UCLA COMPUTER SCIENCE FIELD NOTES

CS M146

Introduction to Machine Learning

Statistical learning, regression, classification, kernels, clustering, EM, PCA, decisions, and reinforcement learning

How these notes are written. I wrote each chapter the way I wish the material had been explained the first time: the real idea, the point where I usually get stuck, the simplest mental model that still stays honest, and a worked decision instead of a polished answer appearing from nowhere. Every chapter closes with practice and a fully written solution. If a sentence sounds impressive but does not help me solve or explain something, it does not belong here.

Daniel Mastick

Curriculum map, working notes, practice, and solutions

Course map

Week	Core thread	What should be solid by the end
1	**Learning problems and generalization**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
2	**Linear regression and regularization**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
3	**Probabilistic classification**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
4	**Margin methods and kernels**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
5	**Trees, ensembles, and calibration**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
6	**Clustering, EM, and latent variables**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
7	**PCA and representation**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
8	**Reinforcement and deep learning overview**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
9	**Integration workshop**	Connect at least three chapters in one end-to-end problem and explain where their contracts meet.
10	**Synthesis and project review**	Audit assumptions and explain a complete solution from interface to failure handling.

Scope anchor: https://catalog.registrar.ucla.edu/course/2025/COMSCIM146. The sequence is aligned to the official catalog description and expanded into a practical ten-week study plan.

How I would use this packet

Treat each numbered section as one chapter. Read the formal explanation, pause at the student interpretation, and see whether the easy version still says the same thing. Rework the example without looking, then cover the solution in the chapter practice box. The cumulative set at the end is deliberately mixed: knowledge becomes durable when I have to choose the tool instead of being told which chapter I am in.

Contents

CHAPTER 1 Learning problems and generalization

The idea

Supervised learning chooses a function from data; risk is expected loss under an unknown distribution. Empirical risk, hypothesis class, regularization, and validation manage the gap between fitting samples and generalizing.

Rebuild the chapter from first principles

The claim I need to own is this: Supervised learning chooses a function from data; risk is expected loss under an unknown distribution. Empirical risk, hypothesis class, regularization, and validation manage the gap between fitting samples and generalizing. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of learning problems and generalization. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach learning problems and generalization on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from the course foundations to linear regression and regularization. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Supervised learning chooses a function from data; risk is expected loss under an unknown distribution.
- **Mechanism.** Empirical risk, hypothesis class, regularization, and validation manage the gap between fitting samples and generalizing.
- **Boundary.** The useful test is whether I can apply learning problems and generalization to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when learning problems and generalization is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I define input, target, loss, and deployment distribution before choosing a model.

The easy version

Supervised learning chooses a function from data; risk is expected loss under an unknown distribution.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why is training error optimistic? The conclusion is The model was selected to reduce that same sample's error. Estimate generalization on held-out data or nested cross-validation when model selection is involved.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why is training error optimistic?
Solution. The model was selected to reduce that same sample's error. Estimate generalization on held-out data or nested cross-validation when model selection is involved.

Practice 2. Give a short oral explanation of learning problems and generalization that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct learning problems and generalization from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a learning problems and generalization problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 2 Linear regression and regularization

The idea

Least squares projects targets onto the feature span. Gradient descent, normal equations, and numerical decompositions solve the fit. Ridge shrinks coefficients; lasso can select sparse features; both encode bias-variance tradeoffs.

Rebuild the chapter from first principles

The claim I need to own is this: Least squares projects targets onto the feature span. Gradient descent, normal equations, and numerical decompositions solve the fit. Ridge shrinks coefficients; lasso can select sparse features; both encode bias-variance tradeoffs. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of linear regression and regularization. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach linear regression and regularization on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from learning problems and generalization to probabilistic classification. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Least squares projects targets onto the feature span.
- **Mechanism.** Gradient descent, normal equations, and numerical decompositions solve the fit.
- **Boundary.** Ridge shrinks coefficients; lasso can select sparse features; both encode bias-variance tradeoffs.
- **Decision test.** I should be able to say when linear regression and regularization is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I read residual plots before celebrating one summary score.

The easy version

Least squares projects targets onto the feature span.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What does ridge add to the objective? The conclusion is It adds lambda times squared coefficient norm, discouraging large weights and stabilizing correlated or weakly identified directions; features usually need comparable scaling.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What does ridge add to the objective?
Solution. It adds lambda times squared coefficient norm, discouraging large weights and stabilizing correlated or weakly identified directions; features usually need comparable scaling.

Practice 2. Give a short oral explanation of linear regression and regularization that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a

four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct linear regression and regularization from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a linear regression and regularization problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 3 Probabilistic classification

The idea

Logistic regression models log odds linearly and optimizes cross-entropy. Bayes decision rules choose actions minimizing expected loss; thresholds should reflect costs and class priors, not default blindly to one half.

Rebuild the chapter from first principles

The claim I need to own is this: Logistic regression models log odds linearly and optimizes cross-entropy. Bayes decision rules choose actions minimizing expected loss; thresholds should reflect costs and class priors, not default blindly to one half. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of probabilistic classification. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative

derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach probabilistic classification on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from linear regression and regularization to margin methods and kernels. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Logistic regression models log odds linearly and optimizes cross-entropy.
- **Mechanism.** Bayes decision rules choose actions minimizing expected loss; thresholds should reflect costs and class priors, not default blindly to one half.
- **Boundary.** The useful test is whether I can apply probabilistic classification to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when probabilistic classification is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

A probability score and a final decision are different layers; I tune the threshold for the real cost.

The easy version

Logistic regression models log odds linearly and optimizes cross-entropy.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Derive the equal-cost Bayes rule. The conclusion is Choose the class with larger posterior probability. With unequal costs, compare posterior expected costs, which shifts the threshold.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Derive the equal-cost Bayes rule.
Solution. Choose the class with larger posterior probability. With unequal costs, compare posterior expected costs, which shifts the threshold.

Practice 2. Give a short oral explanation of probabilistic classification that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct probabilistic classification from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a probabilistic classification problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 4 Margin methods and kernels

The idea

Support vector machines maximize a regularized margin while hinge loss penalizes violations. Kernels compute inner products in implicit feature spaces, allowing nonlinear boundaries without explicitly constructing every feature.

Rebuild the chapter from first principles

The claim I need to own is this: Support vector machines maximize a regularized margin while hinge loss penalizes violations. Kernels compute inner products in implicit feature spaces, allowing nonlinear boundaries without explicitly constructing every feature. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of margin methods and kernels. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach margin methods and kernels on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from probabilistic classification to trees, ensembles, and calibration. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Support vector machines maximize a regularized margin while hinge loss penalizes violations.
- **Mechanism.** Kernels compute inner products in implicit feature spaces, allowing nonlinear boundaries without explicitly constructing every feature.
- **Boundary.** The useful test is whether I can apply margin methods and kernels to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when margin methods and kernels is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: the kernel trick is not free magic; kernel choice still defines what similarity the model believes.

The easy version

Support vector machines maximize a regularized margin while hinge loss penalizes violations.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why do only support vectors affect the boundary? The conclusion is The dual solution gives nonzero coefficients to points on or inside the margin; points safely beyond it have zero multiplier and do not change the optimum.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why do only support vectors affect the boundary?
Solution. The dual solution gives nonzero coefficients to points on or inside the margin; points safely beyond it have zero multiplier and do not change the optimum.

Practice 2. Give a short oral explanation of margin methods and kernels that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct margin methods and kernels from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a margin methods and kernels problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 5 Trees, ensembles, and calibration

The idea

Decision trees partition feature space through greedy splits and can overfit deeply. Bagging and random forests reduce variance; boosting builds a weighted sequence of weak learners. Probability calibration must be checked separately from ranking.

Rebuild the chapter from first principles

The claim I need to own is this: Decision trees partition feature space through greedy splits and can overfit deeply. Bagging and random forests reduce variance; boosting builds a weighted sequence of weak learners. Probability calibration must be checked separately from ranking. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of trees, ensembles, and calibration. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach trees, ensembles, and calibration on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from margin methods and kernels to clustering, em, and latent variables. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Decision trees partition feature space through greedy splits and can overfit deeply.
- **Mechanism.** Bagging and random forests reduce variance; boosting builds a weighted sequence of weak learners.
- **Boundary.** Probability calibration must be checked separately from ranking.
- **Decision test.** I should be able to say when trees, ensembles, and calibration is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I use a tree to expose interactions, then verify whether its apparent rules survive resampling.

The easy version

Decision trees partition feature space through greedy splits and can overfit deeply.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Contrast bagging and boosting. The conclusion is Bagging fits learners independently on resamples and averages to reduce variance. Boosting fits sequentially to emphasize current errors, often reducing bias but increasing sensitivity.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Contrast bagging and boosting.
Solution. Bagging fits learners independently on resamples and averages to reduce variance. Boosting fits sequentially to emphasize current errors, often reducing bias but increasing sensitivity.

Practice 2. Give a short oral explanation of trees, ensembles, and calibration that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct trees, ensembles, and calibration from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a trees, ensembles, and calibration problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 6 Clustering, EM, and latent variables

The idea

K-means alternates assignments and centroid updates to decrease within-cluster squared distance. Gaussian mixtures model soft membership; expectation-maximization alternates posterior responsibilities with parameter updates and converges to a local optimum.

Rebuild the chapter from first principles

The claim I need to own is this: K-means alternates assignments and centroid updates to decrease within-cluster squared distance. Gaussian mixtures model soft membership; expectation-maximization alternates posterior responsibilities with parameter updates and converges to a local optimum. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of clustering, em, and latent variables. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach clustering, em, and latent variables on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from trees, ensembles, and calibration to pca and representation. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** K-means alternates assignments and centroid updates to decrease within-cluster squared distance.
- **Mechanism.** Gaussian mixtures model soft membership; expectation-maximization alternates posterior responsibilities with parameter updates and converges to a local optimum.
- **Boundary.** The useful test is whether I can apply clustering, em, and latent variables to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when clustering, em, and latent variables is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: cluster labels are a model-created summary, not discovered natural truth.

The easy version

K-means alternates assignments and centroid updates to decrease within-cluster squared distance.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why can EM need multiple initializations? The conclusion is Its likelihood improvement is local, and different starting parameters can reach different stationary points or degenerate solutions. Compare runs and regularize covariance estimates.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why can EM need multiple initializations?
Solution. Its likelihood improvement is local, and different starting parameters can reach different stationary points or degenerate solutions. Compare runs and regularize covariance estimates.

Practice 2. Give a short oral explanation of clustering, em, and latent variables that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct clustering, em, and latent variables from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a clustering, em, and latent variables problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 7 PCA and representation

The idea

PCA centers data and finds orthogonal maximum-variance directions through eigenvectors or singular vectors. Projection compresses linear structure, but variance is not the same as usefulness, causality, or fairness.

Rebuild the chapter from first principles

The claim I need to own is this: PCA centers data and finds orthogonal maximum-variance directions through eigenvectors or singular vectors. Projection compresses linear structure, but variance is not the same as usefulness, causality, or fairness. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of pca and representation. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach pca and representation on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from clustering, em, and latent variables to reinforcement and deep learning overview. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** PCA centers data and finds orthogonal maximum-variance directions through eigenvectors or singular vectors.
- **Mechanism.** Projection compresses linear structure, but variance is not the same as usefulness, causality, or fairness.
- **Boundary.** The useful test is whether I can apply pca and representation to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when pca and representation is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I inspect loadings and reconstruction error; a pretty two-dimensional plot is not an explanation by itself.

The easy version

PCA centers data and finds orthogonal maximum-variance directions through eigenvectors or singular vectors.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Relate PCA to SVD. The conclusion is

For centered matrix X=U Sigma V^T, columns of V are principal directions and squared singular values determine explained variance.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Relate PCA to SVD.
Solution. For centered matrix X=U Sigma V^T, columns of V are principal directions and squared singular values determine explained variance.

Practice 2. Give a short oral explanation of pca and representation that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct pca and representation from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a pca and representation problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 8 Reinforcement and deep learning overview

The idea

Reinforcement learning optimizes cumulative reward under sequential state transitions. Values satisfy Bellman relationships; exploration gathers information. Deep networks learn representations but add optimization, data, robustness, and interpretability concerns.

Rebuild the chapter from first principles

The claim I need to own is this: Reinforcement learning optimizes cumulative reward under sequential state transitions. Values satisfy Bellman relationships; exploration gathers information. Deep networks learn representations but add optimization, data, robustness, and interpretability concerns. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of reinforcement and deep learning overview. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach reinforcement and deep learning overview on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from pca and representation to the cumulative course model. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Reinforcement learning optimizes cumulative reward under sequential state transitions.
- **Mechanism.** Values satisfy Bellman relationships; exploration gathers information.
- **Boundary.** Deep networks learn representations but add optimization, data, robustness, and interpretability concerns.
- **Decision test.** I should be able to say when reinforcement and deep learning overview is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I keep reward design separate from the behavior I merely hope it creates.

The easy version

Reinforcement learning optimizes cumulative reward under sequential state transitions.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Write the Bellman optimality equation in words. The conclusion is Optimal state value equals the best action's expected immediate reward plus discounted optimal value of the next state, under the transition model.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Write the Bellman optimality equation in words.
Solution. Optimal state value equals the best action's expected immediate reward plus discounted optimal value of the next state, under the transition model.

Practice 2. Give a short oral explanation of reinforcement and deep learning overview that includes its model, one assumption, one failure mode, and one comparison with a neighboring

method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct reinforcement and deep learning overview from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a reinforcement and deep learning overview problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

Final study check

I should be able to close this packet and reconstruct the core models, not merely recognize their names. My final check is to explain one complete problem aloud: what the inputs mean, which invariant or contract controls the work, why the method is legal, what it costs, how it can fail, and what evidence would convince another engineer.