

Data Science & Machine Learning

A PROJECT BASED APPROACH

By. Larry Miguel R. Cueva

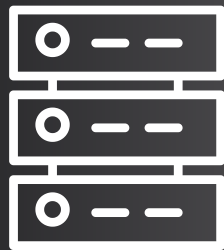
Extracting Data

Every data project be it data science, machine learning/deep learning, data analytics, and data engineering all require some sort of data in order to start building perhaps a predictive model, a data pipeline, or dashboards for analysis



Sources of Data

APIs



Provide access to data from external services. Essential for integrating with other applications and gathering real-time information. Uses HTTP requests to retrieve multiple instances or rows of data

Databases (Relational & NoSQL)



The backbone of many applications. Store structured (Relational) or semi-structured/unstructured (NoSQL) data. Developers use them for persistent storage, querying, and data retrieval for applications, analytics, and ML model training.

Websites



Extracts data from websites when no official API is available. Useful for gathering publicly available information. Source of data coming from websites is not readily available but can be made so via web scraping techniques using open source python libraries like selenium, beautifulsoup, scrapy.

Multimedia



Includes already publicly accessible and easily downloadable files on the internet i.e. images, videos, spreadsheets, javascript object notation (JSON) files, comma separated (CSV) files.

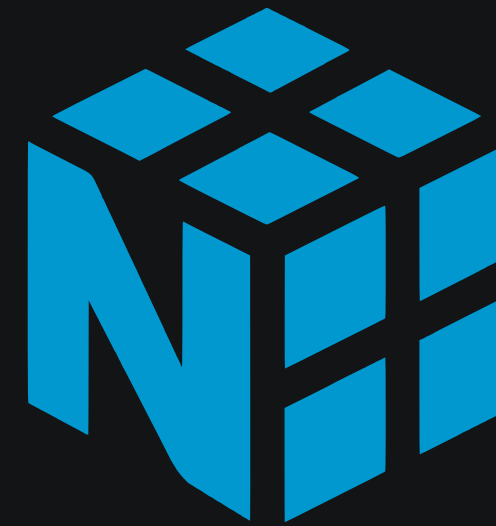
Data Cleaning & Preprocessing

- 1 Handling Missing Values either by Imputation i.e. Filling in missing values with estimated values (e.g., mean, median, mode) or by Removal i.e. Removing rows or columns with too many missing values.
- 2 Removing Duplicates: Identifying and removing duplicate records.
- 3 Correcting Inconsistent Data by either standardizing formats (e.g., date formats, address formats), fixing typos and spelling errors, or resolving conflicting data entries, or all these
- 4 Handling Outliers by identifying and dealing with extreme values that deviate significantly from the rest of the data. This might involve removal, capping, or transformation.

Common tools for the task

data cleaning & preprocessing are two crucial steps in the data preparation process, especially in fields like data science, machine learning, and data analytics. They ensure that the data is accurate, consistent, and in a suitable format for analysis and modeling.

Most common ones include numpy (upper), pandas (left), and scikit-learn



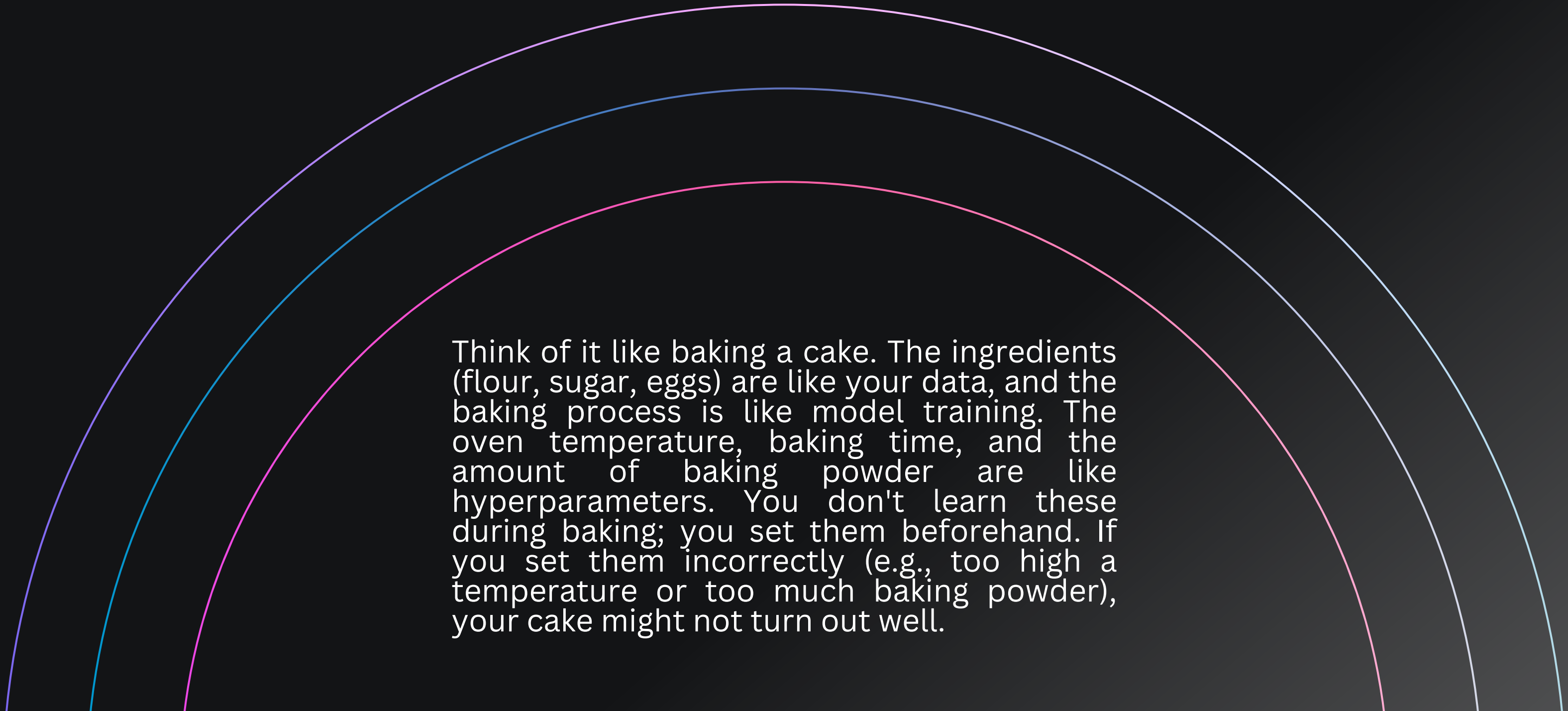
Feature Engineering

Feature engineering is the process of using domain knowledge to create new input features from existing raw data. These new features aim to improve the performance of machine learning models by providing them with more relevant and informative data. It's about crafting the best possible input for your model to learn from.

Think about it this way. A predictive model let's say a Random Forest classifier does not know how to distinguish a cat from a dog if you give it only a set of images, rather you have to transform these images into numerical values such that the model can understand it and finally be able to map out a pattern from these numerical values given its respective labels i.e. a cat or a dog.

Hyperparameter Tuning

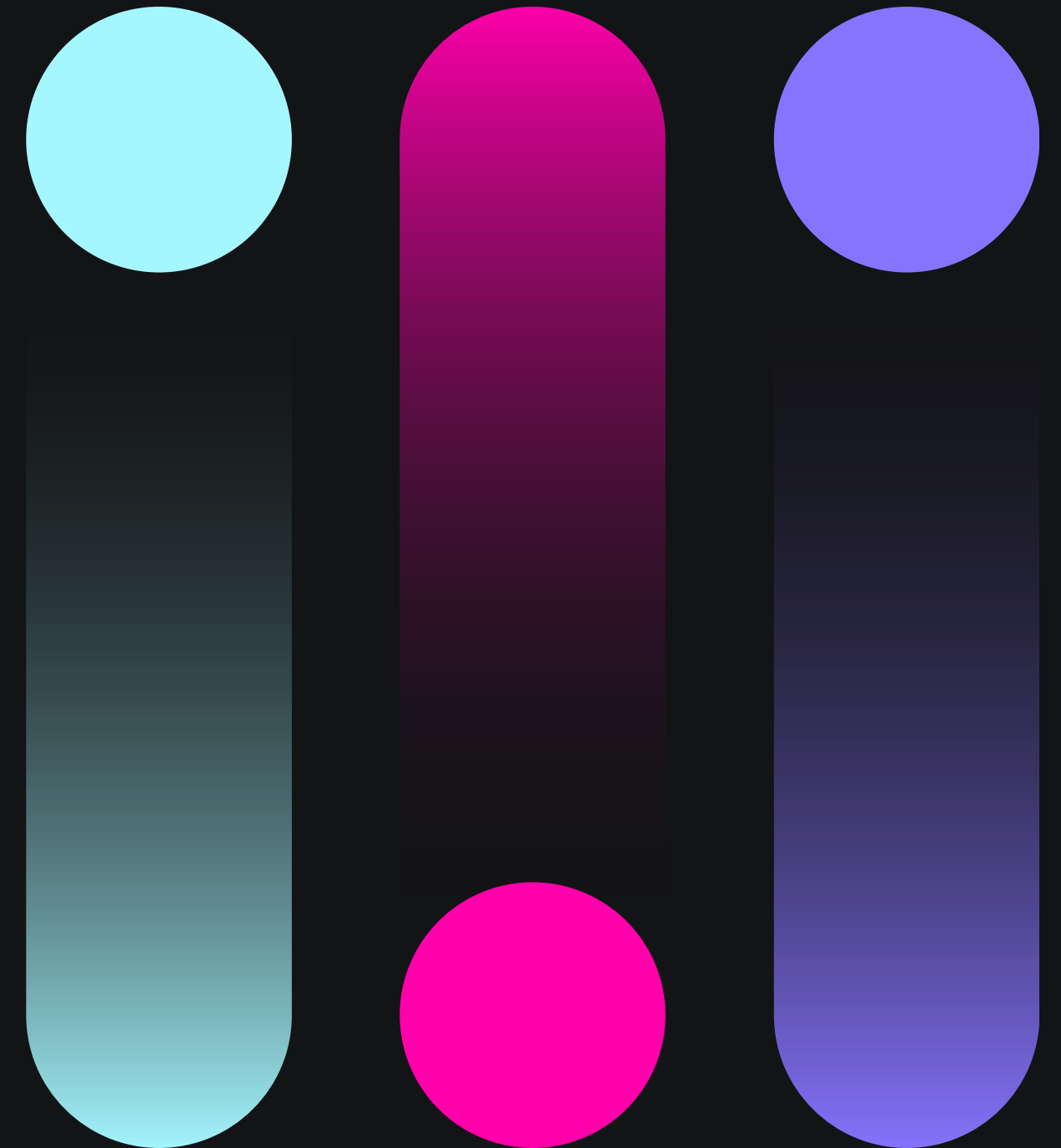
The process of finding the optimal set of hyperparameters for a machine learning model. Unlike model parameters (which are learned during training, like the weights in a neural network), hyperparameters are settings that are set before training begins and control the learning process itself.



Think of it like baking a cake. The ingredients (flour, sugar, eggs) are like your data, and the baking process is like model training. The oven temperature, baking time, and the amount of baking powder are like hyperparameters. You don't learn these during baking; you set them beforehand. If you set them incorrectly (e.g., too high a temperature or too much baking powder), your cake might not turn out well.

Model Training

- Once hyperparameters are obtained model training can now commence
- A process of teaching a machine learning model to recognize patterns in data
- The model adjusts its internal parameters through an optimization algorithm to minimize the difference between its predictions and the actual labels.
- This iterative process continues until the model achieves satisfactory performance
- Ultimately, training aims to create a model that can accurately predict outcomes on new, real-world data.



Model Testing/Evaluation

Model evaluation in machine learning is the process of assessing the performance of a trained model on a dataset, typically a held-out portion of the data not used during training (the test set). It's crucial for understanding how well a model generalizes to unseen data and whether it meets the desired performance criteria for a specific task. Evaluation helps compare different models, tune hyperparameters, and select the best model for deployment.

Underfitting and Overfitting

Overfitting and Underfitting: Evaluation helps identify these issues. Overfitting is when the model performs very well on training data but poorly on test data. Underfitting is when the model performs poorly on both.

Common Metrics for Evaluation

- Accuracy, Precision Recall, F1-Score, ROC-AUC for classification
- Mean Squared Error, Root Mean Squared Error, Mean Absolute Error for regression

Model Deployment & Integration

Model deployment is the process of integrating a trained machine learning model into a production environment, making it available for real-world use. It involves packaging the model, setting up the necessary infrastructure, and creating an interface for accessing its predictions. This often includes building APIs, web applications, or integrating the model into existing software systems. Monitoring and maintenance are crucial post-deployment to ensure continued performance and address potential issues. Ultimately, deployment bridges the gap between model development and practical application, delivering value from machine learning.

1

Scalability: The deployed model should be able to handle increasing volumes of requests or data without significant performance degradation.

2

Monitoring: Continuous monitoring of model performance, data drift, and other relevant metrics is essential to maintain accuracy and identify potential issues.



Additional resources

Videos

- [Andrew Ng Machine Learning Specialization](#)
- [Andrew Ng Deep Learning Specialization](#)
- [Krish Naik NLP Playlist](#)
- [C Playlist](#)
- [Data Structures and Algorithms Playlist](#)
- [Python](#)
- [OOP](#)
- [JavaScript](#)
- [React](#)
- [Django](#)

Documentations

- [Django](#)
 - [Tensorflow](#)
 - [Scikit-Learn](#)
 - [Pandas](#)
 - [Numpy](#)
 - [Matplotlib](#)
- 

Contact



<https://project-alexander.vercel.app/>



MichaelAveuc571@gmail.com



(+63) 970 745 1021

The background features a dark navy blue field. Overlaid on this are several concentric circles and three horizontal lines. The circles are thin and colored in a gradient: the innermost is a vibrant pink, followed by a bright cyan, and the outermost is a light lavender. The three horizontal lines are also thin and colored in the same gradient, with the top line being pink, the middle one cyan, and the bottom one lavender. The text 'Thank You' is centered in a bold, white, sans-serif font.

Thank You