УДК 004.8

APACHE SPARK И PYSPARK

Борисов И. Д., магистрант, МОиАИС БИТ, 1 курс
Семёнов В. А., магистрант, МОиАИС БИТ, 1 курс
Бычкова Я. А., магистрант, МОиАИС БИТ, 1 курс
Чжао М.Н., магистрант, МОиАИС БИТ, 1 курс
Балтийский федеральный университет имени Иммануила Канта
г. Калининград

Apache Spark реализован на языке программирования Scala [1], который выполняется на виртуальной машине Java (JVM) [2]. Для возможности использования Spark в Python, используется PySpark.

Взаимодействие PySpark с самим Spark реализовано через специальную библиотеку Py4J. Она позволяет программам Python динамически обращаться к Java объектам в JVM, транслируя код Scala в JVM. Для большей совместимости PySpark поддерживает парадигму функционального программирования, поскольку Scala является функциональным языком программирования, а функциональный код намного проще распараллелить. Таким образом, PySpark позволяет проводить параллельную обработку без необходимости использования каких-либо модулей Python для потоковой или многопроцессорной обработки. Все сложные коммуникации и синхронизации между потоками и процессами обрабатываются в Spark.

ТОЧКА ВХОДА ЧЕРЕЗ SPARKCONTEXT И SPARKCONF ИЛИ ЧЕРЕЗ SPARKSESSION.

Точкой входа в Spark-приложение для создания DataFrame [3] является SparkSession, в котором определяются параметры конфигурации: название приложения, кластерный менеджер (способ подключения: локально, Kubernates или YARN), количество выделяемых ядер и памяти. Пример инициализации может выглядеть так (рис. 1):

```
from pyspark.sql import SparkSession

spark = SparkSession.builder
.master("local[*]")
.appName("Collecting data")
.config("spark.your.config.option", "your-value")
.getOrCreate()
```

Рис. 1 пример инициализации.

Другой и чаще всего используемый способ инициализации осуществляется через SparkContext и SparkConf, этот способ использовался до появления Spark 2.0. Пример кода на рис. 2:

```
from pyspark import SparkConf
from pyspark import SparkContext
conf = SparkConf().setAppName('AppName').setMaster('local[*]')
sc = SparkContext(conf=conf)
spark = SparkSession(sc)
```

Рис. 2 второй пример инициализации.

PYSPARK.

PySpark включает в себя множество модулей, таких как: SQL, Streaming, ML, MLlib. Pyspark служит для создания DataFrame и включает в себя следующие классы:

- SparkSession — точка входа для создания DataFrame и использования функций SQL [4];
- DataFrame — распределенный набор данных, сгруппированных в именованные столбцы;
- Column — столбец в DataFrame.
- Row — строка в DataFrame;
- GroupedData — агрегационные методы, возвращаемые DataFrame.GroupBy();
- DataFrameNaFunctions — методы обработки Nan значений;
- DataFrameStatFunctions — методы для статистической обработки данных;
- functions — список встроенных в модуль функций, доступных для DataFrame;
- types — список доступных типов данных;
- Модуль Streaming.

Модуль Streaming в PySpark представляет собой расширение основного API Spark, которое позволяет обрабатывать данные в режиме реального времени из различных источников, таких как Kafka, Flume и Amazon Kinesis. Он предоставляет доступ к функциональности потоковой передачи данных и позволяет отправлять обработанные данные в файловые системы, базы данных или дэшборды.

Основой модуля Streaming является DStream (Discretized Stream), который представляет поток данных, разделенный на небольшие пакеты RDD. Эти пакеты могут быть интегрированы с любыми другими компонентами Spark, включая MLlib.

МОДУЛИ ML И MLlib.

PySpark также предлагает два модуля для машинного обучения — ML и MLlib. Они предоставляют различные инструменты для построения моделей

машинного обучения. Модуль ML использует DataFrame, в то время как MLlib использует RDD. Разработчики Spark рекомендуют использовать модуль ML, так как он более удобен в работе.

Модули машинного обучения в PySpark содержат разнообразные инструменты для построения моделей, включая конвейер (pipeline) для составления стадий моделирования, инструменты для извлечения данных, такие как Binarizer, MinMaxScaler, CountVectorizer, Word2Vec и другие классы (всего 51 класс), классификацию (логистическую регрессию, деревья решения, случайные леса и т.д.), кластеризацию (13 алгоритмов, включая k-средние и LDA) и регрессию (линейную, деревья решения и другие 18 алгоритмов) [5].

Список литературы

1. The Scala Programming Language. [Электронный ресурс] URL: https://www.scala-lang.org/. (дата обращения 4.03.2023)
2. Java Virtual Machine. [Электронный ресурс] URL: https://www.java.com/en/download/. (дата обращения 4.03.2023)
3. Pandas.DataFrame Documentation. [Электронный ресурс] URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html. (дата обращения 4.03.2023)
4. Что такое SQL?. [Электронный ресурс] URL: https://aws.amazon.com/ru/what-is/sql/. (дата обращения 4.03.2023)
5. Обзор самых популярных алгоритмов машинного обучения — Tproger. [Электронный ресурс] URL: https://tproger.ru/translations/top-machine-learning-algorithms/. (дата обращения 4.03.2023)