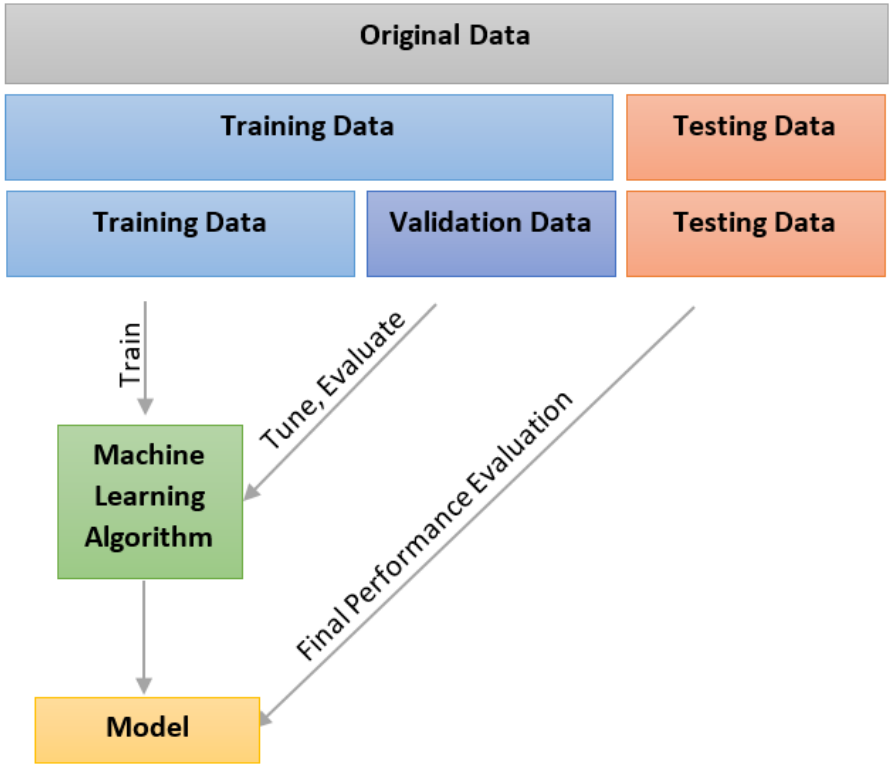
Machine Learning for Predictive Maintenance Applications:
Fault diagnostics

Prof. Dr. Olga Fink

École polytechnique fédérale de Lausanne

October 2025

Training – validation – testing data sets



Statistical tests

What Are Statistical Tests?

- Tools to make **inferences** about populations from sample data.
- Help determine if **observed differences** or **relationships** are **statistically significant**.
- **Why Are They Important?**
  - Distinguish **real effects** from **random variation**.
  - Support **hypothesis testing** in experiments and data analysis.
  - Provide evidence for **decision-making** in research and practice.

Key Concepts

- **Null Hypothesis (H_0):** No effect or difference.
- **Alternative Hypothesis (H_1):** There is an effect or difference.
- **P-value:** Probability of observing data under H_0.
- **Significance Level (α):** Common threshold = **0.05**.

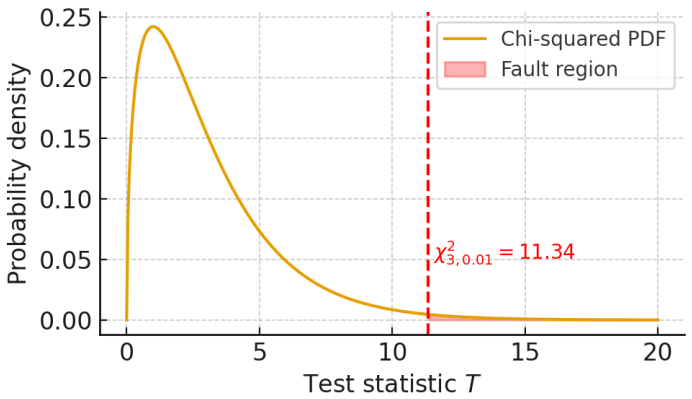
Test	Purpose
t-test	Compare means between two groups
ANOVA	Compare means among three or more groups
Chi-Square Test	Association between categorical variables
Mann-Whitney U	Compare two groups, non-parametric
Wilcoxon Signed-Rank	Compare paired samples, non-parametric

Chi-Squared Test for Residual-Based Fault Detection

- Used to determine if residuals deviate significantly from the healthy model distribution.
- Residuals under healthy operation: $r \sim N(0, \Sigma)$
- Test statistic: $T = r^T \Sigma^{-1} r$
- Decision rule: $T > \chi^2_{n,\alpha} \rightarrow$ Fault detected
- $\chi^2_{n,\alpha}$: critical value of the chi-squared distribution corresponding to significance level α.

Example: for n=3, α=0.01 → $\chi^2_{3,0.01} = 11.34$ defines the fault detection threshold.

- Advantages:
  - – Simple and interpretable statistical test
  - – Direct control of false alarm rate via α
- Limitations:
  - – Assumes Gaussian, uncorrelated residuals
  - – Sensitive to poor covariance estimation



Understanding the Chi-Squared Test

- The Chi-Squared Test evaluates whether residuals deviate significantly from the expected healthy distribution.
- **Test statistic:**
  - $T = r^T \Sigma^{-1} r$
- **where:**
  - r – residual vector (difference between measured and predicted signals)
  - Σ – covariance matrix of residuals under healthy conditions
  - T – test statistic representing the Mahalanobis distance of residuals
  - n – number of residual components (dimension of r)
  - α – significance level (probability of false alarm, e.g., 0.01 or 0.05)
  - $\chi^2_{n,\alpha}$ – threshold from chi-squared distribution; only α% of healthy samples exceed this value
- **Decision rule:**
  - If $T > \chi^2_{n,\alpha} \rightarrow$ fault detected; otherwise, the system is assumed healthy.
- **Interpretation:**
  - The test measures how far current residuals deviate from the healthy model. A high T indicates abnormal behavior.

t-Test

- Compares the means of two groups.
- Types:
  - Independent samples t-test: Compares means of two independent groups.
  - Paired samples t-test: Compares means from the same group at different times.
- Assumptions:
  - Data is continuous and normally distributed.
  - Variances are equal (homogeneity of variance).
  - Observations are independent.
- Example: Comparing the mean strain measurements of two different types of bridge materials under similar load conditions to determine if one material performs significantly better in reducing stress.

t-Test for Detecting Mean Shifts

Test Statistic

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

where:

- $\bar{x}$ — sample mean of the observed data (e.g., residuals)
- μ_0 — expected mean under the healthy (null) hypothesis
- s — sample standard deviation
- n — number of observations (sample size)

Decision Rule

$$|t| > t_{n-1,\alpha} \quad \Rightarrow \quad \text{Reject } H_0$$

Interpretation:

If the test statistic t exceeds the critical value $t_{n-1,\alpha}$ from the **Student's t-distribution** with $n-1$ degrees of freedom,

the mean difference is statistically significant — i.e., a **mean shift (fault)** is detected.

- **Advantages:**
  - Simple to compute and widely used
    Sensitive to small systematic deviations in the mean
- **Limitations:**
  - Assumes data are approximately Gaussian and independent
  - Not effective for variance or distributional changes

Sequential Probability Ratio Test (SPRT)

1. Hypothesis Setup

$$H_0 : r_k \sim p(r|H_0) \quad \text{(healthy)}$$

$$H_1 : r_k \sim p(r|H_1) \quad \text{(faulty)}$$

2. Sequential Test Statistic

$$\Lambda_k = \frac{p(r_k|H_1)}{p(r_k|H_0)}, \quad S_n = \sum_{k=1}^{n} \log \Lambda_k$$

3. Decision Rule

$$\begin{cases} S_n \geq \log A & \Rightarrow \text{Accept } H_1 \text{ (Fault)} \\ S_n \leq \log B & \Rightarrow \text{Accept } H_0 \text{ (Healthy)} \\ \text{Otherwise: continue sampling} \end{cases}$$

$$A = \frac{1-\beta}{\alpha}, \quad B = \frac{\beta}{1-\alpha}$$

α: false alarm (rejecting H_0 when true)

β: missed detection (accepting H_0 when H_1 is true)

Sequential Probability Ratio Test: example

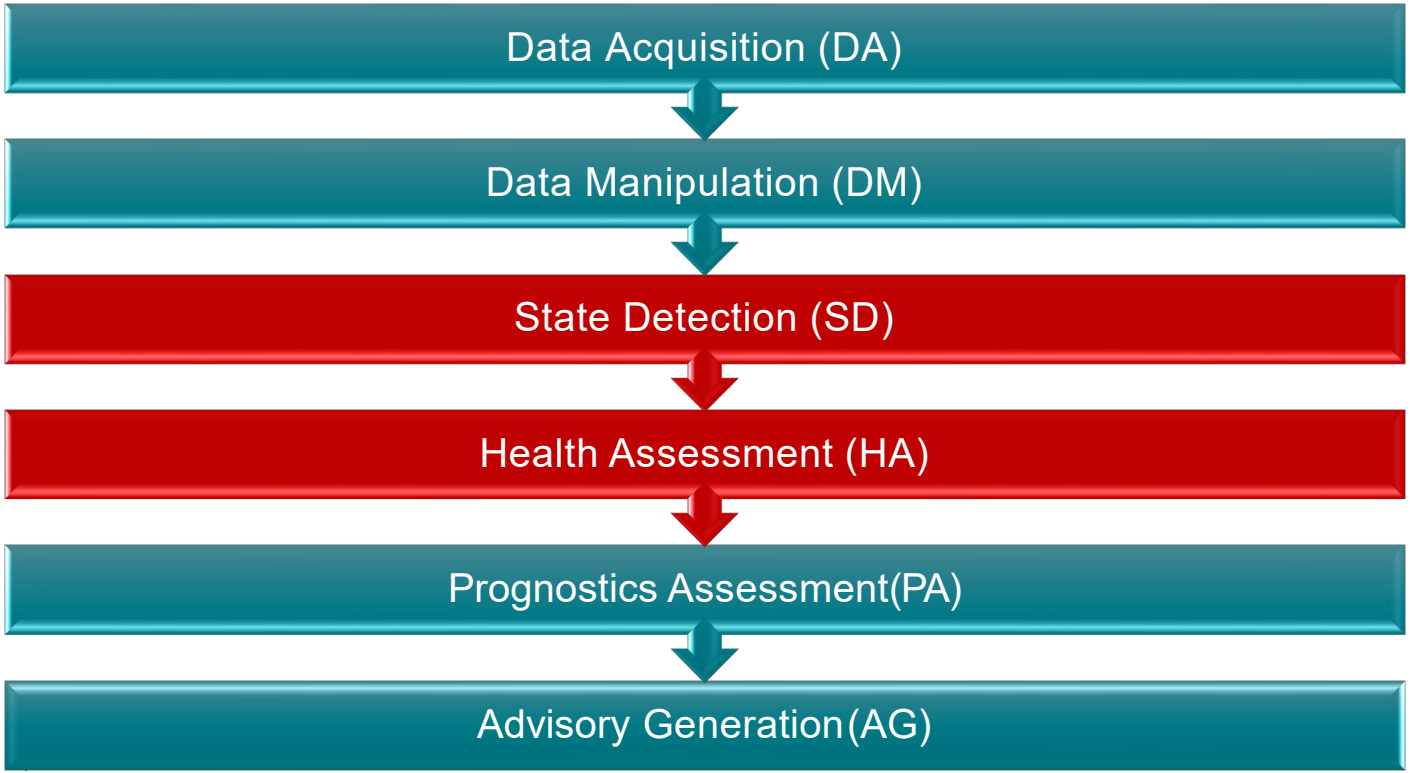
4. Example (Gaussian residuals)

If $r_k \sim \mathcal{N}(0, \sigma^2)$ under H_0

and $r_k \sim \mathcal{N}(\mu_f, \sigma^2)$ under H_1:

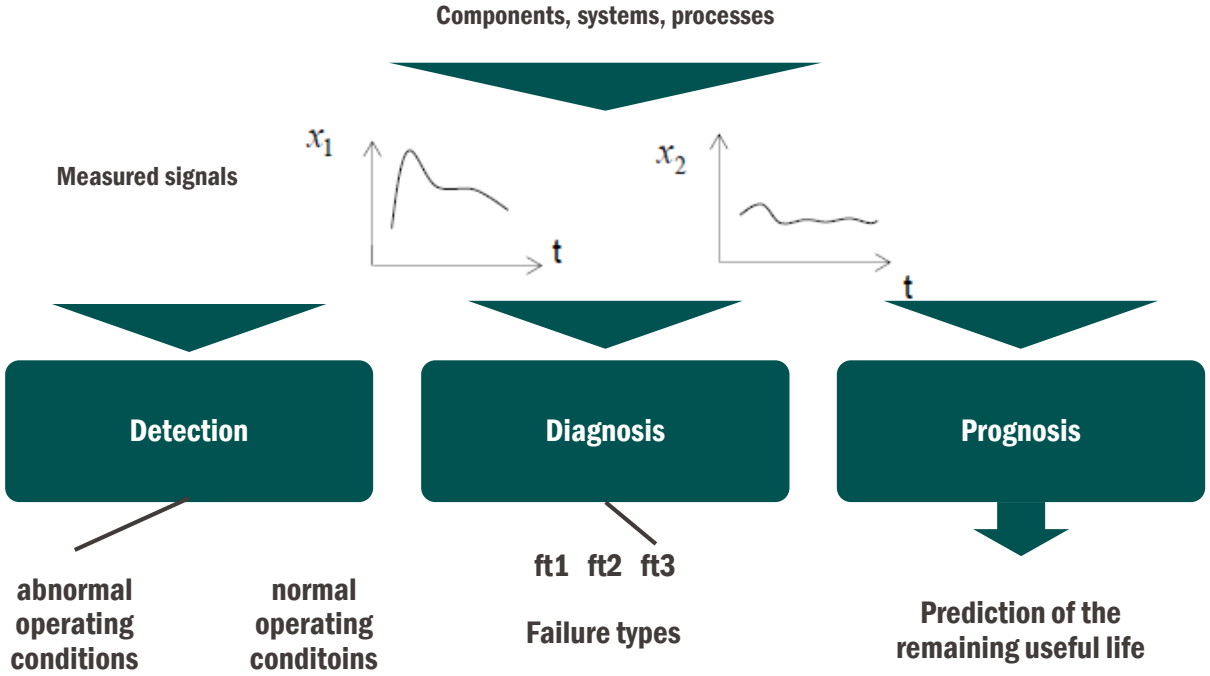
$$\log \Lambda_k = \frac{\mu_f}{\sigma^2} \left(r_k - \frac{\mu_f}{2} \right)$$

PHM Process

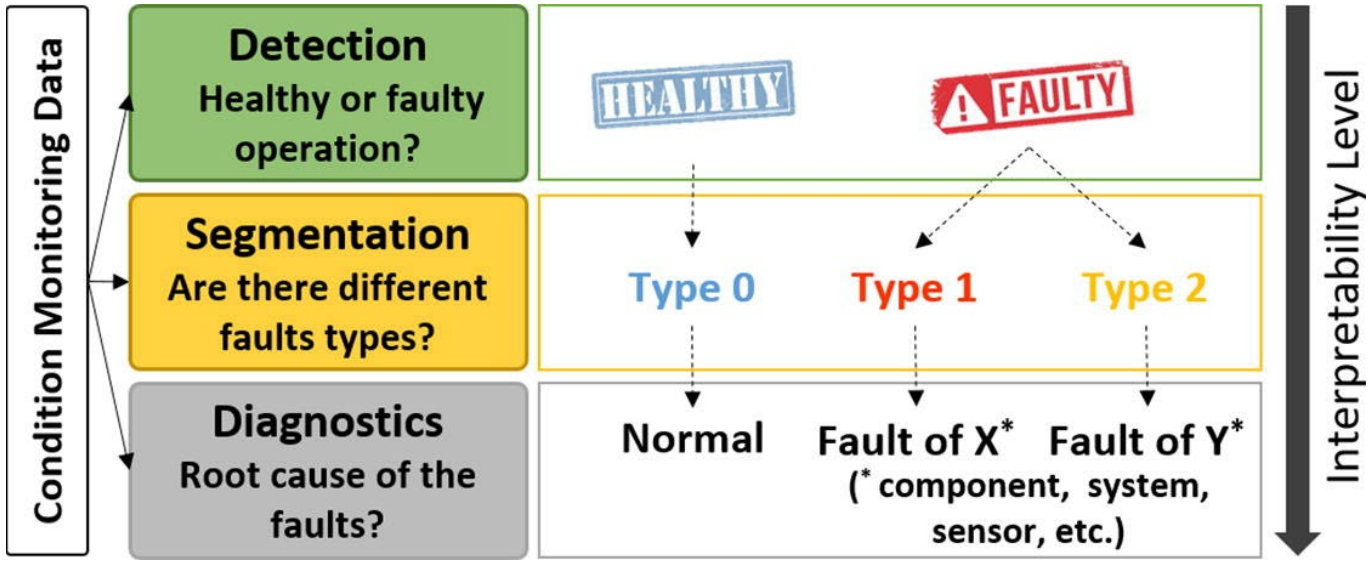

Data Acquisition (DA)
Data Manipulation (DM)
State Detection (SD)
Health Assessment (HA)
Prognostics Assessment(PA)
Advisory Generation(AG)

Diagnostics

Prognostics and health management (PHM)



Fault detection / Fault segmentation / Fault diagnostics



Fault diagnostics

- Typically: Supervised learning
- Unsupervised (combined with fault segmenration+ partial supervision or feedback still required)
- Semi-supervised
- Imbalance challenge needs to be taken into consideration

Fault diagnostics

- Unique Signal Signatures:
  - Different faults produce characteristic patterns in signals (e.g., vibration, acoustic, electrical).
  - These unique signatures help in identifying and differentiating fault types.
- Variations in Signal Amplitude:
  - Faults can cause increases or decreases in signal amplitude.
  - The magnitude of these changes often correlates with the severity and nature of the fault.
- Frequency Content Changes:
  - Faults introduce new frequencies or alter existing ones in the signal spectrum.
  - Spectral analysis reveals these frequency components, aiding in fault detection.
- Time-Domain Pattern Changes:
  - Faults may cause irregularities like spikes, transients, or repetitive patterns over time.
  - Time-domain analysis captures these anomalies for diagnostics.

Different fault types impact captured signals in distinct and detectable ways (1/3)

- Phase Shifts and Timing Differences:
  - Faults can lead to shifts in the phase or timing of signals.
  - Analyzing phase relationships helps localize and identify faults.
- Harmonics and Sidebands Generation:
  - Fault-induced nonlinearities create harmonics or sidebands in signals.
  - These features are indicative of specific fault conditions.
- Statistical Feature Alterations:
  - Faults affect statistical properties like mean, variance, skewness, and kurtosis.
  - Statistical analysis detects deviations from normal operating conditions.
- Energy Distribution Shifts:
  - Faults redistribute signal energy across different frequency bands.
  - Energy-based methods identify abnormal patterns associated with faults.
- Entropy and Complexity Changes:
  - Faults increase the randomness or complexity of signals.
  - Entropy measures help in detecting these changes.

Different fault types impact captured signals in distinct and detectable ways (2/3)

- Signal Correlation and Coherence:
  - Faults alter the relationships between signals from multiple sensors.
  - Cross-correlation and coherence analysis pinpoint inconsistencies due to faults.
- Dynamic System Response Alterations:
  - Faults change system dynamics like stiffness or damping.
  - Monitoring dynamic responses reveals these alterations.
    - Stiffness Alterations: Cracks or material degradation reduce stiffness, resulting in larger deflections under load.
    - Damping Changes: Damage can decrease damping, causing vibrations to persist longer after excitation.
    - Natural Frequency Shifts: Structural damage may lower or raise the bridge's natural frequencies, affecting resonance conditions.
- Thermal Anomalies:
  - Some faults generate excess heat detectable by temperature sensors or thermal imaging.
  - Thermal monitoring identifies hotspots indicating faulty components.
    - E.g. Elevated temperatures at a bridge's expansion joint may indicate excessive friction due to wear or misalignment.
- Acoustic Emission Patterns:
  - Faults emit characteristic acoustic or ultrasonic signals.
  - Acoustic sensors capture these emissions for early fault detection.
    - Detecting high-frequency AE signals in a bridge's truss members can indicate crack initiation or growth.
- Electrical Parameter Variations:
  - Electrical faults affect current, voltage, or impedance.
  - Electrical measurements detect anomalies in circuits and components.

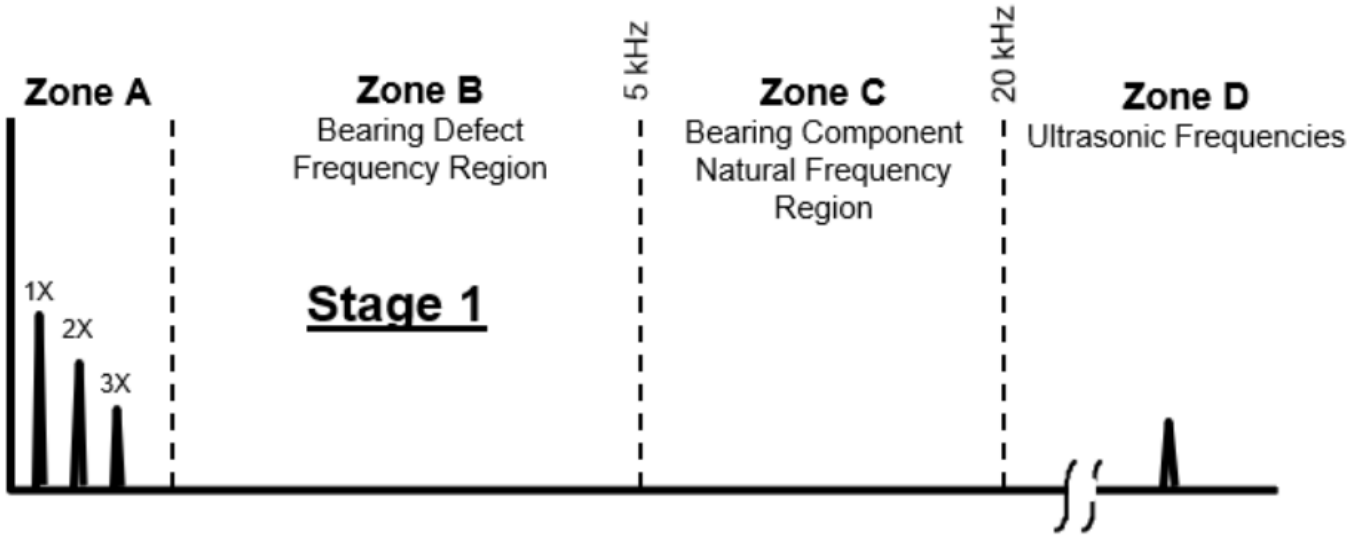
Different fault types impact captured signals in distinct and detectable ways (3/3)

- Chemical and Material Changes:
  - Faults can cause material degradation or chemical reactions (e.g., gas generation).
  - Sensors detect these changes, indicating specific types of faults.
- Load and Operating Condition Dependencies:
  - Fault effects vary with load, speed, or environmental conditions.
  - Observing signal changes under different conditions aids fault identification.
- Control System Deviations:
  - Faults in actuators or sensors cause deviations in control signals.
  - Monitoring control loops helps detect and diagnose these faults.
- Nonlinear Behavior Introduction:
  - Faults introduce nonlinear characteristics into system responses.
  - Nonlinear analysis techniques detect these behaviors.
- Multi-Sensor Data Fusion:
  - Combining data from various sensors provides a comprehensive view.
  - Faults impact different sensors uniquely, and data fusion enhances diagnostics.

Example: fault types of a bearing (particularly rolling element bearings)

- **Inner Race Faults**:
  - **Description**: Defects or damage to the inner raceway surface where the rolling elements make contact.
  - **Causes**: Excessive loads, misalignment, improper installation, or material fatigue.
  - **Effects**: Can lead to increased vibration at specific frequencies associated with the inner race.
- **Outer Race Faults**:
  - **Description**: Damage or wear on the outer raceway surface.
  - **Causes**: Contamination, uneven loading, or improper mounting.
  - **Effects**: Increased vibration and noise due to uneven load distribution, leading to accelerated wear, reduced operational efficiency
- **Rolling Element Faults:**
  - **Description**: Defects on the balls or rollers themselves, such as pitting, spalling, or cracking.
  - **Causes**: Material imperfections, contamination, inadequate lubrication, or overloading.
  - **Effects**: Leads to erratic vibration patterns and potential seizure of the bearing.
- **Cage (Separator) Faults**:
  - **Description**: Damage or deformation of the cage that holds and spaces the rolling elements.
  - **Causes**: High speeds, excessive vibration, improper lubrication, or mechanical stress.
  - **Effects**: Causes uneven spacing of rolling elements, leading to additional stress and potential failure.

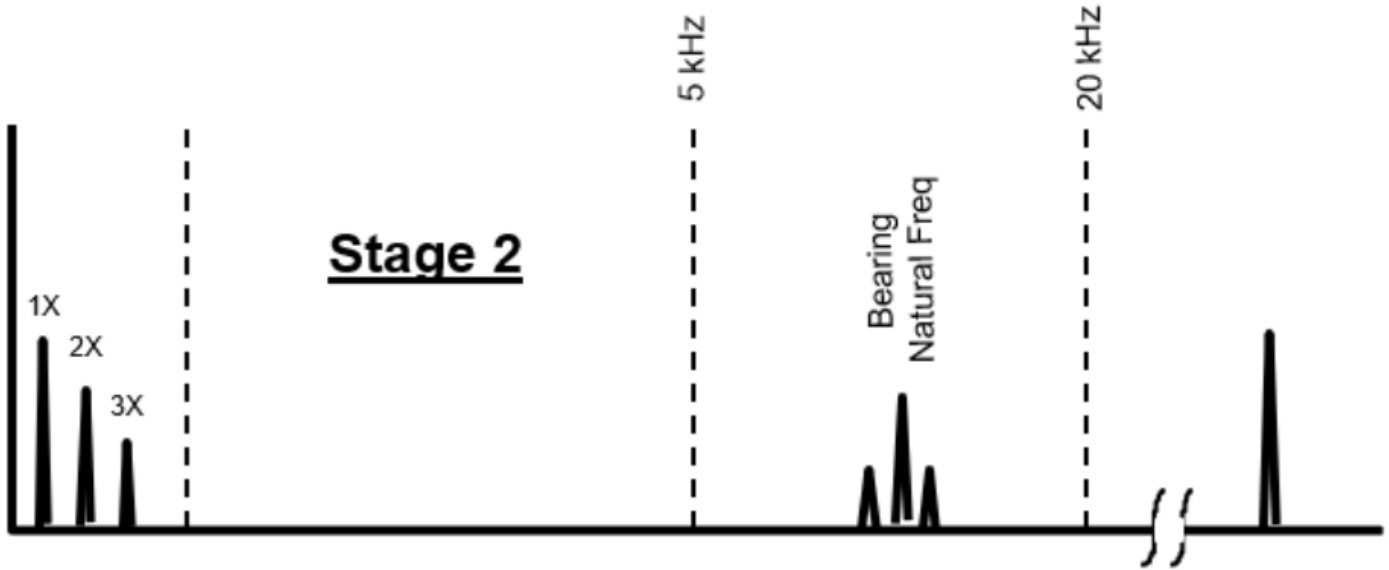
Bearing faults: stage 1



Enveloped Acceleration Spectrum (EAS)

Source: Reliabilityconnect.com

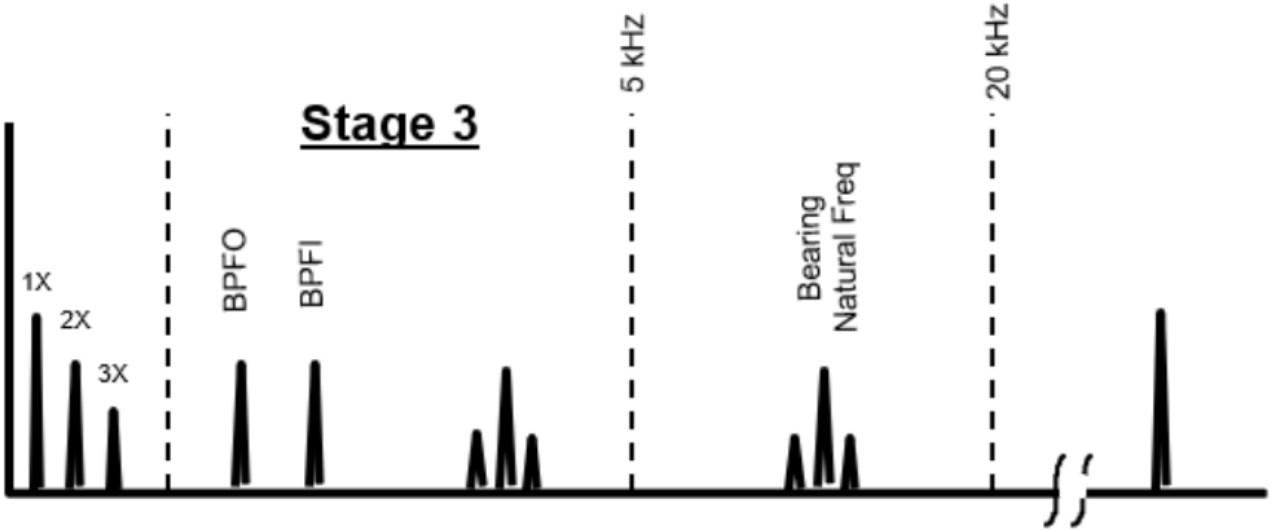
Bearing faults: stage 2



Enveloped Acceleration Spectrum (EAS)

Source: Reliabilityconnect.com

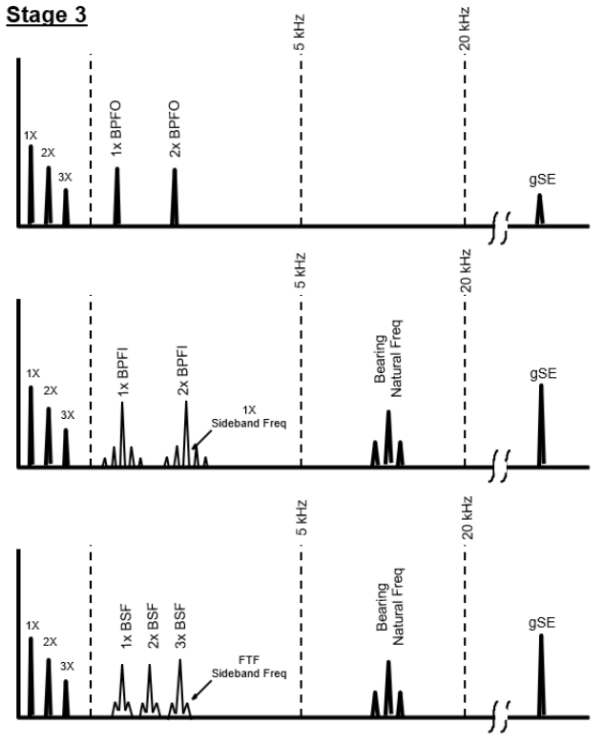
Bearing faults: stage 3

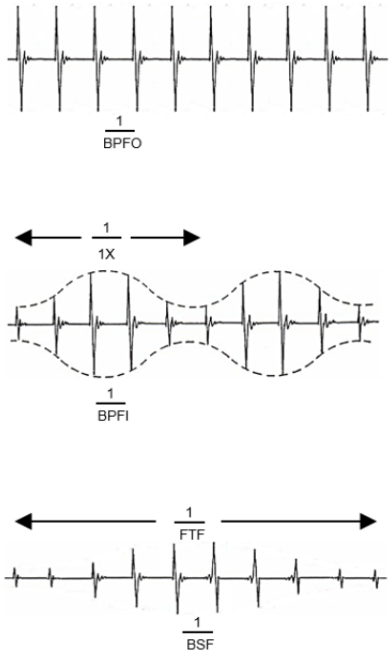


Enveloped Acceleration Spectrum (EAS)

Source: Reliabilityconnect.com

Bearing faults: stage 3

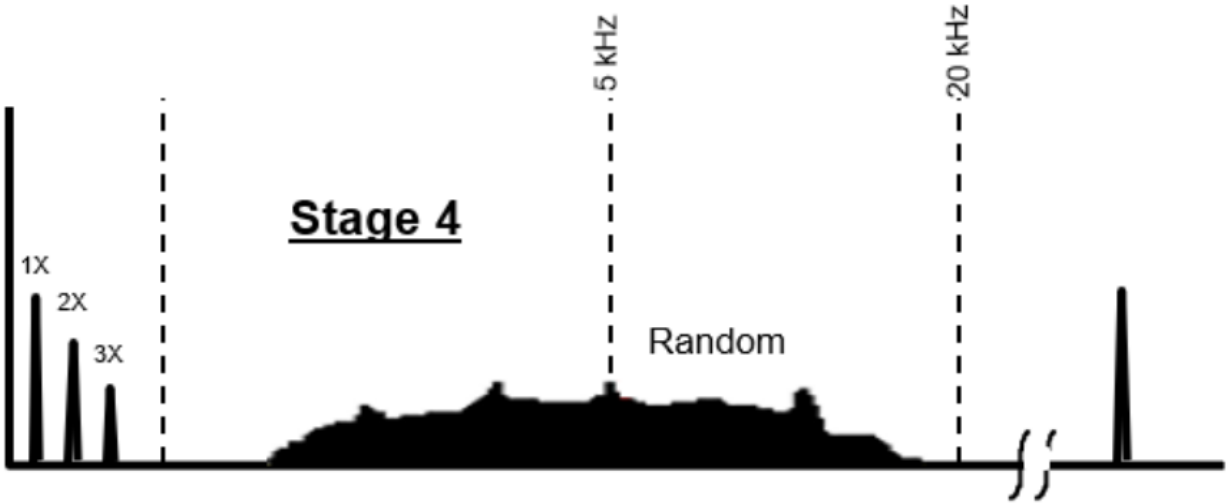




- BPFO (Ball Pass Frequency Outer race): Associated with defects on the outer race.
- BPFI (Ball Pass Frequency Inner race): Associated with defects on the inner race.
- BSF (Ball Spin Frequency): Associated with defects in the rolling elements (balls or rollers).
- FTF (Fundamental Train Frequency): Associated with defects in the bearing cage itself.

Source: Reliabilityconnect.com

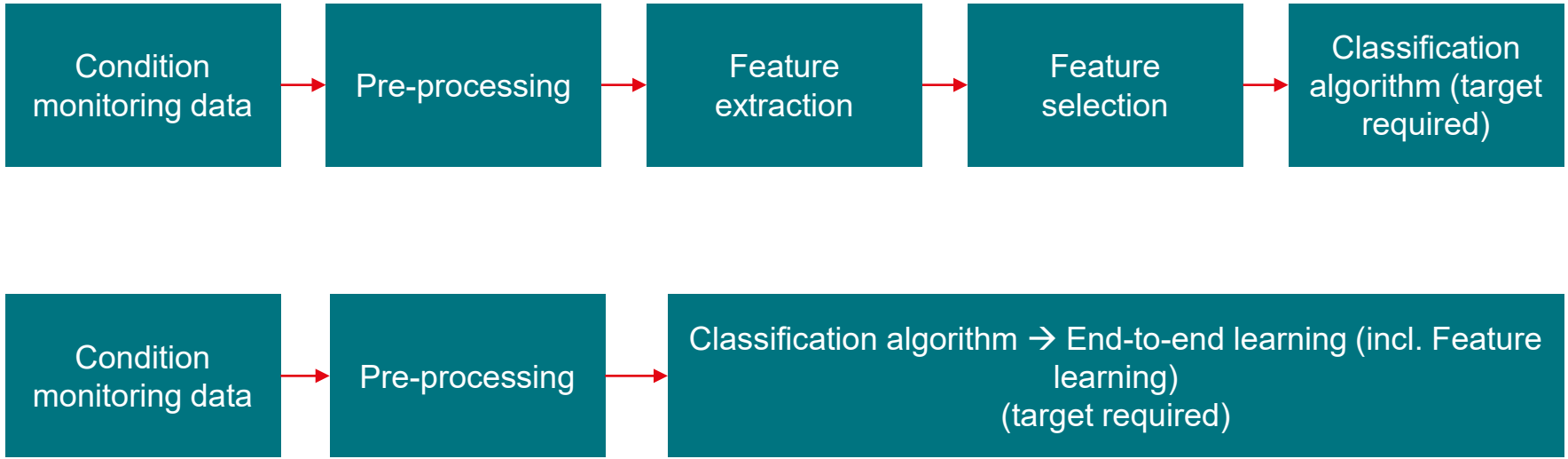
Bearing faults: stage 4



Enveloped Acceleration Spectrum (EAS)

Source: Reliabilityconnect.com

Fault diagnostics steps



Recap: Selected supervised learning algorithms

Recap: Logistic Regression

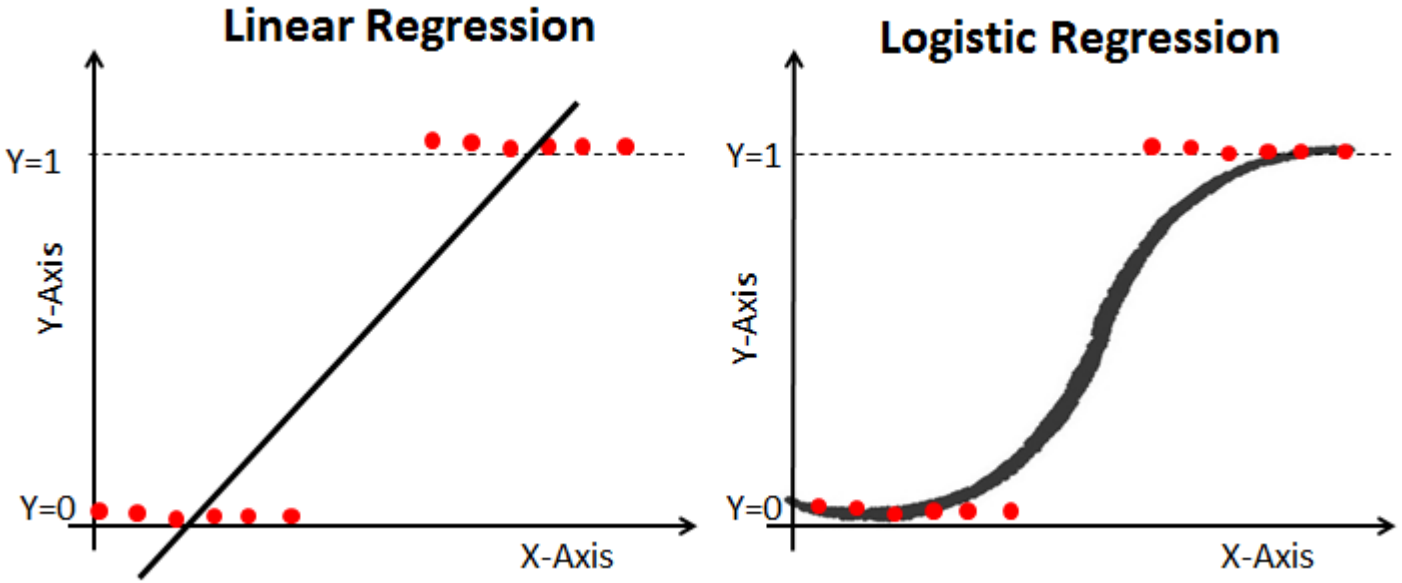
Logistic regression

- Logistic regression is a statistical model used to predict the probability of a binary outcome (i.e., a "yes" or "no" answer) based on one or more predictor variables.
- It is a type of regression analysis commonly used in machine learning and statistics to model the relationship between the input features and the binary target variable.

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

$$f(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}$$

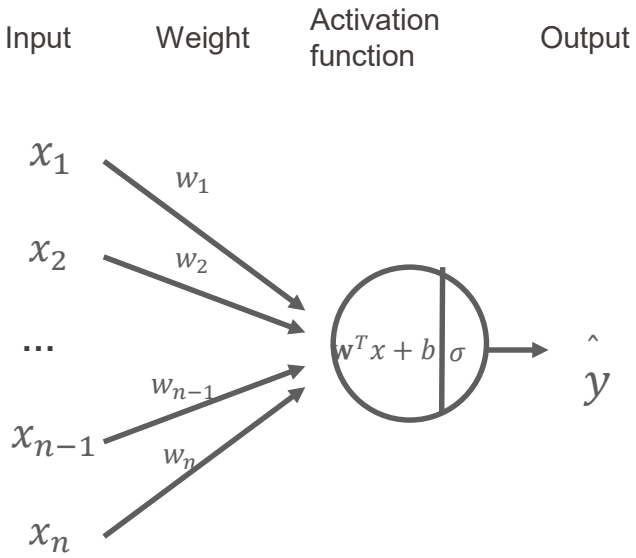
Linear regression vs. Logistic regression



Source: www.towardsdatascience.com

Logistic Regression

Visualisation



1) Linear Regression:

$$z^{(i)} = \mathbf{w}^T \boldsymbol{x}^{(i)} + b$$

2) Activation Function:

$$\hat{y}^{(i)} = \sigma(z^{(i)})$$

3) Classification:

$$if \hat{y}^{(i)} > 0.5 \Rightarrow label1$$

$$if \hat{y}^{(i)} \leq 0.5 \Rightarrow label0$$

Logistic Regression

Loss function

How to learn w?

Binary cross-entropy loss

$$J = -\frac{1}{N}\sum_{i=1}^{N} y^{(i)}\log(h_{\mathbf{w}}(\boldsymbol{x}^{(i)})) + (1-y^{(i)})\log(1-h_{\mathbf{w}}(\boldsymbol{x}^{(i)}))$$

Active when y = 1
Penalize when $h_w(x) \to 0$

Active when y = 0
Penalize when $h_w(x) \to 1$

Logistic Regression

Softmax regression

Softmax function

$$\sigma(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}, i \in \{1,\dots,K\}$$

After applying softmax, each component will be in the interval $(0,1)$ and the component will add up to 1, so that they can be interpreted as probabilities

Softmax regression is a **generalization** of logistic regression to multi-class problems.

Note: For softmax regression, $\mathbf{z}$ is obtained as follows:

- $\mathbf{z} = \mathbf{W}^T\boldsymbol{x} + \mathbf{b}$
- $z_k = \mathbf{w}_k^T\boldsymbol{x} + b_k$ **where** $\mathbf{w}_k$ **is the k-th column of the D x K weight matrix** $\mathbf{W}$**.**

Logistic Regression

Categorical cross-entropy

Categorical Cross-Entropy Loss

$$J(\mathbf{w}, b) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{1}\{y^{(i)} = k\} \log\left(\frac{\exp(\mathbf{w}_k^T \boldsymbol{x}^{(i)} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \boldsymbol{x}^{(i)} + b_j)}\right)$$

where $\mathbf{1}\{y^i = k\}$ is "indicator function", it works as:

$\mathbf{1}\{\textit{True statement}\} = 1$ and $\mathbf{1}\{\textit{False statement}\} = 0$

Assumptions

- Linearity in Log-Odds: Assumes a linear relationship between the independent variables and the log-odds of the dependent variable
- Independence of Observations: Assumes that the observations are independent of each other.
- No or Little Multicollinearity: Assumes that independent variables are not highly correlated.

Recap: k-Nearest Neighbor algorithm

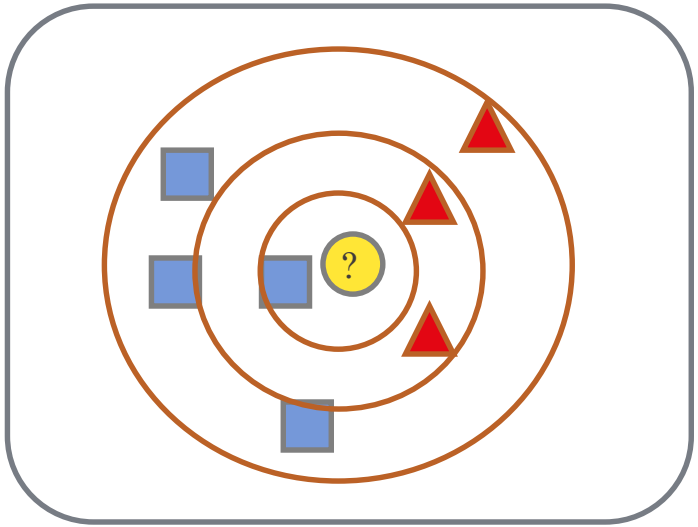
Instance based learning

- Approximating real valued or discrete-valued target functions
- Learning in this algorithm consists of storing the presented training data
- When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query instance
- Construct only local approximation to the target function that applies in the neighborhood of the new query instance
- Instance-based methods can use vector or symbolic representation
- Appropriate definition of „neighboring“ instances
- Disadvantage of instance-based methods is that the costs of classifying new instances can be high
- Nearly all computation takes place at inference time rather than learning time

Source: Fenix, 2015

k-Nearest Neighbor algorithm

- Most basic instance-based method
- Data are represented in a vector space
- Supervised learning algorithm
- Distance measure required
- Requires 3 things:
  - Feature Space (Training Data)
  - Distance metric
    - to compute distance between records
  - The value of *k*
    - the number of nearest neighbors to retrieve from which to get majority class



How kNN Works

- **Choose the Number of Neighbors (k)**:
  - Decide the number of nearest neighbors (k) to consider for making predictions. The choice of k can significantly impact the algorithm's performance.
- **Compute Distances**:
  - For a given input data point, calculate the distance between this point and all points in the training dataset using a chosen distance metric.
- **Identify k Nearest Neighbors**:
  - Select the k training data points with the smallest distances to the input point.
- **Make a Prediction**:
  - **Classification**: Assign the class that is most common among the k nearest neighbors (majority voting).
  - **Regression**: Compute the average (or weighted average) of the target values of the k nearest neighbors.

k-NN Distance measures

- Common Distance Metrics:
  - Euclidean distance (continuous distribution)

$$\|\vec{x} - \vec{y}\| = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

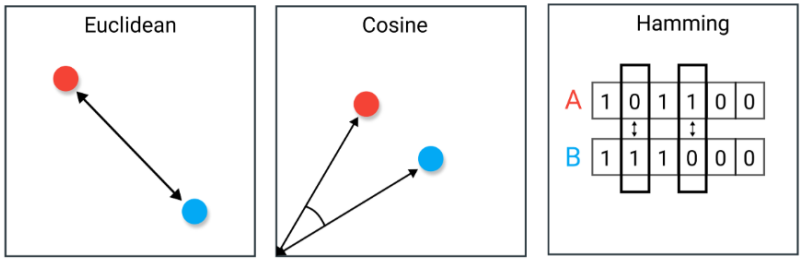
  - Cosine similarity

$$Sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|}$$

  - Hamming distance (overlap metric)

bat (distance = 1)
cat

toned (distance = 3)
roses



  - Discrete Metric(boolean metric)

if x = y then d(x,y) = 0. Otherwise, d(x,y) = 1

- Determine the class from ***k*** nearest neighbor list
  - Take the majority vote of class labels among the k-nearest neighbors
  - Weighted factor: w =1/d (generalized linear interpolation) or $1/d^2$

Source: Connor, 2006

Advantages / disadvantages of kNN

- **Advantages of kNN**
  - **Simplicity**: Easy to understand and implement.
  - **Flexibility**: Applicable to both classification and regression problems.
  - **No Training Phase**: Since it's a lazy learner, there's no time-consuming training process.
  - **Adaptable**: Can handle multi-class problems and adapt to changes in the dataset dynamically.
- **Disadvantages of kNN**
  - **Computationally Intensive**: Requires calculating distances to all training data points during prediction, which can be slow for large datasets.
  - **Memory Usage**: Needs to store the entire training dataset, which can be memory-consuming.
  - **Choice of k**: Selecting the optimal k is crucial. A small k can make the model sensitive to noise, while a large k can smooth out the decision boundaries excessively.
  - **Curse of Dimensionality**: Performance degrades with high-dimensional data because the distance measures become less meaningful.
  - **Sensitive to Irrelevant Features**: Including irrelevant or redundant features can negatively impact the algorithm's performance.

Curse of Dimensionality

- Imagine instances described by 20 features (attributes) but only 3 are relevant to target function
- Curse of dimensionality: nearest neighbor is easily misled when instance space is high-dimensional
- Dominated by large number of irrelevant features

Possible solutions

- Stretch j-th axis by weight z_j, where $z_1,\ldots,z_n$ chosen to minimize prediction error (weight different features differently)
- Use cross-validation to automatically choose weights $z_1,\ldots,z_n$
- Feature subset selection if z_j set zero
- Dimensionality reduction

Source: Fenix, 2015

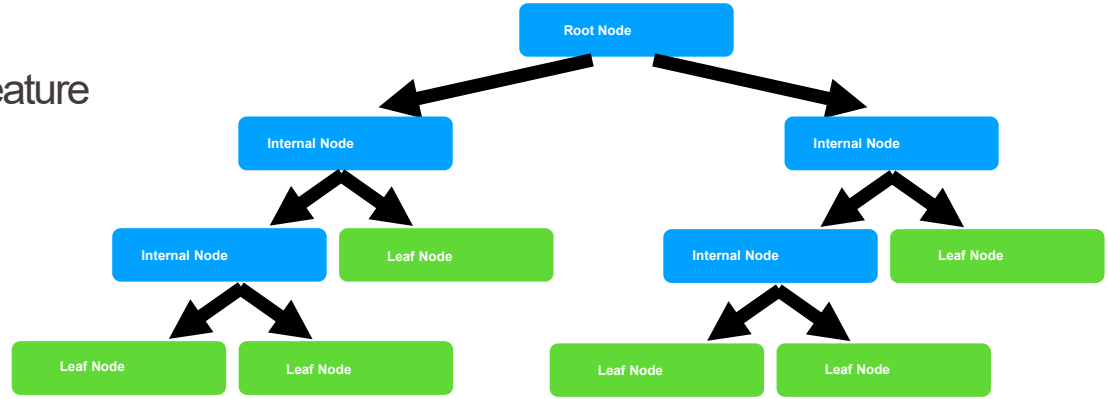
How unserupervised k-NN can work for anomaly detection

- **Data Collection & Preprocessing**
  - Collect sensor data (e.g., vibration, strain, displacement).
  - Extract meaningful features (RMS, peak amplitude, frequencies).
- **Build a Reference Dataset**
  - Use data from normal (healthy) operating conditions.
- **Apply k-NN Algorithm**
  - For each new data point, find its K nearest neighbors from the reference dataset.
  - Calculate the distance (Euclidean or other) to its neighbors.
- **Anomaly Detection Rule**
  - If the average distance is greater than a threshold, classify it as anomalous.
  - Threshold selection: Based on validation data or statistical methods.

Recap: Decision Trees

Decision Trees

- Nodes are checked on a single feature
- Branches are feature values
- Leaves indicate class label



Decision Trees

- Intuitive and versatile supervised learning algorithms used for both classification and regression tasks
- Model decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
- Favored for their simplicity, interpretability, and ability to handle both numerical and
- **Structure:** Composed of nodes and branches:
  - **Root Node:** The topmost node representing the entire dataset.
  - **Internal Nodes:** Represent features or attributes used to split the data.
  - **Leaf Nodes (Terminal Nodes):** Represent the final output or decision (class label or continuous value).
- **Splitting Criteria:**
  - **Classification:**
    - **Gini Impurity:** Measures the frequency at which any element of the dataset will be mislabeled when it is randomly labeled.
    - **Entropy (Information Gain):** Measures the amount of information disorder or randomness.
    - **Information Gain (IG):** The reduction in entropy after a dataset is split on an attribute.
  - **Regression:**
    - **Mean Squared Error (MSE):** Measures the average of the squares of the errors between predicted and actual values.
    - **Mean Absolute Error (MAE):** Measures the average of the absolute differences between predicted and actual values.

Advantages / Limitations

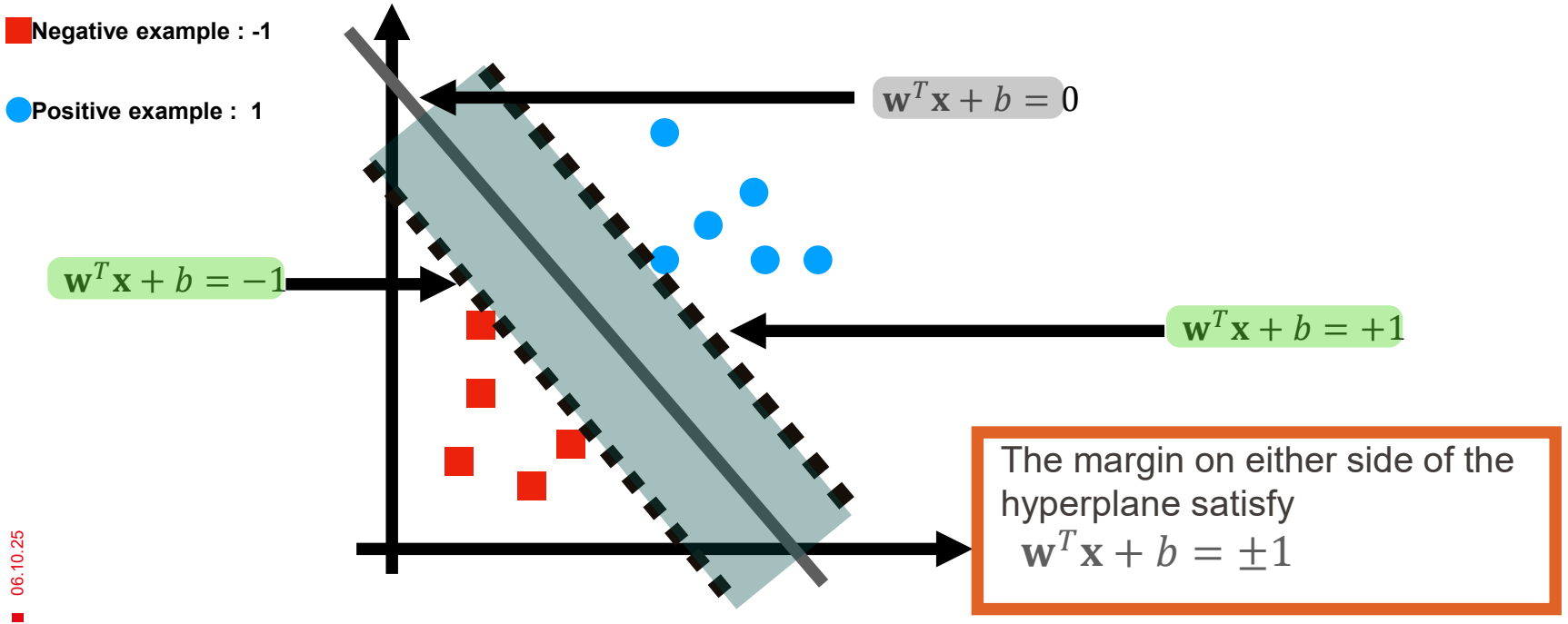
- **Advantages**
  - **Interpretability:** Decision Trees are easy to visualize and understand, making them transparent models where decisions can be traced and explained.
  - **Handling Mixed Data Types:** Capable of handling both numerical and categorical features without the need for extensive preprocessing.
  - **Non-parametric:** No assumptions about the underlying data distribution, making them flexible in modeling complex relationships.
  - **Feature Importance:** Naturally provides insights into feature importance, aiding in feature selection and understanding data.
  - **Robustness to Outliers:** Less sensitive to outliers compared to some other algorithms, especially in regression tasks.
- **Limitations**
  - **Overfitting:** Prone to creating overly complex trees that capture noise in the data, reducing generalization performance.
  - **Instability:** Small changes in the data can lead to significantly different tree structures, affecting consistency.
  - **Bias Toward Dominant Classes:** In classification tasks with imbalanced classes, trees may become biased toward the majority class.
  - **Greedy Algorithms:** Standard tree-building algorithms make locally optimal choices at each node, which may not lead to the globally optimal tree.
  - **Poor Performance on Certain Data Types:** May struggle with capturing smooth relationships in regression tasks compared to models like linear regression.

Recap: Support Vector Machines

Support Vector Machines (SVM)

Optimal hyperplane separation

We obtain our constraints



The margin on either side of the hyperplane satisfy

$$\mathbf{w}^T\mathbf{x} + b = \pm 1$$

SVM: Optimal hyperplane separation

We obtain an optimization problem:

$$\min_{\mathbf{w},b} \frac{\| \mathbf{w} \|^2}{2}$$

Objective

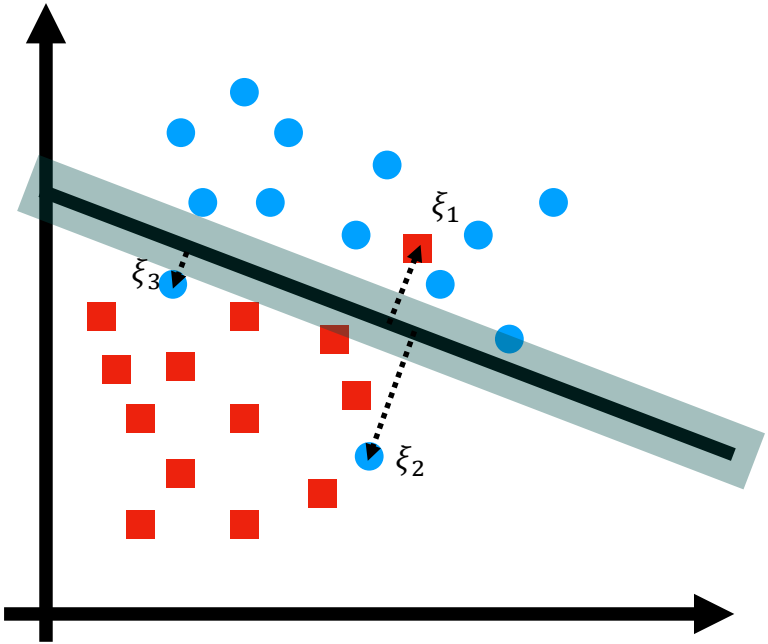
$$\left.\begin{array}{l} \mathbf{w}^T\mathbf{x}^{(i)} + b \geq 1 \text{ when } \; y^{(i)} = +1 \\ \mathbf{w}^T\mathbf{x}^{(i)} + b \leq -1 \text{ when } \; y^{(i)} = -1 \end{array}\right\} y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 \text{ when } \; i = 1,2,\ldots,M$$

Constraints

This is hard-margin SVM, and it work only for separable data

SVM: Dealing with non-separable data

What should we do ?



Constraints relaxed by slack variables ξ_i

$$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \xi_i s.t \xi_i \geq 0 \ \forall i = 1,2,\ldots,N$$

M : number of examples in the margin or misclassified

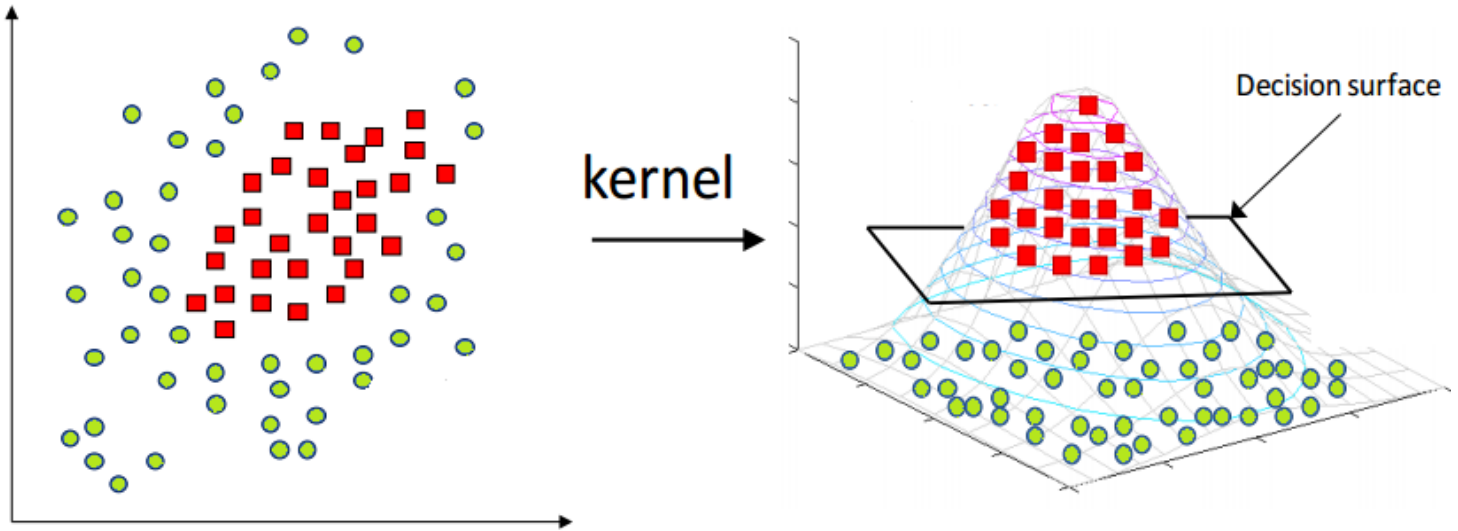
We need to add a penalty for too large slack variable

$$\min_{\mathbf{w},b}\left(\frac{\| \mathbf{w} \|^2}{2} + \frac{C}{N}\sum_{i=1}^{N}\xi_i\right)$$

C > 0 weight the influence of the penalty term

Find trade off between maximizing margin and minimizing the classification errors

SVM: Dealing with non-linear classification



SVM Kernel Functions

- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid
- …

Properties of SVM

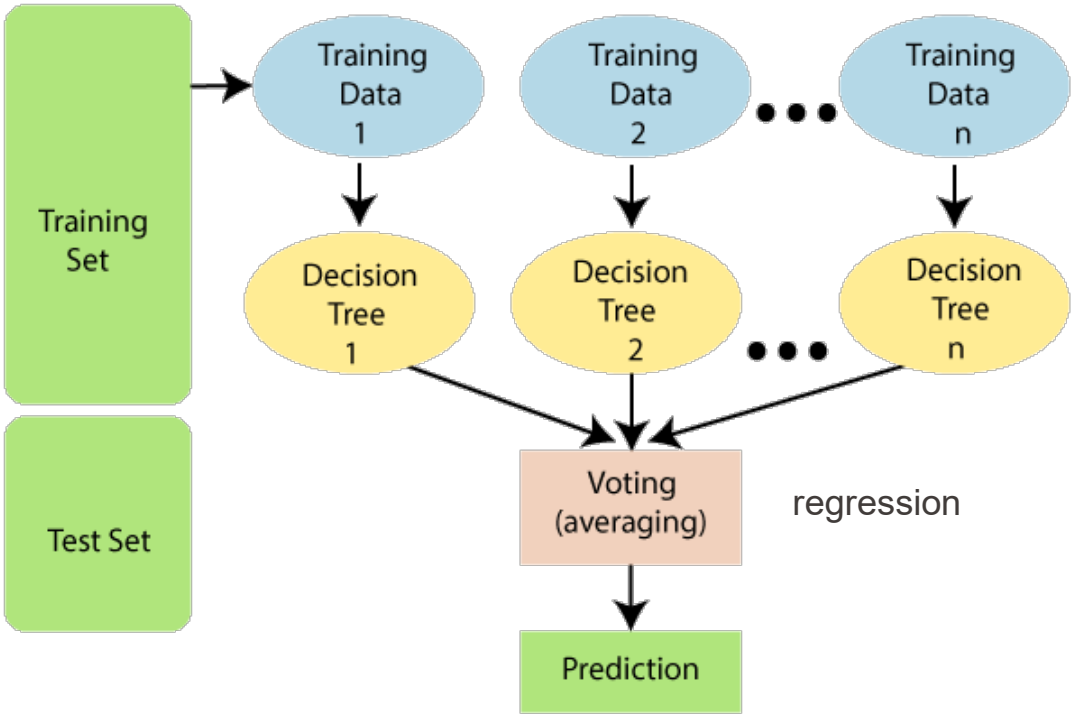
- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
  - only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
  - complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection

Recap: Random forest

Random forest

- Random forest is used for both classification and regression tasks.
- It is an ensemble learning method that combines multiple decision trees to make predictions.
- The name "random forest" comes from the fact that the algorithm creates a "forest" of decision trees that are constructed using a random subset of the training data and a random subset of the features.
- Decision tree:
  - goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features
  - follows a set of if-else conditions to visualize the data and classify it according to the conditions

Basic principle of random forest



regression

Voting (majority)

Classification

Source: www.towardsdatascience.com

Advantages of random forest

- **Diversity:** Not all attributes/variables/features are considered while making an individual tree; each tree is different.
- **Immune to the curse of dimensionality:** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization:** Each tree is created independently out of different data and attributes.
- **Stability/Robustness:** Stability/Robustness arises because the result is based on majority voting/ averaging.
- **Interpretability:** Easier to interpret the single decision trees.

Disadvantages of Random Forest

- **Complexity and Interpretability**
  - **Black Box Nature:**
    Unlike individual decision trees, which are easy to interpret, random forests consist of numerous trees. This ensemble approach makes the overall model less transparent, hindering the ability to understand how specific features influence the final prediction.
  - **Feature Importance Ambiguity:**
    Although random forests provide feature importance scores, these can sometimes be misleading, especially when features are highly correlated. It becomes challenging to discern the true impact of each feature on the model's decisions.
- **Overfitting**
  - **Potential Overfitting:**
    Although random forests are generally robust against overfitting due to the averaging of multiple trees, they can still overfit if the number of trees is excessively large or if individual trees are too deep, especially in the presence of noisy data.
- **Handling Imbalanced Data**
  - **Bias Toward Majority Class:**
    In classification tasks with imbalanced datasets, random forests may become biased toward the majority class, leading to poor performance on minority classes.
- **Parameter Tuning**
  - **Hyperparameter Complexity:**
    Random forests have several hyperparameters (e.g., number of trees, maximum tree depth, number of features to consider at each split) that require careful tuning to achieve optimal performance. This tuning process can be time-consuming and computationally demanding.
- **Feature Scaling and Engineering**
  - **Dependence on Feature Quality:**
    While random forests do not require feature scaling, the quality of the input features significantly impacts model performance. Effective feature engineering remains essential to ensure that the model captures the underlying patterns in the data.

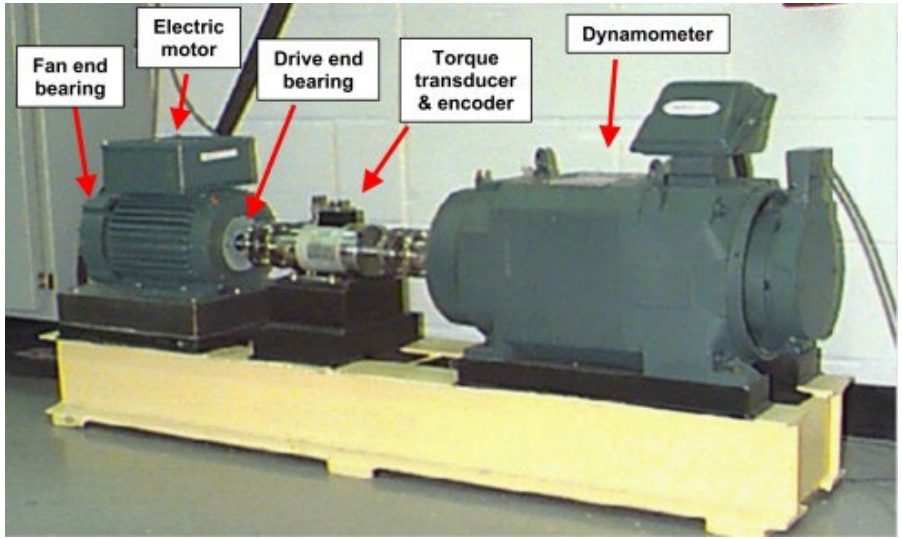
Summary

Supervised Learning Algorithms

- Logistic Regression
  - Advantages: Outputs probabilistic interpretations, can be regularized to avoid overfitting.
  - Disadvantages: Not suitable for complex relationships with non-linear boundaries.
- Decision Trees
  - Advantages: Easy to interpret, handles both numerical and categorical data, requires little data preprocessing.
  - Disadvantages: Prone to overfitting, sensitive to small data changes.
- Random Forests
  - Advantages: Reduces overfitting through ensemble learning, handles large datasets well.
  - Disadvantages: Computationally intensive, less interpretable than single decision trees.
- Support Vector Machines (SVM)
  - Advantages: Effective in high-dimensional spaces, robust against overfitting.
  - Disadvantages: Memory-intensive, tricky to tune, doesn’t scale well with large datasets.
- K-Nearest Neighbors (KNN)
  - Advantages: Simple to understand and implement, no training phase.
  - Disadvantages: High computational cost, sensitive to irrelevant features and the scale of data.

Fault diagnostics: Example

Fault diagnosis (CWRU example)



Three Fault Types (B, IR, OR)
Three different severity levels
→ Ten Classes
→ One healthy class, Nine fault classes

12k Drive End Fault Data
48k Drive End Fault Data
Fan End Fault Data
Benchmark data

Class	0	1	2	3	4	5	6	7	8	9
Severity [mils]	-	7	7	7	14	14	14	21	21	21
Type	N	B	IR	OR	B	IR	OR	B	IR	OR

http://csegroups.case.edu/bearingdatacenter/home

Feature set

Time	Time	Frequency	Entropy	Marginal spectrum energy
Mean T1	Std T6	CF F1	Power Spectrum H1	IMF1 E1
RMS T2	Shape factor T7	MSF F2	Power Spectrum H2	IMF2 E2
Kurtosis T3	Peaking factor T8	RMSF F3	Singularity Spectrum H3	IMF3 E3
Peak-to-peak T4	Pulse factor T9	VF F4	Singularity Spectrum H4	IMF4 E4
Var T5	Margin factor T10	RVF F5	Wavelet Energy H5	IMF5 E5
			Bispectral entropy H6	IMF6 E6

Zhang, Xiao, Boyang Zhao, and Yun Lin. "Machine learning based bearing fault diagnosis using the case western reserve university data: a review." *IEEE Access* 9 (2021): 155598-155608.

Relief

- Instance: A single data point in the dataset.
- Hit: The nearest instance of the same class as the target instance.
- Miss: The nearest instance of a different class than the target instance.
- Feature Weight (W_i): A score representing the relevance of feature i for classification.

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0 \ ; value(A, I_1) = value(A, I_2) \\ 1 \ ; otherwise \end{cases}$$

for nominal attributes.

$$\text{diff}(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{max(A) - min(A)}$$

for numerical attributes.

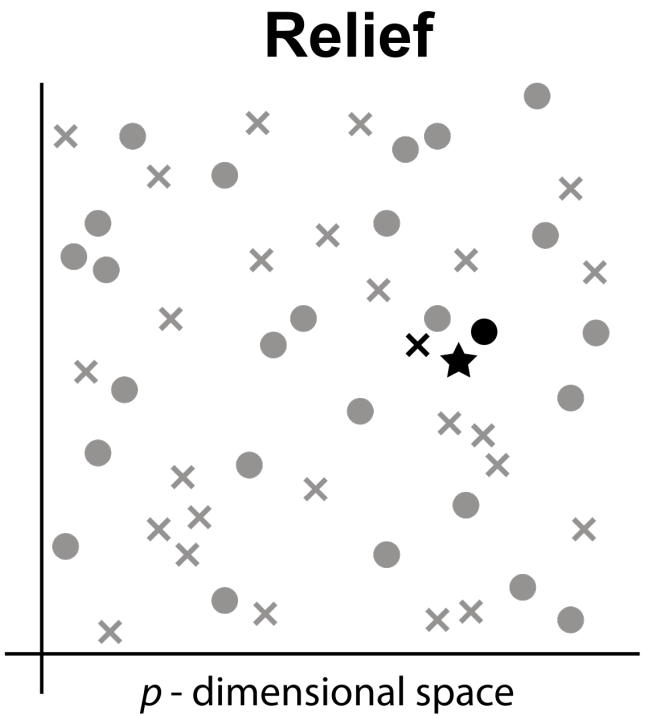
Step-by-Step Process

- **Initialize Feature Weights:**
  - Set the weight $W_i = 0$ for all features *i*
  - **Iterate Through a Sample of Instances:**
  - For each instance R in a randomly selected subset of the dataset:

  a. **Find Nearest Hit and Miss:**
    - **Nearest Hit (H):** The closest instance to R that belongs to the same class.
    - **Nearest Miss (M):** The closest instance to R that belongs to a different class.

  b. **Update Feature Weights:**
    - For each feature i: $$W_i = W_i - \text{diff}(R_i, H_i) + \text{diff}(R_i, M_i)$$
    - **Difference Function (diff):** Measures the difference between feature values (binary, absolute difference, squared difference)
- **Normalize Feature Weights:**
  - After processing all sampled instances, normalize the weights W_i to ensure they are comparable across features.
- **Rank Features:**
  - Features with higher weights are considered more relevant for classification and are ranked accordingly.

Relief feature selection



★ Target Instance (e.g. Class '◯')

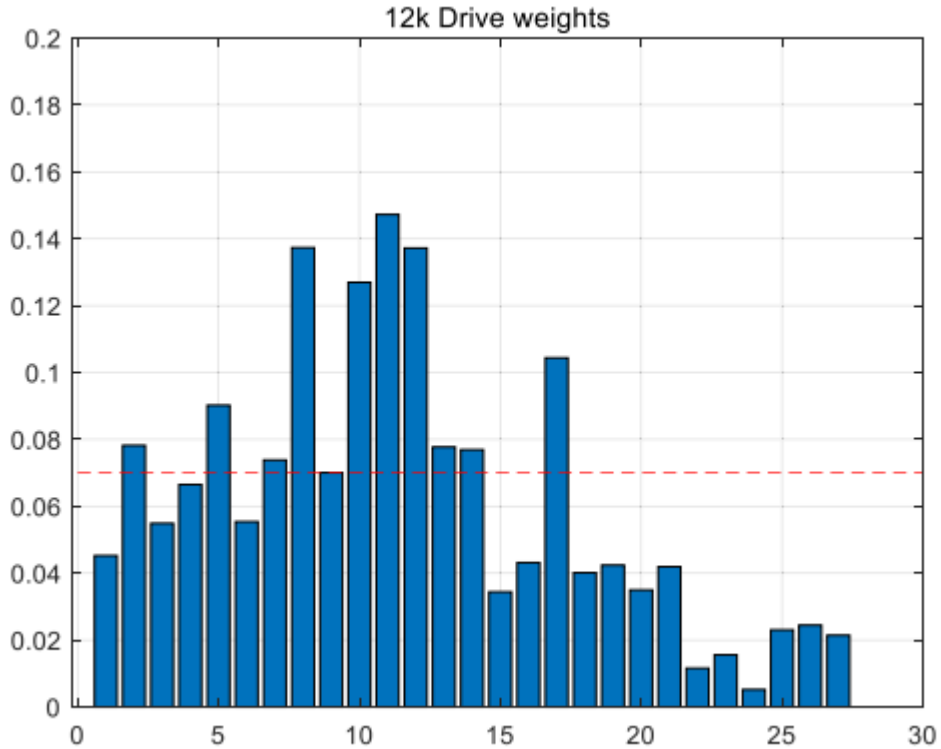
● Instance with Class '◯' (Zero instance weight)

× Instance with Class 'X' (Zero instance weight)

● Instance with Class '◯' Nearest Neighbor(s) (Near)

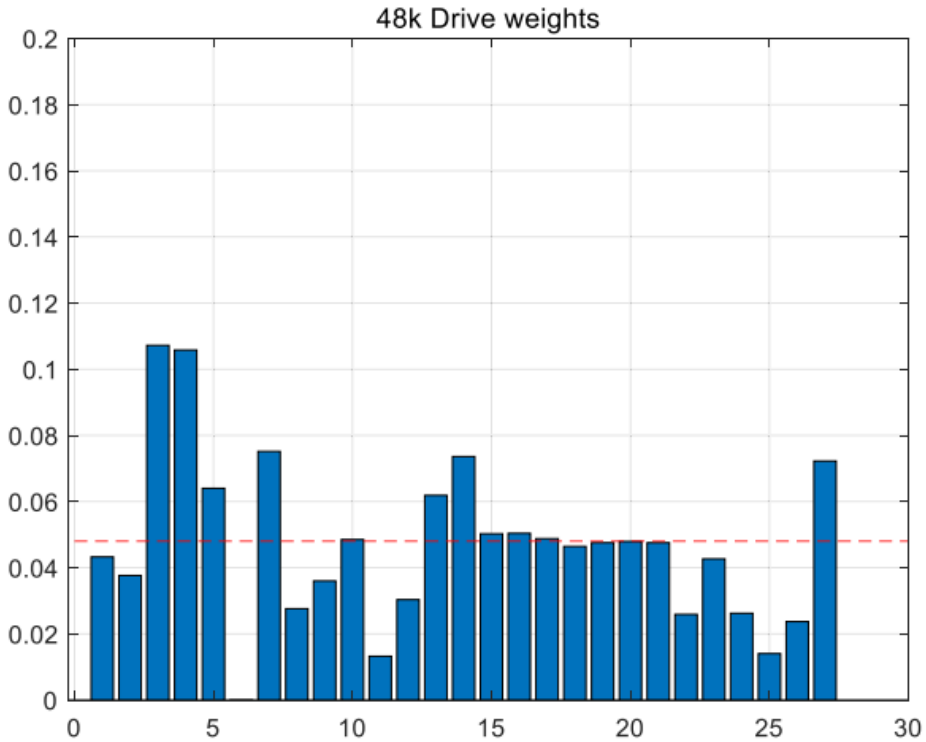
× Instance with Class 'X' Nearest Neighbor(s) (Near)

Feature selection (1/3)



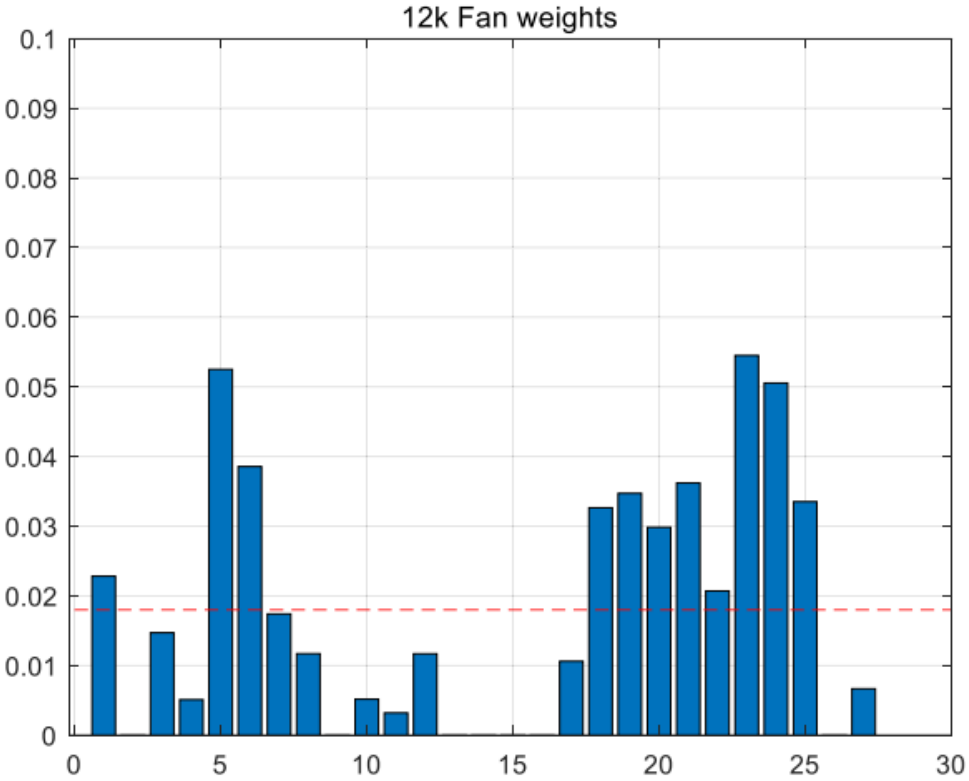
Zhang, Xiao, Boyang Zhao, and Yun Lin. "Machine learning based bearing fault diagnosis using the case western reserve university data: a review." *IEEE Access* 9 (2021): 155598-155608.

Feature selection (2/3)



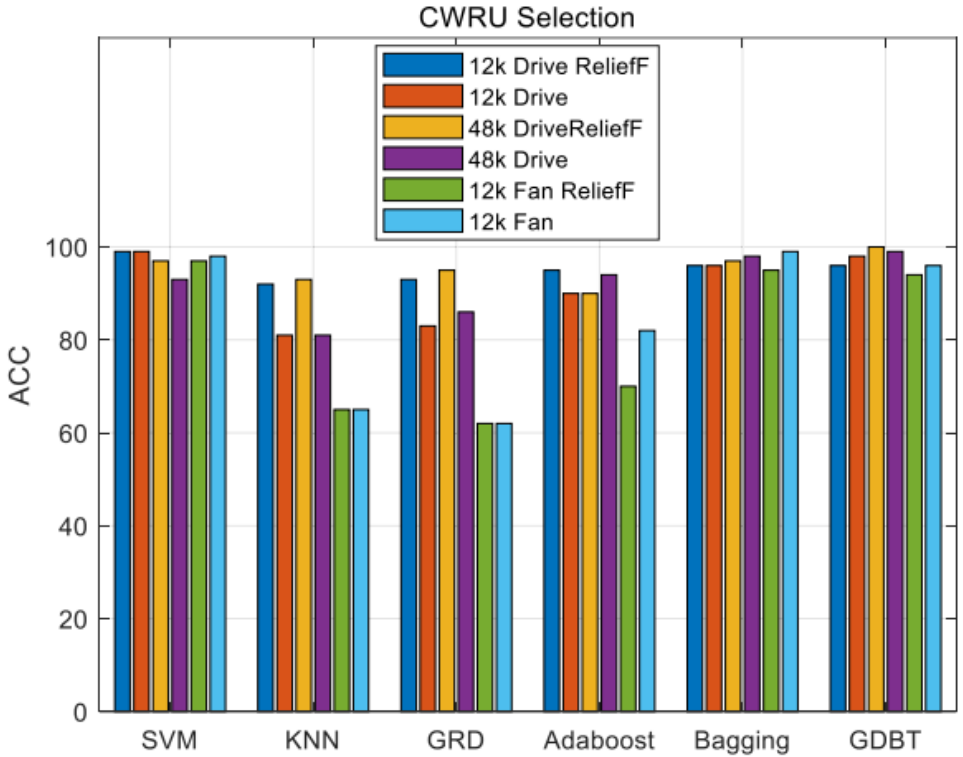
Zhang, Xiao, Boyang Zhao, and Yun Lin. "Machine learning based bearing fault diagnosis using the case western reserve university data: a review." *IEEE Access* 9 (2021): 155598-155608.

Feature selection (3/3)



Zhang, Xiao, Boyang Zhao, and Yun Lin. "Machine learning based bearing fault diagnosis using the case western reserve university data: a review." *IEEE Access* 9 (2021): 155598-155608.

Performance of different ML algorithms



Zhang, Xiao, Boyang Zhao, and Yun Lin. "Machine learning based bearing fault diagnosis using the case western reserve university data: a review." *IEEE Access* 9 (2021): 155598-155608.