = 1, N_2 = 90, N_3 = 10$

(c) $N_1 = 10, N_2 = 100, N_3 = 10$

(d) $N_1 = 10, N_2 = 100, N_3 = 100$

[2 points] Suppose we want to compute 10-Fold Cross-Validation error on 100 training examples. We need to compute error N_1 times, and the Cross-Validation error is the average of the errors. To compute each error, we need to build a model with data of size N_2, and test the model on the data of size N_3.

What are the appropriate numbers for N_1, N_2, N_3?

(a) $N_1 = 10, N_2 = 90, N_3 = 10$

(b) $N_1 = 1, N_2 = 90, N_3 = 10$

(c) $N_1 = 10, N_2 = 100, N_3 = 10$

(d) $N_1 = 10, N_2 = 100, N_3 = 100$

Answer: A.

[2 points] In each round of AdaBoost, the misclassification penalty for a particular training observation is increased going from round t to round $t+1$ if the observation was...

(a) classified incorrectly by the weak learner trained in round t

(b) classified incorrectly by the full ensemble trained up to round t

(c) classified incorrectly by a majority of the weak learners trained up to round t

(d) B and C

(e) A, B, and C

[2 points] In each round of AdaBoost, the misclassification penalty for a particular training observation is increased going from round t to round $t+1$ if the observation was...

(a) classified incorrectly by the weak learner trained in round t

(b) classified incorrectly by the full ensemble trained up to round t

(c) classified incorrectly by a majority of the weak learners trained up to round t

(d) B and C

(e) A, B, and C

Answer: A.

True or False: An important advantage of support vector machines (SVMs) is that they can directly implement classifiers with a large number of classes.

True or False: An important advantage of support vector machines (SVMs) is that they can directly implement classifiers with a large number of classes.

Answer: False.

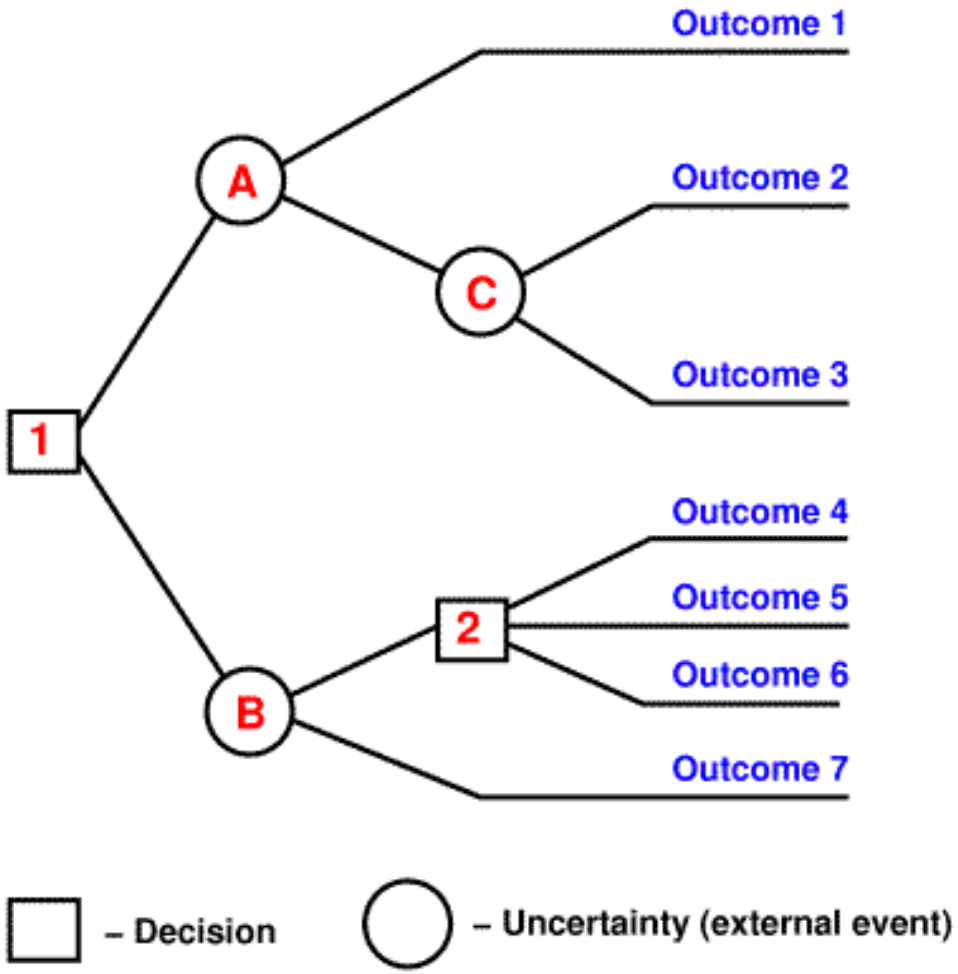
True or False: The ID3 decision tree learning algorithm can only work for binary classification problems.

True or False: The ID3 decision tree learning algorithm can only work for binary classification problems.

Answer: False.

True or False: The ID3 decision tree learning algorithm can only work for binary classification problems.

Answer: False.



True or False: An advantage of using decision trees for machine learning is that the classifiers produced can be easily implemented with rules.

True or False: An advantage of using decision trees for machine learning is that the classifiers produced can be easily implemented with rules.

Answer: True.

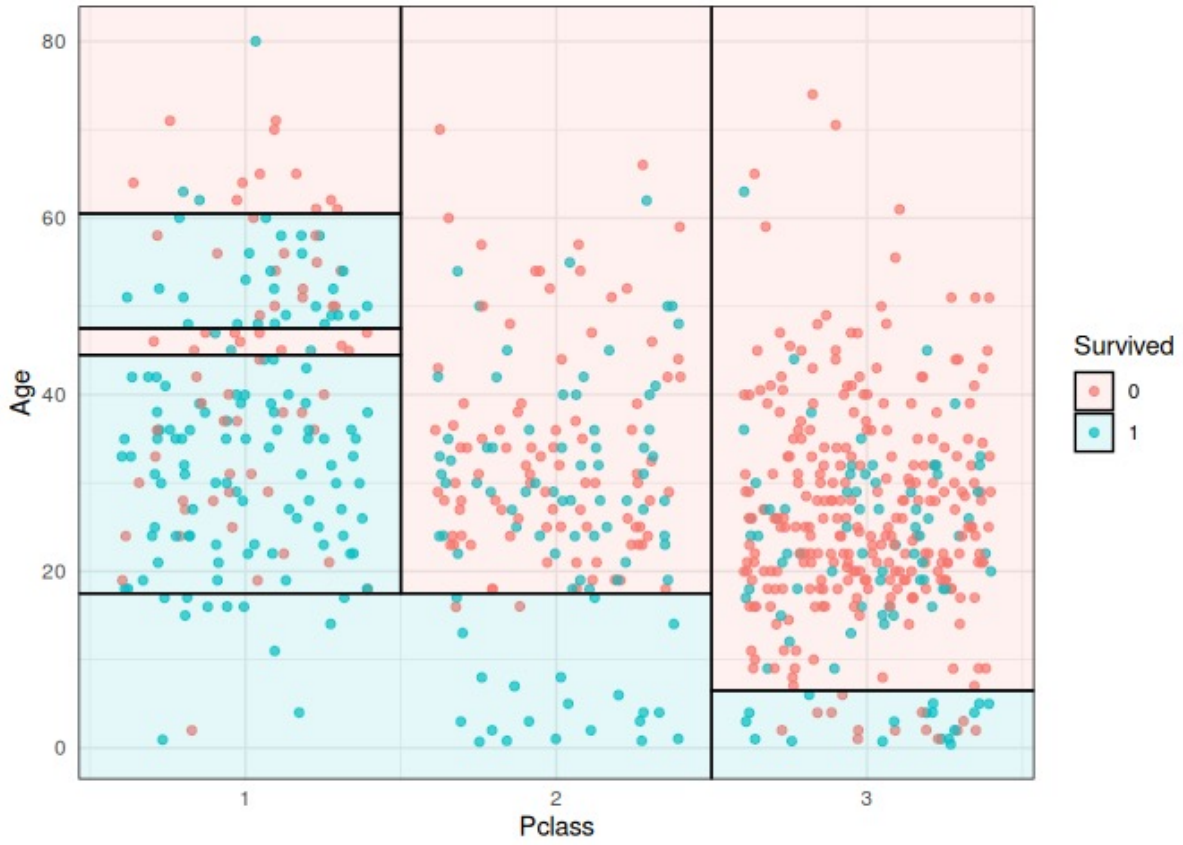
True or False: In theory, a decision tree with N Boolean variables can represent any Boolean function over those N variables.

True or False: In theory, a decision tree with N Boolean variables can represent any Boolean function over those N variables.

Answer: True.

True or False: In theory, a decision tree with N Boolean variables can represent any Boolean function over those N variables.

Answer: True.



In Model based learning methods, an iterative process takes place on the ML models that are built based on various model parameters, called ?

A. mini-batches
B. optimizedparameters
C. hyperparameters
D. superparameters

In Model based learning methods, an iterative process takes place on the ML models that are built based on various model parameters, called ?

A. mini-batches
B. optimizedparameters
C. hyperparameters
D. superparameters

Answer: C

Which of the following models is capable of learning a NON-LINEAR decision boundary?

A. $y = \text{sigmoid}(w_0 + w_1 x_1)$
B. $y = \text{sigmoid}(w_0 + w_1 x_1^3)$
C. $y = \text{sigmoid}(w_0 + w_1 x_1 + w_2 x_2 - w_3 x_3)$
D. All of the above.

Which of the following models is capable of learning a NON-LINEAR decision boundary?

A. $y = \text{sigmoid}(w_0 + w_1 x_1)$
B. $y = \text{sigmoid}(w_0 + w_1 x_1^3)$
C. $y = \text{sigmoid}(w_0 + w_1 x_1 + w_2 x_2 - w_3 x_3)$
D. All of the above.

Answer: B.

Which of the following models is not optimized with a loss function?

A. Logistic Regression
B. Linear Regression
C. Naïve Bayes
D. All of the above use a loss function

Which of the following models is not optimized with a loss function?

A. Logistic Regression
B. Linear Regression
C. Naïve Bayes
D. All of the above use a loss function

Answer: C

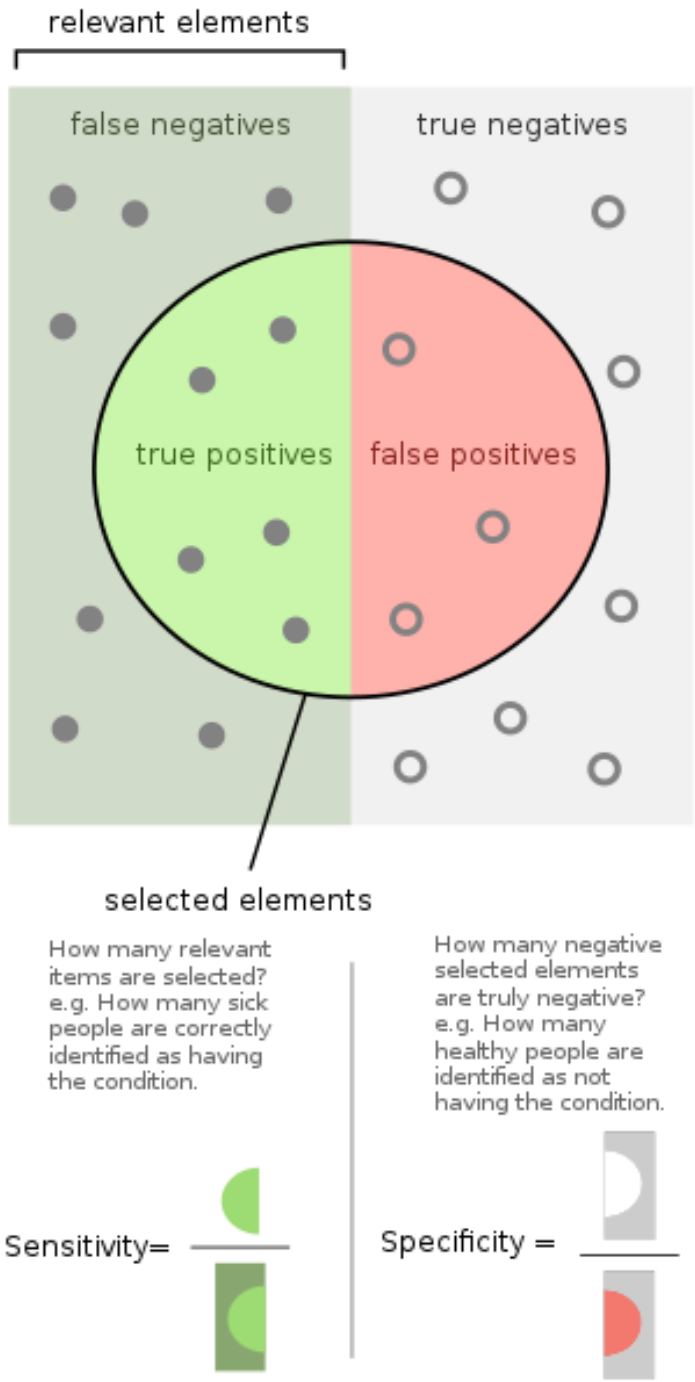
Which evaluation metric below performs poorly with an imbalanced dataset?

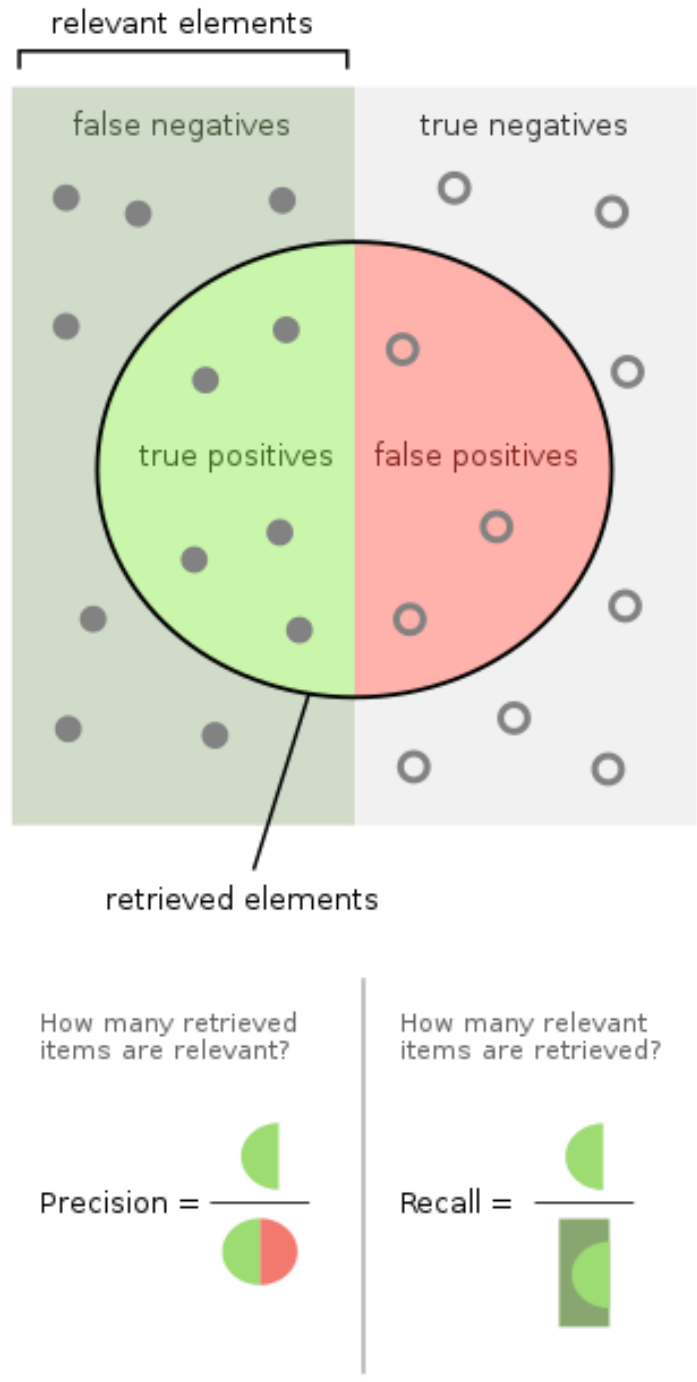
A. Negative Predictive Value
B. Positive Predictive Value
C. Accuracy
D. True Positive Rate
E. True Negative Rate

Which evaluation metric below performs poorly with an imbalanced dataset?

A. Negative Predictive Value
B. Positive Predictive Value
C. Accuracy
D. True Positive Rate
E. True Negative Rate

Answer: C


relevant elements
false negatives
true negatives
true positives
false positives
selected elements
How many relevant items are selected? e.g. How many sick people are correctly identified as having the condition.
Sensitivity=
How many negative selected elements are truly negative? e.g. How many healthy people are identified as not having the condition.
Specificity =


relevant elements
false negatives
true negatives
true positives
false positives
retrieved elements
How many retrieved items are relevant?
Precision =
How many relevant items are retrieved?
Recall =

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

specificity, selectivity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{FPR}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} = 1 - \mathrm{FOR}$$

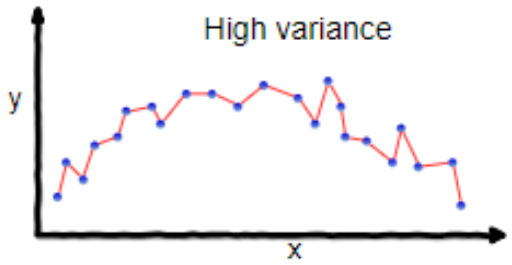
True or False: Simpler models are only preferred over more complex models when there are limited computational resources.

True or False: Simpler models are only preferred over more complex models when there are limited computational resources.

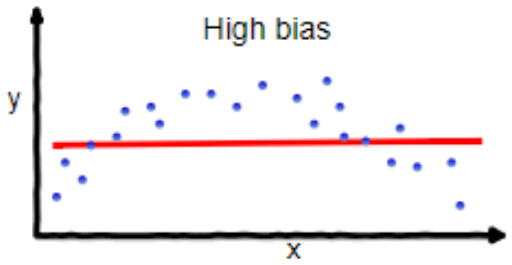
Answer: False.

True or False: Simpler models are only preferred over more complex models when there are limited computational resources.

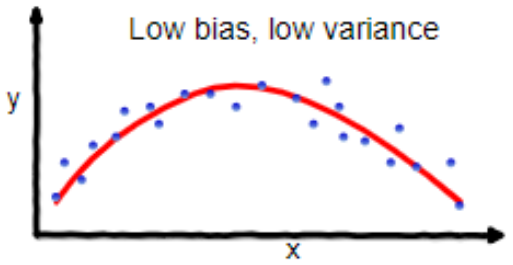
Answer: False.



overfitting



underfitting



Good balance

Which of the gradient descent methods is fastest per iteration?

A. Batch Gradient Descent
B. Mini-Batch Gradient Descent
C. Stochastic Gradient Descent
D. All of the above are equally fast per iteration

Which of the gradient descent methods is fastest per iteration?

A. Batch Gradient Descent
B. Mini-Batch Gradient Descent
C. Stochastic Gradient Descent
D. All of the above are equally fast per iteration

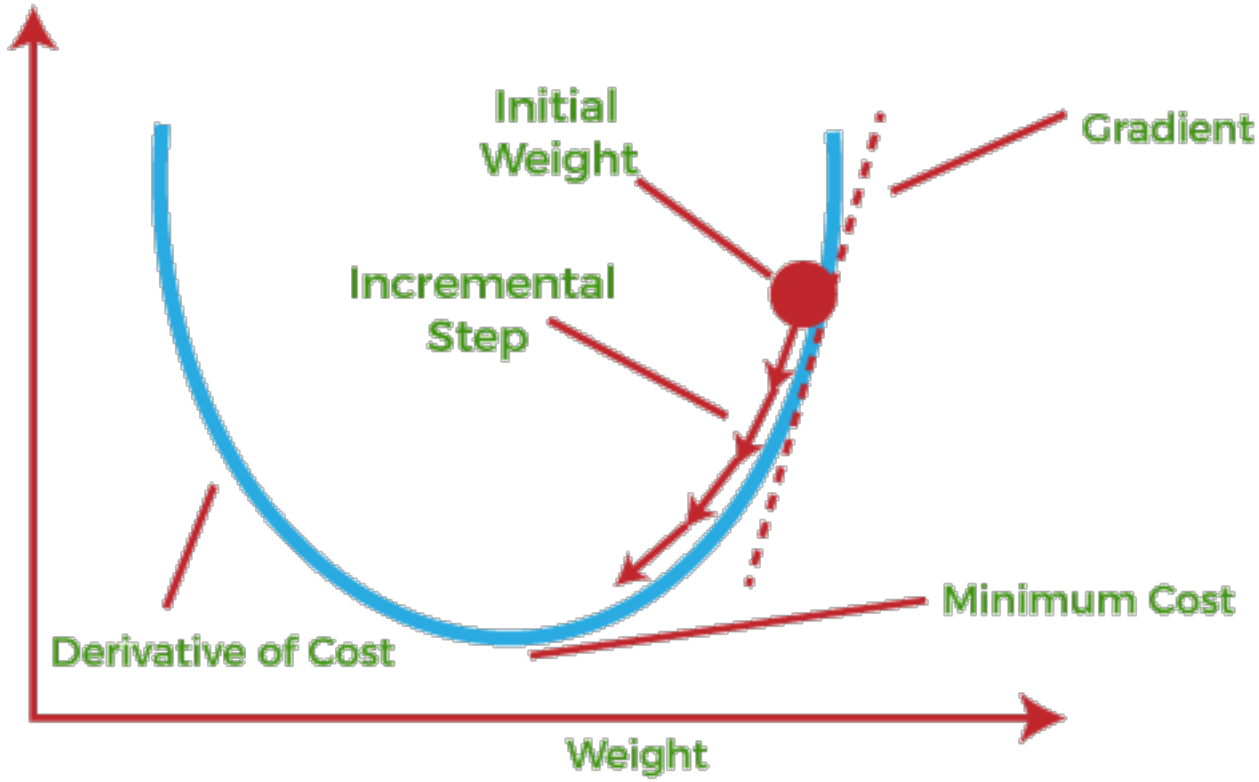
Answer: C

If the loss function is convex, then gradient descent will eventually converge towards a global ____.

A. Maximum
B. Minimum
C. Impossible to tell

If the loss function is convex, then gradient descent will eventually converge towards a global ____.

A. Maximum
B. Minimum
C. Impossible to tell

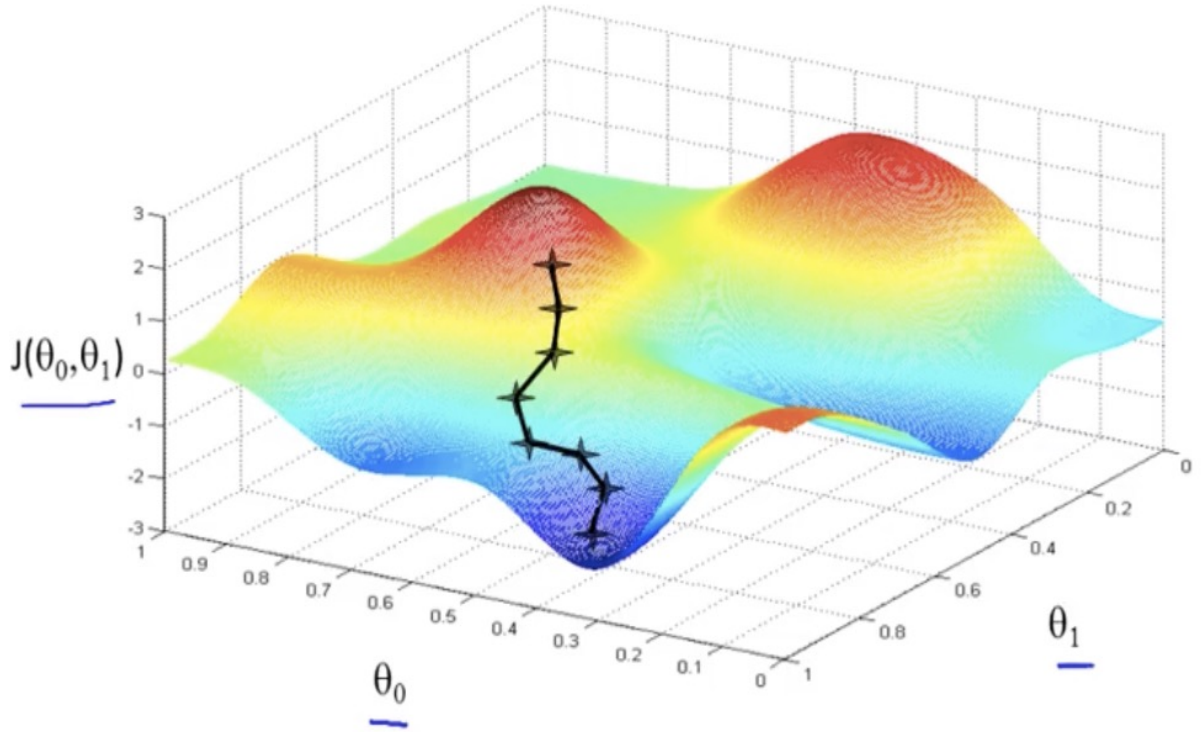


Answer: B

True or False: Gradient Descent only works for 2-dimensional input data.

True or False: Gradient Descent only works for 2-dimensional input data.

Answer: False



True or False: Gradient Descent will work with a non-differentiable loss function.

True or False: Gradient Descent will work with a non-differentiable loss function.

Answer: False

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$