

# VU LEARNING HUB

[vulearninghub.xyz](https://vulearninghub.xyz)

**HANDOUT**

## **CS725**

Data Mining

**CS725 by VU Learning HUB**

Prepared for VU students • Free study resources

<https://vulearninghub.xyz>

# CS725

## Data Mining

---

This handout covers the complete course topic by topic in simple, exam-focused language. Each chapter begins with learning objectives and ends with key points and exam pointers. Definitions are boxed in green-friendly format, examples are worked out step by step, and very important exam points are flagged separately. Study the highlighted points carefully before your papers.

| Course Information |             |
|--------------------|-------------|
| Course Code        | CS725       |
| Course Title       | Data Mining |
| Credit Hours       | 3           |
| Level              | MS          |

## Table of Contents

|                                                |    |
|------------------------------------------------|----|
| Table of Contents                              | 2  |
| 1. The KDD Process and Data Mining Foundations | 3  |
| 2. Advanced Data Preprocessing                 | 4  |
| 3. Frequent Pattern Mining: Beyond Apriori     | 6  |
| 4. Classification: Trees, Rules and Ensembles  | 8  |
| 5. Classification: SVMs and Neural Networks    | 10 |
| 6. Clustering: Advanced Methods                | 12 |
| 7. Outlier Detection and Anomaly Mining        | 13 |
| 8. Mining Data Streams                         | 15 |
| 9. Graph and Network Mining                    | 16 |
| 10. Model Evaluation and Selection             | 18 |
| 11. Privacy, Ethics and Trends                 | 20 |
| Glossary of Key Terms                          | 21 |
| Practice Questions for Exams                   | 21 |
| MCQ Practice Test                              | 27 |
| Exam Strategy Guide                            | 32 |

# 1. The KDD Process and Data Mining Foundations

## Learning Objectives

- Position data mining inside the KDD pipeline
- Classify data mining tasks and data types

**Data mining** is defined as the nontrivial extraction of implicit, previously unknown and potentially useful information from data.

At MS level, data mining is studied as an end-to-end knowledge discovery process: (1) data selection and integration, (2) preprocessing and cleaning, (3) transformation and feature construction, (4) choice and application of mining algorithms, (5) pattern evaluation and knowledge presentation. Failure at any stage — especially preprocessing — invalidates everything downstream.

## Task taxonomy

- Predictive tasks: classification, regression, time-series forecasting — learn a mapping from inputs to a target.
- Descriptive tasks: clustering, association, summarization — characterize the data itself.
- Both are judged by interestingness: objective measures (support, confidence, lift) and subjective measures (novelty, actionability).

## Data types mined

- Relational and data-warehouse data; transactional and spatial data; text, web and multimedia data; data streams and time series; graph and network data.
- Each type demands specialized representations and algorithms — one size does not fit all.

## Key Points to Remember

- KDD = selection, preprocessing, transformation, mining, evaluation.
- Judge patterns by objective and subjective interestingness.

## Exam Pointer

Very important for exams: Contrast predictive vs descriptive tasks with two examples each.

## Research framing

- MS-level work: reproduce, then extend, then publish.
- Read survey papers before diving into algorithms.

- Keep a lab notebook of experiments.

### Interestingness measures

- Objective: support, confidence, lift — computable.
- Subjective: novelty, actionability — needs humans.
- Mine with objective, filter with subjective.

### Paper reading method

- Pass 1: abstract/figures; Pass 2: full read; Pass 3: reproduce.
- Question assumptions, not just results.
- Implement the baseline yourself.

### Thesis topic selection

- Pick a narrow, measurable problem.
- Ensure data and baselines exist.
- Novelty in method OR application suffices.

### Data mining vs machine learning

- ML emphasizes prediction algorithms; mining emphasizes the full KDD process.
- The fields overlap heavily in practice.
- Mining adds databases, warehousing and deployment concerns.
- Job titles use them interchangeably.
- Examiners may ask for the distinction.

### Curse of dimensionality

- Distances concentrate in high dimensions.
- Data becomes sparse exponentially.
- More features can hurt, not help.
- Dimensionality reduction is the remedy.
- Quote Bellman for the term's origin.

## 2. Advanced Data Preprocessing

### Learning Objectives

- Handle high-dimensional and heterogeneous data
- Apply advanced transformation and reduction

- Missing data mechanisms: MCAR, MAR, MNAR — the mechanism decides whether deletion or imputation is valid.
- Imputation: mean/median, kNN imputation, model-based (EM), multiple imputation.
- Outlier treatment: winsorization, trimming, transformation (log, Box-Cox).
- Categorical encoding: one-hot, ordinal, target encoding; high-cardinality features need hashing or embeddings.

## Dimensionality reduction

- Feature selection: filter (correlation, chi-square, mutual information), wrapper (recursive elimination), embedded (LASSO).
- Feature extraction: PCA finds orthogonal directions of maximum variance; used for visualization and de-noising.
- Curse of dimensionality: in high dimensions distances concentrate and data becomes sparse — reduction is not optional.

PCA: project  $X$  onto eigenvectors of the covariance matrix with largest eigenvalues

### Key Points to Remember

- Know your missingness mechanism before imputing.
- PCA = variance-preserving projection; selection keeps original features.

### Exam Pointer

Very important for exams: Explain the curse of dimensionality and two remedies.

## Data leakage

- Scaling before splitting leaks test info into training.
- Always split first, fit preprocessing on train only.
- Leakage is the silent killer of published results.

## Curse of dimensionality demo

- In high-D, all points become nearly equidistant.
- Volume concentrates in shells — intuition breaks.
- Reduce dimensions before distance methods.

## Experiment tracking

- Log every hyperparameter and seed.

- Version datasets, not just code.
- Reproducibility is a research virtue.

### Statistical significance in mining

- Multiple testing inflates false discoveries.
- Bonferroni/Holm corrections adjust.
- Effect sizes matter more than p-values.

### Feature hashing

- Hashes features into fixed-size vectors.
- Handles massive vocabularies in text mining.
- Collisions rarely hurt accuracy.
- Memory-bounded by design.
- Used in large-scale ad prediction.

### SMOTE for imbalance

- Synthesizes minority-class examples.
- Interpolates between neighbours.
- Better than naive oversampling.
- Can blur class boundaries.
- Combine with undersampling often.

## 3. Frequent Pattern Mining: Beyond Apriori

### Learning Objectives

- Explain FP-growth in detail
- Discuss closed and maximal patterns

Apriori suffers from repeated database scans and explosive candidate generation. FP-growth compresses the transaction database into an FP-tree (frequent-pattern tree) and mines it recursively without candidate generation.

### FP-growth steps

1. Scan once: find frequent 1-itemsets, sort by descending frequency.
2. Scan again: insert each transaction (ordered) into the FP-tree, sharing prefixes.
3. Mine recursively: for each item, build its conditional pattern base and conditional FP-tree.

- Result: same frequent itemsets as Apriori, typically orders of magnitude faster on dense data.

## Closed and maximal itemsets

- Closed itemset: no superset has the same support — lossless compression of all frequent itemsets.
- Maximal itemset: no superset is frequent — lossy but compact.
- Constraint-based mining: push constraints (e.g.  $\text{sum}(\text{price}) > 100$ ) deep into the algorithm.

### Key Points to Remember

- FP-growth = compress then divide-and-conquer; no candidates.
- Closed sets preserve all information; maximal sets are a compact summary.

### Exam Pointer

Very important for exams: Build the FP-tree for a small transaction set — the signature MS-level numerical.

## FP-tree example trace

- Order items by frequency; insert transactions sharing prefixes.
- Mine each item's conditional base recursively.
- Trace once by hand to truly understand.

## Maximal vs closed mining

- MAFIA mines maximal sets efficiently.
- CLOSET mines closed sets.
- Choose by whether lossy summary suffices.

## Constraint pushing

- Push constraints deep: monotone, succinct, convertible.
- Prunes search space dramatically.
- Theory meets practice here.

## Pattern set reduction

- Too many patterns overwhelm users.
- Summarize: closed, maximal, top-k, compressed.
- KRIMP uses MDL for non-redundant sets.

## Eclat algorithm

- Vertical format: item -> transaction list.
- Intersections compute support fast.
- Depth-first search like FP-growth.
- Memory-heavy on dense datasets.
- Know its format advantage for exams.

## Closed and maximal itemsets

- Closed: no superset with same support.
- Maximal: no frequent superset at all.
- Both compress the pattern output.
- Lossless vs lossy compression of results.
- CHARM and MAFIA mine them.

# 4. Classification: Trees, Rules and Ensembles

### Learning Objectives

- Compare splitting criteria
- Explain bagging, boosting and random forests

- Splitting criteria: information gain (ID3), gain ratio (C4.5, corrects multi-valued bias), Gini index (CART).
- Rule-based classifiers: extract IF-THEN rules from trees or mine directly (RIPPER); rule ordering by accuracy/coverage.
- Overfitting control: prepruning (stop early), postpruning (cost-complexity), validation sets.

## Ensemble methods

- Bagging: train on bootstrap samples, majority vote — reduces variance (random forests add random feature subsets at each split).
- Boosting (AdaBoost): sequentially reweight misclassified instances; weak learners combine into a strong classifier — reduces bias.
- Random forests: accurate, robust to noise, give feature importance; the default strong baseline for tabular data.
- Stacking: blend diverse models with a meta-learner.

**Key Points to Remember**

- Bagging reduces variance; boosting reduces bias.
- Random forest feature importance guides feature selection.

**Exam Pointer**

Very important for exams: Contrast bagging and boosting on four dimensions: sampling, weighting, error type, parallelization.

**Gradient boosting intuition**

- Each tree fixes the previous errors — sequential correction.
- XGBoost/LightGBM dominate tabular competitions.
- Tune learning rate and depth first.

**CART details**

- Binary splits, Gini index, cost-complexity pruning.
- Handles regression too.
- Foundation of random forests.

**Rule extraction from nets**

- Distil networks into interpretable rules.
- Fidelity vs comprehensibility tradeoff.
- Needed for regulated domains.

**Cost-sensitive trees**

- Weight classes by misclassification cost.
- Alternatively tune the decision threshold.
- Business cost should drive, not accuracy.

**Gradient boosting intuition**

- Each tree corrects the previous errors.
- Shrinkage (learning rate) prevents overfitting.
- XGBoost and LightGBM dominate competitions.
- Handles missing values natively.
- Tune depth and rate together.

**Calibration of classifiers**

- Predicted probabilities should match frequencies.
- Platt scaling and isotonic regression fix miscalibration.
- Matters when probabilities drive decisions.
- Reliability diagrams diagnose it.
- Naive Bayes is famously miscalibrated.

## 5. Classification: SVMs and Neural Networks

### Learning Objectives

- Explain maximum-margin classification and kernels
- Describe neural network training

### Support Vector Machines

- Find the hyperplane maximizing the margin between classes; support vectors are the critical boundary points.
- Soft margin (parameter C) tolerates misclassifications for better generalization.
- Kernel trick: implicitly map to high-dimensional space (RBF, polynomial kernels) to separate nonlinear data without computing the mapping.
- Strengths: effective in high dimensions, memory-efficient; weaknesses: tuning, slow on very large data.

### Neural networks

- Multilayer perceptrons: input → hidden layers → output; universal approximators.
- Backpropagation: gradient descent on the error via the chain rule.
- Regularization: dropout, weight decay, early stopping combat overfitting.
- Deep learning extends this with CNNs (images), RNNs/transformers (sequences) — representation learning replaces manual features.

SVM objective: minimize  $(1/2)||w||^2 + C \times \text{SUM slack}_i$  subject to margin constraints

### Key Points to Remember

- Kernels make linear methods nonlinear; C trades margin width against errors.
- Backprop + regularization make deep nets trainable.

### Exam Pointer

Very important for exams: Explain the kernel trick and why the RBF kernel is popular.

### Kernel choice guide

- Linear: text, high-dim; RBF: general nonlinear; polynomial: interactions.
- Scale features before kernels — always.
- Grid-search C and gamma together.

### Neural net regularization

- Dropout, weight decay, early stopping, batch norm.
- More data beats more regularization.
- Validate everything.

### AutoML landscape

- Searches architectures and hyperparameters.
- Democratizes modelling; experts still win on features.
- Watch compute budgets.

### Transfer learning in mining

- Pretrain on large data, fine-tune on small.
- Embeddings transfer across tasks.
- Reduces labelled-data needs.

### Kernel trick

- Implicitly maps data to high dimensions.
- RBF kernel handles non-linear boundaries.
- Mercer's theorem guarantees validity.
- Scales poorly to millions of rows.
- SVMs made kernels famous.

### Neural network regularization

- Dropout, weight decay, batch normalization.
- Data augmentation multiplies training data.
- Early stopping on validation loss.
- Architecture search automates design.
- Regularization enables deep models.

## 6. Clustering: Advanced Methods

### Learning Objectives

- Explain EM clustering and Gaussian mixtures
- Describe spectral and subspace clustering

### Model-based clustering (EM)

- Assume data comes from a mixture of distributions (e.g. Gaussians); EM alternates: E-step computes soft cluster memberships, M-step updates parameters.
- Soft assignments handle overlapping clusters; model selection via BIC/AIC.
- k-means is the hard-assignment special case of EM with spherical Gaussians.

### Beyond basic clustering

- Spectral clustering: use eigenvectors of the similarity graph Laplacian — finds non-convex clusters.
- Subspace clustering: clusters exist in different attribute subsets (high-dimensional data).
- Constraint-based: must-link / cannot-link background knowledge guides clustering.
- Evaluation without labels: silhouette, Davies-Bouldin index, Dunn index; stability analysis.

### Key Points to Remember

- EM gives probabilistic clusters; spectral clustering exploits graph structure.
- Internal validation indices replace accuracy when labels are absent.

### Exam Pointer

Very important for exams: One EM iteration on a tiny 1-D dataset — tests real understanding.

### GMM vs k-means

- GMM: soft, elliptical clusters; k-means: hard, spherical.
- GMM needs more data but models overlap.
- BIC selects the number of components.

### Spectral clustering steps

- Build similarity graph -> Laplacian -> eigenvectors -> k-means.
- Finds non-convex shapes.
- Choice of sigma/graph is critical.

## Cluster ensembles

- Combine multiple clusterings for robustness.
- Consensus functions merge partitions.
- Stability signals true structure.

## Density-based hierarchies

- HDBSCAN: hierarchy + stability extraction.
- No eps tuning needed.
- Modern favourite for exploration.

## Spectral clustering

- Uses graph Laplacian eigenvectors.
- Finds non-convex clusters.
- Expensive on large graphs.
- $k$  controls the embedding dimension.
- Beautiful theory, niche practice.

## BIRCH algorithm

- Builds a CF-tree in one pass.
- Scales to very large datasets.
- Sensitive to input order.
- Good for initial clustering passes.
- Know its single-pass advantage.

# 7. Outlier Detection and Anomaly Mining

### Learning Objectives

- Taxonomize outlier detection methods
- Apply detection in high-dimensional settings

- Global vs contextual vs collective outliers (a single hot day vs a hot day in winter vs an unusual sequence).
- Statistical: parametric (Gaussian, z-score) and non-parametric (histogram, KDE).
- Proximity: distance to  $k$ -th neighbour; density-based LOF compares local densities.
- Angle-based (ABOD): robust in high dimensions where distances concentrate.
- Isolation Forest: anomalies are isolated quickly by random partitions — fast and effective.

Very important for exams: in high dimensions, distance concentration makes proximity methods fail — use angle-based or isolation-based methods, or reduce dimensionality first.

### Key Points to Remember

- Match the method to the outlier type and dimensionality.
- Isolation Forest is the modern default for tabular anomalies.

### Exam Pointer

Very important for exams: Explain why kNN-based detection degrades in high dimensions.

## LOF worked logic

- Compare a point's density with neighbours' densities.
- LOF  $\gg 1$  signals outlier.
- Sensitive to minPts — test a range.

## Isolation forest intuition

- Anomalies isolate in few random splits.
- Linear time, low memory.
- The modern default for tabular outliers.

## High-D outlier angles

- ABOD uses angle variance — robust where distances fail.
- Expensive: approximations needed.
- Understand why distances concentrate first.

## Outlier ensembles

- Combine detectors like classifiers.
- Feature bagging improves robustness.
- Average scores, rank, or vote.

## Contextual outliers

- Anomalous only in context: 30°C in winter.
- Needs context attributes defined.
- Time-series spikes are contextual.
- Harder than point outliers.
- Domain knowledge defines the context.

## Collective outliers

- Groups anomalous together, normal apart.
- DDoS attacks show this pattern.
- Needs relationship modelling.
- Subgraph mining detects them.
- The hardest outlier type.

## 8. Mining Data Streams

### Learning Objectives

- State stream mining constraints
- Describe stream clustering and classification

- Constraints: infinite volume, single pass, limited memory, concept drift (the target changes over time).
- Sampling, sketching (Count-Min), sliding windows and damped windows summarize streams.
- Stream clustering: CluStream keeps micro-clusters online and macro-clusters offline on demand.
- Stream classification: Hoeffding trees (VFDT) split when the Hoeffding bound guarantees the best attribute; ADWIN detects drift and adapts.

Hoeffding bound: with probability  $1-\delta$ , true mean within  $\epsilon$  =  $\sqrt{R^2 \ln(1/\delta) / 2n}$

### Key Points to Remember

- Streams demand one-pass, adaptive algorithms; drift handling is mandatory.
- Sketches trade exactness for bounded memory.

### Exam Pointer

Very important for exams: List the four stream constraints and one algorithm addressing each.

## Sketch algorithms

- Count-Min: frequency estimates in tiny memory.
- Bloom filters: set membership with no false negatives.
- Probabilistic data structures trade exactness for scale.

### ADWIN drift detector

- Compares windows statistically; shrinks on drift.
- No manual thresholds.
- Pair with any stream learner.

### Hoeffding trees in practice

- VFDT handles millions of examples.
- Ties broken by tau threshold.
- Memory-bounded variants exist.

### Stream regression

- Adaptive windowing for changing targets.
- Fading factors weight recent data.
- Evaluate prequentially.

### Concept drift

- Data distribution changes over time.
- Fraud patterns evolve against detectors.
- Sliding windows adapt models.
- Detect drift before accuracy collapses.
- Retraining pipelines automate the response.

### Sketching algorithms

- Count-Min sketch estimates frequencies.
- Bloom filters test membership.
- Sublinear memory for streams.
- Probabilistic guarantees suffice.
- Essential stream-mining toolkit.

## 9. Graph and Network Mining

### Learning Objectives

- Define graph mining tasks
- Explain PageRank and community detection

- Tasks: frequent subgraph mining (gSpan), link prediction, community detection, node classification.
- PageRank: a page is important if important pages link to it — random surfer model; power iteration computes scores.
- HITS: hubs and authorities reinforce each other.
- Community detection: modularity optimization (Louvain), label propagation.
- Graph neural networks learn node embeddings for downstream tasks.

PageRank:  $PR(u) = (1-d)/N + d \times \sum_v PR(v)/\text{outdeg}(v) \text{ over in-neighbours } u$

### Key Points to Remember

- PageRank = eigenvector centrality via random surfer.
- Communities = densely connected groups; modularity measures quality.

### Exam Pointer

Very important for exams: Two PageRank iterations on a 3-node graph — compact and revealing.

## Louvain method

- Greedy modularity optimization in two phases.
- Fast enough for billion-edge graphs.
- Resolution limit can merge small communities.

## Node embedding uses

- DeepWalk, node2vec: random walks + word2vec.
- Embeddings feed classifiers and link predictors.
- Transductive — new nodes need retraining.

## Graph neural nets

- Message passing aggregates neighbour info.
- GCN, GraphSAGE, GAT variants.
- Oversmoothing limits depth.

## Temporal link prediction

- Who will connect next? Features from graph evolution.
- Applications: recommendations, fraud rings.

- Time-aware embeddings help.

## Community detection

- Louvain method maximizes modularity fast.
- Reveals social and biological modules.
- Overlapping communities need special methods.
- Resolution limit affects small groups.
- Visualize with force-directed layouts.

## Link prediction

- Who will connect next in a network?
- Common neighbours is the baseline.
- Adamic-Adar weights rare neighbours.
- Embeddings learn latent features.
- Drives friend recommendations.

# 10. Model Evaluation and Selection

### Learning Objectives

- Design rigorous evaluation protocols
- Compare models statistically

- Protocol: stratified k-fold cross-validation; nested CV when tuning hyperparameters; hold-out test set touched once.
- Metrics: accuracy, precision/recall/F1, ROC-AUC, PR-AUC (better for imbalance), log-loss, RMSE/MAE for regression.
- Statistical comparison: paired t-test, McNemar test, Wilcoxon signed-rank across datasets.
- Bias-variance tradeoff: model complexity, regularization paths, learning curves diagnose under/overfitting.

Very important for exams: never tune on the test set — that is data leakage and invalidates the evaluation. Report confidence intervals, not point estimates.

### Key Points to Remember

- Nested CV for tuning; test set for final reporting only.
- PR-AUC beats ROC-AUC on imbalanced problems.

**Exam Pointer**

Very important for exams: Design an evaluation protocol for a fraud dataset with 1% positives — justify every choice.

**Nested CV protocol**

- Inner loop tunes, outer loop evaluates — no optimistic bias.
- Expensive but honest.
- Required for small datasets.

**McNemar test**

- Compares two classifiers on the same test set.
- Focuses on disagreements.
- Nonparametric and simple.

**Learning curves**

- Plot error vs training size.
- Diagnose bias (both high) vs variance (gap).
- Decide: more data or better model?

**Model cards**

- Document intended use, limits, biases.
- Standard practice at top labs.
- Write one for your thesis model.

**Nested cross-validation**

- Outer loop estimates; inner loop tunes.
- Prevents optimistic bias from tuning.
- Expensive but honest.
- Required when model selection is part of the method.
- Report both loops in papers.

**Bias-variance tradeoff**

- Error decomposes into bias, variance, noise.
- Simple models: high bias; complex: high variance.
- Cross-validation finds the sweet spot.
- Ensembles reduce variance.

- The central tension of modelling.

## 11. Privacy, Ethics and Trends

### Learning Objectives

- Explain privacy-preserving techniques
- Survey current research directions
- k-anonymity, l-diversity, t-closeness limit re-identification; differential privacy gives formal guarantees via calibrated noise.
- Fairness: demographic parity, equalized odds; audit models for disparate impact.
- Explainability: SHAP, LIME, interpretable models for regulated decisions.
- Trends: automated ML (AutoML), federated learning, mining with large language models, causal discovery beyond correlation.

### Key Points to Remember

- Utility vs privacy is a fundamental tradeoff — quantify it.
- Responsible mining is now part of the professional standard.

### Exam Pointer

Very important for exams: Short essay: argue for or against deploying a black-box model in loan approval.

### Differential privacy basics

- Add calibrated noise; epsilon controls privacy loss.
- Composition: privacy budget depletes across queries.
- The gold standard of formal privacy.

### Fairness metrics

- Demographic parity, equalized odds, calibration.
- Impossible to satisfy all — choose deliberately.
- Report metrics by group.

### Causal discovery teaser

- Beyond correlation: PC algorithm, do-calculus.
- Needs assumptions (faithfulness).

- The next frontier of mining.

## Exam revision checklist

- FP-tree trace, EM iteration, PageRank iterations.
- Nested CV design, LOF logic, Hoeffding bound use.
- One essay: ethics of black-box models.

## Federated learning

- Trains across devices without centralizing data.
- Privacy-preserving by architecture.
- Communication efficiency is the bottleneck.
- Google's Gboard pioneered it.
- Regulatory pressure drives adoption.

## Differential privacy

- Noise calibrated to query sensitivity.
- Epsilon quantifies the privacy budget.
- Composition tracks cumulative loss.
- Apple and Google deploy it.
- The rigorous future of data privacy.

## Glossary of Key Terms

### Learning Objectives

- Revise all key terms quickly before exams

| Term        | Meaning                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------|
| Data mining | the nontrivial extraction of implicit, previously unknown and potentially useful information from data. |

### Exam Pointer

Very important for exams: Revise this glossary the night before the paper.

## Practice Questions for Exams

## Learning Objectives

- Test your preparation with these exam-style questions

## Short questions with model answers

### Example

Q: Explain: position data mining inside the KDD pipeline Model answer: KDD = selection, preprocessing, transformation, mining, evaluation. Judge patterns by objective and subjective interestingness.

### Example

Q: Explain: classify data mining tasks and data types Model answer: KDD = selection, preprocessing, transformation, mining, evaluation. Judge patterns by objective and subjective interestingness.

### Example

Q: Explain: handle high-dimensional and heterogeneous data Model answer: Know your missingness mechanism before imputing. PCA = variance-preserving projection; selection keeps original features.

### Example

Q: Explain: apply advanced transformation and reduction Model answer: Know your missingness mechanism before imputing. PCA = variance-preserving projection; selection keeps original features.

### Example

Q: Explain: explain FP-growth in detail Model answer: FP-growth = compress then divide-and-conquer; no candidates. Closed sets preserve all information; maximal sets are a compact summary.

### Example

Q: Explain: discuss closed and maximal patterns Model answer: FP-growth = compress then divide-and-conquer; no candidates. Closed sets preserve all information; maximal sets are a compact summary.

### Example

Q: Explain: compare splitting criteria Model answer: Bagging reduces variance; boosting reduces bias. Random forest feature importance guides feature selection.

### Example

Q: Explain: explain bagging, boosting and random forests Model answer: Bagging reduces variance; boosting reduces bias. Random forest feature importance guides feature selection.

### Example

Q: Explain: explain maximum-margin classification and kernels Model answer: Kernels make linear methods nonlinear; C trades margin width against errors. Backprop + regularization make deep nets trainable.

### Example

Q: Explain: describe neural network training Model answer: Kernels make linear methods nonlinear; C trades margin width against errors. Backprop + regularization make deep nets trainable.

### Example

Q: Explain: explain EM clustering and Gaussian mixtures Model answer: EM gives probabilistic clusters; spectral clustering exploits graph structure. Internal validation indices replace accuracy when labels are absent.

### Example

Q: Explain: describe spectral and subspace clustering Model answer: EM gives probabilistic clusters; spectral clustering exploits graph structure. Internal validation indices replace accuracy when labels are absent.

### Example

Q: Explain: taxonomize outlier detection methods Model answer: Match the method to the outlier type and dimensionality. Isolation Forest is the modern default for tabular anomalies.

### Example

Q: Explain: apply detection in high-dimensional settings Model answer: Match the method to the outlier type and dimensionality. Isolation Forest is the modern default for tabular anomalies.

### Example

Q: Explain: state stream mining constraints Model answer: Streams demand one-pass, adaptive algorithms; drift handling is mandatory. Sketches trade exactness for bounded memory.

### Example

Q: Explain: describe stream clustering and classification Model answer: Streams demand one-pass, adaptive algorithms; drift handling is mandatory. Sketches trade exactness for bounded memory.

### Example

Q: Explain: define graph mining tasks Model answer: PageRank = eigenvector centrality via random surfer. Communities = densely connected groups; modularity measures quality.

### Example

Q: Explain: explain PageRank and community detection Model answer: PageRank = eigenvector centrality via random surfer. Communities = densely connected groups; modularity measures quality.

### Example

Q: Explain: design rigorous evaluation protocols Model answer: Nested CV for tuning; test set for final reporting only. PR-AUC beats ROC-AUC on imbalanced problems.

### Example

Q: Explain: compare models statistically Model answer: Nested CV for tuning; test set for final reporting only. PR-AUC beats ROC-AUC on imbalanced problems.

### Example

Q: Explain: explain privacy-preserving techniques Model answer: Utility vs privacy is a fundamental tradeoff — quantify it. Responsible mining is now part of the professional standard.

### Example

Q: Explain: survey current research directions Model answer: Utility vs privacy is a fundamental tradeoff — quantify it. Responsible mining is now part of the professional standard.

## Long questions

- Write a detailed note on The KDD Process and Data Mining Foundations.

- Write a detailed note on Advanced Data Preprocessing.
- Write a detailed note on Frequent Pattern Mining: Beyond Apriori.
- Write a detailed note on Classification: Trees, Rules and Ensembles.
- Write a detailed note on Classification: SVMs and Neural Networks.
- Write a detailed note on Clustering: Advanced Methods.
- Write a detailed note on Outlier Detection and Anomaly Mining.
- Write a detailed note on Mining Data Streams.
- Write a detailed note on Graph and Network Mining.
- Write a detailed note on Model Evaluation and Selection.

## Long questions with model answer outlines

### Example

Q: Write a detailed note on The KDD Process and Data Mining Foundations. Answer outline: • Predictive tasks: classification, regression, time-series forecasting — learn a mapping from inputs to a target. • Descriptive tasks: clustering, association, summarization — characterize the data itself. • Both are judged by interestingness: objective measures (support, confidence, lift) and subjective measures (novelty, actionability). • Relational and data-warehouse data; transactional and spatial data; text, web and multimedia data; data streams and time series; graph and network data. • Each type demands specialized representations and algorithms — one size does not fit all.

### Example

Q: Write a detailed note on Advanced Data Preprocessing. Answer outline: • Missing data mechanisms: MCAR, MAR, MNAR — the mechanism decides whether deletion or imputation is valid. • Imputation: mean/median, kNN imputation, model-based (EM), multiple imputation. • Outlier treatment: winsorization, trimming, transformation (log, Box-Cox). • Categorical encoding: one-hot, ordinal, target encoding; high-cardinality features need hashing or embeddings. • Feature selection: filter (correlation, chi-square, mutual information), wrapper (recursive elimination), embedded (LASSO).

### Example

Q: Write a detailed note on Frequent Pattern Mining: Beyond Apriori. Answer outline: • 1. Scan once: find frequent 1-itemsets, sort by descending frequency. • 2. Scan again: insert each transaction (ordered) into the FP-tree, sharing prefixes. • 3. Mine recursively: for each item, build its conditional pattern base and conditional FP-tree. • Result: same frequent itemsets as Apriori, typically orders of magnitude faster on dense data. • Closed itemset: no superset has the same support — lossless compression of all frequent itemsets.

### Example

Q: Write a detailed note on Classification: Trees, Rules and Ensembles. Answer outline: • Splitting criteria: information gain (ID3), gain ratio (C4.5, corrects multi-valued bias), Gini index (CART). • Rule-based classifiers: extract IF-THEN rules from trees or mine directly (RIPPER); rule ordering by accuracy/coverage. • Overfitting control: prepruning (stop early), postpruning (cost-complexity), validation sets. • Bagging: train on bootstrap samples, majority vote — reduces variance (random forests add random feature subsets at each split). • Boosting (AdaBoost): sequentially reweight misclassified instances; weak learners combine into a strong classifier — reduces bias.

### Example

Q: Write a detailed note on Classification: SVMs and Neural Networks. Answer outline: • Find the hyperplane maximizing the margin between classes; support vectors are the critical boundary points. • Soft margin (parameter C) tolerates misclassifications for better generalization. • Kernel trick: implicitly map to high-dimensional space (RBF, polynomial kernels) to separate nonlinear data without computing the mapping. • Strengths: effective in high dimensions, memory-efficient; weaknesses: tuning, slow on very large data. • Multilayer perceptrons: input → hidden layers → output; universal approximators.

### Example

Q: Write a detailed note on Clustering: Advanced Methods. Answer outline: • Assume data comes from a mixture of distributions (e.g. Gaussians); EM alternates: E-step computes soft cluster memberships, M-step updates parameters. • Soft assignments handle overlapping clusters; model selection via BIC/AIC. • k-means is the hard-assignment special case of EM with spherical Gaussians. • Spectral clustering: use eigenvectors of the similarity graph Laplacian — finds non-convex clusters. • Subspace clustering: clusters exist in different attribute subsets (high-dimensional data).

### Example

Q: Write a detailed note on Outlier Detection and Anomaly Mining. Answer outline: • Global vs contextual vs collective outliers (a single hot day vs a hot day in winter vs an unusual sequence). • Statistical: parametric (Gaussian, z-score) and non-parametric (histogram, KDE). • Proximity: distance to k-th neighbour; density-based LOF compares local densities. • Angle-based (ABOD): robust in high dimensions where distances concentrate. • Isolation Forest: anomalies are isolated quickly by random partitions — fast and effective.

### Example

Q: Write a detailed note on Mining Data Streams. Answer outline: • Constraints: infinite volume, single pass, limited memory, concept drift (the target changes over time). • Sampling, sketching (Count-Min), sliding windows and damped windows summarize streams. • Stream clustering: CluStream keeps micro-clusters online and macro-clusters offline on demand. • Stream classification: Hoeffding trees (VFDT) split when the Hoeffding bound guarantees the best attribute; ADWIN detects drift and adapts. • Count-Min: frequency estimates in tiny memory.

### Example

Q: Write a detailed note on Graph and Network Mining. Answer outline: • Tasks: frequent subgraph mining (gSpan), link prediction, community detection, node classification. • PageRank: a page is important if important pages link to it — random surfer model; power iteration computes scores. • HITS: hubs and authorities reinforce each other. • Community detection: modularity optimization (Louvain), label propagation. • Graph neural networks learn node embeddings for downstream tasks.

### Example

Q: Write a detailed note on Model Evaluation and Selection. Answer outline: • Protocol: stratified k-fold cross-validation; nested CV when tuning hyperparameters; hold-out test set touched once. • Metrics: accuracy, precision/recall/F1, ROC-AUC, PR-AUC (better for imbalance), log-loss, RMSE/MAE for regression. • Statistical comparison: paired t-test, McNemar test, Wilcoxon signed-rank across datasets. • Bias-variance tradeoff: model complexity, regularization paths, learning curves diagnose under/overfitting. • Inner loop tunes, outer loop evaluates — no optimistic bias.

### Exam Pointer

Very important for exams: Practice writing answers within the exam time limit.

## MCQ Practice Test

### Learning Objectives

- Test your recall of the key points from every chapter

Q1. Which of the following was listed as a key point of 'The KDD Process and Data Mining Foundations' in this handout?

- A. PCA = variance-preserving projection; selection keeps original features.
- B. KDD = selection, preprocessing, transformation, mining, evaluation.
- C. FP-growth = compress then divide-and-conquer; no candidates.
- D. Know your missingness mechanism before imputing.

Q2. Which of the following was listed as a key point of 'The KDD Process and Data Mining Foundations' in this handout?

- A. PCA = variance-preserving projection; selection keeps original features.
- B. Judge patterns by objective and subjective interestingness.
- C. FP-growth = compress then divide-and-conquer; no candidates.
- D. Know your missingness mechanism before imputing.

Q3. Which of the following was listed as a key point of 'Advanced Data Preprocessing' in this handout?

- A. FP-growth = compress then divide-and-conquer; no candidates.
- B. Know your missingness mechanism before imputing.
- C. Judge patterns by objective and subjective interestingness.
- D. KDD = selection, preprocessing, transformation, mining, evaluation.

Q4. Which of the following was listed as a key point of 'Advanced Data Preprocessing' in this handout?

- A. Judge patterns by objective and subjective interestingness.
- B. PCA = variance-preserving projection; selection keeps original features.
- C. FP-growth = compress then divide-and-conquer; no candidates.
- D. KDD = selection, preprocessing, transformation, mining, evaluation.

Q5. Which of the following was listed as a key point of 'Frequent Pattern Mining: Beyond Apriori' in this handout?

- A. Judge patterns by objective and subjective interestingness.
- B. FP-growth = compress then divide-and-conquer; no candidates.
- C. Know your missingness mechanism before imputing.
- D. KDD = selection, preprocessing, transformation, mining, evaluation.

Q6. Which of the following was listed as a key point of 'Frequent Pattern Mining: Beyond Apriori' in this handout?

- A. Closed sets preserve all information; maximal sets are a compact summary.
- B. Judge patterns by objective and subjective interestingness.
- C. KDD = selection, preprocessing, transformation, mining, evaluation.
- D. Know your missingness mechanism before imputing.

Q7. Which of the following was listed as a key point of 'Classification: Trees, Rules and Ensembles' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.

- B. Bagging reduces variance; boosting reduces bias.
- C. Know your missingness mechanism before imputing.
- D. Judge patterns by objective and subjective interestingness.

Q8. Which of the following was listed as a key point of 'Classification: Trees, Rules and Ensembles' in this handout?

- A. Know your missingness mechanism before imputing.
- B. KDD = selection, preprocessing, transformation, mining, evaluation.
- C. Judge patterns by objective and subjective interestingness.
- D. Random forest feature importance guides feature selection.

Q9. Which of the following was listed as a key point of 'Classification: SVMs and Neural Networks' in this handout?

- A. Know your missingness mechanism before imputing.
- B. Kernels make linear methods nonlinear; C trades margin width against errors.
- C. Judge patterns by objective and subjective interestingness.
- D. KDD = selection, preprocessing, transformation, mining, evaluation.

Q10. Which of the following was listed as a key point of 'Classification: SVMs and Neural Networks' in this handout?

- A. Judge patterns by objective and subjective interestingness.
- B. Backprop + regularization make deep nets trainable.
- C. Know your missingness mechanism before imputing.
- D. KDD = selection, preprocessing, transformation, mining, evaluation.

Q11. Which of the following was listed as a key point of 'Clustering: Advanced Methods' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.
- B. EM gives probabilistic clusters; spectral clustering exploits graph structure.
- C. Know your missingness mechanism before imputing.
- D. Judge patterns by objective and subjective interestingness.

Q12. Which of the following was listed as a key point of 'Clustering: Advanced Methods' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.
- B. Judge patterns by objective and subjective interestingness.
- C. Internal validation indices replace accuracy when labels are absent.
- D. Know your missingness mechanism before imputing.

Q13. Which of the following was listed as a key point of 'Outlier Detection and Anomaly Mining' in this handout?

- A. Match the method to the outlier type and dimensionality.
- B. Know your missingness mechanism before imputing.
- C. KDD = selection, preprocessing, transformation, mining, evaluation.
- D. Judge patterns by objective and subjective interestingness.

Q14. Which of the following was listed as a key point of 'Outlier Detection and Anomaly Mining' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.
- B. Isolation Forest is the modern default for tabular anomalies.
- C. Judge patterns by objective and subjective interestingness.
- D. Know your missingness mechanism before imputing.

Q15. Which of the following was listed as a key point of 'Mining Data Streams' in this handout?

- A. Judge patterns by objective and subjective interestingness.
- B. Know your missingness mechanism before imputing.
- C. KDD = selection, preprocessing, transformation, mining, evaluation.
- D. Streams demand one-pass, adaptive algorithms; drift handling is mandatory.

Q16. Which of the following was listed as a key point of 'Mining Data Streams' in this handout?

- A. Judge patterns by objective and subjective interestingness.
- B. Know your missingness mechanism before imputing.
- C. Sketches trade exactness for bounded memory.
- D. KDD = selection, preprocessing, transformation, mining, evaluation.

Q17. Which of the following was listed as a key point of 'Graph and Network Mining' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.
- B. PageRank = eigenvector centrality via random surfer.
- C. Know your missingness mechanism before imputing.
- D. Judge patterns by objective and subjective interestingness.

Q18. Which of the following was listed as a key point of 'Graph and Network Mining' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.
- B. Judge patterns by objective and subjective interestingness.
- C. Know your missingness mechanism before imputing.

D. Communities = densely connected groups; modularity measures quality.

Q19. Which of the following was listed as a key point of 'Model Evaluation and Selection' in this handout?

- A. Know your missingness mechanism before imputing.
- B. KDD = selection, preprocessing, transformation, mining, evaluation.
- C. Nested CV for tuning; test set for final reporting only.
- D. Judge patterns by objective and subjective interestingness.

Q20. Which of the following was listed as a key point of 'Model Evaluation and Selection' in this handout?

- A. PR-AUC beats ROC-AUC on imbalanced problems.
- B. KDD = selection, preprocessing, transformation, mining, evaluation.
- C. Know your missingness mechanism before imputing.
- D. Judge patterns by objective and subjective interestingness.

Q21. Which of the following was listed as a key point of 'Privacy, Ethics and Trends' in this handout?

- A. Know your missingness mechanism before imputing.
- B. KDD = selection, preprocessing, transformation, mining, evaluation.
- C. Judge patterns by objective and subjective interestingness.
- D. Utility vs privacy is a fundamental tradeoff — quantify it.

Q22. Which of the following was listed as a key point of 'Privacy, Ethics and Trends' in this handout?

- A. KDD = selection, preprocessing, transformation, mining, evaluation.
- B. Judge patterns by objective and subjective interestingness.
- C. Responsible mining is now part of the professional standard.
- D. Know your missingness mechanism before imputing.

### Answer Key

1. B 2. B 3. B 4. B 5. B 6. A 7. B 8. D 9. B 10. B 11. B 12. C 13. A 14. B 15. D 16. C 17. B 18. D 19. C 20. A 21. D 22. C

### Exam Pointer

Very important for exams: Attempt all MCQs without looking back, then check the answer key.

# Exam Strategy Guide

## Learning Objectives

- Attempt papers with a winning strategy

## Before the exam

- Revise the Key Points boxes and glossary first — they carry the highest marks per minute.
- Practice one past paper under timed conditions; examiners repeat question styles.
- Sleep well the night before — an exhausted brain recalls far less.

## Attempting the paper

- Read the whole paper first; start with your strongest question to build momentum.
- Allocate time by marks: a 10-mark question deserves roughly double a 5-mark one.
- Structure long answers: definition, explanation, example/diagram, conclusion.

## Presentation matters

- Use headings, underline key terms, and draw labelled diagrams where possible.
- Attempt all questions — partial credit beats a blank page.
- Review your paper in the last 10 minutes for missed parts.

## Exam Pointer

Very important for exams: Very important for exams: neat, structured answers consistently score higher than longer messy ones.