INDIVIDUAL TASK 3

Feature extraction through experiment

Introduction

To understand feature extraction, we have to look at the world the way a machine does: as a series of cold, hard numbers. A machine learning model doesn't "see" a shopping list or a photo; it sees a multi-dimensional matrix. Feature extraction is the art of transforming raw, messy data into these meaningful numerical representations.
For this thought experiment, we will analyze two vastly different datasets: Credit Card Transactions (Tabular Data) and Urban Street Photos (Image Data).

1. Dataset A: Credit Card Transactions (Fraud Detection)

The goal here is Classification: Is this specific transaction "Legitimate" or "Fraudulent"?

High-Value Features:

- *The "Velocity" Feature: How many transactions have occurred in the last 30 minutes? A sudden burst of 5 transactions is a high-signal feature for a stolen card.
- *Geographic Displacement: The distance between the current transaction and the previous one, divided by the time elapsed. If a card is swiped in New York and then 10 minutes later in London, the "Physical Impossibility" score is 100%.
- Merchant Category Consistency: Does this user normally shop at Steam and Whole Foods, but is now buying $4,000 worth of industrial timber? This is a "Categorical Deviation" feature.
- *Amount Rounding: Fraudsters often test cards with small, "clean" amounts (e.g., $1.00). A feature tracking the "Decimal Profile" can be surprisingly effective.

2. Dataset B: Urban Street Photos (Self-Driving Navigation)

The goal here is Object Detection and Semantic Segmentation. The model needs to know not just that there is a blob of pixels, but what that blob represents.
Low-Level Features (Edges and Textures)
At the earliest layers of a Convolutional Neural Network (CNN), the model extracts:

- *Edge Orientations: Vertical lines often represent buildings or poles; horizontal lines represent the horizon or lane markings.
- *Color Histograms: A large concentration of "Construction Orange" is a feature that suggests a high probability of a road hazard.

Mid-Level Features (Shape and Parts)
As we move deeper into the model, it combines edges into parts:

- *Circular Blobs: Two circular shapes separated by a horizontal bar are a strong feature for a "Bicycle" or "Car Tires."
- Aspect Ratio: A tall, thin rectangle is likely a pedestrian or a signpost; a wide, low rectangle is likely a vehicle.

High-Level Features (Contextual Relationships)

- Vanishing Point: The point where parallel lines converge. This feature tells the model where "Forward" is.
- Spatial Occupancy: Is the pixel "Above" or "Below" the horizon line? A car "above" the horizon is a flying car (noise/error), while a car "below" is a valid obstacle.

3. The Challenge: Dimensionality and "Noise"

In both datasets, more features aren't always better. This is known as the Curse of Dimensionality.

- Redundancy: If we include "Transaction Currency" and "Country of Origin," they often tell the model the same thing. This adds computational weight without adding intelligence.
- Overfitting: If a feature is too specific (e.g., "Transaction happened on a Tuesday when it was raining in Seattle"), the model might think rain causes fraud. It learns "noise" instead of "signal."

4. Conclusion: The "Golden Rule" of Features

A machine learning model is only as "smart" as the features we provide. In the shopping list example, the most important feature isn't the item itself, but the intent behind it.

Buying "Diapers + Beer" is a famous data science correlation for "New Fathers."

Buying "Acetone + High-Wattage Lights" might be a feature for "Industrial Manufacturing" (or something more illicit).

Feature extraction is about translating human context into mathematical significance.