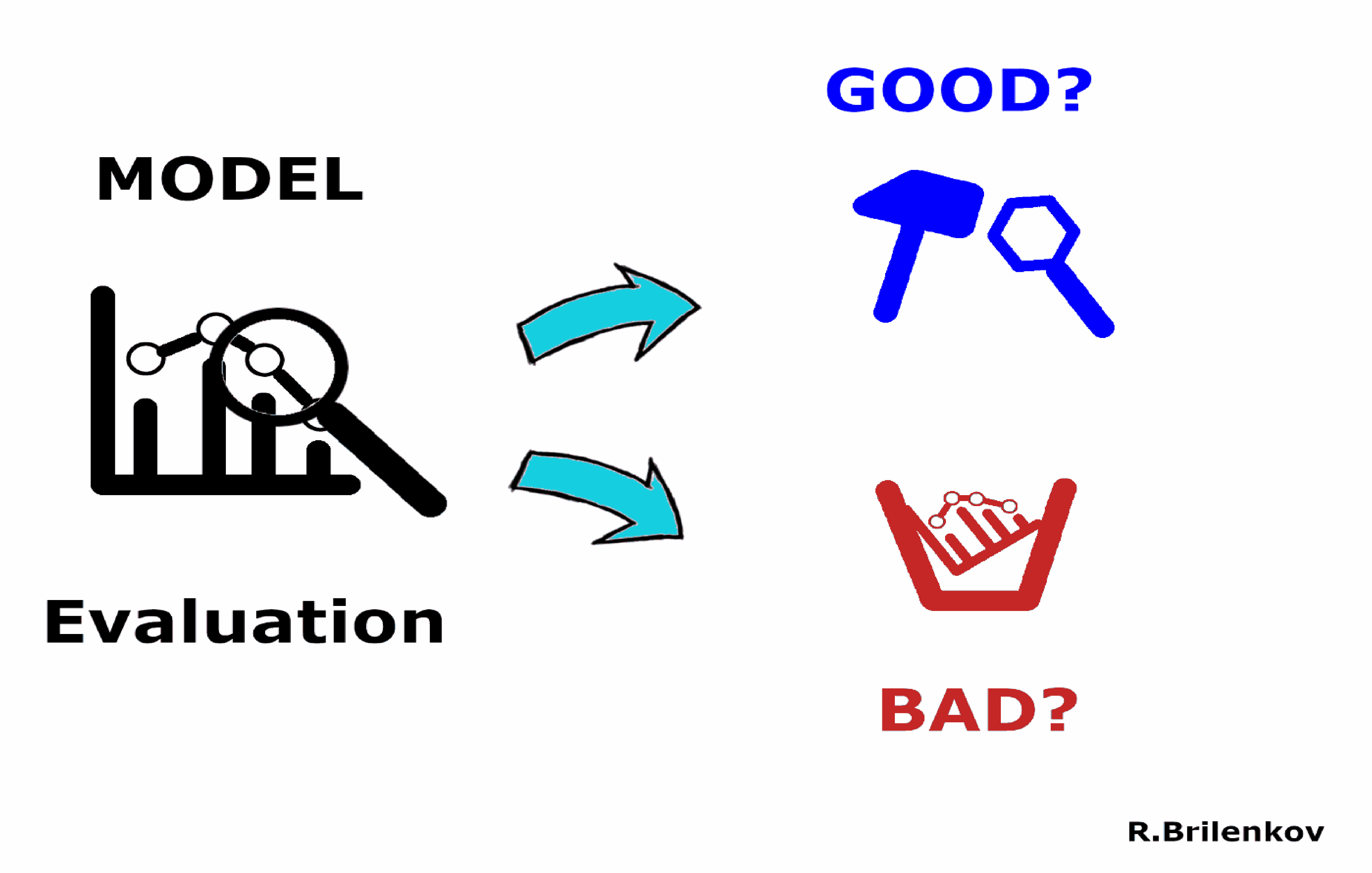
Machine learning

Model Evaluation, Validation sets and PCA

Exercise VI

פיתוח:
משה פרידמן

Model quality evaluation



Confusion matrix - שיערוך המודל

		Predicted	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

- True Positive
  - זיהוי נכון של דוגמא חיובית
- False Positive = False Alarm
  - זיהוי של דוגמא שלילית כחיובית
- True Negative
  - זיהוי נכון של דוגמא שלילית
- False Negative = Miss Detection
  - זיהוי של דוגמא חיובית כשלילית

Credit: Dr. Koby Mike

תרגיל 2א –זיהוי זברות – שערוך המודל – Confusion matrix

		חזוי	
		זברה	לא זברה
אמת	זברה		
	לא זברה		

- לצורך זיהוי תמונות של זברות נבנה מסווג שמטרתו להפריד בין זברות לבין חיות אחרות.
- כאשר המסווג נתקל בתמונה של זברה היא מסווגת תמיד כזברה.
- כאשר המסווג נתקל בתמונה של חיה אחרת, היא מסווגת בטעות כזברה ב- 5% מהמקרים.
- המסווג הופעל על מאגר תמונות בו יחס הזברות לחיות אחרות הוא 1:1000.
- **בנו את Confusion matrix של המסווג.**

Credit: Dr. Koby Mike

תרגיל 2א –זיהוי זברות – שערוך המודל – Confusion matrix - פתרון

- לצורך זיהוי תמונות של זברות נבנה מסווג שמטרתו להפריד בין זברות לבין חיות אחרות.
- כאשר המסווג נתקל בתמונה של זברה היא מסווגת תמיד כזברה.
- כאשר המסווג נתקל בתמונה של חיה אחרת, היא מסווגת בטעות כזברה ב- 5% מהמקרים.
- המסווג הופעל על מאגר תמונות בו יחס הזברות לחיות אחרות הוא 1:1000.
- **בנו את Confusion matrix של המסווג.**

		חזוי	
		זברה	לא זברה
אמת	זברה	1	0
	לא זברה	50	950

Credit: Dr. Koby Mike

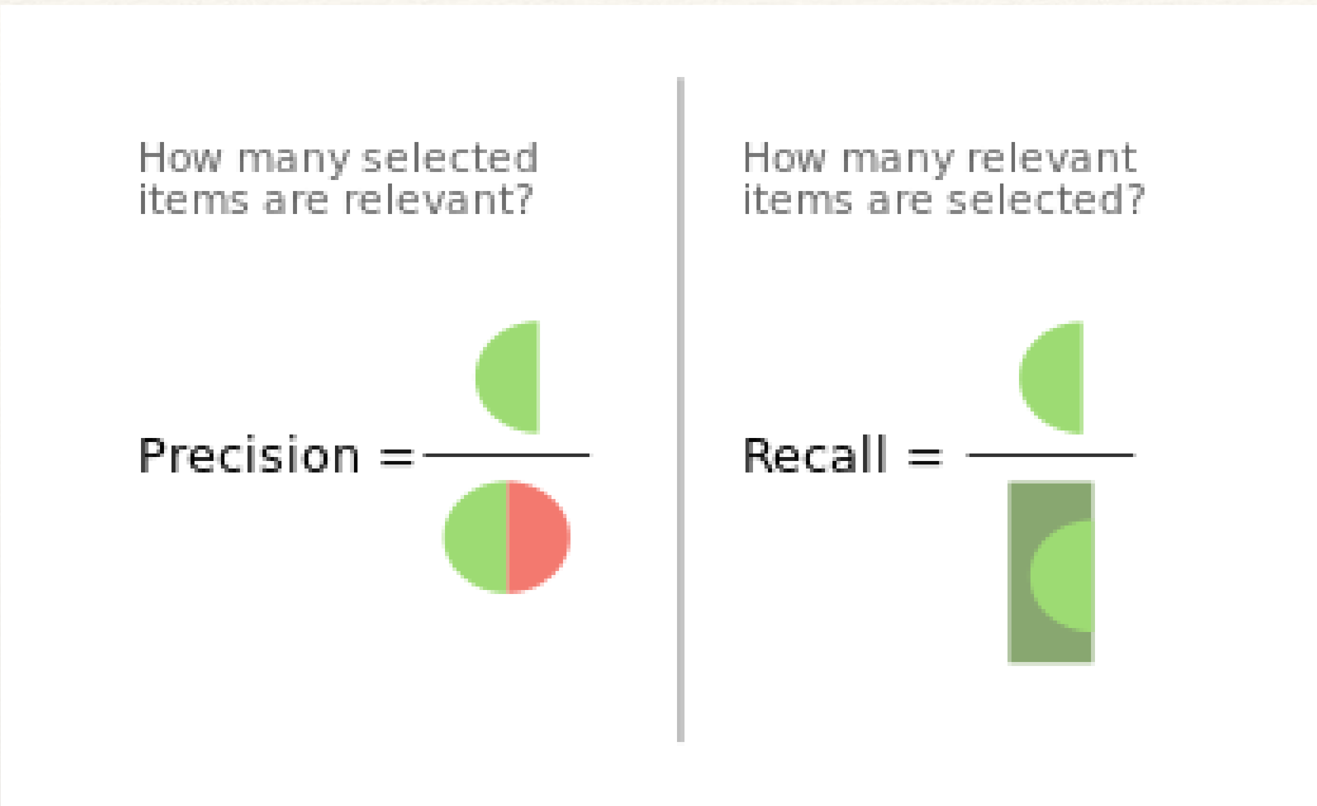
שיערוך המודל – accuracy ו – error-rate

		Predicted	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

- מדד Accuracy - יחס הדגימות שסווגו נכון לסך הדגימות
- מדד Error rate – יחס הדגימות שסווגו בצורה שגויה לסך הדגימות
- יתרונות המדדים:
  - פשוטים לחישוב
  - פשוטים להבנה
- חסרונות המדדים:
  - המדדים מטעים, אם כמות המחלקות לא מאוזנת, בין המחלקה החיובית לשלילית.
  - אין התייחסות למטרת החיזוי (המחלקה החיובית והשלילית מקבלים משקל שווה)

Credit: Dr. Koby Mike

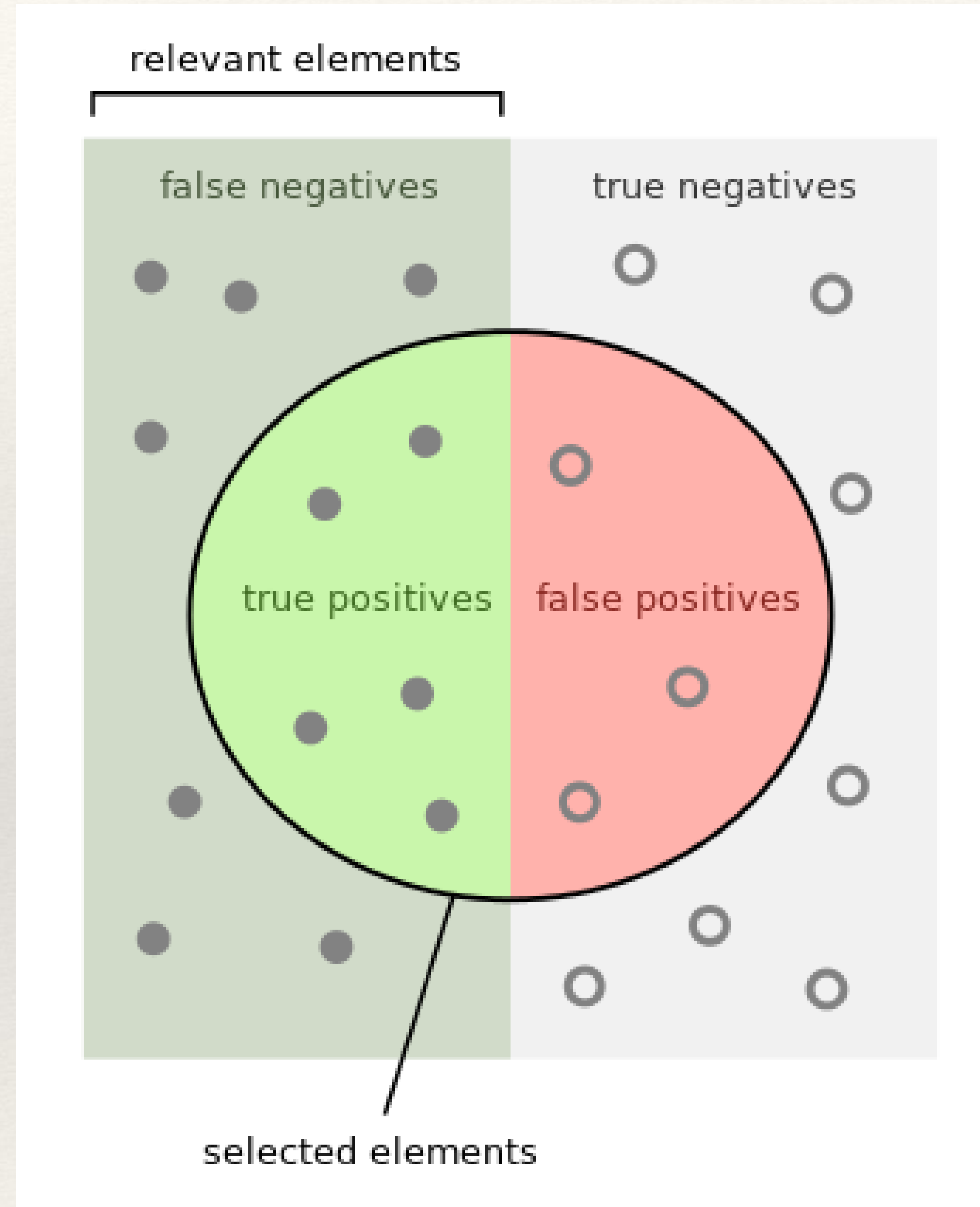
שיערוך המודל – precision ו – recall



		Predicted	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
True	Negative	False Positive (FP)	True Negative (TN)

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$



Credit: Dr. Koby Mike

תרגיל 2ב – זיהוי זברות – שערוך המודל – חשבו את מדדי שיערוך המודל הבאים

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

		חזוי	
		זברה	לא זברה
אמת	זברה	1	0
	לא זברה	50	950

חשבו את מדדי שיערוך המודל:

Accuracy =

Error rate =

Precision =

Recall =

Credit: Dr. Koby Mike

תרגיל 2ב – זיהוי זברות – שערוך המודל –
חשבו את מדדי שיערוך המודל הבאים - פתרון

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

		חזוי	
		זברה	לא זברה
אמת	זברה	1	0
	לא זברה	50	950

חשבו את מדדי שיערוך המודל:

$$\text{Accuracy} = \frac{1+950}{1+950+0+50} \approx 95\%$$

$$\text{Error rate} = \frac{50+0}{1+950+0+50} \approx 5\%$$

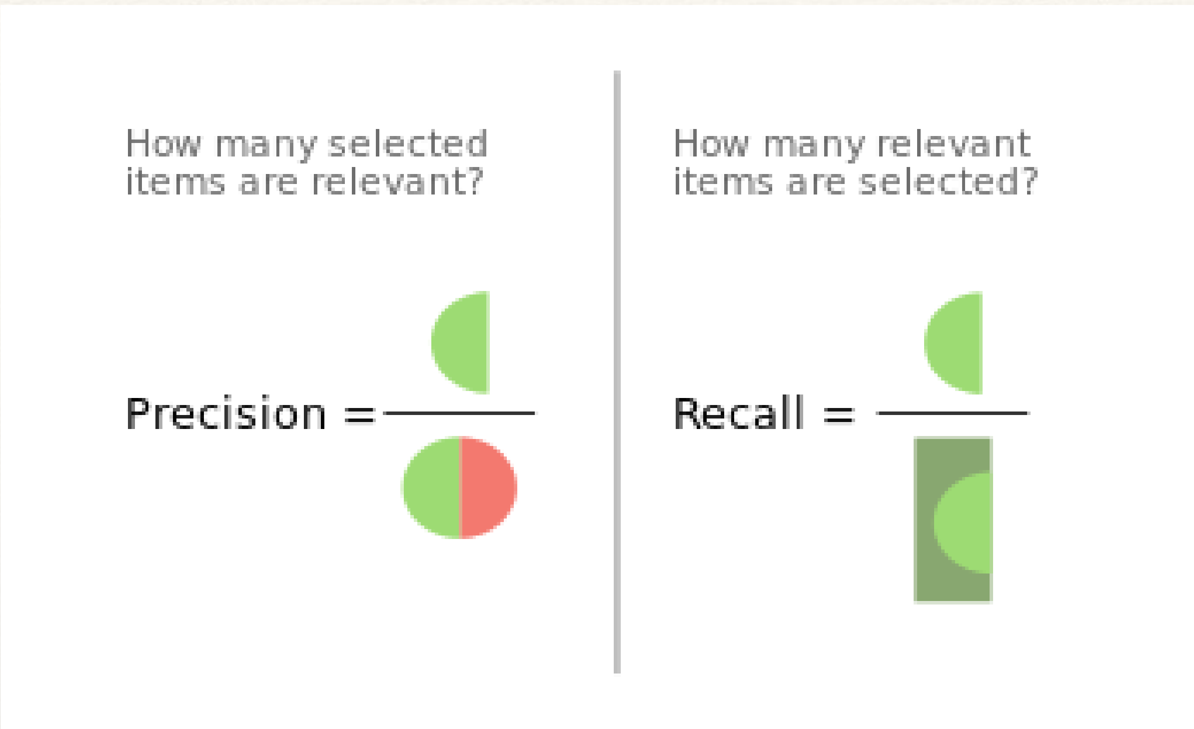
$$\text{Precision} = \frac{1}{1+50} \approx 2\%$$

$$\text{Recall} = \frac{1}{1+0} = 100\%$$

Credit: Dr. Koby Mike

שיערוך המודל – מדד F1

- ממוצע (הרמוני) של מדדי הprecision ו-recall
- נע בין 0-1
- מקבל ערכים גבוהים רק אם:
  - גם ה-precision טוב
  - גם ה-recall טוב



$$precision = \frac{TP}{TP+FP}$$

$$recall = \frac{TP}{TP+FN}$$

		Predicted	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Credit: Dr. Koby Mike

תרגיל 2ג – זיהוי זברות – שערוך המודל – חשבו את מדד F1 - פתרון

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

		חזוי: זברה	חזוי: לא זברה
אמת	זברה	1	0
אמת	לא זברה	50	950

חשבו את מדדי שיערוך המודל:

$$\text{Precision} = \frac{1}{1+50} \approx 0.0196$$

$$\text{Recall} = \frac{1}{1+0} = 1$$

$$\text{F1} =$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Credit: Dr. Koby Mike

תרגיל 2ג – זיהוי זברות – שערוך המודל – חשבו את מדד F1 - פתרון

$$\text{precision} = \frac{TP}{TP+FP}$$

$$\text{recall} = \frac{TP}{TP+FN}$$

		חזוי	
		זברה	לא זברה
אמת	זברה	1	0
	לא זברה	50	950

חשבו את מדדי שיערוך המודל:

$$\text{Precision} = \frac{1}{1+50} \approx 0.0196$$

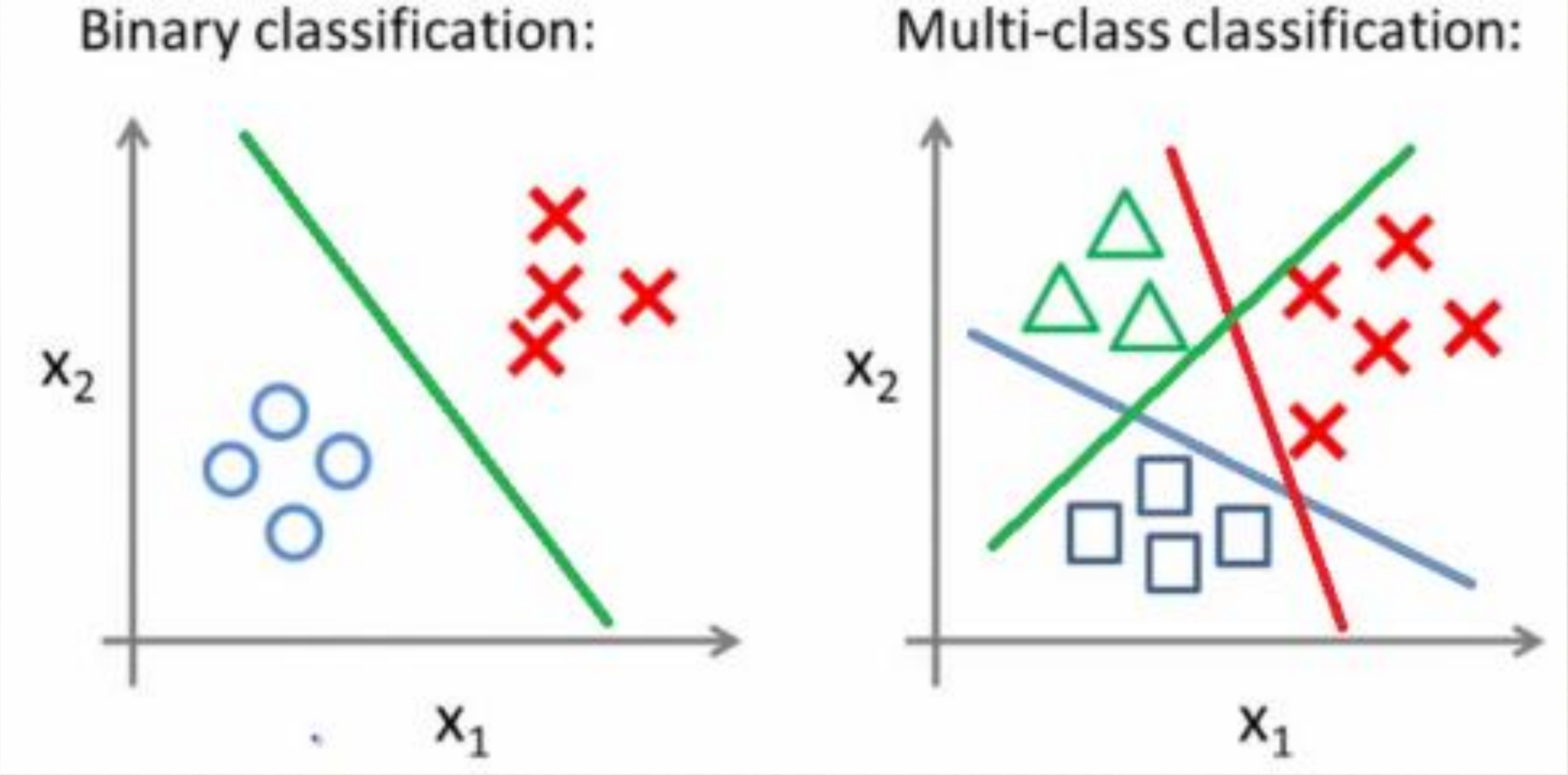
$$\text{Recall} = \frac{1}{1+0} = 1$$

$$F1 \approx \frac{2 \cdot 0.0196 \cdot 1}{0.0196+1} \approx 0.03846$$

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Credit: Dr. Koby Mike

Multi-class classification evaluation



Multi-class example: diamond classification

The target contains 4 types of diamonds:

- ideal, premium, good, and fair.

How could we evaluate the results?

Step 1: confusion matrix

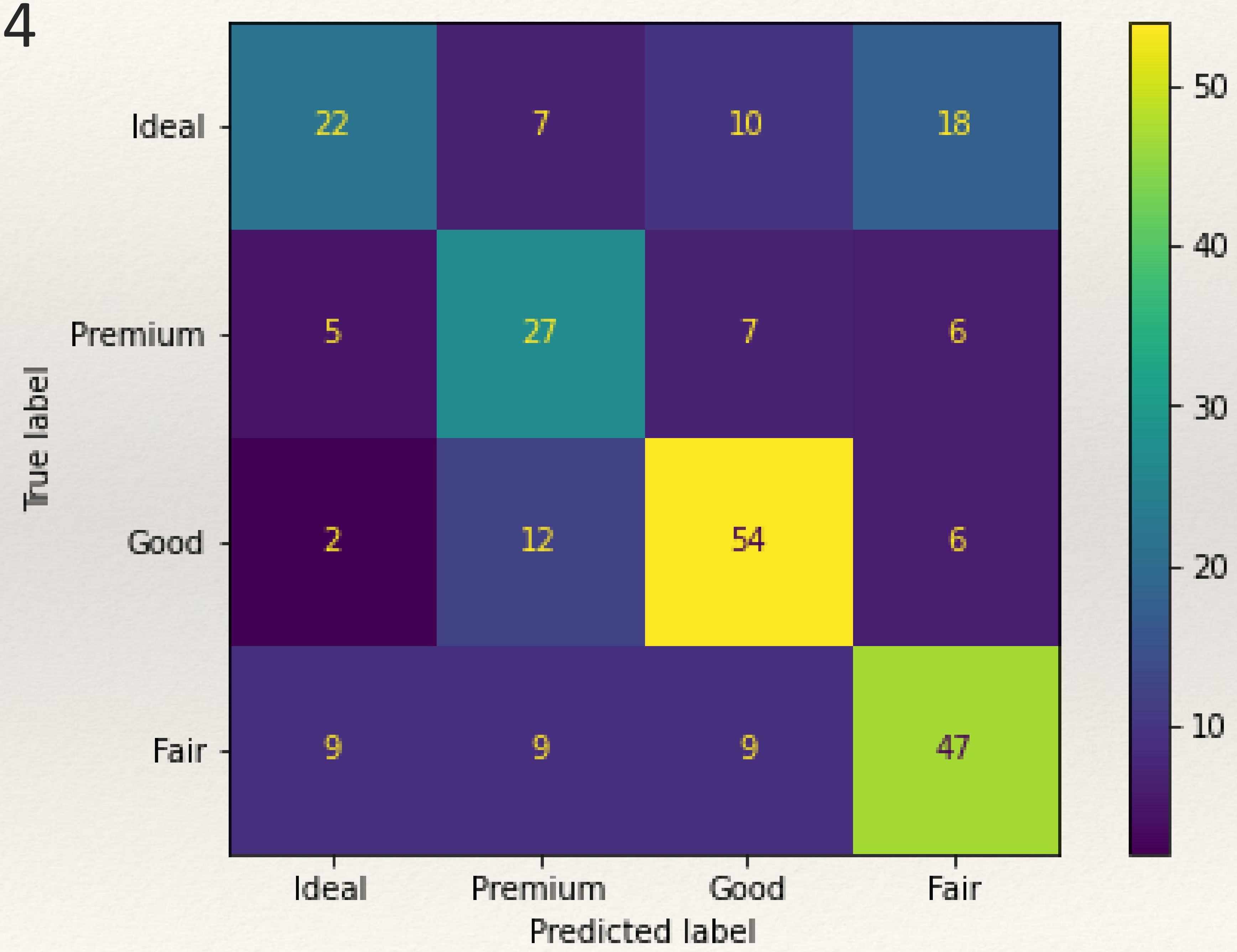
Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification- Confusion matrix

Confusion matrix 4x4

The target contains 4 types of diamonds:

- ideal, premium, good, and fair.



Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

We will use the One-vs-Rest strategy approach to transfer it to 4 binary classification problems:

- Task 1: ideal vs . not ideal [premium, good, fair]
- Task 2: premium vs. not premium [ideal, good, fair]
- Task 3: good vs. not good [ideal, premium, fair]
- Task 4: fair vs. vs. not fair [ideal, premium, good]

For each of these tasks we create a binary confusion matrix and derive the precision, recall and F1

Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

- Task 1: ideal vs. not ideal
- Task 2: premium vs. not premium
- Task 3: good vs. not good
- Task 4: fair vs. vs. not fair

	precision	recall	f1-score	support
Ideal	0.58	0.39	0.46	57
Premium	0.49	0.60	0.54	45
Good	0.68	0.73	0.70	74
Fair	0.61	0.64	0.62	74

Macro averaging: arithmetic mean of all metrics.

Question: What's the macro precision?

Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

- Task 1: ideal vs. not ideal
- Task 2: premium vs. not premium
- Task 3: good vs. not good
- Task 4: fair vs. vs. not fair

	precision	recall	f1-score	support
Ideal	0.58	0.39	0.46	57
Premium	0.49	0.60	0.54	45
Good	0.68	0.73	0.70	74
Fair	0.61	0.64	0.62	74

Macro averaging: arithmetic mean of all metrics.

Question: What's the macro precision?

Answer: (0.58+0.49+0.68+0.61)/4=0.59

Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

Question: What is the TP, FP, FN, TN for every class?

Task 1: ideal vs. not ideal

- TP= , FP= , FN= , TN=

Task 2: premium vs. not premium

- TP= , FP= , FN= , TN=

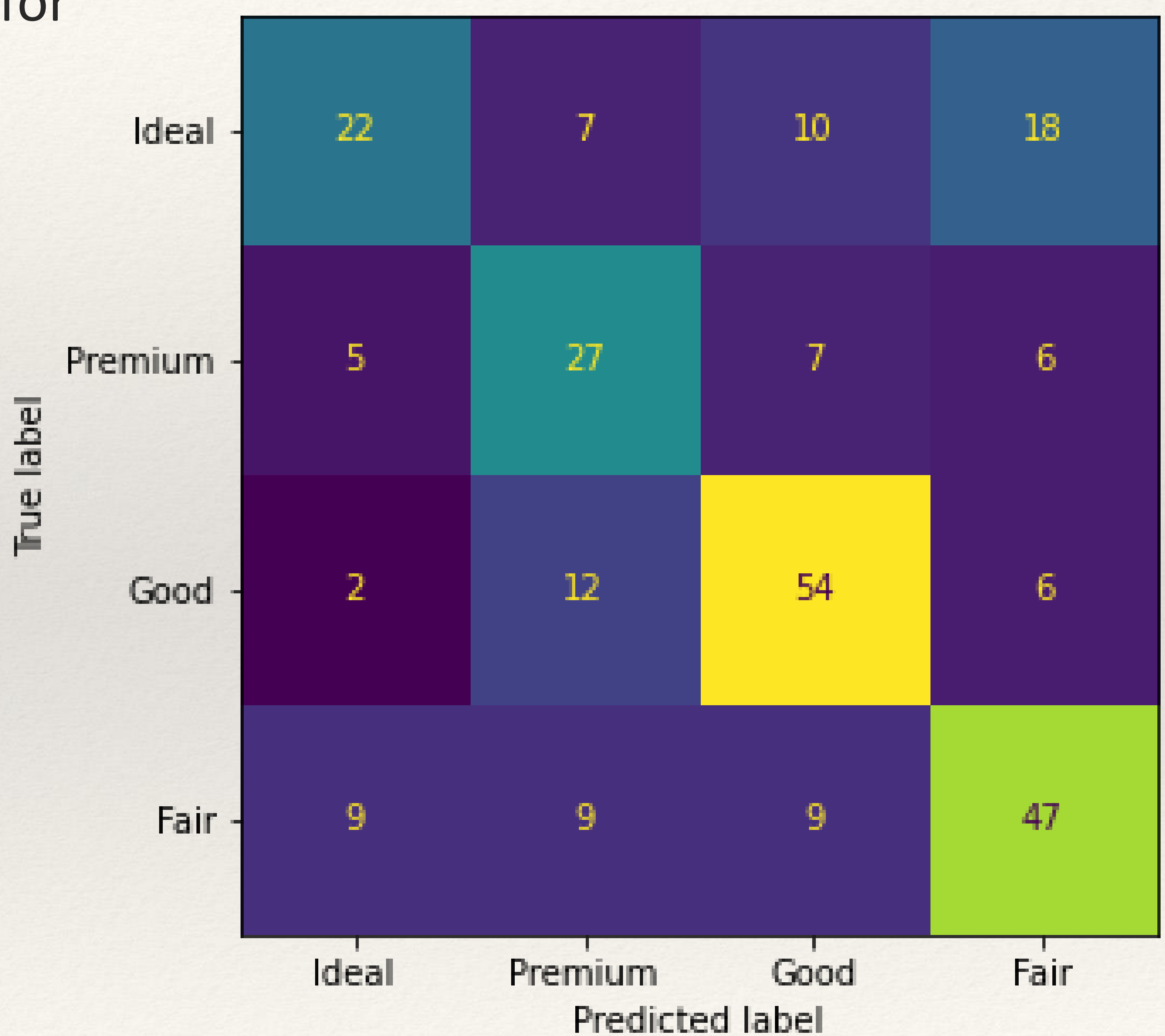
Task 3: good vs. not good

- TP= , FP= , FN= , TN=

Task 4: fair vs. vs. not fair

- TP= , FP= , FN= , TN=

Confusion matrix 4x4



Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

Confusion matrix 4x4

Question: What is the TP, FP, FN, TN for every class?

Task 1: ideal vs. not ideal

- TP= 22, FP=16, FN=35, TN=177

Task 2: premium vs. not premium

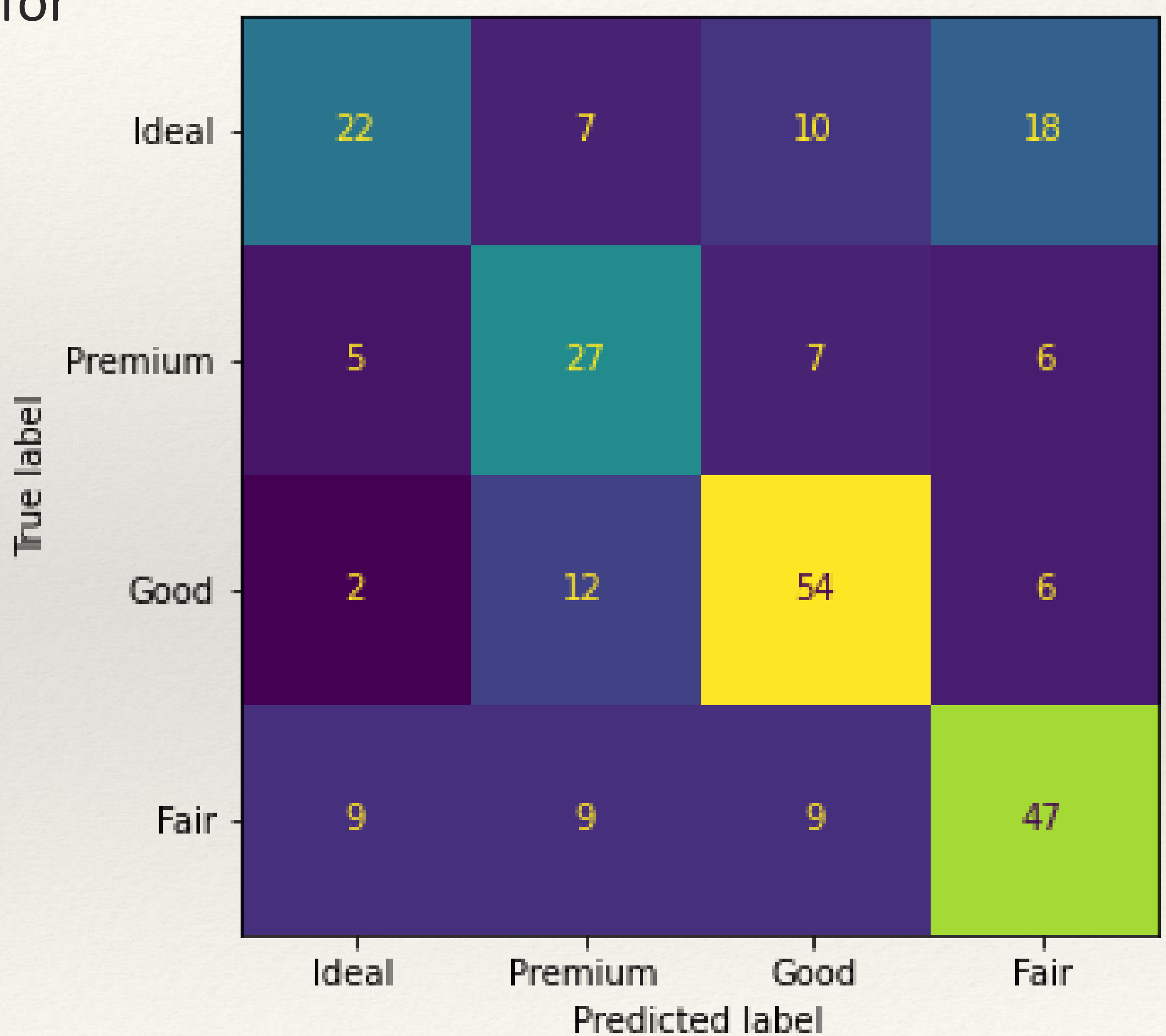
- TP=27 , FP=28, FN=18, TN=177

Task 3: good vs. not good

- TP=54, FP=26, FN=20, TN=150

Task 4: fair vs. vs. not fair

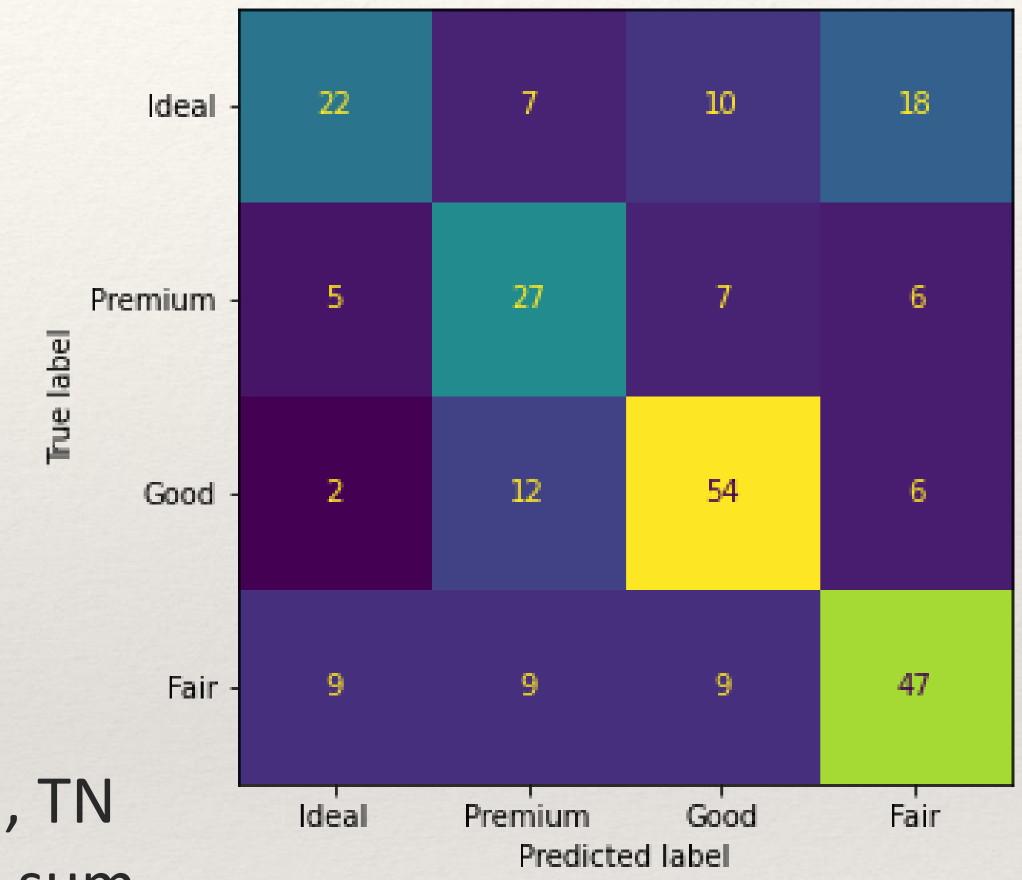
- TP=47 , FP=30, FN=27, TN=146



Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

Confusion matrix 4x4



Task 1: ideal vs. not ideal

- TP= 22, FP=16, FN=35, TN=177

Task 2: premium vs. not premium

- TP=27 , FP=28, FN=18, TN=177

Task 3: good vs. not good

- TP=54, FP=26, FN=20, TN=150

Task 4: fair vs. vs. not fair

- TP=47 , FP=30, FN=27, TN=146

For **micro-average**, we sum each one of the TP, FP, FN, TN and calculate the evaluation average metrics from the sum.

Exercise: calculate total TP and FN and derive micro-average recall

Credit: Bex T. @ towardsdatascience.com

Multi-class example: diamond classification - What about precision, recall and F1?

Task 1: ideal vs. not ideal

- TP= 22, FP=16, FN=35, TN=177

Task 2: premium vs. not premium

- TP=27 , FP=28, FN=18, TN=177

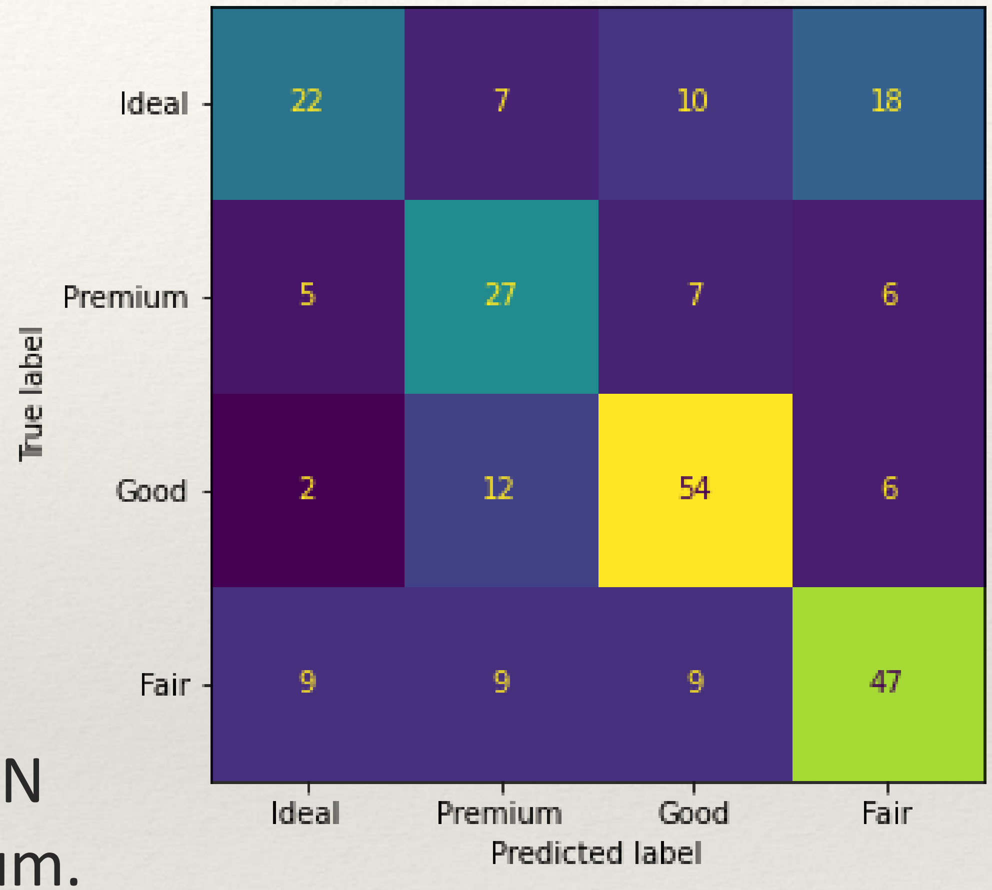
Task 3: good vs. not good

- TP=54, FP=26, FN=20, TN=150

Task 4: fair vs. vs. not fair

- TP=47 , FP=30, FN=27, TN=146

Confusion matrix 4x4



For **micro-average**, we sum each one of the TP, FP, FN, TN and calculate the evaluation average metrics from the sum.

Exercise: calculate total TP and FN and derive micro-average recall

Answer:
TP-total=22+27+54+47=150
FN-total=35+18+20+27=100
Micro-average-Recall=
150/(150+100)=0.6

Credit: Bex T. @ towardsdatascience.com

Validation and validation-sets

Introduction (1)

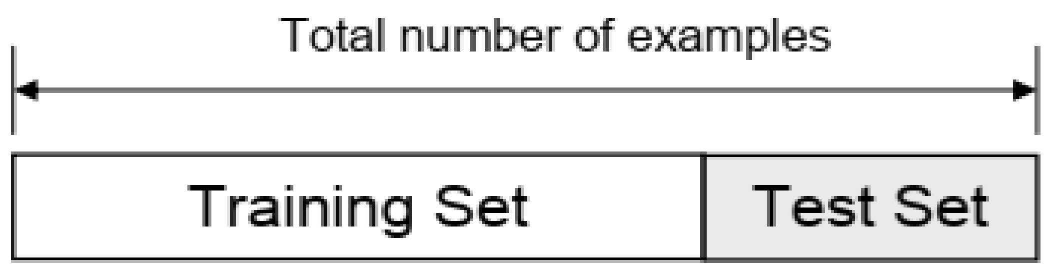
- **Almost invariably, all the pattern recognition techniques that we have introduced have one or more free parameters**
  - The number of neighbors in a kNN Classification Rule
  - The bandwidth of the kernel function in Kernel Density Estimation
  - The number of features to preserve in a Subset Selection problem
- **Two issues arise at this point**
  - **Model Selection**
    - How do we select the "optimal" parameter(s) for a given classification problem?
  - **Validation**
    - Once we have chosen a model, how do we estimate its true error rate?
    - The true error rate is the classifier's error rate when tested on the ENTIRE POPULATION
- **If we had access to an unlimited number of examples, these questions would have a straightforward answer**
  - Choose the model that provides the lowest error rate on the entire population
  - And, of course, that error rate is the true error rate
- **However, in real applications only a finite set of examples is available**
  - This number is usually smaller than we would hope for!
  - Why? Data collection is a very expensive process

Introduction (2)

- **One may be tempted to use the entire training data to select the "optimal" classifier, *then* estimate the error rate**
- **This naïve approach has two fundamental problems**
  - The final model will normally **overfit** the training data: it will not be able to generalize to new data
    - The problem of overfitting is more pronounced with models that have a large number of parameters
  - The error rate estimate will be overly optimistic (lower than the true error rate)
    - In fact, it is not uncommon to have 100% correct classification on training data
- **The techniques presented in this lecture will allow you to make the best use of your (limited) data for**
  - Training
  - Model selection and
  - Performance estimation

The holdout method

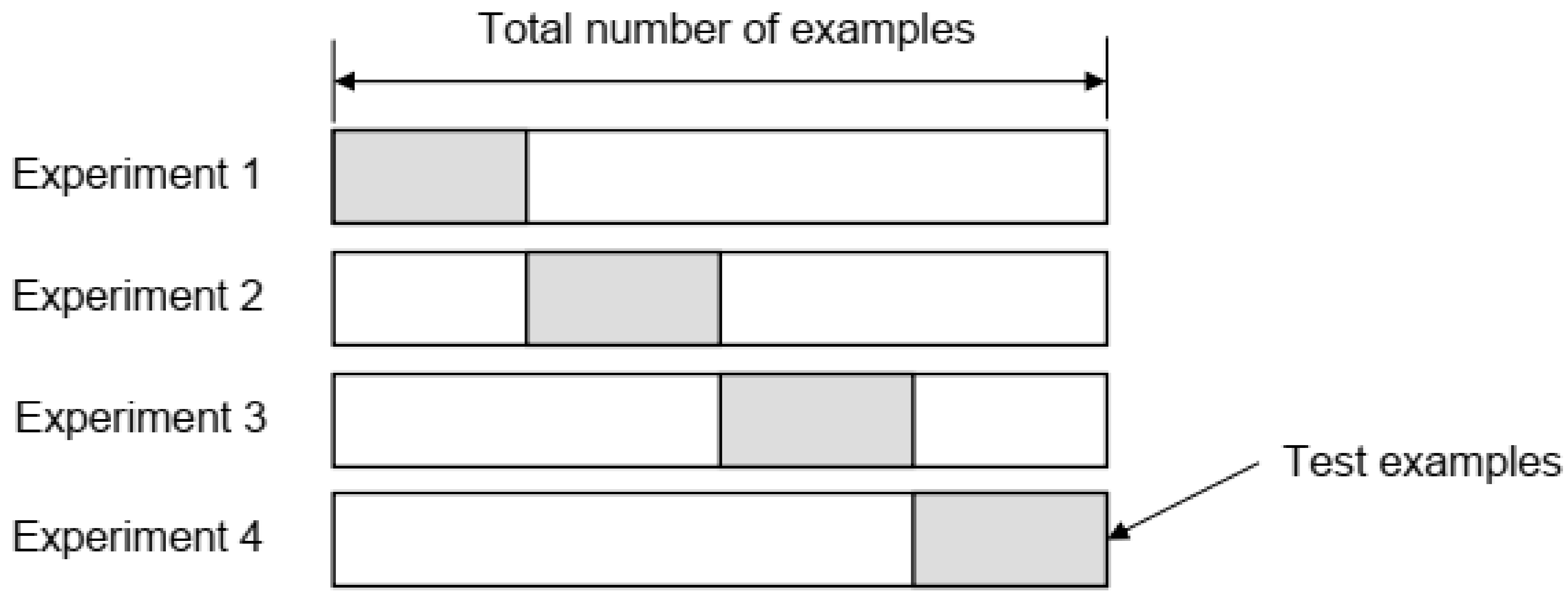
- **Split dataset into two groups**
  - **Training set**: used to train the classifier
  - **Test set**: used to estimate the error rate of the trained classifier



- **The holdout method has two basic drawbacks**
  - In problems where we have a sparse dataset we may not be able to afford the "luxury" of setting aside a portion of the dataset for testing
  - Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an "unfortunate" split
- **The limitations of the holdout can be overcome with a family of re-sampling methods at the expense of higher computational cost**
  - **Cross Validation**
    - K-Fold Cross-Validation **- we will focus on this method**
    - Leave-one-out Cross-Validation
    - Random Subsampling
  - **Bootstrap**

K-Fold Cross-validation

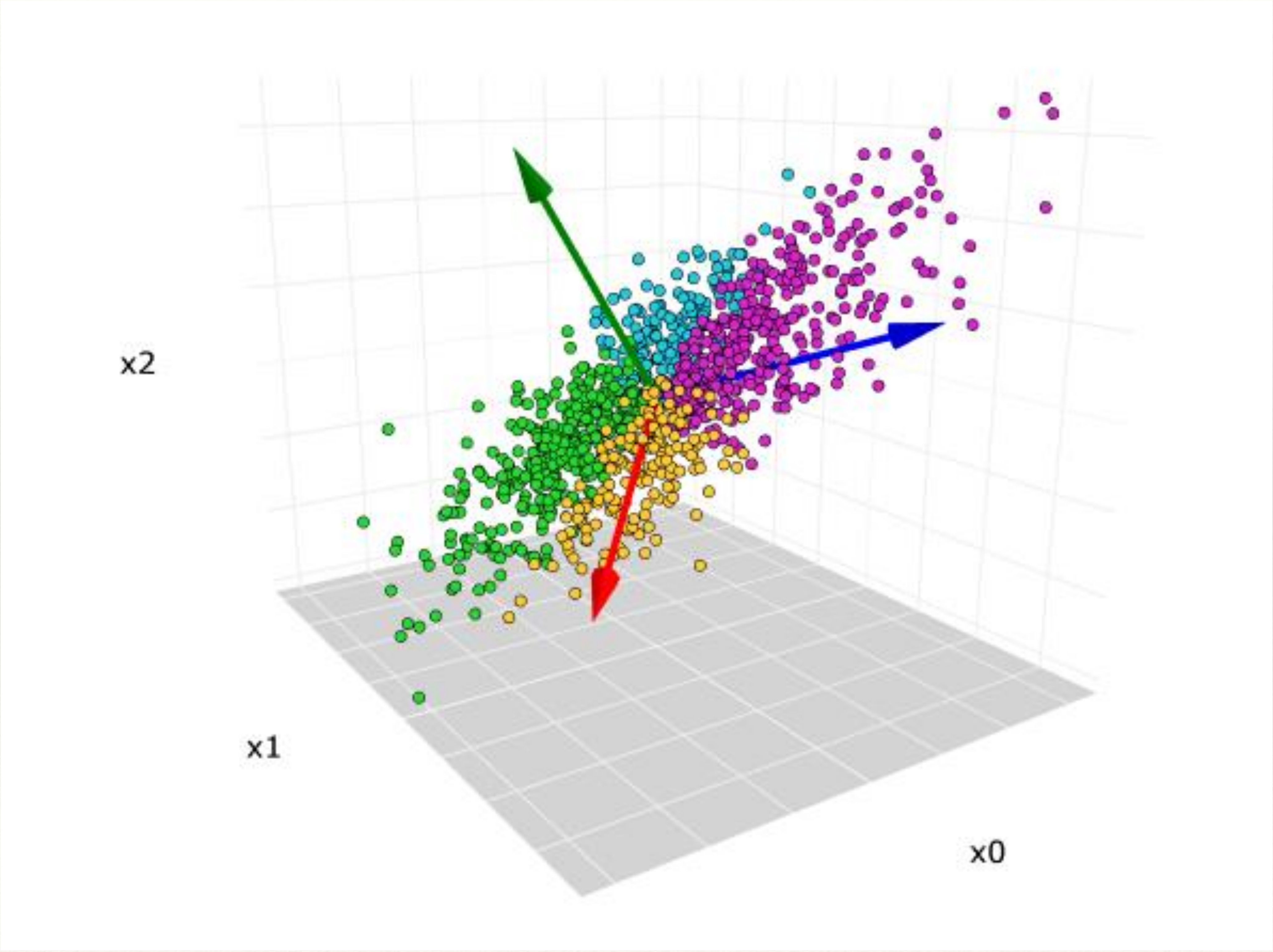
- **Create a K-fold partition of the the dataset**
  - For each of K experiments, use K-1 folds for training and a different fold for testing
    - This procedure is illustrated in the following figure for K=4



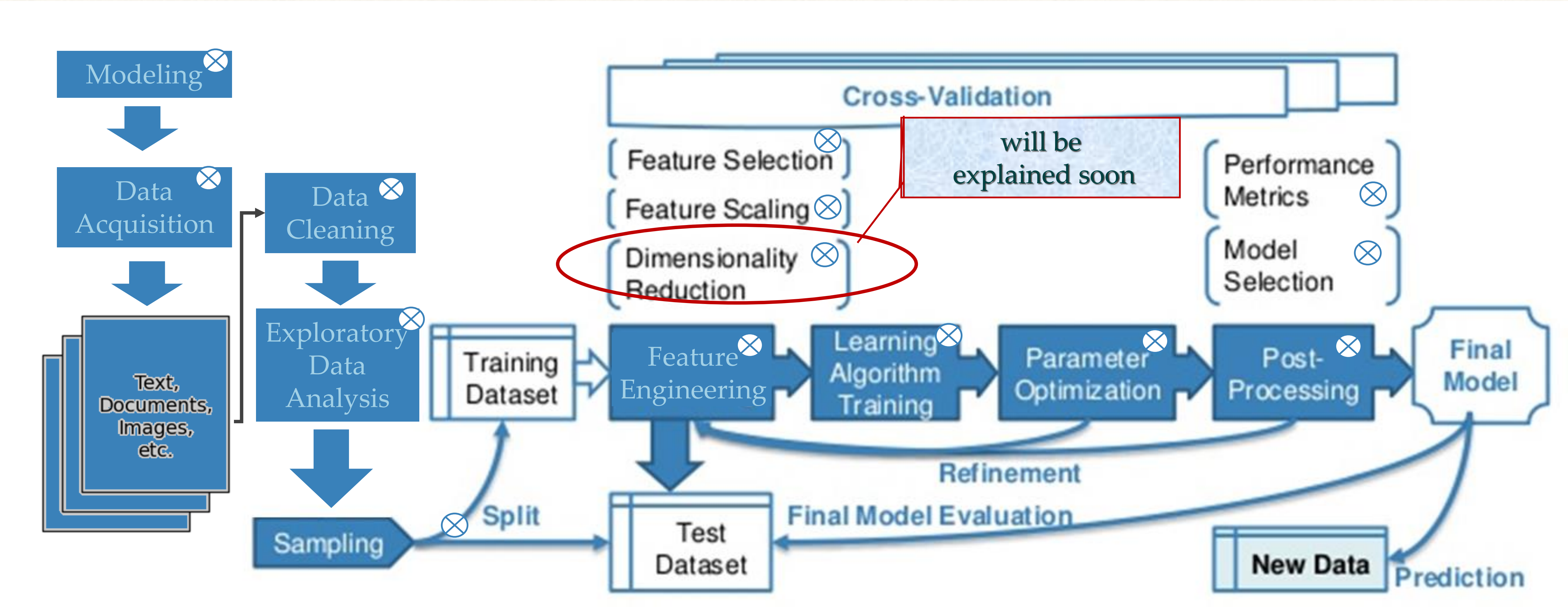
- **K-Fold Cross validation**
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- **the true error is estimated as the average error rate on test examples**

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Dimensionality Reduction - PCA



A typical ML flow - diving in



Data Cleaning

→ Duplications

→ Missing Data

→ Outlier Detection

Train-Test split

+ Validation-set

Data Exploration

→ Statistical and descriptive info

→ Visualization

→ Effecting other steps

Feature Engineering

→ Feature Scaling

→ Feature Selection

→ Dimensionality reduction

Unsupervised Learning algo.

→ PCA

Supervised Learning algo.

→ KNN

→Decision Trees

→Naïve Bayes

Classification Evaluation

→Confusion matrix

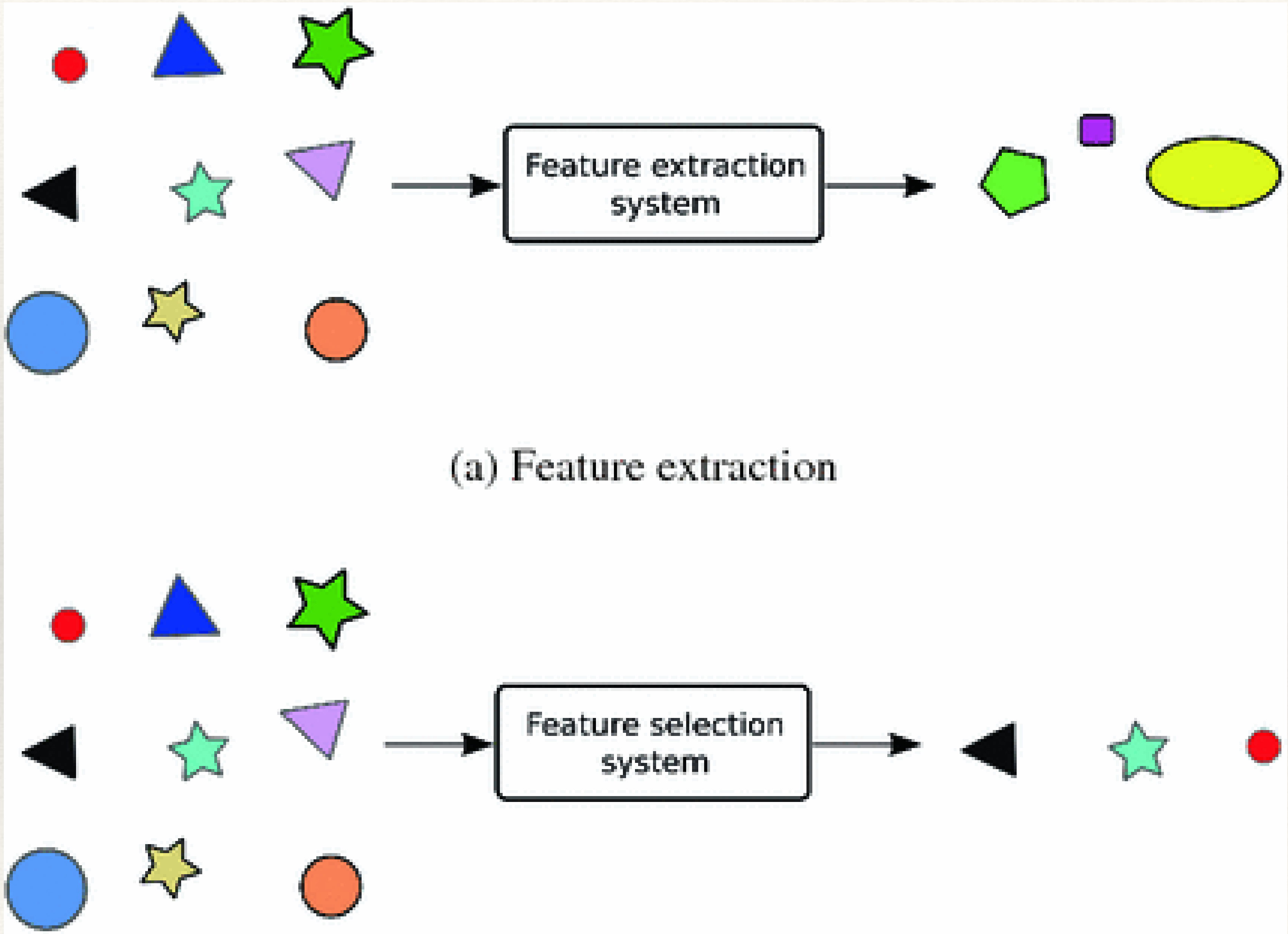
→ Accuracy ,Error (rate)

→ Precision, Recall

→ F1

→ Cross-validation evaluation

Feature selection vs Dimensionality reduction - reminder



(a) Feature extraction

(b) Feature selection

הורדת מימדים – PCA – תזכורת

- PCA – Principal component analysis
- נתונות לנו n דוגמאות במימד d
- נרצה למצוא יצוג לכל הדוגמאות במימד נמוך יותר ($k<d$)
- האמצעי: קומבינציות לינאריות של המאפיינים
  כלומר: הטלה ממימד d למימד k

$$\vec{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}) \qquad z_{i,j} = \vec{w}_j \cdot \vec{x}_i$$

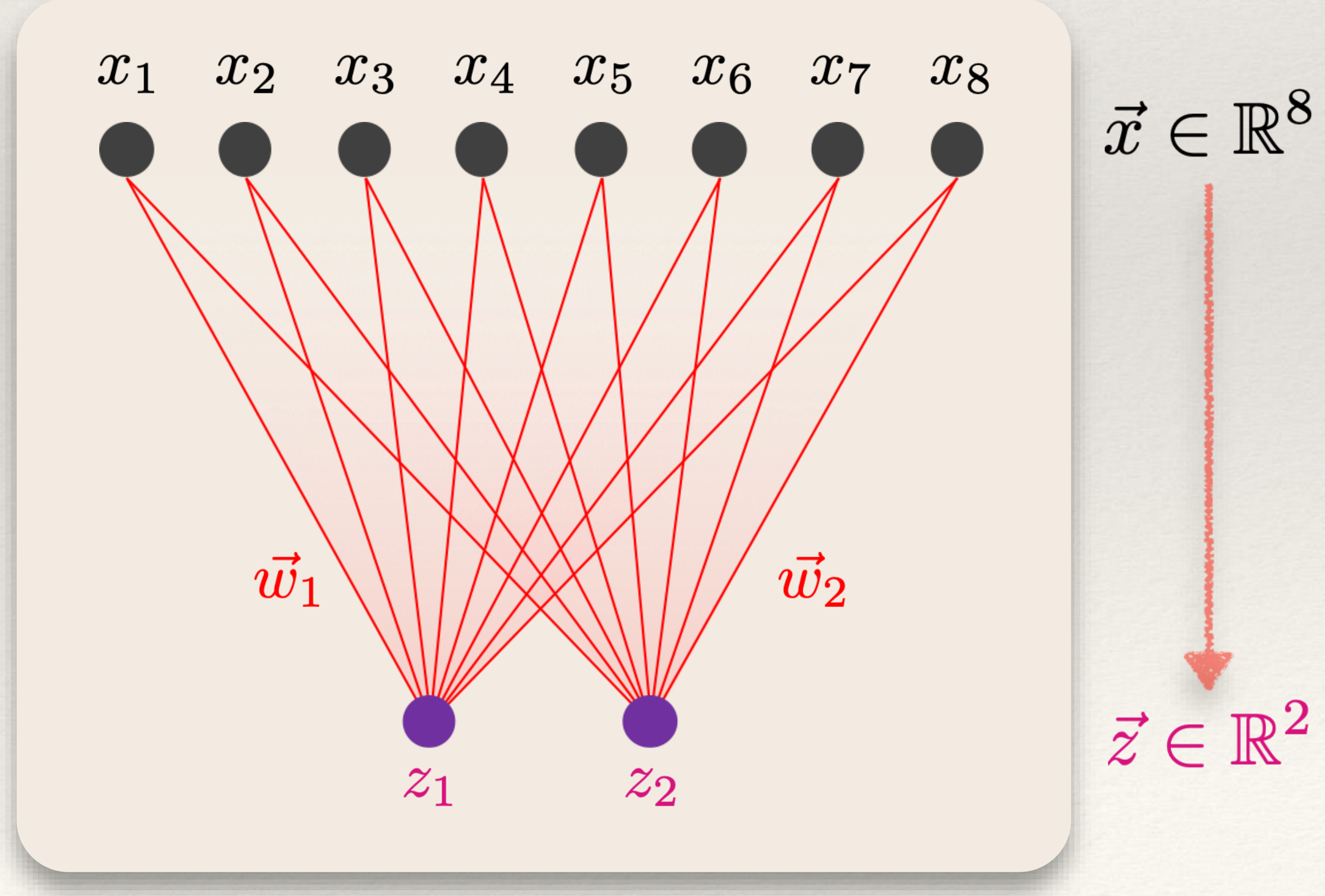
$$\vec{z}_i = \left(\vec{w}_1 \cdot \vec{x}_i \,,\, \vec{w}_2 \cdot \vec{x}_i \,,\, \ldots \,,\, \vec{w}_k \cdot \vec{x}_i\right) = (z_{i,1}, z_{i,2}, \ldots, z_{i,k})$$

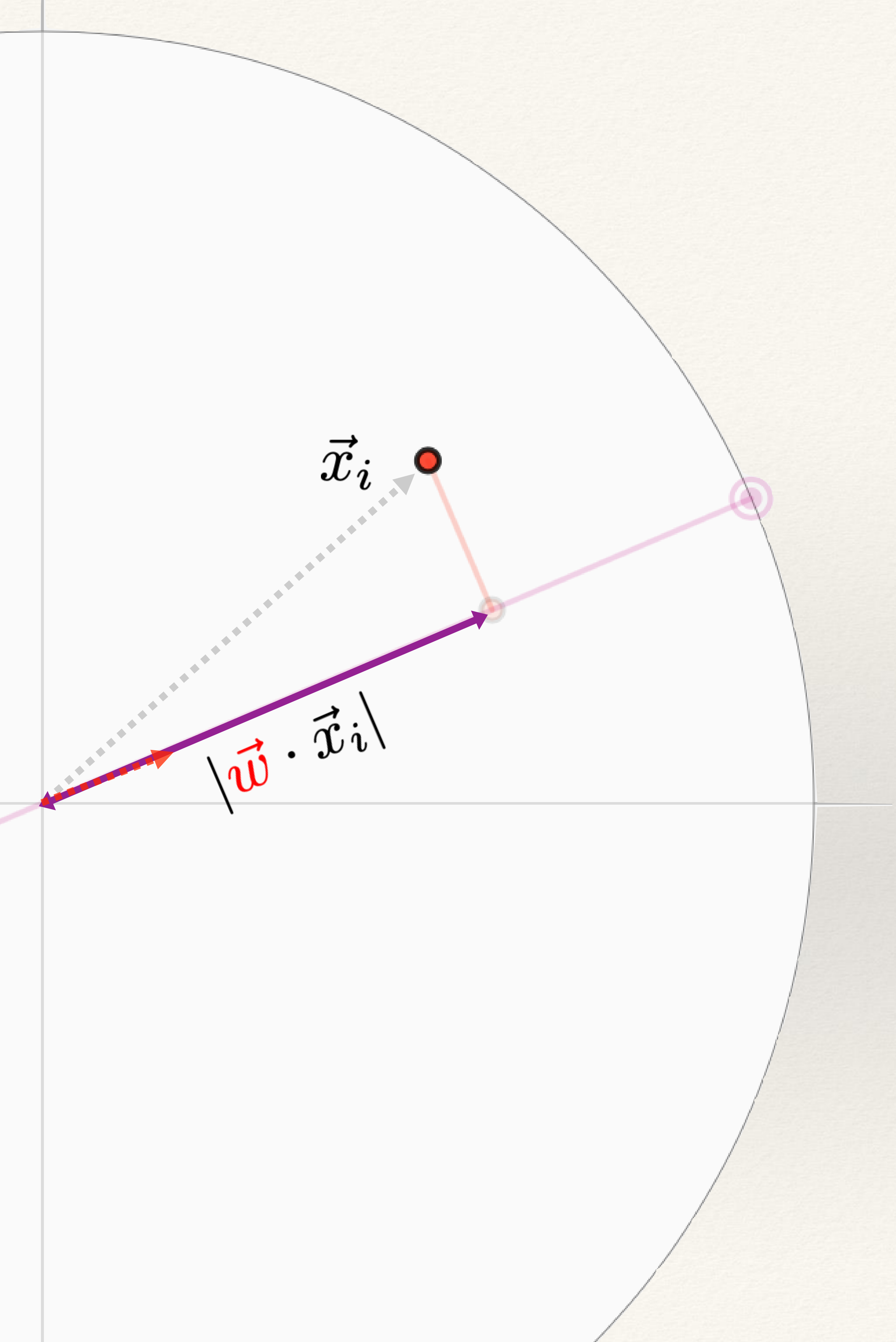
הורדת מימדים – PCA – תזכורת

- מחפשים ליצג את $\vec{x} \in \mathbb{R}^d$ באמצעות $\vec{z} \in \mathbb{R}^k$
- ע״י שימוש בקומבינציות ליניאריות $\vec{w}_1, \dots, \vec{w}_k$ של המאפיינים.

ש: איך נבחר את $\vec{w}_1, \dots, \vec{w}_k$?

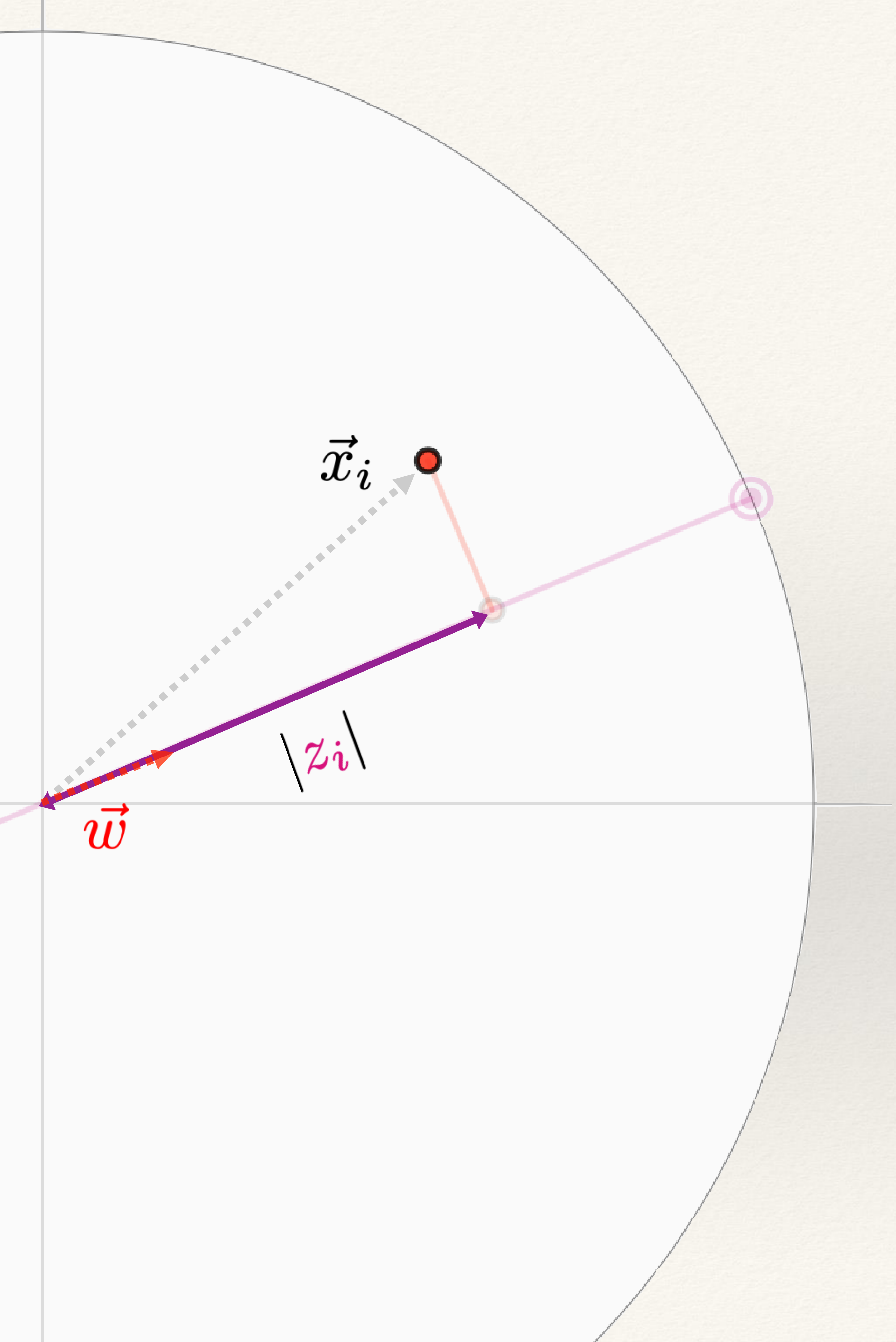
ת: שגיאת שחזור מינימלית.





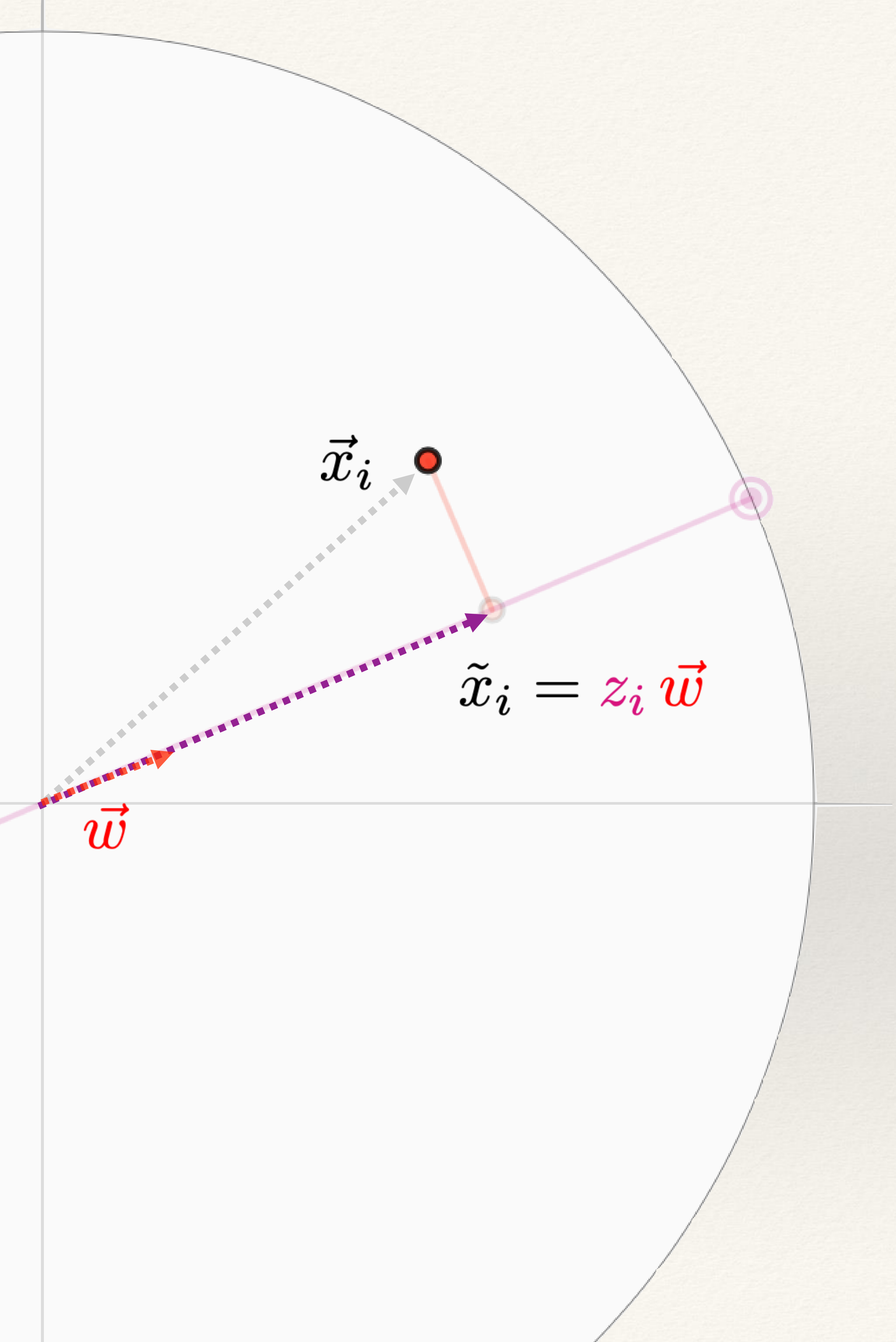
$$\|\vec{w}\|^2 = 1$$

$$z_i = \vec{w} \cdot \vec{x}_i$$



$$\|\vec{w}\|^2 = 1$$

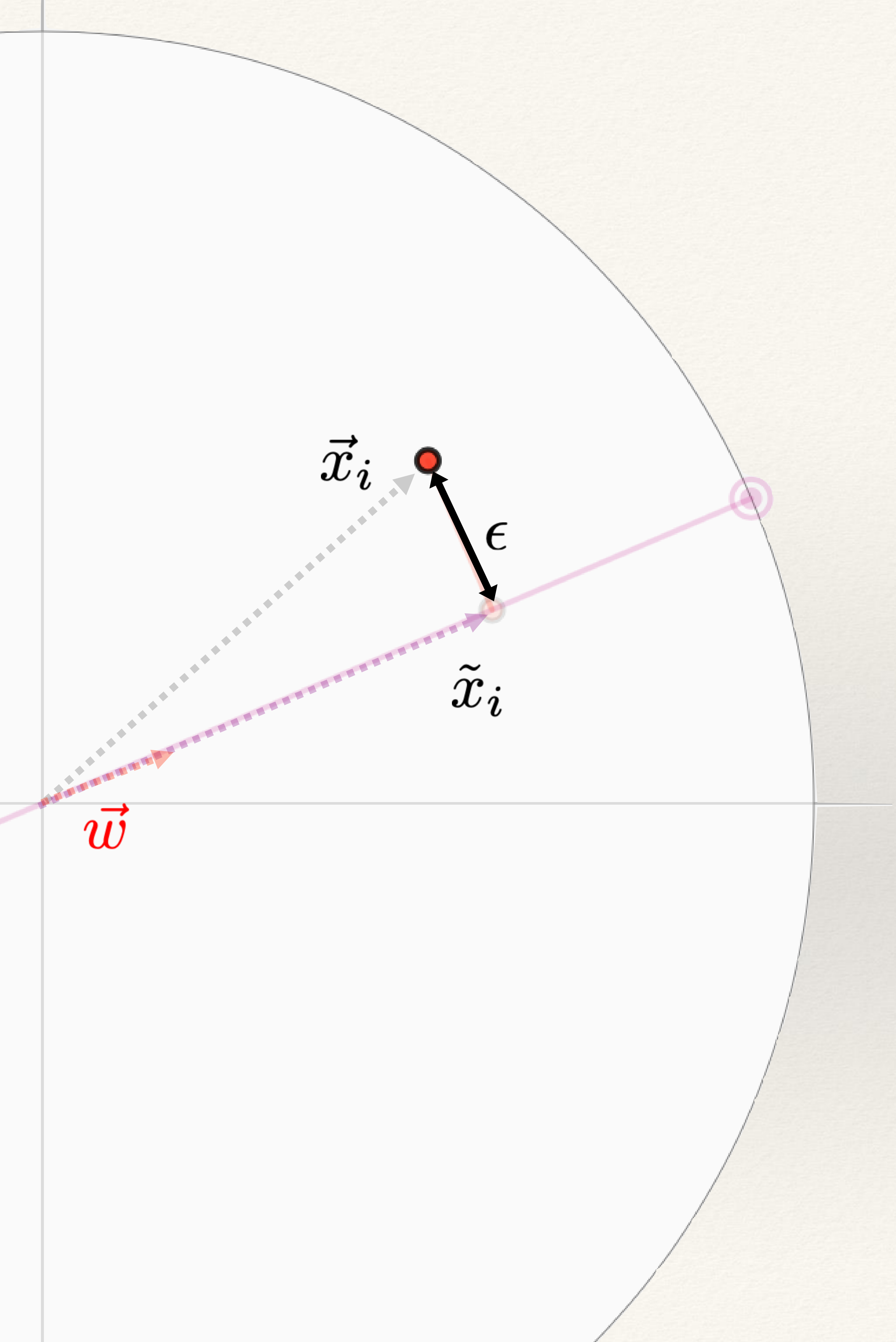
$$z_i = \vec{w} \cdot \vec{x}_i$$



$$\left\|\vec{w}\right\|^2 = 1$$

$$z_i = \vec{w} \cdot \vec{x}_i$$

$$\tilde{x}_i = z_i\,\vec{w}$$



$$\|\vec{w}\|^2 = 1$$

$$z_i = \vec{w} \cdot \vec{x}_i$$

$$\tilde{x}_i = z_i \, \vec{w}$$

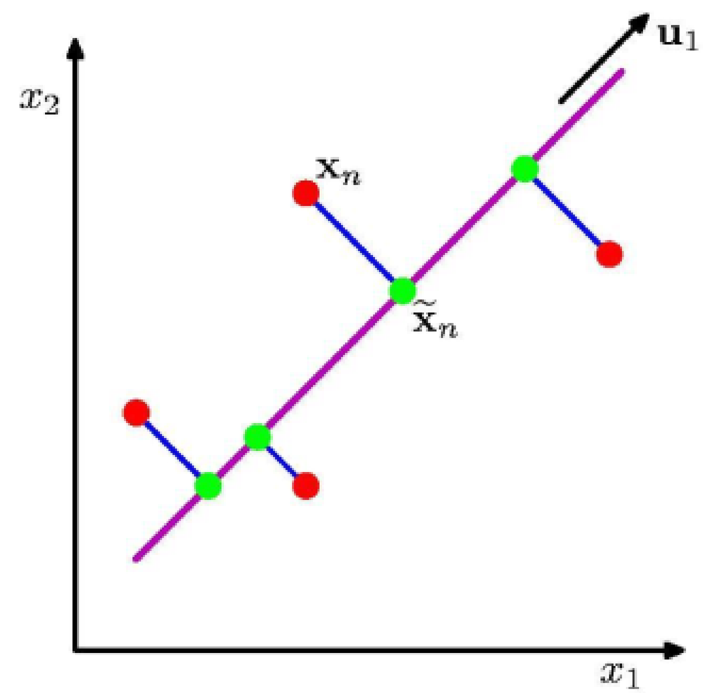
$$\epsilon\left(\vec{x}_i\right) := \left\|\vec{x}_i - \tilde{x}_i\right\|^2$$

שגיאת שיחזור ריבועית

PCA: Motivation - reminder

- Choose directions such that a total variance of data will be maximum
  - Maximize Total Variance
- Choose directions that are orthogonal
  - Minimize correlation
- Choose $k<d$ orthogonal directions which maximize total variance

PCA: Motivation - reminder



PCA:

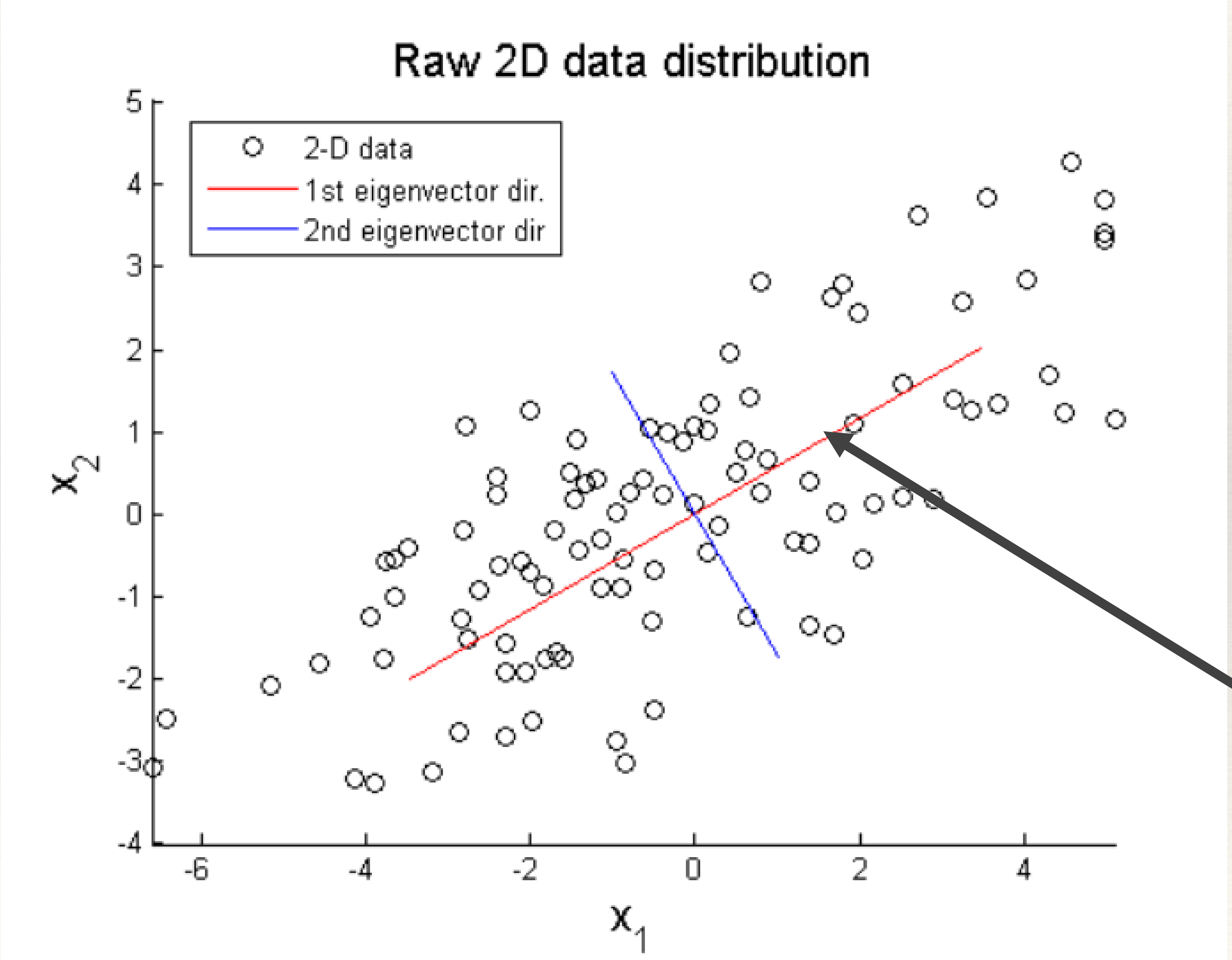
Orthogonal projection of the data onto a lower-dimension linear space that...

- maximizes variance of projected data (purple line)

- minimizes the mean squared distance between
  - data point and
  - projections (sum of blue lines)

PCA 2D example

PCA is implemented to project one hundred of 2-D data $X \in R_{2x100}$ *on 1-D space.*

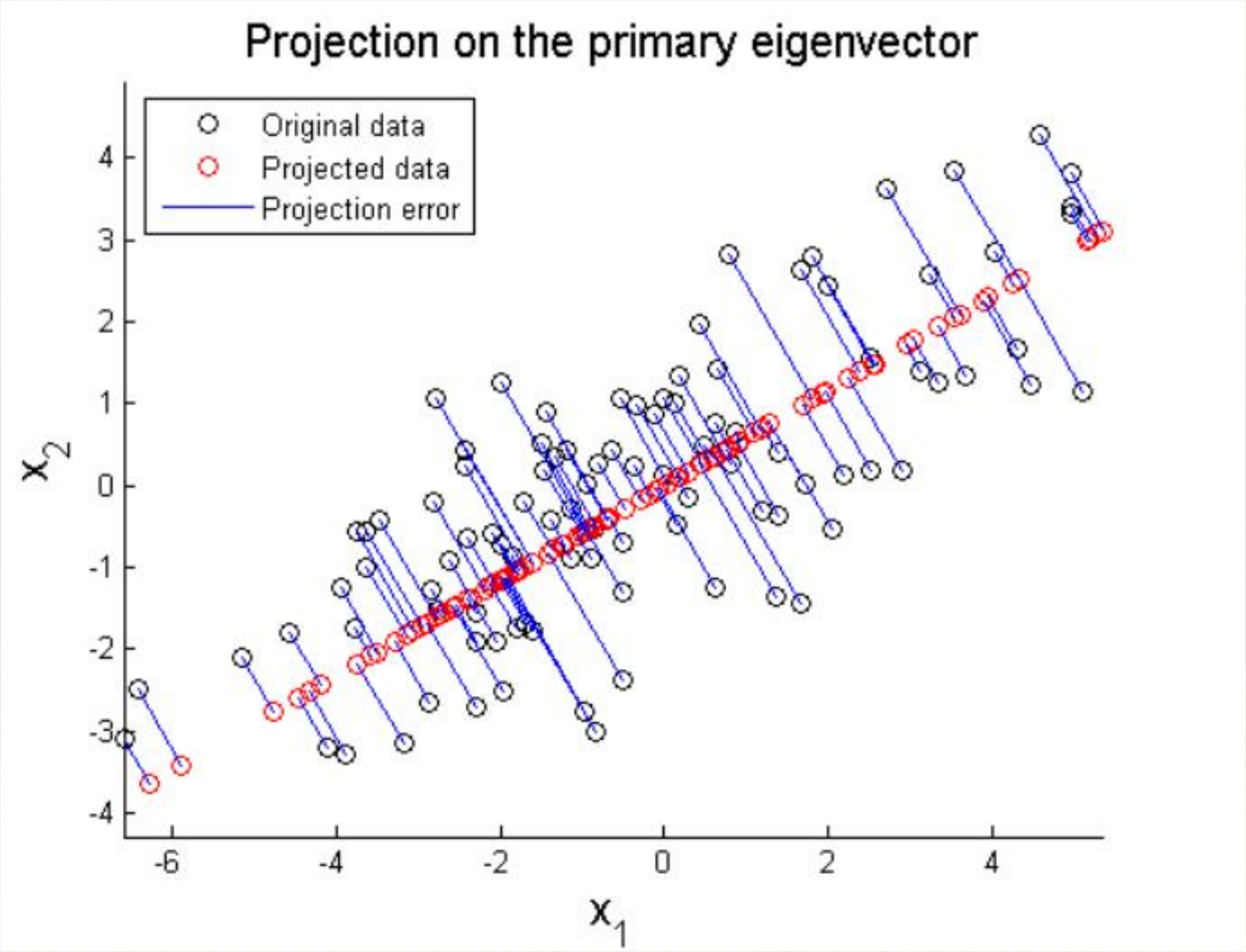


1st 2D eigenvector
Maximal projected variance

https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

PCA 2D example

Project to the first eigenvector to reduce the dimension *to 1-D space.*



https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

PCA example on image

In this example, PCA is applied for the compression of 512-by-512 grey-scale image

https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

PCA example on image

first 30 eigenvalues

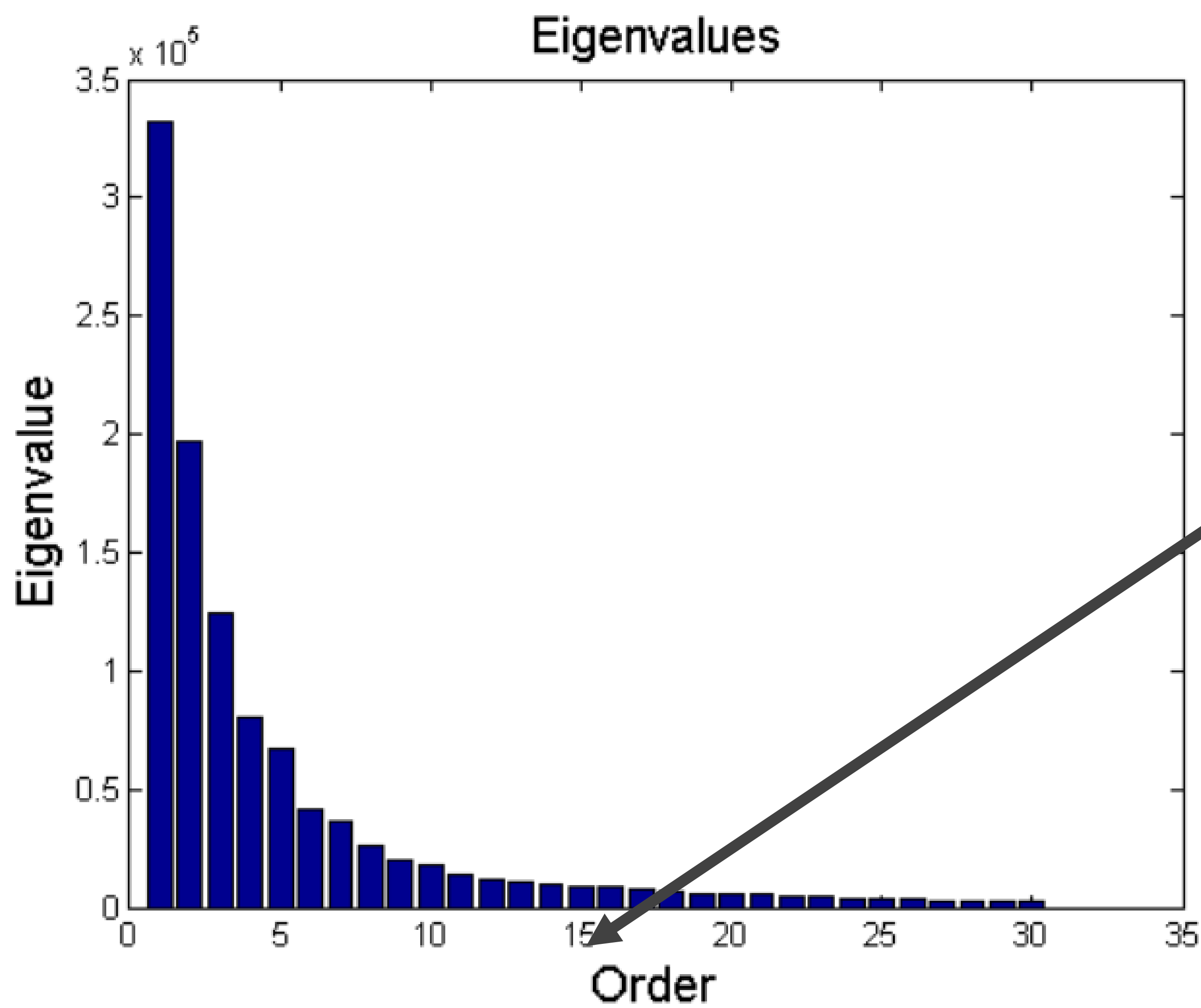


Figure 6. Ordered eigenvalues.

How many eigenvectors to use?

https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

PCA example on image

(a) 1 principal component
(b) 5 principal component
(c) 9 principal component
(d) 13 principal component
(e) 17 principal component
(f) 21 principal component
(g) 25 principal component
(h) 29 principal component

https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

How to choose k – explained

- Proportion of Variance (PoV) explained

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

*The top **k** eigen values*

All *eigen values*

when λ_i are sorted in descending order

- Typically, stop at PoV>0.9
- Screen graph plots of PoV vs *k*, stop at "elbow"