00:00-08:22

Feature Engineering and Dimensionality Reduction Techniques

In this article, we will explore various feature engineering and dimensionality reduction techniques to enhance our data analysis process. We will begin by examining the steps involved in creating new features and then delve into reducing the dimensions of our dataset. These techniques will aid us in outlier detection and removal, ensuring the reliability of our analysis.

Creating the Price per Square Feet Feature

To start, let's consider a real estate dataset where we have a data frame with various columns. One important feature in the real estate market is the price per square feet, which can be useful for outlier detection. We will create a new column, "Price per Square Feet," by dividing the price column by the square foot area column. This will provide us with a valuable feature for later stages of analysis.

```
df5'Price per Square Feet' = df5'Price' /
df5'Square Foot Area'
```

With the addition of the new feature, our data frame now includes the "Price per Square Feet" column. This column will contribute to identifying and addressing outliers in subsequent steps.

Exploring the Location Feature

Next, let's focus on the location column in our dataset. It is a categorical feature that contains information about the various locations. We aim to determine the number of unique locations and the corresponding number of rows in our dataset.

```
unique_locations = df5'Location'.unique()
num_locations = len(unique_locations)
num_rows = df5.shape0
```

After executing the above code, we find that we have a large number of locations, with a total of 1,300 unique locations in our dataset. However, dealing with such a large number of locations can pose challenges, specifically regarding the dimensionality of the data.

Addressing the Dimensionality Curse

High dimensionality, also known as the "dimensionality curse," can negatively impact our data analysis. When we convert categorical features like location into dummy columns using one-hot encoding, we end up with an overwhelming number of columns. In this case, we would have 1,300 columns, which is impractical and can lead to computational inefficiencies.

To combat this issue, we can introduce the concept of an "other" category. This category encompasses locations that have a limited number of data points, such as only one or two instances. By identifying these locations, we can reduce the dimensions of our dataset effectively.

To begin, we need to determine the number of data points available for each location. Let's clean the location column by removing any extra spaces using a lambda function:

```
df5'Location' = df5'Location'.apply(lambda x:
x.strip())
```

Now, we can calculate the statistics on location by grouping the data frame based on the location column and aggregating the count for each location. This will provide us with the number of data points available for each location.

```
location_stats =
df5.groupby('Location')'Location'.count()
```

Sorting the resulting statistics in descending order will reveal the locations with the maximum number of data points. We can use this information to establish a threshold, such as considering locations with less than ten data points as "other" locations.

```
location_stats =
location_stats.sort_values(ascending=False)
threshold = 10
other_locations = location_statslocation_stats <
threshold
```

By applying the condition mentioned above, we find that there are 1,052 locations out of the original 1,293 that have less than ten data points. These locations will be consolidated into a general category called "other."

```
df5'Location' = df5'Location'.apply(lambda x:
'other' if x in other_locations else x)
```

After transforming the data frame, we observe that the number of unique locations reduces to 242. This reduction in dimensions will significantly aid us in subsequent data analysis steps.

Conclusion

In this article, we covered essential techniques for feature engineering and dimensionality reduction. We explored the creation of the "Price per Square Feet" feature, which facilitates outlier detection. Additionally, we addressed the issue of high dimensionality by introducing an "other" category for locations with a limited number of data points. By implementing these techniques, we successfully reduced the dimensionality of our dataset, enhancing the efficiency and accuracy of our analysis. In the next article, we will dive into outlier detection and removal, further refining our data analysis process.