exp 6

Experiment Number: 6

AIM: To explore and apply different encoding techniques (Label Encoding, One-Hot Encoding, Binary Encoding) to convert categorical variables into quantitative variables in Python, using the Titanic dataset as an example.

Objectives:

- To understand the importance of encoding categorical variables in data analysis and machine learning.
- To learn about different encoding methods: Label Encoding, One-Hot Encoding, and Binary Encoding.
- To apply these methods to the Titanic dataset using Scikit-learn and category_encoders libraries.
- To analyze and compare the different encoded representations of categorical variables.

Course Outcomes: CO3, CO4 (These likely relate to data preprocessing, feature engineering, and data transformation in your course)

Resource/Tools: Jupyter Notebook/Python with Pandas, Scikit-learn, and category_encoders libraries.

Theory:

Many machine learning algorithms are designed to work with numerical data. Categorical variables, which represent data in categories or groups, need to be converted into a numerical format before they can be used in these algorithms. This process is called encoding.

Encoding Techniques:

- **Label Encoding:** Assigns a unique integer to each category in a categorical variable. This is suitable for ordinal variables where the categories have a meaningful order (e.g., "low", "medium", "high").
- **One-Hot Encoding:** Creates new binary (0 or 1) variables for each category in a categorical variable. This is useful for nominal variables where there is no inherent order among the categories (e.g., colors, cities).
- **Binary Encoding:** Represents each category with a unique binary code. This is a combination of label encoding and one-hot encoding and can be useful for high-cardinality categorical variables (variables with many unique categories).

Algorithm:

1. **Load the dataset:** Import the Titanic dataset using Pandas.
2. **Identify categorical variables:** Determine the categorical columns in the dataset (e.g., `Sex`, `Embarked`).
3. **Apply Label Encoding:** Use `LabelEncoder` to transform ordinal categorical variables into numerical labels.
4. **Apply One-Hot Encoding:** Use `OneHotEncoder` to create binary variables for nominal categorical features.
5. **Apply Binary Encoding:** Use `BinaryEncoder` from the `category_encoders` library to encode categorical features into binary codes.
6. **Analyze and compare:** Examine the encoded data to understand the different representations.

Code:

Python

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
import category_encoders as ce

# Load the dataset
data = pd.read_csv('titanic.csv')

# Display information about the dataset
print("First 5 rows of the dataset:")
print(data.head())
print("\nColumn information:")
print(data.info())

# Select categorical features for encoding
categorical_features = ['Sex', 'Embarked']

# 1. Label Encoding
label_encoder = LabelEncoder()
for feature in categorical_features:
    data[feature + '_label'] = label_encoder.fit_transform(data[feature])
    print(f"\nUnique values of {feature}: {data[feature].unique()}")
    print(f"Unique values of {feature}_label: {data[feature +
'_label'].unique()}")

# 2. One-Hot Encoding
onehot_encoder = OneHotEncoder(sparse_output=False, handle_unknown='ignore')
encoded_data = onehot_encoder.fit_transform(data[categorical_features])
```

```
encoded_df = pd.DataFrame(encoded_data,

columns=onehot_encoder.get_feature_names_out(categorical_features))
data = pd.concat([data, encoded_df], axis=1)
print("\nOne-hot encoded columns:\n", encoded_df.head())

# 3. Binary Encoding
binary_encoder = ce.BinaryEncoder(cols=categorical_features)
data = binary_encoder.fit_transform(data)
print("\nBinary encoded columns:\n", data[['Sex_0', 'Sex_1', 'Embarked_0',
'Embarked_1']].head())

# Display the encoded data
print("\nEncoded data:")
print(data.head())
```

Output:

The code will output:

- The first 5 rows and column information of the Titanic dataset.
- The unique values of the original and label-encoded categorical features.
- The one-hot encoded columns.
- The binary encoded columns.
- The first 5 rows of the dataset with all the encoded columns.

Result:

The experiment successfully demonstrates the application of different encoding techniques to convert categorical variables into numerical representations. The output shows how each method transforms the categorical data, allowing you to compare and understand their effects.

Learning Outcomes:

- Gained practical experience in applying various encoding techniques for categorical variables.
- Understood the differences between Label Encoding, One-Hot Encoding, and Binary Encoding.
- Developed skills in using Scikit-learn and category_encoders for data preprocessing.
- Enhanced understanding of how to prepare data for machine learning algorithms by encoding categorical features.