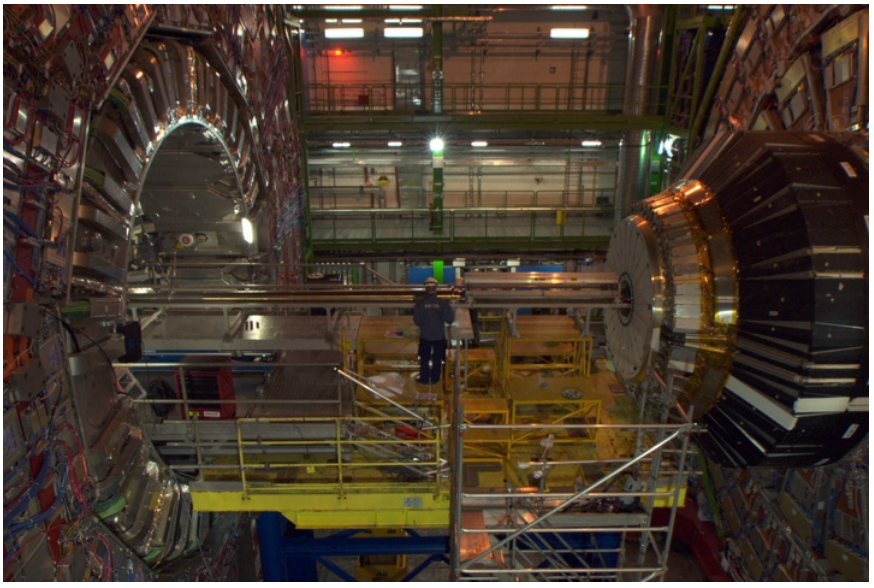
CERN seeks to predict new and popular data sets

Apache Spark eyed as potential framework for big data analysis at one of the world's most prominent nuclear research organizations.

By Siddha Ganju, March 22, 2016



Large Hadron Collider.
(source: Nikolaus Geyrhalter Filmproduktion on Wikimedia Commons).

Editor's note: Siddha Ganju will talk more about big data analysis at CERN in a talk "Atom smashing using machine learning at CERN" at Strata + Hadoop World San Jose, March 28-31, 2016.

Last summer, I worked as a summer openlab intern at the European Organization for Nuclear Research (CERN), in Geneva. The focus of my work was on exploring

the Apache Spark MLlib framework for CERN's big data analyses.

A major experiment at CERN is the Compact Muon Solonoid (CMS), which envisages contribution to the understanding of subatomic particles. This experiment is taking place at CERN's Large Hadron Collider (LHC). The LHC is a particle accelerator that pushes subatomic particles at an express speed and is visualized by the CMS detector, a huge multi-story digital camera that records images per second of debris produced by the LHC particle collision. The CMS experiment collects data on the order of O(10) PB/year. The data increases enormously with each collision over time. The maximum data production rate can reach around 600 MB/second, and presents a considerable processing challenge. This data is stored and processed by a multi-tier computing infrastructure on the worldwide LHC Computing Grid.

Since data placement is an essential component of the experiments at CERN, we are looking for a different way to improve this task and have developed a pilot project, "Evaluation of Apache Spark as Analytics Framework for CERN's Big Data Analytics Infrastructure." The objective of this pilot project is to make appropriate predictions from the CMS data, improve resource utilization, and obtain an in-depth understanding of the metrics and the framework. (This will be the focus of my talk at the upcoming Strata + Hadoop World conference in San Jose, March 28-31, 2016.)

SESSION

Atom smashing using machine learning at CERN
STRATA + HADOOP WORLD IN SAN JOSE 2016

Understanding popular CMS data sets

The first stage of CERN's pilot project is predicting new and popular CMS data sets. Popularity is defined as data sets frequently used for research. These are considered popular because they are regularly accessed by physicists and hence need to be replicated at various data centers around the world. Recognizing popular data sets improves the efficiency of analysis and helps to identify data

sets that might become hot topics of high-energy physics, such as Higgs boson and supersymmetric particles.

Figure 1 shows the popularity of random data sets over a span of weeks in 2014, using the logarithmic scale. Each line represents a different data set. The data set represented by the black line has been accessed 20-30% more during weeks 1-20, indicating the popularity of the data set. In contrast, the data set represented by the yellow line has not been accessed at all, indicating it is an unpopular data set.

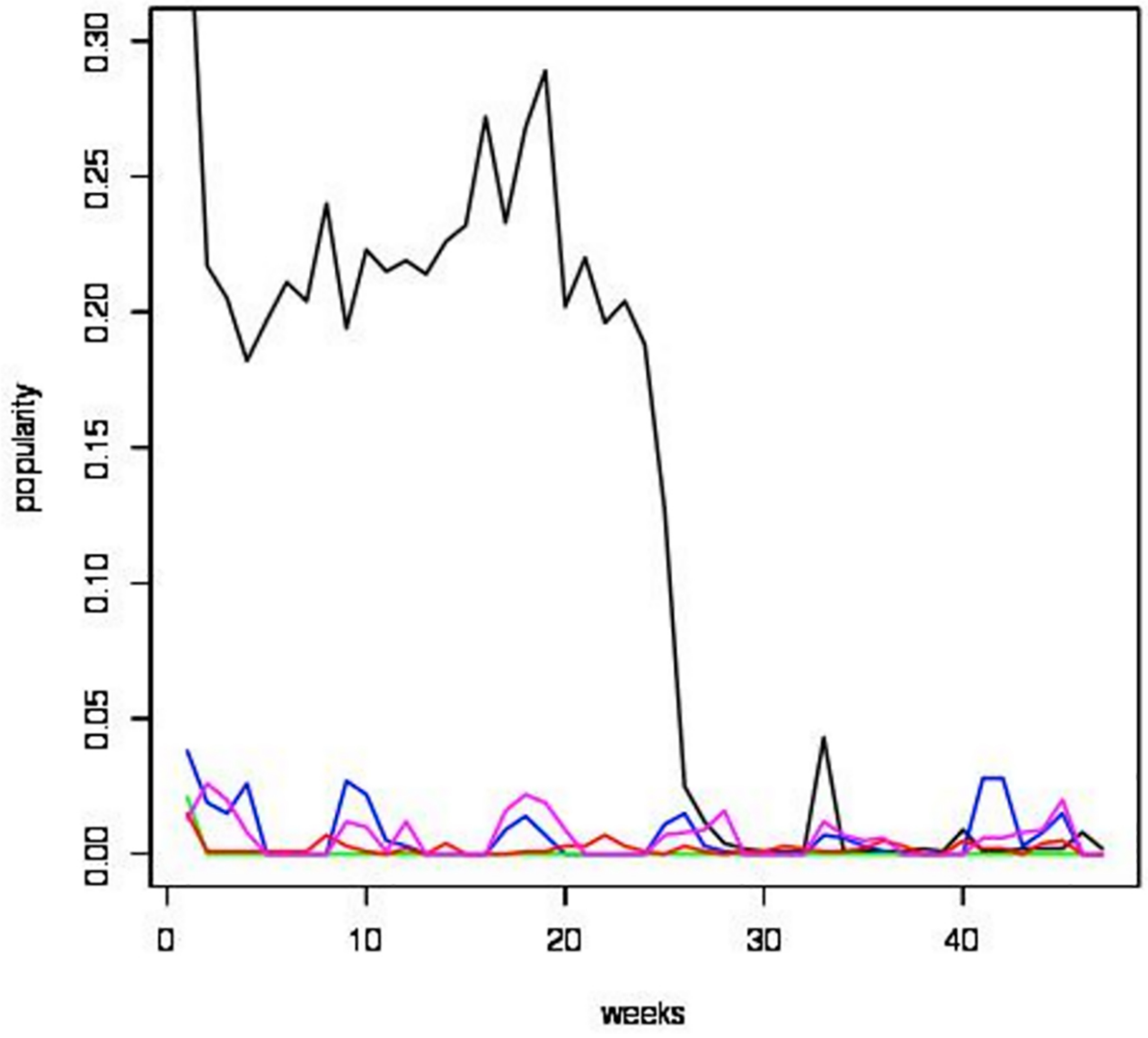


Figure 1. Popularity of random CMS data sets over a span of weeks. Chart courtesy of Valentin Kuznetsov, used with permission.

Relative popularity can also be gauged by making cloud plots based on a single popularity metric like naccess, totcpu (see Figure 2), or nusers (see Figure 3) distributions. Naccess is the count of individual accesses to a dataset, totcpu is the total CPU hours spend to analyse a dataset, and nusers is the total number of users who accessed the dataset. These users can be physicists, students or researchers. The numbers you see in the figures represent the data set name. The data set nomenclature includes the date, software version and format, and is defined by three distinct parts: process, software, and tier. These three parts are important because they help to replicate the process.

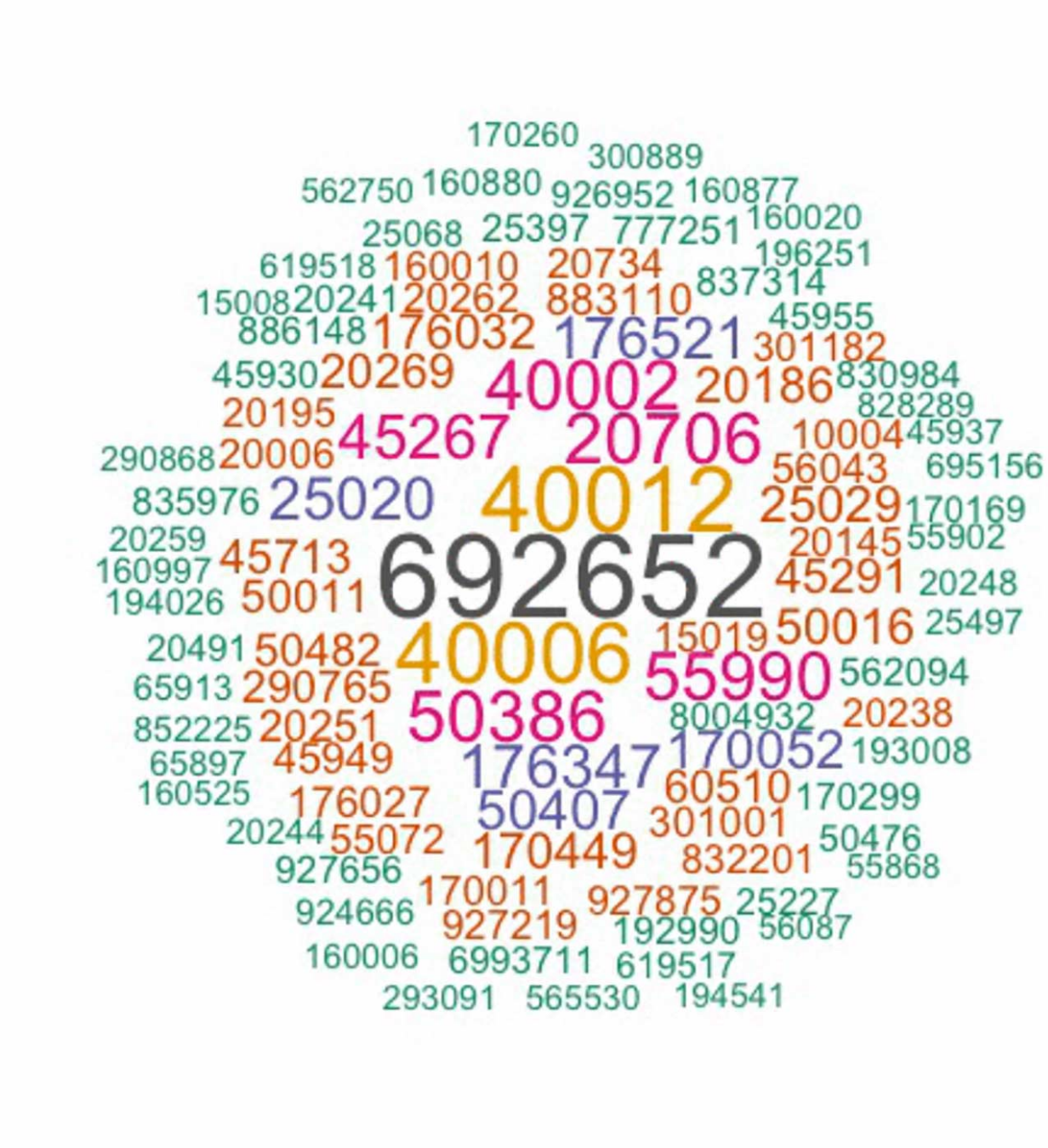


Figure 2. Popular CMS data sets using the totcpu metric represented in a cloud plot. Courtesy of Valentin Kuznetsov, used with permission.

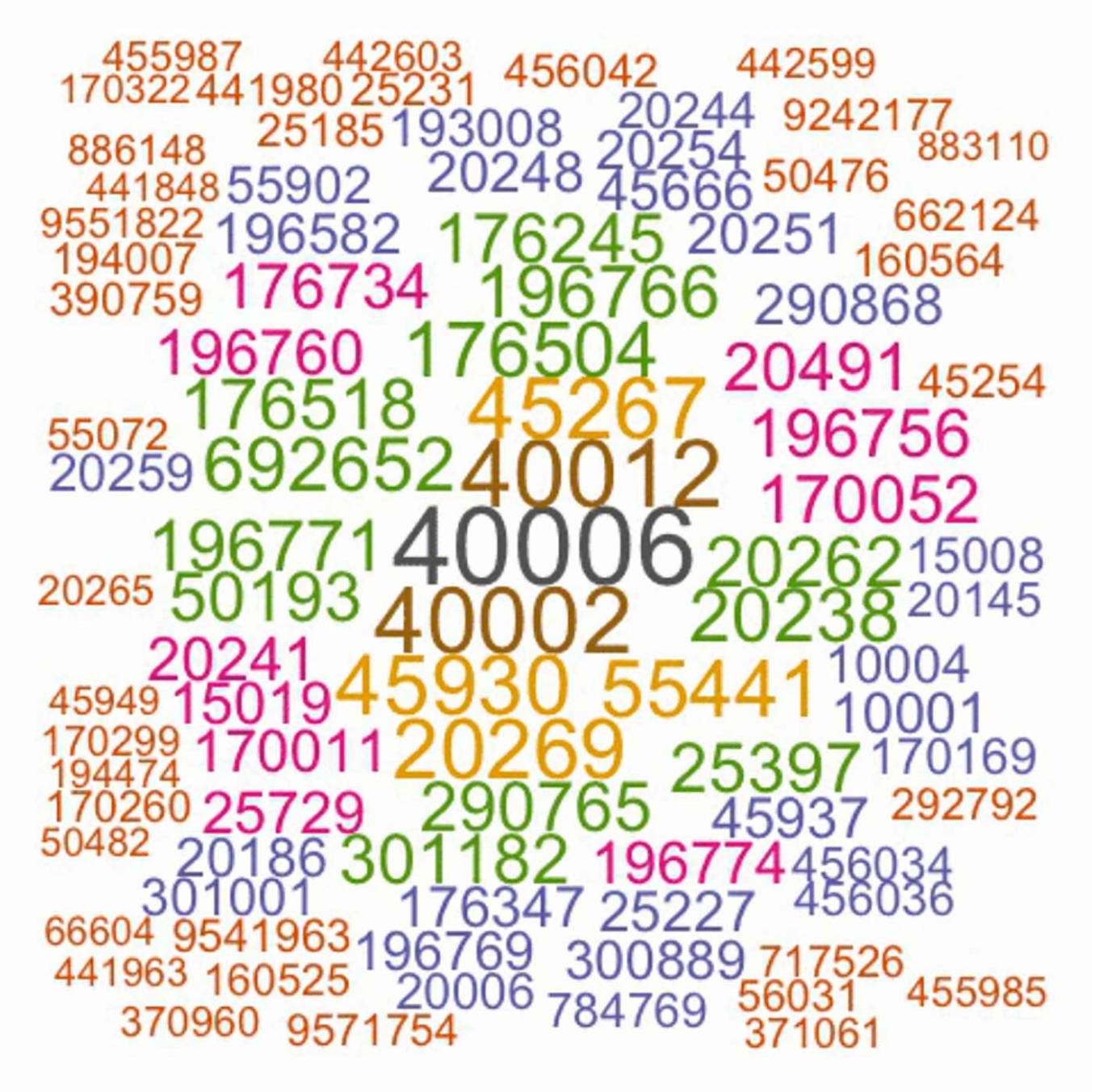


Figure 3. Popular CMS data sets using the nusers metric shown in a cloud plot. Courtesy: Valentin Kuznetsov, used with permission.

The CMS data sets most frequently accessed in 2014 are shown in Figure 4.

Get O'Reilly's weekly data newsletter

1. Predictive modeling in regulated industries

Here's **how to strike a balance between accuracy and interpretability when you're using machine learning models in regulated industries**.

Related resources:

- **How the machine learning wave is changing the way organizations look at analytics** (free webcast)
- **Finance-related sessions** at Strata + Hadoop World in San Jose
- ***Data Science, Banking, and Fintech*** (free report)

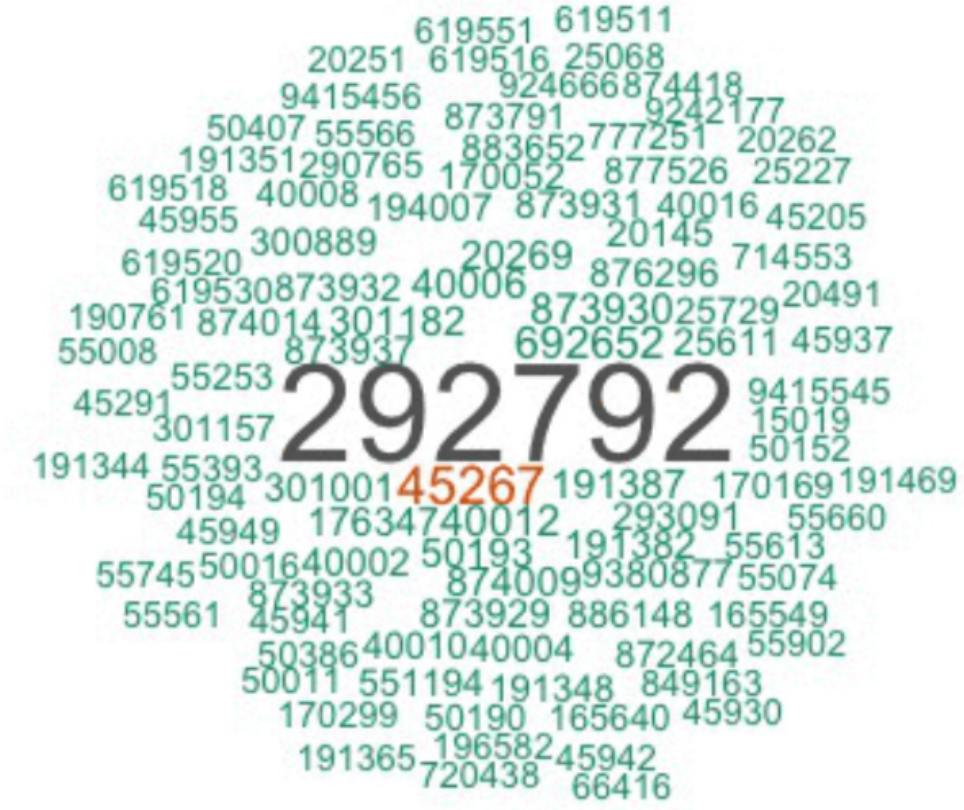


Figure 4. The 100 most frequently accessed CMS data sets in 2014 shown in a cloud plot. Courtesy: Valentin Kuznetsov, used with permission.

Using Apache Spark to predict new and popular CMS data sets

Machine learning algorithms are able to run predictive models and suggest data sets that will become popular over time. I evaluated Apache Spark as a tool to streamline the different predictive prototypes that are capable of gathering information from CMS data services. When compared to earlier results obtained by the dynamic data placement method, the accuracy Spark provided was similar

—the big difference was that results were obtained in real time. Because Spark can analyze streaming data as it comes off the wire, a rolling forecast yielded popularity results as the data was being produced. Predicting popular data sets was then done by implementing machine learning algorithms using Spark's native machine learning library (MLlib) and Python. These algorithms mainly included Naïve Bayes, Stochastic Gradient Descent, and Random Forest.

Each week's data was added to the existing data and a new model was created, which led to better data analysis. These models were then combined into an ensemble and evaluated using true positive, true negative, false positive, or false negative values. I also used scikit-learn in Python and compared the values obtained from different frameworks.Through this process, I was able to determine the quality of individual models.MLlib has state-of-the-art implementations of almost all machine learning algorithms and yielded better results in terms of CPU consumption and memory used, in comparison to Python frameworks.

Conclusion

The accuracy measurements of model prototypes from both Spark and scikit-learn were almost similar. By performing principle component analysis, I was able to choose the best predictive model(s) for new data sets interactively. Other factors critical to CMS data analyses are parallel and fast distributed data processing. The Spark framework offers a simple programming abstraction that provides powerful caching and persistence capabilities, along with high speed. In conclusion, I found that the components of Spark—Spark Streaming and MLlib—immensely simplified the analysis of CMS data and can be successfully applied to CMS data sets.

Siddha Ganju

Siddha Ganju is a Master's student of computational data science at Carnegie Mellon University and was a 2015 summer openlab intern at CERN. Her research is at the junction of machine learning, natural language processing, information retrieval, and deep learning. Siddha has a Bachelor's degree from the National Institute of Technology, Hamirpur, India.

How intelligent data platforms are powering smart cities

By Ben Lorica

Smart cities and smart nations run on data.

Beyond algorithms: Optimizing the search experience

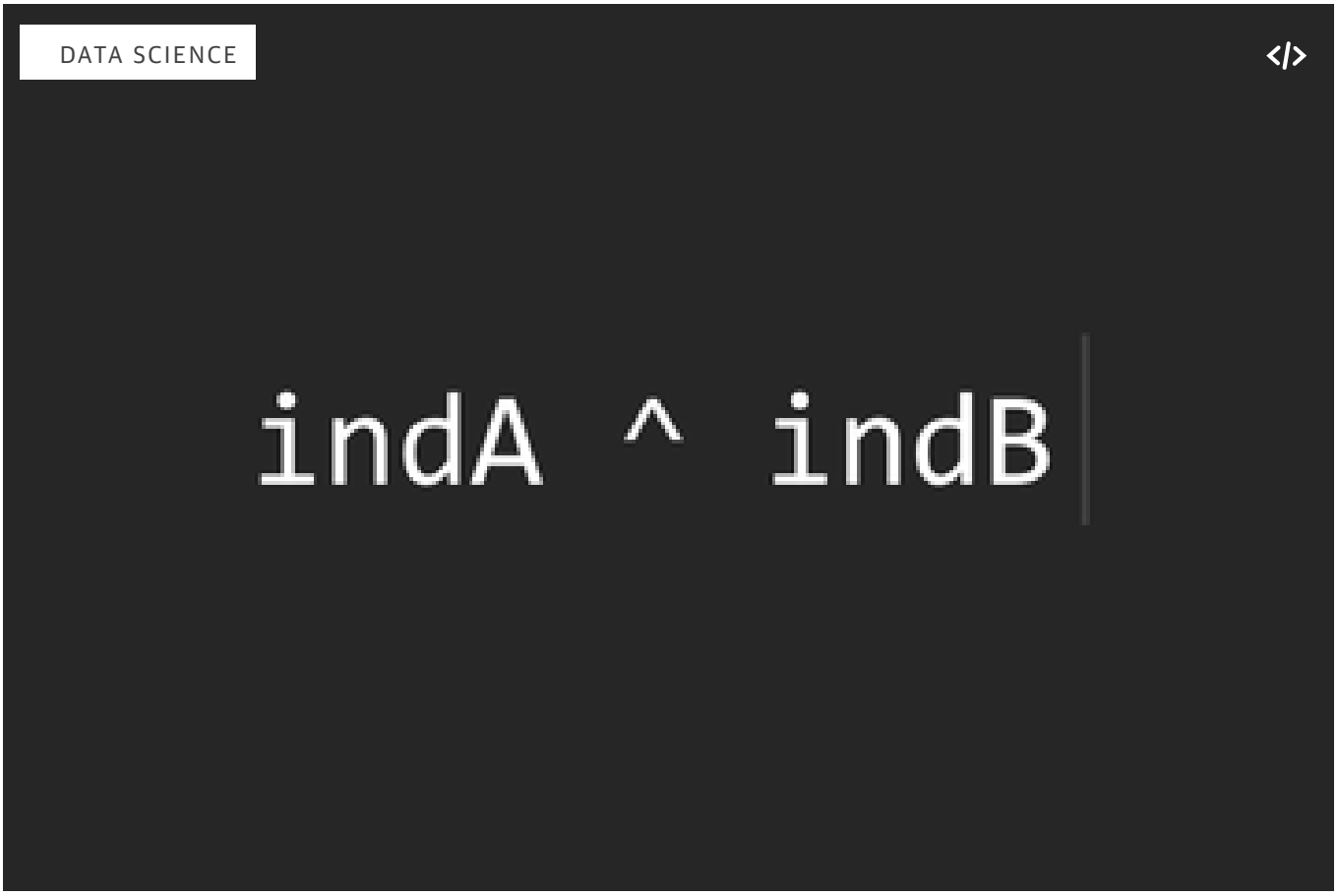
By Daniel Tunkelang

Making search smarter through better human-computer interaction.

Handling missing data

By Jake VanderPlas

Python Data Science Handbook: Early Release



Introducing Pandas Objects

By Jake VanderPlas

Python Data Science Handbook: Early Release