### **Phase 1: Foundation (Weeks 1–4)**

#### **Week 1–2: Python Basics + Jupyter Notebooks**

##### **Practical Exercises:**

1. **Python Basics**:
   * **Task**: Write Python code to solve simple problems:
     + Print a message to the console.
     + Perform basic arithmetic operations (addition, subtraction, multiplication, division).
     + Write functions to perform mathematical operations (e.g., a function to calculate the area of a circle given its radius).
   * **Goal**: Familiarize yourself with Python syntax, variables, loops, and functions.
2. **Using Jupyter Notebooks**:
   * **Task**: Open a Jupyter Notebook and execute the following steps:
     + Create a new notebook and practice writing code in different cells.
     + Install and import key Python libraries (NumPy, pandas).
     + Practice writing simple Python scripts in the notebook (e.g., basic list operations, string manipulations).
   * **Goal**: Understand the interactive environment of Jupyter Notebooks and how to run code in segments (cells).
3. **Working with Data using Pandas**:
   * **Task**: Use pandas to load a CSV dataset (e.g., any dataset from Kaggle or UCI Machine Learning Repository).
     + Inspect the data with .head() and .info().
     + Access and manipulate columns in the DataFrame.
     + Filter rows based on conditions (e.g., filter data for a specific year or value).
   * **Goal**: Learn basic data manipulation techniques with pandas (dataframes, indexing, filtering).
4. **Simple Data Visualizations**:
   * **Task**: Create basic plots using Matplotlib and Seaborn.
     + Create a line plot, scatter plot, and histogram with your dataset.
     + Customize plot labels, titles, and legends.
   * **Goal**: Understand basic data visualization concepts, such as customizing and interpreting plots.

##### **Project: Simple Python Calculator**

* **Task**: Build a basic calculator using Python. This project will incorporate loops, functions, and user input.
  1. Ask the user to input two numbers.
  2. Provide an option to perform basic operations (addition, subtraction, multiplication, and division).
  3. Allow the user to continue using the calculator or exit.
  4. Write the Python code and handle user input safely (e.g., prevent division by zero).
* **Goal**: Practice using Python to solve real problems, integrate functions, and handle user inputs.

#### **Week 3–4: Data Manipulation and Analysis**

##### **Practical Exercises:**

1. **Data Cleaning and Preprocessing with Pandas**:  
   * **Task**: Take a dataset (e.g., housing prices, car sales, etc.) and clean it:
     + Identify and handle missing values (e.g., filling or dropping missing values).
     + Remove duplicate rows if they exist.
     + Convert data types for columns (e.g., convert a string column to datetime).
   * **Goal**: Get comfortable with data cleaning, a vital skill for data scientists.
2. **Exploratory Data Analysis (EDA) with Pandas**:  
   * **Task**: Perform EDA on a given dataset:
     + Summary statistics (mean, median, standard deviation) using .describe().
     + Identify correlations between variables with .corr().
     + Visualize data distributions using histograms or boxplots.
   * **Goal**: Develop the ability to perform basic EDA to uncover insights from a dataset.
3. **Advanced Data Visualizations**:  
   * **Task**: Create more advanced visualizations:
     + Boxplots to check for outliers.
     + Correlation heatmaps using Seaborn (sns.heatmap()).
     + Bar charts to show categorical data distribution.
   * **Goal**: Improve your ability to visualize data and convey insights visually.

##### **Project: EDA Project**

* **Task**: Choose a dataset (e.g., Titanic dataset, Iris dataset, or any Kaggle dataset) and perform EDA:
  1. Clean the data by handling missing values and duplicates.
  2. Perform summary statistics and visualizations to understand distributions and relationships.
  3. Write a short report summarizing key insights (e.g., which features are most correlated with the target variable, any trends you notice).
  4. If possible, try to predict a target variable (e.g., predict survival on the Titanic using logistic regression).
* **Goal**: Showcase your ability to clean data, perform in-depth analysis, and present findings effectively.

### **Phase 2: Data Science Fundamentals (Weeks 5–8)**

#### **Week 5–6: Statistics for Data Science + Exploratory Data Analysis (EDA)**

##### **Practical Exercises:**

1. **Descriptive Statistics**:  
   * **Task**: Calculate basic statistical measures on a dataset using pandas:
     + Mean, median, mode, variance, standard deviation.
     + Use df.describe() and df.value\_counts() for categorical variables.
   * **Goal**: Get comfortable using pandas to summarize data.
2. **Probability and Distributions**:  
   * **Task**: Simulate data using Python and understand different distributions:
     + Use numpy to generate random numbers from uniform, normal, and binomial distributions.
     + Plot the generated distributions using Matplotlib.
   * **Goal**: Understand how data can follow different probability distributions.
3. **Hypothesis Testing**:  
   * **Task**: Perform a simple hypothesis test (e.g., t-test or chi-squared test):
     + Check if there is a significant difference between two groups (e.g., comparing the average income of two groups).
     + Use scipy.stats to perform the test and interpret the p-value.
   * **Goal**: Learn how to use hypothesis testing to make data-driven decisions.

##### **Project: Statistical Analysis on a Dataset**

* **Task**: Choose a dataset and perform a hypothesis test:
  1. Identify a question that can be answered with a hypothesis test (e.g., "Does the average age of buyers differ between two regions?").
  2. Preprocess the data and prepare it for analysis.
  3. Apply the appropriate statistical test (e.g., t-test) and interpret the results.
  4. Present the findings, including the p-value and whether you reject or accept the null hypothesis.
* **Goal**: Demonstrate your understanding of hypothesis testing and statistical analysis.

#### **Week 7–8: SQL + Data Cleaning**

##### **Practical Exercises:**

1. **SQL Basics**:  
   * **Task**: Use SQL queries to extract and manipulate data:
     + Write SELECT statements to filter, order, and aggregate data.
     + Use JOIN to combine tables and retrieve related data.
     + Group and aggregate data using GROUP BY.
   * **Goal**: Learn to query databases using SQL and practice on datasets.
2. **Data Cleaning in SQL**:  
   * **Task**: Perform data cleaning in SQL:
     + Remove duplicate rows using DISTINCT.
     + Handle missing data by replacing or filtering out rows.
   * **Goal**: Practice cleaning data directly in SQL before bringing it into Python for further analysis.

##### **Project: SQL Data Cleaning and Analysis**

* **Task**: Use SQL to clean and analyze a dataset (e.g., sales data or customer information):
  1. Write SQL queries to clean the data (e.g., remove duplicates, handle missing values).
  2. Use SQL to analyze the dataset: Find the top-selling products, average customer age, etc.
  3. Bring the cleaned dataset into Python and perform further analysis or modeling.
* **Goal**: Showcase your ability to clean and query data using SQL, and use the data for analysis.

### **Phase 3: Machine Learning Basics (Weeks 9–12)**

#### **Week 9–10: Supervised Learning (Regression & Classification)**

##### **Practical Exercises:**

1. **Linear Regression**:  
   * **Task**: Implement linear regression to predict a continuous variable:
     + Use scikit-learn’s LinearRegression() to predict housing prices or sales.
     + Evaluate the model using metrics like R-squared and Mean Absolute Error (MAE).
   * **Goal**: Understand regression analysis and how to evaluate a model.
2. **Logistic Regression**:  
   * **Task**: Implement logistic regression to classify binary outcomes:
     + Use scikit-learn’s LogisticRegression() to predict whether a customer will buy a product (binary classification).
     + Evaluate the model using accuracy, precision, recall, and F1-score.
   * **Goal**: Learn classification algorithms and evaluation metrics.

##### **Project: Predicting Housing Prices**

* **Task**: Use linear regression to predict housing prices based on features like size, location, etc.:
  1. Load a housing dataset and perform EDA and cleaning.
  2. Train a linear regression model to predict housing prices.
  3. Evaluate model performance using metrics like Mean Squared Error (MSE) or R-squared.
* **Goal**: Apply regression techniques to real-world data and evaluate model performance.

#### **Week 11–12: Unsupervised Learning (Clustering, Dimensionality Reduction)**

##### **Practical Exercises:**

1. **K-means Clustering**:  
   * **Task**: Apply K-means clustering to group data into clusters:
     + Use scikit-learn’s KMeans() to cluster a dataset (e.g., customer segmentation or document clustering).
     + Visualize the clusters with scatter plots or pair plots.
   * **Goal**: Learn how to use clustering algorithms for unsupervised learning.
2. **Principal Component Analysis (PCA)**:  
   * **Task**: Apply PCA for dimensionality reduction:
     + Use PCA() to reduce the number of features in a dataset.
     + Visualize the results and explain how PCA is reducing the complexity of data.
   * **Goal**: Understand dimensionality reduction and its importance in data science.

##### **Project: Customer Segmentation using K-means**

* **Task**: Use K-means clustering to segment customers based on purchasing behavior:
  1. Load a customer dataset (e.g., retail transaction data).
  2. Apply K-means to segment customers into different clusters.
  3. Analyze each cluster and interpret the customer segments.
* **Goal**: Demonstrate clustering and pattern recognition skills.

### **Phase 4: Advanced Data Science + Projects (Weeks 13–16)**

#### **Week 13–14: Deep Learning**

##### **Practical Exercises:**

1. **Build a Simple Neural Network**:
   * **Task**: Implement a simple neural network using Keras or TensorFlow:
     + Build a neural network with one hidden layer to classify handwritten digits (MNIST dataset).
     + Train and evaluate the model using accuracy.
   * **Goal**: Understand the basics of neural networks and deep learning.

##### **Project: Image Classification with a Neural Network**

* **Task**: Build a convolutional neural network (CNN) to classify images:
  1. Use a dataset like MNIST or CIFAR-10.
  2. Implement a CNN using Keras or TensorFlow to classify images.
  3. Evaluate the performance and try improving the model (e.g., by adding layers).
* **Goal**: Apply deep learning techniques to a real-world problem.

Here’s a list of **video resources** (primarily free) for each phase of your data science learning journey. These videos will complement the schedule and give you the step-by-step guidance you need to master each phase.

### **Phase 1: Foundation (Weeks 1–4)**

#### **Week 1–2: Python Basics + Jupyter Notebooks**

1. **Python Programming for Beginners** (Intro to Python)  
   * **Video**: [Python for Beginners - FreeCodeCamp](https://www.youtube.com/watch?v=rfscVS0vtbw)
   * **Length**: 4 hours 26 minutes
   * **Focus**: Introduction to Python basics, including variables, loops, functions, and more.
2. **Intro to Jupyter Notebooks**
   * **Video**: [Jupyter Notebooks for Beginners - Tech with Tim](https://www.youtube.com/watch?v=HW29067qVWk)
   * **Length**: 8 minutes
   * **Focus**: Basic walkthrough of how to use Jupyter Notebooks for data science.
3. **Python for Data Science: Pandas Basics**
   * **Video**: [Python Pandas Tutorial - FreeCodeCamp](https://www.youtube.com/watch?v=vmEHCJofslg)
   * **Length**: 4 hours
   * **Focus**: Learn how to use Pandas for data manipulation (importing, cleaning, and exploring data).
4. **Matplotlib and Seaborn for Data Visualization**
   * **Video**: [Python Data Visualization with Matplotlib and Seaborn - FreeCodeCamp](https://www.youtube.com/watch?v=0P7S4sWpgA0)
   * **Length**: 3 hours
   * **Focus**: Learn how to create visualizations using Python libraries like Matplotlib and Seaborn.

#### **Week 3–4: Data Manipulation and Analysis**

1. **Data Cleaning and Preprocessing in Python**
   * **Video**: [Data Cleaning with Python - Data Science Tutorials](https://www.youtube.com/watch?v=wi1vnVYvlmA)
   * **Length**: 1 hour 7 minutes
   * **Focus**: Learn the techniques of handling missing data, removing duplicates, and general preprocessing.
2. **Exploratory Data Analysis (EDA) with Pandas and Seaborn**
   * **Video**: [Exploratory Data Analysis (EDA) in Python](https://www.youtube.com/watch?v=7JHh0w_h7Xo)
   * **Length**: 45 minutes
   * **Focus**: Learn to explore datasets using summary statistics and visualizations in Python.
3. **Advanced Data Visualizations with Seaborn**
   * **Video**: [Advanced Visualization with Seaborn - Data School](https://www.youtube.com/watch?v=GXkxoPj1c8g)
   * **Length**: 1 hour 30 minutes
   * **Focus**: Learn advanced plotting techniques with Seaborn.
4. **Practical Data Science with Python (Beginner EDA Project)**
   * **Video**: [Practical Data Science with Python - Data Science Tutorials](https://www.youtube.com/watch?v=0y8rmFuSHck)
   * **Length**: 1 hour
   * **Focus**: Walkthrough of a beginner-friendly data analysis project.

### **Phase 2: Data Science Fundamentals (Weeks 5–8)**

#### **Week 5–6: Statistics for Data Science + Exploratory Data Analysis (EDA)**

1. **Intro to Statistics for Data Science**
   * **Video**: [Statistics for Data Science - Data Science Tutorials](https://www.youtube.com/watch?v=xxj3LYt-Yl0)
   * **Length**: 1 hour
   * **Focus**: Learn the fundamentals of statistics for data science, including mean, median, variance, and more.
2. **Probability and Statistics for Data Science**
   * **Video**: [Probability and Statistics for Data Science - Data Science Dojo](https://www.youtube.com/watch?v=2xVZGzSz_JA)
   * **Length**: 1 hour 40 minutes
   * **Focus**: A deeper dive into probability and distribution concepts relevant for data science.
3. **Hypothesis Testing in Python**
   * **Video**: [Hypothesis Testing for Data Science - StatQuest](https://www.youtube.com/watch?v=0zZYBALbZ8s)
   * **Length**: 15 minutes
   * **Focus**: Learn the basics of hypothesis testing, including t-tests, p-values, and confidence intervals.
4. **Exploratory Data Analysis and Hypothesis Testing**
   * **Video**: [Exploratory Data Analysis and Hypothesis Testing - Python for Data Science](https://www.youtube.com/watch?v=6FOph9U5A_U)
   * **Length**: 50 minutes
   * **Focus**: Combining EDA and hypothesis testing for a comprehensive data analysis approach.

#### **Week 7–8: SQL + Data Cleaning**

1. **SQL for Data Science**
   * **Video**: [SQL for Data Science - Coursera (free)](https://www.youtube.com/watch?v=9Pzj7Aj25lw)
   * **Length**: 4 hours 15 minutes
   * **Focus**: SQL basics, including querying databases, filtering, and joining tables.
2. **SQL Data Cleaning**
   * **Video**: [SQL Data Cleaning - Data Science Tutorials](https://www.youtube.com/watch?v=7VQ4JTHXfiI)
   * **Length**: 50 minutes
   * **Focus**: Techniques for cleaning data using SQL queries (removing duplicates, handling null values, etc.).
3. **SQL Joins Explained (Inner, Left, Right, Full)**
   * **Video**: [SQL Joins Explained](https://www.youtube.com/watch?v=9yeOJ0Z3U6s)
   * **Length**: 20 minutes
   * **Focus**: Learn different types of SQL joins to merge datasets efficiently.

### **Phase 3: Machine Learning Basics (Weeks 9–12)**

#### **Week 9–10: Supervised Learning (Regression & Classification)**

1. **Linear Regression Tutorial with Python**
   * **Video**: [Linear Regression with Python - Data Science Tutorials](https://www.youtube.com/watch?v=6a7bQjxwGFM)
   * **Length**: 1 hour
   * **Focus**: Introduction to linear regression and implementing it with Python.
2. **Logistic Regression Tutorial**
   * **Video**: [Logistic Regression Explained - Data Science Tutorials](https://www.youtube.com/watch?v=zM4VZR0px8E)
   * **Length**: 1 hour 30 minutes
   * **Focus**: Learn logistic regression for classification tasks and how to evaluate it.
3. **Introduction to Scikit-Learn**
   * **Video**: [Intro to Scikit-learn - Machine Learning with Python](https://www.youtube.com/watch?v=xxwV-JgQxII)
   * **Length**: 1 hour
   * **Focus**: Understand how to use the Scikit-learn library for machine learning tasks.
4. **Evaluating Machine Learning Models**
   * **Video**: [Evaluating ML Models - Data Science Dojo](https://www.youtube.com/watch?v=85dtiMz7T0k)
   * **Length**: 1 hour 10 minutes
   * **Focus**: Learn to evaluate models using metrics such as accuracy, precision, recall, and F1-score.

#### **Week 11–12: Unsupervised Learning (Clustering, Dimensionality Reduction)**

1. **K-means Clustering Explained**
   * **Video**: [K-Means Clustering - Machine Learning with Python](https://www.youtube.com/watch?v=4b5k7gQ74Mw)
   * **Length**: 18 minutes
   * **Focus**: Learn how K-means clustering works and how to implement it with Python.
2. **PCA for Dimensionality Reduction**
   * **Video**: [Principal Component Analysis - Data School](https://www.youtube.com/watch?v=FgakZw6K1QQ)
   * **Length**: 15 minutes
   * **Focus**: Understand PCA and its application in reducing dimensionality.
3. **Clustering Algorithms Overview**
   * **Video**: [Overview of Clustering Algorithms](https://www.youtube.com/watch?v=EJeQKh4HsmY)
   * **Length**: 1 hour
   * **Focus**: Learn different clustering algorithms, including K-means, DBSCAN, and hierarchical clustering.
4. **Unsupervised Learning: Clustering and Dimensionality Reduction**
   * **Video**: [Clustering and Dimensionality Reduction - Python](https://www.youtube.com/watch?v=J0WkB5AnldI)
   * **Length**: 50 minutes
   * **Focus**: A walkthrough of clustering and dimensionality reduction techniques using Scikit-learn.

### **Phase 4: Advanced Data Science + Projects (Weeks 13–16)**

#### **Week 13–14: Deep Learning**

1. **Introduction to Neural Networks**
   * **Video**: [Neural Networks from Scratch - Data School](https://www.youtube.com/watch?v=aircAruvnKk)
   * **Length**: 1 hour 30 minutes
   * **Focus**: Learn the fundamentals of neural networks and how they work.
2. **Deep Learning with TensorFlow**
   * **Video**: [TensorFlow 2.0 Tutorial](https://www.youtube.com/watch?v=tPYj3fFJGjk)
   * **Length**: 4 hours 26 minutes
   * **Focus**: Introduction to deep learning using TensorFlow 2.0.
3. **Convolutional Neural Networks (CNNs)**
   * **Video**: [CNNs for Image Classification - DeepLizard](https://www.youtube.com/watch?v=YRhxdVk_sIs)
   * **Length**: 40 minutes
   * **Focus**: Learn the basics of CNNs and how to apply them to image classification tasks.
4. **Deep Learning in Python with Keras**
   * **Video**: [Deep Learning with Keras - Data Science Tutorials](https://www.youtube.com/watch?v=V1YauM7Rpd8)
   * **Length**: 1 hour 30 minutes
   * **Focus**: Introduction to deep learning using Keras for building neural networks.