**AI/ML Interns - Bhavya**

**Day - 3 Task**

**Learning Topics(Task):**

* Data preprocessing basics
* Handling missing data
* Encoding, normalization
* Train/Test split

**Data Processing Basics:**

Data processing in machine learning (ML) and artificial intelligence (AI) is like preparing ingredients before cooking a meal. For a computer to learn from data, it needs clean, organized information. Imagine teaching a robot to predict if someone will like a movie. You can’t just give it messy notes—you need to tidy up the data first.

1. **Collecting Data:** The initial phase involves data collection. Researchers aggregate pertinent information—variables such as age, genre preferences, and historical ratings, for example—methodically compiling datasets that reflect the domain of interest.
2. **Cleaning Data:** Following collection, the data frequently requires cleaning. This stage addresses incomplete entries, inaccuracies, or inconsistencies. If certain values (e.g., age) are missing, one may impute them using statistical measures like the mean. Similarly, standardization is essential; disparate labels such as “comedy” and “funny” are unified under a single category to maintain coherence. This step is akin to refining measurements or correcting anomalies in experimental data.
3. **Organizing Data:** Subsequently, the data must be organized into a format suitable for computational processing. Qualitative attributes are encoded numerically—gender might be translated to 0 or 1, genres transformed into binary columns—thus rendering the information machine-readable.
4. **Splitting Data:**The process continues with data splitting. For robust evaluation, the dataset is partitioned into training and testing subsets—commonly at an 80/20 ratio. This practice ensures that the model’s performance can be assessed objectively, mirroring the methodological rigor of dividing experimental trials into calibration and validation sets.
5. **Scaling Data:** Finally, scaling is implemented. Numerical variables are normalized to comparable ranges, preventing any single feature from disproportionately influencing the model due to differences in scale. This ensures that all variables contribute equitably to the learning process.

**Handling missing data**:

In the context of machine learning, missing data is analogous to encountering a recipe with incomplete instructions; critical information is absent, making it challenging for models to function optimally. Machine learning algorithms require comprehensive datasets to generate reliable predictions—for instance, forecasting whether an individual will purchase a house based on variables such as age, income, or occupation. When key data points are missing, the accuracy of these predictions can be compromised.

Several techniques have been developed to address the issue of missing data. Among the most commonly used are:

**Mean/Mode Imputation:** This method involves substituting missing numerical values with the mean—essentially, the arithmetic average—of the available data. For categorical variables, the mode, or most frequently occurring value, is used. For example, if age data is missing for 20 participants, one would calculate the average age (e.g., 35) and assign this value to all missing entries. Similarly, if “teacher” is the most common occupation, missing job data would be filled accordingly. The Titanic dataset frequently employs this approach: missing ages are replaced with the average age, and missing embarkation ports default to the most common (e.g., “Southampton”).

**Random Sample Imputation:** Rather than using a summary statistic, this approach fills missing values by randomly selecting from the observed data. For instance, missing ages might be filled by randomly choosing existing ages (25, 40, 32, etc.). This method preserves the original distribution and variability, reducing the risk of introducing bias. In the context of the Titanic data, if a passenger’s age is unknown, a randomly selected age from another passenger could be imputed.

**End of Tail Imputation:** Here, missing values are replaced with extreme values at the tail ends of the data distribution. For example, if the mean age is 35 and the maximum observed age is 70, a missing value might be imputed as 75. This technique is useful for marking missing data as potential outliers and is sometimes employed to flag anomalies in datasets such as the Titanic manifest, where missing ages might be set to 80.

**Constant Value Imputation:** This straightforward method uses a designated value—such as “Unknown” for categorical fields or 0 for numerical ones—to fill in missing entries. In the Titanic dataset, missing embarkation points might be assigned the value “Unknown,” clearly indicating the absence of information.

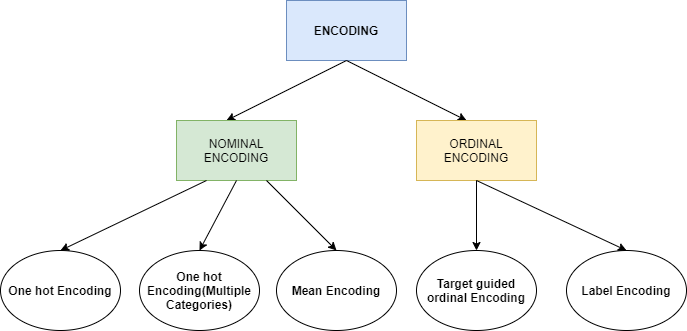
**Encoding and Normalization**:

In machine learning, preparing data is critical for effective model performance. Two foundational steps in this process are encoding (converting text data into numbers) and normalization (scaling numerical values). This summary outlines the key concepts and techniques involved.

**Encoding: Transforming Text Into Numbers**

Machine learning models require numerical input. Encoding handles the conversion of categorical (text) data into numerical format. Two common encoding methods are:

* **Label Encoding:**
  + Assigns a unique number to each category (e.g., “boy” = 0, “girl” = 1).
  + Practical for binary categories.
  + Caution: For multiple categories, assigned numbers may mistakenly imply order or ranking.
* **One-Hot Encoding:**
  + Creates a new binary column for each category (e.g., “Red,” “Blue,” “Green” for favorite color).
  + Each entry is marked with a 1 in its respective category column and 0 in others.
  + Prevents any implied hierarchy among categories.



**Normalization: Scaling Data for Fairness**

Raw numerical data can have differing ranges (e.g., age from 0-100, price from 1-50), which might bias the model. Normalization ensures all features contribute equally.

* **Min-Max Normalization:**
  + Scales all values to a range between 0 and 1.
  + Example: Age 50 out of 100 becomes 0.5; price $25 out of $50 also becomes 0.5.
* **Standardization:**
  + Adjusts data to have a mean of 0 and standard deviation of 1.
  + Centers features, ensuring no single variable dominates due to its scale.

**Train and Split:**

A train/test split is a fundamental process in machine learning. It involves dividing a dataset into two distinct parts: one for training a model and another for testing its performance. This approach ensures that the model can generalize its learning to new, unseen data, rather than simply memorizing patterns from the training set.

#### **Why Split the Data?**

* **Prevent Overfitting:**  
  By withholding a portion of the data for testing, we can check whether the model truly understands the patterns rather than just memorizing them.
* **Model Evaluation:**  
  The test set provides an objective means to evaluate the model’s predictive accuracy on data it has never encountered.

#### **The Process**

* **Training Set (Typically 80% of Data):**
  + Used to teach the model how features (e.g., age, game genre) relate to outcomes (e.g., game preferences).
  + For example, with 100 gamer profiles, 80 are used for the model’s learning process.
* **Test Set (Typically 20% of Data):**
  + Reserved for evaluating the model’s performance.
  + The model predicts outcomes for these 20 profiles, and results are compared to actual preferences to measure accuracy.

#### **Example: Game Preference Dataset**

* **Dataset Size:** 500 gamer records.
* **Split:**
  + **Training Set:** ~400 records for model training.
  + **Test Set:** ~100 records for performance evaluation.

This split allows the model to learn from the majority of the data while reserving a portion to objectively assess its ability to predict new instances.