Feature Engineering Assignment - 3

February 13, 2024

Q1. What is data encoding? How is it useful in data science?

Data encoding is the process of transforming data from one format or representation to another. This is done to ensure that the data is compatible with a particular system, application, or communication protocol.

In data science, data encoding plays an important role in data preprocessing, which is a crucial step in preparing data for analysis.

Machine learning algorithms typically require numerical data, but many datasets contain categorical or textual data. Encoding techniques can be used to transform categorical or textual data into numerical data, making it suitable for machine learning.

Q2. What is nominal encoding? Provide an example of how you would use it in a real-world scenario.

Nominal encoding, also known as one-hot encoding, is a technique used to transform categorical data into numerical data. In nominal encoding, each unique category value is assigned a binary value, with one binary feature being created for each category value.

For example, suppose we have a dataset of customer purchases, and one of the categorical features is the payment method used for the purchase, with three possible values: cash, credit card, and debit card. To use this data in a machine learning algorithm, we need to encode this feature numerically. We can use nominal encoding to create three new binary features, one for each payment method, as follows:

[1]:
```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
df = pd.DataFrame({'Payment_Method': ['Cash', 'Credit Card', 'Debit Card']
})
df.head()
```

[1]:
```
  Payment_Method
0           Cash
1    Credit Card
2     Debit Card
```

[2]:
```
encoder = OneHotEncoder()
encoded = encoder.fit_transform(df[['Payment_Method']])
encoded_df = pd.DataFrame(encoded.toarray(), columns=encoder.
 ↪get_feature_names_out())
```

[2]:

	Payment_Method_Cash	Payment_Method_Credit Card	Payment_Method_Debit Card
0	1.0	0.0	0.0
1	0.0	1.0	0.0
2	0.0	0.0	1.0

[3]:
```
new_df = pd.concat([df, encoded_df], axis=1)
new_df
```

[3]:

	Payment_Method	Payment_Method_Cash	Payment_Method_Credit Card	Payment_Method_Debit Card
0	Cash	1.0	0.0	0.0
1	Credit Card	0.0	1.0	0.0
2	Debit Card	0.0	0.0	1.0

Q3. In what situations is nominal encoding preferred over one-hot encoding? Provide a practical example.

Nominal encoding and one-hot encoding are actually the same thing, and the terms are often used interchangeably. One-hot encoding is a type of nominal encoding where each category value is assigned a binary value, and it is the most commonly used nominal encoding technique in data science.

However, there is another type of nominal encoding called label encoding, where each unique category value is assigned a numerical label. Label encoding can be useful in situations where the categorical values have an inherent order or ranking, such as rating scales or levels of education.

For example, in a dataset of job applicants, we might have a feature for the level of education, with values such as high school, bachelor's degree, and master's degree. We could use label encoding to assign numerical labels to each of these values, with high school as 1, bachelor's degree as 2, and master's degree as 3. This would allow us to preserve the inherent order of the values while still transforming them into numerical data for use in machine learning algorithms.

Q4. Suppose you have a dataset containing categorical data with 5 unique values. Which encoding technique would you use to transform this data into a format suitable for machine learning algorithms? Explain why you made this choice.

If we have a dataset containing categorical data with 5 unique values, we could use nominal encoding techniques such as one-hot encoding to transform this data into a format suitable for machine learning algorithms. In one-hot encoding, we would create 5 new binary features, one for each unique category value, and assign a value of 1 to the corresponding feature for each data point.

The reason why we would choose one-hot encoding in this scenario is that nominal encoding techniques such as one-hot encoding are preferred for categorical data because they can accurately represent the categorical data in numerical form without creating false relationships between categories. Other encoding techniques, such as label encoding, can create false relationships between categories by assigning numerical labels that imply an order or ranking to the categories.

Q5. In a machine learning project, you have a dataset with 1000 rows and 5 columns. Two of the columns are categorical, and the remaining three columns are numerical. If you were to use nominal encoding to transform the categorical data, how many new columns would be created? Show your calculations.

If we use nominal encoding to transform the two categorical columns in the dataset, we would create new binary features for each unique category value in each column. The number of new binary features created for each column would depend on the number of unique category values in each column.

Let's assume that the first categorical column has 4 unique category values, and the second categorical column has 6 unique category values. To perform one-hot encoding on these columns, we would create 4 new binary features for the first column (one for each unique category value), and 6 new binary features for the second column (again, one for each unique category value). Each row in the original dataset would then be represented by the original three numerical columns, as well as the 4 binary features for the first categorical column and the 6 binary features for the second categorical column.

Therefore, the total number of new columns created through one-hot encoding would be: $4 + 6 + 3 = 13$. So, we would have 13 columns in the transformed dataset after nominal encoding.

Q6. You are working with a dataset containing information about different types of animals, including their species, habitat, and diet. Which encoding technique would you use to transform the categorical data into a format suitable for machine learning algorithms? Justify your answer.

For transforming the categorical data in the animal dataset, I would use nominal encoding techniques, such as one-hot encoding. This is because nominal encoding techniques are preferred for categorical data since they can accurately represent the categorical data in numerical form without creating false relationships between categories.

In the animal dataset, we have categorical variables such as species, habitat, and diet. One-hot encoding would be a suitable technique for encoding these variables. For example, we could create binary features for each unique value in the species variable, such as lion, tiger, and leopard. Similarly, we could create binary features for each unique value in the habitat and diet variables, such as forest, grassland, and carnivorous.

Q7. You are working on a project that involves predicting customer churn for a telecommunications company. You have a dataset with 5 features, including the customer's gender, age, contract type, monthly charges, and tenure. Which encoding technique(s) would you use to transform the categorical data into numerical data? Provide a step-by-step explanation of how you would implement the encoding.

For transforming the categorical data in the customer churn dataset into numerical data, I would use nominal encoding techniques, such as one-hot encoding, since it is one of the most commonly used techniques for encoding categorical data. Here is how I would implement the encoding step-by-step: 1. Identify the categorical variables in the dataset. In this case, the only categorical variable is the customer's gender. 2. Apply one-hot encoding to the categorical variable. This involves creating a new binary feature for each unique category value in the gender variable (i.e., male and female). We can achieve this by using the get_dummies() function in Python's Pandas library. This function creates new binary columns for each unique category value and assigns a value of 1 to the corresponding column for each data point. 3. Drop the original categorical variable (gender)

from the dataset. We no longer need this variable since we have already encoded it using one-hot encoding. 4. The remaining four features (age, contract type, monthly charges, and tenure) are numerical and do not require any encoding.

[ ]: