

UCLA-ALIGNED COURSE FIELD NOTES

STATS 101C

Introduction to Statistical Models and Data Mining

Supervised learning, unsupervised structure, validation, regularization, diagnostics, and responsible model use

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	Prediction problems and honest data partitions	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
2	Linear prediction and regularization	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
3	Classification, probability, and decision thresholds	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
4	Trees, ensembles, and nonlinear structure	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
5	Nearest neighbors, margins, and feature geometry	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
6	Clustering and dimension reduction	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
7	Model assessment, resampling, and leakage	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
8	Interpretability, fairness, and deployment monitoring	Reconstruct the model, work an application, test its assumptions, and explain the result in ordinary language.
9	Integration studio	Combine several chapters in one case, preserve their assumptions, and reconcile the result across representations.
10	Cumulative review	Retrieve the full course map from memory and solve mixed problems without chapter labels.

Scope anchor: <https://catalog.registrar.ucla.edu/course/2025/STATS101C>. 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

1	Prediction problems and honest data partitions	5
1.1	Rebuild the chapter from first principles	5
1.2	Details I would refuse to skip	5
1.3	How this chapter connects	6
1.4	The full working model	6
2	Linear prediction and regularization	7
2.1	Rebuild the chapter from first principles	7
2.2	Details I would refuse to skip	8
2.3	How this chapter connects	8
2.4	The full working model	8
3	Classification, probability, and decision thresholds	10
3.1	Rebuild the chapter from first principles	10
3.2	Details I would refuse to skip	10
3.3	How this chapter connects	11
3.4	The full working model	11
4	Trees, ensembles, and nonlinear structure	12
4.1	Rebuild the chapter from first principles	12
4.2	Details I would refuse to skip	13
4.3	How this chapter connects	13
4.4	The full working model	14
5	Nearest neighbors, margins, and feature geometry	15
5.1	Rebuild the chapter from first principles	15
5.2	Details I would refuse to skip	15
5.3	How this chapter connects	16
5.4	The full working model	16
6	Clustering and dimension reduction	18
6.1	Rebuild the chapter from first principles	18
6.2	Details I would refuse to skip	18
6.3	How this chapter connects	19

6.4	The full working model	19
7	Model assessment, resampling, and leakage	20
7.1	Rebuild the chapter from first principles	20
7.2	Details I would refuse to skip	21
7.3	How this chapter connects	21
7.4	The full working model	21
8	Interpretability, fairness, and deployment monitoring	23
8.1	Rebuild the chapter from first principles	23
8.2	Details I would refuse to skip	23
8.3	How this chapter connects	24
8.4	The full working model	24

CHAPTER 1 Prediction problems and honest data partitions

The idea

Data mining starts with an outcome, observation unit, loss function, baseline, and deployment population. Training, validation, and test partitions separate fitting, tuning, and final evaluation; preprocessing belongs inside the training pipeline.

Rebuild the chapter from first principles

The claim I need to own is this: Data mining starts with an outcome, observation unit, loss function, baseline, and deployment population. Training, validation, and test partitions separate fitting, tuning, and final evaluation; preprocessing belongs inside the training pipeline. 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 prediction problems and honest data partitions. 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 prediction problems and honest data partitions 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 prediction 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.

The full working model

- **Core claim.** Data mining starts with an outcome, observation unit, loss function, baseline, and deployment population.
- **What moves.** Training, validation, and test partitions separate fitting, tuning, and final evaluation; preprocessing belongs inside the training pipeline.
- **What keeps the answer honest.** The model becomes useful only when I can apply prediction problems and honest data partitions to an unfamiliar case and defend the assumptions.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

I define what one future prediction represents and how a mistake costs us before comparing models.

The easy version

Data mining starts with an outcome, observation unit, loss function, baseline, and deployment population.

Worked example

Question. Why should the test set remain untouched during tuning?

Walkthrough. Repeated test feedback turns it into another validation set and overfits decisions to it. Use it once for an honest final estimate after the pipeline is fixed.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. Why should the test set remain untouched during tuning?

Solution. Repeated test feedback turns it into another validation set and overfits decisions to it. Use it once for an honest final estimate after the pipeline is fixed.

Practice 2. Explain prediction problems and honest data partitions to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct prediction problems and honest data partitions 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 prediction problems and honest data partitions 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 prediction and regularization

The idea

Linear models provide an interpretable predictive baseline. Ridge shrinks correlated coefficient directions, lasso can set coefficients to zero, and elastic net blends both; tuning controls the bias-variance tradeoff.

Rebuild the chapter from first principles

The claim I need to own is this: Linear models provide an interpretable predictive baseline. Ridge shrinks correlated coefficient directions, lasso can set coefficients to zero, and elastic net blends both; tuning controls the bias-variance tradeoff. 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 prediction 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 prediction 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 prediction problems and honest data partitions to classification, probability, and decision thresholds. 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.

The full working model

- **Core claim.** Linear models provide an interpretable predictive baseline.
- **What moves.** Ridge shrinks correlated coefficient directions, lasso can set coefficients to zero, and elastic net blends both; tuning controls the bias-variance tradeoff.

- **What keeps the answer honest.** The model becomes useful only when I can apply linear prediction and regularization to an unfamiliar case and defend the assumptions.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

My shorthand for this chapter is: regularization is a bet that a little bias will buy a larger reduction in unstable variance.

The easy version

Linear models provide an interpretable predictive baseline.

Worked example

Question. Why standardize predictors before ridge or lasso?

Walkthrough. The penalty acts on coefficient magnitude. Without standardization, measurement units determine how strongly each predictor is penalized.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. Why standardize predictors before ridge or lasso?

Solution. The penalty acts on coefficient magnitude. Without standardization, measurement units determine how strongly each predictor is penalized.

Practice 2. Explain linear prediction and regularization to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct linear prediction 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 prediction 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 Classification, probability, and decision thresholds

The idea

Logistic models, discriminant methods, and other classifiers estimate scores or probabilities. Confusion matrices, sensitivity, specificity, ROC, precision-recall, calibration, and cost-sensitive thresholds answer different decision questions.

Rebuild the chapter from first principles

The claim I need to own is this: Logistic models, discriminant methods, and other classifiers estimate scores or probabilities. Confusion matrices, sensitivity, specificity, ROC, precision-recall, calibration, and cost-sensitive thresholds answer different decision questions. 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 classification, probability, and decision thresholds. 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 classification, probability, and decision thresholds 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 prediction and regularization to trees, ensembles, and nonlinear structure. 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.

The full working model

- **Core claim.** Logistic models, discriminant methods, and other classifiers estimate scores or probabilities.
- **What moves.** Confusion matrices, sensitivity, specificity, ROC, precision-recall, calibration, and cost-sensitive thresholds answer different decision questions.
- **What keeps the answer honest.** The model becomes useful only when I can apply classification, probability, and decision thresholds to an unfamiliar case and defend the assumptions.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

A probability model and a decision rule are separate objects; I can change the threshold without refitting the probability model.

The easy version

Logistic models, discriminant methods, and other classifiers estimate scores or probabilities.

Worked example

Question. Why can accuracy be misleading when positives are rare?

Walkthrough. Predicting the majority class can achieve high accuracy while finding no positives. Use class-specific errors, precision-recall, calibration, and decision costs.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. Why can accuracy be misleading when positives are rare?

Solution. Predicting the majority class can achieve high accuracy while finding no positives. Use class-specific errors, precision-recall, calibration, and decision costs.

Practice 2. Explain classification, probability, and decision thresholds to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct classification, probability, and decision thresholds 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 classification, probability, and decision thresholds 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 Trees, ensembles, and nonlinear structure**The idea**

Decision trees partition feature space through recursive splits. Bagging and random forests reduce variance by averaging decorrelated trees; boosting builds a sequence focused on residual errors. Tuning depth and learning controls complexity.

Rebuild the chapter from first principles

The claim I need to own is this: Decision trees partition feature space through recursive splits. Bagging and random forests reduce variance by averaging decorrelated trees; boosting builds a sequence focused on residual errors. Tuning depth and learning controls complexity. 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 nonlinear structure. 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 nonlinear structure 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 classification, probability, and decision thresholds to nearest neighbors, margins, and feature geometry. 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.

The full working model

- **Core claim.** Decision trees partition feature space through recursive splits.
- **What moves.** Bagging and random forests reduce variance by averaging decorrelated trees; boosting builds a sequence focused on residual errors.
- **What keeps the answer honest.** Tuning depth and learning controls complexity.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

My shorthand for this chapter is: a tree is a chain of local questions; an ensemble trades some readability for stability and predictive strength.

The easy version

Decision trees partition feature space through recursive splits.

Worked example

Question. How does a random forest decorrelate its trees?

Walkthrough. Each tree uses a bootstrap sample and considers a random subset of predictors at each split, reducing the chance that the same dominant feature drives every tree.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. How does a random forest decorrelate its trees?

Solution. Each tree uses a bootstrap sample and considers a random subset of predictors at each split, reducing the chance that the same dominant feature drives every tree.

Practice 2. Explain trees, ensembles, and nonlinear structure to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct trees, ensembles, and nonlinear structure 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 nonlinear structure 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 Nearest neighbors, margins, and feature geometry

The idea

K-nearest neighbors predicts from local examples, while support-vector machines use a maximum-margin boundary and kernels for nonlinear separation. Scaling, distance choice, dimensionality, and tuning determine what neighborhood or margin means.

Rebuild the chapter from first principles

The claim I need to own is this: K-nearest neighbors predicts from local examples, while support-vector machines use a maximum-margin boundary and kernels for nonlinear separation. Scaling, distance choice, dimensionality, and tuning determine what neighborhood or margin means. 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 nearest neighbors, margins, and feature geometry. 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 nearest neighbors, margins, and feature geometry 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 nonlinear structure to clustering and dimension reduction. 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.

The full working model

- **Core claim.** K-nearest neighbors predicts from local examples, while support-vector machines use a maximum-margin boundary and kernels for nonlinear separation.
- **What moves.** Scaling, distance choice, dimensionality, and tuning determine what neighborhood or margin means.
- **What keeps the answer honest.** The model becomes useful only when I can apply nearest neighbors, margins, and feature geometry to an unfamiliar case and defend the assumptions.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

If distance drives the model, units and irrelevant features are part of the model whether I acknowledge them or not.

The easy version

K-nearest neighbors predicts from local examples, while support-vector machines use a maximum-margin boundary and kernels for nonlinear separation.

Worked example

Question. What is the bias-variance effect of increasing k in k -nearest neighbors?

Walkthrough. Larger k smooths over more neighbors, usually increasing bias and reducing variance. Cross-validation chooses a useful balance.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. What is the bias-variance effect of increasing k in k -nearest neighbors?

Solution. Larger k smooths over more neighbors, usually increasing bias and reducing variance. Cross-validation chooses a useful balance.

Practice 2. Explain nearest neighbors, margins, and feature geometry to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct nearest neighbors, margins, and feature geometry 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 nearest neighbors, margins, and feature geometry 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 and dimension reduction

The idea

K-means, hierarchical clustering, and density methods search for structure without labeled outcomes. Principal components find orthogonal high-variance directions. Scaling, initialization, stability, and substantive validation matter because clusters are not automatically real populations.

Rebuild the chapter from first principles

The claim I need to own is this: K-means, hierarchical clustering, and density methods search for structure without labeled outcomes. Principal components find orthogonal high-variance directions. Scaling, initialization, stability, and substantive validation matter because clusters are not automatically real populations. 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 and dimension reduction. 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 and dimension reduction 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 nearest neighbors, margins, and feature geometry to model assessment, resampling, and leakage. 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.

The full working model

- **Core claim.** K-means, hierarchical clustering, and density methods search for structure without labeled outcomes.
- **What moves.** Principal components find orthogonal high-variance directions.
- **What keeps the answer honest.** Scaling, initialization, stability, and substantive validation matter because clusters are not automatically real populations.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

My shorthand for this chapter is: unsupervised output is a proposed map of the data, not a discovered law of nature.

The easy version

K-means, hierarchical clustering, and density methods search for structure without labeled outcomes.

Worked example

Question. Why can two variables measured on large scales dominate k-means?

Walkthrough. Euclidean distance squares raw coordinate differences. Standardize or use a substantively justified metric so units do not define the clusters by accident.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. Why can two variables measured on large scales dominate k-means?

Solution. Euclidean distance squares raw coordinate differences. Standardize or use a substantively justified metric so units do not define the clusters by accident.

Practice 2. Explain clustering and dimension reduction to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct clustering and dimension reduction 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 and dimension reduction 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 Model assessment, resampling, and leakage

The idea

Cross-validation estimates generalization when folds match deployment. Hyperparameter search, nested validation, learning curves, subgroup metrics, uncertainty, and temporal splits expose overfitting and drift. Leakage occurs whenever future or outcome-derived information enters training.

Rebuild the chapter from first principles

The claim I need to own is this: Cross-validation estimates generalization when folds match deployment. Hyperparameter search, nested validation, learning curves, subgroup metrics, uncertainty, and temporal splits expose overfitting and drift. Leakage occurs whenever future or outcome-derived information enters training. 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 model assessment, resampling, and leakage. 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 model assessment, resampling, and leakage 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 and dimension reduction to interpretability, fairness, and deployment monitoring. 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.

The full working model

- **Core claim.** Cross-validation estimates generalization when folds match deployment.
- **What moves.** Hyperparameter search, nested validation, learning curves, subgroup metrics, uncertainty, and temporal splits expose overfitting and drift.

- **What keeps the answer honest.** Leakage occurs whenever future or outcome-derived information enters training.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

I audit the entire data pipeline, not just the estimator, because leakage usually hides before the model call.

The easy version

Cross-validation estimates generalization when folds match deployment.

Worked example

Question. A feature records whether a loan was sent to collections. Can it predict default at approval time?

Walkthrough. No. It occurs after the decision and encodes the outcome. Using it would create target leakage and impossible deployment performance.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. A feature records whether a loan was sent to collections. Can it predict default at approval time?

Solution. No. It occurs after the decision and encodes the outcome. Using it would create target leakage and impossible deployment performance.

Practice 2. Explain model assessment, resampling, and leakage to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct model assessment, resampling, and leakage 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 model assessment, resampling, and leakage 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 Interpretability, fairness, and deployment monitoring

The idea

Feature importance, partial dependence, local explanations, calibration, error analysis, and sensitivity clarify model behavior without turning association into causation. Deployment requires documentation, privacy, subgroup review, drift monitoring, fallback rules, and human accountability.

Rebuild the chapter from first principles

The claim I need to own is this: Feature importance, partial dependence, local explanations, calibration, error analysis, and sensitivity clarify model behavior without turning association into causation. Deployment requires documentation, privacy, subgroup review, drift monitoring, fallback rules, and human accountability. 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 interpretability, fairness, and deployment monitoring. 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 interpretability, fairness, and deployment monitoring 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 model assessment, resampling, and leakage 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.

The full working model

- **Core claim.** Feature importance, partial dependence, local explanations, calibration, error analysis, and sensitivity clarify model behavior without turning association into causation.
- **What moves.** Deployment requires documentation, privacy, subgroup review, drift monitoring, fallback rules, and human accountability.
- **What keeps the answer honest.** The model becomes useful only when I can apply interpretability, fairness, and deployment monitoring to an unfamiliar case and defend the assumptions.
- **Mastery standard.** I can define the objects, rebuild the rule, work a fresh case, compare alternatives, and diagnose a broken solution without relying on recognition.

How I actually think about it

My shorthand for this chapter is: a model is not finished when the notebook score is high; it is finished when its use, limits, and failure response are explicit.

The easy version

Feature importance, partial dependence, local explanations, calibration, error analysis, and sensitivity clarify model behavior without turning association into causation.

Worked example

Question. A hiring model has equal overall accuracy but double the false-negative rate for one group. What should be done?

Walkthrough. Report the group-specific harm, inspect data and threshold causes, evaluate relevant fairness and legal criteria, redesign or constrain the system, and establish monitoring rather than hiding behind aggregate accuracy.

Common miss

The fastest way to get this chapter wrong is to use the visible procedure without naming its assumption, units, time period, reference group, or boundary. I put that condition beside the work and verify the result independently.

Solved chapter practice

Practice 1. A hiring model has equal overall accuracy but double the false-negative rate for one group. What should be done?

Solution. Report the group-specific harm, inspect data and threshold causes, evaluate relevant fairness and legal criteria, redesign or constrain the system, and establish monitoring rather than hiding behind aggregate accuracy.

Practice 2. Explain interpretability, fairness, and deployment monitoring to a classmate using one definition, one concrete example, one assumption, and one failure mode.

Solution. Begin with the core claim above, use the worked case as the concrete example, state the condition highlighted by the warning, and choose a failure where that condition does not hold. A complete explanation connects all four pieces instead of listing them.

Mastery practice A. Reconstruct interpretability, fairness, and deployment monitoring 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 interpretability, fairness, and deployment monitoring 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.

Cumulative study plan

I use this packet in three passes. First I reconstruct each model and work its examples without looking. Second I mix chapters so the tool is no longer named for me. Third I explain a complete case aloud, including assumptions, units, uncertainty, failure modes, and what evidence would change my conclusion.