Feature Engineering in Machine Learning

Ayush Singh[1]

Antern Department of Artificial Intelligence
ayush@antern.co

Abstract. This document contains contents on data preparation in machine learning and we also cover several components of data preparation like feature engineering, feature selection, dimensionality reduction, etc. We provide readers with several traditional and modern techniques to handle complicated data tasks.

Key words: data preparation, feature engineering, feature selection, data cleansing, data transformation, dimensionality reduction

0.1 One Hot Encoding

For each distinct category in a nominal categorical variable, binary (0/1) features are created using the one-hot encoding technique. According to this method, a new binary column is created for each distinct category, with the existence of the category in an observation being represented by 1 and the absence by 0.

Advantages and Disadvantages:

Advantages	**Disadvantages**
Interpretability: One-hot encoding generates a binary feature for each category, making the connections between categories and the target variable simple to understand. No Artificial Order: One-hot encoding is useful for nominal categorical variables because, unlike label encoding, it does not impose an artificial order on the categories. Disadvantages:.	Increased Dimensionality: When the categorical variable has a large number of distinct categories, one-hot encoding can greatly increase the dimensionality of the dataset. This may result in the "curse of dimensionality" and increasing computational complexity.

Table 1. Advantages and disadvantages of one-hot encoding

Worked Example: Consider a dataset with the variable 'Animal' representing different animal species:

Animal
Dog
Cat
Elephant
Dog
Elephant

Using one-hot encoding, we create a new binary column for each unique category:

Dog	Cat	Elephant
1	0	0
0	1	0
0	0	1
1	0	0
0	0	1

To perform one-hot encoding in Python, you can use the get_dummies function from the pandas library:

```
import pandas as pd

# Create a sample dataset
data = {'Animal': ['Dog', 'Cat', 'Elephant', 'Dog', 'Elephant']}
df = pd.DataFrame(data)

# Apply one-hot encoding to the 'Animal' column
encoded_df = pd.get_dummies(df, columns=['Animal'])

# Display the encoded dataset
print(encoded_df)
```

Animal_Cat	Animal_Dog	Animal_Elephant
0	1	0
1	0	0
0	0	1
0	1	0
0	0	1

Table 2. Output