Report on Handling Text and Categorical Attributes

Processing Categorical Attributes

In our dataset, categorical attributes were identified and processed to ensure compatibility with machine learning models. Since most models require numerical inputs, we transformed these categorical features into a numerical format using appropriate encoding techniques.

Encoding Techniques Used

- **One-Hot Encoding (OHE):**
    - Applied to categorical attributes with a small number of unique categories.
    - Justification: One-hot encoding prevents an ordinal relationship assumption in nominal data and allows models to interpret each category independently.
- **Label Encoding:**
    - Used for categorical variables with an inherent order.
    - Justification: Label encoding is suitable for ordinal categories where a ranking exists (e.g., low, medium, high).

Handling Missing Categorical Values

- **Mode Imputation:**
    - If a categorical variable had missing values, they were replaced with the most frequent category (mode).
    - Justification: This approach ensures minimal impact on data distribution.
- **Separate Category ('Unknown'):**
    - For features where missing values had significant representation, a new category labeled 'Unknown' was introduced.
    - Justification: Preserves missing information rather than imputing an artificial value.

Handling Text Attributes

Preprocessing Steps

To ensure the text data was clean and suitable for numerical conversion, we performed the following preprocessing steps:

- **Text Cleaning:** Removed special characters, numbers, and excessive whitespace.
- **Lowercasing:** Converted all text to lowercase for uniformity.
- **Tokenization:** Split text into individual words or tokens.
- **Stopword Removal:** Eliminated common stopwords (e.g., "the," "and," "is") to focus on meaningful words.
- **Stemming/Lemmatization:** Reduced words to their root form (e.g., "running" → "run").

Text to Numerical Representation

- **TF-IDF (Term Frequency-Inverse Document Frequency):**
    - Used to convert text into numerical vectors based on word frequency while considering the importance of words across the dataset.
    - Justification: TF-IDF helps reduce the impact of common words and enhances meaningful text representation.
- **Word Embeddings (Optional for Advanced NLP Models):**
    - Considered for deep learning models if necessary.
    - Justification: Word embeddings capture contextual relationships between words better than traditional frequency-based methods.

Challenges and Solutions

1. **High Dimensionality in One-Hot Encoding:**
    - Solution: Used feature selection techniques or dimensionality reduction (e.g., PCA) to manage high-cardinality categorical variables.
2. **Handling Imbalanced Categorical Features:**
    - Solution: Merged infrequent categories into an "Other" category to avoid sparsity issues.
3. **Processing Large Text Data Efficiently:**
    - Solution: Used TF-IDF with n-grams to capture context while keeping computational costs reasonable.

Conclusion

By implementing appropriate encoding techniques for categorical data and transforming text into meaningful numerical representations, we ensured our dataset was optimized for machine learning models. These preprocessing steps enhanced model interpretability and performance while addressing common challenges in handling categorical and textual data.