Executive Summary

Label encoding is the recommended and simplest approach for converting your 6-class target variable into numeric form, assigning each class an integer while preserving the classification task structure. Other encoding methods like one-hot or binary encoding are unsuitable for target variables because they transform the problem into multi-label or bit-wise tasks, increase output dimensionality, and risk unintended relationships. Implementing a split-first pipeline ensures that encoding is learned solely from the training data, preventing leakage and delivering reliable, production-ready evaluations.

---

1. Label Encoding for Target Variables

What It Is

Label encoding assigns each unique target class a distinct integer value (e.g., DDoS → 0, DoS → 1, …, MITM → 5) using transformers like `LabelEncoder` from scikit-learn.

Why Use It

- **Native Support and Simplicity:** `LabelEncoder` is explicitly designed for target transformation, mapping classes to an integer range without expanding the output dimension.
- **Compatibility with Loss Functions:** Many classifiers and loss functions (e.g., sparse categorical cross-entropy) operate directly on integer labels, avoiding the need for vector outputs.
- **Efficient Representation:** Compared to one-hot, label encoding uses a single output column, reducing memory and computation overhead.

---

2. Why Other Encoding Techniques Are Unsuitable

2.1 One-Hot Encoding

One-hot encoding on the target transforms single-class labels into binary vectors, effectively converting classification into a multi-label problem. This increases complexity, data sparsity, and

model output size, often degrading performance on limited data. (we still didn't finally decided are we going to use it or no)

2.2 Binary Encoding

Binary encoding maps labels to binary digits across multiple columns, which conflicts with standard single-output classifiers and can confuse the model by introducing bit-level relationships that don’t exist among classes.

2.3 Feature-Focused Encodings

Methods like target encoding and frequency encoding are intended for features and incorporate target statistics into the encoding process, risking leakage if misapplied to the target and offering no benefit for label transformation.

---

3. Split-First Encoding Pipeline

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier

# 1. Split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 2. Fit LabelEncoder on training labels only
encoder = LabelEncoder().fit(y_train)

# 3. Transform both sets
y_train_enc = encoder.transform(y_train)
y_test_enc  = encoder.transform(y_test)

# 4. Build and evaluate pipeline
pipeline = Pipeline([
    ('clf', RandomForestClassifier())
])
pipeline.fit(X_train, y_train_enc)
print("Test Accuracy:", pipeline.score(X_test, y_test_enc))
```

This workflow guarantees that encoding mappings are learned exclusively on the training data, preventing any insight from the test labels from influencing the model.

---

4. Conclusion

- **Use Label Encoding** for target variables to maintain a clear, integer-based representation that aligns with most classifiers.
- **Avoid One-Hot and Binary Encoding** for the target to prevent task alteration and unnecessary output expansion.
- **Always Split First** to eliminate data leakage, ensuring credible performance metrics and reproducible pipelines.

References

- LabelEncoder — scikit-learn 1.6.1 documentation
- Why should LabelEncoder from sklearn be used only for the target variable? — Stack Overflow
- Difference between One-Hot Encoding and Label Encoding of target output (LabelEncoder vs OHE) — Stack Overflow
- Types of Categorical Data Encoding — Analytics Vidhya
- Target encoding in test data and target leakage — CrossValidated (Stats.SE)
- Preventing data leakage during data preparation — Machine Learning Mastery
- One-Hot Encoding vs Label Encoding in Machine Learning — Analytics Vidhya
- Label Encoding in Python | GeeksforGeeks
- What to do when the target (Y value) is not numerical? — Kaggle Discussions
- Label Encoding vs One Hot Encoding — Medium article