

VU LEARNING HUB

vulearninghub.xyz

HANDOUT

CS513

Advanced Data Analytics and Business
Intelligence

CS513 by VU Learning HUB

Prepared for VU students • Free study resources

<https://vulearninghub.xyz>

VU STUDY HANDOUT

CS513

Advanced Data Analytics and Business Intelligence

This handout covers the full analytics pipeline: data warehousing and ETL, OLAP, data mining, statistical learning, visualization, and decision support - the technologies that turn raw organizational data into business intelligence.

13 lecture-wise topics | key terms | exam pointers

Table of Contents

UNIT 1: BUSINESS INTELLIGENCE FOUNDATIONS

Lecture 1: What is Business Intelligence?

Lecture 2: Data Warehousing Concepts

Lecture 3: OLAP and Multidimensional Analysis

UNIT 2: DATA MINING

Lecture 4: Data Mining Process and Tasks

Lecture 5: Classification Algorithms

Lecture 6: Ensemble Methods and Advanced Models

Lecture 7: Clustering and Association Rules

UNIT 3: ADVANCED ANALYTICS

Lecture 8: Regression and Time Series Forecasting

Lecture 9: Text Mining and Sentiment Analysis

Lecture 10: Data Visualization and Dashboards

UNIT 4: DECISION SUPPORT AND STRATEGY

Lecture 11: Decision Support Systems

Lecture 12: Big Data Technologies

Lecture 13: BI Strategy, Governance and Ethics

UNIT 1: BUSINESS INTELLIGENCE FOUNDATIONS

Lecture 1: What is Business Intelligence?

Learning Objectives

- Define BI and its components
- Explain the DIKW hierarchy
- Describe BI architecture

Key Terms & Definitions

Business Intelligence (BI)

Technologies and practices that transform data into actionable insight for decision-making.

DIKW hierarchy

Data -> Information -> Knowledge -> Wisdom: the value ladder of analytics.

Decision Support System (DSS)

Interactive system helping managers make semi-structured decisions.

Business Intelligence is the **umbrella** over data warehousing, reporting, OLAP, dashboards, data mining and analytics - everything that converts an organization's data exhaust into decisions. The classic **BI architecture**: source systems -> ETL -> data warehouse/data marts -> OLAP cubes -> dashboards/reports -> decision makers.

The **DIKW hierarchy** explains the value chain: raw **data** (transactions) becomes **information** (aggregated reports) becomes **knowledge** (patterns: "sales drop every Ramadan afternoon") becomes **wisdom** (decisions: adjust staffing). Analytics maturity climbs this ladder from descriptive to predictive to prescriptive.

DSS history matters: 1970s model-driven DSS (what-if spreadsheets), 1990s data-driven DSS (warehouses/OLAP), 2000s document/knowledge-driven, today AI-driven. Modern BI (Power BI, Tableau) is **self-service**: business users build their own dashboards without IT tickets.

- **Descriptive**: what happened (dashboards); **Diagnostic**: why (drill-down); **Predictive**: what will happen (ML); **Prescriptive**: what to do (optimization).
- **Self-service BI** shifts dashboard building from IT to analysts.
- **KPIs** must be SMART and tied to strategy (balanced scorecard).

VERY IMPORTANT — Exam Pointer

Very important: name the four analytics types with one business example each.

Section Summary

BI is the pipeline from operational data to decisions; maturity moves from describing the past to prescribing the future.

Lecture 2: Data Warehousing Concepts

Learning Objectives

- Define data warehouse characteristics
- Contrast OLTP vs OLAP
- Explain ETL

Key Terms & Definitions

Data warehouse

Subject-oriented, integrated, time-variant, non-volatile collection of data for decision support (Inmon).

OLTP vs OLAP

Online Transaction Processing (operational, normalized) vs Online Analytical Processing (analytical, denormalized).

ETL

Extract, Transform, Load: the pipeline feeding the warehouse.

A data warehouse is **subject-oriented** (organized around customers/products, not applications), **integrated** (consistent codes/formats), **time-variant** (history kept, e.g. 5-10 years), and **non-volatile** (loaded, not updated in place). Inmon's top-down (enterprise warehouse -> marts) vs Kimball's bottom-up (marts -> conformed warehouse) are the two design schools.

OLTP systems (POS, ERP) are normalized for fast writes; **OLAP** needs denormalized **star schemas** for fast reads: a central **fact table** (measures: sales amount, quantity) joined to **dimension tables** (time, product, store, customer). **Snowflake** normalizes dimensions further; **fact constellations** share dimensions across facts.

ETL: Extract (from sources, CDC - change data capture), Transform (clean, conform, deduplicate, handle slowly changing dimensions), Load (bulk into warehouse, indexed). Modern **ELT** loads raw into a lake then transforms in-database (dbt).

- **Star schema**: one fact table + denormalized dimensions - the analytical workhorse.
- **SCD Type 2**: keep history by versioning dimension rows (e.g. customer address changes).
- **Data marts**: departmental subsets (sales mart, finance mart).
- **Data lake**: raw schema-on-read storage; **lakehouse**: lake + warehouse ACID (Delta Lake).

Example

A retail warehouse: fact SALES (date_key, product_key, store_key, quantity, amount); dimensions TIME, PRODUCT, STORE enable "sales by category per quarter" in one scan.

VERY IMPORTANT — Exam Pointer

Exam pointer: draw/describe a star schema and explain SCD Type 2 with an example.

Section Summary

Warehouses denormalize history for analysis; ETL/ELT is the unglamorous work that makes BI possible.

Lecture 3: OLAP and Multidimensional Analysis

Learning Objectives

- Explain the data cube
- Perform OLAP operations
- Compare MOLAP/ROLAP/HOLAP

Key Terms & Definitions

Data cube

Multidimensional array of measures indexed by dimensions (time x product x region).

Slice/dice, drill-down/roll-up

Core OLAP navigation operations.

MOLAP / ROLAP

Multidimensional OLAP (cube storage) vs Relational OLAP (SQL on star schema).

The **data cube** generalizes the 2D crosstab: measures (sales) indexed by dimensions (time, product, geography). A cube with d dimensions has 2^d cuboids (all aggregation combinations); **materialized views** precompute the most-used ones.

OLAP operations: **slice** (fix one dimension: "2024 only"), **dice** (subcube: "2024, electronics, Punjab"), **drill-down** (year $\rightarrow$ quarter $\rightarrow$ month), **roll-up** (month $\rightarrow$ year), **pivot** (rotate axes). These give managers interactive exploration at "the speed of thought" (Codd's OLAP rules).

MOLAP (e.g. SSAS) stores cubes natively - fast but sparse-data hungry; **ROLAP** translates OLAP to SQL over star schemas - scales better; **HOLAP** hybrids. Modern in-memory columnar engines (Power BI VertiPaq, ClickHouse) blur the distinction.

- **Slice**: one value of a dimension; **Dice**: ranges on multiple dimensions.
- **Drill-down/roll-up**: move along dimension hierarchies.
- **Pre-aggregation** trades storage for query speed.

VERY IMPORTANT — Exam Pointer

Very important: define slice, dice, drill-down and roll-up with a sales-cube example.

Section Summary

OLAP cubes make multidimensional exploration interactive; operations navigate hierarchies and subcubes.

UNIT 2: DATA MINING

Lecture 4: Data Mining Process and Tasks

Learning Objectives

- Describe CRISP-DM

- List data mining tasks
- Explain preprocessing importance

Key Terms & Definitions

CRISP-DM

Cross-Industry Standard Process for Data Mining: business understanding -> deployment.

Classification / Clustering

Supervised prediction of labels vs unsupervised grouping.

Data preprocessing

Cleaning, integration, transformation, reduction - typically 70-80% of project effort.

CRISP-DM structures projects: (1) business understanding (define the decision), (2) data understanding, (3) data preparation, (4) modeling, (5) evaluation, (6) deployment - with loops back. Skipping business understanding is the top cause of failed analytics projects.

Task taxonomy: **classification** (churn yes/no), **regression** (forecast sales), **clustering** (customer segments), **association rules** (market basket), **anomaly detection** (fraud), **sequence/time-series** patterns. Supervised tasks need labeled data; unsupervised find structure.

Preprocessing decides success: handle missing values (impute vs delete - and know why missing), outliers (cap, transform), normalization (min-max, z-score for distance methods), encoding categoricals (one-hot, target encoding), and **feature engineering** (ratios, lags, RFM features) - the highest-leverage step.

- **CRISP-DM phases in order** - memorize the six and the iteration loops.
- **"Garbage in, garbage out"**: preprocessing is 70-80% of the work.
- **Feature engineering** beats fancier algorithms most of the time.
- **Train/validation/test split** (or cross-validation) prevents overfitting optimism.

VERY IMPORTANT — Exam Pointer

Exam pointer: list the six CRISP-DM phases and explain why business understanding comes first.

Section Summary

Data mining is a disciplined process; tasks divide into supervised/unsupervised, and preparation dominates effort.

Lecture 5: Classification Algorithms

Learning Objectives

- Explain decision trees
- Describe Naive Bayes and kNN
- Compare classifiers

Key Terms & Definitions

Decision tree

Recursive splits on features maximizing information gain / minimizing Gini.

Naive Bayes

Probabilistic classifier assuming feature independence given the class.

Overfitting

Memorizing training noise; controlled by pruning, regularization, cross-validation.

Decision trees split data recursively: at each node choose the feature/threshold maximizing **information gain** (entropy reduction) or minimizing **Gini impurity**. Trees are interpretable and handle mixed types, but single trees overfit - **pruning** (or ensembles) controls this.

Naive Bayes applies Bayes' theorem with the "naive" conditional-independence assumption: $P(\text{class}|\text{features})$ proportional to $P(\text{class}) \times \text{product of } P(\text{feature}|\text{class})$. Despite the unrealistic assumption it is fast, needs little data, and excels at text classification.

k-NN classifies by majority vote of the k nearest training points (distance metric matters - normalize first); lazy, no training, but slow at prediction. **Evaluation**: confusion matrix -> accuracy, precision, recall, F1, ROC-AUC; for imbalanced churn/fraud data, **precision-recall** and business cost matrices beat raw accuracy.

- **Entropy/Gini**: impurity measures driving tree splits.
- **Pruning + max depth + min samples**: anti-overfitting controls.
- **Naive Bayes**: strong baseline for text; handles high dimensions.
- **Always report precision/recall**, not just accuracy, on imbalanced data.

Example

Churn prediction: decision tree finds "contract=monthly AND tenure<6 months AND complaints>2" as the top churn segment - directly actionable for retention offers.

VERY IMPORTANT — Exam Pointer

Very important: define information gain and explain overfitting with two remedies.

Section Summary

Trees give interpretable rules, Naive Bayes gives fast probabilistic baselines; honest evaluation uses more than accuracy.

Lecture 6: Ensemble Methods and Advanced Models

Learning Objectives

- Explain bagging and boosting
- Describe random forests and gradient boosting
- Introduce SVMs and neural nets

Key Terms & Definitions

Bagging

Bootstrap aggregating: average many trees on resampled data (random forest).

Boosting

Sequentially fit weak learners to residuals (AdaBoost, gradient boosting, XGBoost).

SVM

Support Vector Machine: maximum-margin classifier, kernel trick for non-linearity.

Random forest bags hundreds of deep trees trained on bootstrap samples with random feature subsets; voting/averaging slashes variance while keeping low bias. **Out-of-bag (OOB)** samples give free validation. Feature importance scores guide insight.

Boosting builds trees sequentially, each correcting predecessors' errors: **AdaBoost** reweights misclassified points; **gradient boosting** (XGBoost, LightGBM, CatBoost) fits residuals with regularization - the dominant tabular-data winner in competitions and industry. Tune learning rate, depth, and use early stopping.

SVMs maximize the margin between classes; the **kernel trick** (RBF) implicitly maps to high dimensions. **Neural networks** (MLPs) learn hierarchical features - overkill for small tabular data, essential for text/image. Rule of thumb: start with gradient boosting for tables.

- **Bagging reduces variance; boosting reduces bias** (but can overfit - tune it).
- **XGBoost/LightGBM**: default choice for tabular classification/regression.
- **SVM + RBF kernel**: strong on medium-sized, high-dimensional data.
- **No free lunch**: validate with cross-validation; simplest adequate model wins.

VERY IMPORTANT — Exam Pointer

Exam pointer: contrast bagging vs boosting in one sentence each, and name when to prefer gradient boosting.

Section Summary

Ensembles dominate tabular analytics: bagging stabilizes, boosting sharpens; validate everything.

Lecture 7: Clustering and Association Rules

Learning Objectives

- Explain k-means
- Describe hierarchical clustering
- Mine association rules with Apriori

Key Terms & Definitions

k-means

Partition data into k clusters minimizing within-cluster variance; Lloyd's algorithm.

Support / Confidence / Lift

Association rule metrics: frequency, conditional probability, beyond-chance strength.

Apriori

Level-wise frequent itemset mining using the downward-closure property.

k-means: choose k , initialize centroids, assign points to nearest centroid, recompute centroids, repeat. Fast and simple; weaknesses: must pre-specify k (use elbow/silhouette), spherical-cluster assumption, outlier sensitivity. **k-means++** initialization helps. **Hierarchical clustering** (agglomerative dendrogram) needs no k upfront - cut the tree where convenient.

DBSCAN finds arbitrary-shaped clusters via density (core/border/noise points) - great for spatial/anomaly data, no k needed. **Choosing k** : elbow method, silhouette score, gap statistic - plus business interpretability.

Association rules (market basket): **support** $P(A \text{ and } B)$, **confidence** $P(B|A)$, **lift** = confidence / $P(B)$ (lift > 1 means real association). **Apriori** exploits downward closure (subsets of frequent itemsets are frequent) to prune the exponential search; FP-Growth is faster via tree compression.

- **Standardize features** before k-means (distance-based).
- **Elbow + silhouette** for choosing k ; validate clusters make business sense.
- **Lift > 1** is the interestingness test; high confidence alone misleads on frequent items.
- **Applications**: customer segmentation (RFM + k-means), cross-sell rules, fraud (DBSCAN outliers).

Example

{diapers} → {beer} (support 3%, confidence 60%, lift 2.1): place them adjacently; lift confirms it beats chance.

VERY IMPORTANT — Exam Pointer

Very important: define support, confidence and lift, and explain the Apriori pruning principle.

Section Summary

Clustering segments customers; association rules find cross-sell patterns; both need business validation.

UNIT 3: ADVANCED ANALYTICS

Lecture 8: Regression and Time Series Forecasting

Learning Objectives

- Build and interpret linear regression
- Explain ARIMA components
- Forecast with exponential smoothing

Key Terms & Definitions

Linear regression

Models y as a linear function of features; OLS minimizes squared errors.

ARIMA(p,d,q)

AutoRegressive Integrated Moving Average: the classical forecasting model.

Stationarity

Constant mean/variance over time - required by ARIMA (achieved by differencing).

Multiple linear regression: $y = b_0 + b_1x_1 + \dots + e$; OLS estimates minimize sum of squared residuals. Interpret coefficients *ceteris paribus*; check **R-squared**, residual plots (linearity, homoscedasticity, normality), and **multicollinearity** (VIF). **Regularization** (Ridge/Lasso) shrinks coefficients; Lasso performs feature selection.

Time series components: trend, seasonality, cycles, noise. **Decomposition** separates them; **exponential smoothing** (Holt-Winters) handles level/trend/seasonality with smoothing parameters. **ARIMA(p,d,q)**: AR(p) regresses on past values, I(d) differences to stationarity, MA(q) models past errors - Box-Jenkins methodology: identify (ACF/PACF), estimate, diagnose.

Modern practice: **Prophet** (additive trend + seasonality + holidays, robust to missing data) and gradient boosting on lag features often beat ARIMA in business forecasting; always hold out a recent period for honest evaluation (**MAPE**, **RMSE**).

- **Regression assumptions:** linearity, independent homoscedastic normal errors - check residuals.
- **Differencing** makes series stationary for ARIMA.
- **Seasonal naive** is the baseline every forecaster must beat.
- **Forecast intervals**, not just point forecasts, for inventory decisions.

VERY IMPORTANT — Exam Pointer

Exam pointer: expand ARIMA(p,d,q) - what each letter means - and state why stationarity is required.

Section Summary

Regression explains drivers; ARIMA/Prophet forecast the future; honest backtesting separates signal from luck.

Lecture 9: Text Mining and Sentiment Analysis

Learning Objectives

- Describe the text mining pipeline
- Explain TF-IDF
- Perform sentiment analysis

Key Terms & Definitions

TF-IDF

Term Frequency-Inverse Document Frequency: weights terms by distinctiveness.

Sentiment analysis

Classifying text polarity (positive/negative/neutral) or emotion.

Topic modeling (LDA)

Discovers latent topics as word distributions in a document collection.

Pipeline: collect (reviews, tweets, support tickets) -> clean (lowercase, remove noise) -> tokenize -> remove stopwords -> stem/lemmatize -> vectorize. **Bag-of-words** counts terms; **TF-IDF** downweights ubiquitous words ("the", "product") and upweights distinctive ones - the classic retrieval weighting.

Sentiment analysis: lexicon methods (VADER for social text) vs ML classifiers (Naive Bayes/logistic on TF-IDF) vs transformers (BERT fine-tunes) for nuance, sarcasm, aspect-level sentiment ("battery great, camera poor"). Business use: brand monitoring, review triage, NPS drivers.

Topic modeling (LDA): each document is a mixture of topics, each topic a word distribution - discovers "what customers talk about" unsupervised. **Word embeddings** (Word2Vec) capture semantics for search and recommendation.

- **TF-IDF** = $tf \times \log(N/df)$: memorize the intuition (frequent here, rare elsewhere).
- **Aspect-based sentiment** beats document-level for product feedback.
- **LDA topics** need human labeling - coherence scores help choose topic count.

Example

Telecom reviews: TF-IDF + logistic regression flags "signal drop" complaints; LDA discovers an emerging "billing app crash" topic before support tickets spike.

VERY IMPORTANT — Exam Pointer

Very important: explain TF-IDF components and one advantage of aspect-based over document-level sentiment.

Section Summary

Text mining turns unstructured feedback into measurable insight: what is said, how it feels, and about what.

Lecture 10: Data Visualization and Dashboards

Learning Objectives

- Choose the right chart
- Apply visualization best practices
- Design effective dashboards

Key Terms & Definitions

Pre-attentive attributes

Color, size, position processed instantly by vision - use for emphasis.

Dashboard

Single-screen operational view of KPIs with drill-down.

Lie factor

Tufte's ratio of visual effect to data effect - keep near 1.

Chart selection: comparison -> bar charts; trend -> line; composition -> stacked bar (avoid pie beyond 2-3 slices); distribution -> histogram/boxplot; relationship -> scatter; geography -> choropleth. **Dual axes** mislead - prefer indexed lines or small multiples.

Best practices (Tufte, Few): maximize **data-ink ratio**, start bars at zero, label directly instead of legends where possible, use color for meaning (sequential/diverging palettes, colorblind-safe), declutter gridlines. **Pre-attentive** highlights (one red bar) guide the eye to the insight.

Dashboard design: one screen, top-left = most important KPI, consistent filters, drill-down paths, refresh cadence labeled, alerts on thresholds. Distinguish **operational** (real-time monitoring), **analytical** (exploration), and **strategic** (executive scorecards) dashboards.

- **Bar for comparison, line for trend** - the two rules that prevent most bad charts.
- **Start bars at zero**; truncated axes exaggerate.
- **One dashboard, one decision**: know its user and question.
- **Annotate the insight**, not just the data ("Sales fell 12% after price hike").

VERY IMPORTANT — Exam Pointer

Exam pointer: given a scenario, choose and justify the right chart type.

Section Summary

Visualization is decision design: right chart, honest scales, and dashboards built around one question.

UNIT 4: DECISION SUPPORT AND STRATEGY

Lecture 11: Decision Support Systems

Learning Objectives

- Classify DSS types
- Explain what-if and goal-seeking analysis
- Describe optimization in DSS

Key Terms & Definitions

What-if analysis

Varying inputs to see outcome changes (scenario/sensitivity analysis).

Goal seeking

Working backward: find inputs achieving a target outcome.

Linear programming

Optimizing a linear objective under linear constraints (simplex).

DSS types: model-driven (spreadsheets, optimization), data-driven (warehouses/OLAP), document-driven (search), knowledge-driven (expert systems), communication-driven (collaboration). A modern BI platform blends all five.

What-if/sensitivity analysis: data tables varying price and volume to map profit; **scenario manager** (base/best/worst); **goal seek** (what sales achieve Rs 10m profit?); **Monte Carlo simulation** propagates input uncertainty into outcome distributions - essential for risk-aware planning.

Optimization: linear programming allocates limited resources (production mix, media budget) via the simplex method; **decision trees** (different sense: rollback analysis with EMV) structure sequential decisions under uncertainty. DSS embeds these so managers decide with numbers, not gut.

- **Sensitivity analysis** finds which assumptions actually matter.
- **Monte Carlo** turns point estimates into risk profiles.
- **EMV (expected monetary value)** rolls back decision trees.
- **DSS success** = used in real decisions, not technical sophistication.

Example

A bank's credit DSS: scorecard (model) + bureau data (data-driven) + policy rules (knowledge-driven) auto-decides small loans, routing edge cases to officers.

VERY IMPORTANT — Exam Pointer

Very important: distinguish what-if, goal-seeking and optimization with one example each.

Section Summary

DSS operationalizes analytics inside decisions through scenarios, simulation and optimization.

Lecture 12: Big Data Technologies

Learning Objectives

- Define the 5 Vs of big data
- Explain MapReduce and Spark
- Describe NoSQL categories

Key Terms & Definitions

5 Vs

Volume, Velocity, Variety, Veracity, Value - the big data dimensions.

MapReduce

Distributed programming model: map (transform) then reduce (aggregate).

Spark

In-memory distributed computing; 10-100x faster than Hadoop MapReduce for iterative jobs.

When data exceeds single-machine capacity, **distributed systems** take over. **HDFS** stores blocks replicated across nodes; **MapReduce** ships computation to data (map -> shuffle -> reduce). **YARN** manages cluster resources.

Spark keeps data in memory (**RDDs/DataFrames**), making iterative ML 10-100x faster than MapReduce; Spark SQL, MLlib, and Structured Streaming unify batch, ML and streaming. The **Lambda architecture** (batch + speed layers) is giving way to **Kappa** (streaming-only) and lakehouse designs.

NoSQL: document (MongoDB), key-value (Redis), column-family (Cassandra/HBase), graph (Neo4j) - chosen by access pattern, sacrificing joins/ACID for scale. **Stream processing** (Kafka + Flink/Spark Streaming) powers fraud detection and recommendations in real time.

- **Use big data tech only when data truly doesn't fit** - a laptop with DuckDB/pandas beats a cluster for GBs.
- **Spark DataFrames** > RDDs for optimization (Catalyst).
- **CAP theorem**: consistency/availability/partition-tolerance trade-offs in NoSQL.
- **Kafka**: the standard event-streaming backbone.

VERY IMPORTANT — Exam Pointer

Exam pointer: list the 5 Vs and explain one scenario where Spark beats MapReduce.

Section Summary

Distributed computing scales analytics horizontally; choose the tool by data size and latency needs.

Lecture 13: BI Strategy, Governance and Ethics

Learning Objectives

- Align BI with business strategy
- Explain data governance
- Discuss analytics ethics

Key Terms & Definitions

Data governance

Policies for data quality, ownership, security and lifecycle.

Single version of truth

One governed definition per KPI across the enterprise.

Algorithmic bias

Models reproducing historical discrimination (hiring, lending).

BI strategy starts from decisions, not data: map strategic objectives to key decisions to required analytics (the **balanced scorecard** links financial, customer, process, learning perspectives to KPIs). Maturity models (Gartner) stage the journey from ad-hoc spreadsheets to embedded AI.

Data governance: data owners/stewards, **data catalogs**, quality SLAs, master data management (one customer record), lineage tracking, and access control. Without governance, self-service BI produces **multiple versions of truth** - five dashboards, five "revenues".

Ethics: biased training data -> biased lending/hiring models (test disparate impact); **privacy** (PDPA/GDPR: consent, minimization, right to erasure); **explainability** rights; surveillance concerns in people analytics. Responsible BI includes bias audits and human-in-the-loop for high-stakes decisions.

- **Start from decisions**, then data - not the reverse.
- **One KPI, one definition:** governed metric layer.
- **Bias audit:** test models across demographic slices before deployment.
- **Privacy by design:** minimize collection, anonymize, secure.

Example

A retailer's five "monthly sales" figures traced to different returns-handling definitions; a governed metric layer with one certified definition ended the debate.

VERY IMPORTANT — Exam Pointer

Very important: explain "single version of truth" and name two ethical risks of predictive analytics.

Section Summary

BI succeeds through strategy alignment, governance discipline, and ethical guardrails - technology is the easy part.

Glossary of Key Terms

BI — Business Intelligence: data-to-decision technologies and practices.

ETL — Extract, Transform, Load pipeline into the warehouse.

Star schema — Fact table plus denormalized dimensions for analytics.

OLAP — Online Analytical Processing over multidimensional cubes.

CRISP-DM — Standard six-phase data mining process.

TF-IDF — Term weighting by local frequency and global rarity.

Lift — Association strength beyond chance; >1 is interesting.

ARIMA — Autoregressive Integrated Moving Average forecasting model.