1. OOP concept of Python
2. Numpy
3. Pandas
4. Data Preprocessing
5. Simple Linear Regression (Supervised Learning)
6. Multiple Linear Regression (Supervised Learning)
7. Evaluation metrics (Regression)
8. Implement Logistic Regression in Python. Concepts and implementation.
9. Implement k-NN in Python. Concepts and implementation.
10. Learn about evaluation metrics: Accuracy, Precision, Recall, F1 Score.
11. Project (Implement Linear regression and Logistic regression on the dataset)

**Week 3: Advanced Supervised Learning**

**Day 15-17: Decision Trees and Random Forests**

* **Day 15:**
  + Study Decision Tree Algorithm: Concepts and implementation.
  + Implement Decision Trees in Python.
* **Day 16:**
  + Study Ensemble Methods: Bagging, Random Forests.
  + Implement Random Forests in Python.
* **Day 17:**
  + Compare Decision Trees and Random Forests.
  + Evaluate their performance on a sample dataset.

**Day 18-20: Support Vector Machines (SVM)**

* **Day 18:**
  + Study Support Vector Machines (SVM): Concepts and implementation.
  + Implement SVM in Python.
* **Day 19:**
  + Learn about the Kernel Trick and its applications.
  + Implement SVM with different kernels.
* **Day 20:**
  + Evaluate the performance of SVM on a sample dataset.
  + Compare SVM with other classification algorithms.

**Day 21: Project**

* **Day 21:**
  + Choose a dataset (e.g., Titanic survival prediction).
  + Implement Decision Trees and SVM on the dataset.
  + Document the process and results.

**Week 4: Unsupervised Learning**

**Day 22-24: Clustering**

* **Day 22:**
  + Study k-Means Clustering: Concepts and implementation.
  + Implement k-Means Clustering in Python.
* **Day 23:**
  + Study Hierarchical Clustering: Concepts and implementation.
  + Implement Hierarchical Clustering in Python.
* **Day 24:**
  + Evaluate clustering results using different metrics.
  + Compare k-Means and Hierarchical Clustering.

**Day 25-27: Dimensionality Reduction**

* **Day 25:**
  + Study Principal Component Analysis (PCA): Concepts and implementation.
  + Implement PCA in Python.
* **Day 26:**
  + Study t-SNE: Concepts and implementation.
  + Implement t-SNE in Python.
* **Day 27:**
  + Compare PCA and t-SNE for dimensionality reduction.
  + Visualize high-dimensional data using PCA and t-SNE.

**Day 28: Project**

* **Day 28:**
  + Choose a dataset (e.g., customer segmentation).
  + Implement Clustering and PCA on the dataset.
  + Document the process and results.

**Week 5: Neural Networks and Deep Learning**

**Day 29-31: Introduction to Neural Networks**

* **Day 29:**
  + Study Perceptron and Activation Functions.
  + Implement a simple Perceptron in Python.
* **Day 30:**
  + Learn about Forward and Backward Propagation.
  + Implement a basic neural network from scratch.
* **Day 31:**
  + Study different activation functions (ReLU, Sigmoid, Tanh).
  + Implement and compare their performance.

**Day 32-34: Deep Learning with TensorFlow/Keras**

* **Day 32:**
  + Introduction to TensorFlow and Keras.
  + Build a simple neural network using Keras.
* **Day 33:**
  + Learn about model training and evaluation in Keras.
  + Train and evaluate the neural network on a sample dataset.
* **Day 34:**
  + Experiment with different network architectures.
  + Fine-tune hyperparameters for better performance.

**Day 35-37: Convolutional Neural Networks (CNNs)**

* **Day 35:**
  + Study Convolutional Neural Networks (CNNs): Concepts and layers.
  + Implement a basic CNN in Keras.
* **Day 36:**
  + Learn about common CNN architectures (e.g., LeNet, AlexNet).
  + Implement and train a CNN on an image classification task.
* **Day 37:**
  + Experiment with data augmentation techniques.
  + Evaluate the CNN model on a validation dataset.

**Day 38: Project**

* **Day 38:**
  + Choose a dataset (e.g., MNIST dataset).
  + Build and train a CNN model on the dataset.
  + Document the process and results.

**Week 6: Advanced Deep Learning**

**Day 39-41: Recurrent Neural Networks (RNNs)**

* **Day 39:**
  + Study Recurrent Neural Networks (RNNs): Concepts and architecture.
  + Implement a simple RNN in Keras.
* **Day 40:**
  + Learn about Long Short-Term Memory (LSTM) Networks.
  + Implement an LSTM network for sequence prediction.
* **Day 41:**
  + Compare RNN and LSTM performance.
  + Evaluate the model on a sample sequence dataset.

**Day 42-44: Natural Language Processing (NLP)**

* **Day 42:**
  + Introduction to NLP and text preprocessing.
  + Implement text cleaning and tokenization.
* **Day 43:**
  + Study Sentiment Analysis: Concepts and implementation.
  + Implement sentiment analysis on a text dataset.
* **Day 44:**
  + Learn about text classification techniques.
  + Implement a text classification model.

**Day 45-47: Generative Models**

* **Day 45:**
  + Study Autoencoders: Concepts and implementation.
  + Implement an Autoencoder in Keras.
* **Day 46:**
  + Learn about Generative Adversarial Networks (GANs).
  + Implement a basic GAN in Keras.
* **Day 47:**
  + Experiment with generating data using GANs.
  + Evaluate the quality of generated data.

**Day 48: Project**

* **Day 48:**
  + Choose a sequence prediction task (e.g., stock price prediction).
  + Implement an RNN or LSTM for the task.
  + Document the process and results.

**Week 7: Reinforcement Learning**

**Day 49-51: Introduction to Reinforcement Learning**

* **Day 49:**
  + Study key concepts: Agent, Environment, State, Action, Reward.
  + Learn about Markov Decision Process (MDP).
* **Day 50:**
  + Study basic Reinforcement Learning algorithms.
  + Implement a simple RL algorithm.
* **Day 51:**
  + Learn about Q-Learning: Concepts and implementation.
  + Implement Q-Learning on a simple environment.

**Day 52-54: Deep Q-Networks (DQN)**

* **Day 52:**
  + Study Deep Q-Networks (DQN): Concepts and architecture.
  + Implement a basic DQN in Python.
* **Day 53:**
  + Train and evaluate the DQN model on a sample environment.
  + Experiment with hyperparameter tuning.
* **Day 54:**
  + Compare DQN with traditional Q-Learning.
  + Evaluate the performance improvements.

**Day 55: Project**

* **Day 55:**
  + Choose a simple reinforcement learning task (e.g., CartPole).
  + Implement and train a DQN model on the task.
  + Document the process and results.

**Week 8: Capstone Projects and Advanced Topics**

**Day 56-57: Capstone Project Planning**

* **Day 56:**
  + Define the problem statement for the capstone project.
  + Collect and preprocess data relevant to the project.
* **Day 57:**
  + Explore and analyze the dataset.
  + Plan the model architecture and approach.

**Day 58-60: Capstone Project Execution**

* **Day 58:**
  + Build and train the model.
  + Evaluate the model performance.
* **Day 59:**
  + Fine-tune the model and perform additional experiments.
  + Document the entire process.
* **Day 60:**
  + Prepare a presentation of the project results.
  + Present the findings and insights.