Machine Learning Course Outline (Detailed Version)

Week 00: Statistics and Mathematics for ML

Module 01: Descriptive Statistics and Distributions

- Mean, median, mode: when each is appropriate
- Variance and standard deviation as measures of spread
- Percentiles, quartiles, IQR, and z-score
- Distribution shapes: symmetric vs skewed, long tails; outlier detection with IQR fences
- When median and IQR are preferred over mean and standard deviation

Module 02: Probability Basics for ML

- Events, sample space, and basic probability rules
- Conditional probability and independence
- Bayes' theorem (intuitive, formula-level) and base-rate effects
- Sensitivity, specificity, false positive/negative rates, PPV, and NPV
- Class imbalance and its impact on interpreting errors

Module 2.5: Worked examples on center, spread, IQR fences, and z-scores

Module 03: Data Quality, Scaling, and Encoding

- Missingness types (MCAR, MAR, MNAR) and simple imputation rationale
- Standardization, min–max scaling, and robust scaling (formulas and when to use)
- Nominal vs ordinal variables, one-hot vs ordinal encoding, and geometric implications
- Vectors, dot product, norms, Euclidean and Manhattan distance, and cosine similarity
- Covariance and correlation; PCA idea as directions of maximum variance (conceptual only)

Module 3.5: Bayes and confusion-rate worksheets (no coding)

Module 4: Week 00 concept quiz (pen-and-paper calculations and short explanations)

Week 01 – Getting Started with Machine Learning

Module 01: Introduction to Machine Learning

- What is ML? Types of ML (Supervised, Unsupervised, Reinforcement)
- Key concepts: features, labels, training, testing, generalization
- The ML pipeline: data collection → preprocessing → training → evaluation

- Overview of common applications and challenges (bias, overfitting, underfitting)

Module 02: Data Preprocessing and Feature Engineering

- Handling missing values, encoding categorical data, and normalization/scaling
- Outlier detection and handling
- Feature transformation (polynomial, binning)
- Feature construction and domain-driven feature creation

Module 2.5:
Colab: Load 3 tiny CSVs, identify feature and target columns, classify task type, and sketch the pipeline in the notebook with a simple diagram cell.

Module 03: Exploratory Data Analysis (EDA)

- Understanding data distributions, correlations, and patterns
- Visualizing data (histograms, scatter plots, heatmaps, pairplots)
- Identifying data imbalance and relationships between variables
- EDA best practices before model building

Module 3.5:
Colab: Run a non-destructive audit on a candidate dataset using pandas only to profile missingness and categorical levels; draft a preprocessing checklist as markdown in the notebook.

Module 04:
Week 01 quiz with classify tasks from mini-scenarios, short EDA interpretation tasks, and a mini CSV to reason about.

Project Tasks:

- Brainstorm 2–3 project ideas and write one-paragraph problem statements.
- Scout datasets for each idea; check license, size, features, target, imbalance, and quality.
- Define success metrics (e.g., MAE, F1, ROC-AUC) and why they fit.
- Start a literature matrix and collect at least 8 closely related papers.
- Create an EDA and a preprocessing checklist that you will execute later.
- Set up repository structure.

Weekly Deliverables:

- Topic shortlist document + dataset shortlist with links.
- Dataset card v1 for the leading dataset.
- Metrics plan table.
- Literature matrix (≥8 entries).
- Repo scaffold with README.

Week 02 – Regression Fundamentals

Module 01: Linear Regression

- Concept of regression and line fitting
- Cost function, gradient descent, and optimization
- Model evaluation metrics (R^2, MAE, RMSE)
- Assumptions and limitations of linear regression

Module 02: Logistic Regression

- Transition from regression to classification
- Sigmoid function and decision boundary
- Model evaluation (accuracy, precision, recall, F1-score)
- ROC curve, AUC, and confusion matrix

Module 2.5:
Colab: Given y_true/y_pred arrays, compute R^2/MAE/RMSE; inspect residual patterns; note assumption violations.

Module 03: Decision Trees

- Structure and working of decision trees
- Entropy, Gini index, and information gain
- Pruning and avoiding overfitting
- Interpreting decision tree visualizations

Module 3.5:
Colab: Vary classification thresholds to see precision/recall trade-offs; plot a provided ROC from probabilities and compute AUC.

Module 04:
Quiz: Metric selection and thresholding on small tables; manual info gain split on a toy example.

Project Tasks:

- Lock the final topic and single dataset after the feasibility check.
- Complete literature review (15–20 focused citations grouped by themes).
- Finalize evaluation protocol: split or CV, seeds, primary and secondary metrics, baseline list.
- Write a short risk and ethics note.

Weekly Deliverables:

- Final topic statement and locked dataset with updated dataset card.
- Literature review document.
- Evaluation protocol document (including baselines).
- Ethics and risks note.

Week 03 – Advanced Supervised Learning

Module 01: Support Vector Machines (SVM)

- Concept of hyperplanes and margins
- Kernel trick (linear, polynomial, RBF)
- Regularization and soft margin classification
- Visualization of decision boundaries

Module 02: Naïve Bayes Classifier

- Bayes' theorem and the independence assumption
- Types: Gaussian, Multinomial, Bernoulli NB
- Pros, cons, and use cases (text classification, spam filtering)

Module 2.5:
Colab: Train SVM (linear vs RBF) on toy 2D data; visualize boundaries; show effect of scaling.

Module 03: Ensemble Learning

- What and why of ensemble methods
- Bagging vs. Boosting vs. Stacking
- Random Forests (Bagging-based) and voting classifiers overview

Module 3.5:
Colab: Tiny text classification (10–20 sentences) with bag-of-words + Multinomial NB; inspect top tokens.

Module 04:
Quiz: SVM margin/kernel intuition, NB independence caveat, ensemble bias–variance trade-offs.

Project Tasks:

- Implement preprocessing: imputation, encoding, scaling; prevent leakage (fit on train only).
- Engineer 1–3 justified features if appropriate.

- Execute full EDA with 6–8 plots and short interpretations.
- Save train/validation/test splits with fixed seeds and a split manifest.

Weekly Deliverables:

- Preprocessing notebook or script plus saved transformers/artifacts.
- EDA report notebook with plots and a one-page narrative.
- Updated dataset card and ethics note.
- Split the manifest file.

Week 04 – Ensemble and Dimensionality Reduction

Module 01: Boosting Techniques

- AdaBoost, Gradient Boosting, and XGBoost concepts
- Comparison with bagging
- Hyperparameter tuning and feature importance

Module 02: K-Nearest Neighbors (KNN)

- Distance metrics and k-value selection
- Lazy learning and computational cost
- Strengths, weaknesses, and applications

Module 2.5:
Colab: Run a tiny tuning grid on a boosting model (3–5 configs); plot train vs validation to discuss overfitting.

Module 03: Dimensionality Reduction

- Curse of dimensionality and need for feature reduction
- PCA (Principal Component Analysis) and visualization
- LDA (Linear Discriminant Analysis)
- t-SNE and UMAP (conceptual overview)

Module 3.5:
Colab: KNN across k values with/without scaling; compare Euclidean vs cosine on same features; optional PCA 2D scatter.

Module 04:
Quiz: Boosting vs bagging scenarios, KNN distance effects, PCA variance interpretation.

Project Tasks:

- Select at least one baseline and one stronger model that are suitable for the data.
- Implement training loop using the fixed evaluation protocol.
- Try a small hyperparameter grid; if useful, apply PCA or simple feature selection.
- Produce an initial results table and key diagnostic plots.

Weekly Deliverables:

- Results table comparing at least two models.
- Plots: confusion matrix or residuals, ROC/PR where relevant; optional 2D DR scatter.
- Clean run instructions are included in the README, along with a brief progress memo.

Week 05 – Project Week (Step-by-Step Build)

Module 01: Project Setup and Repository Scaffold

- Lock topic, final dataset, and licenses; finalize dataset card
- Create repo structure (data/, notebooks/, src/, reports/, figures/) and gitignore
- Fix random seeds; define and save train/validation/test splits
- Add requirements.txt (or environment.yml) and a short README with run instructions

Module 02: Preprocessing Pipeline (Final)

- Implement the final imputation, encoding, and scaling choices
- Fit transformers on train only; apply to validation/test (no leakage)
- Save fitted preprocessors and the transformed train/val/test artifacts
- Quick sanity checks: class balance, feature ranges, and drift between splits

Module 03: Baseline Implementation and Logging

- Implement a clear training script/notebook for the baseline model
- Record metrics with a fixed evaluation protocol (hold-out or CV)
- Export key plots (confusion/residuals, ROC/PR where relevant)
- Log all settings (seed, split, hyperparameters) and save a results table

Module 04: Model Improvement and Tuning

- Select 1–2 stronger models based on Weeks 02–04 learnings
- Run structured hyperparameter sweeps; address imbalance (class weights/sampling) if needed
- Optional: feature selection or PCA, and report the impact on metrics
- Compare against baseline with clean tables and learning/validation curves

Module 05: Final Evaluation, Interpretation, and Paper Scaffold

- Evaluate the chosen model on the locked test set; freeze results
- Produce final figures and tables; add error analysis and slice-wise metrics
- Add simple interpretability (feature importance or brief explanation)
- Draft **Methods** and **Results** sections in the conference template with numbered figures/tables and cross-references

Module 06:

Milestone replication check: run end-to-end from a clean clone and match a target metric band.

Week 06 – Report Writing and Submission

Module 01: Paper Writing and Polish

- Write Abstract, Introduction, and finalize Related Work
- Refine Methods and Results based on Week 05 outputs
- Add Discussion, Conclusion, and explicit Limitations
- Ensure flow, clarity, and consistency of terminology and notation

Module 02: Ethics, Reproducibility, and References

- Add data license, bias, and privacy considerations; note any IRB or ethics aspects
- Complete reproducibility checklist and code/data availability statements
- Fix citations and references to the target format (IEEE or ACM)
- Verify figure and table numbering, captions, and in-text references; run plagiarism and template checks

Module 2.5: Live session on addressing queries and issues of Module 01 & Module 02

Module 03: Presentation and Submission Package

- Prepare a 5-minute slide deck with speaker notes; optional poster overview
- Build the final PDF and source bundle of the paper
- Create a final tagged GitHub release with instructions and artifacts
- Submit the three deliverables: dataset package or documented link, implementation repo, conference-format report PDF (plus slides)

Module 3.5: Live session on addressing queries and issues of Module 03

Module 04:

Final submission plus a short viva-style defense (slides + Q&A).
[Need a team for this exam module]