ML DAY 2-3

09 June 2025 22:35

1. Transitioning from Regression to Classification

- Covers how to identify when a problem should be modeled using **classification** (predicting discrete labels) instead of **regression** (predicting continuous values).

2. Key Differences

- **Regression** predicts numeric values (e.g., house prices).
- **Classification** predicts discrete categories (e.g., email is spam or not).
- Introduces **binary classification** (two classes) and **multi-class classification** (more than two classes).

3. Data Preparation

- Shows how to convert a continuous target into **categorical classes**, such as binning age into groups or profit margins into "low/medium/high".

4. Choosing the Right Model

- Presents common classification algorithms:
    - **Logistic Regression**
    - **K-Nearest Neighbors (KNN)**
    - **Decision Trees**
    - **Support Vector Machines (SVM)**
- Highlights that regression models unlike logistic models aren't suitable for discrete class outputs.

5. Training & Model Evaluation

- Foreshadows splitting data into **train/test sets**, fitting classification models, and evaluating performancE

1. Introduction to the ML Lifecycle

- Explains the purpose: helps developers **navigate the complexities** of building ML solutions.

2. Key Stages of the Lifecycle

A. Understanding ML & Gathering Insights

- Defines ML as the ability for systems to **learn without explicit programming**.
- Highlights importance of grasping the foundational concept: learning from data.

B. Framing the Business Problem

- Emphasizes beginning with the right question.
- Must define business metrics and objectives *before* picking a model.

C. Data Collection & Understanding

- Gather relevant data.
- Use exploratory data analysis (EDA) to understand structure, patterns, and quality.

D. Feature Engineering (Preparation)

- Manipulate raw data to improve model input:
    - Create new meaningful features

    - Handle correlations and missing values
    - Encode categorical data through techniques like one-hot encoding (e.g., transforming 'ocean proximity' into binary flags)

E. Building the ML Pipeline

- Combine preprocessing, feature engineering, and model training into a **reproducible pipeline**.
- Helps maintain consistent transformations throughout experimentation and deployment.

F. Model Training & Evaluation

- Train models and evaluate them on key metrics:
    - Classification: accuracy, precision, recall, confusion matrix
    - Regression: RMSE, MAE
- Compare against baseline models.

G. Operationalizing the Pipeline

- Deploy the end-to-end pipeline into production:
    - Ensure it's **scalable, maintainable**, and version-controlled.
    - Include monitoring for data drift and retraining.