

Study Delight Publications

From
Data
to
Decisions

MACHINE LEARNING

Simplified

PART 2

Algorithms & Implementation

Learn
Implement
Experiment
Improve



Supervised
Learning



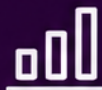
Unsupervised
Learning



Step-by-Step
Approach



Code
Examples



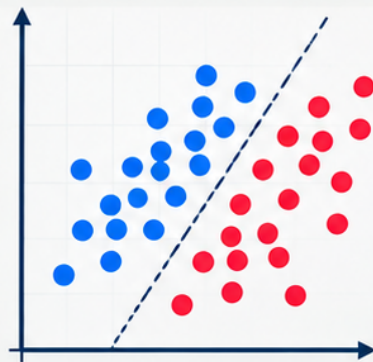
Visual
Learning



Real Datasets



Build Practical
Skills



● Class A

● Class B



Load Data



Train Model



Evaluate



Make Predictions



Decision
Trees



Ensemble
Methods

pythonforaiml.com

PART 2 • MACHINE LEARNING ALGORITHMS

Learn the major algorithms, the intuition behind them, and when to use each one.

30-PAGE ROADMAP



01 Part 2: Algorithm Thinking

02 Linear Regression

03 Best-Fit Line

04 Cost Function

05 Gradient Descent

06 Multiple Linear Regression

07 Polynomial Regression

08 Logistic Regression

09 K-Nearest Neighbors (KNN)

10 Decision Trees

11 Random Forest

12 Naive Bayes

13 Support Vector Machines

14 Feature Scaling

15 Encoding Categories

16 Training a Classifier

17 Confusion Matrix

18 Precision & Recall

19 F1 Score

20 Cross-Validation

21 Hyperparameter Tuning

22 K-Means Clustering

23 Elbow Method

24 Principal Component Analysis (PCA)

25 Model Comparison

26 End-to-End ML Pipeline

27 House Price Predictor Project

28 Customer Classifier Project

29 Algorithm Map

30 Master Checklist

LEARN • PRACTICE • BUILD • GROW

Use this index as your quick navigation map while studying.

PART 2 • MACHINE LEARNING ALGORITHMS

Now we move from ideas to algorithms

EXPLANATION

Part 1 gave us the language of machine learning. In this part we study the algorithms that actually learn patterns from data. The goal is to understand the intuition, the mathematics behind the important ideas, and when each algorithm is useful.

NOTE / FORMULA

Algorithm ladder: Regression → Classification → Trees → Neighbors → SVM → Clustering → Dimensionality Reduction

KEY IDEA

An algorithm is a method for learning a relationship from data. Different algorithms make different assumptions.

REMEMBER

Draw the algorithm ladder and circle the two methods you most want to learn first.

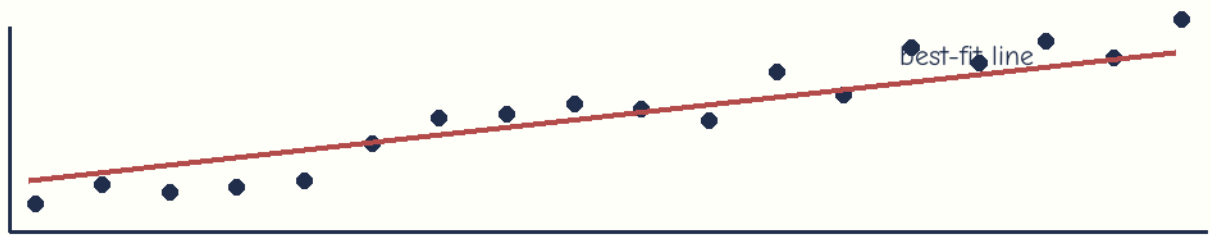
PRACTICE

Understand ☒ visualize ☒ code ☒ evaluate

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

LINEAR REGRESSION

Find the line that best represents the data



EXPLANATION

Linear regression predicts a continuous numeric value. Imagine plotting house area on the x-axis and price on the y-axis. If the points show an approximately straight trend, a line can summarize that relationship. The model learns the line's parameters from examples.

NOTE / FORMULA

$y = mx + b$ x = feature m = slope b = intercept y = predicted value

KEY IDEA

Key idea: the line is not expected to pass through every point. It should represent the overall relationship as well as possible.

REMEMBER

Given $x = 10$, $m = 3$ and $b = 5$, calculate y .

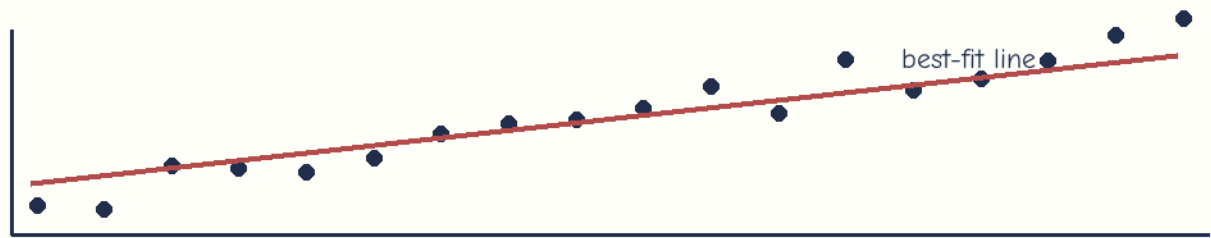
PRACTICE

Concept ☒ graph ☒ equation ☒ prediction

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

BEST-FIT LINE

Why one line is better than another



EXPLANATION

Many lines could be drawn through a scatter of points. Linear regression chooses parameters that minimize an error measure. The vertical gap between an actual point and the predicted line is called a residual or prediction error.

NOTE / FORMULA

Actual point • | residual ↓ —•— ← predicted line Residual = actual - predicted

KEY IDEA

A good line keeps the overall errors small rather than trying to perfectly fit every noisy observation.

REMEMBER

Sketch five points and draw a reasonable best-fit line. Mark one residual.

PRACTICE

Plot ☒ compare ☒ adjust

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

COST FUNCTION

Turn model mistakes into one number

EXPLANATION

A cost function summarizes how badly the model is performing. For ordinary linear regression, mean squared error is a common objective. Squaring makes large errors count more strongly and avoids positive and negative errors cancelling each other.

NOTE / FORMULA

$MSE = (1/n) \sum (y_i - \hat{y}_i)^2$ actual y_i predicted $\hat{y}_i$ error = $y_i - \hat{y}_i$

KEY IDEA

Training tries to find parameter values that make the chosen cost small.

REMEMBER

If actual values are [10, 12] and predictions are [8, 13], calculate the MSE.

PRACTICE

Predict $\boxtimes$ calculate error $\boxtimes$ square $\boxtimes$ average

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

GRADIENT DESCENT

How a model searches for better parameters

EXPLANATION

Gradient descent is an optimization method. Imagine the cost function as a landscape. The algorithm repeatedly checks the direction in which the cost increases and moves in the opposite direction. The learning rate controls the step size.

NOTE / FORMULA

Cost $\uparrow$ | $\bullet$ start | / | $\bullet$ | $\bullet$ | $\bullet$ _____ $\rightarrow$ parameter $\downarrow$ lower cost

KEY IDEA

Too large a learning rate can overshoot. Too small a rate can make learning unnecessarily slow.

REMEMBER

Explain why gradient descent needs both a direction and a step size.

PRACTICE

Measure slope $\boxtimes$ move downhill $\boxtimes$ repeat

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

MULTIPLE LINEAR REGRESSION

Use several features together

EXPLANATION

Real problems rarely depend on one variable. Multiple linear regression uses several features to predict one numeric target. Each feature receives a coefficient that represents its contribution while the other included features are held in the model.

NOTE / FORMULA

$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3$ Area $\neg$ Rooms $\mapsto$ Model $\rightarrow$ Price Age $\neg$

KEY IDEA

A coefficient is not automatically a causal statement. It describes the fitted mathematical relationship under the model assumptions.

REMEMBER

List three features you might use to predict house price.

PRACTICE

Features $\boxtimes$ coefficients $\boxtimes$ prediction

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

POLYNOMIAL REGRESSION

When the relationship bends

EXPLANATION

Sometimes a straight line is too simple. Polynomial regression adds powers such as x^2 or x^3 so the model can represent curved relationships. It is still based on linear regression in its coefficients, even though the curve in the original feature space is nonlinear.

NOTE / FORMULA

$y = b_0 + b_1x + b_2x^2$ Straight line: — Polynomial:  / \

KEY IDEA

More flexibility can help, but high-degree polynomials can overfit. Always compare performance on unseen data.

REMEMBER

Draw a curved trend and explain why a straight line might underfit it.

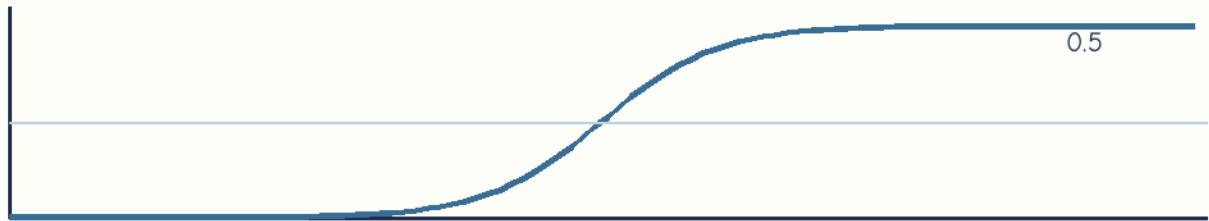
PRACTICE

Add complexity ☒ validate ☒ compare

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

LOGISTIC REGRESSION

A classic model for classification



EXPLANATION

Despite its name, logistic regression is commonly used for classification. It converts a linear score into a probability using the sigmoid function. A threshold can then turn that probability into a class decision.

NOTE / FORMULA

$\sigma(z) = 1 / (1 + e^{-z})$ probability 1 | — | / .5 | —●— | / 0 | —
 +————→ z

KEY IDEA

The sigmoid output lies between 0 and 1, making it useful as a probability-like score for binary classification.

REMEMBER

If the model gives probability 0.82 and the threshold is 0.5, what class is predicted?

PRACTICE

Score $\boxtimes$ probability $\boxtimes$ threshold $\boxtimes$ class

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

K-NEAREST NEIGHBORS

Let nearby examples vote

EXPLANATION

KNN predicts a new point by looking at nearby training examples. For classification, the nearest neighbors can vote on the class. For regression, their target values can be averaged. Distance is therefore central to the algorithm.

NOTE / FORMULA

Class A ● ● \ ★ new point / Class B ● Nearest points influence ★

KEY IDEA

K controls how many neighbors participate. Small K can be sensitive to noise; larger K can create smoother decisions.

REMEMBER

Why can feature scaling be important for KNN?

PRACTICE

Measure distance ☒ find neighbors ☒ vote

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

DECISION TREES

Ask a sequence of questions

EXPLANATION

A decision tree splits data using questions about features. Each split tries to create groups that are more useful for prediction. Trees are intuitive because the final prediction can be followed through a sequence of decisions.

NOTE / FORMULA

Age < 30? / \ YES NO / \ Spend > 5k? Low risk / \ High Low

KEY IDEA

A deep tree can memorize training examples. Limiting depth and using validation can improve generalization.

REMEMBER

Create a two-level tree for deciding whether to recommend a product.

PRACTICE

Question $\boxtimes$ split $\boxtimes$ branch $\boxtimes$ prediction

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

RANDOM FOREST

Many trees can work together

EXPLANATION

Random forest combines many decision trees. Each tree sees a randomized view of the training process, and their predictions are combined. For classification this often means voting; for regression it commonly means averaging.

NOTE / FORMULA

DATA / | \ Tree Tree Tree \ | / \ | / FINAL PREDICTION

KEY IDEA

The strength comes from combining diverse trees instead of relying on one tree's particular mistakes.

REMEMBER

Explain why many imperfect trees can outperform one unstable tree.

PRACTICE

Build trees ✕ combine ✕ generalize

STUDY LOOP: Understand ✕ visualize ✕ code ✕ evaluate

NAIVE BAYES

Use probability to classify

EXPLANATION

Naive Bayes uses Bayes' theorem with a simplifying independence assumption. It is especially well known for text classification because it can be fast and effective with high-dimensional word features.

NOTE / FORMULA

$P(\text{class} \mid \text{features}) \propto P(\text{class}) \times P(\text{features} \mid \text{class})$ Message $\rightarrow$ word
evidence $\rightarrow$ class probability

KEY IDEA

The "naive" assumption is often unrealistic, but the method can still work surprisingly well.

REMEMBER

Give three words that might increase the probability that an email is spam.

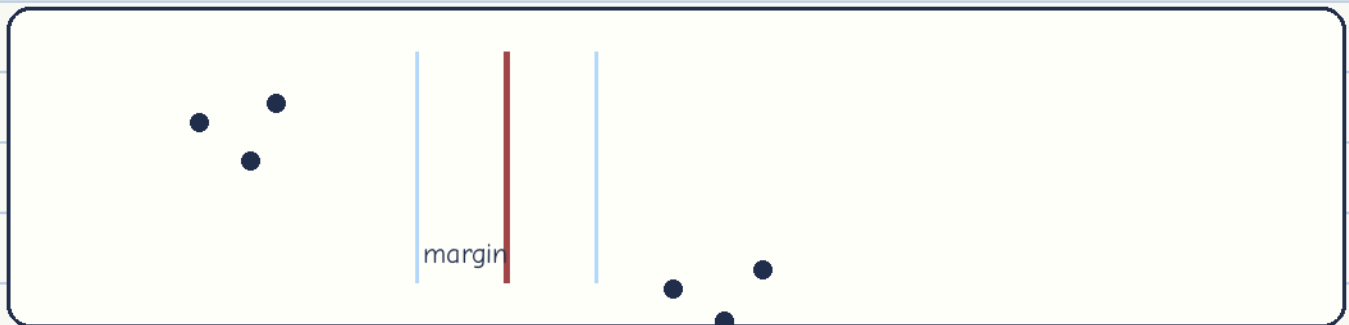
PRACTICE

Evidence $\boxtimes$ probability $\boxtimes$ class

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

SUPPORT VECTOR MACHINE

Find a boundary with a useful margin



EXPLANATION

Support Vector Machines search for a decision boundary that separates classes while maximizing the margin between the boundary and the closest training examples. Those closest examples are support vectors.

NOTE / FORMULA

Class A ● ● ● | ← margin → | ● | boundary | Class B ● ● | | ●

KEY IDEA

A larger margin can improve generalization. Kernel methods can also represent nonlinear boundaries.

REMEMBER

What does the word margin mean in an SVM?

PRACTICE

Separate ☒ maximize margin ☒ predict

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

FEATURE SCALING

Put features on comparable scales

EXPLANATION

Distance-based and gradient-based methods can be strongly affected when one feature has a much larger numerical range than another. Scaling transforms features so their magnitudes are more comparable. Standardization is a common approach.

NOTE / FORMULA

Before: Age: 20–60 Income: 20,000–200,000 After scaling: features have comparable numerical magnitude

KEY IDEA

Scaling is especially important for methods such as KNN, K-Means and many gradient-based models. Tree-based models generally do not require it in the same way.

REMEMBER

Give an example where one feature's units could dominate a distance calculation.

PRACTICE

Inspect ranges ☒ scale ☒ train

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

ENCODING CATEGORIES

Turn labels into model-friendly numbers

EXPLANATION

Machine-learning models usually require numerical representations. A categorical feature such as city can be represented using one-hot encoding, where each category becomes a separate binary column. The encoding must be handled consistently between training and future data.

NOTE / FORMULA

City Pune $\rightarrow$ [1,0,0] Mumbai $\rightarrow$ [0,1,0] Delhi $\rightarrow$ [0,0,1]

KEY IDEA

Do not assign arbitrary numbers such as Pune=1, Mumbai=2, Delhi=3 when those numbers would falsely imply an ordered distance.

REMEMBER

Encode three product categories using one-hot vectors.

PRACTICE

Categorical value $\rightarrow$ representation $\rightarrow$ model

STUDY LOOP: Understand $\rightarrow$ visualize $\rightarrow$ code $\rightarrow$ evaluate

TRAINING A CLASSIFIER

A practical scikit-learn pattern

EXPLANATION

Once X and y are prepared, many scikit-learn estimators follow the same interface: create the model, call `fit` on training data, call `predict` on unseen data, and then evaluate the predictions.

NOTE / FORMULA

```
from sklearn.tree import DecisionTreeClassifier model =  
DecisionTreeClassifier(max_depth=3) model.fit(X_train, y_train) pred =  
model.predict(X_test)
```

KEY IDEA

`fit()` learns from training data. `predict()` applies the learned model to new feature rows.

REMEMBER

Write the four core lines for training a classifier.

PRACTICE

Prepare $\boxtimes$ fit $\boxtimes$ predict $\boxtimes$ evaluate

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

CONFUSION MATRIX

See which mistakes the classifier makes

EXPLANATION

A confusion matrix compares predicted classes with actual classes. For binary classification it separates true positives, true negatives, false positives and false negatives. This is more informative than one accuracy number when the costs of errors differ.

NOTE / FORMULA

Predicted Positive Negative Actual +ve TP FN -ve FP TN

KEY IDEA

False positives and false negatives are different kinds of mistakes. The important one depends on the application.

REMEMBER

For a disease-screening example, explain why false negatives may be especially important.

PRACTICE

Count ☒ inspect ☒ interpret

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

PRECISION & RECALL

Different questions about positive predictions

EXPLANATION

Precision asks: of the examples predicted positive, how many were actually positive? Recall asks: of all truly positive examples, how many did we find? They emphasize different types of errors.

NOTE / FORMULA

Precision = $TP / (TP + FP)$ Recall = $TP / (TP + FN)$ Precision → avoid false alarms Recall → avoid missed positives

KEY IDEA

There is often a trade-off between precision and recall as the decision threshold changes.

REMEMBER

Give a situation where high recall is more important than high precision.

PRACTICE

Ask the right metric question

F1 SCORE

Balance precision and recall

EXPLANATION

F1 is the harmonic mean of precision and recall. It is useful when you want a single score that rewards both high precision and high recall. Because it is a harmonic mean, a very low value in one component pulls the score down.

NOTE / FORMULA

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) \div (\text{Precision} + \text{Recall})$$

KEY IDEA

F1 is not automatically the best metric. The correct metric depends on the actual decision problem.

REMEMBER

If precision is high but recall is very low, will F1 also be high? Why?

PRACTICE

Precision + recall $\neq$ balance

STUDY LOOP: Understand $\rightarrow$ visualize $\rightarrow$ code $\rightarrow$ evaluate

CROSS-VALIDATION

Use several validation folds

EXPLANATION

Cross-validation repeatedly divides the training data into folds. Each fold becomes a validation set while the others are used for training. The scores are then combined. This gives a more stable estimate than relying on one random split, especially for smaller datasets.

NOTE / FORMULA

Fold 1: TEST | train train train Fold 2: train | TEST train train Fold 3: train train | TEST train Fold 4: train train train | TEST

KEY IDEA

Cross-validation helps compare models or hyperparameters while preserving a final test set for honest evaluation.

REMEMBER

Explain why every example gets a chance to be in a validation fold.

PRACTICE

Split ✕ rotate ✕ score ✕ average

STUDY LOOP: Understand ✕ visualize ✕ code ✕ evaluate

HYPERPARAMETER TUNING

Choose settings using evidence



two discovered groups

EXPLANATION

Hyperparameters are configuration choices such as tree depth, K in KNN or regularization strength. Tuning means trying reasonable choices and comparing them using validation or cross-validation rather than choosing based on training performance alone.

NOTE / FORMULA

depth=2 → score depth=4 → score depth=6 → score depth=8 → score Choose using validation evidence

KEY IDEA

A good search is systematic and constrained. Trying every possible setting is not always practical.

REMEMBER

Pick three K values you would test for KNN and explain your choice.

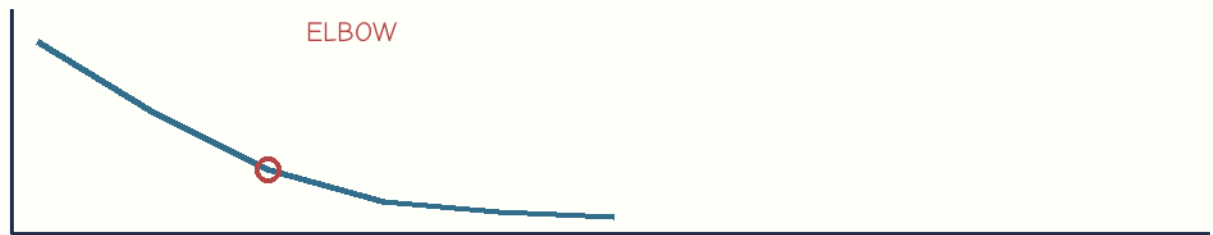
PRACTICE

Choose ☒ validate ☒ compare

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

K-MEANS CLUSTERING

Discover groups without labels



EXPLANATION

K-Means is an unsupervised algorithm that divides examples into K clusters. It alternates between assigning points to the nearest centroid and moving each centroid to the mean of its assigned points.

NOTE / FORMULA

● ● ○ ○ ● + ○ + = centroid 1. assign points 2. move centroids 3. repeat

KEY IDEA

You must choose K. The resulting clusters are mathematical groupings; they still need human interpretation.

REMEMBER

What could three customer clusters represent in a retail dataset?

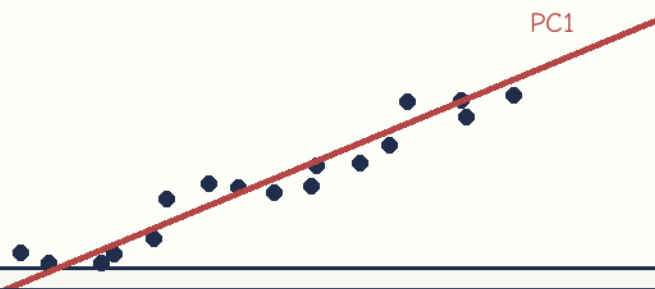
PRACTICE

Initialize ☒ assign ☒ update ☒ repeat

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

ELBOW METHOD

A practical clue for choosing K



EXPLANATION

The elbow method compares clustering cost for different K values. As K increases, within-cluster variation generally decreases. We look for a point where additional clusters provide diminishing improvement—the “elbow.”

NOTE / FORMULA

Cost ↑ | ● | ● | ● | ● — + —————→ K ↑ elbow

KEY IDEA

The elbow is a heuristic, not a universal law. Domain knowledge and other validation methods can also help.

REMEMBER

Sketch a cost-vs-K curve and mark the elbow.

PRACTICE

Try ☒ plot ☒ inspect ☒ justify

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

PCA

Reduce dimensions while preserving important variation

EXPLANATION

Principal Component Analysis transforms features into new directions called principal components. The first component captures the greatest amount of variance possible, the next captures the greatest remaining variance under the method's constraints, and so on.

NOTE / FORMULA

Original axes $\uparrow \cdot \cdot \mid \cdot \cdot \mid ______ \rightarrow \nearrow$ PC1 PC1 follows the main direction of variation

KEY IDEA

PCA can help visualize high-dimensional data and compress representations, but the new components may be harder to interpret.

REMEMBER

Why might reducing 20 features to 2 help visualization?

PRACTICE

Center $\boxtimes$ transform $\boxtimes$ inspect variance

STUDY LOOP: Understand $\boxtimes$ visualize $\boxtimes$ code $\boxtimes$ evaluate

MODEL COMPARISON

Accuracy is only one part of the decision

EXPLANATION

Suppose three models have different scores. The best model is not necessarily the one with the highest training accuracy. Compare validation/test metrics, inference cost, interpretability, stability, data requirements and the consequences of mistakes.

NOTE / FORMULA

Model A: high accuracy, hard to explain Model B: slightly lower, easy to explain Model C: fast, compact, moderate score Choose for the problem—not the leaderboard.

KEY IDEA

A production choice balances technical performance with practical constraints.

REMEMBER

Create a five-column model comparison table on paper.

PRACTICE

Measure ☒ compare ☒ contextualize

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

END-TO-END ML PIPELINE

Connect preprocessing, training and evaluation

EXPLANATION

A real ML workflow chains multiple steps. Data is loaded, cleaned, transformed, split correctly, used to train a baseline and then evaluated. Pipelines help ensure preprocessing is applied consistently and reduce accidental leakage.

NOTE / FORMULA

Raw data ↓ Clean ↓ Transform ↓ Train model ↓ Evaluate ↓ Save + monitor

KEY IDEA

Start with a simple baseline. Only add complexity when evidence shows it helps.

REMEMBER

Draw the pipeline for a customer-churn classifier.

PRACTICE

Build small ☒ measure ☒ improve

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

PROJECT • HOUSE PRICE PREDICTOR

A complete regression exercise

EXPLANATION

Build a small regression project using features such as area, bedrooms and age. Explore the data, handle missing values, split into training and test sets, fit a baseline linear regression model, evaluate with MAE, and compare it with a tree-based model.

NOTE / FORMULA

```
X = df[['Area', 'Bedrooms', 'Age']] y = df['Price'] Train →  
LinearRegression Test → MAE Compare → Tree model
```

KEY IDEA

Do not focus only on the final score. Explain what data was used, what assumptions were made and where the model may fail.

REMEMBER

Write the project steps in your own words before coding.

PRACTICE

Question ☒ data ☒ baseline ☒ evaluate ☒ compare

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

PROJECT • CUSTOMER CLASSIFIER

A practical classification exercise

EXPLANATION

Build a classifier that predicts a customer category such as likely-to-renew versus unlikely-to-renew. Define the target carefully, prepare numeric and categorical features, split the data, train a baseline model and inspect the confusion matrix.

NOTE / FORMULA

Customer data ↓ Features + target ↓ Train classifier ↓ Predictions ↓
Confusion matrix ↓ Precision / Recall / F1

KEY IDEA

The metric should match the business consequence. If missing a likely-renewal customer is costly, recall may matter more than accuracy alone.

REMEMBER

Choose a threshold or metric you would monitor and explain why.

PRACTICE

Define ☒ train ☒ inspect mistakes ☒ improve

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

PART 2 • MASTER CHECKLIST

Before moving to deep learning

EXPLANATION

At this stage you should understand not only algorithm names but the questions behind them: What type of target do I have? What assumptions does the model make? Does scale matter? How should categories be represented? How will I evaluate mistakes? Could the model be overfitting or leaking information?

NOTE / FORMULA

Algorithm map: Regression → numeric target Classification → class target Clustering → unlabeled groups PCA → representation / dimensionality reduction Evaluation → evidence for decisions

KEY IDEA

The next part moves into neural networks and deep learning, where these foundations become building blocks for more powerful models.

REMEMBER

Explain linear regression, logistic regression, trees, KNN, SVM and K-Means in one sentence each.

PRACTICE

Recall ☒ explain ☒ choose ☒ build

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate

PART 2 • FINAL ALGORITHM MAP

Choose the algorithm by the problem, data and evidence

EXPLANATION

Do not choose an algorithm simply because it is popular. First identify the target, the amount and type of data, the required interpretability, the cost of mistakes and the evaluation metric. Then establish a baseline and compare alternatives using validation evidence.

NOTE / FORMULA

Regression → Linear / Polynomial Classification → Logistic / KNN / Tree / Forest / SVM Unsupervised → K-Means Representation → PCA

KEY IDEA

Every algorithm is a tool with strengths and weaknesses. The best model is the one that solves the actual problem reliably.

REMEMBER

Before Part 3, explain when you would try Linear Regression, Logistic Regression, Decision Trees, KNN, Random Forest, SVM and K-Means.

PRACTICE

Problem ☒ choose tool ☒ validate ☒ explain

STUDY LOOP: Understand ☒ visualize ☒ code ☒ evaluate