**📌 Semester Plan (University AI + Self ML Projects)**

**🔹 Weeks 1–4 → Foundation**

**University (AIMA basics):**

* Intro to AI (what AI is, history, applications).
* Problem-solving & search algorithms (DFS, BFS, A\*).

**Self-Learning (ML):**

* Learn Regression (Linear, Multiple, Polynomial).
* Learn Regression metrics (MSE, RMSE, R2R^2R2).
* First ML project → **House Price Prediction**.

**🔹 Weeks 5–8 → Logic & Probability**

**University (AIMA basics):**

* Logic-based AI (propositional logic, first-order logic).
* Reasoning under uncertainty (Bayes’ theorem, probability).

**Self-Learning (ML):**

* Learn Classification (Logistic Regression, Decision Trees, Random Forest).
* Learn Classification metrics (Accuracy, Precision, Recall, F1, ROC).
* Project → **Titanic Survival Prediction**.
* Bonus mini-project → **Iris Flower Classification**.

**🔹 Weeks 9–12 → Advanced AI Concepts**

**University (AIMA basics):**

* Knowledge representation.
* Planning & decision-making.

**Self-Learning (ML):**

* Learn Clustering (K-Means, DBSCAN).
* Learn Dimensionality Reduction (PCA).
* Project → **Customer Segmentation (Clustering)**.
* Start learning **Feature Engineering & Scaling** (StandardScaler, One-Hot Encoding).

**🔹 Weeks 13–15 → Bringing It Together**

**University (AIMA basics):**

* Intro to Machine Learning (overview, not too deep).
* Wrap-up: AI ethics, real-world applications.

**Self-Learning (ML):**

* Learn Ensemble Methods (Bagging, Boosting → XGBoost).
* Do **End-to-End ML Pipeline Project**:
  + Data Cleaning → EDA → Preprocessing → Model → Evaluation.
  + Example: **Loan Approval Prediction**.

**🔹 Week 16 → Final Review**

**University:** Revise AIMA concepts for exams (summaries, definitions, examples).  
**Self-Learning:** Polish your ML projects, upload to **GitHub/Kaggle**.

**📌 What You’ll Have at End of Semester**

✅ University side → Strong **AI theory** (AIMA basics).  
✅ Self-study side → 4–5 **ML projects**:

1. House Price Prediction (Regression).
2. Titanic Survival Prediction (Classification).
3. Iris Classification (Classification).
4. Customer Segmentation (Clustering).
5. Loan Approval (End-to-End Pipeline + Ensemble).

👉 That’s enough to show **real ML skills + academic AI knowledge** on your CV.

**📌 AI/ML Roadmap (Comprehensive – with all small things included)**

**1. Foundations (Before ML)**

These are the basics you need to understand AI/ML properly:

* **Math for ML:**
  + Linear Algebra → vectors, matrices, dot product, matrix multiplication.
  + Probability & Statistics → distributions, expectation, variance, conditional probability, Bayes theorem.
  + Calculus (basics) → derivatives, gradients, chain rule, partial derivatives.
  + Optimization basics → what is "minimizing a function", convex vs non-convex.
* **Programming:**
  + Python (must) → lists, loops, functions, classes (OOP).
  + Numpy, Pandas (data handling).
  + Matplotlib/Seaborn (visualization).
* **Data Concepts:**
  + Structured vs unstructured data.
  + Data cleaning (handling missing values, duplicates, outliers).
  + Feature scaling (normalization, standardization).
  + Train/test split & cross-validation.

**2. Core Machine Learning**

* **Types of Learning:**
  + Supervised learning.
  + Unsupervised learning.
  + Reinforcement learning (just basics at this stage).
* **Key Algorithms:**
  + Linear Regression.
  + Logistic Regression.
  + k-Nearest Neighbors (kNN).
  + Decision Trees.
  + Random Forest.
  + Naive Bayes.
  + Support Vector Machines (SVM).
  + k-Means Clustering.
  + PCA (Dimensionality reduction).
* **Important Concepts:**
  + Loss function (MSE, Cross-Entropy).
  + Cost function.
  + Bias-variance tradeoff.
  + Underfitting vs Overfitting.
  + Regularization (L1, L2).
  + Gradient Descent (and variants: SGD, Mini-batch).
  + Evaluation metrics:
    - Regression: RMSE, MAE, R².
    - Classification: Accuracy, Precision, Recall, F1-score, ROC-AUC.
  + Confusion Matrix.

**3. Advanced ML Topics**

* Feature Engineering (encoding categorical data, polynomial features).
* Feature Selection (filter, wrapper, embedded methods).
* Hyperparameter Tuning (GridSearchCV, RandomizedSearchCV).
* Ensemble Methods:
  + Bagging.
  + Boosting (AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost).
* Imbalanced Data handling (SMOTE, class weights).
* Model Deployment basics (Flask/FastAPI, Streamlit).
* Pipelines (Scikit-learn Pipelines).

**4. Deep Learning (Neural Networks)**

* **Core Concepts:**
  + Perceptron, Activation functions (ReLU, Sigmoid, Tanh, Softmax).
  + Forward propagation.
  + Backpropagation.
  + Loss functions (Cross-entropy, MSE).
  + Optimizers (SGD, Adam, RMSprop).
* **Neural Network Architectures:**
  + Feedforward Neural Network (MLP).
  + Convolutional Neural Networks (CNN).
  + Recurrent Neural Networks (RNN).
  + LSTMs & GRUs.
  + Transformers (basic intro).
* **Important Deep Learning Concepts:**
  + Overfitting in Neural Nets → Dropout, Early Stopping, Batch Normalization.
  + Weight initialization.
  + Vanishing/Exploding gradients.
* **Libraries/Frameworks:**
  + TensorFlow / Keras.
  + PyTorch (highly recommended for modern AI research).

**5. AI Specializations (Choose Your Path)**

Once comfortable with ML + DL, pick a specialization:

* **Computer Vision** → CNNs, Object Detection (YOLO, Faster R-CNN), Image Segmentation, OpenCV.
* **Natural Language Processing (NLP)** → Text preprocessing, Word embeddings (Word2Vec, GloVe, BERT), Transformers, Chatbots.
* **Reinforcement Learning** → Markov Decision Process (MDP), Q-Learning, Deep Q Networks (DQN).
* **Generative AI** → GANs, Diffusion Models, Large Language Models (LLMs).
* **MLOps (Deployment & Scaling)** → Docker, Kubernetes, Model monitoring, CI/CD for ML.

**6. Tools & Extras (Often Ignored but Crucial)**

* **Version Control** → Git/GitHub.
* **SQL** → Querying structured data.
* **Cloud** → Google Colab, AWS, GCP, Azure.
* **Experiment Tracking** → MLflow, Weights & Biases.
* **Data Handling** → Big Data basics (Spark, Hadoop – optional for now).
* **Soft Skills** → Report writing, Presenting results, Communication with non-technical people.

**7. Practice & Projects**

* Start with small datasets: Titanic, Iris, MNIST.
* Do **Kaggle competitions** (learn from notebooks).
* Projects ideas:
  + Spam email classifier.
  + House price prediction.
  + Movie recommendation system.
  + Sentiment analysis of tweets.
  + Image classification (cats vs dogs).
* Later, do a **bigger project**: end-to-end pipeline → Data collection → Cleaning → Model → Deployment.