



ThAIMed Initiative Session 2

*Introduction to Machine Learning/AI for
healthcare professionals*

12th July 2024
Liam Barrett

Session 1 Recap

- Introduction to data types, structure, and representation
- Data quality and data cleaning
- Data transformation
- Feature selection/reduction
- Used Google Colab Notebooks to analyse and visualise clinical data

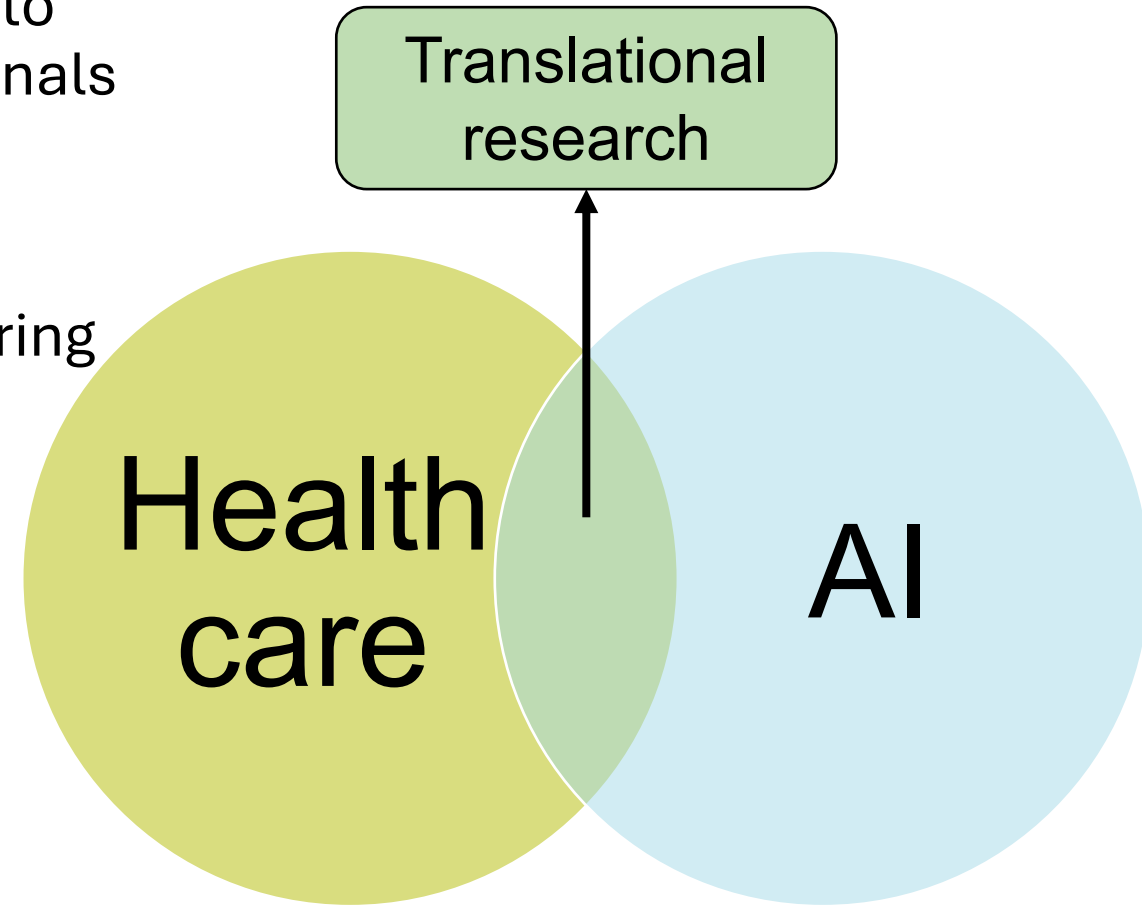
Session 2 Overview

1. Introduction to model training
2. Running model training on clinical dataset (practical)
3. Introduction to model evaluation
4. Evaluating our models from section 1. (practical)
5. Summary
6. Q&A

The practical sessions
follow a real journal
publication using the same
data!
(An Dinh et al., 2019)

Importance of translational input

- Machine learning and AI have huge potentials to improve service for both health care professionals and patients
- It also has a large potential to go wrong
- Multidisciplinary experts (you) are key to ensuring actual improvements are delivered
- AI is highly susceptible to misalignment
I.e., your model



Session 2 Overview

1. Introduction to model training
2. Running model training on clinical dataset (practical)
3. Introduction to model evaluation
4. Evaluating our models from section 1. (practical)
5. Summary
6. Q&A

1. Introduction to model training

- Machine learning is a subfield of artificial intelligence that focuses on algorithms that can automatically learn patterns and relationships from data and use this knowledge to make predictions or decisions.
- Key components
 - Data
 - Algorithms
 - Models

1.1 Supervised vs. unsupervised learning

Supervised

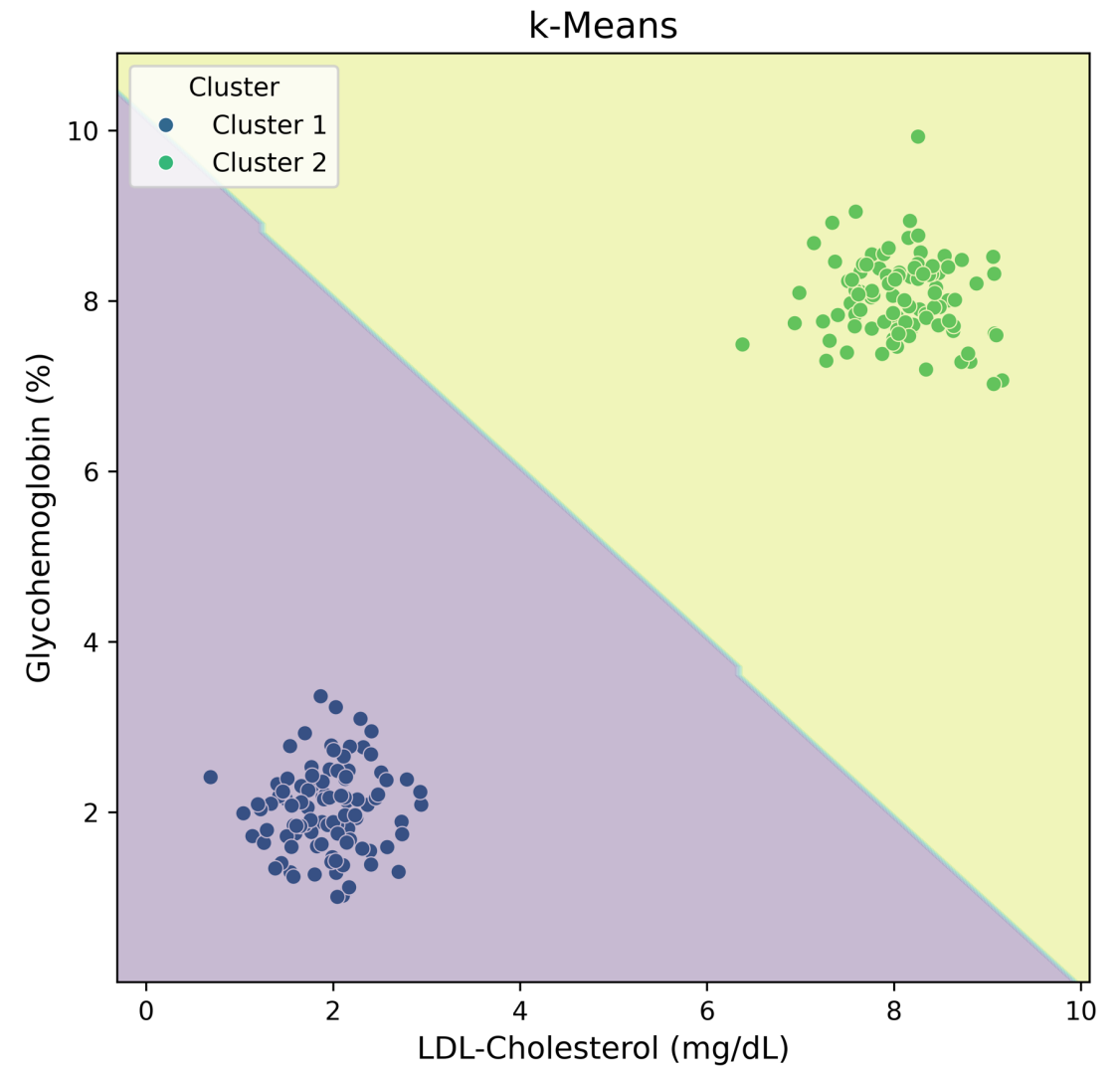
- Learns from labelled data,
- The ground truth (target) is known
- The goal is to learn a mapping function from input features to output label
- Examples include predicting disease outcomes based on patient data.

Unsupervised

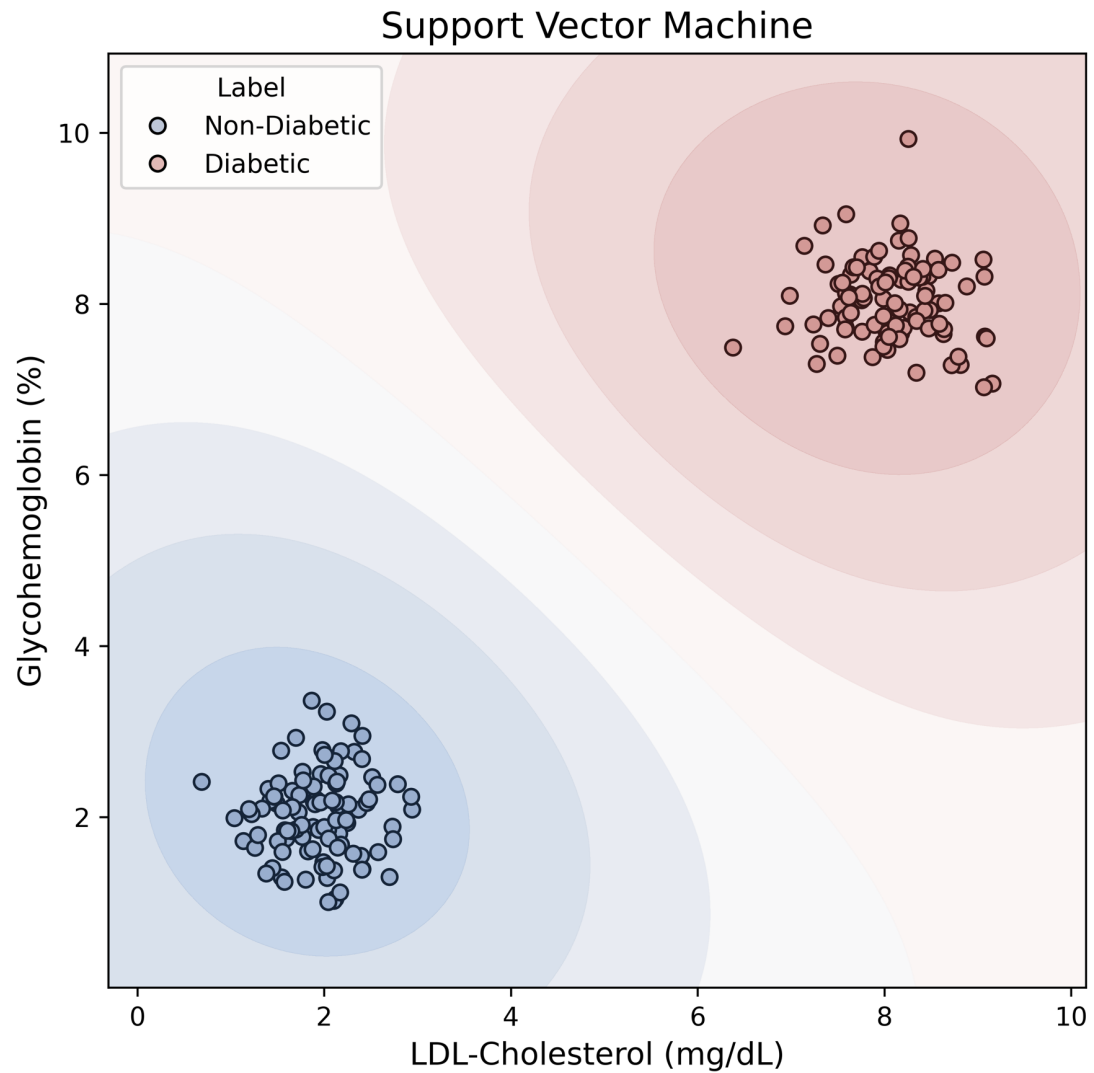
- Learns from unlabelled data
- No target values are provided – ground truth not always known
- The goal is to discover hidden structures or patterns in the data
- Examples include clustering patients with similar characteristics or identifying anomalies in medical images.

Supervised

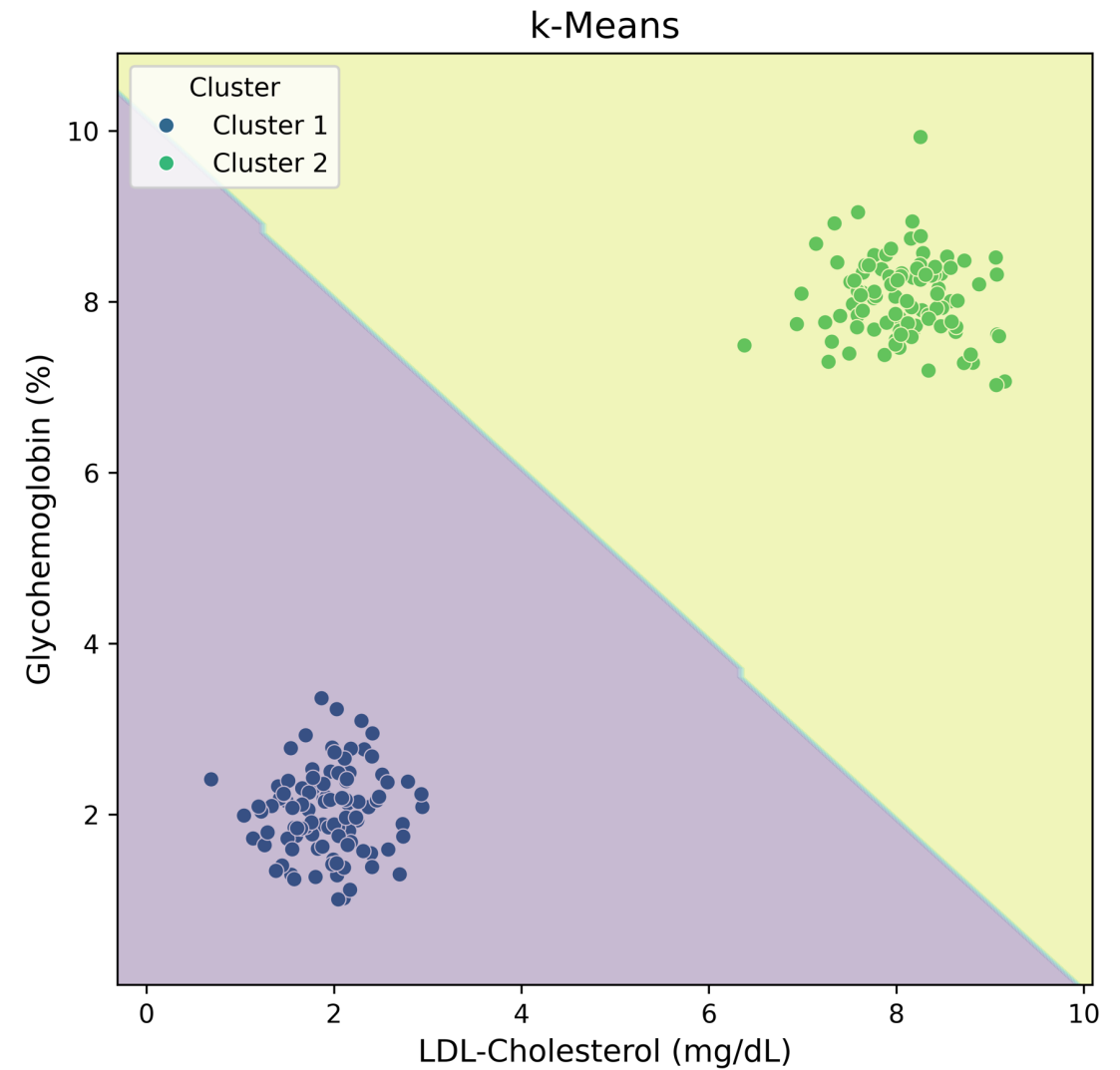
Unsupervised



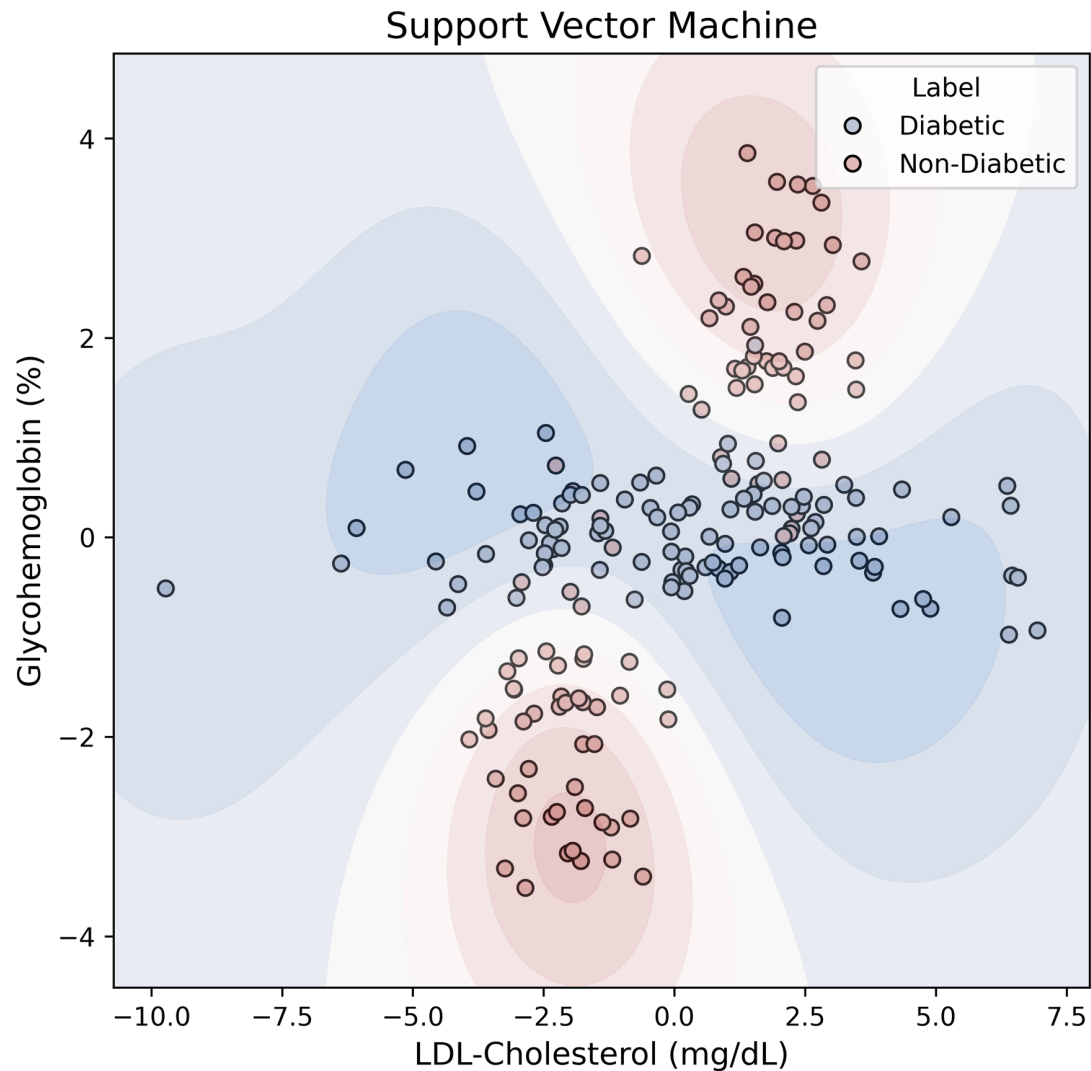
Supervised



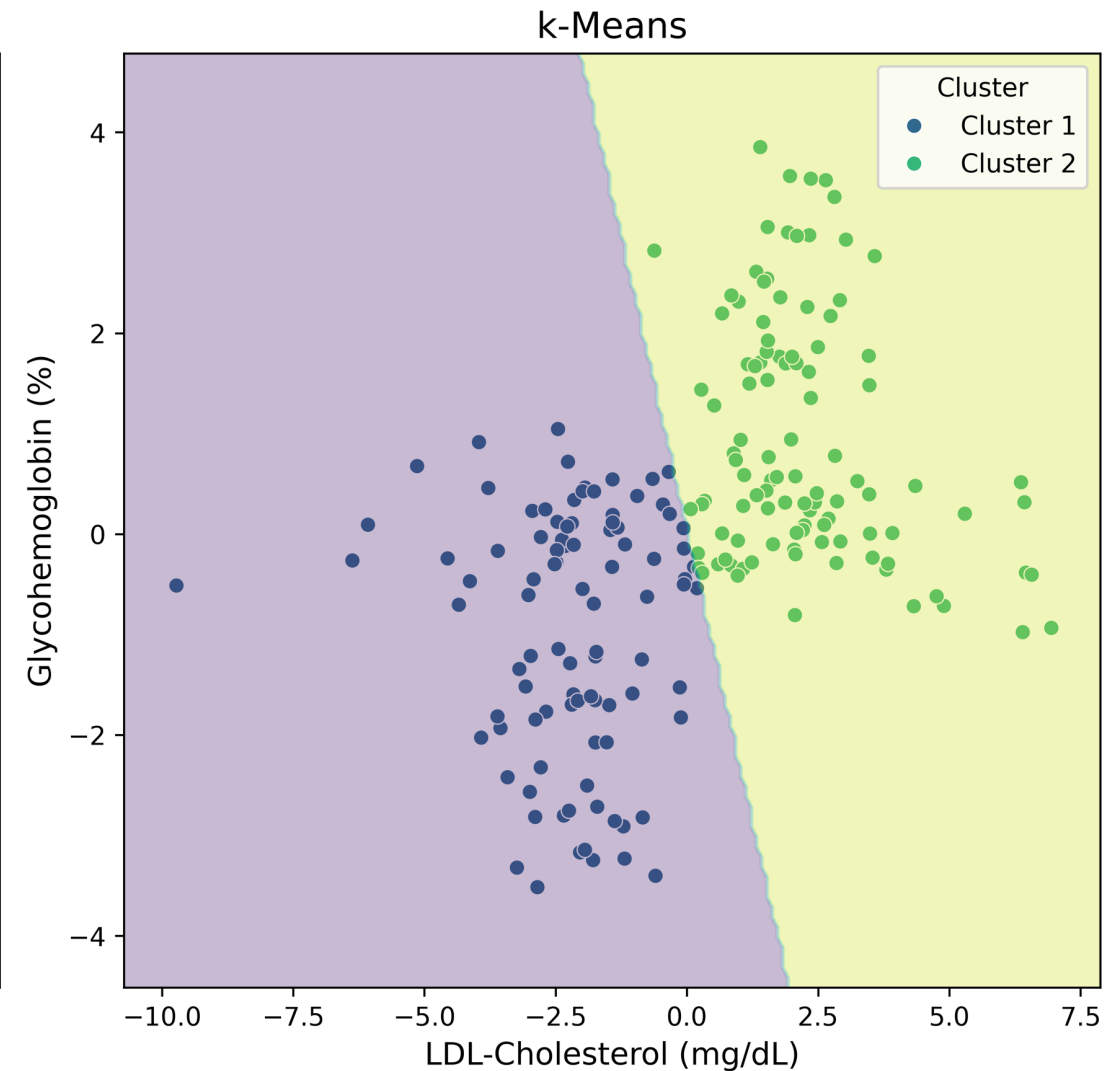
Unsupervised



Supervised



Unsupervised



1.2 Classification vs. regression

Classification

- The goal is to predict a categorical target variable
- The model learns to assign input instances to predefined categories
- Examples include classifying tumours as benign or malignant

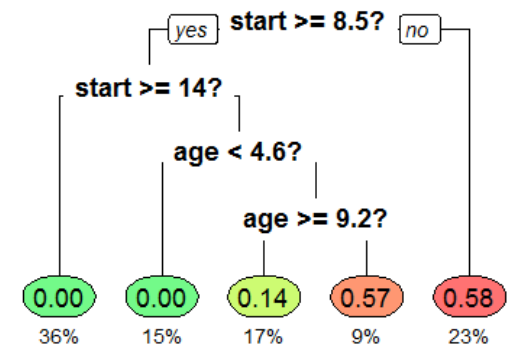
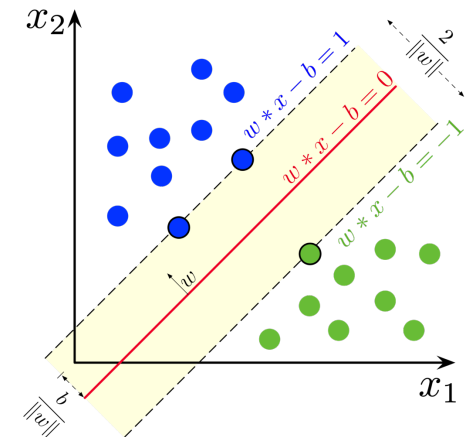
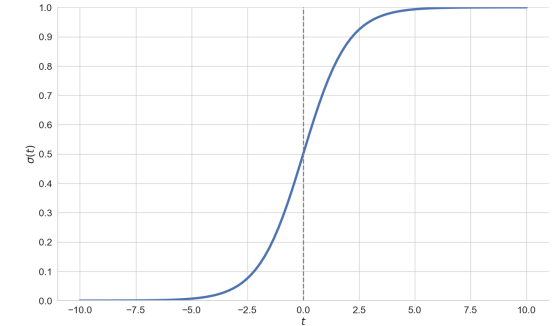
Regression

- The goal is to predict a continuous target variable
- The model learns to estimate a numerical value based on input features
- Examples include predicting patient survival time

Both are for supervised learning problems

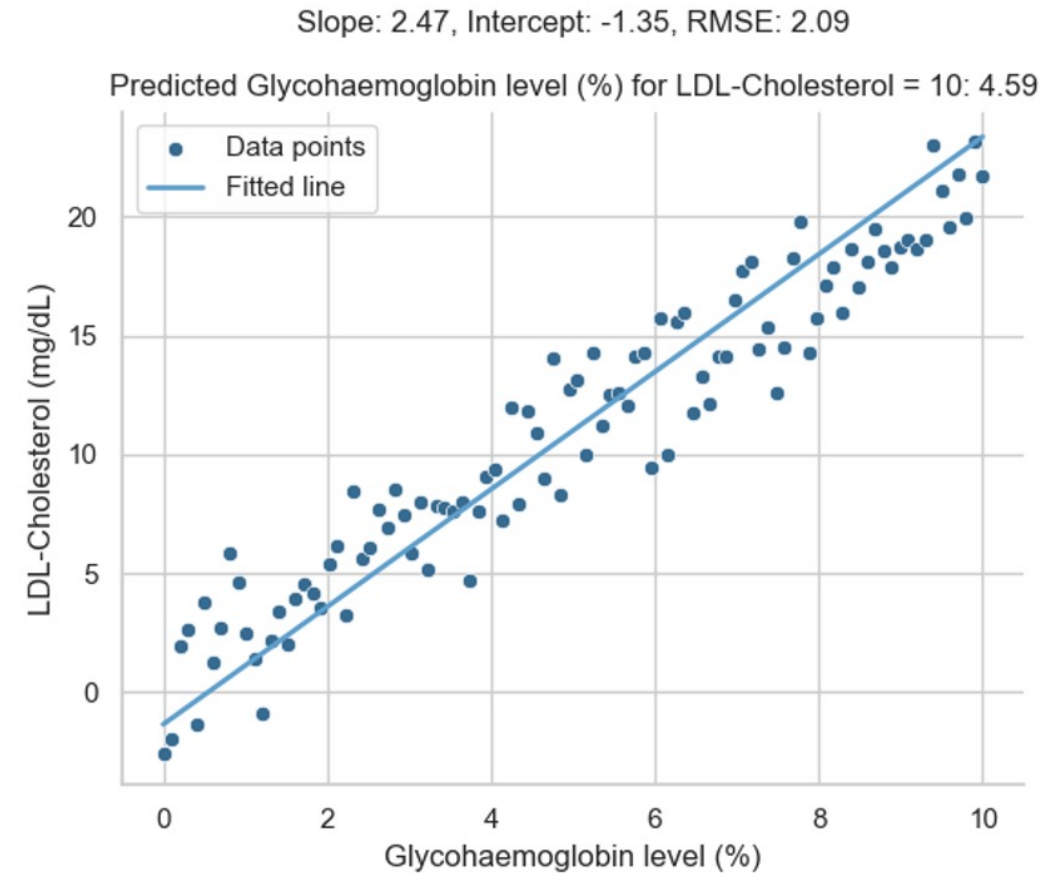
1.3 Supervised Learning

- The objective of supervised learning is to learn a function that maps input features to their corresponding output labels.
- The learned function, known as a model, should be able to predict the correct output label for new, unseen input data.
- The model learns by example, generalizing from the labeled training data to make predictions on new data.
- Classic examples of supervised learning models include,
 - Logistic Regression
 - Support Vector Machines
 - Decision Tree's



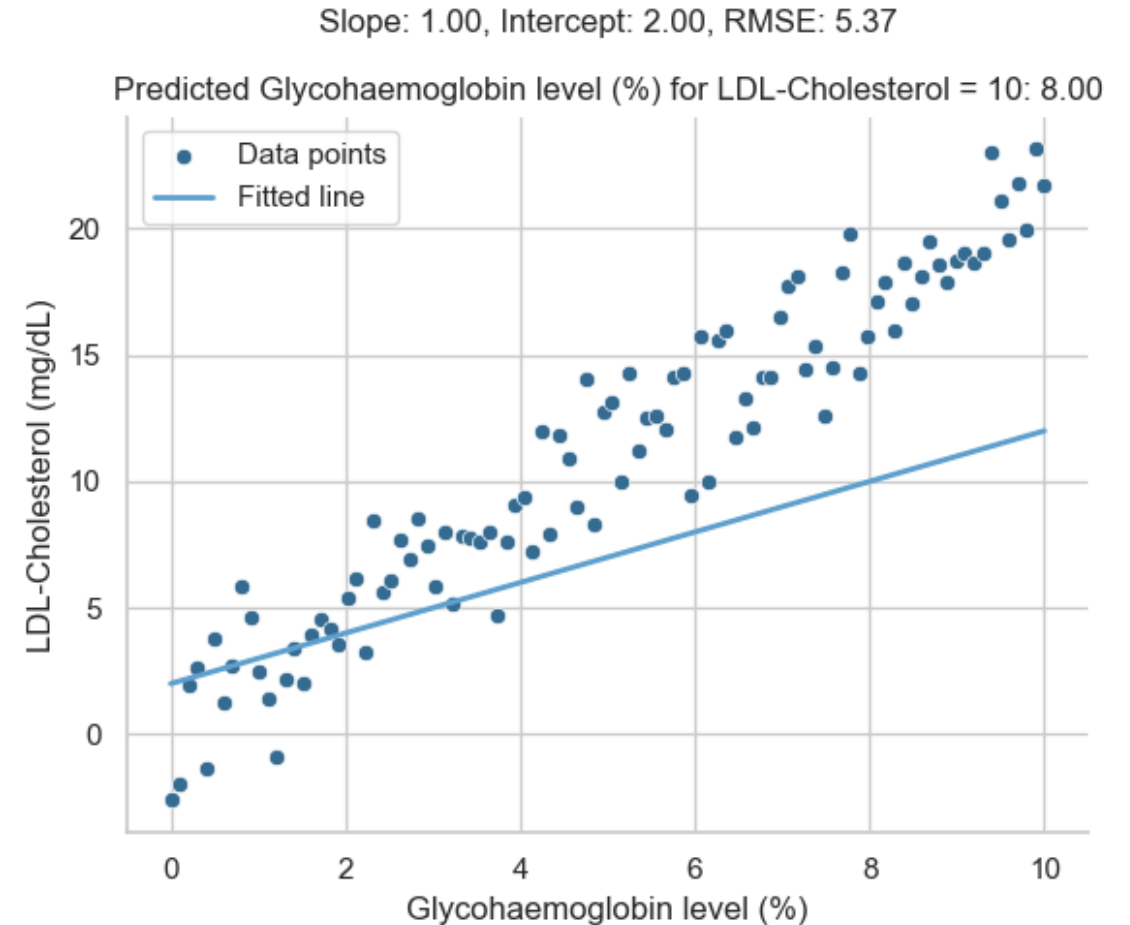
1.4 Training Process

- Assuming your data has been prepared as in session 1...
- During training, the model's parameters are iteratively adjusted to minimize the difference between its predictions and the true labels
- This process is repeated until the model's performance converges or a stopping criterion is met

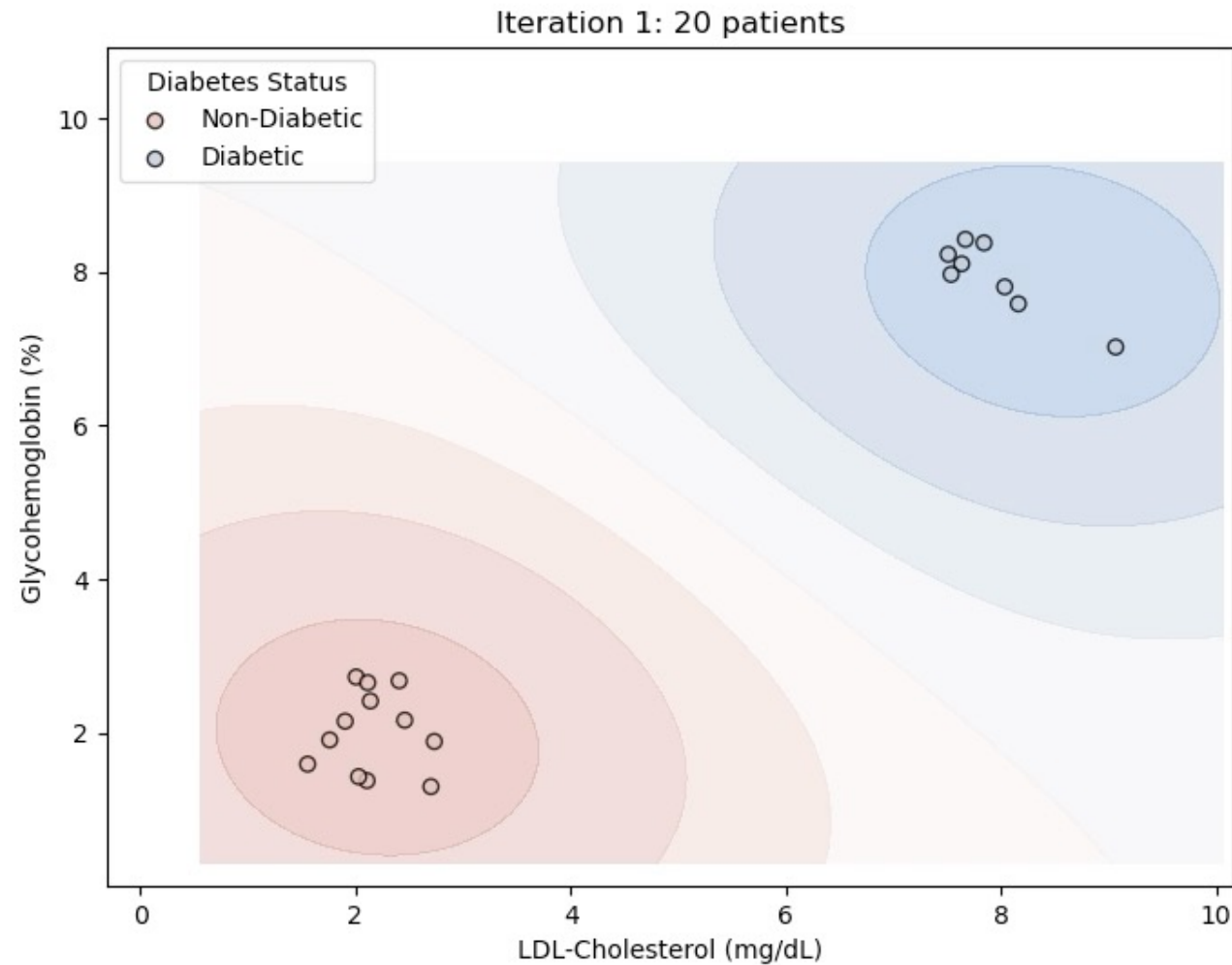


1.4 Training Process

- Assuming your data has been prepared as in session 1...
- During training, the model's parameters are iteratively adjusted to minimize the difference between its predictions and the true labels
- This process is repeated until the model's performance converges or a stopping criterion is met



1.4 Training Process



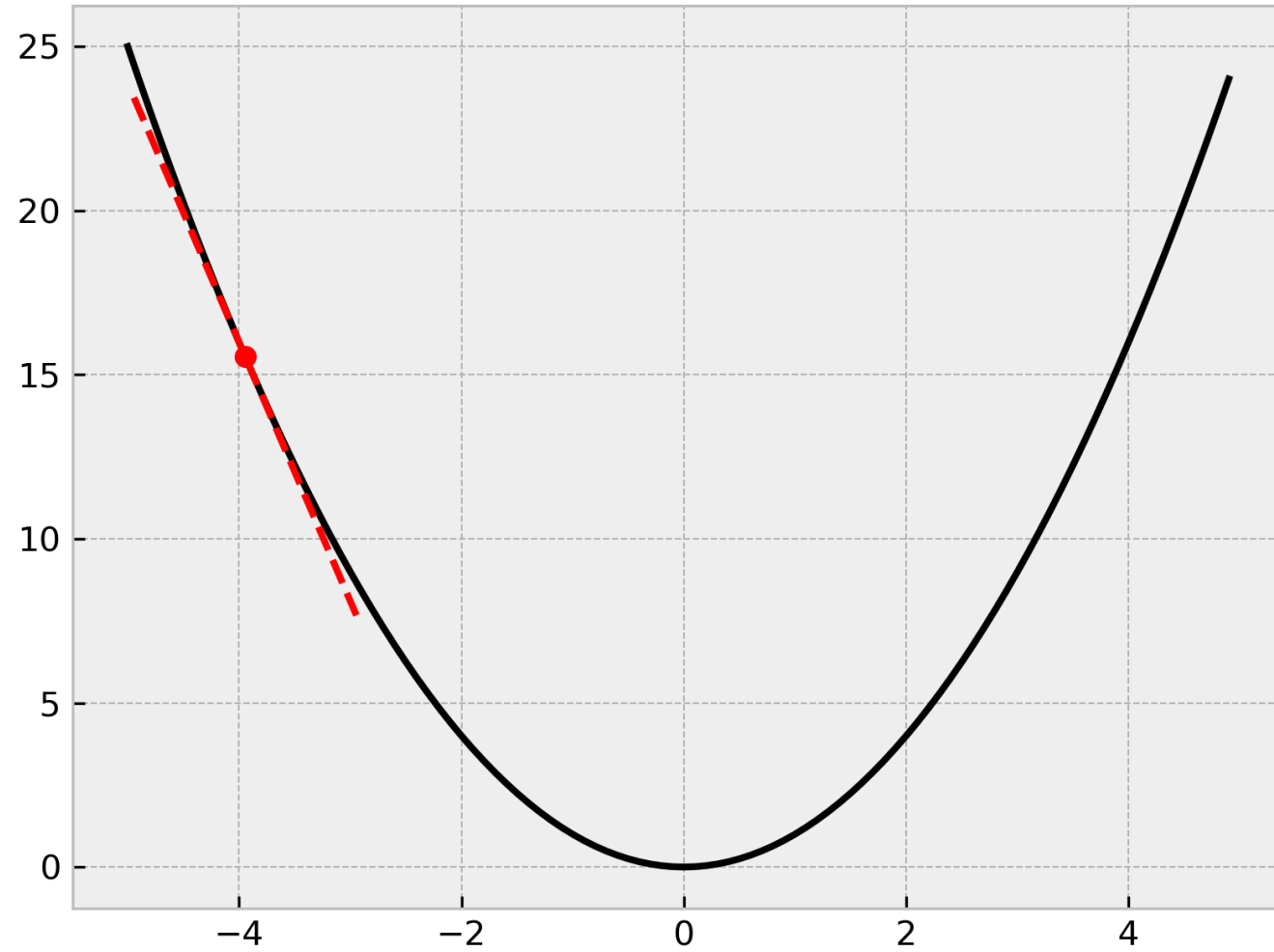
1.5 Loss Functions

- Quantifying the difference between predicted and actual outputs
- Loss functions measure the discrepancy between the model's predictions and the true labels.
- The choice of loss function depends on the task and the specific problem
- The goal of training is to minimize the loss function, indicating that the model's predictions are getting closer to the ground truth.

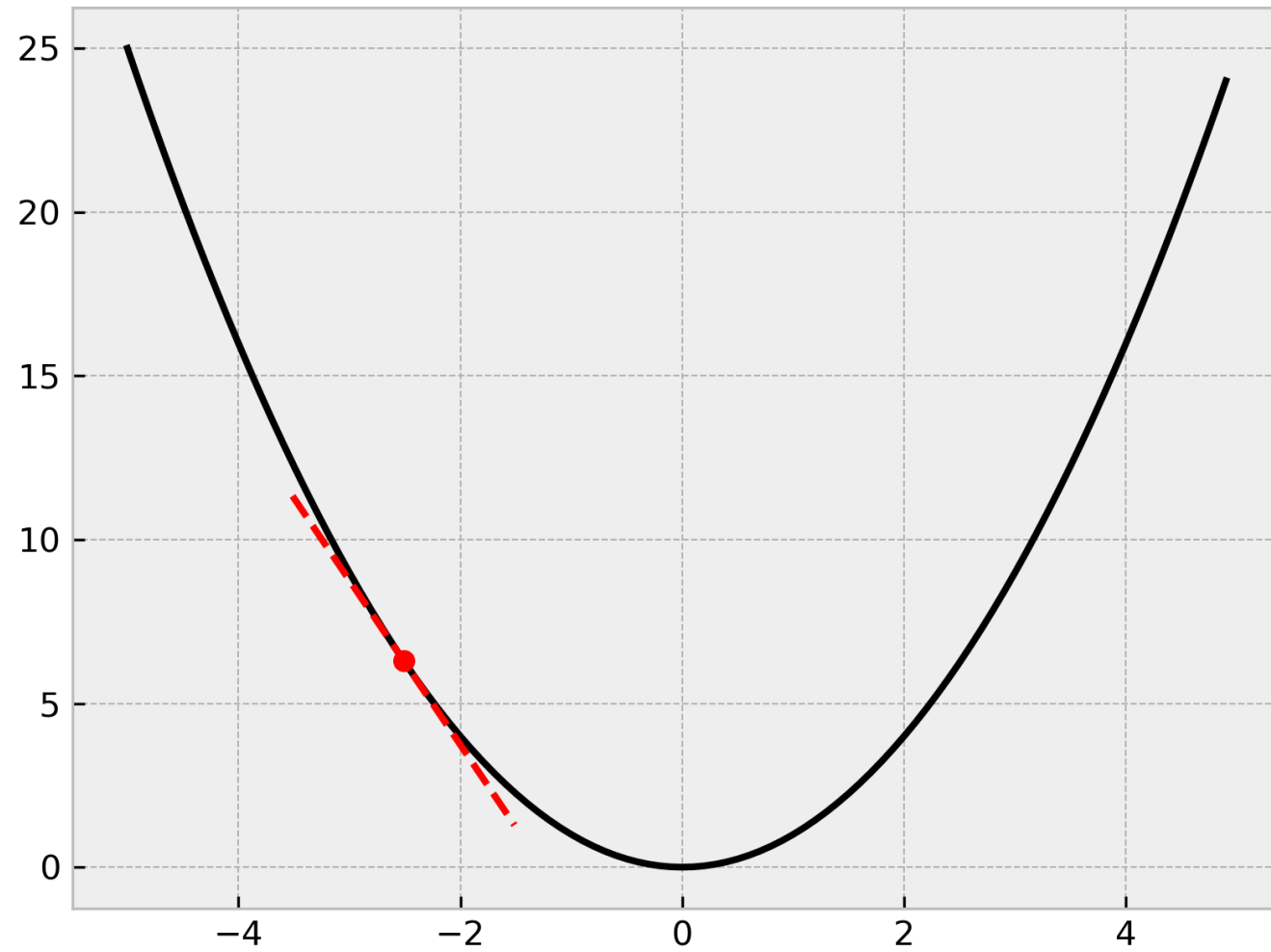
1.6 Gradient Descent

- Optimization algorithm for minimizing the loss function
- The process of iteratively adjusting model parameters to find the minimum of the loss function
- Intuition: Taking steps in the direction of steepest descent to reach the minimum
- Learning rate: Determines the size of steps taken during each iteration

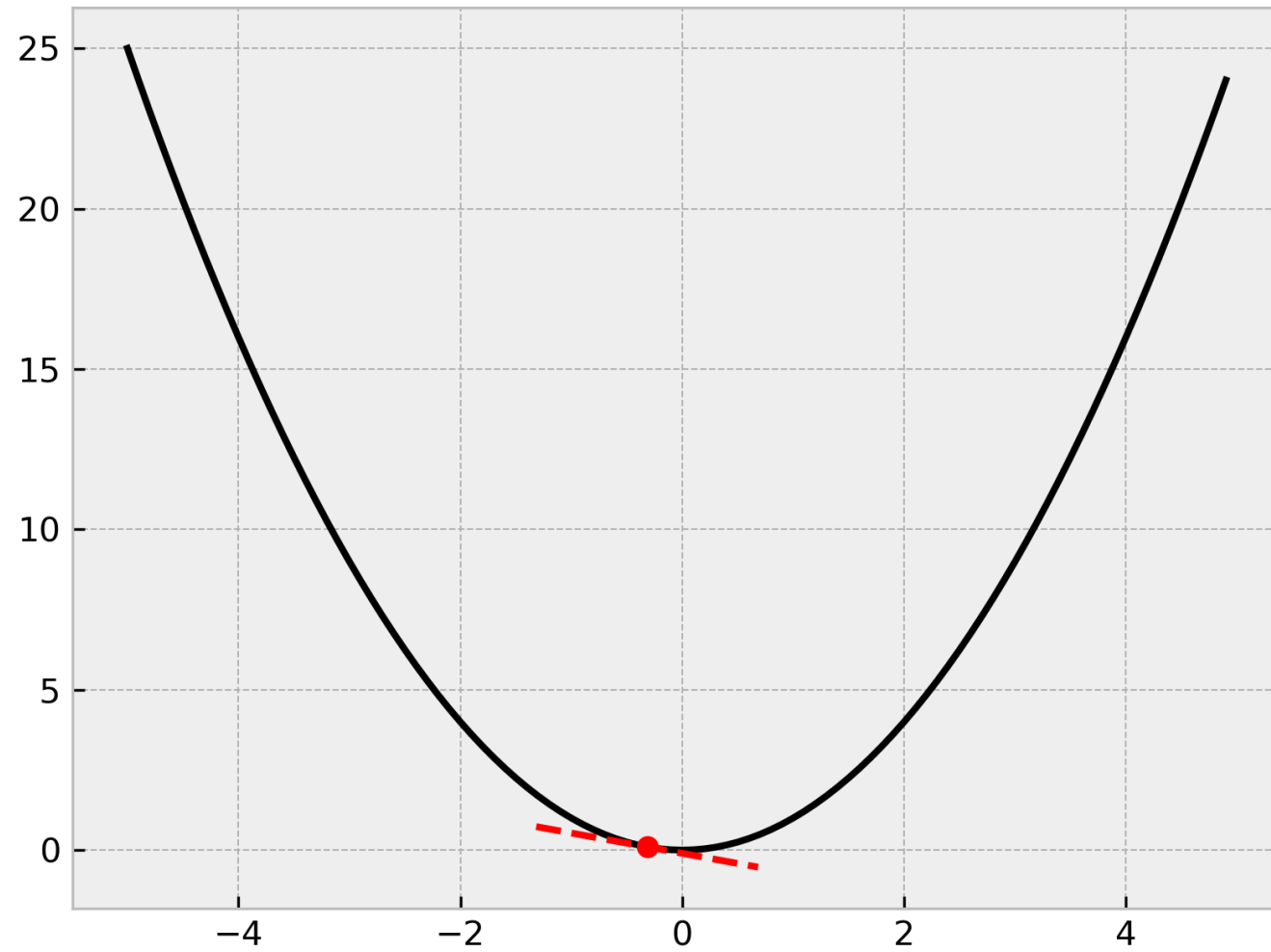
Gradient descent



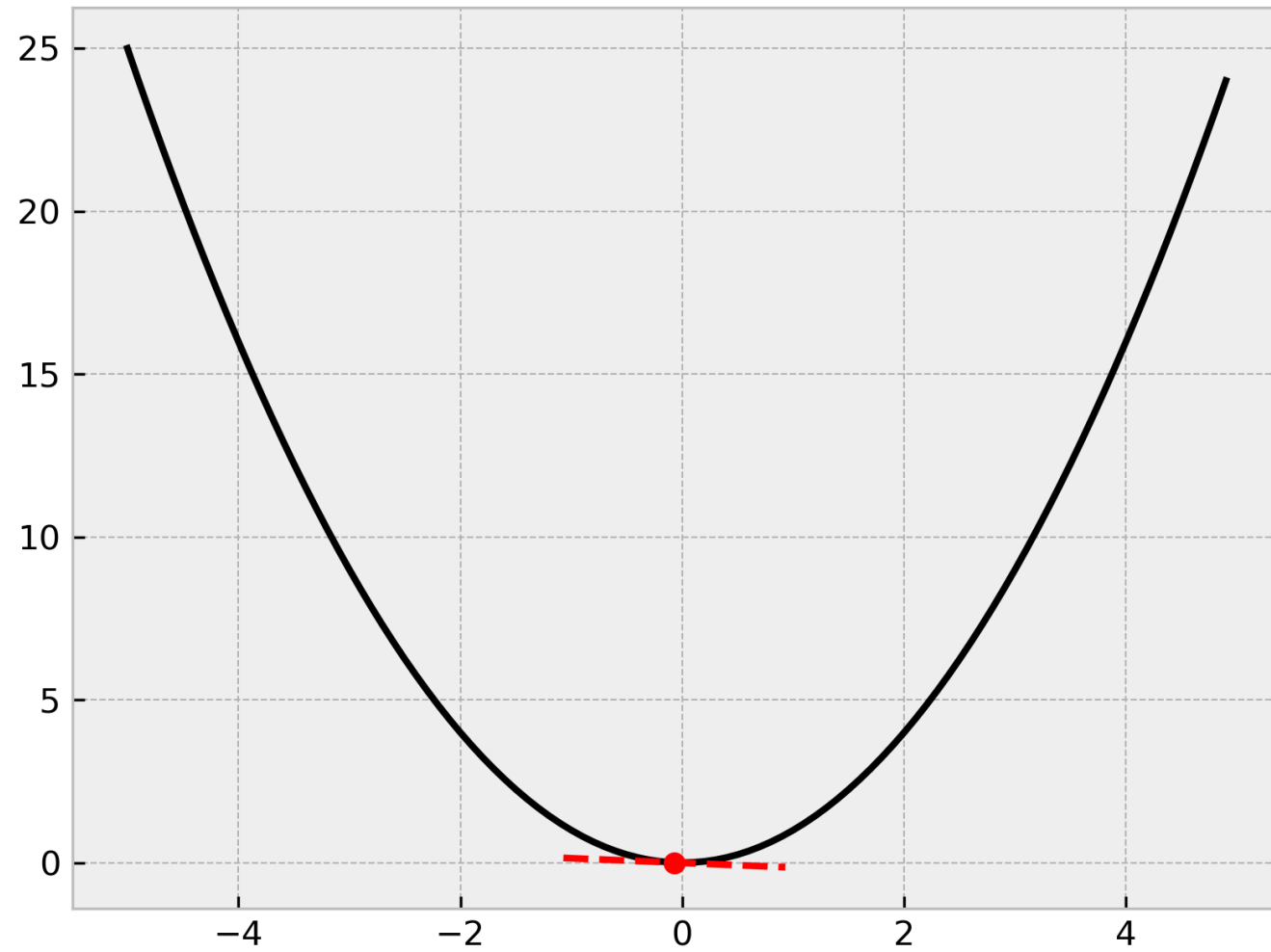
Gradient descent



Gradient descent



Gradient descent



1.7 Overfitting and Underfitting

- These are both common problems with ML models pertaining to the model and unseen (test) data

Underfitting

- Model is too simple to capture the underlying pattern in the data
- High bias, low variance
- Poor performance on both training and test data

Overfitting

- Model is too complex, capturing noise in the training data
- Low bias, high variance
- Excellent performance on training data, poor performance on test data

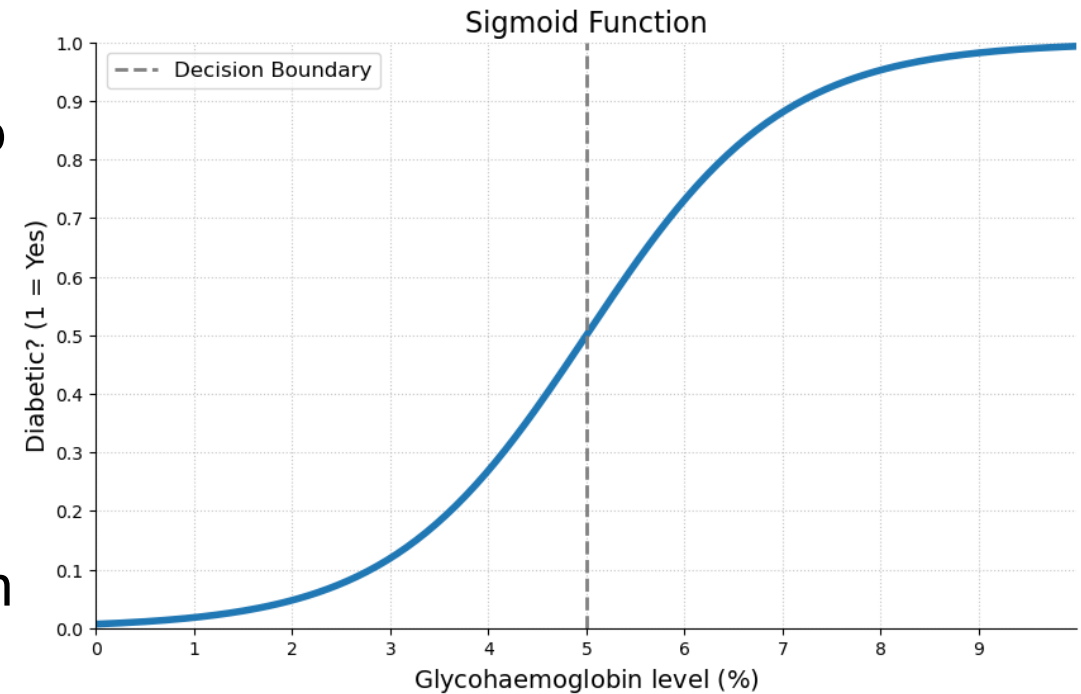
1.9 Model Selection

- Now we have seen how ML models learn – in simplistic terms – we should now consider which ML model type best serves our purpose
- Considerations:
 - Interpretability
 - Dataset size
 - Data characteristics
 - Problem type
 - Performance metrics

1.10 Common model types

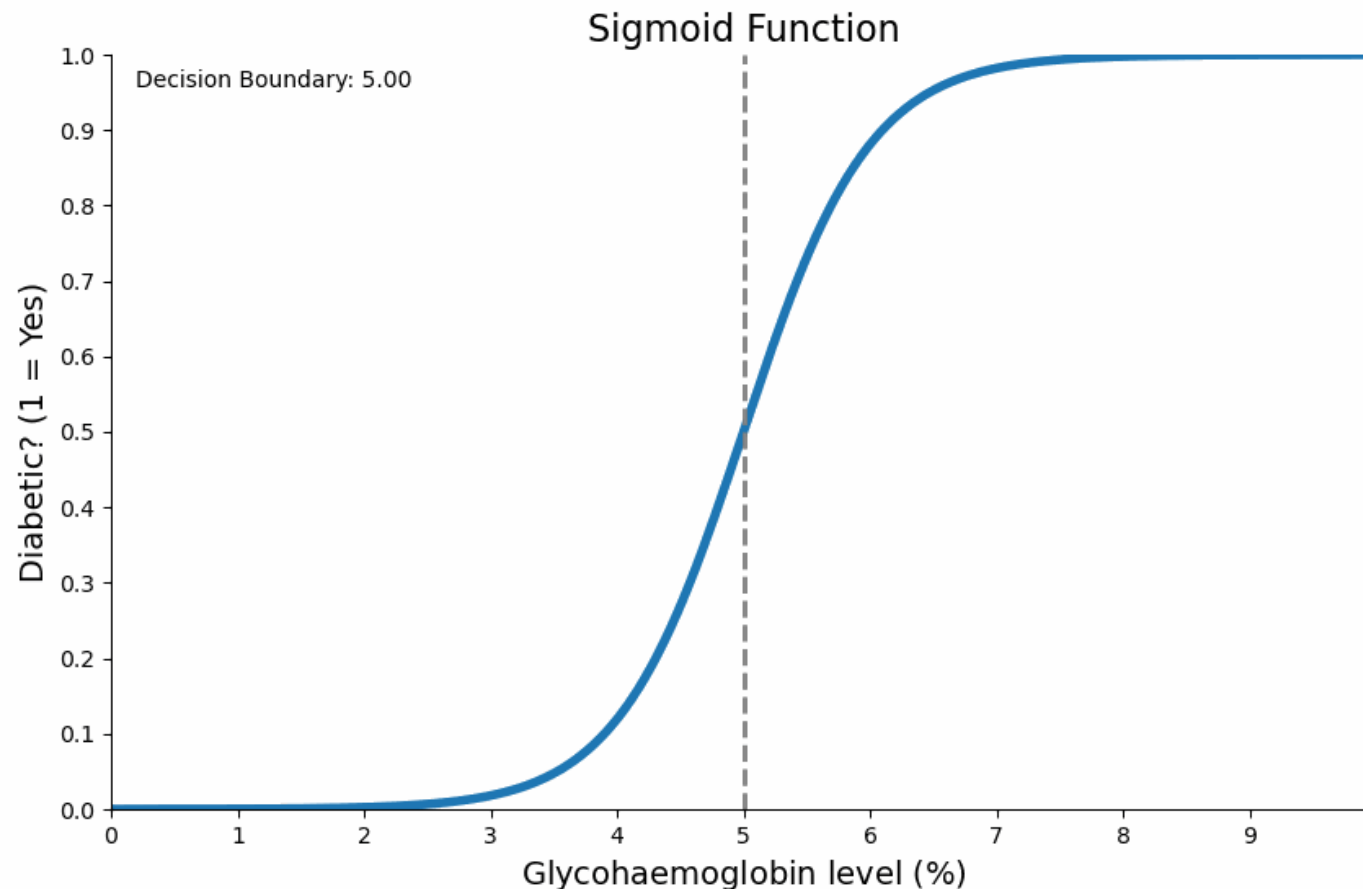
Logistic Regression

- Transforms linear combinations of features into probabilities
- Best uses are:
 - When interpretability is crucial
 - For linearly separable problems
 - As a baseline model for binary classification



1.10 Common model types

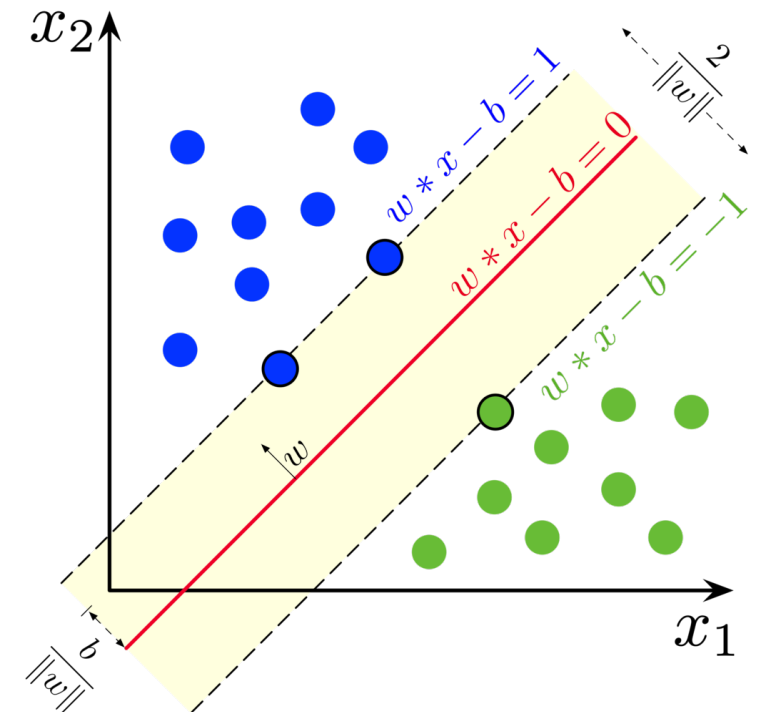
Logistic Regression



1.10 Common model types

Support Vector Machines

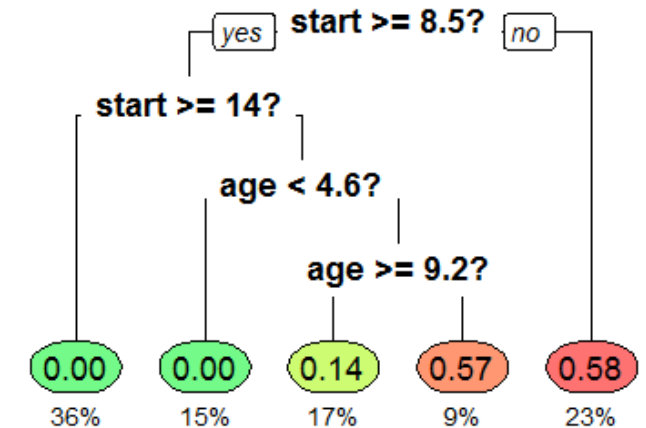
- Finds the hyperplane that maximizes the margin between classes
- Uses kernel tricks to handle non-linear decision boundaries
- Best uses are:
 - For small to medium-sized datasets
 - When dealing with high-dimensional spaces
 - When a clear margin of separation exists between classes



1.10 Common model types

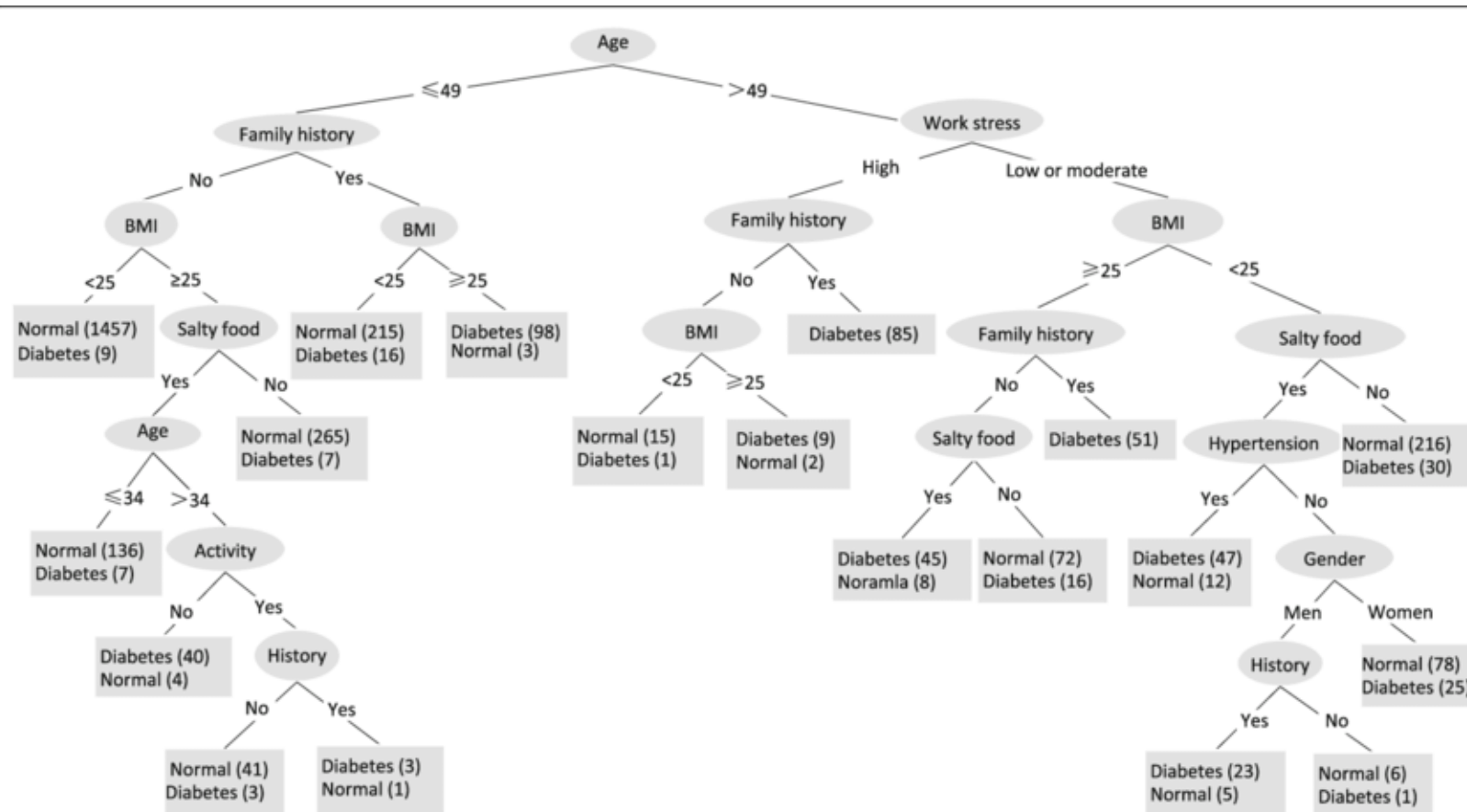
Decision Trees/Random Forests

- Recursively splits the data based on feature values to maximize information gain
- Grows the tree until a stopping criterion is met (e.g., max depth, min samples per leaf)
- Best uses are:
 - Interpretability and visualizing decision rules
 - Non-linear relationships without assuming a specific form



1.10 Common model types

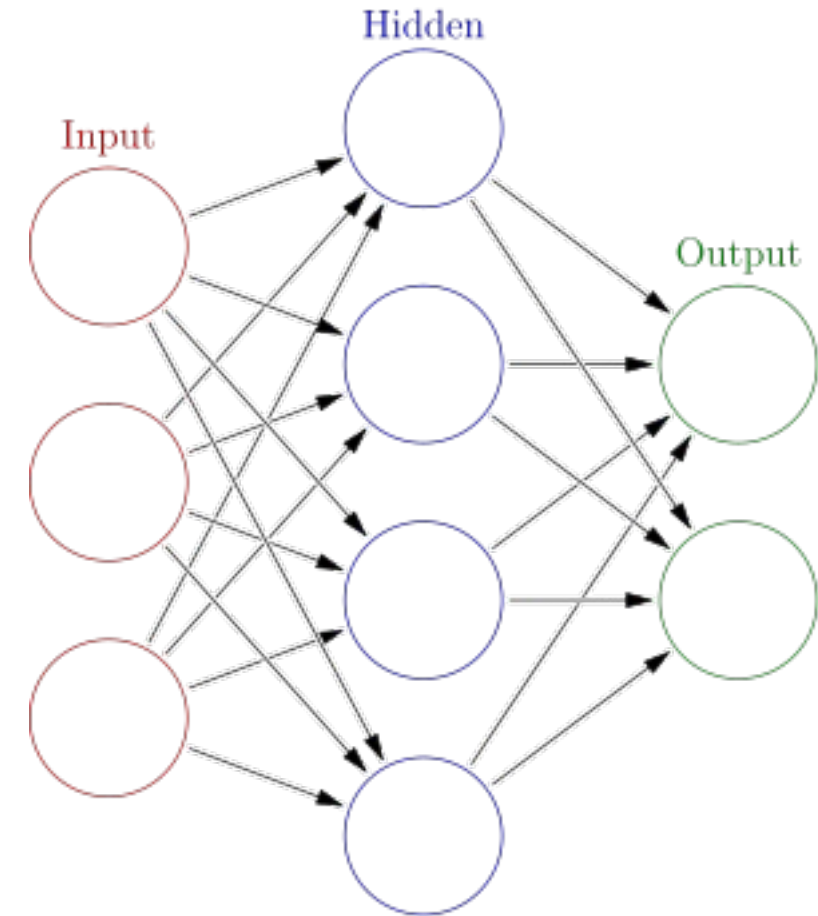
Decision Trees/Random Forests



1.10 Common model types

Neural Networks

- Learns hierarchical representations through multiple layers of non-linear transformations
- Uses backpropagation to compute gradients of the loss function with respect to weights
- Best uses are:
 - For large, complex datasets
 - When dealing with unstructured data (images, text, audio)
 - Interpretability is less critical



1.10 Common model types

Resources

- A cheat sheet of these models and cross-comparison is available as an appendix.

1.11 Data splitting

- Data splitting is a crucial step in the machine learning workflow
- Proper splitting helps assess model performance and generalization ability
- Typically, this is split into Training, Validation and Test data ‘folds’
 - Why also validation?
- An example splitting technique is *K*-Fold Cross-Validation...

1.11 Data splitting

K -Fold Cross-Validation

- A resampling technique used to evaluate model performance
- Split the data into K equal-sized folds
- For each fold i ($i = 1$ to K):
 - Use fold i as the validation set
 - Use the remaining $K-1$ folds as the training set
 - Train the model and evaluate on the validation set

$n = 8$

Model 1



1.11 Data splitting

Considerations

- Is this information known immediately or from a future visit? (Time-based)
- Make sure equal number of patient types across training and test sets (Stratification)
- Make sure data is representative of the target population not just the sample
 - For example, Multi-centre Studies

1. Introduction to model training

Summary

- **Types of Learning**
 - Supervised: Learn from labelled data (e.g., disease prediction)
 - Unsupervised: Discover patterns in unlabelled data (e.g., patient clustering)
- Loss functions and Gradient Descent
- Underfitting, Overfitting & Regularisation
- Common model types and their uses
- Preparing data for implementation

Questions?

Session 2 Overview

1. Introduction to model training
2. Running model training on clinical dataset (practical)
3. Introduction to model evaluation
4. Evaluating our models from section 1. (practical)
5. Summary
6. Q&A

2. Running model training on clinical dataset (practical)

- **Use the NHANES data from session 1**
- **Consider how an ML model is implemented in Python**
- **Look at how the concepts we have learnt about impact training**

2.1 The dataset

- We will be using Demographic, Examination Laboratory and, Questionnaire data
 - Demographic: Age, Gender, etc.
 - Examination: Weight (kg), Blood pressure, etc.
 - Lab: LDL-Cholesterol, Glycohemoglobin
 - Questionnaire: Doctor told you have diabetes (Y/N)
- Remember, for supervised machine learning we have two key data categories:
 - Input: The ‘features’ we use to predict with (Age, Weight, Cholesterol, etc.)
 - Output: The ‘targets’ we are trying to predict (Diabetes (Y/N), Glycohemoglobin)

Session 2 Overview

1. Introduction to model training
2. Running model training on clinical dataset (practical)
3. Introduction to model evaluation
4. Evaluating our models from section 1. (practical)
5. Summary
6. Q&A

3. Introduction to model evaluation

- So far, we have looked at how models learn patterns and relationships within the data
- We now have models trained on our data that can make predictions on new inputs!
 - In our current example, given a patient's demographic, exam and lab data, we can make a prediction whether this patient has diabetes or not
 - In the regression example, we can estimate this patients glycohaemoglobin levels
- We now want to know: *'How good are these predictions?'*

3. Introduction to model evaluation

Classification

- Think, do our predicted classes match our true classes
- E.g., if we predict patient A has diabetes – does patient A actually have diabetes?

Regression

- Think, is our prediction close to the true measurement
- E.g., if we predict a glycohaemoglobin level of 6.2% for patient A, how close is this to patient A's actual glycohaemoglobin level?

3.1 Model evaluation: Classification

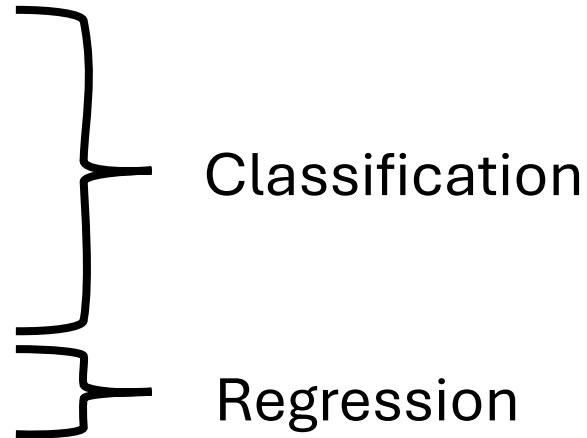
- Let's start with the classification example: Diabetes (Y/N)?
- A useful way to breakdown the outputs is by:
 - Polarity: Positive-Negative or, Diabetes=Yes vs. Diabetes=No
 - Validity: True-False or, Prediction=Correct vs. Prediction=incorrect

	True	False
Positive	Diabetes=Yes; Correct	Diabetes=Yes; Incorrect
Negative	Diabetes=No; Correct	Diabetes=No; Incorrect

3.2 Model evaluation: Metrics

- From our Punnett square, we can generate pretty much every evaluation metric:

- Accuracy
- Precision
- Sensitivity (Recall)
- Specificity
- Mean Square Error



- Which metric you consider depends highly on your dataset and goals
- (cheat sheet in appendices)

3.2 Model evaluation: Metrics

- Let's go through them and assume that each one has metric=90%
- We will consider how this is calculated and,
- What this means in our interpretation of the model
- Finally, the relevant clinical consideration

3.3 Model evaluation: Accuracy

- Accuracy is one of the most intuitive metric and represents the overall ‘correctness’ of the model's predictions.
- It is the ratio of **correct predictions** to the **total number of predictions**
- For our diabetes classifier, an accuracy of 90% means that the classifier correctly predicts diabetes 90% of the time
- While accuracy is easy to understand, it may not be the best metric when dealing with imbalanced datasets or when the “cost” of false positives and false negatives differs significantly.

	True (Yes)	False (No)
Predicted: Positive	Diabetes=Yes; Correct; TP	Diabetes=Yes; Incorrect; FP
Predicted: Negative	Diabetes=No; Correct' TN	Diabetes=No; Incorrect; FN

3.4 Model evaluation: Precision

- Precision is the proportion of **true positive predictions** among **all positive predictions**.
- It answers the question: "Out of all the instances predicted as positive, how many are actually positive?"
- For our diabetes classifier, given a positive prediction we have 90% confidence that the prediction is correct
- Precision is important when the cost of false positives is high, such as in medical diagnosis, where a false positive may lead to unnecessary treatments or interventions.

	True (Yes)	False (No)
Predicted: Positive	Diabetes=Yes; Correct; TP	Diabetes=Yes; Incorrect; FP
Predicted: Negative	Diabetes=No; Correct; TN	Diabetes=No; Incorrect; FN

3.5 Model evaluation: Sensitivity

- Sensitivity is the proportion of **correctly predicted positives** out of **all positive instances**.
- It answers the question: "Out of all the actual positive instances, how many did the model correctly identify?"
- It tells us that if we consider *all the people who do **not** have diabetes*, the classifier will correctly identify 90% of them
- Sensitivity is crucial when the cost of false negatives is high, such as in cancer screening, where missing a positive case can have severe consequences.

	True (Yes)	False (No)
Predicted: Positive	Diabetes=Yes; Correct; TP	Diabetes=Yes; Incorrect; FP
Predicted: Negative	Diabetes=No; Correct' TN	Diabetes=No; Incorrect; FN

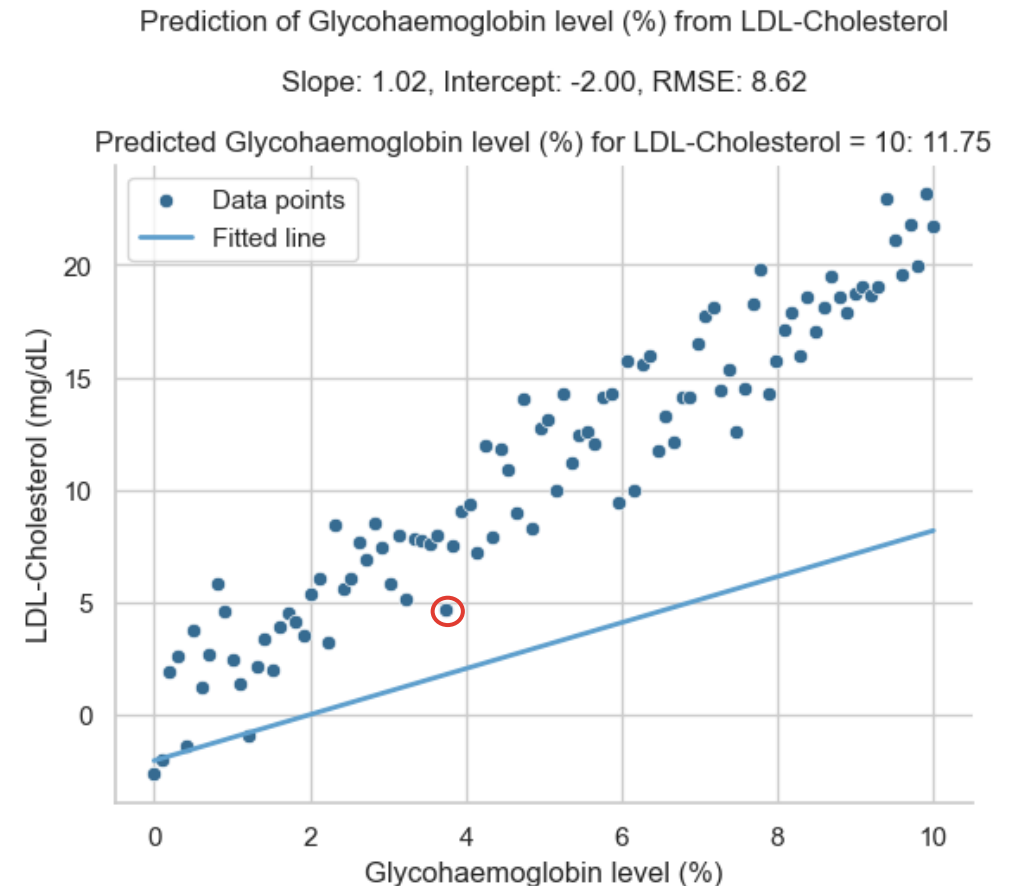
3.6 Model evaluation: Specificity

- Specificity is the proportion the proportion of **correctly predicted negatives** out of **all negative instances**.
- It answers the question: “Out of all the actual negative instances, how many did the model correctly identify?”
- High specificity is crucial when a false positive result can lead to unnecessary further testing, invasive procedures, or psychological distress for the patient.

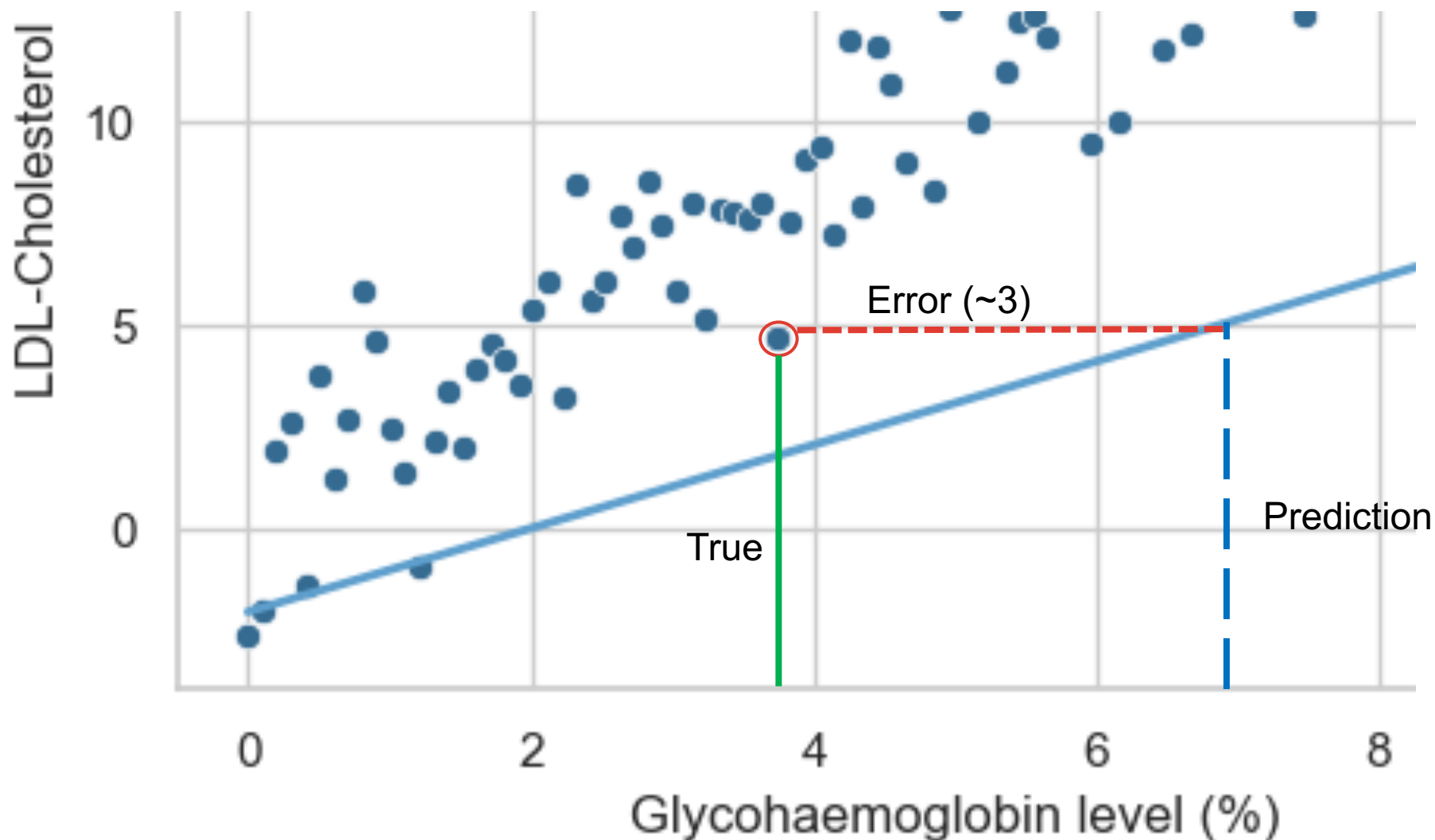
	True (Yes)	False (No)
Predicted: Positive	Diabetes=Yes; Correct; TP	Diabetes=Yes; Incorrect; FP
Predicted: Negative	Diabetes=No; Correct' TN	Diabetes=No; Incorrect; FN

3.7 Model evaluation: Regression

- Intuitively, evaluating regression models is measuring the distance between a predicted value and a true value of a test
- So, if patient A has LDL of 5 and a glycol- level of 4%, we want our predict value to be as close to that as possible.
- We do this by taking the predicted value minus the true value. E.g., ~7% predicted minus 4%



3.7 Model evaluation: Regression



3.7 Model evaluation: RMSE

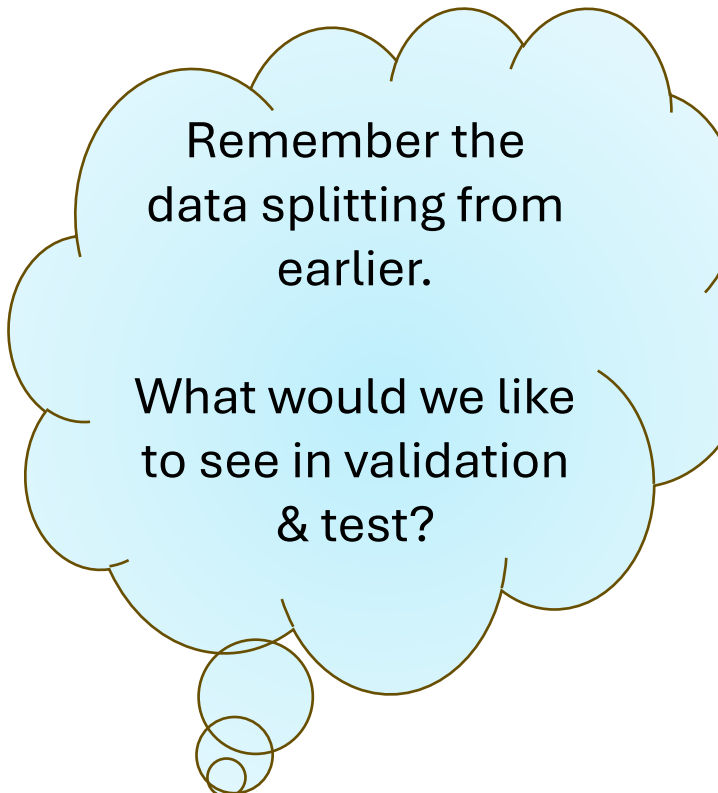
- Root Mean Squared Error (RMSE) measures the difference between the predicted and actual values.
 - Provides a balanced measure of a model's performance
- For the glycohemoglobin regression problem:
 - If the RMSE is 0.7%, it means that, on average, the model's predictions deviate from the actual glycohemoglobin levels by 0.7%
- While the model will try to reduce RMSE as much as it can, it is up to **you** to decide an acceptable error margin
 - In hearing health for example, hearing level is measured within 5 dB. Therefore, a $\text{RMSE} < 5 \text{ dB}$ is clinically acceptable

Think of a continuous measurement in your field.

What would an acceptable error level be?

3.8 Metric Considerations

- When selecting evaluation metrics for your machine learning model, consider the following:
 - Problem Domain and Objectives
 - Class Imbalance
 - Trade-offs between Metrics
 - Improving one metric (e.g., sensitivity) may come at the cost of another (e.g., specificity)
 - Domain-Specific Metrics
 - Validation and Testing



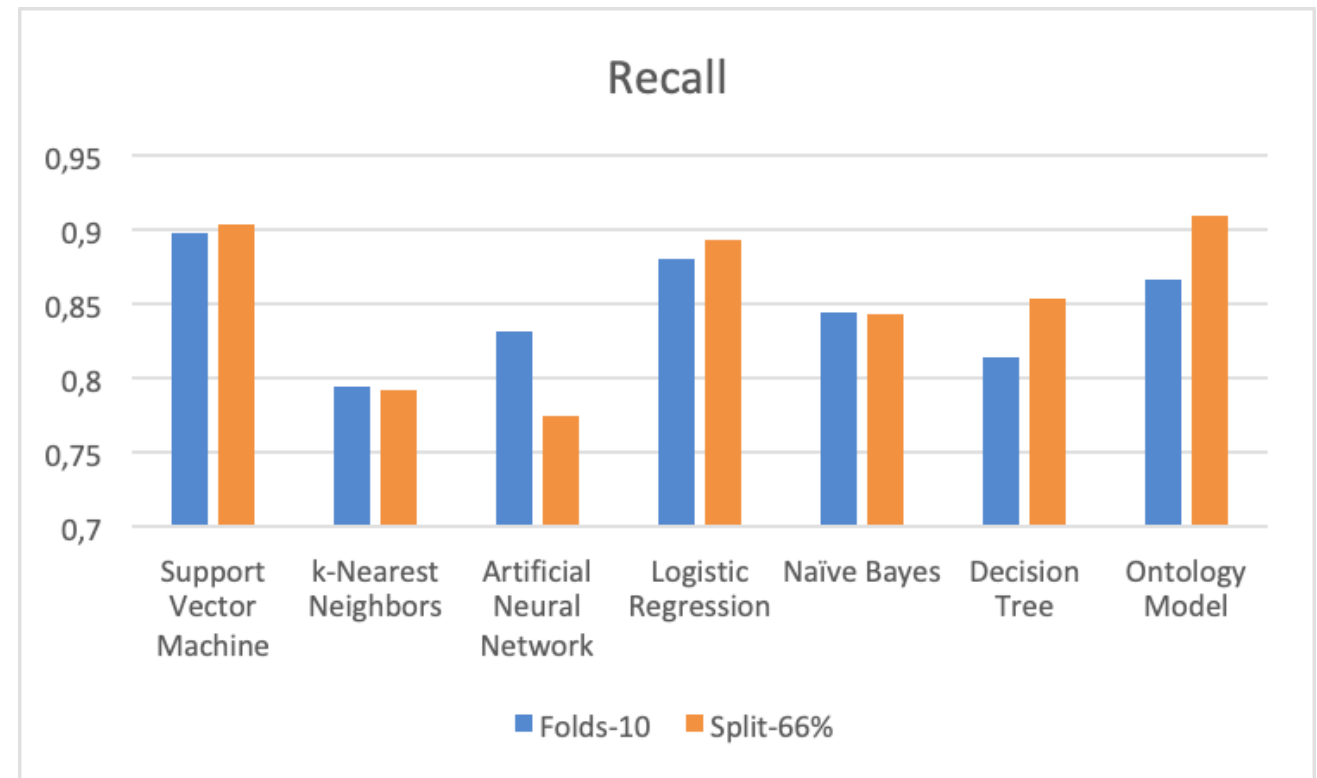
Remember the
data splitting from
earlier.

What would we like
to see in validation
& test?

3.9 Model evaluation: Benchmarking

- Benchmarking is crucial to assess the performance of clinical AI models against the current state-of-the-art.
- Look online for benchmark datasets, systematic reviews and previous works
 - From the existing literature, we can observe a sensitivity of $\sim 90\%$ for SVMs
- Consider the specific clinical context, patient population, and data characteristics when interpreting benchmark results

Sensitivity or Recall for models of diabetes prediction



From El Massari et al. (2022)

3.8 Ethical Considerations

- Consider the risks associated with incorrect predictions, such as misdiagnosis (FP+FN), delayed treatment (FN), or unnecessary interventions (FP)
- Ensure that the model's performance is consistent and fair across different patient subgroups (Obermeyer et al., 2019)
- Consider the explainability of the model's predictions
- Communicate the limitations and uncertainties associated with the model's predictions
- Governance & policy

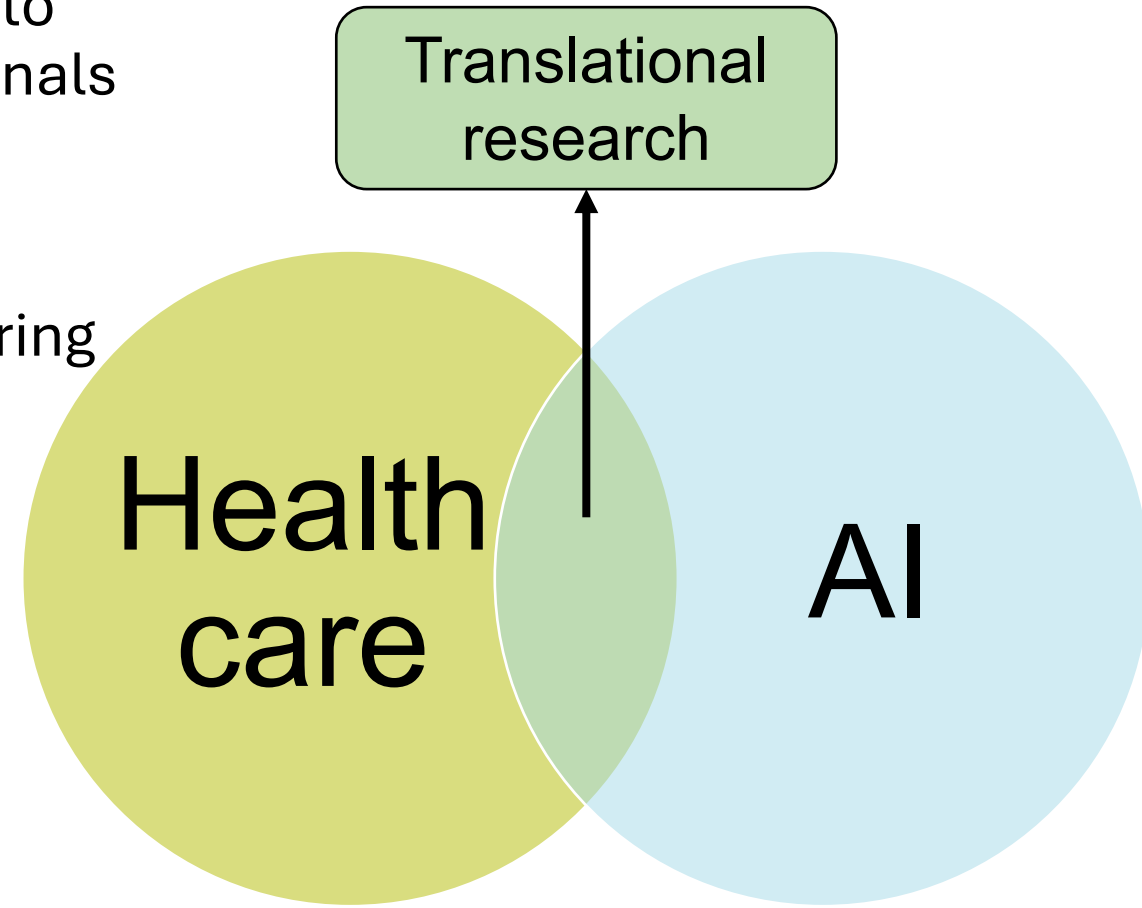
FP = False Positive
FN = False Negative

Session 2 Overview

1. Introduction to model training
2. Running model training on clinical dataset (practical)
3. Introduction to model evaluation
4. Evaluating our models from section 1. (practical)
5. Summary
6. Q&A

Importance of translational input

- Machine learning and AI have huge potentials to improve service for both health care professionals and patients
- It also has a large potential to go wrong
- Multidisciplinary experts (you) are key to ensuring actual improvements are delivered
- AI is highly susceptible to misalignment
I.e., your model



5. Session 2 Summary

- Model Training
 - You know the difference between classification and regression problems **and** when it is appropriate to apply either
 - You know how a machine learning model is trained on data at a conceptual level
 - You have learnt about advanced ML topics such as, regularisation
 - You have considered various standard ML models and where best to apply them
 - You have trained ML models and considered the effect on changes to the training process

Recall that
regularisation
helps prevent
overfitting

5. Session 2 Summary

- Model Evaluation
 - You know different methods for evaluating classification and regression models
 - You are aware of considerations when choose the appropriate metric given your goal
 - You are aware of ethical considerations when evaluating your model
 - Benchmarking is key to assess model performance
 - You have evaluated ML models and considered how to choose your final model

**Well done, Thank you and,
Questions?**

References and attributions

Dinh, A., Miertschin, S., Young, A. et al. A data-driven approach to predicting diabetes and cardiovascular disease with machine learning. BMC Med Inform Decis Mak 19, 211 (2019). <https://doi.org/10.1186/s12911-019-0918-5>

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

El Massari, H., Sabouri, Z., Mhammedi, S., & Gherabi, N. (2022). Diabetes prediction using machine learning algorithms and ontology. Journal of ICT Standardization, 10(2), 319-337.

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. Science, 366(6464), 447-453. <https://doi.org/10.1126/science.aax2342>

References and attributions

Figure of SVM decision boundary: Larhmam, CC BY-SA 4.0
<<https://creativecommons.org/licenses/by-sa/4.0>>, via Wikimedia Commons
https://commons.wikimedia.org/wiki/File:SVM_margin.png

Figure of Decision Tree: Stephen Milborrow, CC BY-SA 4.0
<<https://creativecommons.org/licenses/by-sa/4.0>>, via Wikimedia Commons
https://commons.wikimedia.org/wiki/File:Cart_tree_kyphosis.png

Figure of Neural Network: Glosser.ca, CC BY-SA 3.0
<<https://creativecommons.org/licenses/by-sa/3.0>>, via Wikimedia Commons
https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg

Figure of Cross-validation: MBanuelos22 - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=87684543>

Appendix A: Key Differences Between Example Models

Aspect	Logistic Regression	SVM	Decision Trees	Neural Networks
Model Complexity	Low	Medium	Medium	High
Interpretability	High	Medium	High	Low
Feature Scaling	Required	Required for some kernels	Not required	Required
Handling Non-linearity	Poor (without feature engineering)	Good (with non-linear kernels)	Good	Excellent
Training Speed	Fast	Medium to Slow	Fast	Slow
Prediction Speed	Fast	Fast (linear kernel), Medium (non-linear kernel)	Fast	Fast
Memory Usage	Low	Medium to High	Low	High
Handling High-dimensional Data	Poor to Medium	Good	Medium	Good
Sensitivity to Outliers	Medium	Low to Medium	Low	Medium
Handling Imbalanced Data	Poor (without adjustments)	Good	Medium	Medium (depends on design)
Hyperparameter Tuning Effort	Low	Medium	Medium	High
Scalability to Large Datasets	Good	Poor to Medium	Good	Excellent (with appropriate hardware)

Appendix A: Metrics

Aspect	Accuracy	Precision	Sensitivity (Recall)	Specificity	Mean Square Error (MSE)	Root Mean Square Error (RMSE)
Type of Problem	Classification	Classification	Classification	Classification	Regression	Regression
Interpretation	Higher is better. Proportion of correct predictions.	Higher is better. Proportion of true positive predictions among all positive predictions.	Higher is better. Proportion of actual positives correctly identified.	Higher is better. Proportion of actual negatives correctly identified.	Lower is better. Average squared difference between predicted and actual values.	Lower is better. Root of average squared difference between predicted and actual values.
Clinical Relevance	Good for balanced datasets. E.g., overall correct diagnosis rate.	Important when false positives are costly. E.g., avoiding unnecessary treatments.	Critical when missing positives is dangerous. E.g., cancer screening.	Important when falsely identifying negatives as positives is costly. E.g., avoiding unnecessary biopsies.	Useful for comparing models, but units are squared. E.g., predicting blood pressure, but squared differences.	More interpretable than MSE as it's in the same units as the target variable. E.g., predicting patient recovery time in days.