

180-Day Data Science Journey

(Beginner to Expert)

Welcome to your 6-month roadmap to become a Data Scientist. You're not just learning you're leveling up. Let's break it down week by week.

Phase 1: Python (Week 1–4)

“Lay your foundation — this is your weapon for everything ahead.”

Week 1: Basics – Variables, Data Types, Input/Output, Loops, Conditionals

- Q1. Write a Python program to swap two variables.
- Q2. Take user input and display it back to the user.
- Q3. Write a program to check if a number is even or odd.
- Q4. Create a program that prints the multiplication table of a given number.
- Q5. Write a program to find the largest of three numbers.
- Q6. Convert temperature from Celsius to Fahrenheit.
- Q7. Write a program to calculate the factorial of a number using a loop.
- Q8. Create a program to count the number of vowels in a string.
- Q9. Write a Python script to reverse a given string.
- Q10. Check if a number is a palindrome.
- Q11. Write a program to find the sum of first N natural numbers.
- Q12. Create a number guessing game.
- Q13. Write a program to print all prime numbers between 1 and 100.
- Q14. Check if a given year is a leap year or not.
- Q15. Create a program to print the Fibonacci series up to N terms.
- Q16. Write a program to find the GCD of two numbers.
- Q17. Write a program to find the LCM of two numbers.
- Q18. Check whether a character is a vowel or consonant.
- Q19. Write a program to calculate the sum of digits of a number.

- Q20. Create a program to find the second largest number in a list.
- Q21. Write a program to count the number of digits in an integer.
- Q22. Create a program to print all Armstrong numbers between 1 to 1000.
- Q23. Write a Python program to print a pattern of stars in a triangle.
- Q24. Create a calculator app using if-else.
- Q25. Write a program to display the ASCII value of a character.
- Q26. Convert a decimal number to binary using loops.
- Q27. Create a program to find the square root of a number.
- Q28. Write a program to find the sum of all even numbers in a list.
- Q29. Create a program to check whether a number is prime or not.
- Q30. Write a program to display the cube of the number up to an integer.

Week 2: Functions, Lists, Tuples, Dictionaries, Sets

- 1. Write a function to check if a number is even.
- 2. Create a list and find the sum of all its elements.
- 3. Write a program to find the maximum and minimum in a list.
- 4. Create a program that removes duplicates from a list.
- 5. Write a function to reverse a list.
- 6. Create a tuple and access its elements.
- 7. Convert a list into a tuple and vice versa.
- 8. Write a program to merge two dictionaries.
- 9. Write a function to count the frequency of elements in a list.
- 10. Create a dictionary of squares of numbers from 1 to 10.
- 11. Write a program to sort a list in ascending order.
- 12. Create a program to check if a key exists in a dictionary.
- 13. Create a set and perform union, intersection, and difference.

14. Write a function to find common elements in two lists.
15. Write a function that returns the factorial of a number.
16. Create a function that checks whether a string is a palindrome.
17. Write a function to count vowels in a string.
18. Create a dictionary and iterate over its keys and values.
19. Write a function to remove all punctuation from a string.
20. Write a function to capitalize the first letter of each word in a string.
21. Create a list comprehension to get squares of all even numbers in a range.
22. Write a function to check if a string is an anagram.
23. Create a nested dictionary to represent student records.
24. Write a function to flatten a nested list.
25. Write a program to find the second highest number in a list.
26. Create a function to rotate a list left by k positions.
27. Write a function to find the missing number from a list of 1 to N.
28. Write a program to remove all None values from a list.
29. Write a function to merge two dictionaries and handle key collisions by summing values.
30. Create a function to find unique elements present in only one of two lists.

Week 3: File Handling, Error Handling, Modules, Comprehensions

- Q1. Write a Python script to read a file and print its contents.
- Q2. Create a file and write your name into it.
- Q3. Handle a ZeroDivisionError using try-except.
- Q4. Write a program to handle file not found error.
- Q5. Create a module with a function and import it in another file.
- Q6. Use a list comprehension to filter even numbers from a list.
- Q7. Write a generator that yields even numbers up to N.
- Q8. Create a program to count lines and words in a file.

Q9. Write a program to read a CSV file and print its contents.

Q10. Handle multiple exceptions in a single try block.

Week 4: OOPs Concepts + Solve 50–100 Logic Building Problems

Q1. Write a Python program to check if a string has all unique characters.

Q2. Create a program that removes all duplicate characters from a string.

Q3. Write a script to count the frequency of each character in a string.

Q4. Write a program that accepts a sentence and calculates the number of upper and lower case letters.

Q5. Create a program to find the longest word in a sentence.

Q6. Write a program that takes a string and returns the string in reverse order without using `[::-1]`.

Q7. Create a Python function to check if a string is a pangram.

Q8. Write a Python script to sort words in a sentence alphabetically.

Q9. Write a program to check if two strings are anagrams.

Q10. Write a Python program to capitalize the first letter of each word in a sentence.

Q11. Create a program that extracts numbers from a string and returns their sum.

Q12. Write a program to replace all spaces in a string with underscores.

Q13. Write a function to count how many times a substring appears in a string.

Q14. Write a script to convert a string into title case without using `.title()`.

Q15. Write a Python program to merge two dictionaries into one.

Q16. Create a program to filter out all non-alphabetic characters from a string.

Q17. Write a function that returns True if a string ends with a given suffix.

Q18. Create a program that counts words, characters, and lines in a paragraph.

Q19. Write a script to encode a string using Caesar cipher (shift = 3).

Q20. Write a program that accepts a string and counts vowels and consonants.

Q21. Create a script to convert binary string to decimal.

Q22. Write a program to count the number of words starting with a vowel in a string.

Q23. Create a script that takes a sentence and removes all stop words.

Q24. Write a Python program to split a sentence into words and reverse each word.

- Q25. Write a function that returns a new string made of every third character of the original string.
- Q26. Write a program to find all palindromic substrings in a string.
- Q27. Write a function that compresses a string using run-length encoding.
- Q28. Write a Python program to count the frequency of each word in a file.
- Q29. Write a script that extracts hashtags from a tweet.
- Q30. Write a function to remove punctuation from a string.
- Q31. Create a program that finds the first non-repeating character in a string.
- Q32. Write a script that converts camelCase to snake_case.
- Q33. Write a function to generate acronyms from a sentence.
- Q34. Write a script to check if a file contains a specific word.
- Q35. Write a Python program to find and replace text in a file.
- Q36. Write a script that checks if all characters in a string are digits.
- Q37. Write a program to calculate the average word length in a sentence.
- Q38. Create a function that removes all HTML tags from a string.
- Q39. Write a program to parse a date string and display it in a different format.
- Q40. Write a script that finds all email addresses in a given text.
- Q41. Write a program that counts the occurrence of each vowel in a paragraph.
- Q42. Create a function that validates an email address format.
- Q43. Write a script to check if a string is a valid URL.
- Q44. Write a program that extracts all integers from a given text.
- Q45. Create a script to find duplicate words in a paragraph.
- Q46. Write a program that converts a sentence to Pig Latin.
- Q47. Write a script that finds the longest sentence in a paragraph.
- Q48. Write a Python program to read a file and display all lines that contain a given keyword.
- Q49. Write a script to clean a text file by removing extra spaces and blank lines.
- Q50. Write a Python program to count how many sentences are in a paragraph.

Step 2: Maths (Week 5–8)

"Understand the numbers — because Data Science is 80% about interpreting data correctly."

Recommended Resource: Watch our 5-hour full Statistics session on Sheryians AI School (YouTube) —

<https://www.youtube.com/watch?v=eF7HoC-cLRM&t=17267s>

Week 5: Statistics Basics – Descriptive Statistics

Mean, Median, Mode – Concepts & Formulas

Variance & Standard Deviation – Why do we need them?

Range, Quartiles, IQR

Understanding Data Distribution (Normal vs Skewed)

Hands-on with Python: `mean()`, `median()`, `std()`, `describe()` in NumPy & Pandas

Q1. Create a NumPy array of 10 random integers between 1 and 100. Find its mean.

Q2. Calculate the median of the array: [10, 5, 8, 12, 3, 7].

Q3. Generate an array of 15 random numbers between 1 and 50. Find its standard deviation.

Q4. Create a 3x3 array with values from 1 to 9. Find the row-wise mean.

Q5. For the array [4, 6, 8, 10, 12], calculate mean, median, std.

Q6. Generate a 5x5 NumPy array of random integers (0 to 100). Find the overall mean.

Q7. Create an array of 50 random integers and calculate the standard deviation.

Q8. Find indices of non-zero elements from [1,2,0,0,4,0]

Q9. You have an 2D array, print array element [[7,4]]:

[[2,3,4

5,7,4

8,9,0]]

Q10. Create 2 array of 2-Dimension and then perform matrix multiplication on them.

Week 6: Probability & Combinatorics

Basic Probability Rules (Addition, Multiplication)

Conditional Probability & Bayes Theorem

Permutations & Combinations (Simple problems)

Expected Value & Real-World Examples

Q1. A dice is thrown once. Find the probability of getting a 2 or 5.

Q2. A card is drawn from a deck of 52 cards. Find the probability of getting a red card or a king.

Q3. A number is chosen from 1 to 10. Find the probability that the number is even or divisible by 3.

Q4. In a bag, there are 3 red and 2 blue balls. A ball is drawn randomly. Find the probability of getting a red ball or a blue ball.

Q5. A spinner has 6 equal parts numbered 1 to 6. Find the probability of spinning a 1 or 6.

Q6. A card is drawn twice with replacement. Find the probability that both cards are spades.

Q7. A card is drawn from a deck and not replaced, then another card is drawn. Find the probability that both are aces.

Q8. A coin is tossed twice. Find the probability of getting head on both tosses.

Q9. A dice is rolled twice. Find the probability that first die shows 3 and second die shows an even number.

Q10. A bag contains 4 white and 2 black balls. Two balls are drawn without replacement. Find the probability that both are black.

Q11. A coin is tossed once. Find the probability of getting a tail.

Q12. A dice is thrown once. Find the probability of getting a number greater than 4

Q13. A card is drawn from a deck. Find the probability that it is a queen or a red card.

Q14. A box contains 2 red, 3 blue, and 5 green balls. One ball is drawn. Find the probability that it is green.

Q15. A dice is rolled twice. Find the probability that both numbers are odd.

Q16. A coin is tossed 3 times. Find the probability of getting exactly 2 heads.

Q17. A bag has 4 yellow and 1 black ball. A ball is drawn and not replaced, then another ball is drawn. Find the probability that both balls are yellow.

Q18. Two cards are drawn together from a deck. Find the probability that both are kings.

Q19. A spinner with 4 equal sections numbered 1 to 4 is spun twice. Find the probability that both spins show the same number.

Q20. A box has 5 red and 5 blue balls. One ball is drawn at random. Find the probability that it is not blue.

Week 7: Inferential Statistics – Hypothesis Testing

Population vs Sample

Sampling Techniques

Hypothesis Testing: Null & Alternate Hypotheses

p-values, Significance Levels (α)

T-tests & Chi-Square Test (Intuition only, no heavy math)

Q1. A coin is tossed 50 times and shows 30 heads. State the null and alternate hypotheses to check if the coin is fair.

Q2. A factory claims its bulbs last 1000 hours on average. Formulate the null and alternate hypotheses for testing this claim.

Q3. A new teaching method is applied to a class. Formulate hypotheses to test whether the new method improves average student marks.

Q4. A company claims 40% of customers prefer online shopping. Formulate hypotheses to test this claim.

Q5. If a hypothesis test gives p-value = 0.03 and $\alpha=0.05$, should you reject the null hypothesis?

Q6. If a test gives p-value = 0.15 and $\alpha = 0.05$, what is your decision about the null hypothesis?

Q7. Explain what it means if p-value < α in hypothesis testing.

Q8. If $\alpha = 0.01$, what is the confidence level of the test?

Q9. You have the heights of 10 male and 10 female students. Which test will you perform to check if their average heights are different?

Q10. A training program is introduced. Employee performance is measured before and after. Which T-test will check if the program improved performance?

Q11. You roll a dice 120 times and record how many times each number appears. Use a Chi-Square test to check if the dice is fair.

Q12. A chips brand claims the average packet weight is 50g. You weigh 10 random packets:

[49, 52, 51, 50, 48, 49, 50, 51, 52, 50]

Use a One-Sample T-Test to check if the claim is true.

Q13. A hospital claims a new medicine has a 70% success rate. In a trial of 20 patients, 12 recovered.

Use a proportion test to check if the success rate matches the claim.

Q14. A survey of 30 people records gender (Male/Female) and favorite drink (Tea/Coffee).

Perform a Chi-Square test to check if drink preference depends on gender.

Q15. A factory claims its bulbs last 1000 hours on average. You test 8 bulbs:

[980, 1005, 995, 1010, 1000, 985, 1020, 990]

Use a One-Sample T-Test to verify the claim.

Step 3: Data Preprocessing & Visualization (Week 9–11)

Videos - Numpy - <https://www.youtube.com/watch?v=Utgwk0r9Zq4&t=365s>

Pandas - <https://www.youtube.com/watch?v=QUaSmqBeR9w>

Visualization - <https://www.youtube.com/watch?v=-jTD74eEy2I&t=5057s>

"Now that you understand data, learn to clean and visualize it."

Week 9: Data Cleaning – Missing Values, Outliers, Encoding, Scaling

Identifying & Handling Missing Data (drop, fill, impute)

Detecting & Treating Outliers (IQR, Z-score)

Encoding Categorical Variables (Label, One-Hot)

Feature Scaling (Min-Max, Standardization)

- Q1. Count how many missing values are in a DataFrame.
- Q2. Show which values are missing in a DataFrame (True/False).
- Q3. Remove all rows with missing values.
- Q4. Fill missing values in a column with 0.
- Q5. Fill missing values in a column with "Not Available".
- Q6. Print the maximum and minimum value of a column.
- Q7. Find values in a column that are greater than 100.
- Q8. Replace all values greater than 100 with 100.
- Q9. Sort a column to check for extreme values.
- Q10. Remove the highest value from a numeric column.
- Q11. Convert column Gender with Male/Female into 0/1.
- Q12. Create separate columns for each City from a City column.
- Q13. Count how many times each category appears in a column.
- Q14. Change all values in a column to lowercase.
- Q15. Divide all values in a numeric column by 100.
- Q16. Subtract the minimum value from a numeric column.
- Q17. Convert all column values to range 0 to 1 manually.
- Q18. Show the mean and standard deviation of a numeric column.
- Q19. Subtract column mean from each value (basic standardization).
- Q20. Round all values in a numeric column to 2 decimals.

Week 10: pandas + numpy Deep Dive – DataFrames, Aggregations, Joins

pandas for DataFrames (create, read, filter, modify)

Aggregations (groupby, agg, pivot_table)

Merging & Joining multiple datasets

numpy for numerical operations (arrays, stats, reshaping)

Q1. Create a DataFrame of 5 students with columns: Name, Age, Marks.

Q2. Display the first 3 rows and last 2 rows of the DataFrame.

Q3. Show the shape of the DataFrame (rows × columns).

Q4. Access only the Name column from the DataFrame.

Q5. Filter all rows where Marks are greater than 50.

Q6. Add a new column Result with values: "Pass" or "Fail".

Q7. Update the Marks of the first student to 95.

Q8. Delete the Result column from the DataFrame.

Q9. Sort the DataFrame by Marks in descending order.

Q10. Count the number of unique names in the Name column.

Q11. Create a DataFrame of Sales with columns: Region, Product, Sales.

Q12. Calculate the total sales.

Q13. Calculate the average sales per product.

Q14. Count how many times each product appears in the dataset.

Q15. Find maximum and minimum sales per region using groupby and agg.

Q16. Create a pivot table showing total sales per Region vs Product.

Q17. Create a pivot table showing average sales per Region.

Q18. Create a NumPy array of numbers from 1 to 10.

Q19. Find the sum, mean, and standard deviation of the array.

Q20. Reshape an array of numbers from 1 to 12 into a 3×4 matrix.

Q21. Create a 2D NumPy array of random integers (3×3).

Q22. Find the sum of each row and sum of each column.

Q23. Multiply two 3×3 matrices using NumPy.

Q24. Find the maximum and minimum value in the array and their indices.

Q25. Slice a NumPy array to get only even numbers.

Week 11: Data Visualization – matplotlib, seaborn, EDA Techniques

matplotlib basics: Line, Bar, Scatter, Pie

seaborn for advanced plots: Heatmaps, Boxplots, Pairplots

Exploratory Data Analysis (EDA): Identify trends & patterns

Visual Storytelling: Making insights clear

Q1. Create a dataset with the following columns-> Products,Regions,Sales,Profit and 10 rows:

Q2. Plot a line chart of Sales for all 10 rows (index as X-axis).

Q3. Plot a bar chart showing total sales for each product.

Q4. Plot a scatter plot of Sales vs Profit.

Q5. Create a pie chart showing total sales by region.

Q6. Plot a histogram of the Sales column.

Q7. Create a boxplot of Profit to detect outliers.

Q8. Create a countplot showing how many products are sold in each region.

Q9. Create a heatmap to show correlation between Sales and Profit.

Q10. Show total and average sales per region using pandas.

Step 4: SQL (Week 11–13)

"Welcome to the world of real-world data – learn to query like a pro."

Week 12: Basics – SELECT, WHERE, ORDER BY, LIMIT, Aliasing

Introduction to Databases & SQL

SELECT statements to retrieve data

Filtering with WHERE (AND, OR, BETWEEN, IN, LIKE)

Sorting results with ORDER BY

Limiting results with LIMIT & OFFSET

Aliasing columns and tables

Q1. Create and Insert Data. Create a table named Employees with the following columns: EmployeeID, Name, Department, Salary, JoiningDate.

Q2. Select all columns from the Employees table.

Q3. Display only Name and Department columns.

Q4. Show all employees who work in the IT department.

Q5. Retrieve employees with a Salary greater than 45,000.

Q6. Show employees who joined after 2020-01-01.

Q7. Retrieve employees with a salary between 40,000 and 55,000.

Q8. Display employees whose department is either HR or Finance.

Q9. Retrieve employees whose name starts with 'S'.

Q10. Show employees whose name ends with 'a'.

Q11. Display employees ordered by salary in descending order.

Q12. Display the first 3 employees based on joining date.

Q13. Retrieve employees skipping the first 2 rows using OFFSET.

Q14. Show employee names as Employee_Name using alias.

Q15. Display Department as Dept and Salary as Income.

Q16. Combine aliasing with sorting: Show top 3 highest paid employees with columns Employee_Name and Income.

Q17. Find the highest salary in the Employees table.

Q18. Find the total number of employees in each department.

Q19. Show the average salary of all employees.

Q20. Display the employee(s) with the lowest salary.

Week 13: GROUP BY, HAVING, Joins (Inner, Left, Right, Full), Subqueries

GROUP BY & Aggregation functions (SUM, AVG, COUNT, MAX, MIN)

Filtering groups with HAVING

INNER, LEFT, RIGHT, FULL Joins – Combining multiple tables

Writing Subqueries for filtering and aggregations

Q1. Count how many employees are in each Department.

Q2. Find the average salary per department.

Q3. Show the highest and lowest salary in each department.

Q4. Show the total salary paid per department.

Q5. Count how many employees joined in each year (use YEAR(JoiningDate)).

Q6. Find departments that have more than 1 employee.

Q7. Find departments where average salary > 50,000.

Q8. Find joining years where more than 2 employees joined.

Q9. Perform an INNER JOIN to show Employee Name with their Department Manager.

Q10. Perform a LEFT JOIN to list all employees and their managers, even if manager info is missing.

Q11. Show total salary per department using JOIN + GROUP BY.

- Q12. Find the employee with the highest salary using a subquery.
- Q13. Find all employees who earn more than the average salary.
- Q14. Find the second highest salary using a subquery.
- Q15. Find employees who joined after the employee with the lowest salary.
- Q16. List all departments that have any employee earning more than 60,000.
- Q17. Find the total number of employees and total salary in each department.
- Q18. List all employees whose salary is the maximum in their department.
- Q19. Show all departments where no employee earns less than 45,000 (use HAVING).
- Q20. Find employees whose joining year is the same as any HR department employee (use subquery).

Week 14: Advanced SQL – Window Functions, CTEs, Nested Queries + Projects

Window Functions (ROW_NUMBER, RANK, DENSE_RANK, PARTITION BY)

Common Table Expressions (CTEs)

Nested Queries for complex filtering

Building end-to-end SQL queries

Mini Projects:

Project 1: Analyze an E-commerce dataset: Top-selling products, most active customers, revenue trends.

Project 2: Build a Student Management Dashboard using SQL: Average scores, top performers, pass/fail stats.

Step 5: Machine Learning (Week 15–21)

Videos - <https://youtube.com/playlist?list=PLaldQ9PzZd9qT0KsKJ7yCq70iFFP3MFJ5&si=zPjf0Wb98hFbaYLx>

"Now you're the magician – build models, make predictions, automate decisions."

Week 15: Intro to ML + Linear Regression

What is Machine Learning? (Types: Supervised, Unsupervised, Reinforcement)

Understanding Features & Labels

Simple & Multiple Linear Regression

Implementing Linear Regression with scikit-learn

Mini Projects:

Project 1: Predict Student Exam Scores: Predict a student's exam score based on study hours and other features like parental education and lunch type.

Project 2: House Price Prediction: Predict the price of a house using features like area, number of bedrooms, and bathrooms.

Week 16: Classification (Logistic Regression, KNN)

Logistic Regression (Binary & Multiclass)

K-Nearest Neighbors (KNN) – Concept & Implementation

Decision Boundaries & Model Interpretability

Project 1: Predict Student Pass/Fail using Study Hours and Scores

Project 2: Classify Iris Flowers into 3 Species using Petal and Sepal Measurements

Project 3: Predict if a Person has Diabetes based on Health Metrics (PIMA Diabetes Dataset)

Project 4: Classify Titanic Passengers as Survived or Not Survived based on Age, Gender, and Class

Week 17: Decision Tree, Random Forest

Decision Trees: Splitting Criteria (Gini, Entropy)

Overfitting & Pruning Techniques

Random Forests – Bagging & Feature Importance

Project 1: Predict Mushroom Type (Edible or Poisonous) using Physical Features (Mushroom Dataset)

Project 2: Classify Mobile Price Range (Low/Medium/High) based on Technical Specifications

Week 18: SVM, Naive Bayes

Support Vector Machines: Margins & Kernels

Gaussian Naive Bayes: Probabilistic Classification

When to use SVM vs Naive Bayes

Project 1: Classify Emails as Spam or Not Spam using the SpamBase Dataset

Project 2: Predict Handwritten Digits (0–9) using the MNIST Dataset

Week 19: Clustering (K-Means, DBSCAN)

K-Means Clustering – Centroids & Inertia

DBSCAN for density-based clustering

Choosing K (Elbow Method, Silhouette Score)

Project 1: Customer Segmentation using Mall Customers Dataset

Project 2: Group Similar Movies based on Ratings or Features using MovieLens Dataset

Week 20: Model Evaluation – Confusion Matrix, Accuracy, Precision, Recall, ROC

Confusion Matrix & Classification Metrics

Precision, Recall, F1-Score – Tradeoffs

ROC Curve & AUC

Practice:

Project 1: Predict Loan Approval using the Loan Prediction Dataset

Project 2: Diagnose Diabetes using the Pima Indians Diabetes Dataset

Week 21: Cross-Validation, Hyperparameter Tuning + Mini ML Project

K-Fold Cross Validation

Grid Search & Random Search for Hyperparameter Tuning

Wrap-up: Build a mini end-to-end ML project

Mini Project: Build a model to predict employee attrition (classification) or house prices (regression).

Step 6: Deep Learning (Week 22–25)

"Let's mimic the human brain — welcome to neural networks."

Week 22: Basics – Perceptron, Activation Functions, Forward + Backprop

What is a Neural Network?

Perceptron: The simplest neuron

Activation Functions: Sigmoid, ReLU, Tanh, Softmax

Forward Propagation – How predictions are made

Backpropagation – How networks learn

Q1. Compute the Sigmoid, ReLU, and Tanh values for the array $[-2, -1, 0, 1, 2]$.

Q2. Calculate the weighted sum and Sigmoid activation for input $[2, 3]$ with weights $[0.5, 0.2]$ and bias 0.1 .

Q3. Implement a perceptron to simulate the AND gate.

Q4. Compute the Mean Squared Error (MSE) for $y_{\text{true}}=[1, 0, 1, 0]$ and $y_{\text{pred}}=[0.8, 0.2, 0.9, 0.1]$.

- Q5. Perform one step of weight update for a single neuron using gradient descent (manual backprop).
- Q6. Plot the Sigmoid, ReLU, and Tanh functions for values from -5 to 5.
- Q7. Create a step activation function and test it on input [0.2, -0.3, 1.5, -2].
- Q8. Calculate the derivative of the Sigmoid function for input [0,1,2].
- Q9. Initialize random weights and biases for a single neuron with 3 inputs.
- Q10. Compute the output of a perceptron for OR gate using randomly initialized weights and a step activation function.

Week 23: ANN (TensorFlow/Keras) – Build from scratch + Keras implementation

Building a Neural Network from scratch using NumPy

Introduction to TensorFlow & Keras

Defining models, compiling, training, and evaluating

Hands-on: Predicting house prices or MNIST digits

Project 1: Predict house prices using a simple Artificial Neural Network (Boston Housing Dataset – available in `sklearn.datasets`).

Project 2: Classify handwritten digits using ANN (MNIST Dataset – available in Keras).

Week 24: CNN – Image Data, Filters, Pooling, Image Classification

Convolution: Filters & Feature Maps

Pooling (Max, Average) & Flattening

Building CNNs for Image Classification

Using Pre-trained Models (Transfer Learning)

Project 1: Classify handwritten digits using CNN (MNIST Dataset – available in Keras).

Project 2: Build a CNN to classify cats vs dogs (Kaggle: “Dogs vs Cats” dataset).

Week 25 : RNN – Sequential Data, LSTM/GRU + DL Project

Recurrent Neural Networks: Why we need them

LSTM & GRU – Handling long-term dependencies

Sequence data examples: Text, Time-series

End-to-End Deep Learning Project

Project 1: Predict next words in a sentence using LSTM (Text Generation on small dataset).

Project 2: Stock price prediction using LSTM on historical stock data.

Step 7: Specialization (Week 25–26)

“Now choose your path: become a domain expert.”

Week 25: Choose 1 track —

NLP (Text, Sentiment, Transformers)

CV (Computer Vision)

BI Tools (Power BI/Tableau)

Data Engineering (Airflow, ETL)

Week 26: Build & deploy a capstone project + Make GitHub portfolio + Resume prep