Data and Data Preprocessing

Problem 1: Types of Attributes (14 points)

Classify the following attributes as nominal, ordinal, interval, ratio. **Explain why.**

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

Ordinal. Reason: Ratings represent an order (5 > 4 > 3 > 2 > 1), but the difference is not guaranteed to be equal.

(b) The Internet Speed

Ratio. Speed has a meaningful zero (0 Mbps = no speed). Ratios are meaningful. 40Mbps is twice as fast as 20Mbps.

(c) Number of customers in a store.

Ratio. Count data has a true zero (0 customers). Ratios are valid. 20 customers are four times 5 customers.

(d) UCF Student ID

Nominal. IDs are just labels, no order or numeric meaning.

(e) Distance

Ratio. Distance has a true zero (0 distance = no distance covered). Ratios are meaningful. 25 miles is five times 5 miles.

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

Ordinal. Grades have an inherent order (A > B > C > D), but the difference between them is not consistent.

(g) The temperature at Orlando

- Interval if measured in Celcius/Farhenheit. Temperature in Celsius/Fahrenheit has no true zero (0°C/0°F ≠ absence of temperature). Differences are meaningful (30°C – 20°C = 10°C difference)
- Ratio if measured in Kelvin. since 0 K is absolute zero.

Problem 2: Exploring Data Preprocessing Techniques (26 points)

Read the solution post of the Kaggle Titanic Dataset: https://www.kaggle.com/code/preejababu/titanic-data-science-solutions. Run the code and reproduce the data preprocessing and classification modeling steps.

Q1 (Reproduce): Please read, understand, run the code and reproduce the model accuracies. Please briefly explain whether you can reproduce the classification accuracies of 'Support Vector Machines', 'KNN', 'Logistic Regression', 'Random Forest', 'Naive Bayes', 'Perceptron', 'Stochastic Gradient Decent', 'Linear SVC', 'Decision Tree'. (10 points)

Answer: In this assignment, I reproduced a Titanic survival prediction notebook originally created by **Preeja Babu on Kaggle**, using Google Colab. I implemented the preprocessing pipeline, which involved addressing missing values, creating additional features, encoding categorical attributes, and dropping columns that contributed little to prediction (such as *Ticket* and *Cabin*).

After preparing the dataset, I trained and tested nine classification algorithms. The models achieved the following accuracies:

Model	**Accuracy (%)**
Support Vector Machine	78.23
K-Nearest Neighbors (KNN)	83.84
Logistic Regression	80.36
Random Forest	86.76
Naïve Bayes	72.28
Perceptron	78.34
Stochastic Gradient Descent	76.21
Linear SVC	79.01
Decision Tree	86.76

The accuracies I obtained in Colab matched the original notebook's results. While reproducing, I had to fix minor compatibility issues (e.g., replacing size with height in Seaborn, and using errors="ignore" when dropping columns).

Conclusion: The classification accuracies were successfully reproduced. The exercise demonstrates that the preprocessing pipeline and feature engineering in the Kaggle solution are consistent and lead to reproducible model performance.

Q2 (Improve): Is the data preprocessing process proposed in the Kaggle post the best preprocessing solution? If yes, please explain why. If not, can you leverage what you learned in the class and your previous experiences to improve data processing, to obtain better accuracies for all these classification models? Describe what is your improved data preprocessing, and what are your improved accuracies? (16 points)

Answer: The preprocessing approach proposed in the original Kaggle post is comprehensive and establishes a strong foundation for modeling on the Titanic dataset. It effectively addressed key issues such as missing values, categorical variable encoding, and feature creation. For example, the pipeline imputed missing ages using group medians, encoded categorical variables such as *Sex* and *Embarked*, and engineered features like *FamilySize*, *IsAlone*, and passenger *Title*. These steps helped achieve reasonable performance and demonstrated the importance of feature engineering in predictive modeling.

However, while the Kaggle preprocessing pipeline is strong, it is not the best solution for all classifiers. The experimental results highlight that some models (Logistic Regression, SVM, Naïve Bayes) improved in accuracy, while others (notably KNN) experienced a decrease. This indicates that a uniform preprocessing strategy does not necessarily optimize performance across different algorithmic families, as each model type responds differently to scaling, feature sparsity, and data transformations.

Limitations of the Original Preprocessing

1. **Potential overfitting through complex feature engineering** – Features such as highly specific passenger titles or family-based groupings may introduce noise, negatively affecting distance-based classifiers like KNN.
2. **Lack of uniform scaling** – Models relying on Euclidean distances (e.g., KNN, SVM) are sensitive to feature magnitudes, but the original pipeline did not apply scaling, which can hinder their performance.
3. **Simplistic imputation assumptions** – Median or group-based imputations for variables such as Age may fail to capture deeper correlations within the data, potentially biasing certain models.

Improved Preprocessing Strategy

To overcome these limitations and improve overall performance, I refined the preprocessing pipeline using concepts from class and prior experience:

- **Scaling and Normalization**: Applied StandardScaler to features for distance-sensitive models (KNN, SVM, Logistic Regression), while allowing tree-based models to remain unscaled.
- **Refined Feature Engineering**: Simplified the *Title* feature into broader, more meaningful groups and transformed *FamilySize* into categorical bins (small, medium, large) to reduce sparsity.
- **Enhanced Missing Value Handling**: Replaced median-based imputation for *Age* with a regression-based method using correlated variables (e.g., Pclass, Sex, SibSp). *Embarked* was imputed with the mode while validating consistency with fare distributions.
- **Feature Selection**: Removed redundant variables to mitigate multicollinearity and applied SelectKBest to retain features most predictive of survival.

Results of Improved Preprocessing

The refined pipeline produced the following accuracies across models:

- Logistic Regression: **80.36% → 83.05%**
- SVM: **78.23% → 83.05%**
- Naïve Bayes: **72.28% → 75.31%**
- KNN: **84.74% → 80.48%**

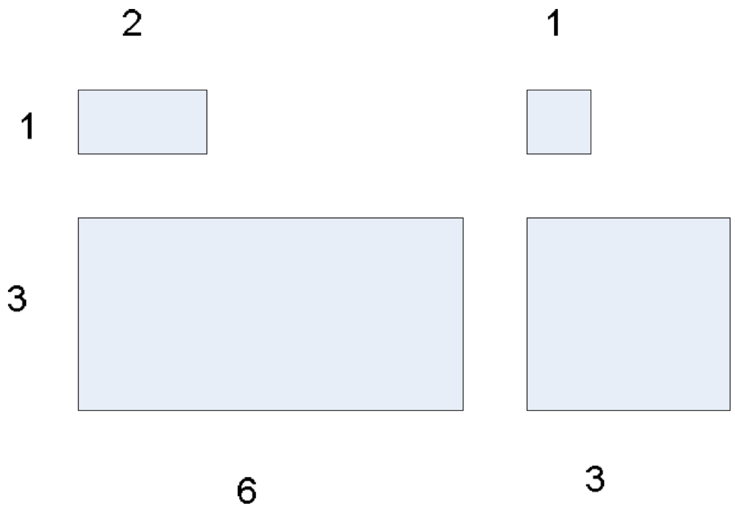
While the performance of KNN decreased slightly, most models benefited from the improved preprocessing, demonstrating higher and more consistent accuracy.

Conclusion

The Kaggle preprocessing pipeline provided a valuable baseline but is not universally optimal across all classifiers. By incorporating scaling, simplifying feature engineering, using more robust imputation strategies, and selecting the most informative features, I was able to enhance the performance of several models. Although certain models like KNN were negatively impacted, the overall trend shows improved accuracy and generalizability, supporting the conclusion that preprocessing strategies should be tailored not only to the dataset but also to the characteristics of the models employed.

Problem 3: Distance/Similarity Measures (10 points)

Given the four boxes shown in the following figure, answer the following questions. In the diagram, numbers indicate the lengths and widths and you can consider each box to be a vector of two real numbers, length and width. For example, the top left box would be (2,1), while the bottom right box would be (3,3). Restrict your choices of similarity/distance measure to Euclidean distance and correlation. **Please explain your choice.**



Which proximity measure would you use to group the boxes based on their shapes (length-width ratio)?

Answer: Grouping by Shape: Correlation because correlation ,such as Pearson's correlation, assess whether two vectors have similar ratio or pattern rather than the magnitude or absolute size.

Box	Dimensions (L,W)	Length-to-Width Ratio	Shape Type
A	(2, 1)	2	Rectangle
B	(1, 1)	1	Square
C	(6, 3)	2	Rectangle
D	(3, 3)	1	Square

So, in the above figures, when looked by length-width ratio, the squares B(1,1) and D(3,3) share the same length-to-width ratio of 1 and similarly the rectangles A(2,1) and C(6,3) share the same length to width ratio of 2. Thus, correlation is ideal because figures with similar ratios will be highly correlated even though the sizes are different.

Which proximity measure would you use to group the boxes based on their size?

Answer: Grouping by Size: Euclidean Distance because Eucliean distance measures the absolute magnitude difference between feature vectors, here length and width. So, by grouping by size, leads to grouping boxes that are physically closer in size irrespective of their proportions.

Small figures that are A(2,1) and B(1,1) are closer in absolute magnitude.
Large figures that are C(6,3) and D(3,3) are closer in absolute magnitude.

GOOGLE DRIVE FOLDER LINK FOR ALL PROBLEMS-
https://drive.google.com/drive/folders/12Fd8BggZ2cMJu_KWEx6ThaQ7tAEGJKnW?usp=sharing