Introduction to Data Science

What is Data Science?

Welcome to this course on Data Science. If you are active on Fb/Insta/X, **share your Data Science journey on these platforms and tag me**. I would love to see your progress.

I am so happy you decided to learn Data Science with me in my style. Lets step back and talk a bit about why do we need data science. What kind of tasks we will do as a data scientist in an organization

What is Data Science?

At its core, **Data Science** is the field of study that uses **mathematics, statistics, programming, and domain knowledge** to extract **meaningful insights** from data. Well that sounds like a bookish definition which you are free to write in your semester exams but Data Science pretty much blends various techniques from different disciplines to analyze large amounts of information and solve real-world problems.

Simple Definition:

Let me give you a very simple non fancy definition of Data Science

> Data Science is the process of collecting, cleaning, analyzing, and interpreting data to make informed decisions.

Why is Data Science Important?

In today's world, **data is everywhere**—from your online shopping habits to the sensors in smart devices. Companies use this data to:

- **Make better decisions** (e.g., which product to launch next).
- **Predict outcomes** (e.g., weather forecasts or stock prices).
- **Automate processes** (e.g., self-driving cars).
- **Personalize experiences** (e.g., Netflix recommendations, YouTube recommendations).

Key Steps in Data Science:

1. **Data Collection** – Gathering raw data from various sources (websites, databases, IoT devices, etc.).
2. **Data Cleaning** – Fixing errors and handling missing values (this takes up **80% of a data scientist's time**).
3. **Data Analysis** – Using statistical methods to find patterns and insights.
4. **Model Building** – Applying **machine learning** to make predictions.
5. **Interpretation & Communication** – Presenting findings in a clear way to help decision-making.

Where Do We See Data Science in Action?

- **Healthcare:** Predicting disease outbreaks and improving diagnoses.
- **E-commerce:** Personalized product recommendations.
- **Finance:** Detecting fraudulent transactions by recognizing patterns.
- **Entertainment:** Content recommendations on platforms like YouTube and Netflix.

Why Should You Care About Data Science?

- **High Demand:** There's a global shortage of skilled data scientists.
- **Great Pay:** The average salary of a data scientist in the US is **$120,000+ per year**; in India, ₹10-30 LPA for experienced professionals.
- **Future-Proof Career:** It powers everything from AI to business strategy—this field is only growing.

Note: The content you just read will be supplied to you as a downloadable PDF. Since I am writing a summarized version of the video content which will serve as revision notes, I recommend you try to read them along with watching the videos. Thanks and see you in the next lecture!

Data Science Lifecycle

The **Data Science Lifecycle** refers to the structured process used to extract insights from data. It involves several stages, from gathering raw data to delivering actionable insights. Here is a breakdown of each step:

1. Problem Definition

> Understanding the problem you want to solve.

- Identify business objectives and define the question to answer.
- Later, we will do a very interesting project called "Coders of Delhi" where we start by understanding the business objective.
- Example: "Can we predict customer churn?" or "What factors drive sales?". In "Coders of Delhi" project, the problem is how to find potential friends of a given person in a social network

Key Activities:

- Collaborate with stakeholders.
- Define success metrics.
- Set project goals and deliverables.

---

2. Data Collection

> Gathering relevant data from multiple sources.

- Sources may include databases, APIs, web scraping, or third-party datasets. Sometimes if this step is taken care of by another team or it's a data dump given by another team, we don't care where it came from.

Key Activities in data collection:

- Identify data sources (structured vs. unstructured).
- Collect data using SQL, Python, or automated pipelines.
- Ensure data relevance and completeness.

---

3. Data Cleaning (Data Preprocessing)

> Preparing raw data for analysis.

- This step addresses **missing values, duplicates, and inconsistencies**.

Key Activities:

- Handle missing or incorrect data.
- Standardize formats and remove duplicates.
- Manage outliers and inconsistencies.

Fun Fact: Data scientists spend **80% of their time** cleaning data!

---

4. Data Exploration (EDA – Exploratory Data Analysis)

> Analyzing data patterns and relationships.

- Understand data distributions and detect anomalies using visualizations.

Key Activities:

- Summarize data using statistics (mean, median, etc.).

- Visualize patterns (using **Matplotlib**, **Seaborn**, etc.).
- Identify correlations and outliers. (correlation is how two variables move in relation to each other and outlier is a data point that stands out as unusually different from the rest. Eg. A 190 Kg heavyweight person is an outlier)

---

5. Model Building

> Creating and training machine learning models.

- Use algorithms to predict outcomes or classify data.

Key Activities:

- Choose appropriate models (e.g., regression, decision trees, neural networks).
- Split data into training and testing sets.
- Train and fine-tune models.

Common Tools: Scikit-learn, TensorFlow, PyTorch.

---

6. Model Evaluation

> Measuring model performance and accuracy.

- Evaluate models using metrics to ensure reliability.

Key Activities:

- Use performance metrics (e.g., accuracy, RMSE, ROC curve) to answer questions like - *"How often is my model correct?"*
- Perform cross-validation for robustness. Train using some part of the data and test using some part and average out the accuracy. We will study this in detail later
- Compare multiple models for best outcomes.

Key Metrics:

- Classification: Accuracy, Precision, Recall, F1-Score.

- Regression: RMSE, R-squared.

---

7. Deployment

> Integrating the model into production systems.

- Deliver actionable results through APIs or dashboards.

Key Activities:

- Package the model for deployment (Usually done using web frameworks like Flask, and FastAPI).
- Automate pipelines for continuous learning (MLOps).
- Monitor performance post-deployment.

---

8. Communication & Reporting

> Sharing insights with stakeholders. At the end of the day the ML model solves a problem and proper reporting it to the concerned department is very important

Key Activities:

- Create dashboards
- Present findings clearly and concisely.
- Document the process and results.

---

9. Maintenance & Iteration

> Keeping the model accurate and up-to-date.

Key Activities:

- Monitor model performance.

- Update models with new data.
- Refine features and parameters.

Summary

The **Data Science Lifecycle** is a continuous, iterative process involving:

1. Problem Definition
2. Data Collection
3. Data Cleaning
4. Data Exploration
5. Model Building
6. Model Evaluation
7. Deployment
8. Communication & Reporting
9. Maintenance & Iteration

By following this lifecycle, data scientists transform raw data into meaningful insights that drive better decision-making.

Data Science Tools

I am enjoying teaching so far. When working in data science, the right tools make your work easier, faster, and more efficient. When I started my data science journey at IIT Kharagpur, I used to code using Pycharm and regular Python installation. I knew about Jupyter but wasn't familiar with its capabilities. From writing code to visualizing data, there are many options to choose from. Here is a breakdown of popular data science tools and why **Anaconda** with **Jupyter Notebook** is an excellent choice for beginners and advanced users.

How to run Python programs

The easiest way to run Python programs is by installing VS Code and using pip to install packages but we will use Anaconda and Jupyter notebooks

1. Jupyter Notebook (with Anaconda Distribution)

> An open-source web application that allows you to create and share documents with live code, equations, visualizations, and text.

Why Use Anaconda with Jupyter Notebook?

- **User-Friendly:** Interactive coding and easy-to-follow outputs, perfect for beginners.
- **All-in-One Package:** Anaconda includes essential libraries (NumPy, Pandas, Matplotlib + 1,500 other popular packages) pre-installed.
- **Ideal for Data Science:** Quick prototyping, data visualization, and exploratory analysis.
- **Environment Management:** Easily create isolated environments to manage package dependencies.

Common Use Cases:

- Data exploration and visualization
- Machine learning experiments
- Sharing research and reports

Installation: Althought we will do a detailed installation of Anaconda in the later sections, you can download and install the Anaconda distribution from anaconda.com. It includes Jupyter Notebook by default.

Command to Launch Jupyter Notebook:

```
jupyter notebook
```

Don't worry, we will do all these things step by step in the next section

---

2. Google Colab

> A free, cloud-based Jupyter Notebook environment provided by Google.

Why Use Google Colab?

- **Free GPU/TPU Access:** Great for deep learning without requiring expensive hardware.
- **Cloud-Based:** No local setup—just log in and start coding.
- **Collaboration:** Share notebooks via links for easy collaboration.

Common Use Cases:

- Machine learning and deep learning projects
- Quick experiments without local setup
- Collaborative projects

Access: Use Google Colab directly in your browser at colab.research.google.com.

---

3. VS Code (Visual Studio Code)

> A lightweight and powerful code editor by Microsoft with robust extensions.

Why Use VS Code?

- **Customizable:** Extensive extensions for Python and data science (e.g., Python extension by Microsoft).
- **Integrated Jupyter Support:** You can run Jupyter Notebooks directly in VS Code.
- **Debugging Tools:** Advanced debugging capabilities.

Common Use Cases:

- Large-scale data science projects
- Working with multiple languages (Python, R, etc.)
- Integrated development (data pipelines, APIs)

Installation: Can be downloaded from code.visualstudio.com but we will install and use Anaconda distribution throughout this Data Science course.

---

4. PyCharm

> A powerful, professional IDE for Python development by JetBrains.

Why Use PyCharm?

- **Professional Features:** Advanced code navigation, refactoring, and debugging.
- **Environment Management:** Virtual environment and package management built-in.
- **Scientific Mode:** Built-in support for Jupyter notebooks.

Common Use Cases:

- Large, production-level data science projects
- Building Python-based machine learning applications

Installation: Download from jetbrains.com/pycharm.

---

5. Cursor AI

> An AI-powered code editor designed for enhanced productivity with machine learning assistance.

Why Use Cursor AI?

- **AI Integration:** Code suggestions and completions for faster development.
- **Context-Aware:** Understands complex data science workflows.
- **Collaborative:** Works well with team-based projects.

Common Use Cases:

- Assisted coding for data science
- Accelerating research and prototyping
- Team collaboration

Access: You can visit cursor.so to download cursor AI but I don't recommend using it just yet. Its important to understand the basics of data science and programming before you use such AI assistants

---

Which Tool Should You Choose?

Tool	Best For	Key Advantage
Jupyter Notebook	Interactive analysis, education	Easy to use and visualize data
Google Colab	Deep learning, cloud-based projects	Free GPU/TPU and no local setup
VS Code	Large projects, debugging	Lightweight with advanced features
PyCharm	Enterprise-level, complex applications	Professional IDE with deep features
Cursor AI	AI-assisted coding, productivity	AI-enhanced code suggestions
Spyder	Academic research, scientific computing	MATLAB-like interface

Recommendation: If you're starting out or want a hassle-free experience, **Anaconda with Jupyter Notebook** is the best choice. For advanced AI and big data projects, **Google Colab** is an excellent free alternative. For robust, large-scale development, **VS Code** or **PyCharm** provides advanced capabilities. Since we are just starting our learning journey, I will be using Anaconda and Jupyter for the most part of this course

---

Summary

1. **Anaconda + Jupyter Notebook:** Best for beginners and interactive analysis.
2. **Google Colab:** Ideal for cloud-based work and deep learning.

3. **VS Code:** Perfect for integrated, large-scale projects.
4. **PyCharm:** A professional-grade IDE for Python development.
5. **Cursor AI:** AI-assisted productivity for fast development.

Choosing the right tool depends on your project size, complexity, and hardware needs. For most data science workflows, **Anaconda with Jupyter Notebook** offers the best balance of simplicity, flexibility, and power.