(a) Rating of an Amazon product by a person on a scale of 1 to 5

- **Type: Ordinal**
- **Reason:** The numbers represent an order (5 is better than 4, etc.), but the differences between consecutive values are not guaranteed to be equal (the gap between "3" and "4" in a person's perception may not be the same as between "4" and "5").

(b) The Internet Speed

- **Type: Ratio**
- **Reason:** Internet speed (e.g., Mbps) is measured on a numeric scale with a **true zero** (0 Mbps means no connection). Ratios are meaningful (e.g., 20 Mbps is twice as fast as 10 Mbps).

(c) Number of customers in a store

- **Type: Ratio**
- **Reason:** The count of customers starts at a **true zero** (0 customers means none). Ratios are valid: 10 customers is twice as many as 5.

(d) UCF Student ID

- **Type: Nominal**
- **Reason:** Student IDs are identifiers. They are **labels only** and do not carry numeric meaning, order, or arithmetic interpretation.

(e) Distance

- **Type: Ratio**
- **Reason:** Distance has a **true zero** (0 distance means no separation). Ratios are meaningful (10 km is twice as far as 5 km).

(f) Letter grade (A, B, C, D)

- **Type: Ordinal**
- **Reason:** Letter grades have an inherent **order** (A > B > C > D), but the intervals are not uniform (the difference between an A and B is not necessarily equal to that between C and D).

(g) The temperature at Orlando

- **Type: Interval** (in °C or °F)

- **Reason:** Temperature has ordered values, and differences are meaningful (30°C − 20°C = 10°C). However, the zero point is **arbitrary** (0°C does not mean “no temperature”). Ratios are not meaningful (40°C is not “twice as hot” as 20°C).

https://github.com/HazelTChikara/titanic_assignment/tree/main/Documents/SCHOOL/DATA%20MINING/TITANIC%20Assignment

Improved Data Preprocessing and Model Accuracies

The preprocessing pipeline proposed in the Kaggle Titanic solution is a strong starting point, but it is **not the best solution** for maximizing classification performance. While the Kaggle workflow introduces useful engineered features such as *Title*, *FamilySize*, and *IsAlone*, it has several shortcomings:

1. **Leakage risk**: Statistics for imputing missing values and binning were often computed on the full dataset instead of within cross-validation folds.
2. **Information loss**: Hard binning continuous features like *Age* and *Fare* removes variability that linear and margin-based models can exploit.
3. **Discarded features**: Variables such as *Cabin* were dropped, despite the first letter (Deck) being informative.
4. **Limited interaction terms**: Relationships such as *Sex* × *Pclass* or *Fare per person* were not considered.
5. **Single hold-out split**: Reliance on one validation set leads to high variance in reported results.

Improved Preprocessing Pipeline

To address these limitations, I designed a **leakage-safe pipeline** using scikit-learn’s Pipeline and Column Transformer. All imputations, encodings, and rare-category groupings are learned inside cross-validation folds, preventing leakage. Key improvements:

- **Feature Engineering**
  - *Title* from Name (grouping rare titles).
  - *Deck* from Cabin initial.
  - *Ticket Prefix* and *GroupSizeByTicket* (captures group survival effects).
  - *Family Size* and *IsAlone*.

    - *FarePerPerson* = Fare ÷ FamilySize.
    - *AgeClass* = Age × Pclass.
    - Optional coarse bins (*AgeBin*, *FareBin*) to support tree models.
- **Data Handling**
    - Iterative imputers and median imputation per fold for numeric features.
    - Rare-category grouper for infrequent labels (mapped to "Rare").
    - One-hot encoding with handle_unknown='ignore'.
    - Scaling applied only for linear/SVM/KNN models, not for trees.
- **Evaluation**
    - **Stratified K-Fold CV (5 folds)** for fair and stable accuracy estimates.
    - Identical folds across models to ensure comparability.

Results: Improved Accuracies

After applying the improved pipeline and tuning models with GridSearchCV, the following accuracies were observed (mean ± std across 5 folds):