Pandas: Handling Categorical Data

Farivar Zarvandeh

July 27, 2023

1 Introduction

In this document, we will explore various methods in pandas for handling and working with categorical data in a Series. Categorical data represents data that falls into specific categories, and pandas provides powerful tools to manipulate and analyze such data efficiently.

2 List of Methods

1. `pd.Categorical()`

   ```
   pd.Categorical(data, categories, ordered, dtype)
   ```

   Convert data into a pandas Categorical data type. This method is used to create a categorical variable from existing data. It is helpful for memory optimization and represents data with a fixed set of categories.

2. `Series.astype()`

   ```
   Series.astype(dtype)
   ```

   Convert the data type of the Series to the specified data type. You can use this method to convert a Series to a Categorical data type.

3. `Series.cat`
   A property of the Series that provides access to categorical methods.

4. `Series.cat.codes`
   Return the category codes of each element in the Series. It returns an integer array representing the codes of the categories.

5. `Series.cat.categories`
   Return the categories of the categorical data. It returns an Index containing the unique categories.

6. `Series.cat.ordered`
   Return a boolean indicating whether the categories have an ordered relationship.

7. `Series.cat.add_categories()`

   ```
   Series.cat.add_categories(new_categories)
   ```

   Add new categories to the existing categorical data.

8. `Series.cat.remove_categories()`

   ```
   Series.cat.remove_categories(removals)
   ```

   Remove specific categories from the existing categorical data.

9. `Series.cat.reorder_categories()`

   ```
   Series.cat.reorder_categories(new_categories, ordered)
   ```

   Reorder the categories of the categorical data. It can also change the ordering if the 'ordered' parameter is provided.

10. `Series.cat.set_categories()`

    ```
    Series.cat.set_categories(new_categories, ordered)
    ```

    Set the categories of the categorical data. It can also set the ordering if the 'ordered' parameter is provided.

11. `Series.cat.rename_categories()`

    ```
    Series.cat.rename_categories(new_categories)
    ```

    Rename the categories of the categorical data.

12. `Series.cat.as_ordered()`
    Set the categorical data as ordered.

13. `Series.cat.as_unordered()`
    Set the categorical data as unordered.

14. `Series.cat.remove_unused_categories()`
    Remove categories that are not present in the Series.

15. `Series.cat.set_ordered()`
    Deprecated method to set categorical data as ordered.

16. `Series.cat.set_categories()`
    Deprecated method to set the categories of the categorical data.