# ECON7880 Midterm Theory Cheat Sheet (Q1–20)

*Time‑savvy bullets. Plain language. Use Ctrl/⌘‑F to jump to a keyword.*

| ## 0) One‑screen map of what’s being tested - **Q1–5: Supervised vs Unsupervised.** - *Supervised* = you have a clear target/label to predict (yes/no, number). - *Unsupervised* = no labels; you’re grouping/finding structure. - **Q6–10: True/False model basics.** Trees, complexity/overfitting, SVM, logistic regression, SVM C. - **Q11–20: Quick multiple‑choice concepts.** Laplace smoothing, R² & MSE, train vs test, fitting graph, model complexity, tree params, what logistic regression is/isn’t, SVM margin/hinge loss, SVM distances. |
| --- |
| ## 1) Fast rules: Supervised vs Unsupervised (Q1–5) - **Predict default / purchase / next quarter sales?** → **Supervised** (classification/regression). - **“Separate/Group/Cluster into types with no given labels”?** → **Unsupervised** (clustering/segmentation). - **Using known gender/purchases to split customers?** If the *goal* is grouping without a target label → **Unsupervised**. |

## 2) True/False quick hits (Q6–10)

* **Decision Trees are non‑parametric** (no fixed form; they grow to fit data) → **True**.
* **Adding complexity usually improves *test* performance** → **False**. It fits train better, can hurt generalization (overfitting).
* **SVMs can model non‑linear relations with kernels** → **True**.
* **“Logistic regression is unregularized”** → **False**. Modern implementations include regularization (e.g., L2) to control coefficient size.
* **SVM C smaller ⇒ stronger regularization** (wider margin, tolerate more violations) → **True**.

| ## 3) Laplace smoothing for leaf probabilities (Q11) **When a decision tree leaf has class counts and you want a non‑zero, smoothed probability:** - Let **N** = total in leaf, **k** = number of classes, **n\_c** = count of class **c**. - **Laplace estimate:** (P(cleaf) = ) - *Example:* counts = {High: 30, Medium: 10, Low: 10}, **N = 50**, **k = 3** → (P(High) = (30+1)/(50+3)). - **Why:** avoids zero‑probability and is conservative for small leaves. |
| --- |
| ## 4) Linear regression model metrics (Q12) - **R² (coefficient of determination):** fraction of variance in **y** explained by the model. **Higher is better** (0 to 1). - **MSE (mean squared error):** average squared residual. **Lower is better**. |

## 5) Train vs test behavior (Q13)

* Perfect on **train** (0 error) **≠** perfect on **test**. Test error typically **> 0** due to generalization gap and noise.

| ## 6) Fitting graph facts (Q14) - **Shows** training & generalization (validation/test) performance **vs model complexity** (e.g., tree depth, polynomial degree). - **Doesn’t show** generalization **vs training data size** (that’s a **learning curve**, different plot). |
| --- |
| ## 7) Model complexity truths (Q15) - **Complex models**: - **Overfit more** (memorize noise) → generalization **worse**. - **Less interpretable**. - **Usually slower** to train/infer than simpler ones (NOT “faster”). |

## 8) DecisionTreeClassifier size controls (Q16)

Use these to stop trees from overgrowing: - max\_depth — cap depth. - max\_leaf\_nodes — cap number of leaves. - min\_samples\_leaf — minimum samples per leaf (prevents tiny leaves). **Not a tree control:** C is an **SVM** regularization parameter, not for trees.

| ## 9) Logistic regression essentials (Q17–18) - **Task type:** *Linear* model used for a **discrete** outcome (classification). - **Link:** linear score → **log‑odds**; probability via the **sigmoid**: (P(y=1x) = (w^x + b) = ). - **Interpretation:** linear score **= log‑odds**, **not** odds. Odds = (e^{score}). - **Features:** must be **numeric**; encode categories first (one‑hot). |
| --- |
| ## 10) SVM quick box (Q19–20) - **Geometric margin & C:** smaller **C** ⇒ stronger regularization (wider margin, more tolerance to violations); larger **C** ⇒ fit training harder. - **Hinge loss:** ((0, 1 - y f(x))). Points **inside margin** but correctly classified ((0 < y f(x) < 1)) incur **some** loss; points on the **wrong side** ((y f(x) < 0)) incur **> 1** loss. Therefore, “violating the hyperplane” gives **larger** hinge loss than “violating margin only”. - **Kernels:** enable non‑linear decision boundaries by mapping to high‑dimensional feature spaces. - **Signed distance to hyperplane:** For hyperplane (w^x + w\_0 = 0), ((x) = ). Using the **label** (y {}), the **signed, label‑corrected distance** is (). If (w = 1), this reduces to **(y,(w^x + w\_0))** — i.e., perpendicular distance to the hyperplane with the correct sign. |

## 11) Overfitting & avoidance (for several Qs)

* **Symptoms:** train gets better as complexity ↑; test peaks then worsens.
* **Controls:** regularization (e.g., SVM **C**, logistic **penalty**), tree **stopping** params, cross‑validation, more data, simpler features.

| ## 12) Mini “identify the task” crib - **Predict next quarter revenue:** Supervised **regression**. - **Will default? buy/not buy?** Supervised **classification**. - **Group customers by behavior w/o labels:** **Clustering** (unsupervised). - **Split transactions into types by time/location w/o labels:** **Clustering**. |
| --- |
| ## 13) Handy formulas & snippets - **Laplace:** ((n\_c + 1)/(N + k)). - **R²:** ( - ). Higher better. - **MSE:** ((y - y)^2). Lower better. - **Logistic:** (P=1/(1+e{-(wx + b)})); score = **log‑odds**. - **SVM hinge:** ((0, 1 - y f(x))). Smaller **C** = stronger reg. |

## 14) Common trick questions

* **“Complexity helps test performance.”** Only up to a point; then hurts.
* **“Logistic scores are odds.”** They are **log‑odds**; odds = (e^{score}).
* **“Tree uses C to control size.”** No; that’s SVM.
* **“Fitting graph shows performance vs data size.”** That’s a **learning curve**, not a fitting graph.

## 15) 5‑second checks

* **Overfit?** Train ≪ Test.
* **Underfit?** Both bad, close together.
* **Right zone?** Train good, Test near its best.
* **Need prob. in leaf?** Laplace.
* **SVM C?** Smaller = softer/more regularized.
* **LogReg output?** Probability via sigmoid; linear part is **log‑odds**.