# Koalas: 让 pandas 开发者轻松过渡到 Apache Spark

今年的 Spark + AI Summit 2019 databricks 开源了几个重磅的项目，比如 Delta Lake，Koalas 等，Koalas 是一个新的开源项目，它增强了 PySpark 的 DataFrame API，使其与 pandas 兼容。

Python 数据科学在过去几年中爆炸式增长，pandas 已成为生态系统的关键。
当数据科学家拿到一个数据集时，他们会使用 pandas 进行探索。
它是数据清洗和分析的终极工具。 事实上，pandas 的 read_csv
函数通常是学生在数据科学旅程中的第一个命令。

那么这么用的问题是什么呢？ pandas
不能很好地适应大数据，它专为单机处理小型数据集而设计的。另一方面，Apache Spark
已成为大数据 workloads 的事实标准。今天，许多数据科学家将 pandas
用于课程作业，个人业余项目（pet
projects）和小型数据任务，但是当他们使用非常大的数据集时，他们必须迁移到 PySpark
以便可以利用 Spark，或者对其数据进行下采样，以便他们可以使用 pandas。

现在有了 Koalas，数据科学家可以从单机过渡到分布式环境，而无需学习新的框架。
正如您在下面所看到的，只需将一个包替换为另一个包，就可以使用 Koalas 在 Spark
上扩展我们的 pandas 代码。
pandas:

```
import pandas as pd
df = pd.DataFrame({'x': [1, 2], 'y': [3, 4], 'z': [5, 6]})
# Rename columns
df.columns = [‘x’, ‘y’, ‘z1’]
# Do some operations in place
df[‘x2’] = df.x * df.x
```

Koalas:

```
import databricks.koalas as ks
df = ks.DataFrame({'x': [1, 2], 'y': [3, 4], 'z': [5, 6]})
# Rename columns
df.columns = [‘x’, ‘y’, ‘z1’]
# Do some operations in place
df[‘x2’] = df.x * df.x
```

# pandas 作为 Python 数据科学的标准词汇

随着 Python 成为数据科学的主要语言，社区基于最重要的库构建了一些词汇表，包括 pandas，matplotlib 和 numpy。
当数据科学家使用这些库时，他们可以充分表达他们的想法，并根据这个想法得出结论。
他们可以概念化某些东西并立即执行。

但是当他们不得不使用他们词汇表之外的库时，他们会遇到许多问题，他们每隔几分钟检查一次 StackOverflow，并且必须中断他们的工作流程才能使他们的代码工作。 尽管 PySpark 使用起来很简单并且在很多方面类似于 pandas，但他们仍然需要学习不同的词汇。

在 Databricks，我们相信 Spark 上的 pandas 将大大提高数据科学家和数据驱动型组织的工作效率，原因如下：

- Koalas 无需决定是否对给定的数据集使用 pandas 或 PySpark
- 对于最初用 pandas 编写的单机程序，Koalas 允许数据科学家通过 pandas 和 Koalas 的轻松切换来扩展在 Spark 上的代码；
- Koalas 为组织中的更多数据科学家解锁大数据，因为他们不再需要学习 PySpark 来利用 Spark；

下面我们展示了两个简单而强大的 pandas 方法示例，这些方法可以直接在 Spark with Koalas 上运行。

## 具有分类变量的特征工程

数据科学家在构建 ML 模型时经常会遇到分类变量。
一种流行的技术是将分类变量编码为虚拟变量。
在下面的示例中，有几个分类变量，包括呼叫类型，邻域和单元类型。 pandas 的get_dummies 方法是一种方便的方法。 下面我们将展示如何使用 pandas：

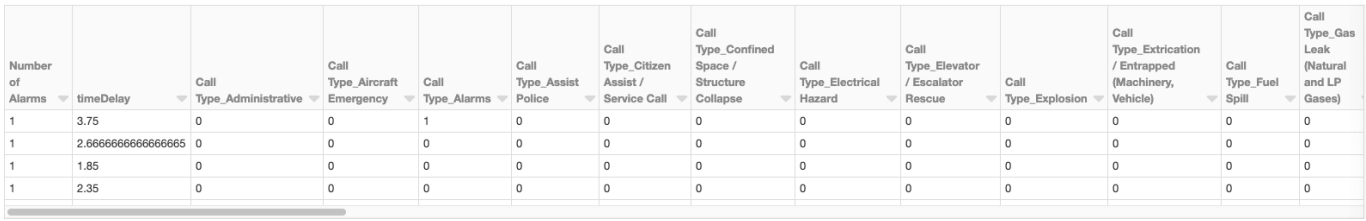
```
import pandas as pd
data = pd.read_csv("fire_department_calls_sf_clean.csv", header=0)
display(pd.get_dummies(data))
```

原始的 DataFrame

| Call Type | Neighborhooods - Analysis Boundaries | Number of Alarms | Original Priority | Unit Type | timeDelay |
|---|---|---|---|---|---|
| Alarms | Mission | 1 | 3 | TRUCK | 3.75 |
| Medical Incident | South of Market | 1 | 3 | MEDIC | 2.6666666666666665 |
| Medical Incident | Financial District/South Beach | 1 | 3 | MEDIC | 1.85 |
| Medical Incident | Excelsior | 1 | 3 | ENGINE | 2.35 |
| Medical Incident | Tenderloin | 1 | 2 | PRIVATE | 5.616666666666666 |

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公众号：iteblog_hadoop

变换后的 DataFrame



| Number of Alarms | timeDelay | Call Type_Administrative | Call Type_Aircraft Emergency | Call Type_Alarms | Call Type_Assist Police | Call Type_Citizen Assist / Service Call | Call Type_Confined Space / Structure Collapse | Call Type_Electrical Hazard | Call Type_Elevator / Escalator Rescue | Call Type_Explosion | Call Type_Extrication / Entrapped (Machinery, Vehicle) | Call Type_Fuel Spill | Call Type_Gas Leak (Natural and LP Gases) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3.75 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 2.6666666666666665 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1.85 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 2.35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公众号：iteblog_hadoop

有了 Koalas 之后，我们可以通过一些调整在 Spark 上做到这一点：

```
import databricks.koalas as ks
data = ks.read_csv("fire_department_calls_sf_clean.csv", header=0)
display(ks.get_dummies(data))
```

## 带时间戳的算术

数据科学家一直使用时间戳，但正确处理它们可能会变得非常困难。pandas 提供了一个优雅的解决方案。 假设您有一个日期的 DataFrame：

```
import pandas as pd
import numpy as np
date1 = pd.Series(pd.date_range('2012-1-1 12:00:00', periods=7, freq='M'))
date2 = pd.Series(pd.date_range('2013-3-11 21:45:00', periods=7, freq='W'))
df = pd.DataFrame(dict(Start_date = date1, End_date = date2))
print(df)
```

```
  End_date           Start_date
0 2013-03-17 21:45:00 2012-01-31 12:00:00
1 2013-03-24 21:45:00 2012-02-29 12:00:00
2 2013-03-31 21:45:00 2012-03-31 12:00:00
3 2013-04-07 21:45:00 2012-04-30 12:00:00
4 2013-04-14 21:45:00 2012-05-31 12:00:00
5 2013-04-21 21:45:00 2012-06-30 12:00:00
6 2013-04-28 21:45:00 2012-07-31 12:00:00
```

要使用 pandas 从结束日期中减去开始日期，您只需运行：

```
df['diff_seconds'] = df['End_date'] - df['Start_date']
```

```
df['diff_seconds'] = df['diff_seconds']/np.timedelta64(1,'s')
print(df)
```

结果

```
  End_date          Start_date        diff_seconds
0 2013-03-17 21:45:00 2012-01-31 12:00:00 35545500.0
1 2013-03-24 21:45:00 2012-02-29 12:00:00 33644700.0
2 2013-03-31 21:45:00 2012-03-31 12:00:00 31571100.0
3 2013-04-07 21:45:00 2012-04-30 12:00:00 29583900.0
4 2013-04-14 21:45:00 2012-05-31 12:00:00 27510300.0
5 2013-04-21 21:45:00 2012-06-30 12:00:00 25523100.0
6 2013-04-28 21:45:00 2012-07-31 12:00:00 23449500.0
```

现在要在 Spark 上做同样的事情，你需要做的就是用 Koalas 替换 pandas：

```
import databricks.koalas as ks
df = ks.from_pandas(pandas_df)
df['diff_seconds'] = df['End_date'] - df['Start_date']
df['diff_seconds'] = df['diff_seconds'] / np.timedelta64(1,'s')
print(df)
```

就这么简单。

## 接下来的安排和 Koalas 入门

我们创建了 Koalas ，是因为我们遇到了许多不愿意处理大数据的数据科学家。我们相信 Koalas 会通过让他们很容易的在 Spark 上扩展他们程序，从而使得他们能够做更多的事。

到目前为止，我们已经实现了常见的 DataFrame 操作方法，以及 pandas 中强大的索引技术。以下是我们路线图中的一些即将推出的项目，主要侧重于改善覆盖范围：

- 用于处理文本数据的字符串操作；
- 时间序列数据的日期/时间操作。

该计划尚处于初期阶段，但正在迅速发展。 如果您有兴趣了解更多有关 Koalas

及入门的信息，请查看该项目的 GitHub 地址。

本文翻译自 Koalas: Easy Transition from pandas to Apache Spark

本博客文章除特别声明，全部都是原创！
转载本文请加上：转载自过往记忆（https://www.iteblog.com/）
本文链接: 【】（）