**Problem 1: Types of Attributes (14 points)**

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

1. **Rating of an Amazon product by a person on a scale of 1 to 5**

Answer: Ordinal

Reason: The ratings (1, 2, 3, 4, 5) indicate order (5 is better than 4, 4 is better than 3, etc.), but the difference between values is not guaranteed to be uniform or meaningful (the gap between 1→2 may not feel the same as 4→5).

1. **The Internet Speed**

Answer: Ratio

Reason: Internet speed is measured in Mbps or Gbps. It has a true zero (0 Mbps means no connection), ratios are meaningful (100 Mbps is twice as fast as 50 Mbps)

1. **Number of customers in a store.**

Answer: Ratio

Reason: The number of customers is a count variable. It has a true zero (0 customers = empty store) and supports meaningful ratios (10 customers is twice as many as 5).

1. **UCF Student ID**

Answer: Nominal

Reason: A student ID is just a label for identification. It doesn’t indicate order, difference, or meaningful zero. It’s purely categorical.

1. **Distance**

Answer: Ratio

Reason: Distance has a true zero (0 means no distance) and ratios are meaningful (20 km is twice as far as 10 km).

1. **Letter grade (A, B, C, D)**

Answer: Ordinal

Reason: Grades reflect order (A > B > C > D), but the gaps between them are not uniform or quantifiable.

1. **The temperature at Orlando**

Answer: Interval

If in Celsius or Fahrenheit -> **Interval**

Explanation: These scales have arbitrary zero points (0°C ≠ absence of temperature, and negative values exist). Ratios are not meaningful (40°C is not “twice as hot” as 20°C). So, interval.

If in Kelvin -> **Ratio**

Explanation: Kelvin has an absolute zero (0 K = no thermal energy). Ratios are meaningful.

**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: I successfully executed the notebook and replicated both the preprocessing pipeline and model training procedures. The classification accuracies I achieved closely matched the results documented in the Kaggle post. Small differences of 1-2% were observed, which can be attributed to inherent randomness in data partitioning and model initialization processes. The overall reproduction was highly successful.

|  |  |
| --- | --- |
| **Model** | **Score (%)** |
| Random Forest | 86.76 |
| Decision Tree | 86.76 |
| KNN | 84.74 |
| Logistic Regression | 80.36 |
| Linear SVC | 78.90 |
| Perceptron | 78.34 |
| Support Vector Machines | 78.23 |
| Stochastic Gradient Descent | 75.87 |
| Naïve Bayes | 72.28 |

Link to Google Colab Notebook: <https://drive.google.com/file/d/1havoXO54YfoffR0gpxAqIgpmf7s-4GXz/view?usp=sharing>

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 in the Kaggle post is good but can be slightly modified to get better accuracies for few models. Here are my data preprocessing steps:-

1. **Title**: Extracted from Name (e.g., Mr, Miss, Mrs, Dr).
2. **Ticket features**:

* Ticket\_2letter: first two characters of the ticket string.
* Ticket\_len: number of characters in the ticket string.

1. **Cabin features**:

* Cabin\_num: number of cabins assigned (count of entries when splitting cabin string).
* Cabin\_letter: first letter of the cabin (deck indicator).

1. **Family features**:

* Fam\_size: total family size = SibSp + Parch + 1.
* Fam\_type: family size grouped into categories → Solo, Small, Big, Very big.

1. **Numerical pipeline**

* Impute missing numeric values (e.g., Fare) using the **median**.
* Apply **StandardScaler** to normalize numerical values.

1. **Categorical pipeline**

* Impute missing categorical values using the **most frequent** value.
* Apply **OneHotEncoder** (with handle\_unknown='ignore') to convert categories into dummy variables.

1. **ColumnTransformer**

* Combines both pipelines:
  + Numerical steps applied to Fare.
  + Categorical steps applied to Pclass, Title, Embarked, Fam\_type, Ticket\_len, Ticket\_2letter.

|  |  |  |
| --- | --- | --- |
| **Model** | **Old Score (%)** | **New Score (%)** |
| LinearSVC | 78.90 | 82.68 |
| Logistic Regression | 80.36 | 81.01 |
| Support Vector Machines | 78.23 | 81.01 |
| Stochastic Gradient Descent | 75.87 | 79.33 |
| Random Forest | 86.76 | 81.56 |
| Decision Tree | 86.76 | 79.89 |
| KNN | 84.74 | 81.01 |

Link to Google Colab Notebook: <https://drive.google.com/file/d/1tPLlBLMTjN9XKnmhnlX5AH2W9XWGmv3E/view?usp=sharing>

**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.**

**2 1**

1

3

6 3

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

Answer: Shape depends on proportionality (the ratio of length to width).

For (2,1), ratio is 2.0,

(6,3), ratio is 2.0

(1,1), ratio is 1.0

(3,3), ratio is 1.0

So (2,1) & (6,3) have the same shape (rectangles stretched twice as long) and (1,1) & (3,3) have the same shape (squares).

To capture similarity of ratios (ignoring scale), **Correlation** is the right choice. Correlation focuses on linear relationshipsand proportionality. It ignores magnitude and just sees whether the vectors have the same pattern.

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

Answer: Size depends on actual magnitudes of length and width.We want boxes with similar dimensions grouped together. (1,1) is much closer in size to (2,1) than to (6,3).Euclidean distance measures absolute closeness in terms of vector length.

Therefore, **Euclidean distance** is best for grouping by size.

We have 4 boxes as vectors: A = (2,1), B = (1,1), C = (6,3), D = (3,3) Euclidean distance formula between two points (x1, y1) and (x2, y2):

d = √(x1-x2)2 + (y1-y2)2

1. A–B = (2,1) vs (1,1)

√(2-1)2 + (1-1)2 =√12+02=√1=1.0

1. A–C = (2,1) vs (6,3)

√(2-6)2 + (1-3)2 ≈ 4.472

1. A–D = (2,1) vs (3,3)

√(2-3)2 + (1-3)2 ≈ 2.236

1. B–C = (1,1) vs (6,3)

√(1-6)2 + (1-3)2 ≈ 5.385

1. B-D = (1,1) vs (3,3)

√(1-3)2 + (1-3)2 ≈ 2.828

1. C-D = (6,3) vs (3,3)

√(6-3)2 + (3-3)2 = 3.0