Unnecessary Iteration

Context

Loops are typically time-consuming and verbose, while developers can usually use some vectorized solutions to replace the loops.

Problem

As stated in the Pandas documentation: “Iterating through pandas objects is generally slow. In many cases, iterating manually over the rows is not needed and can be avoided”. In `EffectiveTensorflow` github repository, it is also stated that the slicing operation with loops in TensorFlow is slow, and there is a substitute for better performance.

Please, consider that this problem is currently denoted for Pandas library when iterating through Pandas objects, and Tensorflow library for slicing operations. This is not applicable to other libraries where the the iteration can not be replaced with a specific library API call.

Solution

Pandas’ **built-in methods (e.g., join, groupby) are vectorized.**
It is therefore recommended to **use Pandas built-in methods as an alternative to loops**.

In TensorFlow, **using the `tf.reduce_sum()` API to perform reduction operation is much faster than combining slicing operation and loops.**

Existing Stage	**Effect**
Data Cleaning	Efficiency

Examples at next page.

Example

Python

```python
### Pandas
import pandas as pd
df = pd.DataFrame([1, 2, 3])
- result = []
- for index, row in df.iterrows():
-     result.append(row[0] + 1)
- result = pd.DataFrame(result)
+ result = df.add(1)
### TensorFlow 2
import tensorflow as tf
x = tf.random.uniform([500, 10])
- z = tf.zeros([10])
- for i in range(500):
-     z += x[i]

+ z = tf.reduce_sum(x, axis=0)
```