CCA-4: Feature Scaling and Normalization

- Overview
- Objectives
- Steps
  - Feature Scaling
  - Normalization
- Need for Scaling and Normalization
- Results Obtained
- Next Steps
- Conclusion

Overview

This section covers the process of feature scaling and normalization. These steps are crucial to prepare the dataset for modeling by ensuring consistency across features and improving the performance of machine learning algorithms.

Objectives

- Apply scaling techniques such as Min-Max Scaling or Standard Scaling.
- Normalize the data to maintain consistency across features.

Steps

Feature Scaling

In feature scaling, the features are re-scaled into a specified range, usually [0, 1] in Min-Max Scaling. This enables consistency between the features.

Code Snippet

```python
import os
import pandas as pd
import json
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Helper function to find the project root directory
def find_project_root(filename='config.json'):
    current_dir = os.path.abspath('')
    while True:
        if filename in os.listdir(current_dir):
            return current_dir
        parent_dir = os.path.dirname(current_dir)
        if parent_dir == current_dir:
            raise FileNotFoundError(f"{filename} not found in any parent directories.")
        current_dir = parent_dir

# Find the project root directory
root_dir = find_project_root()
```

```python

# Load configuration
config_path = os.path.join(root_dir, 'config.json')
with open(config_path, 'r') as f:
    config = json.load(f)

# Load the cleaned dataset using the dynamic path from the config file
cleaned_data_path = os.path.join(root_dir, config['processed_data_path'])
df = pd.read_csv(cleaned_data_path)
print("Cleaned dataset loaded successfully.")
print(df.head())

# Apply Min-Max Scaling
min_max_scaler = MinMaxScaler()
scaled_df = pd.DataFrame(min_max_scaler.fit_transform(df), columns=df.columns)
print("Min-Max Scaling applied successfully.")
print(scaled_df.head())
```

Explanation

- **Dynamic Path Configuration**: Ensures the script can dynamically find the project's root directory.
- **Loading the Cleaned Dataset**: The cleaned dataset is loaded using the path specified in the `config.json` file.
- **Applying Min-Max Scaling**: The `MinMaxScaler` from `sklearn.preprocessing` is used to rescale the features to a range of [0, 1].

Normalization

In this process, normalization of the numeric features is the adjustment of values of a feature such that the mean will be zero with standard deviation one. This will be beneficial when working with a lot of machine learning algorithms to achieve better results.

Code Snippet:

```python
# Apply Standard Scaling
standard_scaler = StandardScaler()
normalized_df = pd.DataFrame(standard_scaler.fit_transform(df), columns=df.columns)
print("Standard Scaling applied successfully.")
print(normalized_df.head())

```

Explanation

- **Applying Standard Scaling**: The `StandardScaler` from `sklearn.preprocessing` is used to normalize the features, ensuring they have a mean of 0 and a standard deviation of 1.

Need for Scaling and Normalization

1. **Model Performance:** Better performance, as many machine learning algorithms are sensitive to the scale of the input features.
2. **Consistency:** Helps maintain the scaling of features to be consistent, very important in proper model training and evaluation.
3. **Avoids Bias:** It avoids domination by those larger-scaled features in the modeling process.

Results Obtained

These were the results obtained after performing feature scaling and normalization.

1. **Min-Max Scaling Applied:** All features were rescaled to fall within the range [0, 1].
2. **Applied Standard Scaling:** Done scaling and centering all features to a mean of 0 and standard deviation of 1.

3. **Data Ready for Modeling:** The dataset is now at scale and has been normalized; it will be good to go into the process of modeling.

Next Steps

Now that data is scaled and normalized, the next steps are as follows:

1. **Exploratory Data Analysis:** Descriptive exploratory data analysis is to be carried out on the scaled and normalized dataset.
2. **Data Visualization:** Visualize the distribution of data and patterns or anomalies.
3. **Feature Engineering:** Create new features that increase the predictability of models, transform existing features to more representatively represent the data.
4. **Data Splitting:** Separate the scaled and normalized dataset into training and testing sets to be used in the evaluation of the model.
5. **Model training and evaluation:** Through the application of the training set for model learning, one can perform predictive modeling and, at last, test performance using the testing set.
6. **Reporting:** Develop reports and visualizations to clearly present the findings from the analysis and modeling.

When all these steps are carried out, we have a full data pipeline that retains data quality and enhances successful data analysis and modeling.

Conclusion

The process of feature scaling and normalization is quite important while doing data engineering. This task is done to ensure that the data set remains homogeneous and free from redundancy, hence it is further used in the analysis and modeling process. In this documentation, a clear and detailed explanation of how to perform these tasks is given, making sure everything is well described for the process to be reproducible and understandable.