Introduction to Data Science

Praveen K. Chandaliya

DoAI, SVNIT Surat

July 31, 2025

"Data Science is the most sought-after profession of the 21st century. If Data is the new **oil**, *then Data Science is the* **combustion engine** *driving the digital revolution."*

What is Data Science?

Definition

- Data Science is an interdisciplinary field that uses statistics, computer science, and domain expertise to extract insights and knowledge from structured and unstructured data.
- Data science is the science of collecting, storing, processing, describing, and mapping data.

Tasks of Data Science

Task	Description
Problem Formulation	Define business/research objectives and data-driven goals.
Data Acquisition	Collect raw data from DB, APIs, web scraping, sensors.
Data Cleaning	Handle missing values, remove noise, and ensure consistency and correctness.
EDA	Summarize data using statistics and visualizations to uncover patterns.
Feature Engineering	Create, select, or transform variables to improve model effectiveness.
Modeling	Apply statistical or machine learning algorithms to extract insights or make predictions.
Model Evaluation	Assess model performance using metrics like accuracy, precision, recall, RMSE.
Deployment	Deploy models into real-world systems or applications.
Monitoring	Model behavior, Changes in input data, Model update, and Ensure Fairness.
Communication	Present findings via reports, dashboards, or visualizations

Tools and Technologies

Languages

- Python, R, SQL

Libraries

- pandas, NumPy, scikit-learn
- TensorFlow, PyTorch

Tools

- Jupyter, Tableau, Power BI
- Apache Spark, Hadoop

Databases

- MySQL, MongoDB, PostgreSQL

Data Collection

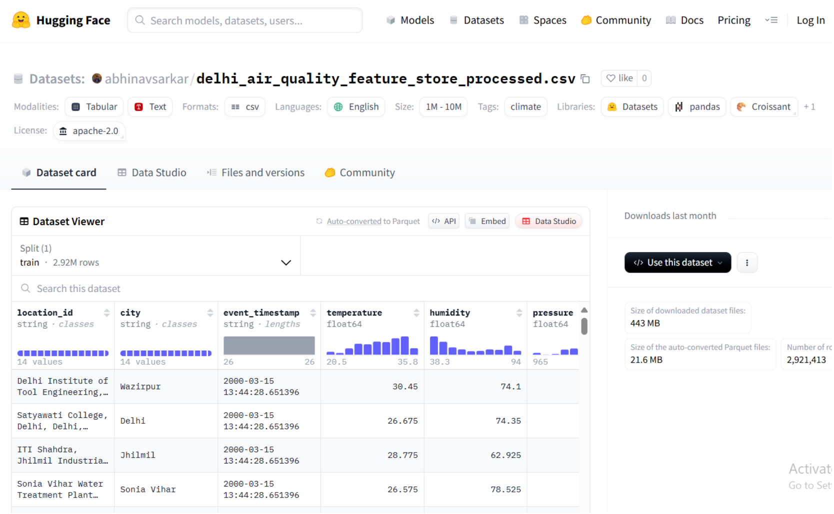
A Key Step in the Data Science Lifecycle

Air Pollution Prediction in Delhi

Data Sources and Methods

- **CPCB (Central Pollution Control Board) API** – Real-time air pollutant data (PM2.5, PM10, NO_2, CO) via REST API.
- **IoT Sensors** – Temperature, humidity, and wind speed from rooftop devices (via MQTT/HTTP).
- **Google Maps API** – Live traffic congestion data to estimate vehicular emissions.
- **Twitter API** – Public complaints and smog alerts using hashtag-based tweet streaming.
- **NASA Satellite Feeds** – Regional aerosol density and weather info from satellite imagery.
- **HuggingFace**- `https://huggingface.co/datasets/abhinavsarkar/delhi_air_quality_feature_store_processed.csv`

HuggingFace: Delhi Air Quality



Real-Time Data Collection: OpenAQ API

Steps to Use OpenAQ API

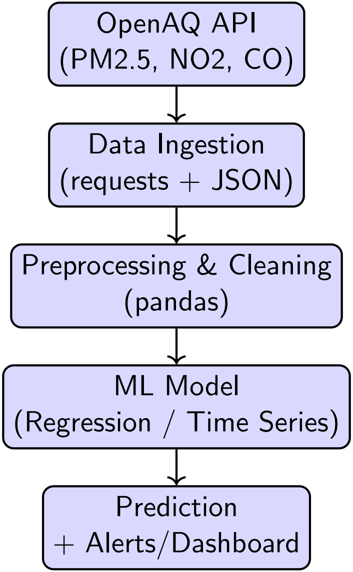
1. **Register:** Sign up at `https://accounts.openaq.org` and create a free account.
2. **Get API Key:** Go to the dashboard $\rightarrow$ API Keys $\rightarrow$ Generate a key.
3. **Set Headers:** Include the key in request headers using `X-API-Key`.
4. **Make Request:** Use endpoints like `/v3/measurements` to fetch data.

Example: Python Script to Fetch PM2.5 Data (Delhi)

Python Code

```
import requests
url = "https://api.openaq.org/v3/measurements"
params = {
    "country": "IN",
    "city": "Delhi",
    "parameter": "pm25",
    "limit": 5,
    "sort": "desc"
}
headers = {
    "accept": "application/json",
    "X-API-Key": "YOUR_API_KEY"  # Replace with your actual key
}
response = requests.get(url, params=params, headers=headers)
if response.status_code == 200:
    data = response.json()
    for result in data["results"]:
        print(f"{result['location']} | PM2.5: {result['value']} {result['unit']} |
else:
    print("Error:", response.status_code, response.text)
```

Real-Time Air Quality Monitoring: Data Pipeline



Data Collection in Political Campaigns

Objective

Collect multi-source data to analyze voter behavior, sentiment, and optimize campaign strategy.

Data Sources

- **Voter Registry:** Age, gender, location, and past voting history.
- **Social Media:** Posts, hashtags, likes, and engagement from Twitter, Facebook.
- **News Media:** Sentiment from news articles, debates, and TV coverage.
- **Polls and Surveys:** Voter intent, issue priorities, satisfaction scores.
- **Event Participation:** Rally attendance, volunteer sign-ups, and feedback.
- **Web and App Analytics:** Traffic to official websites and campaign apps.
- **Call/SMS Logs:** Responses from voter outreach and call center engagement.

Data Collection Sources for Political Campaigns

Source	**Type of Data**	**Collection Method**
Voter Registry	Age, gender, location, past voting behavior	Public electoral rolls, third-party providers
Social Media	Posts, hashtags, likes, sentiment	Twitter/Facebook APIs, scraping tools
News	Public sentiment, political narratives	News APIs, text scraping, NLP processing
Surveys & Polls	Voter opinions, issue priorities, satisfaction levels	Online/field surveys, IVR, Google Forms
Event Participation	Rally attendance, volunteer sign-ups	QR code scans, app/web registration logs
Web & App Analytics	Website traffic, user engagement	Google Analytics, Matomo, backend logs
Call/SMS Logs	Responses to outreach, voter feedback	CRM data, call center software

Table: Key Data Sources for Political Campaign Analytics

COVID-19 Data Collection

Source	Type of Data	Description
MoHFW, India	Daily case stats	State-wise COVID-19 cases, deaths, and testing numbers
ICMR	Testing data	RT-PCR, antigen test results, lab coverage
Johns Hopkins University (JHU)	Global time-series	Confirmed cases, recoveries, deaths worldwide
COVID19-India API	District-level data	Cases, vaccination progress, hospital bed availability
Google Mobility Reports	Mobility trends	Visits to parks, workplaces, retail from smartphone data
Twitter	Social signals	Public sentiment, outbreak signals, misinformation detection
Aarogya Setu	Contact tracing	Exposure risk based on Bluetooth and location history

Table: COVID-19 Data Collection Sources

Fetch Global COVID-19 Data from JHU GitHub

Download and Filter India-Specific Cases

```
import pandas as pd import matplotlib.pyplot as plt
# JHU confirmed global time-series data
url = "https://raw.githubusercontent.com/CSSEGISandData/" \
      "COVID-19/master/csse_covid_19_data/" \
      "csse_covid_19_time_series/" \
      "time_series_covid19_confirmed_global.csv"
df = pd.read_csv(url)
# Filter for India
india_df = df[df['Country/Region'] == 'India']
india_df = india_df.drop(['Province/State', 'Lat', 'Long'], axis=1)
india_df = india_df.set_index('Country/Region').T
india_df.index = pd.to_datetime(india_df.index)
# Plot daily confirmed cases
india_df.plot(title="COVID-19 Confirmed Cases in India", figsize=(1
plt.ylabel("Cases") plt.xlabel("Date")
plt.grid() plt.show()
```

Cricket Series Data Collection

- **Data Sources:**
  - **ESPN Cricinfo API / Web Scraping**
  - **Cricbuzz API (Unofficial)**
  - **Google Sports Widgets (live data)**
  - **Open-source cricket datasets (e.g., Cricsheet)**
- **Collected Data:**
  - Match summaries (scorecard, date, venue)
  - Ball-by-ball data (runs, wickets, player stats)
  - Player performance (batting average, strike rate)
  - Toss decisions, win margins, pitch info

Agricultural Data Sources in India

Major Platforms for Agricultural Data Collection:

Source	Data Type	Website
Ministry of Agriculture	Crop area, production, yield statistics	https://agricoop.nic.in
Directorate of Economics and Statistics (DES)	District-wise agricultural indicators	https://eands.dacnet.nic.in
Agmarknet (DACFW)	Mandi arrivals and prices of commodities	https://agmarknet.gov.in
Open Government Data (OGD) India	Rainfall, irrigation, fertilizer use	https://data.gov.in
FAOSTAT	Global and country-level agri statistics	https://www.fao.org/faostat
IndiaStat (Paid)	Historical agricultural data	https://www.indiastat.com

Specialized Agricultural Datasets in India

Thematic Datasets for Agri-Analytics:

Dataset	Content	Source
Rainfall Data	Daily/monthly rainfall by district/block	https://mausam.imd.gov.in
Irrigation Statistics	Canal, borewell, tank irrigation stats	https://eands.dacnet.nic.in
Fertilizer Consumption	District-level use of NPK fertilizers	https://fert.nic.in
Satellite Vegetation Indices	NDVI, LAI from remote sensing satellites	https://bhuvan.nrsc.gov.in
District-level Crop Production	Crop-wise (e.g., Wheat, Sugarcane) yield data	https://data.gov.in/catalogs/agriculture

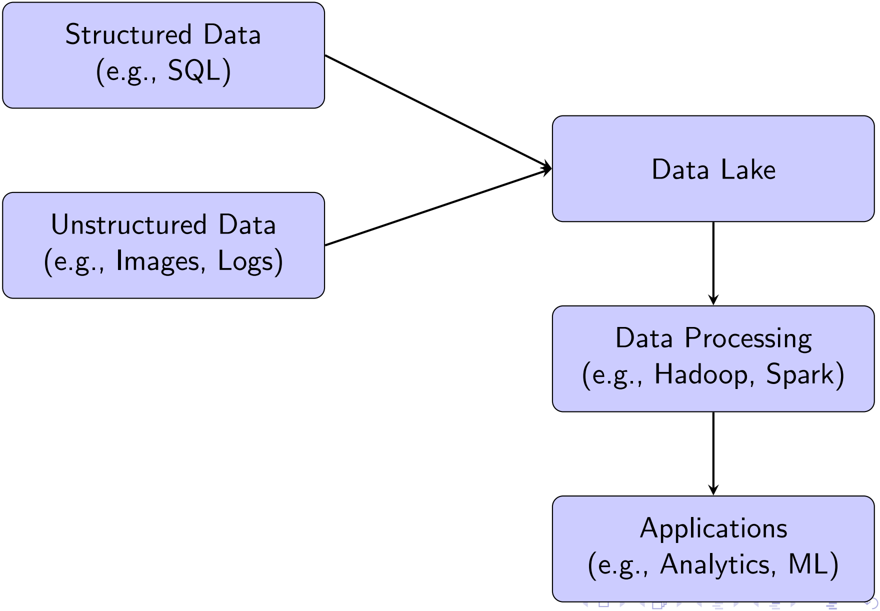
Data Storage

A Key Step in the Data Science Lifecycle

Types of Data Storage in Data Science

- **Structured Data:** Organized into rows and columns, stored in relational databases (e.g., MySQL, PostgreSQL).
- **Unstructured Data:** No fixed format; includes text, images, videos, etc. (e.g., social media posts, emails).
- **Data Lake:** Centralized repository storing structured and unstructured data at any scale (e.g., Hadoop, Amazon S3).

Application Sketch: Data Storage Workflow



Real-Time Application Examples

Data Type	Format	Storage	Application
Structured	Tabular	MySQL, Oracle	Fraud Detection, Statement Generation, Banking Transactions
Unstructured	Text, Images, Videos	MongoDB, Amazon S3	Sentiment Analysis, Trend Prediction
Data Lake	Structured + Unstructured	Amazon S3, Hadoop	Recommendation Systems, E-commerce Platform

Processing Data (Colab: Wrangling.ipynb)

A Key Step in the Data Science Lifecycle

https://colab.research.google.com/drive/1r_Dj0_tdUO-rSxgeoFhWUqHnHt_TogxG?usp=sharing

Data Wrangling

Definition:
Data wrangling refers to the process of cleaning, transforming, and organizing raw data into a more usable format for analysis or modeling.

Tasks Involved:

- Handling missing values
- Correcting data types (e.g., string to datetime)
- Filtering or removing outliers
- Normalization or scaling
- Merging or joining datasets
- Encoding categorical variables

Goal: Make data clean, structured, and analysis-ready.

Data Munging (Exploratory Understanding)

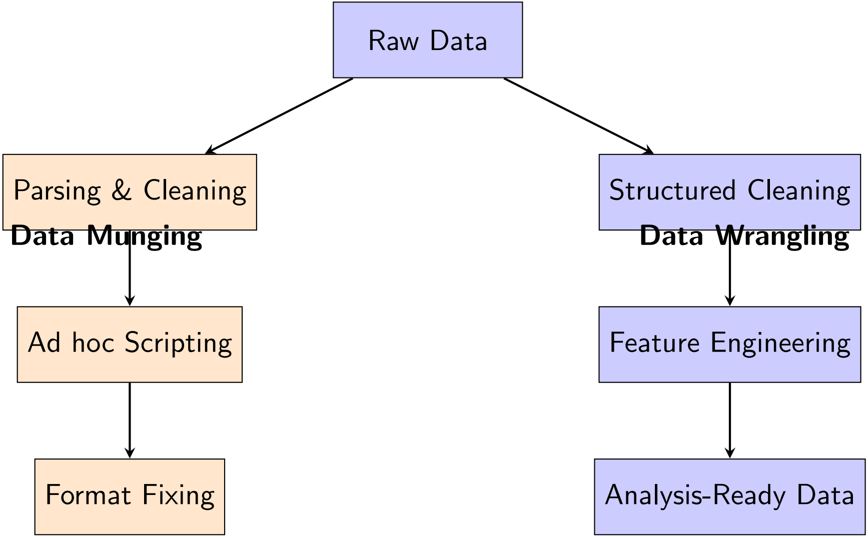
Definition:
"Data Munging" can also be interpreted as deeply understanding, appreciating, and exploring data before modeling—essentially, *"embracing"* the data.

Key Concepts:

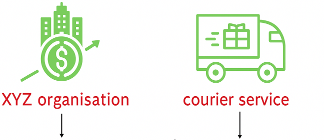
- Exploratory Data Analysis (EDA)
- Visualizing distributions and relationships (scatter plots, histograms)
- Understanding context, domain knowledge, and biases
- Asking key questions:
    - What is this data trying to tell?
    - Is there bias or noise?
    - What is the story behind each variable?

Goal: Build intuitive understanding before modeling or assumptions.
Example:
Plotting pairwise feature relationships, seasonal trend analysis, or real-world data correlation before training.

Data Munging vs Data Wrangling



Identifying the Data Sources



Item	timestamp	First Name	Last Name
book	1578079800	John	Doe

```
{
package_contedns: "book"
delivery date: "03-Jan-20
delivery time: "19:30:00"
receiver: "John Doe"
}
```

XYZ Organisation (Structured Data):

- Format: Database table or spreadsheet
- **Fields:** item, timestamp, First Name, Last Name
- **Example:** book, 1578079800, John, Doe

Courier Service (Semi-Structured Data):

- Format: JSON
- **Fields:** package_contents, delivery date, delivery time, receiver
- **Example:** "book", "03-Jan-2020", "19:30:00", "John Doe"

The Wrangling and Munging Process

Key Steps:

- **Data Integration:** Match receiver = First Name + Last Name; match item = package_contents
- **Data Parsing & Formatting:**
  - Convert Unix timestamp (1578079800) to date/time: Jan 3, 2020, 19:30:00
  - Combine First Name and Last Name into one field
- **Data Cleansing & Transformation:**
  - Resolve naming format differences
  - Ensure consistent datetime formats across sources
- **Field Mapping:** item → package_contents, First Name + Last Name → receiver

The Goal of Wrangling and Munging

Unified Dataset Objective:

- Merge data to show that an item was ordered and delivered to the same person at the same time.

Applications:

- **Analytics:** Analyze delivery times, satisfaction, purchasing patterns
- **Reporting:** Create delivery and order reports
- **Database Updates:** Enrich master tables with delivery records

Summary:

The process transforms and aligns multiple data formats into a clean, analyzable structure—this is the heart of **data wrangling and munging**.

Python Examples: Wrangling vs Munging

Data Wrangling (Cleaning/Transforming):

```
import pandas as pd
df = pd.read_csv("raw_data.csv")
# Handle missing values
df.fillna(df.median(), inplace=True)
# Convert data types
df['date'] = pd.to_datetime(df['date'])
# Encode categorical variable
df['gender'] = df['gender'].map('M': 0, 'F': 1)
# Normalize numeric column
df['income'] = (df['income'] - df['income'].mean()) / df['income'].s
```

Data Munging (Exploratory Analysis):

```
import seaborn as sns import matplotlib.pyplot as plt
# Visualize distributions
sns.histplot(df['income'])
# Check pairwise relationships
sns.pairplot(df[['income', 'age', 'spending_score']], hue='gender')
# Identify correlations print(df.corr())
```

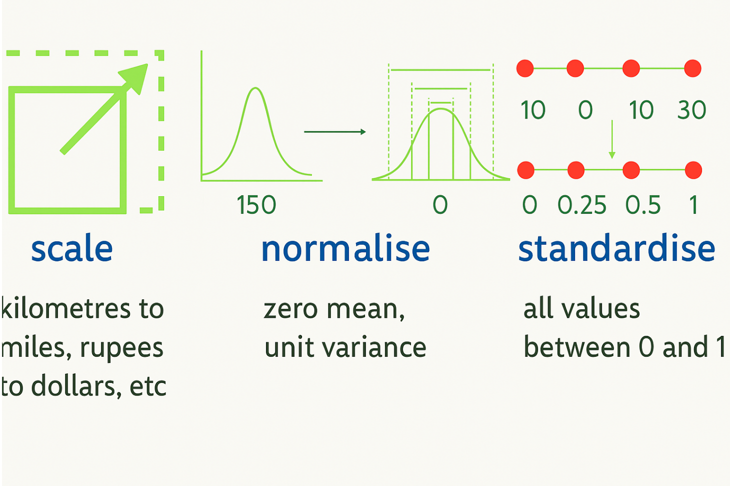
Data Wrangling vs Data Munging

Aspect	Data Wrangling	Data Munging
Definition	Structured process of cleaning, transforming, and organizing data	Ad hoc or script-based transformation of raw, messy data
Focus	Preparing data for analysis or machine learning	Making raw data machine-readable or structured
Common Tasks	Handling missing data, normalization, joining datasets	Parsing formats, regex cleanup, formatting
Tools	Python (Pandas), R (dplyr), SQL	Scripting (Python, Bash), awk, sed
Typical Use	Pipelines for modeling and analytics	Quick preprocessing or ETL steps

Date cleaning

- Fill missing values
- Standardize keywords tags
- Correct spelling errors
- Identifying and remove the outliers

Scaling, Normalizing, Standardizing



Describing Data (Colab: 1-EDA And Feature Engineering.ipynb)

A Key Step in the Data Science Lifecycle

https://drive.google.com/file/d/1XSqDZPkmCsiw8_nNyiWOdYHinF9IUPMQ/view?usp=sharing

i) Data Visualization

Definition:
Graphical representation of data using charts and plots to identify trends and patterns.

Example: COVID-19 Cases per Country

```
countries = ['USA', 'India', 'Brazil', 'UK']
cases = [32000, 28000, 21000, 15000]
plt.bar(countries, cases, color='skyblue')
plt.title('COVID-19 Cases per Country')
plt.xlabel('Country')
plt.ylabel('Number of Cases')
plt.show()
```

ii) Summarization of Data

Definition:
Statistical summarization helps condense data into key figures like mean, median, and standard deviation.

Example: Student Exam Scores

```
scores = [88, 76, 92, 85, 69, 94, 78]
mean_score = np.mean(scores)
median_score = np.median(scores)
std_dev = np.std(scores)
print(f"Mean: mean_score, Median: median_score, Std_Dev: std_dev")
```

Zomato Restaurants Data

- Sourced from Zomato API and Open dataset [1].
- Format: CSV, suitable for data cleaning, EDA, and modeling.
- Contains attributes such as:
  - Name, Location, Country, City
  - Cuisines, Rating, Votes, Cost for Two
  - Delivery and Booking Flags
- **Data Cleaning:** Handled missing values, removed duplicates.
- **EDA:** Used Pandas, Matplotlib, Seaborn.
- **Feature Engineering:** Encoded categorical data, grouped cuisines.
- **Modeling:** Optional - clustering, classification, or recommendation.

[1] `https://github.com/krishnaik06/5-Days-Live-EDA-and-Feature-Engineering`

Applications of Data Science

- Fraud Detection in Banking
- Predictive Maintenance in Industry
- Personalized Recommendations
- Healthcare Diagnosis and Drug Discovery
- Image and Speech Recognition
- Climate Modeling and Forecasting

Related Fields

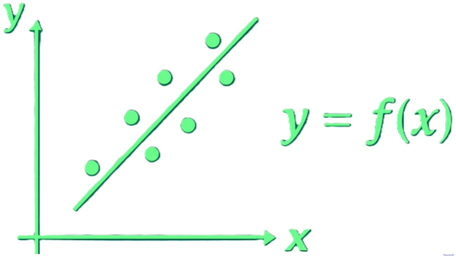
- **Machine Learning** – Algorithms and prediction
- **Big Data** – Handling massive datasets
- **Artificial Intelligence** – Decision making, planning, reasoning
- **Business Analytics** – Data-driven business decision making

Statistical Modeling Data vs Algorithm Modeling

Data Science, Machine Learning, and Deep Learning

Statistical Modeling Data

- A statistical model is a mathematical framework used to describe the relationship between different variables within a dataset. Its primary purpose is to approximate reality, allowing us to make predictions, infer relationships, and understand underlying patterns.
- A simple statistical model for this problem could be a linear regression model. Its job is to find the best-fit line (or hyper-plane) that describes the relationship between the Predictor variables/Independent Variables (x) and the Outcome variable/Dependent Variable (y).



Statistical Modeling: Example

Problem:
What is the relationship between the number of treatment days and blood sugar level?

Data:

Days (X)	Blood Sugar (Y)
1	180
2	174
3	170
4	165
5	162

Model: Linear Regression
$Y = mX + c$

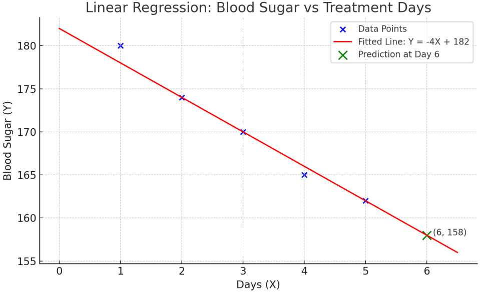
Model Estimation and Prediction

Estimated Parameters: $m = -4, \quad c = 182$

Fitted Model: Blood Sugar = -4 $\times$Days + 182

Prediction: $On Day 6$:Y = -4 $\times$6 + 182 = 158

Interpretation: Every additional treatment day reduces blood sugar by 4 units.



Algorithmic modeling

- Algorithmic modeling (Machine Learning (ML)) refers to the use of computational algorithms to model **complex relationships between inputs (features) and outputs (targets)**. These models are trained on historical data and are then used to predict or classify new, unseen data.
- **Examples:**
  - Decision Trees
  - Random Forests
  - Support Vector Machines (SVM)
  - Neural Networks

Key Components:

- Input Variables (Features)
- Output (Target variable)
- Loss Function & Optimization

Statistical vs Algorithmic Modeling

Statistical Modeling	**Algorithmic Modeling**
Assumes a data distribution	No assumption on data distribution
Suited for low-dimensional data	Work with higher-dimensional data
Emphasizes interpretability	Emphasizes predictive accuracy
Data lean models	Data hungry
Examples: Linear regression, Logistic Regression, PCA, SVD, ANOVA	Examples: Random Forest, SVM, K-NN, Neural Network

Understanding Deep Learning

When working with large volumes of high-dimensional data and aiming to learn complex relationships between inputs and outputs, a specialized class of machine learning models collectively referred to as **Deep Learning** is employed. These models utilize **deep neural networks** to automatically extract features and capture intricate patterns that traditional models may struggle to represent.

Conclusion

- Data Science drives decision-making across industries.
- It requires a blend of technical, analytical, and domain-specific skills.
- The field is rapidly evolving with advancements in AI and computing.