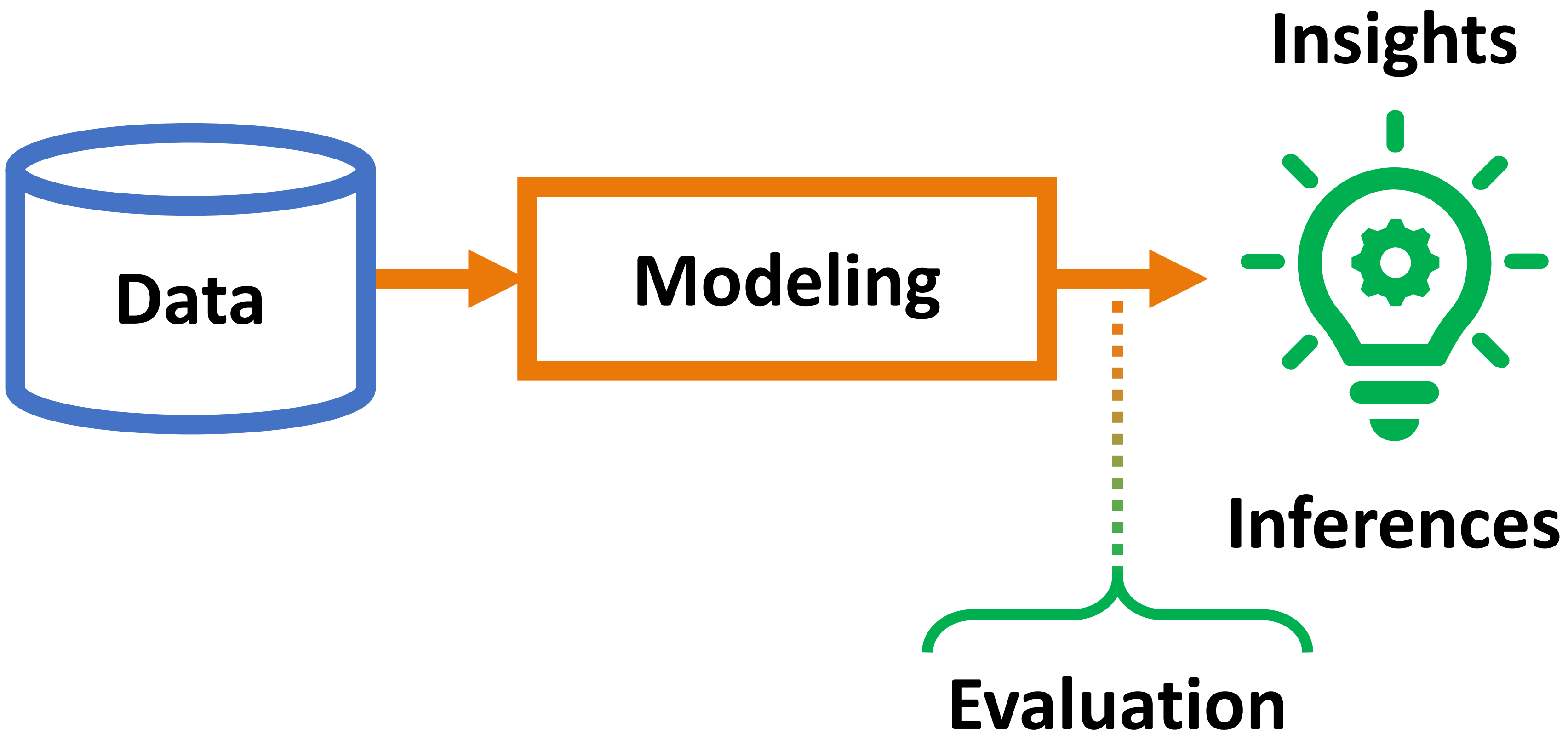
Data Processing

8A

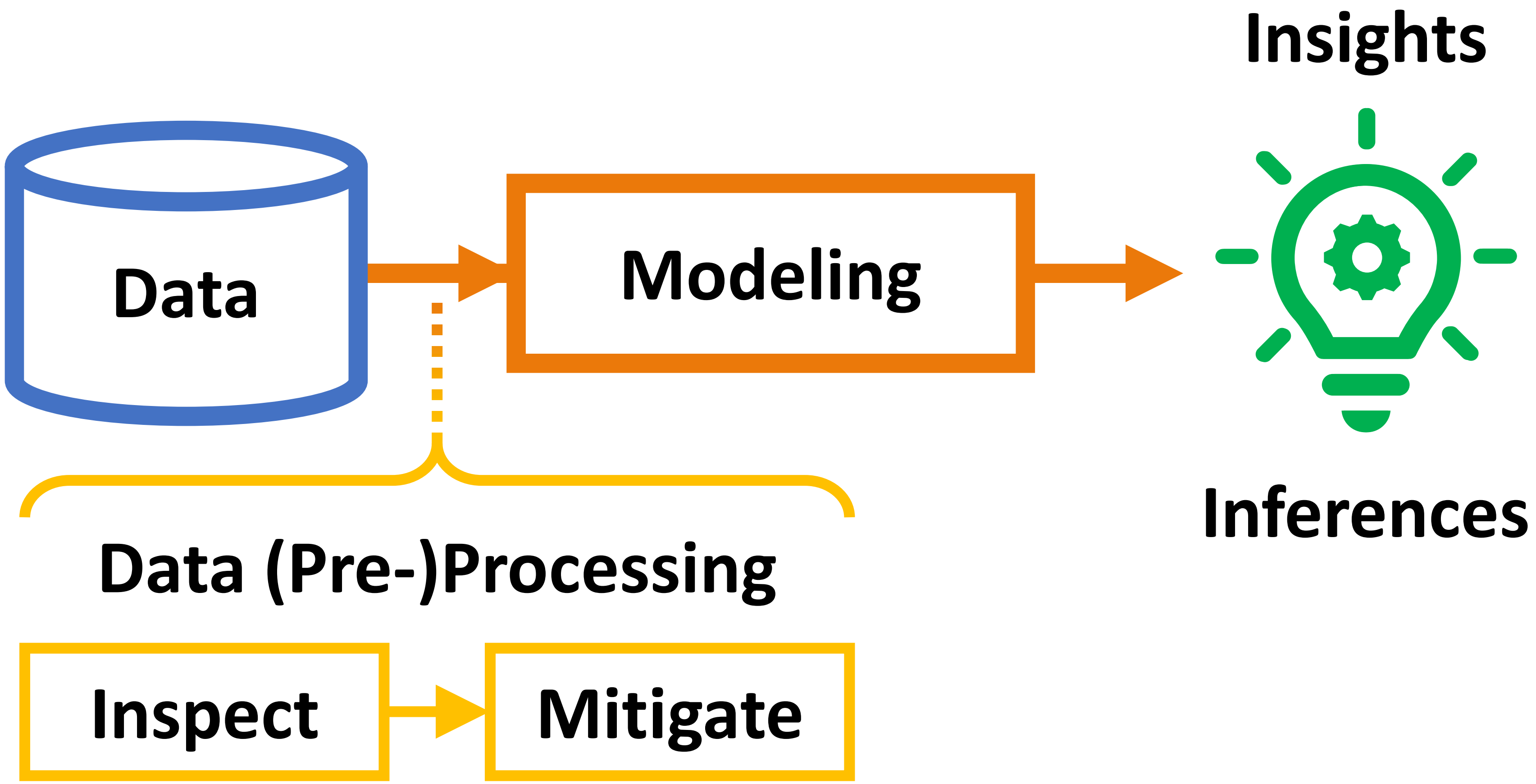
CS 3244
Machine Learning

NUS National University of Singapore | Computing

Machine Learning Pipeline



Machine Learning Pipeline



W08 Pre-Lecture Task

Read

1. Discover Feature Engineering, How to Engineer Features and How to Get Good at It by Jason Brownlee
2. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset by Jason Brownlee

Task

1. Identify cases of **bad data** in machine learning
2. Propose **mitigation strategies**

   Tip: you can your own projects too; you don't have to be correct
3. Post a 1–2 sentence answer to the topic in your tutorial group: #tg-xx

1. Identify cases of **bad data** in machine learning
2. Propose **mitigation strategies**

1 Erroneous data

An example of bad data is that the data is unclean (having a lot of noise). To reduce this, outliers have to be removed so that it does not affect the model's performance.

Bad formatting: Spend more time filtering and reformatting the data

2 Irrelevant data [W08b]

1. Too many irrelevant features can affect the learning of the model, affecting its potential performance on a given problem statement
2. Feature selection can be done to ensure that we only allow the model to learn relevant features

3 Imbalanced data

1. imbalance data: Porto Seguro's Safe Driver Prediction from kaggle (only 3.78% of the data (claim) was filed for the policy holder) https://datascience.foundation/sciencewhitepaper/understanding-imbalanced-datasets-and-techniques-for-handling-them
2. mitigation strategies: under-sampling the data by decreasing the instances of majority class so the count is decreased to match the minority class

4 Missing data

1. When dealing with stocks data, it is often the case that there are missing data in some of the days because it is supposed to be tracked daily, and sometimes it is hard to do so.
2. To impute the missing data, we could just copy the previous day's price, because it is logical since the day to day movement of stock prices is minimal, compared to, say a longer term. Or another way is we could use the previous 5 days average because of similar reason.

5 Too many features (high dimensionality)

High dimensionality of data is another commonly faced problem. Some popular tactics to perform feature selection (reduce dimensionality) are lasso regression and principal component analysis.

Week 08: Learning Outcomes

Data **Issues**

1. Linear Separability
2. Curse of Dimensionality
3. Imbalanced Data

Issue Template

1. **What** is the issue?
2. **Why** is it a problem?
3. **When** would it happen?
4. **How** to **check** for it?
5. **How** to **mitigate** it?

For each issue, which of the following techniques can:

a) **Check** for the issue?

b) **Mitigate** the issue?

Issue	a) Check	b) Mitigate
1) Linear Separability		
2) Curse of Dimensionality		
3) Imbalanced Data		

1. Feature Engineering
2. Feature Extraction (extract new features)
3. Information Gain
4. Linear Discriminant Analysis (LDA)
5. Principle Components Analysis (PCA)
6. SMOTE
7. Support Vector Machine
8. Visualize Histogram
9. Visualize Scatterplot

Emote (react) in Slack #lecture channel <u>one or more options</u> (MRQ) for each issue

Exercise W08b.1 Solution

For each issue, which of the following techniques can:

a) **Check** for the issue?

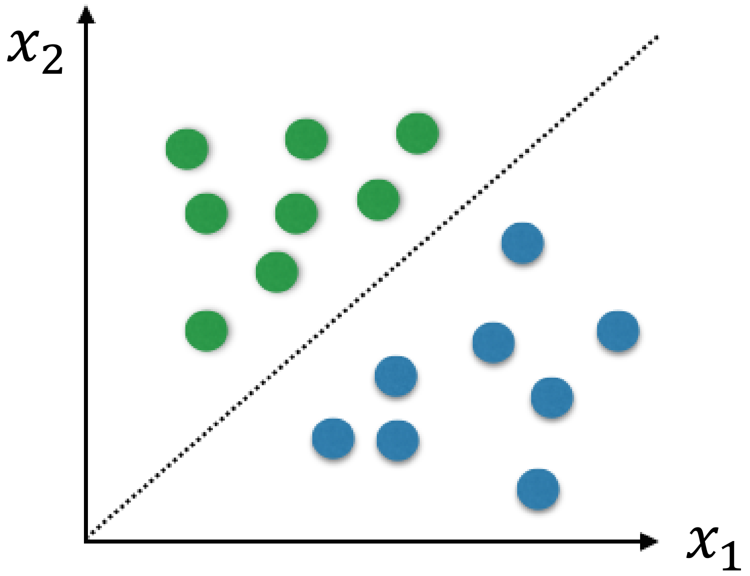
b) **Mitigate** the issue?

Issue	a) Check	b) Mitigate
1) Linear Separability	9 Visualize Scatterplot 7 Support Vector Machine 4 5 Check Basis Vectors (with LDA, PCA)	1 Feature Engineering 2 Feature Extraction 4 5 Matrix Factorization (with LDA, PCA)
2) Curse of Dimensionality	8 Visualize Histogram (of distances)	3 Feature Selection (using Information Gain) 4 5 Dimensionality Reduction (with LDA, PCA)
3) Imbalanced Data	8 Visualize Histogram	6 SMOTE

Linear Separability

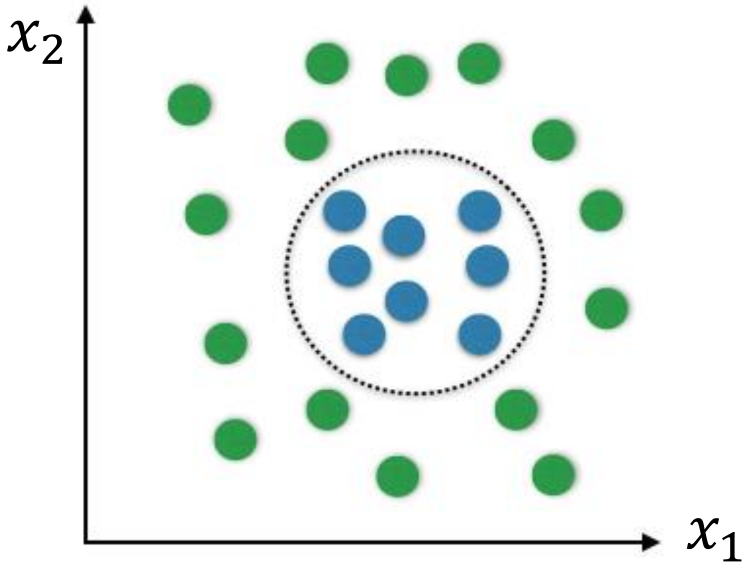
Linearly Separable?

Yes



$$\boldsymbol{x} = \binom{x_1}{x_2}$$

Not without data processing



How to make linearly separable?

$$\boldsymbol{x}' = \begin{pmatrix} (x_1 - \bar{x}_1)^2 \\ (x_2 - \bar{x}_2)^2 \end{pmatrix} = (\boldsymbol{x} - \overline{\boldsymbol{x}})^\top(\boldsymbol{x} - \overline{\boldsymbol{x}})$$

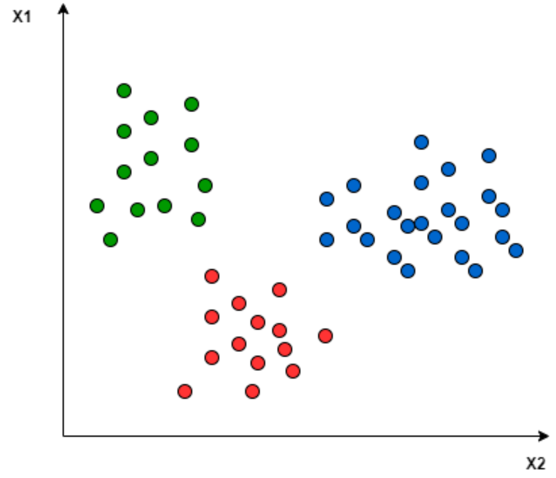
$\boxed{\text{Feature Engineering!}}$

Image Credit: [Sebastian Raschka]

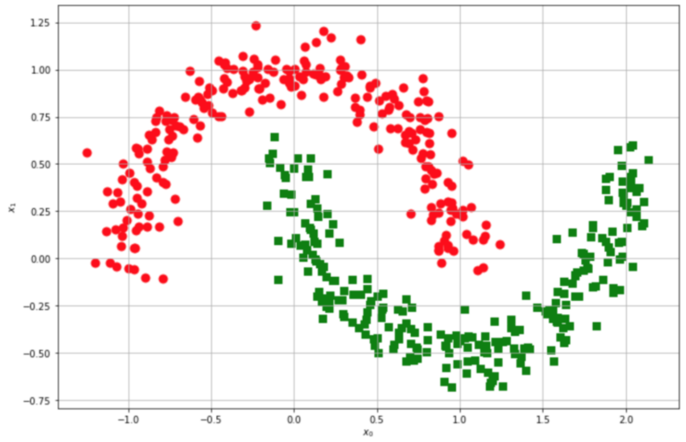
Which of the following is:

1. Is Linearly separable (📏:straight_ruler:)?
2. Can it be made Linearly Separable (➰:curly_loop:)? How? (Write in **thread**)

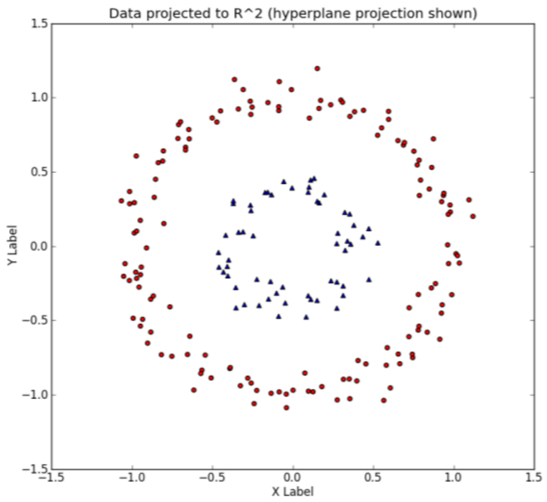
a)



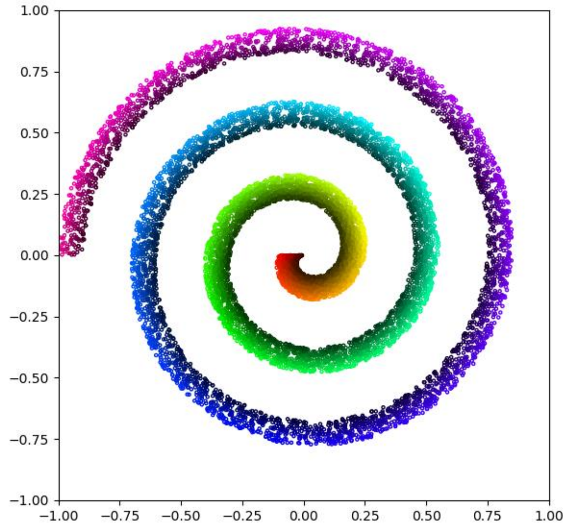
b)



c)



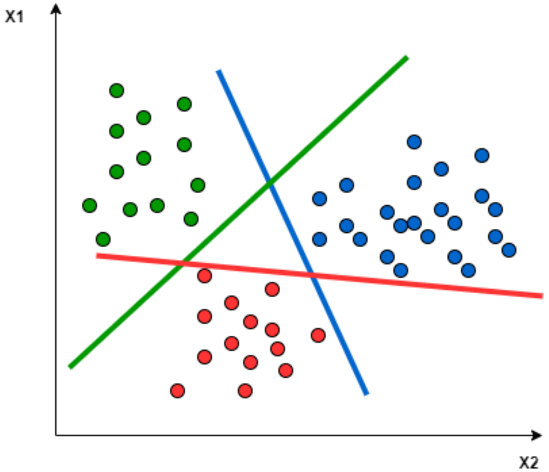
d)



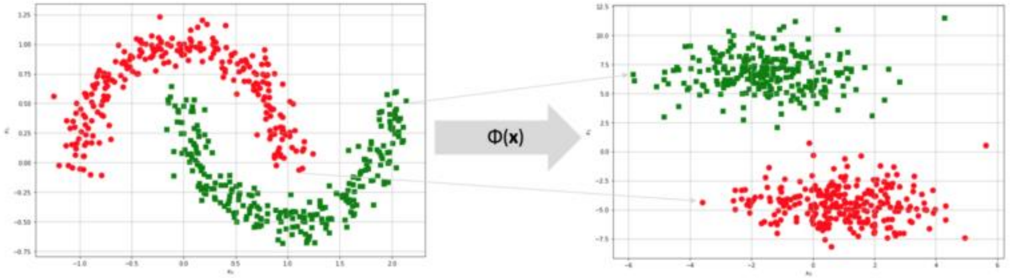
Which of the following is:

1. Is Linearly separable (📏:straight_ruler:)?
2. Can it be made Linearly Separable (➰:curly_loop:)? How? (Write in **thread**)

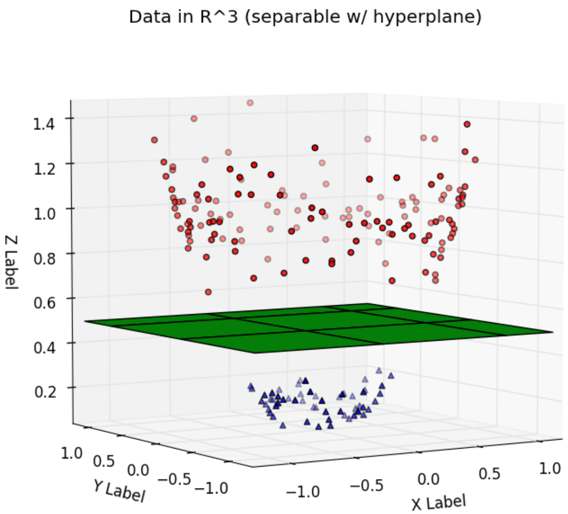
a) ➰📏



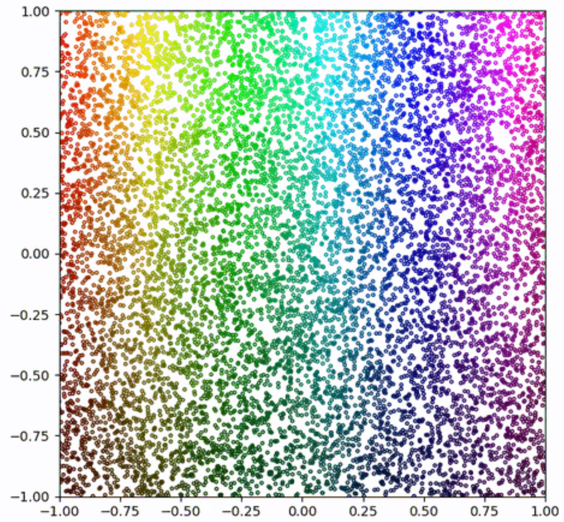
b) ➰



c) ➰



d) ➰



Issue: Linear Separability

1. **What** is the issue?
    1. Many models assume that data features are **linearly** separable
    2. Does your data satisfy this **assumption**?
2. **Why** is it a problem?
    1. Irrelevant features will be uninformative to train the model to discriminate between prediction labels
    2. If features are not linearly separable, you cannot learn a good **linear model**
    3. Need to use more complex models
3. **When** would it happen?
    1. Most of the time, for “fresh” unprocessed data.
    2. Especially for unstructured (non-tabular) data, e.g., images, time, text

Issue: Linear Separability

4. **How** to **check** for it?
   1. Visualize
      - 2D: **Scatterplot** of x_1 by x_2 graph
      - >2D: **Scatterplot Matrix**
      - 500 dimensions?

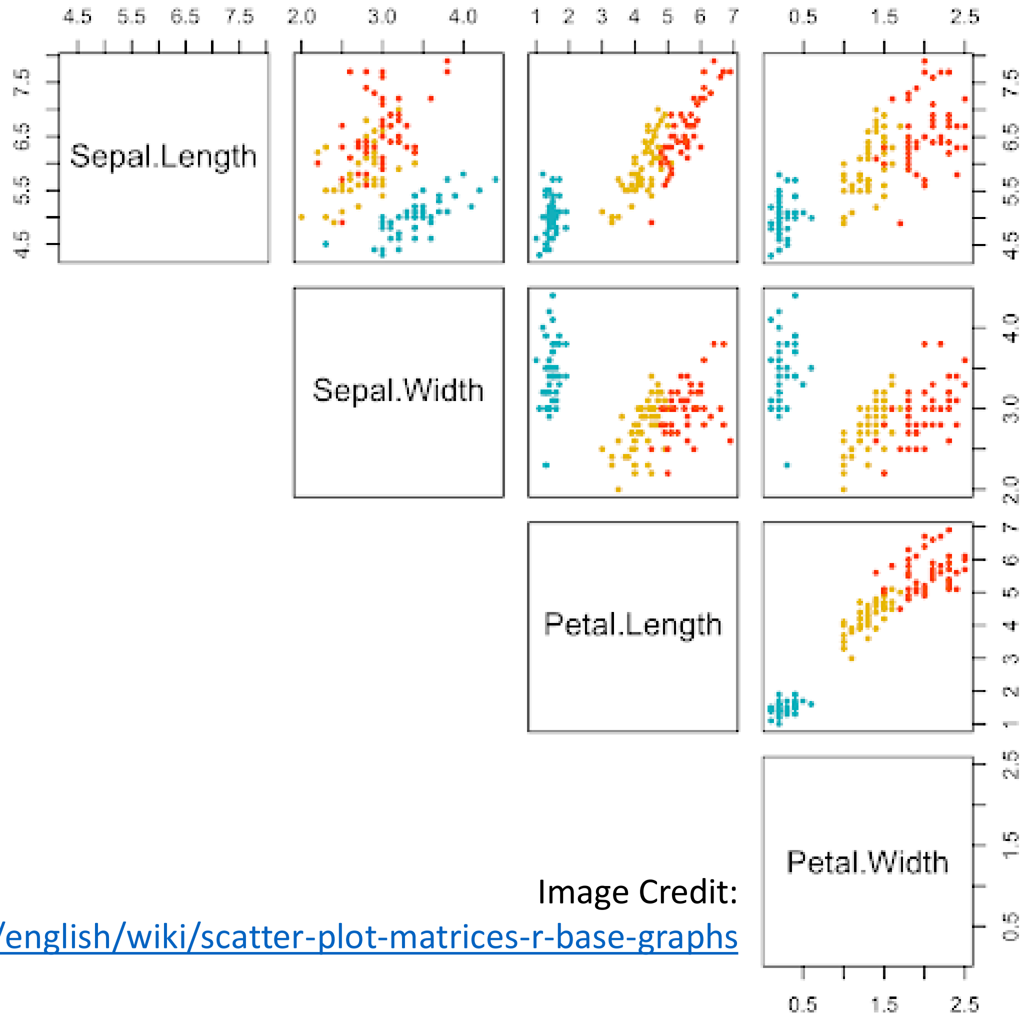


Image Credit:
http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs

Issue: Linear Separability

4. **How** to **check** for it?
   1. Visualize
   2. Computational metrics
      1. Linear SVM [W04b]

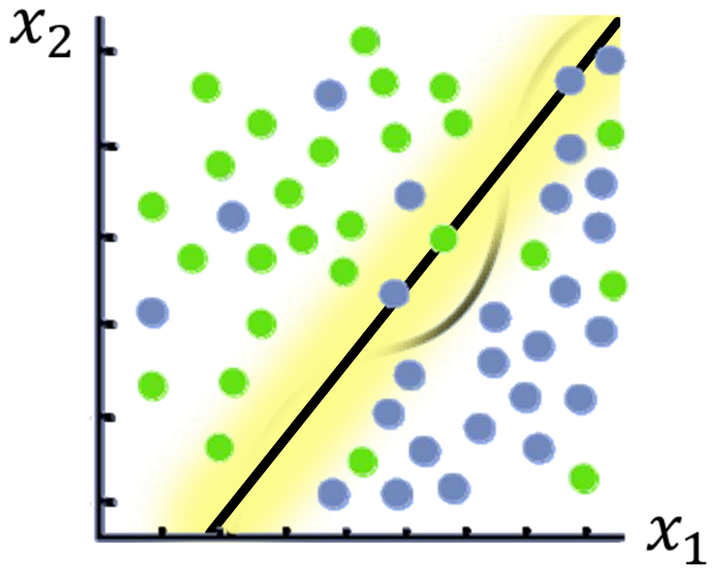


Image Credit:
http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs

Cost Function w Slack Variables

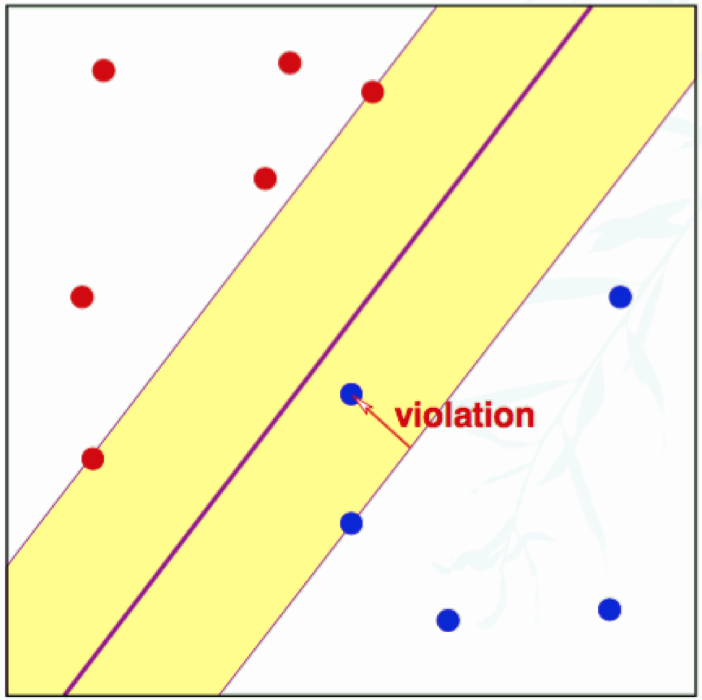
Margin violation: $y^{(*)}\left(\boldsymbol{\theta}^{\top}\mathbf{x}^{(*)} + b\right) \geq 1$ fails

Quantify this:

$y^{(*)}\left(\boldsymbol{\theta}^{\top}\mathbf{x}^{(*)} + b\right) \geq 1 - \xi^{(*)}$
where $\xi^{(*)} \geq 0$

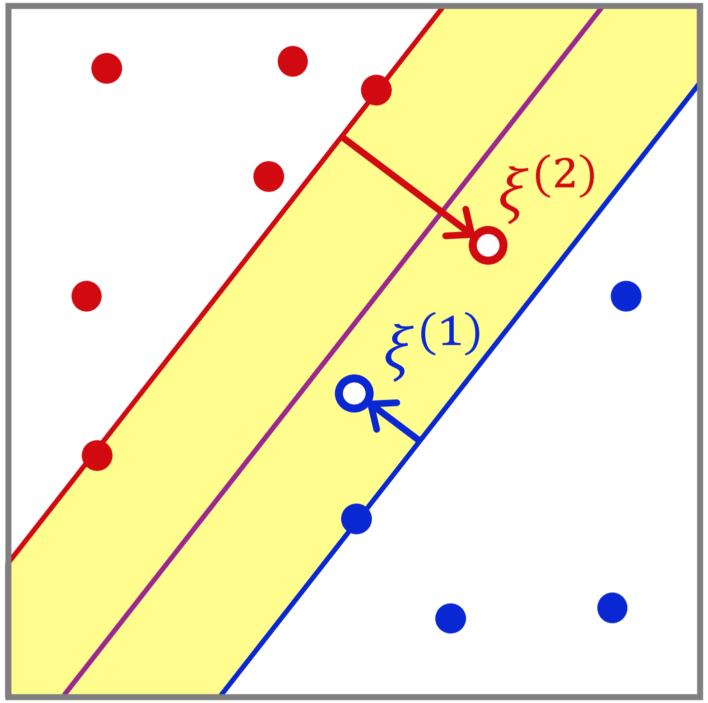
Slack variable: Soft error on $(x^{(*)}, y^{(*)})$

Total violation: $\sum_{j=1}^{m} \xi^{(j)}$



Testing Linear Separability with Linear Soft-Margin SVM

- Each $\xi^{(j)}$ is the **distance** that the misclassified point j is from its correct margin
- Total violation: $\sum_{j=1}^{m} \xi^{(j)}$
- Calculating the **total violation** indicates how **linearly separable** the data is in terms of its features
- Higher violation => Less linearly separable



Further Reading: https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe

Issue: Linear Separability

4. **How** to **check** for it?
   1. Visualize
   2. Computational metrics
      1. Linear SVM [W04b]
      2. Reduce dimensions (LDA, PCA), then visualize separability (for separation by "diagonal planes")
      3. Others: Linear programming, Convex Hulls

Only these are **examinable**

Issue: Linear Separability

5. **How** to **mitigate** it?
   - Find useful features
     - Feature extraction (collect new features of your data)
   - Transformation of features
     - Feature Engineering (e.g., $\boldsymbol{x} \rightarrow \boldsymbol{x}^{2}$)
     - Change Basis Vectors (e.g., PCA, LDA)
     - Kernel trick (e.g., for kernel SVM [W04b])
     - Feature Learning (e.g., Neural Networks [W09/10])

Issue: Linear Separability

5. How to **mitigate** it?

- Find useful features
  - Feature extraction (collect new features of your data)
- Transformation of features
  - Feature Engineering (e.g., $\boldsymbol{x} \rightarrow \boldsymbol{x}^{\boldsymbol{2}}$)
  - **Change Basis Vectors (e.g., PCA, LDA)**
  - Kernel trick (e.g., for kernel SVM [W04b])
  - Feature Learning (e.g., Neural Networks [W09/10])

Vector Spaces and Basis Vectors

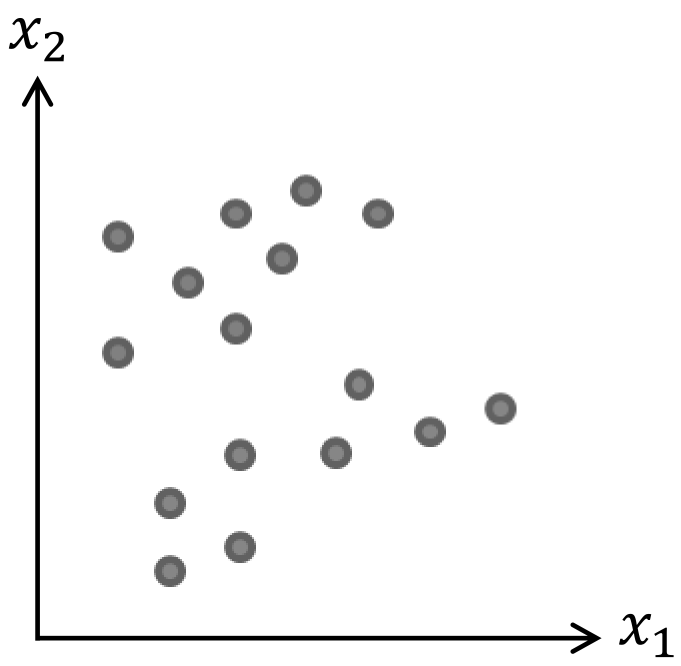


Image Credit: https://nirpyresearch.com/classification-nir-spectra-linear-discriminant-analysis-python/

Vector Spaces and Basis Vectors

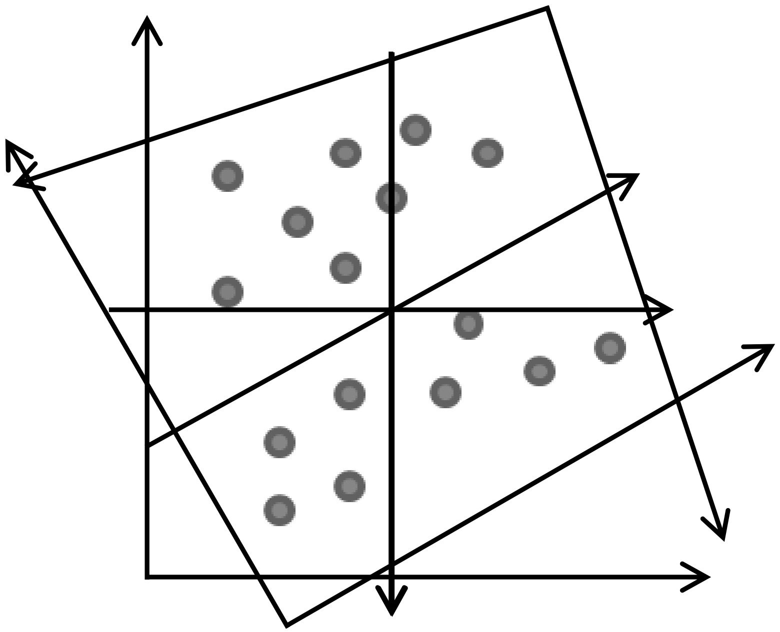
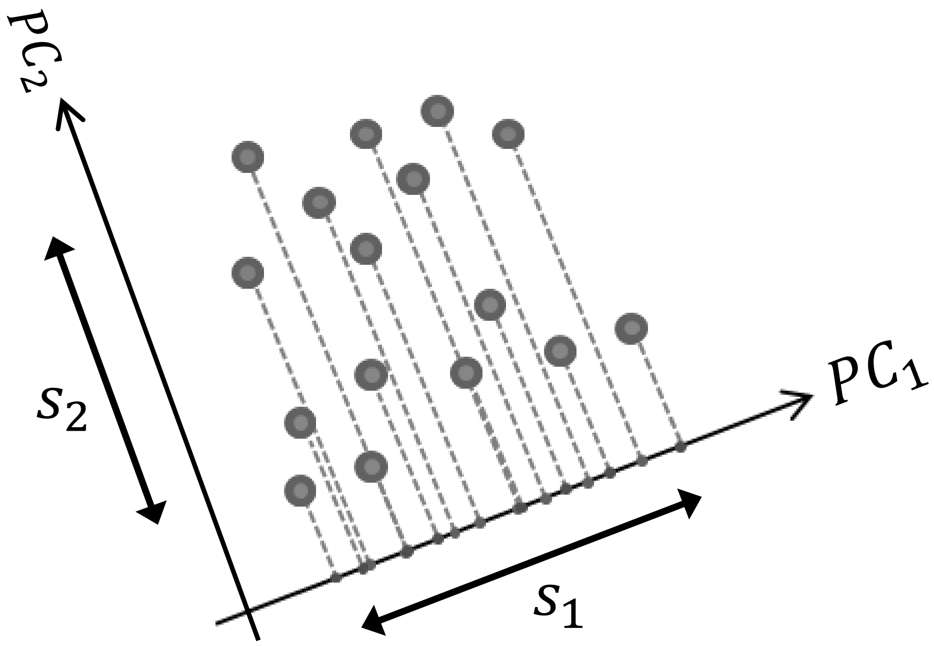


Image Credit: https://nirpyresearch.com/classification-nir-spectra-linear-discriminant-analysis-python/

Principal Component Analysis (PCA)

What axis best **describes the variation** in the data?



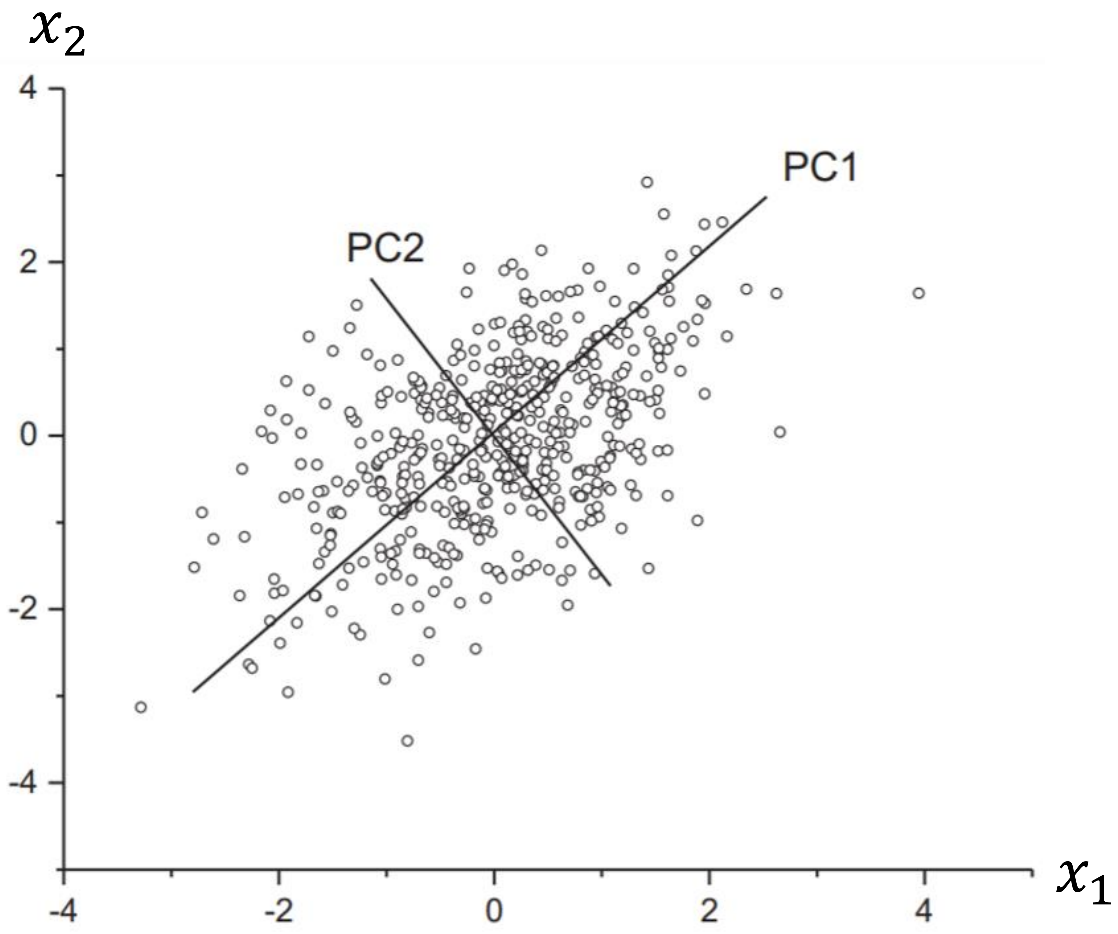
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \xrightarrow{\text{PCA Projection}} \begin{pmatrix} PC_1 \\ PC_2 \end{pmatrix}$$

$$s_1 > s_2$$

$$\begin{pmatrix} PC_1 \\ PC_2 \end{pmatrix} \xrightarrow{\text{Reduce Dimensions}} (PC_1)$$

Further reading:
[PCA 1: the basics - simply explained](#) by [TileStats](#),
[StatQuest: Principal Component Analysis (PCA), Step-by-Step](#) by [StatQuest with Josh Starmer](#)
Image Credit: https://nirpyresearch.com/classification-nir-spectra-linear-discriminant-analysis-python/

Principal Component Analysis (PCA)



$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \xrightarrow[]{\substack{\text{PCA} \\ \text{Projection}}} \begin{pmatrix} PC_1 \\ PC_2 \end{pmatrix}$$

$$s_1 > s_2$$

$$\begin{pmatrix} PC_1 \\ PC_2 \end{pmatrix} \xrightarrow[]{\substack{\text{Reduce} \\ \text{Dimensions}}} (PC_1)$$

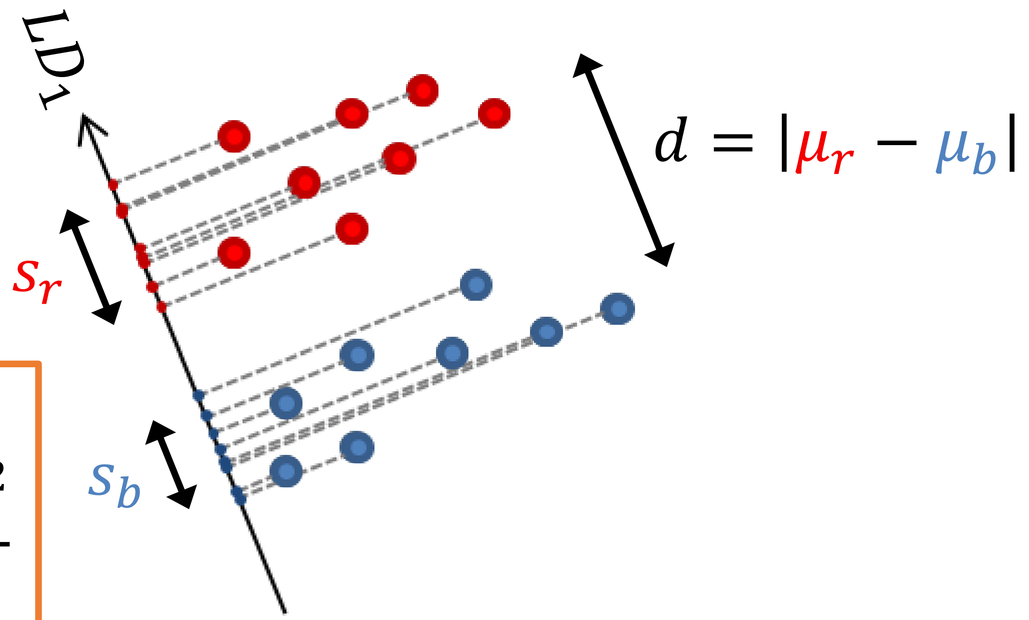
Image Credit: https://ekamperi.github.io/mathematics/2021/02/23/pca-limitations.html

Linear Discriminant Analysis (LDA)

What axis best **distinguishes classes** in the data?

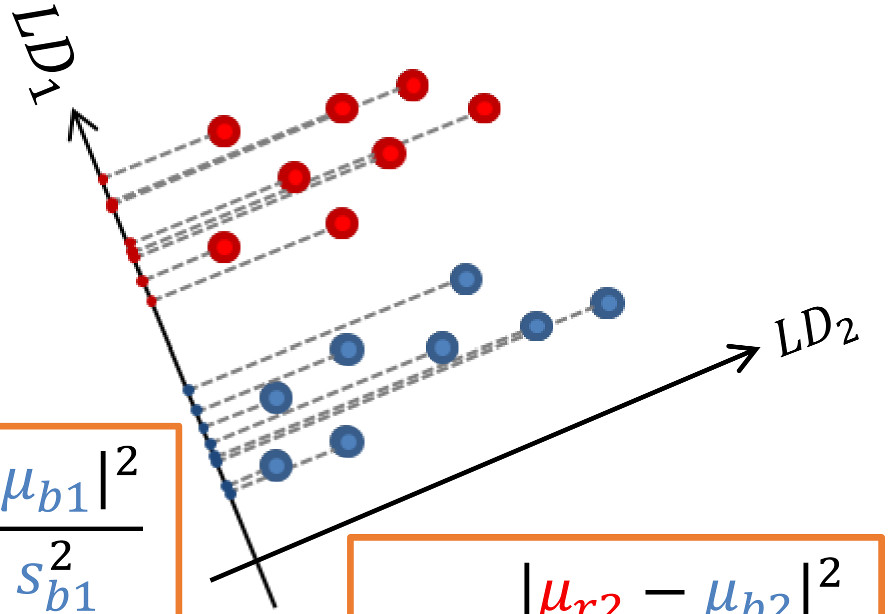
Maximize

$$F = \frac{|\mu_r - \mu_b|^2}{s_r^2 + s_b^2}$$



Further reading: Linear discriminant analysis (LDA) - simply explained by TileStats,
StatQuest: Linear Discriminant Analysis (LDA) clearly explained by StatQuest with Josh Starmer
Image Credit: https://nirpyresearch.com/classification-nir-spectra-linear-discriminant-analysis-python/

Linear Discriminant Analysis (LDA)



$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \xrightarrow[\quad]{\text{LDA Projection}} \begin{pmatrix} LD_1 \\ LD_2 \end{pmatrix}$$

$$F_1 > F_2$$

$$F_1 = \frac{|\mu_{r1} - \mu_{b1}|^2}{s_{r1}^2 + s_{b1}^2}$$

$$F_2 = \frac{|\mu_{r2} - \mu_{b2}|^2}{s_{r2}^2 + s_{b2}^2}$$

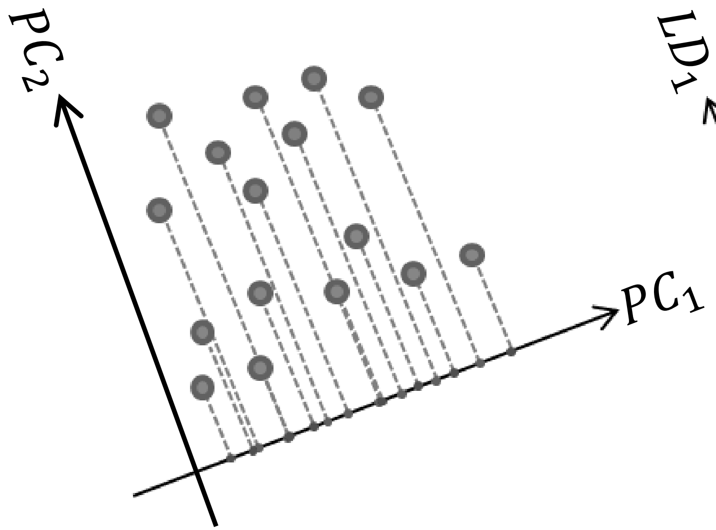
$$\begin{pmatrix} LD_1 \\ LD_2 \end{pmatrix} \xrightarrow[\quad]{\text{Reduce Dimensions}} (LD_1)$$

Further reading: Linear discriminant analysis (LDA) - simply explained by TileStats,
StatQuest: Linear Discriminant Analysis (LDA) clearly explained by StatQuest with Josh Starmer
Image Credit: https://nirpyresearch.com/classification-nir-spectra-linear-discriminant-analysis-python/

PCA and LDA for **Dimensionality Reduction**

PCA

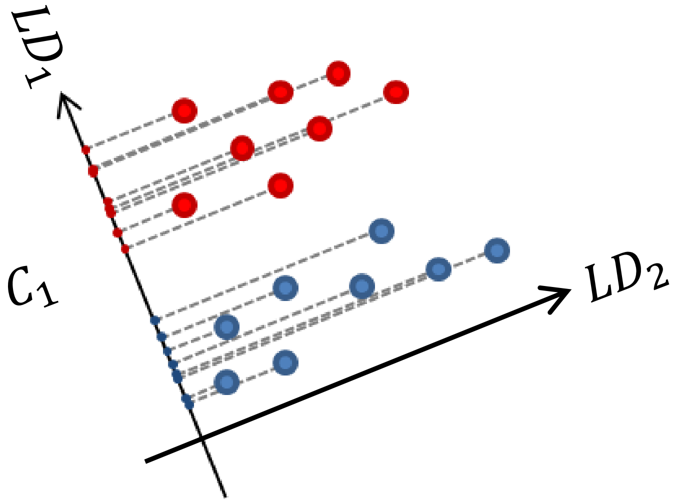
Maximize Data Variance



Good for
for <u>supervised learning</u>
and <u>unsupervised learning</u>

LDA

Maximize Class Separation



Better for
for <u>supervised classification</u>.
Not for <u>unsupervised learning</u>

Steps

- All axes are orthogonal (independent)

1. **Identify** basis vectors
2. **Rank** basis vectors by importance
3. **Truncate** selection of basis vectors
   - Keeps more important features
   - Performs dimensionality reduction

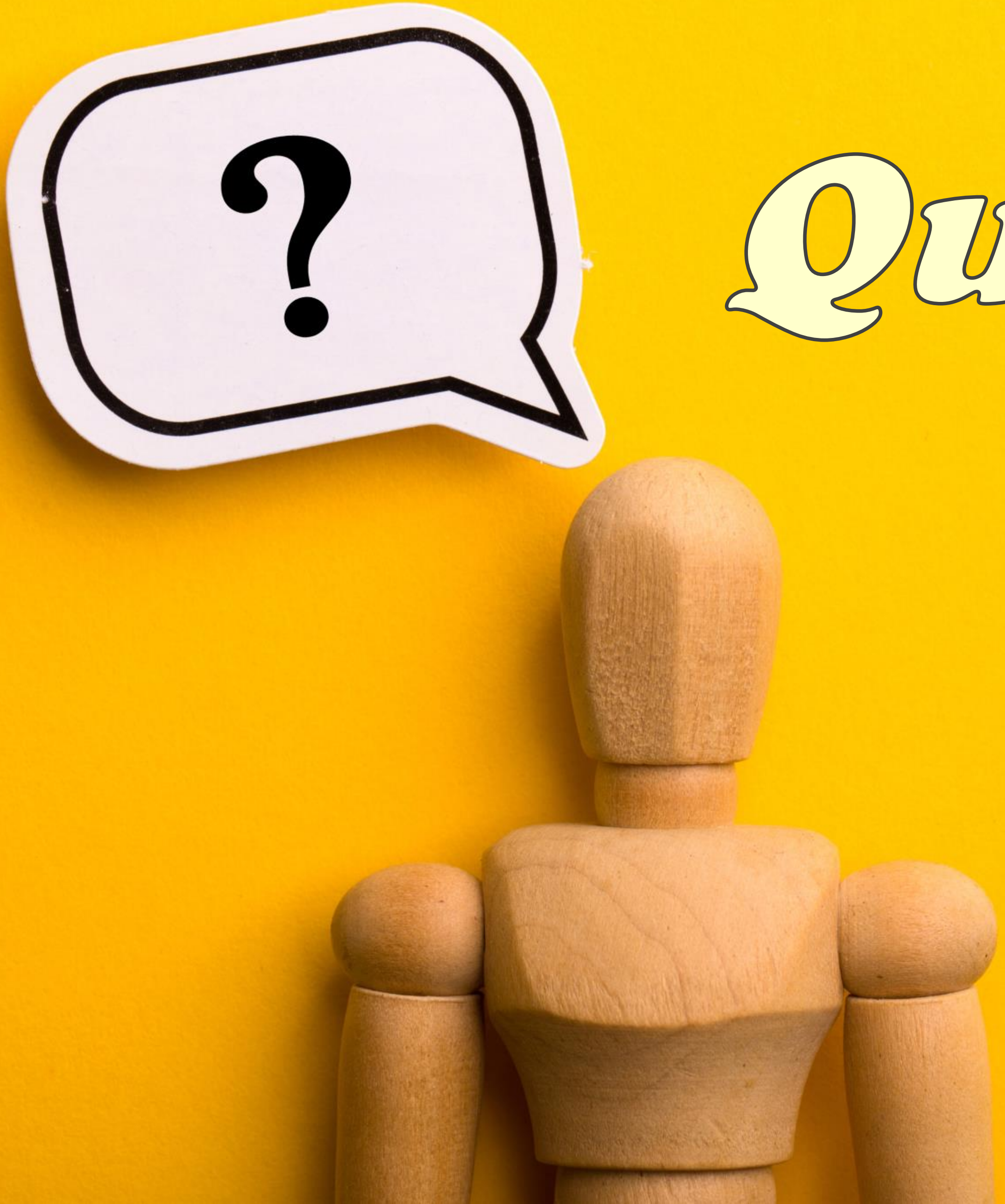
Issue: Linear Separability

5. **How** to **mitigate** it?

- Find useful features
  - Feature extraction (collect new features of your data)
  - Feature selection (keep fewer, more useful features)
- Transformation of features
  - Feature Engineering (e.g., $x \rightarrow x^2$)
  - Change Basis Vectors (e.g., PCA, LDA)
  - Kernel trick (e.g., for kernel SVM [W04b])
  - Feature Learning (e.g., Neural Networks [W09/10])

Reducing the number of non-linearly separable dimensions, makes it easier/faster to find the optimal decision boundary, i.e., practical benefit.

Questions!

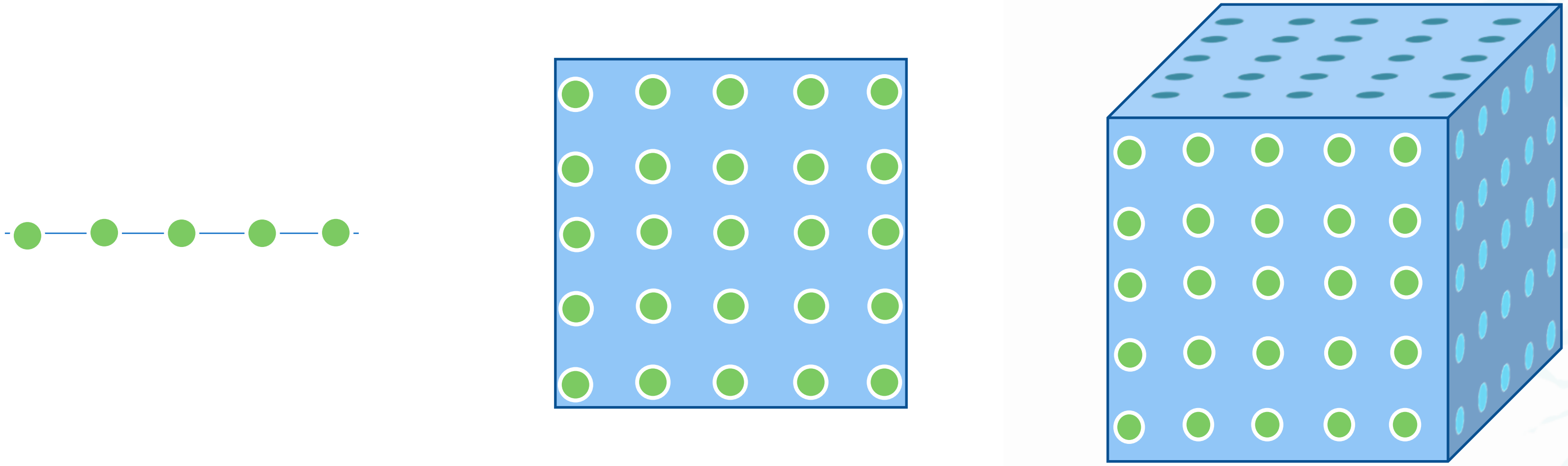


have a

Curse of Dimensionality

Sparsity with high dimensions

$m = 5$	$m = 25$	$m = 125$
$n = 1$	$n = 2$	$n = 3$



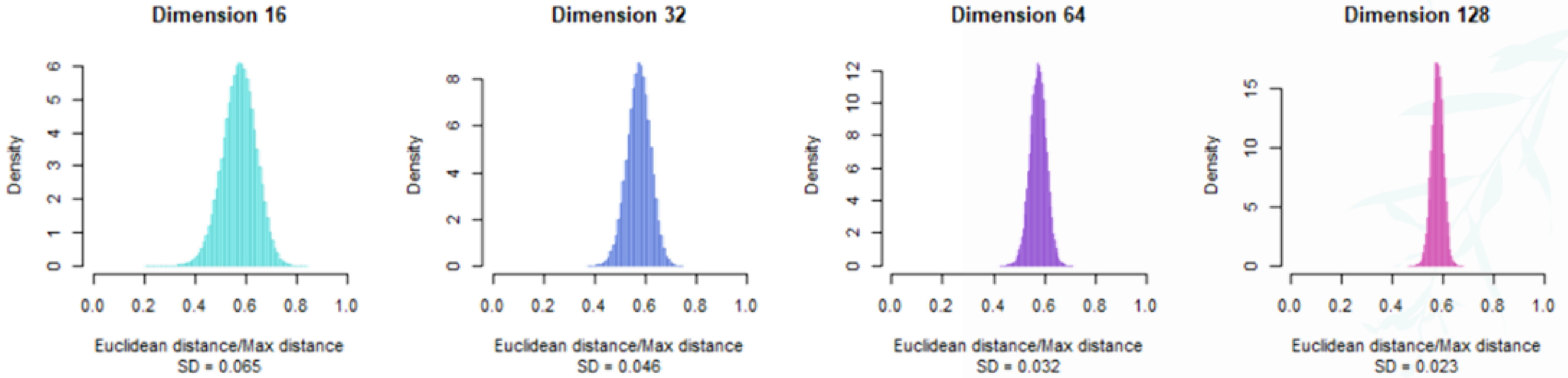
Sparsity problem: maintaining density of samples depends on exponential growth of the data

Chart illustration idea: Stanford CS229N

Curse of Dimensionality

In high dimensional space, most points are nearly the same distance away.

The result: learners that depend on distance break down in high dimensions.



https://stats.stackexchange.com/questions/451027/mathematical-demonstration-of-the-distance-concentration-in-high-dimensions

Issue: Curse of Dimensionality

1. **What** is the issue?
    1. Too many **features**
2. **Why** is it a problem?
    1. Data too sparse to inform about true decision boundary (for classification)
       => Too easy to fit a model on sparse training data => Overfitting
    2. Distances are too similar (bad for kNN [W02], clustering [W11])
3. **When** would it happen?
    1. Extracted more features than data instances (i.e., $n \gtrsim m$)
    2. Unstructured data (e.g., features as image pixels, sensor readings)

Issue: Curse of Dimensionality

4. How to **check** for it?

- Visualize histogram of **distances** (check for **variance** σ^2)

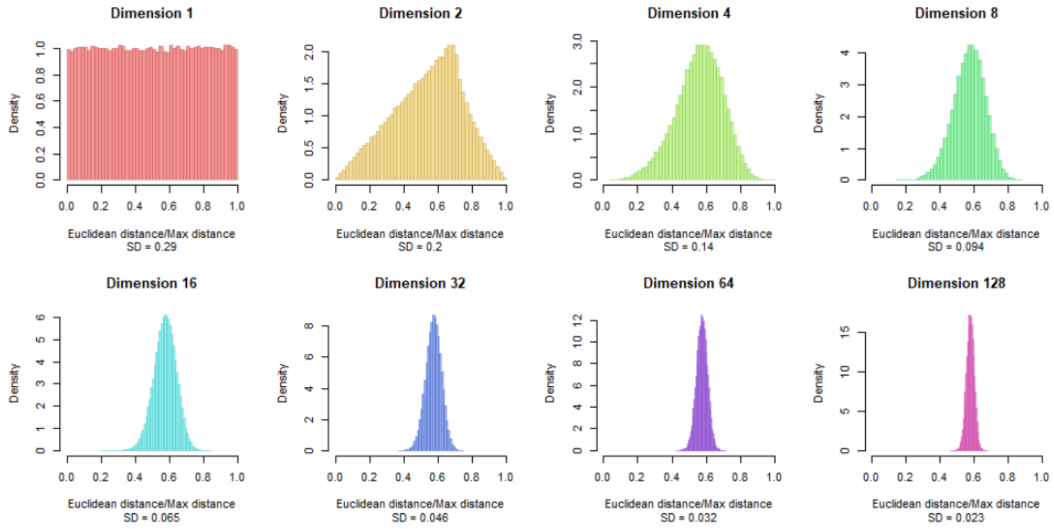


Image Credit: https://www.mygreatlearning.com/blog/understanding-curse-of-dimensionality/

Generally tedious to analyze this; just aim for: $n < m/5$

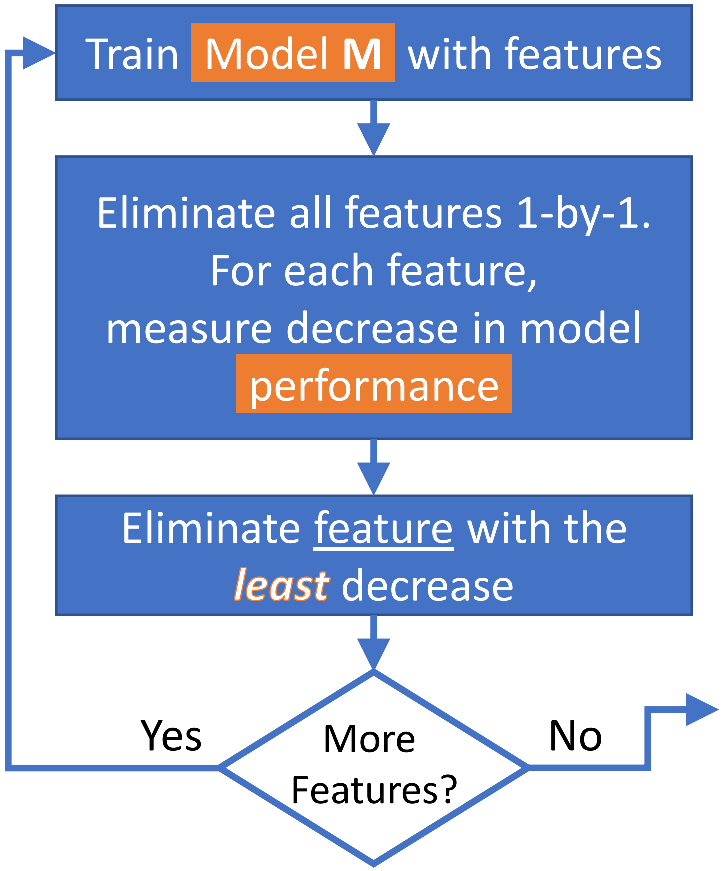
Issue: Curse of Dimensionality

5. **How** to **mitigate** it?
   - Feature Selection
     - Wrapper methods
     - Filter methods

Issue: Curse of Dimensionality

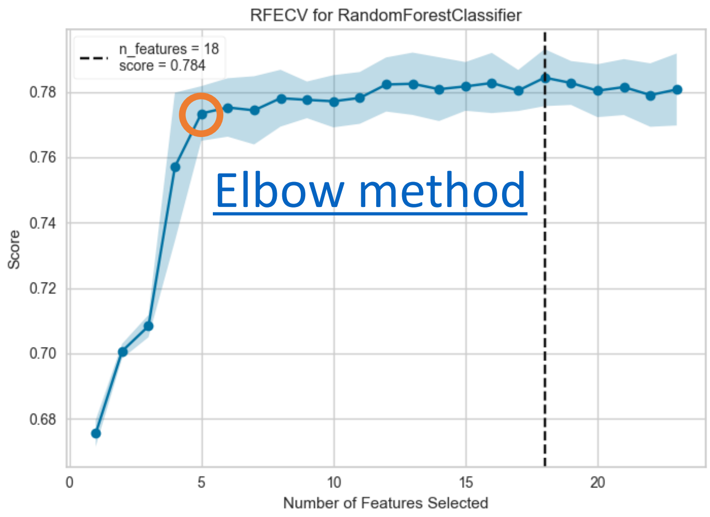
5. **How** to **mitigate** it?

- Feature Selection
  - **Wrapper methods** (e.g., RFE)
  - Filter methods



Recursive Feature Elimination (RFE)

Image Credit: https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html



Issue: Curse of Dimensionality

5. **How** to **mitigate** it?

- Feature Selection
  - Wrapper methods
  - **Filter methods**
    - Mutual Information = Information Gain [W03b]
    - Correlation

Recap W03a (slides 22-28)

Information gain

A chosen feature x_i divides the example set S into subsets $S_1, S_2, \ldots, S_c$ according to the C_i distinct values for x_i.

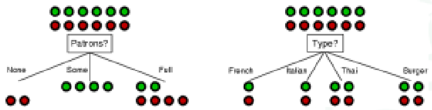
The entropy then reduces to the entropy of the subsets $S_1, S_2, \ldots, S_c$:

$$\text{remainder}(S, x_i) = \sum_{j=1}^{C_i} \frac{|S_j|}{|S|} H(S_j)$$

Information Gain (IG; "reduction in entropy") from knowing the value of x_i.
Choose the attribute with the largest IG:

$$\boxed{\text{IG}(S, x_i) = H(S) - \text{remainder}(S, x_i)}$$

For the training set at the root,
$p = n = 6,\ H\left(\frac{6}{12}, \frac{6}{12}\right) = 1$ bit.



Consider the attributes Patrons and Type:

$$IG(Patrons) = 1 - \left[\frac{2}{12}H(0,1) + \frac{4}{12}H(1,0) + \frac{6}{12}H\left(\frac{2}{6}, \frac{4}{6}\right)\right] = 0.541 \text{ bits}$$

$$IG(Type) = 1 - \left[\frac{2}{12}H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{2}{12}H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{12}H\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{4}{12}H\left(\frac{2}{4}, \frac{2}{4}\right)\right] = 0 \text{ bits}$$

Patrons has the highest IG, and so is chosen by DTL as the root.

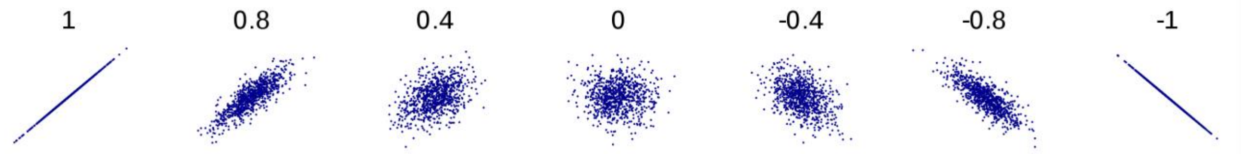
Further Reading: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/, https://towardsdatascience.com/feature-selection-for-machine-learning-3-categories-and-12-methods-6a4403f86543

Issue: Curse of Dimensionality

5. **How** to **mitigate** it?

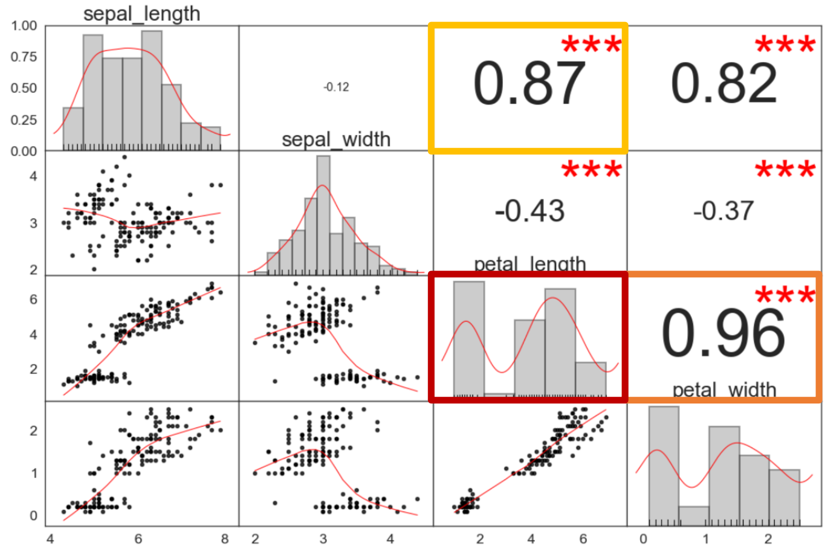
- Feature Selection
  - Wrapper methods
  - **Filter methods**
    - Mutual Information
    - Correlation

Pearson Correlation Coefficient



Higher **magnitude** => more **correlated**. $r > 0.7$ is very **high**.
Further reading: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Pearson Correlation Coefficients
for Iris (flower) dataset



`petal_length` is highly correlated to **`petal_width`** and **`sepal_length`**
=> Should remove, due to redundancy

Issue: Curse of Dimensionality

5. **How** to **mitigate** it?
   - Feature Selection
   - Dimensionality Reduction
     - Linear Matrix Factorization (e.g., PCA, LDA)
     - Non-linear Manifold Learning (e.g., SOM, MDS, t-SNE, UMAP)
     - Deep Auto-Encoders [W12]

Only these are **examinable**

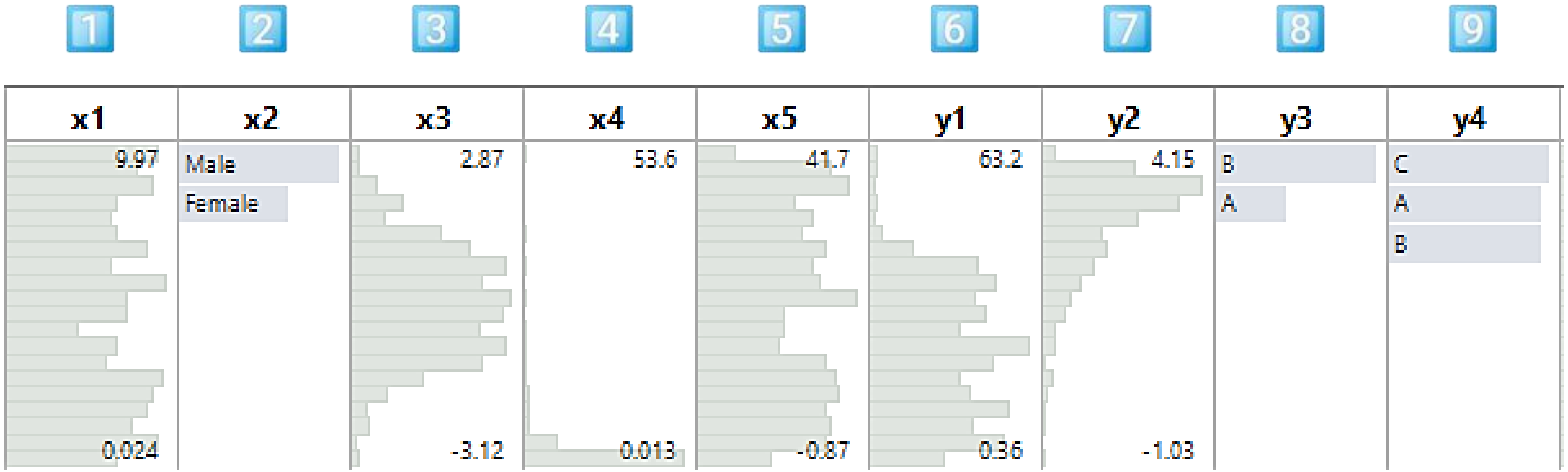
Further reading:
https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/

Benefits of Feature Selection

- Avoid Curse of Dimensionality
- Faster model training (optimizing fewer parameters on fewer features)
- Fewer features to read => easier to interpret

Imbalanced Data

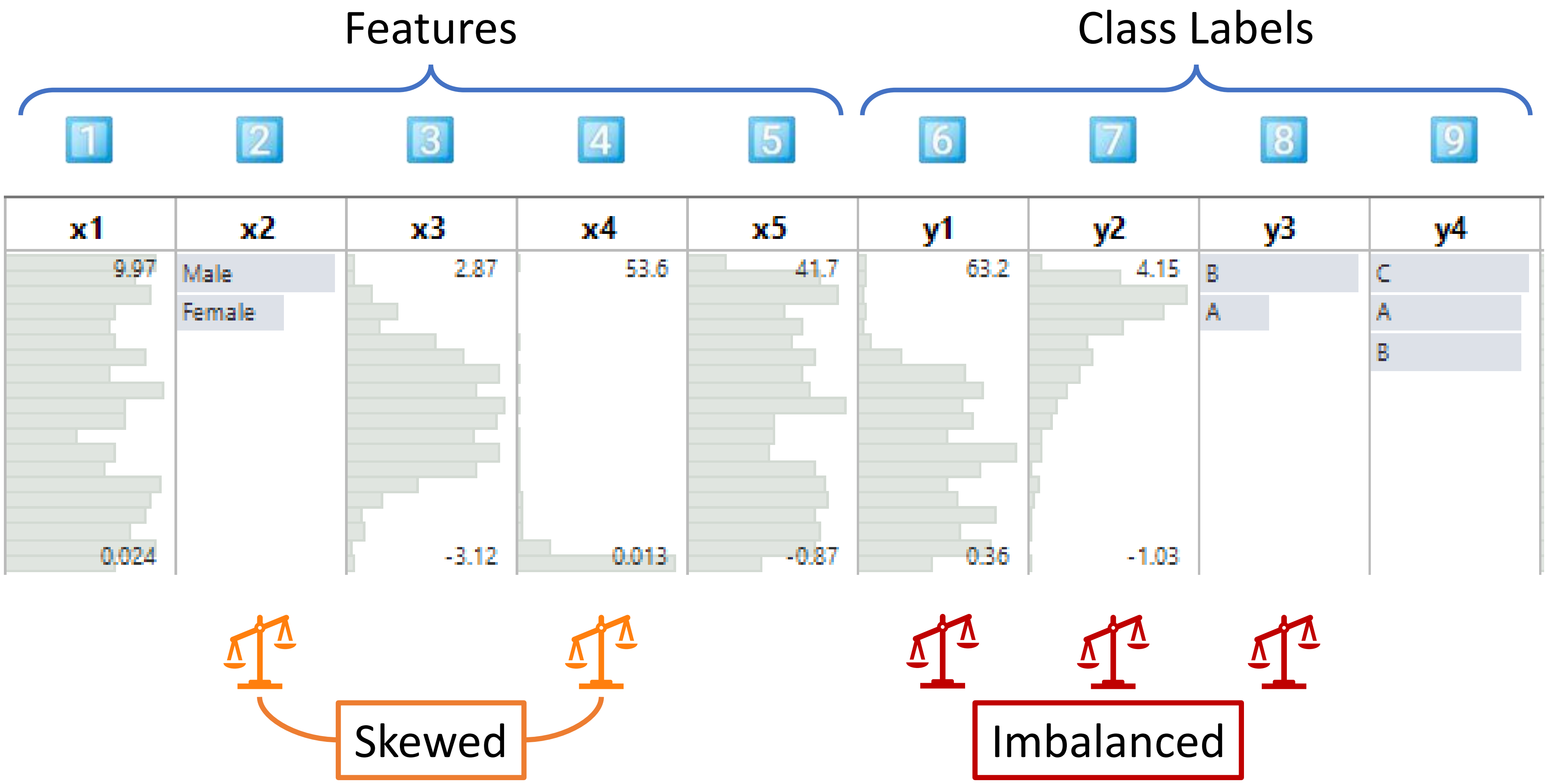
Which of the following variables (columns) are imbalanced?



Emote (react) in Slack #lecture channel <u>one or more options</u> (MRQ)

Exercise W08b.3 Solution

Which of the following variables (columns) are imbalanced?

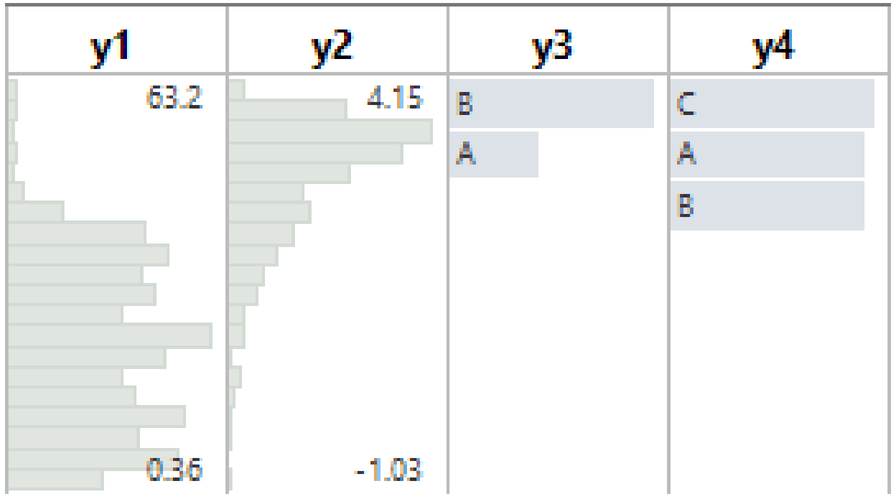


Issue: Imbalanced Data

1. **What** is the issue?
    1. Values not evenly distributed in feature
    2. Data may be skewed
2. **Why** is it a problem?
    1. Evaluation metrics become misleading to interpret [W07b]
    2. Models overfit to majority class
3. **When** would it happen?
    1. When events unevenly occur (e.g., rare cancer)
    2. When data collection is uneven (e.g., only positive survey respondents)

Issue: Imbalanced Data

4. **How** to **check** for it?
   - Visualize **histogram** or **bar chart** of feature values



Issue: Imbalanced Data

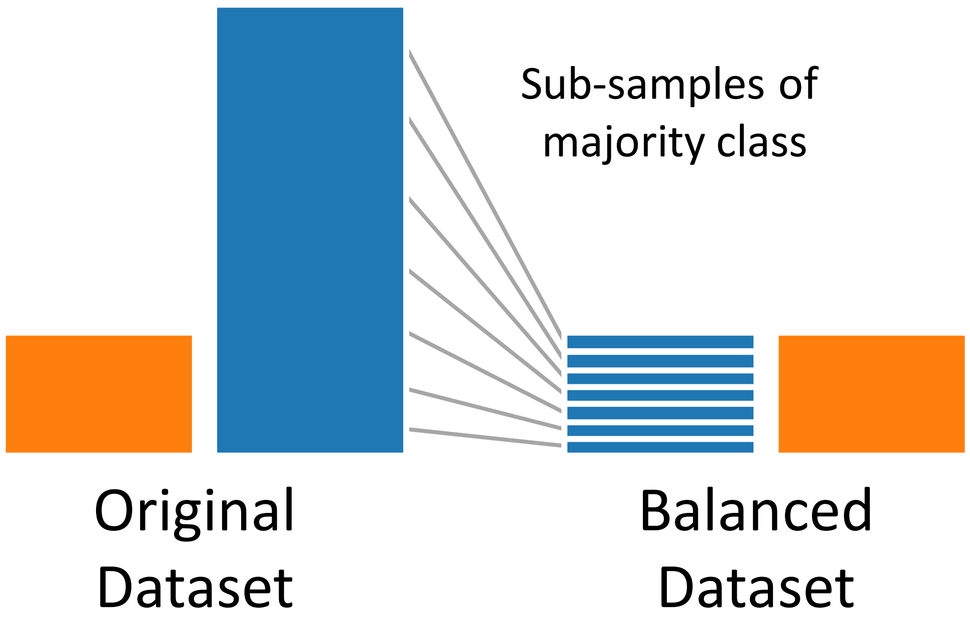
5. **How** to **mitigate** it?
    - Collect more data instances
    - Resample instances (e.g., Undersampling, Oversampling, SMOTE)

Further reading:
https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/

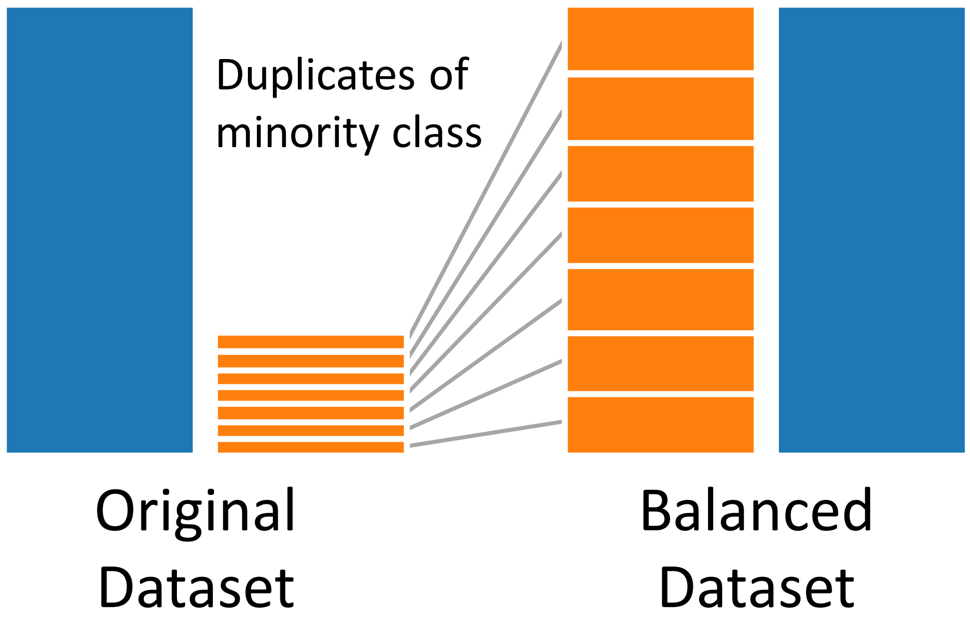
Data Resampling

Image credit:
https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/

Undersampling

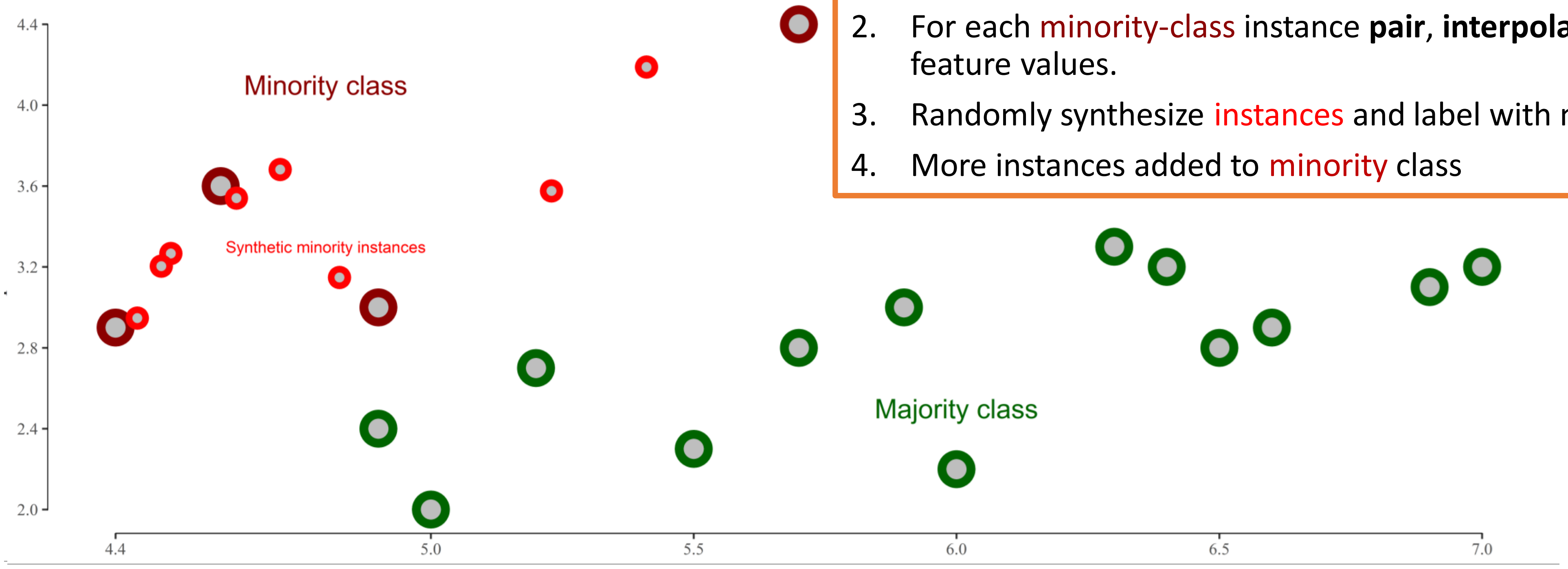


Oversampling



Data leakage (snooping): remember to **first split** dataset to train–test, then **resample** train and test datasets separately

Synthetic Minority Oversampling Technique (SMOTE)

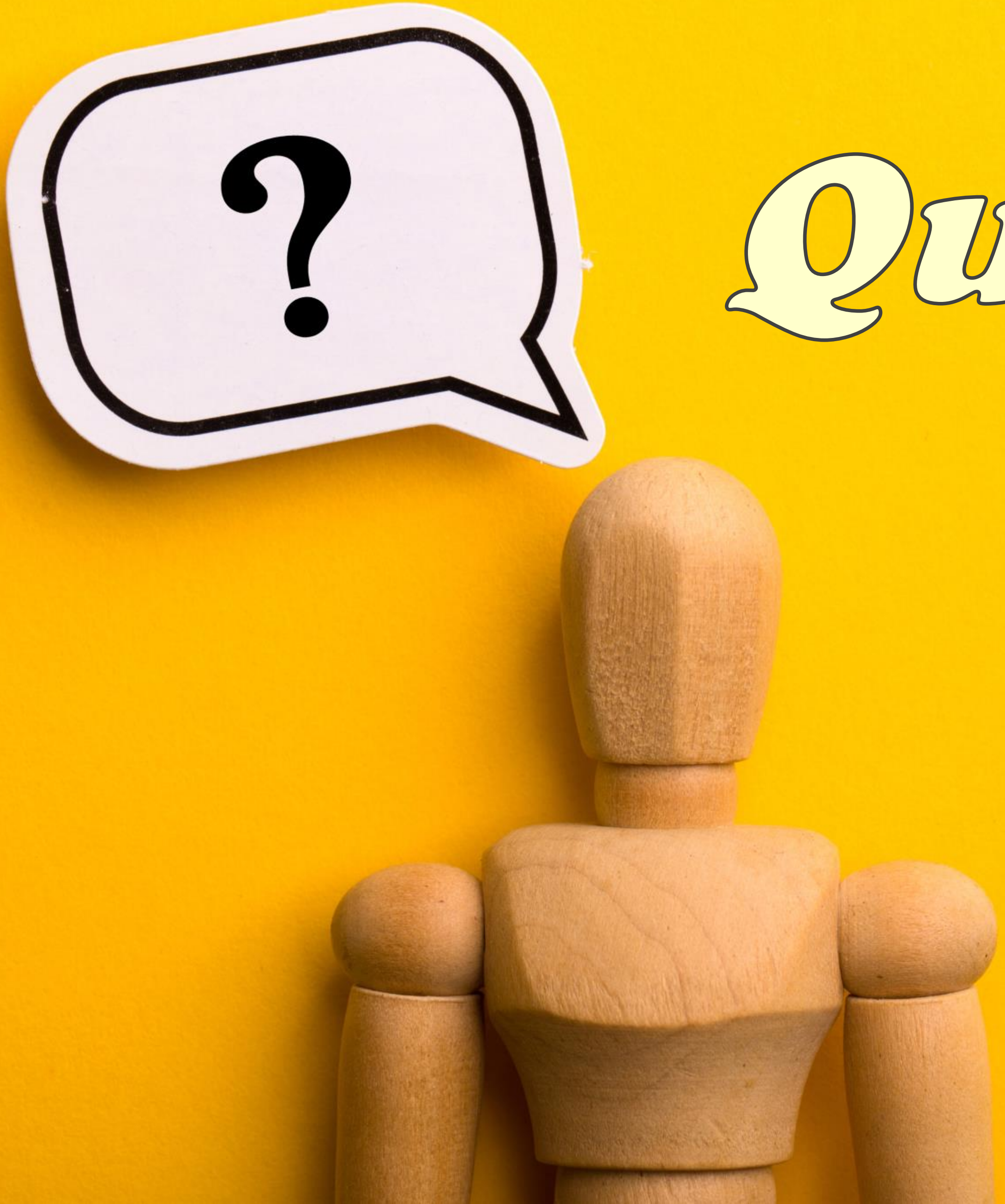


Steps

1. Consider minority and majority instances in vector space.
2. For each minority-class instance **pair**, **interpolate** their feature values.
3. Randomly synthesize instances and label with minority class
4. More instances added to minority class

Image Credit: https://www.quora.com/Can-you-explain-me-SMOTE-Synthetic-Minority-Over-sampling-Technique-in-simple-terms

Questions!



Wrapping Up

What did we learn?

Data **Issues**

1. Linear Separability
2. Curse of Dimensionality
3. Imbalanced Data

Issue Template

1. **What** is the issue?
2. **Why** is it a problem?
3. **When** would it happen?
4. **How** to **check** for it?
5. **How** to **mitigate** it?

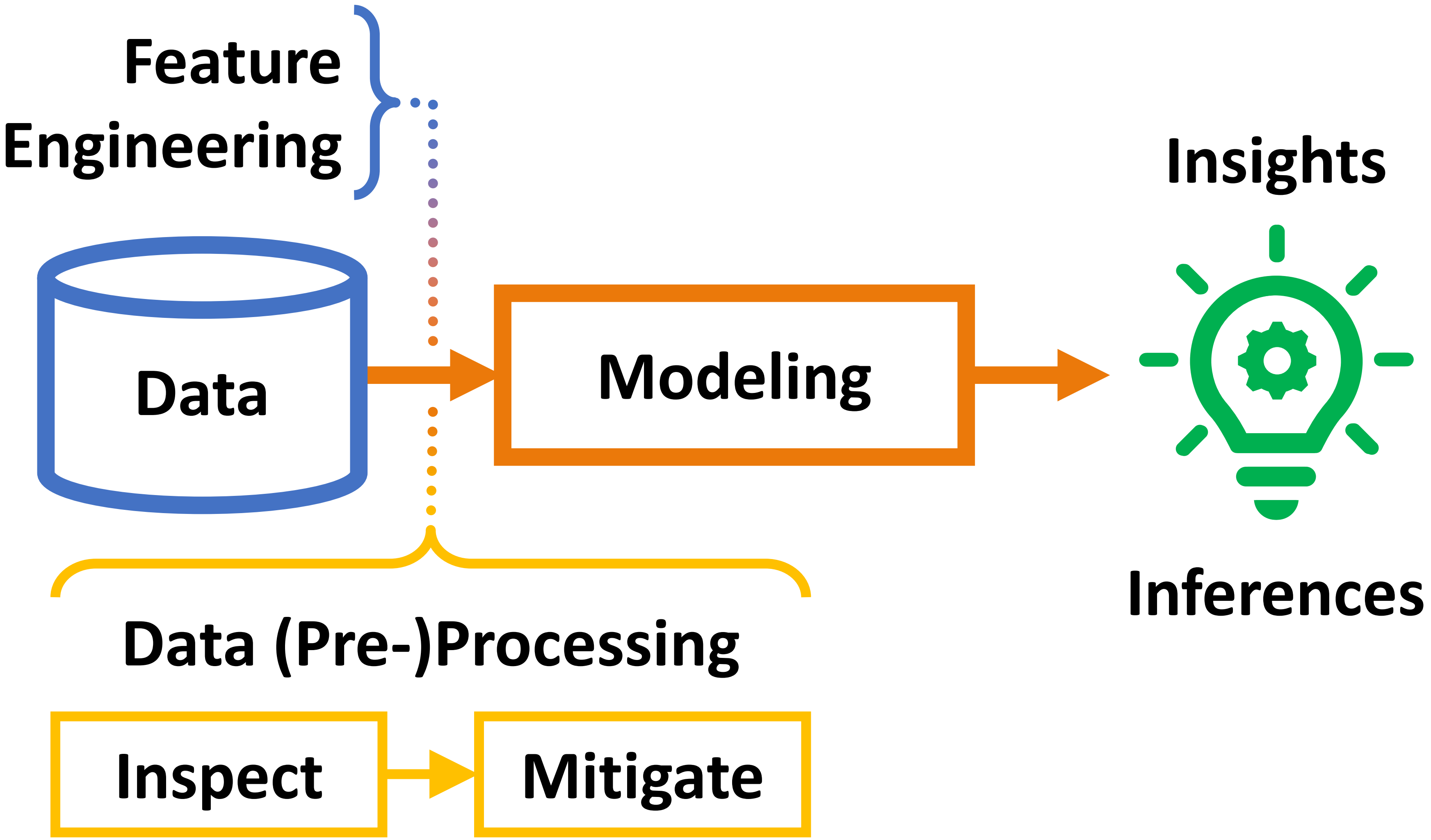
Checks

1. Linear SVM, PCA, LDA
2. Visualize Histograms

Mitigations

1. Dimensionality Reduction
   (PCA, LDA, Deep Auto-Encoders)
2. Feature Selection
   (Recursive Feature Elimination, Correlation, Mutual Information)
3. Resampling
   (Under/Oversampling, SMOTE)

Machine Learning Pipeline



On Thursday: Feature Engineering