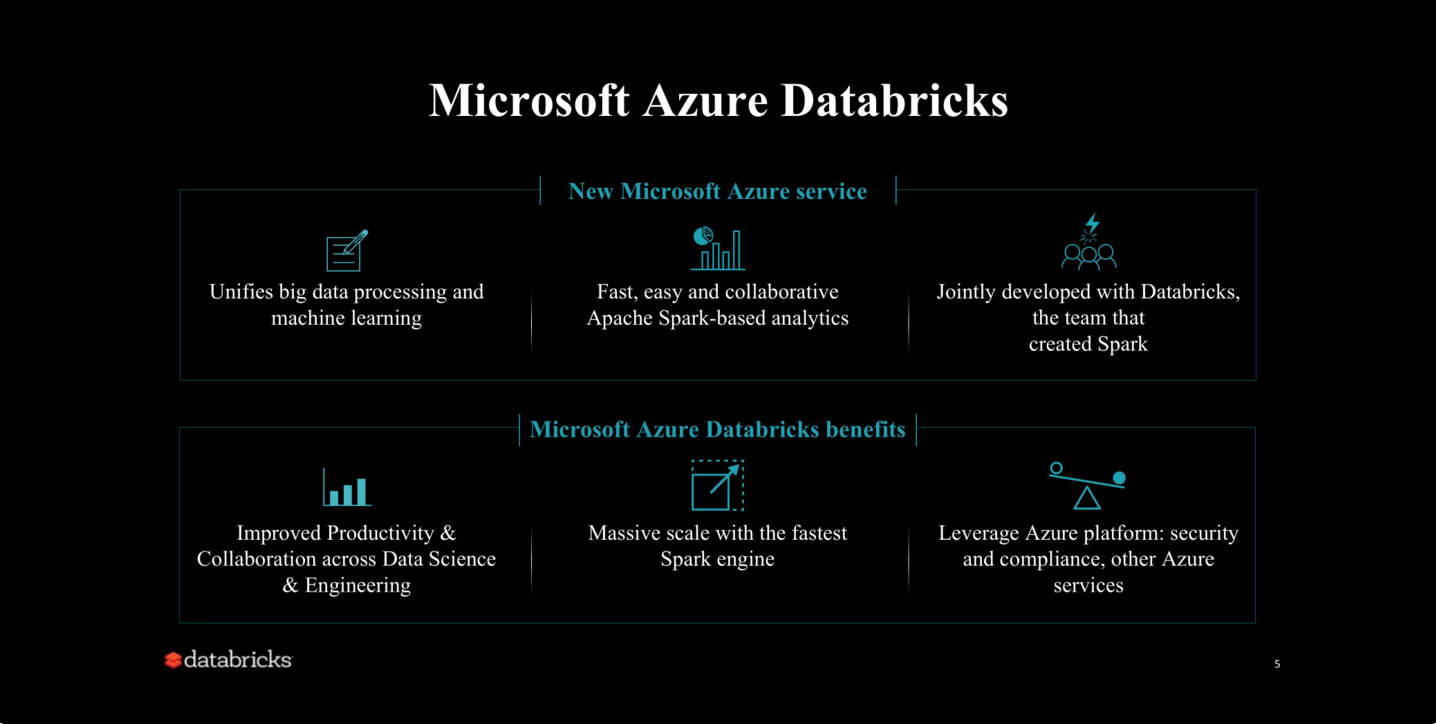
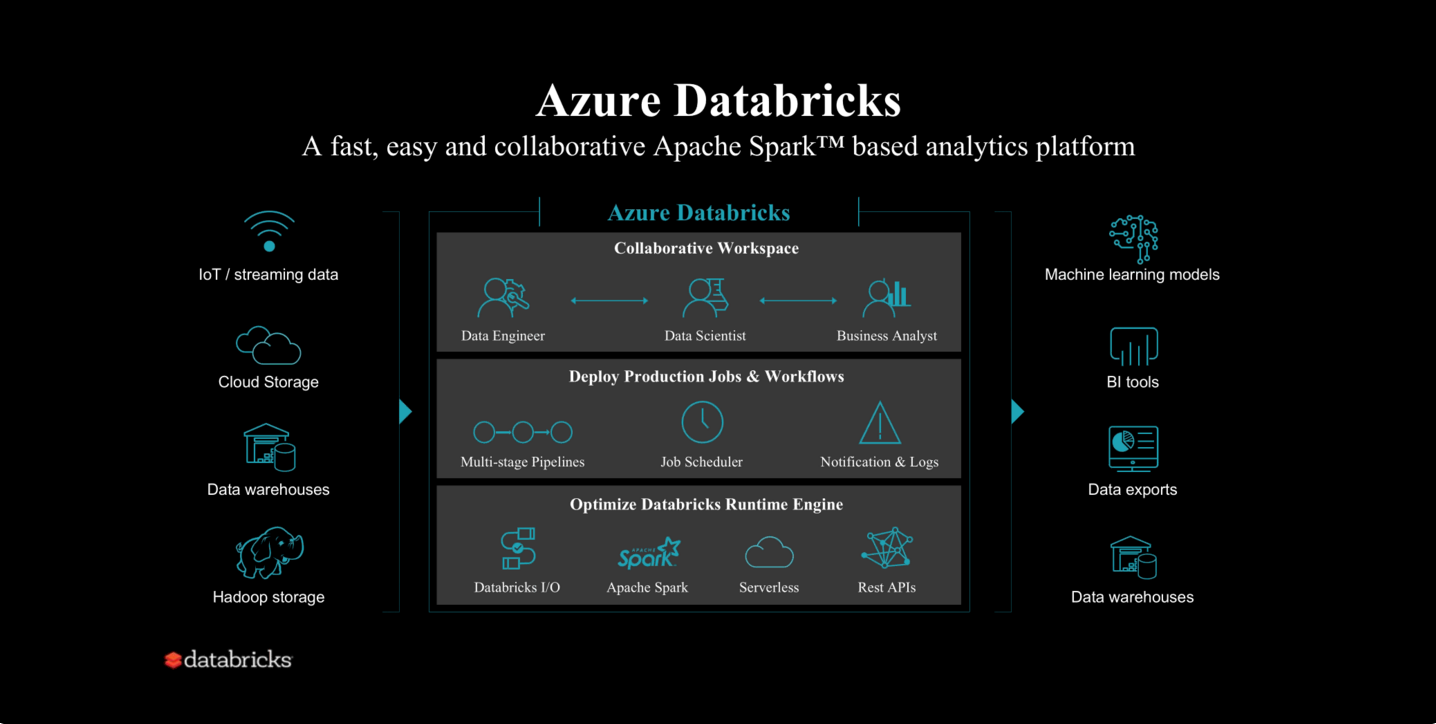
Azure Databricks:

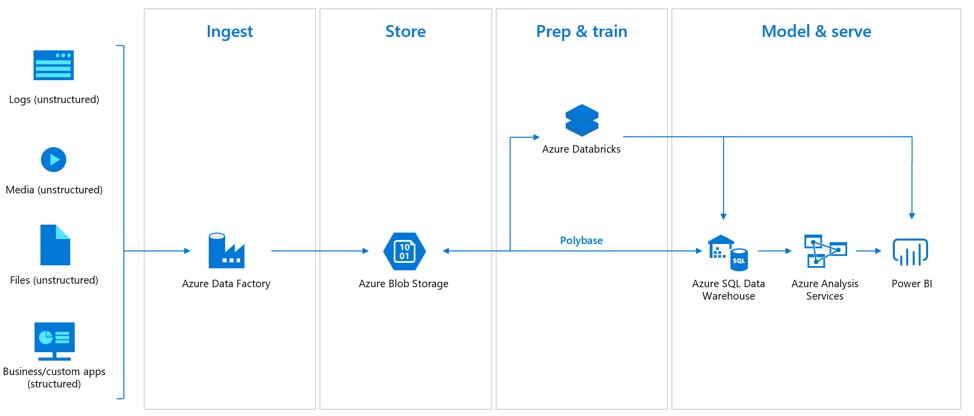
Azure Databricks provide fully managed, highly scalable and available unified compute platform using Databricks distribution of open source Spark. Azure makes it very simple by automating many steps in setting up and getting started using Databricks spark. Developers can get started in minutes to create workspace, configure and start cluster and build data models using R, Python, Scala and SQL APIs. It also provides popular deep learning frameworks such as TensorFlow, Pytorch, ScikitLearn and Spark MLib. It is also integrated with Azure Storage, Azure CosmosDB, EventHub, Azure Data Factory and Azure Active Directory. Azure Databricks makes it simple, productive and collaborative environment for data scientist, data engineers and data analyst of the organization.



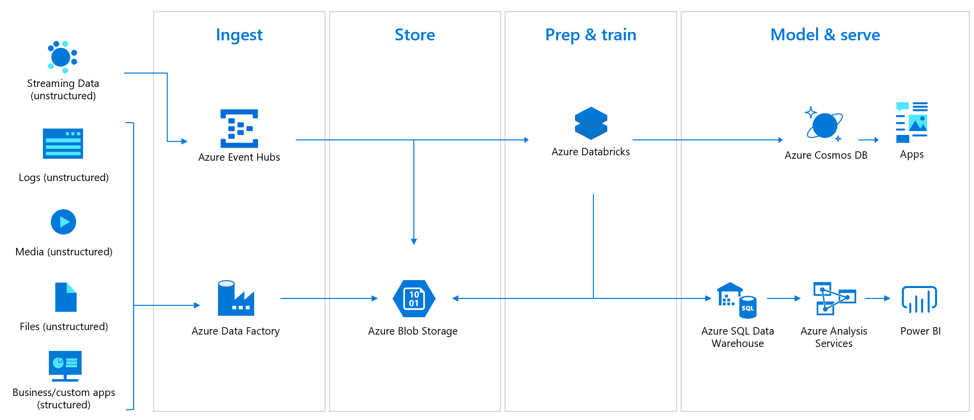
Azure Databricks integrated with other Azure Data & Analytics services provides comprehensive solution for collecting data, preparing features, building, training and testing machine learning model. Event hub and ADF can load stream and batch data into azure data storage for scalable and low-cost data storage. Azure Databricks can use spark APIs to prepare data and build models. Azure CosmosDB can serve model at scale through web and mobile interface. You can leverage powerful distributed computing for scalable data processing of Spark to build Data Warehouse and Analytics platform, Model Machine Learning and DataSceince Platform and Real time analytics platform.



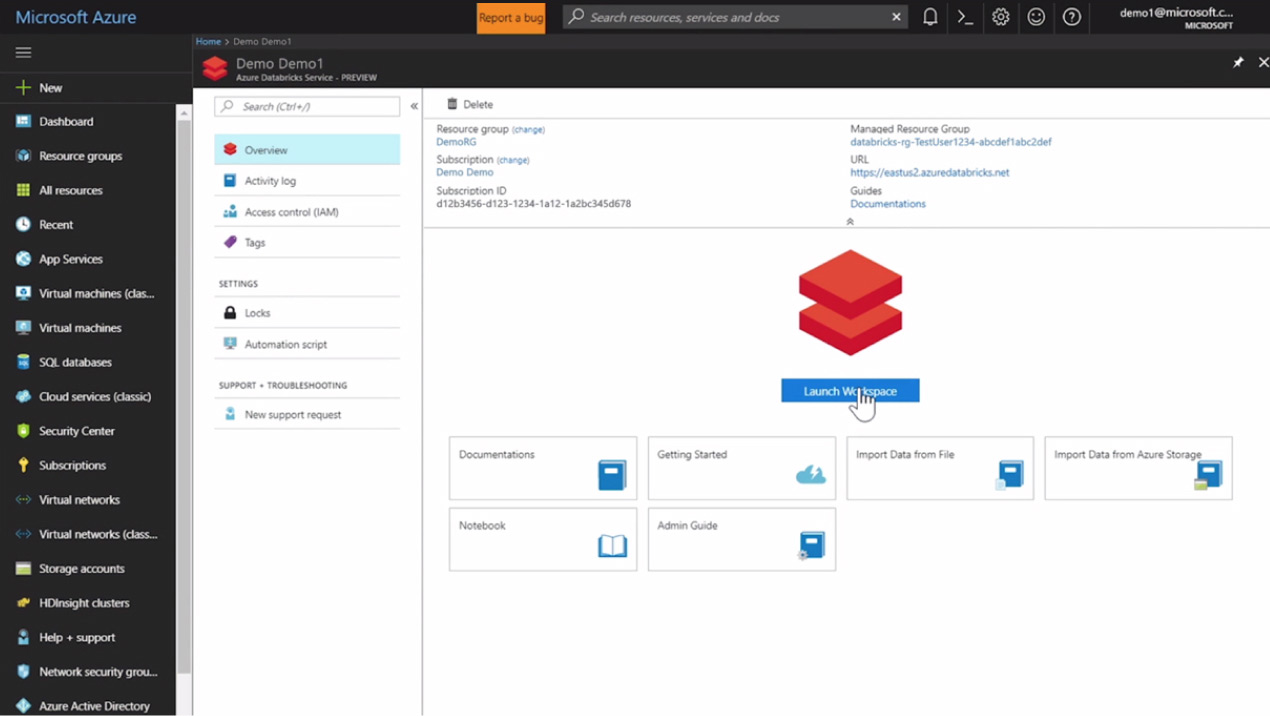
**Modern Data Warehouse & Advanced Analytics:** Collect all data types from all sources, process and transform data at scale using Azure Data Factory and Azure Databricks, analyze using Azure SQL DW, Analysis service and Power BI visualization.

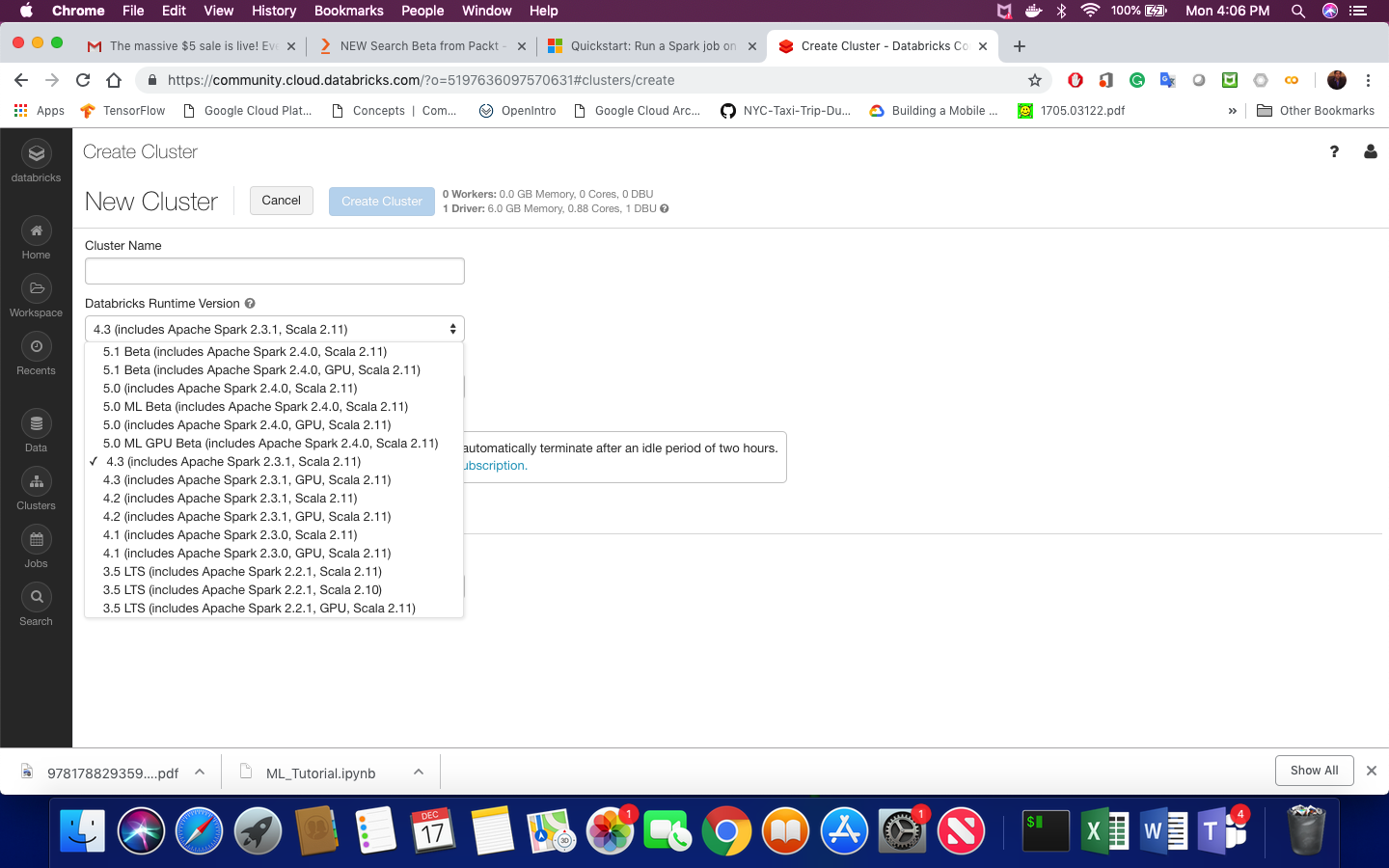


**Real time Analytics:** Get insights from streaming data with ease. Capture data continuously from any streaming source, or logs from website clickstreams, and process it in near-real time using Databricks and serve using CosmosDB.



**Azure Workspace:** To use azure Databricks first thing is to create an azure Databricks workspace. Now I am not going to go step by step how to get started with azure Databricks here because there is a great tutorial <https://docs.microsoft.com/en-us/azure/azure-databricks/quickstart-create-databricks-workspace-portal> that you can refer to learn how to get here. At this tutorial, I will assume you have azure subscription (there is always a free trial which get you $200.00 to get started), azure Databricks workspace and a cluster created. You need to select 5.0 ML Beta Databricks runtime while creating cluster for ML application. It will have pretty much all the API pre-installed.

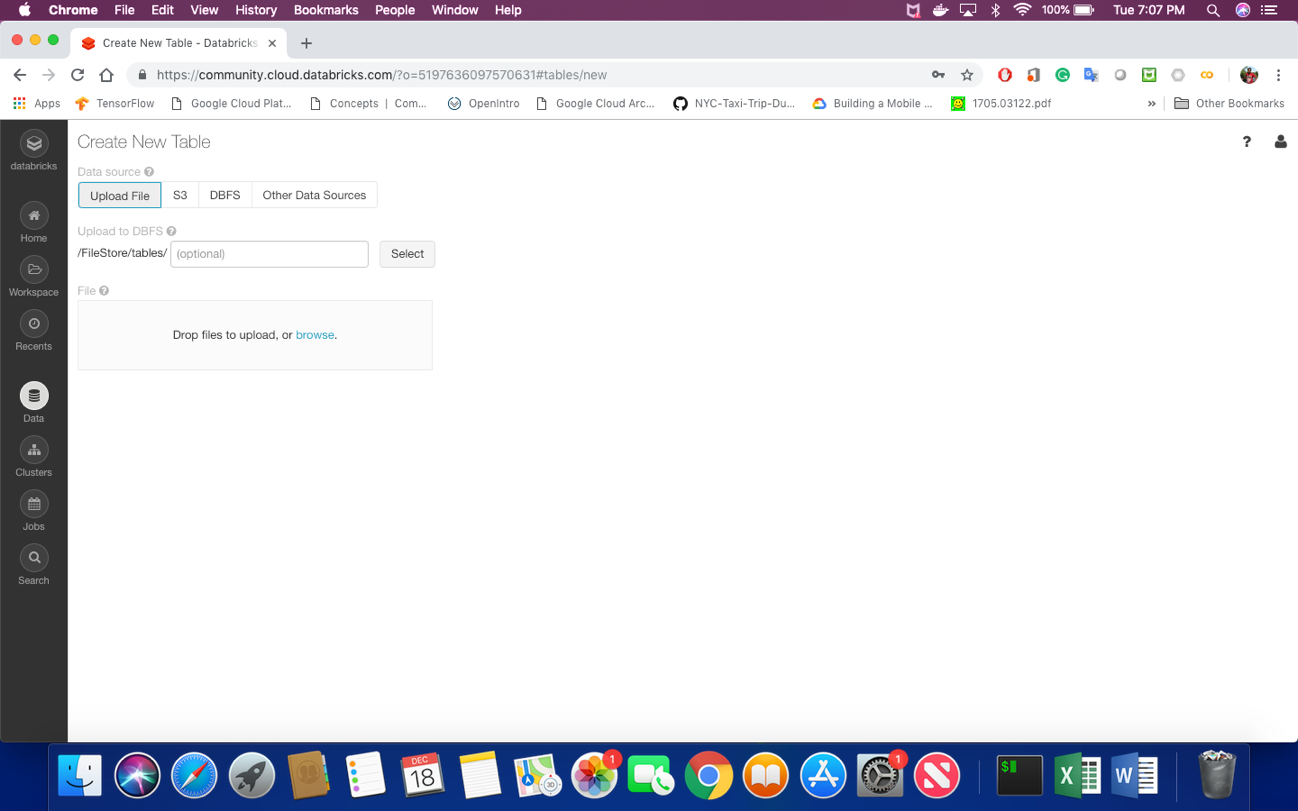




Machine Learning in Azure Databricks:

Now, we will develop a simple classification model that will classify customer data in to categories weather customer will churn or not. For this tutorial, we'll be using the Orange Telecoms Churn Dataset. It consists of cleaned customer activity data (features), along with a churn label specifying whether the customer canceled their subscription or not. The data can be fetched from BigML's S3 bucket, [churn-80](https://bml-data.s3.amazonaws.com/churn-bigml-80.csv) and [churn-20](https://bml-data.s3.amazonaws.com/churn-bigml-20.csv). The two sets are from the same batch, but have been split by an 80/20 ratio. We'll use the larger set for training and cross-validation purposes, and the smaller set for final testing and model performance evaluation.

We have to load the data first to Databricks file store using azure Databricks workspace as seen here.



Once data is loaded open a notebook using python option and start coding as shown here or go straight in the github here.

Lets look at the file using Databricks %fs magic command

%fs ls /FileStore/tables

Check the data in the file using head command

%fs head /FileStore/tables/churn.csv

[Truncated to first 65536 bytes] State,Account\_length,Area\_code,International\_plan,Voice\_mail\_plan,Number\_vmail\_messages,Total\_day\_minutes,Total\_day\_calls,Total\_day\_charge,Total\_eve\_minutes,Total\_eve\_calls,Total\_eve\_charge,Total\_night\_minutes,Total\_night\_calls,Total\_night\_charge,Total\_intl\_minutes,Total\_intl\_calls,Total\_intl\_charge,Customer\_service\_calls,Churn KS,128,415,No,Yes,25,265.1,110,45.07,197.4,99,16.78,244.7,91,11.01,10,3,2.7,1,FALSE OH,107,415,No,Yes,26,161.6,123,27.47,195.5,103,16.62,254.4,103,11.45,13.7,3,3.7,1,FALSE NJ,137,415,No,No,0,243.4,114,41.38,121.2,110,10.3,162.6,104,7.32,12.2,5,3.29,0,FALSE OH,84,408,Yes,No,0,299.4,71,50.9,61.9,88,5.26,196.9,89,8.86,6.6,7,1.78,2,FALSE OK,75,415,Yes,No,0,166.7,113,28.34,148.3,122,12.61,186.9,121,8.41,10.1,3,2.73,3,FALSE AL,118,510,Yes,No,0,223.4,98,37.98,220.6,101,18.75,203.9,118,9.18,6.3,6,1.7,0,FALSE MA,121,510,No,Yes,24,218.2,88,37.09,348.5,108,29.62,212.6,118,9.57,7.5,7,2.03,3,FALSE MO,147,415,Yes,No,0,157,79,26.69,103.1,94,8.76,211.8,96,9.53,7.1,6,1.92,0,FALSE WV,141,415,Yes,Yes,37,258.6,84,43.96,222,111,18.87,326.4,97,14.69,11.2,5,3.02,0,FALSE RI,74,415,No,No,0,187.7,127,31.91,163.4,148,13.89,196,94,8.82,9.1,5,2.46,0,FALSE IA,168,408,No,No,0,128.8,96,21.9,104.9,71,8.92,141.1,128,6.35,11.2,2,3.02,1,FALSE MT,95,510,No,No,0,156.6,88,26.62,247.6,75,21.05,192.3,115,8.65,12.3,5,3.32,3,FALSE IA,62,415,No,No,0,120.7,70,20.52,307.2,76,26.11,203,99,9.14,13.1,6,3.54,4,FALSE ID,85,408,No,Yes,27,196.4,139,33.39,280.9,90,23.88,89.3,75,4.02,13.8,4,3.73,1,FALSE VT,93,510,No,No,0,190.7,114,32.42,218.2,111,18.55,129.6,121,5.83,8.1,3,2.19,3,FALSE VA,76,510,No,Yes,33,189.7,66,32.25,212.8,65,18.09,165.7,108,7.46,10,5,2.7,1,FALSE TX,73,415,No,No,0,224.4,90,38.15,159.5,88,13.56,192.8,74,8.68,13,2,3.51,1,FALSE FL,147,415,No,No,0,155.1,117,26.37,239.7,93,20.37,208.8,133,9.4,10.6,4,2.86,0,FALSE

A more convenient way to view and process data in databricks using python is to create a dataframe as