

Hands on Introduction to IBM's Data Science Experience



Power of data. Simplicity of
design. Speed of innovation.

Joel Patterson
Bernie Beekman
Asad Mahmood
Davin Shearer

Agenda

Time	Description
8:30 AM - 9:00 AM	Registration and Coffee
9:00 AM - 10:00 AM	Overview of the Watson Data Platform and IBM Data Science Experience (DSX)
10:00 AM - 11:30 AM	Lab 1 - Setting Up Your First DSX Notebook
11:30 AM - 12:30 PM	Lab 2 - Machine Learning with Spark ML
12:30 PM - 1:30 PM	Lunch Provided
1:30 PM - 2:30 PM	Lab 3 - R, Shiny, and GUI Interfaces
2:30 PM - 4:00 PM	Lab 4 - Choose From Two Options
4:00 PM - 4:30 PM	Questions and Wrap Up

Participant Background

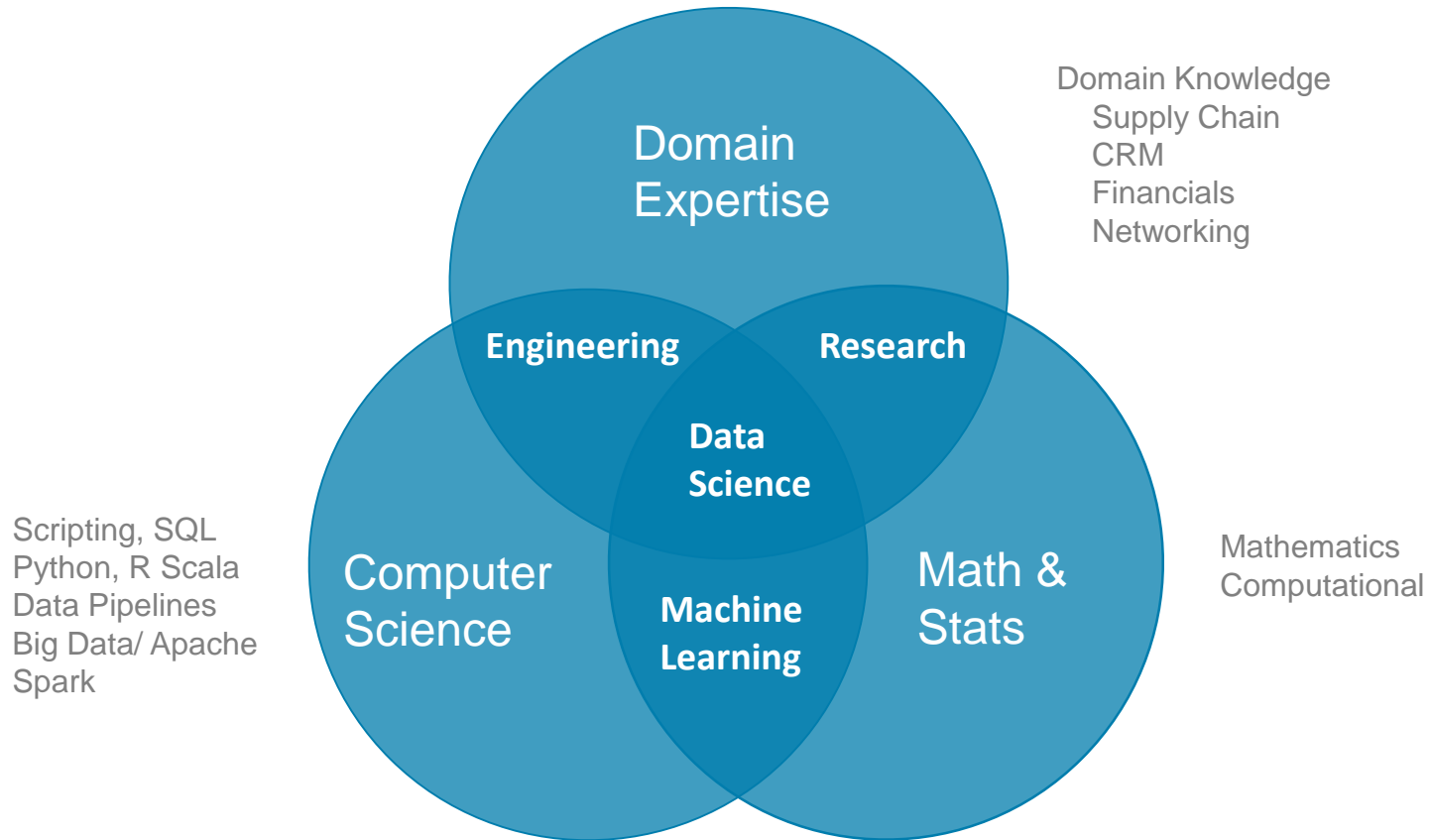
Open Source

- R/Python/Scala
- Jupyter Notebook
- Spark
- Hadoop

IBM

- Bluemix

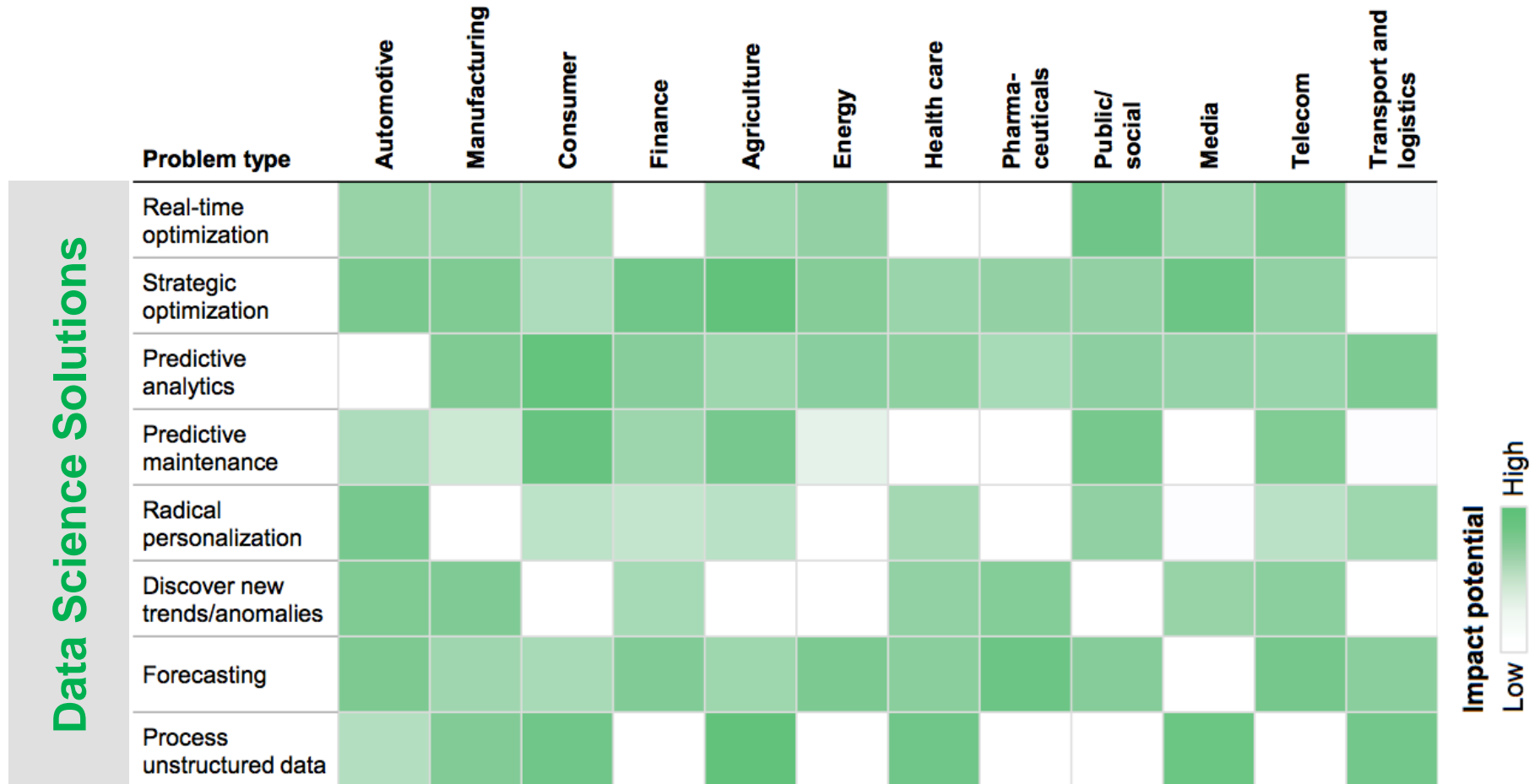
What is Data Science?



Data Science Projects Require Multiple Skills

Data Science Impact Across Industries and Use Cases

\$10s of Billions in each industry and use case



SOURCE: McKinsey Global Institute analysis

Challenges in delivering value with Data Science

Data

- Data resides in silos and difficult to access
- Detailed data was never stored
- Unstructured and external data wasn't considered

Skills

- Data Science skills are in low supply and high demand
- Nurturing new data professionals is challenging

Governance

- Self-service isn't a reality, if the data isn't secure
- Understanding lineage and getting to a system of truth

Infrastructure

- Need an environment that enables collaboration and deployment to production
- Discrete tools present barriers to progress

IBM Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality.

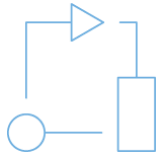
Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Watson Data Platform

IBM Watson Data Platform

Mission: Make Data Simple and Accessible to All



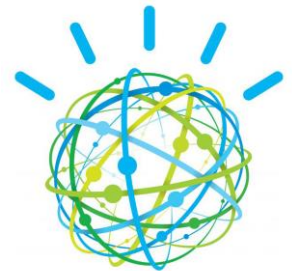
Platform.



Method.



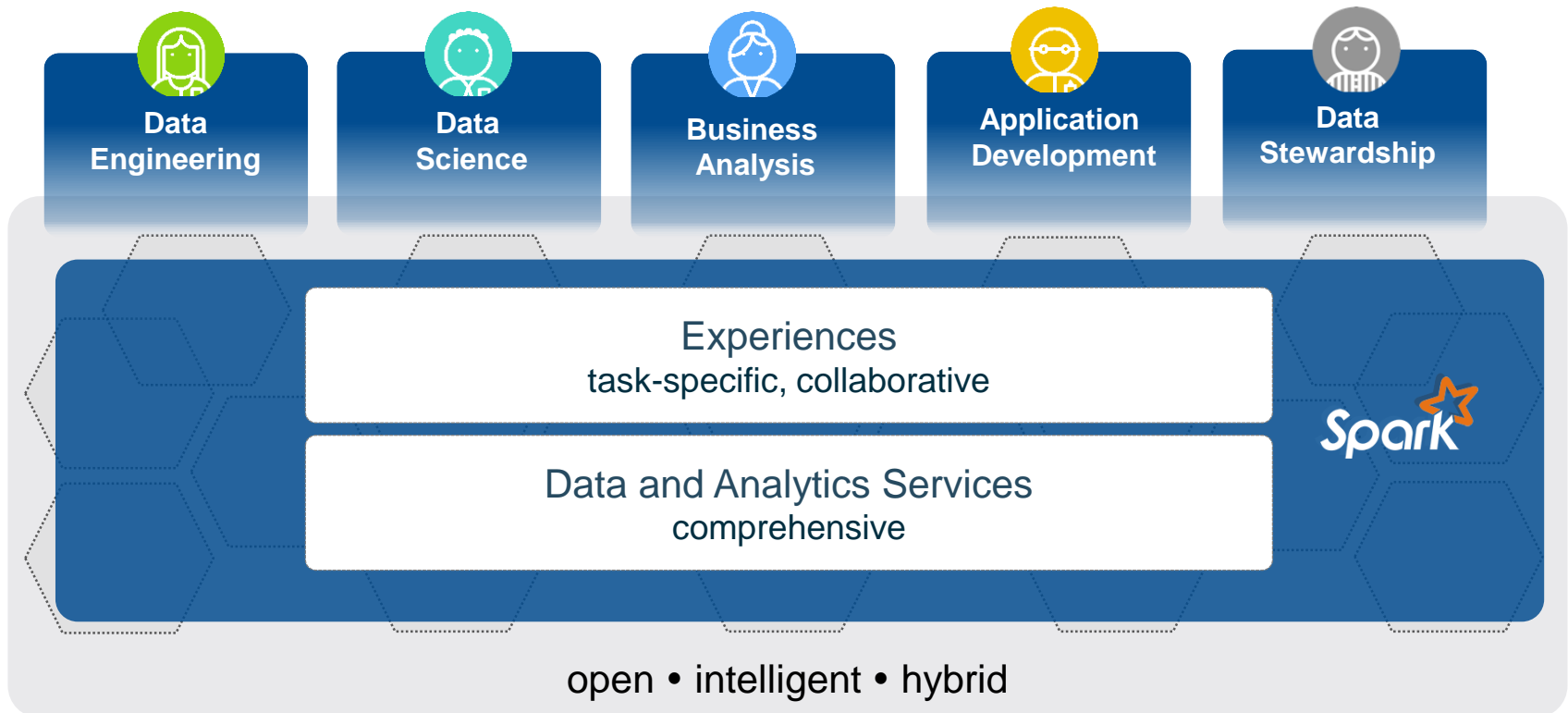
Ecosystem.



<http://ibm.co/makedatasimple>

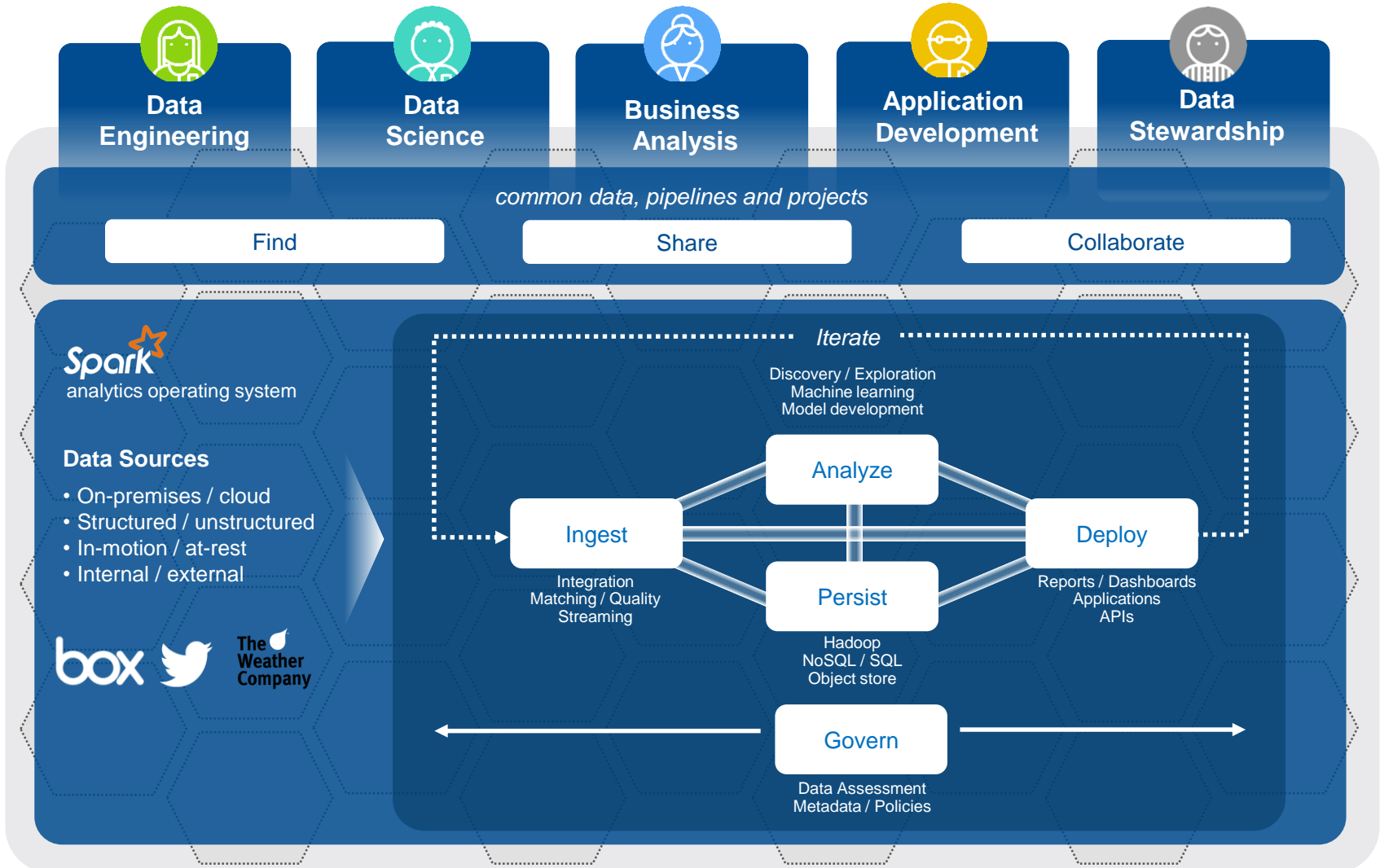
IBM Watson Data Platform

Experience New Ways To Put Data To Work



IBM Watson Data Platform

Connects Users to Data and Analytics

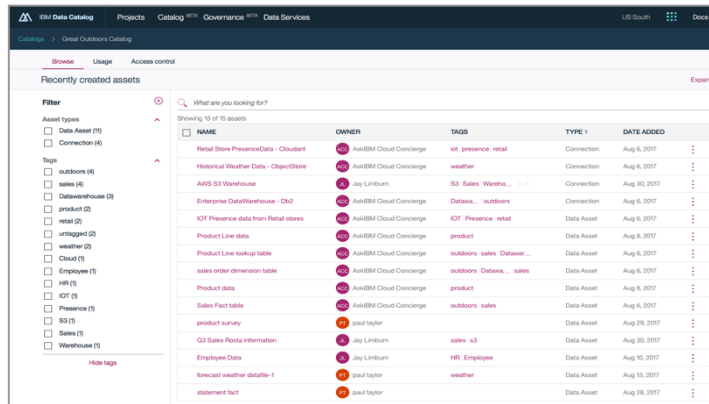


Data Catalog – Open Beta

Unlock tribal knowledge to unleash your data professionals



Data
Steward



IBM Data Catalog Projects Catalog Governance Data Services

Recently created assets

Filter

Asset Type

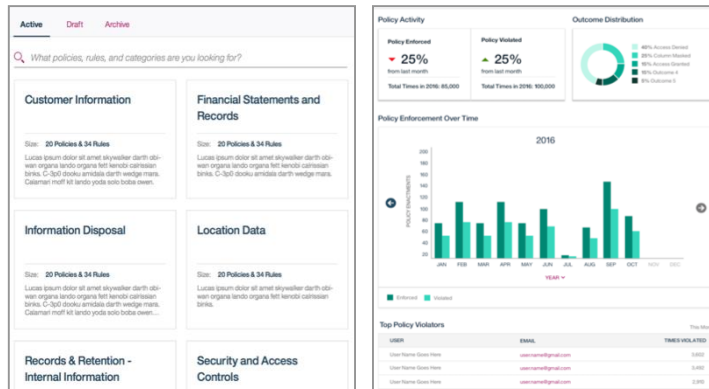
- Data Asset (1)
- Connection (6)

Tags

- outdoors (4)
- sales (6)
- Datalakehouse (3)
- product (2)
- retail (2)
- untagged (2)
- weather (2)
- Cloud (1)
- Employee (1)
- HR (1)
- KOT (1)
- Presence (1)
- ES (1)
- Sales (1)
- Warehouse (1)

Hide tags

NAME	OWNER	TAGS	TYPE	DATE ADDED
Retail Store Presence Data - Cloudant	Aksham Cloud Conco	ret, presence, retail	Connection	Aug 6, 2017
Historical Weather Data - ObjectStore	Aksham Cloud Conco	weather	Connection	Aug 6, 2017
AWB S3 Warehouse	Jay Limbun	S3 Sales, Warehouse	Connection	Aug 30, 2017
Enterprise Data Warehouse - Clou	Aksham Cloud Conco	Default, outdoors	Connection	Aug 6, 2017
KOT Presence data from Retail stores	Aksham Cloud Conco	KOT, Presence, retail	Data Asset	Aug 6, 2017
Product Line data	Aksham Cloud Conco	product	Data Asset	Aug 6, 2017
Product Line lookup table	Aksham Cloud Conco	outdoors, sales, Datalake...	Data Asset	Aug 6, 2017
sales order dimension table	Aksham Cloud Conco	outdoors, Datalake, sales	Data Asset	Aug 6, 2017
Product data	Aksham Cloud Conco	product	Data Asset	Aug 6, 2017
Sales Fact table	Aksham Cloud Conco	outdoors, sales	Data Asset	Aug 6, 2017
product survey	paal taylor		Data Asset	Aug 29, 2017
Q3 Sales Results Information	Jay Limbun	sales, s3	Data Asset	Aug 30, 2017
Employee Data	Jay Limbun	HR, Employee	Data Asset	Aug 10, 2017
forecast weather details-1	paal taylor	weather	Data Asset	Aug 18, 2017
statement fact	paal taylor		Data Asset	Aug 26, 2017



Active Draft Active

What policies, rules, and categories are you looking for?

Customer Information

Size: 20 Policies & 34 Rules

Lucas poeun doer all amet skywalker dorth oba-wan organa londo organa fett kenderi cappelisan binks C-3PO droids amida darth wedge mauls Calamari moff kit lando poda solo boba owen.

Financial Statements and Records

Size: 20 Policies & 34 Rules

Lucas poeun doer all amet skywalker dorth oba-wan organa londo organa fett kenderi cappelisan binks C-3PO droids amida darth wedge mauls Calamari moff kit lando poda solo boba owen.

Information Disposal

Size: 20 Policies & 34 Rules

Lucas poeun doer all amet skywalker dorth oba-wan organa londo organa fett kenderi cappelisan binks C-3PO droids amida darth wedge mauls Calamari moff kit lando poda solo boba owen.

Location Data

Size: 20 Policies & 34 Rules

Lucas poeun doer all amet skywalker dorth oba-wan organa londo organa fett kenderi cappelisan binks C-3PO droids amida darth wedge mauls Calamari moff kit lando poda solo boba owen.

Records & Retention - Internal Information

Security and Access Controls

Policy Activity

Policy Enforced: 25% (from last month)

Policy Violated: 25% (from last month)

Outcome Distribution

Policy Enforcement Over Time

2016

Top Policy Violators

USER	EMAIL	TIMES VIOLATED
User Name Given Here	username@gmail.com	5,000
User Name Given Here	username@gmail.com	5,000
User Name Given Here	username@gmail.com	2,000

Discover

Automatic data profiling and classification, using machine learning to categorize data and assign metatags to quickly organize it into searchable stores.

Catalog

Cloud-based enterprise metadata repository that lets you securely catalog your data sources and assets

Govern

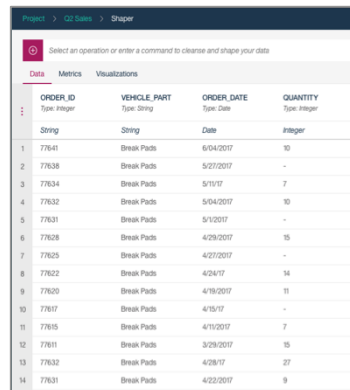
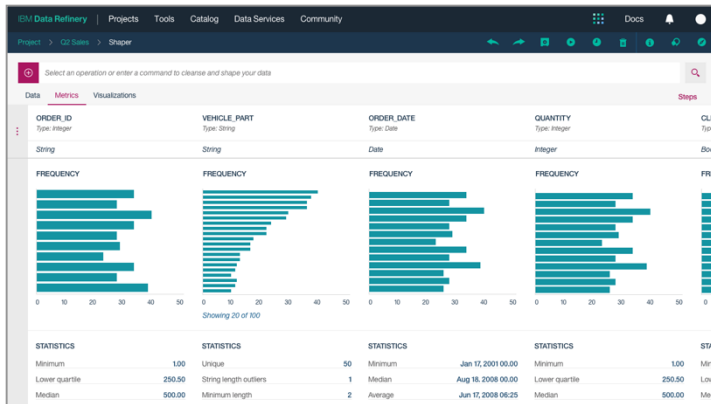
Robust governance capabilities let you define and enforce policies so that the right data are being accessed by the right people



Data
Engineer

Data Refinery – Open Beta

A Breakthrough Approach to Explore and Prepare Data



The screenshot shows the IBM Data Refinery interface with the 'Data' tab active. It displays a table with columns: ORDER_ID (Type: Integer), VEHICLE_PART (Type: String), ORDER_DATE (Type: Date), and QUANTITY (Type: Integer). The table contains 14 rows of data.

ORDER_ID	VEHICLE_PART	ORDER_DATE	QUANTITY
77641	Break Pads	6/04/2017	10
77638	Break Pads	5/27/2017	-
77634	Break Pads	5/19/17	7
77632	Break Pads	5/04/2017	10
77631	Break Pads	5/1/2017	-
77628	Break Pads	4/29/2017	15
77625	Break Pads	4/27/2017	-
77622	Break Pads	4/24/17	14
77620	Break Pads	4/19/2017	11
77617	Break Pads	4/15/17	-
77615	Break Pads	4/10/2017	7
77611	Break Pads	3/29/2017	15
77632	Break Pads	4/28/17	27
77631	Break Pads	4/22/2017	9



Wrangle

Interactively explore, resolve quality issues, enrich, standardize, and summarize data.

Flow

Create data flows visually, schedule for repeatability, monitor and notify

Adapt

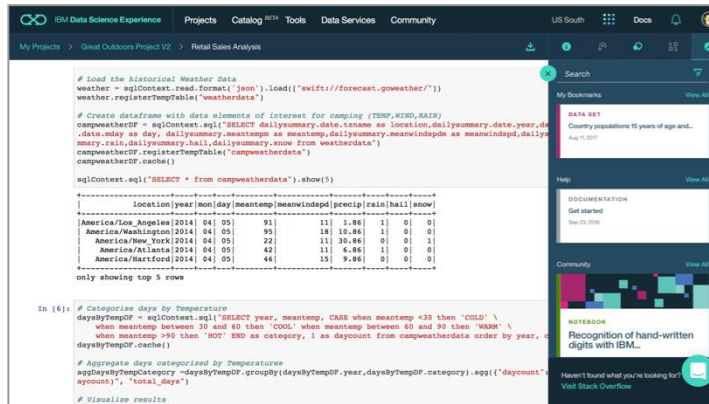
Connect to 30+ cloud and on-premises stores and scale on demand with cataloging and governance

Data Science Experience



Data
Scientist

Brings together everything a Data Scientist needs to be successful



```
# Load the historical weather data
weather = eqContext.read.format('json').load(['swift://forecast.goweather'])
weather.registerTempTable('weatherdata')

# Create dataset with data elements of interest for camping (TEMP, WIND, RAIN)
campweatherDF = eqContext.sql('SELECT dailysummary.date, time as location, dailysummary.date, time as day, dailysummary.meantemp as meantemp, dailysummary.meandepth as meandepth, dailysummary.rain, dailysummary.hail, dailysummary.snow from weatherdata')
campweatherDF.registerTempTable('campweatherdata')
campweatherDF.cache()

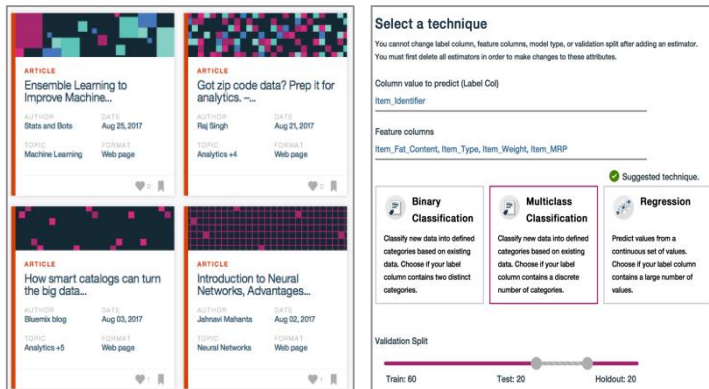
eqContext.sql('SELECT * from campweatherdata').show(5)

+-----+-----+-----+-----+-----+-----+
| location|year|month|day|meantemp|meandepth|precip|rain|hail|snow|
+-----+-----+-----+-----+-----+-----+
| America/Tokyo|2014|04|05| 91| 11| 3.86| 1| 0| 0|
| America/Washington|2014|04|05| 95| 18| 10.86| 1| 0| 0|
| America/New York|2014|04|05| 22| 11| 30.86| 0| 0| 1|
| America/Atlanta|2014|04|05| 42| 11| 6.86| 1| 0| 0|
| America/Bartford|2014|04|05| 46| 15| 9.86| 0| 0| 0|
+-----+-----+-----+-----+-----+-----+
only showing top 5 rows

In [6]: # Categorize days by Temperature
daysByTempDF = eqContext.sql('SELECT year, meantemp, CASE WHEN meantemp < 30 THEN 'COLD' \
WHEN meantemp BETWEEN 30 AND 60 THEN 'COOL' WHEN meantemp BETWEEN 60 AND 90 THEN 'WARM' \
WHEN meantemp > 90 THEN 'HOT' END as category, 1 as daycount from campweatherdata order by year, dayByTempDF.cache()

# Aggregate days categorized by Temperatures
aggDaysByTempCategory = daysByTempDF.groupBy(daysByTempDF.year, daysByTempDF.category).agg('count(*) as count', 'sum(daycount) as total_days')

# Visualize results
```



Select a technique

You cannot change label column, feature columns, model type, or validation split after adding an estimator. You must first delete all estimators in order to make changes to these attributes.

Column value to predict (Label Col)
Item_Identifier

Feature columns
Item_Fat_Content, Item_Type, Item_Weight, Item_MRP

Suggested technique.

- Binary Classification**
Classify new data into defined categories based on existing data. Choose if your label column contains two distinct categories.
- Multiclass Classification**
Classify new data into defined categories based on existing data. Choose if your label column contains a discrete number of categories.
- Regression**
Predict values from a continuous set of values. Choose if your label column contains a large number of values.

Validation Split
Train: 60 Test: 20 Holdout: 20

Learn

Built-in learning to get started or go the distance with advanced tutorials

Create

The best of open source and IBM value-add to create state-of-the-art data products

Collaborate

Data and Analytic assets are contained within projects which can be shared with other users.

IBM Cloud PaaS

Rich Platform and Service APIs for your developers

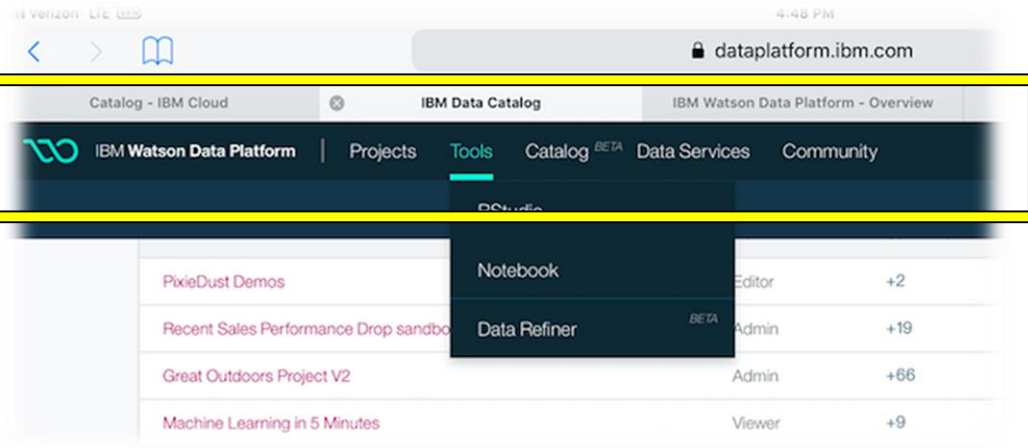


Application
Developer

The screenshot displays the IBM Cloud API Docs interface in a web browser. The browser's address bar shows the URL: `https://console.bluemix.net/apidocs/1084-watson-data-platform-core-services-v2?&language=node#introduction`. The page title is "IBM Cloud API Docs". On the left, there is a navigation sidebar with sections: "OVERVIEW" (containing links for Introduction, Authentication, Versioning, Sorting, Filtering, Searching, Rate Limiting, and Error Handling) and "REFERENCE" (containing a link for Analytics Engines). The main content area is titled "Retrieve details of specified customization request ID." and includes a description: "Retrieves the status of running the specified customization request, along with pointers to log files generated during the run." Below this, there is a "Request" section with two sub-sections: "CUSTOM HEADERS" and "PATH PARAMETERS". The "CUSTOM HEADERS" section lists "Authorization" as a string representing an Identity Access Management (IAM) bearer token. The "PATH PARAMETERS" section lists "instance_guid" as a service instance guid and "request_id" as a customization request id. To the right of the "Request" section, there is an "Example Request" section showing a Node.js code snippet for making a GET request to the API endpoint: `/v2/analytics_engines/{instance_guid}/customization_requests/{request_id}`. The code snippet includes the following logic:

```
var http = require("https");
var options = {
  "method": "GET",
  "hostname": "api.dataplatform.ibm.com",
  "port": null,
  "path": "/v2/analytics_engines/{instance_guid}/customization_requests/{request_id}",
  "headers": {
    "content-type": "application/json",
    "authorization": "SOME_STRING_VALUE"
  }
};
var req = http.request(options, function (res) {
  var chunks = [];
  res.on("data", function (chunk) {
    chunks.push(chunk);
  });
  res.on("end", function () {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });
});
```

Intelligent data fabric provides consistent platform experience



This fabric remains consistent throughout the Watson Data Platform experience – regardless if you are ingesting data, shaping data, building algorithms, deploying models and more...

How does WDP help fulfill the promise of your data?

Data

Puts every important data source at the fingertips of the teams that need it wherever resides

Governance

Enforces your policies without getting in the way of delivering insights

Skills

Makes the most of the data professionals you have and helps them grow and learn from each other as a team

Infrastructure

Delivers the foundation for your first data project through to the complete transformation of your business

Data Science Experience

Core Attributes of the Data Science Experience



IBM Data Science Experience

Community

- Find tutorials and datasets
- Read articles and papers
- Connect with Data Scientists
- Share comments
- Copy and share notebooks

Open Source

- Code in Scala/Python/R/SQL
- Jupyter Notebooks
- RStudio IDE and Shiny
- Apache Spark
- Your favorite libraries

IBM Added Value


- IBM Machine Learning
- SPSS Modeler Canvas
- Prescriptive Analytics - DOpnexcloud
- Projects and Version Control
- Managed Spark Service

Powered by IBM **Watson Data Platform**

DSX Architecture

DSX architecture

Last updated: June 27, 2017

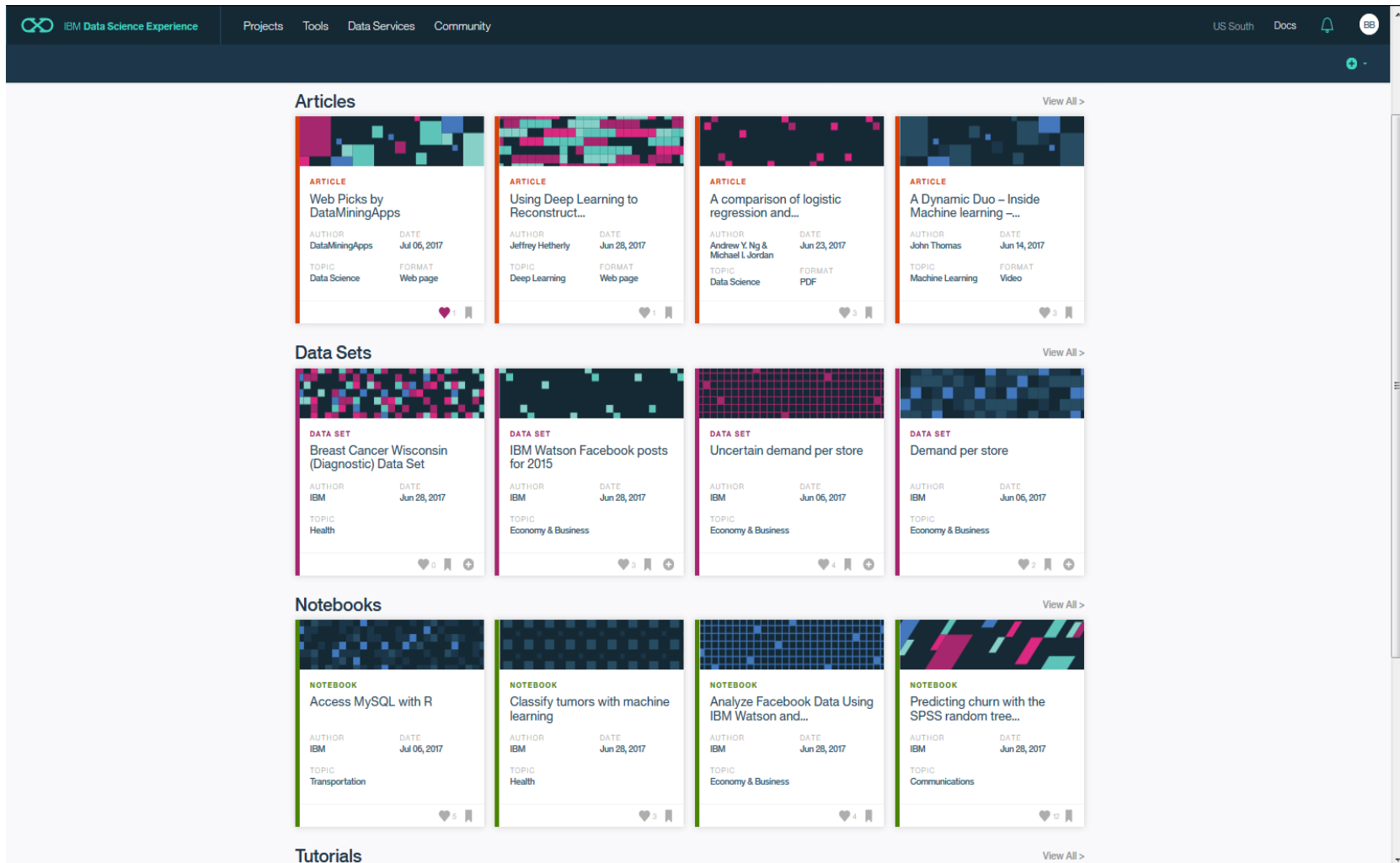
 Search this document



DSX provides you with the environment and tools to solve your business problems by collaboratively analyzing data. This illustration shows how the architecture of DSX is centered around the project. A project is how you organize your resources for solving a business problem.



Community Cards provide in-context learning

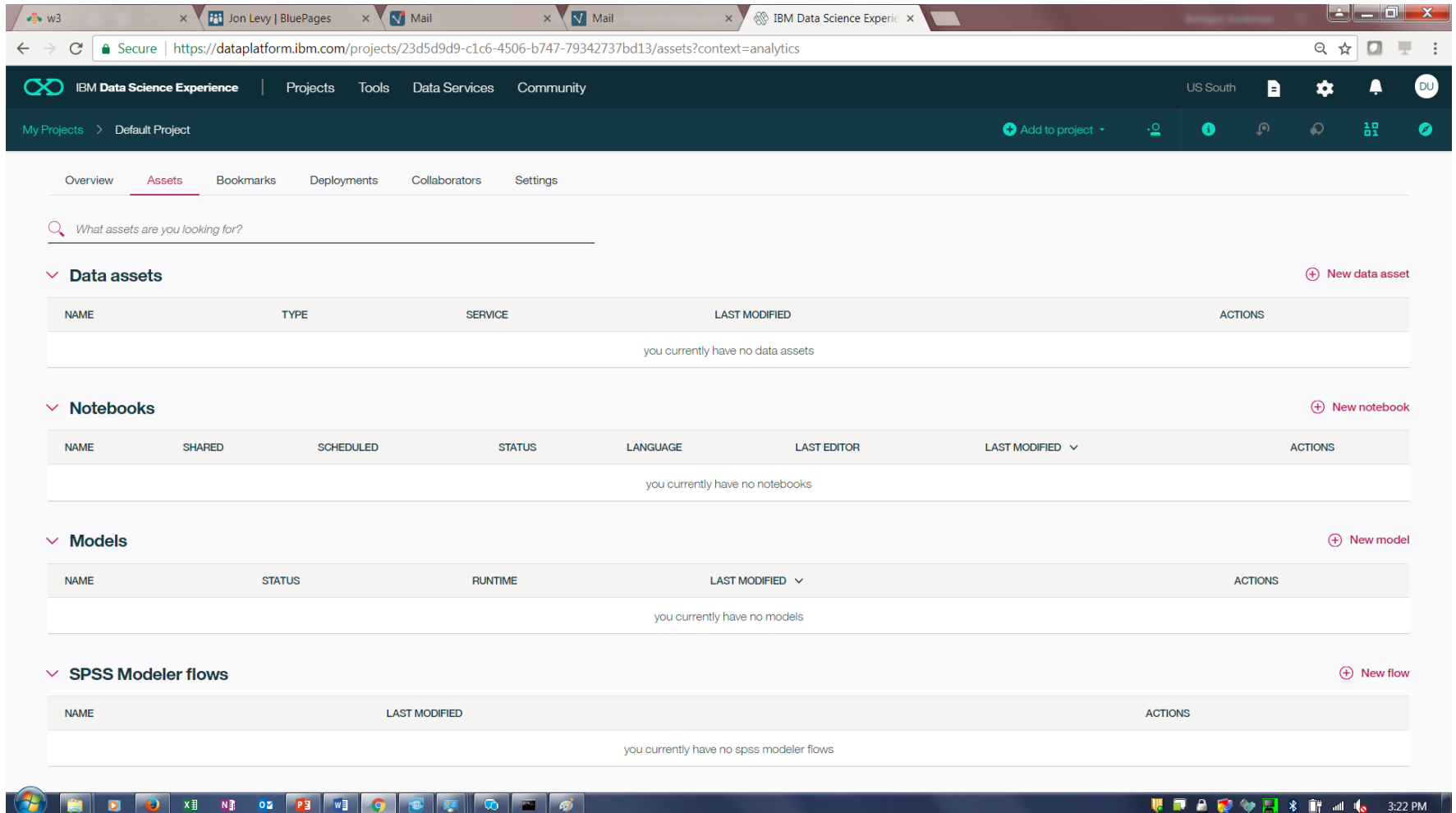


The screenshot displays the IBM Data Science Experience Community Cards interface. The top navigation bar includes the IBM Data Science Experience logo, a menu with Projects, Tools, Data Services, and Community, and user information (US South, Docs, a bell icon, and a profile icon labeled BB).

The main content area is divided into four sections, each with a "View All >" link:

- Articles:** Displays four article cards. Each card includes a title, author, date, topic, format, and a heart icon for likes.
 - Article 1: "Web Picks by DataMiningApps" by DataMiningApps, dated Jul 06, 2017, Topic: Data Science, Format: Web page, 1 like.
 - Article 2: "Using Deep Learning to Reconstruct..." by Jeffrey Hetherly, dated Jun 28, 2017, Topic: Deep Learning, Format: Web page, 3 likes.
 - Article 3: "A comparison of logistic regression and..." by Andrew Y. Ng & Michael L. Jordan, dated Jun 23, 2017, Topic: Data Science, Format: PDF, 3 likes.
 - Article 4: "A Dynamic Duo – Inside Machine learning ..." by John Thomas, dated Jun 14, 2017, Topic: Machine Learning, Format: Video, 3 likes.
- Data Sets:** Displays four data set cards. Each card includes a title, author, date, topic, and a heart icon for likes.
 - Data Set 1: "Breast Cancer Wisconsin (Diagnostic) Data Set" by IBM, dated Jun 06, 2017, Topic: Health, 0 likes.
 - Data Set 2: "IBM Watson Facebook posts for 2015" by IBM, dated Jun 28, 2017, Topic: Economy & Business, 3 likes.
 - Data Set 3: "Uncertain demand per store" by IBM, dated Jun 06, 2017, Topic: Economy & Business, 4 likes.
 - Data Set 4: "Demand per store" by IBM, dated Jun 06, 2017, Topic: Economy & Business, 2 likes.
- Notebooks:** Displays four notebook cards. Each card includes a title, author, date, topic, and a heart icon for likes.
 - Notebook 1: "Access MySQL with R" by IBM, dated Jul 06, 2017, Topic: Transportation, 5 likes.
 - Notebook 2: "Classify tumors with machine learning" by IBM, dated Jun 28, 2017, Topic: Health, 3 likes.
 - Notebook 3: "Analyze Facebook Data Using IBM Watson and..." by IBM, dated Jun 28, 2017, Topic: Economy & Business, 4 likes.
 - Notebook 4: "Predicting churn with the SPSS random tree..." by IBM, dated Jun 28, 2017, Topic: Communications, 2 likes.
- Tutorials:** This section is partially visible at the bottom of the screenshot.

Collaborate Using Projects



The screenshot shows the IBM Data Science Experience interface. The browser address bar displays the URL: <https://dataplatfrom.ibm.com/projects/23d5d9d9-c1c6-4506-b747-79342737bd13/assets?context=analytics>. The page header includes the IBM Data Science Experience logo and navigation links: Projects, Tools, Data Services, and Community. The user is logged in as 'DU' (David Uhlir) in the 'US South' region. The main navigation bar shows 'My Projects' and 'Default Project'. The 'Assets' tab is selected, displaying a search bar with the placeholder text 'What assets are you looking for?'. Below the search bar, there are four sections: 'Data assets', 'Notebooks', 'Models', and 'SPSS Modeler flows'. Each section has a table with columns for asset details and an 'Actions' column. All sections indicate that there are currently no assets, notebooks, models, or flows.

Overview **Assets** Bookmarks Deployments Collaborators Settings

What assets are you looking for?

▼ Data assets + New data asset

NAME	TYPE	SERVICE	LAST MODIFIED	ACTIONS
you currently have no data assets				

▼ Notebooks + New notebook

NAME	SHARED	SCHEDULED	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
you currently have no notebooks							

▼ Models + New model

NAME	STATUS	RUNTIME	LAST MODIFIED	ACTIONS
you currently have no models				

▼ SPSS Modeler flows + New flow

NAME	LAST MODIFIED	ACTIONS
you currently have no spss modeler flows		

Add Collaborators to a Project

Add New Collaborator

Add users to your project for collaboration. Users with write access can add services to your project...

Type name or email address

Select

Viewer



Editor


Admin

Cancel

Add

GitHub Integration



Data Science Experience 

Settings

Integrations

[Profile](#)[Services](#)[Integrations](#)

GitHub Integration

Want to publish your notebooks on GitHub?

Before you can publish to GitHub, you need to create an access token. Visit [GitHub personal access tokens](#), select repo scope and generate a token.

Paste generated personal access token here

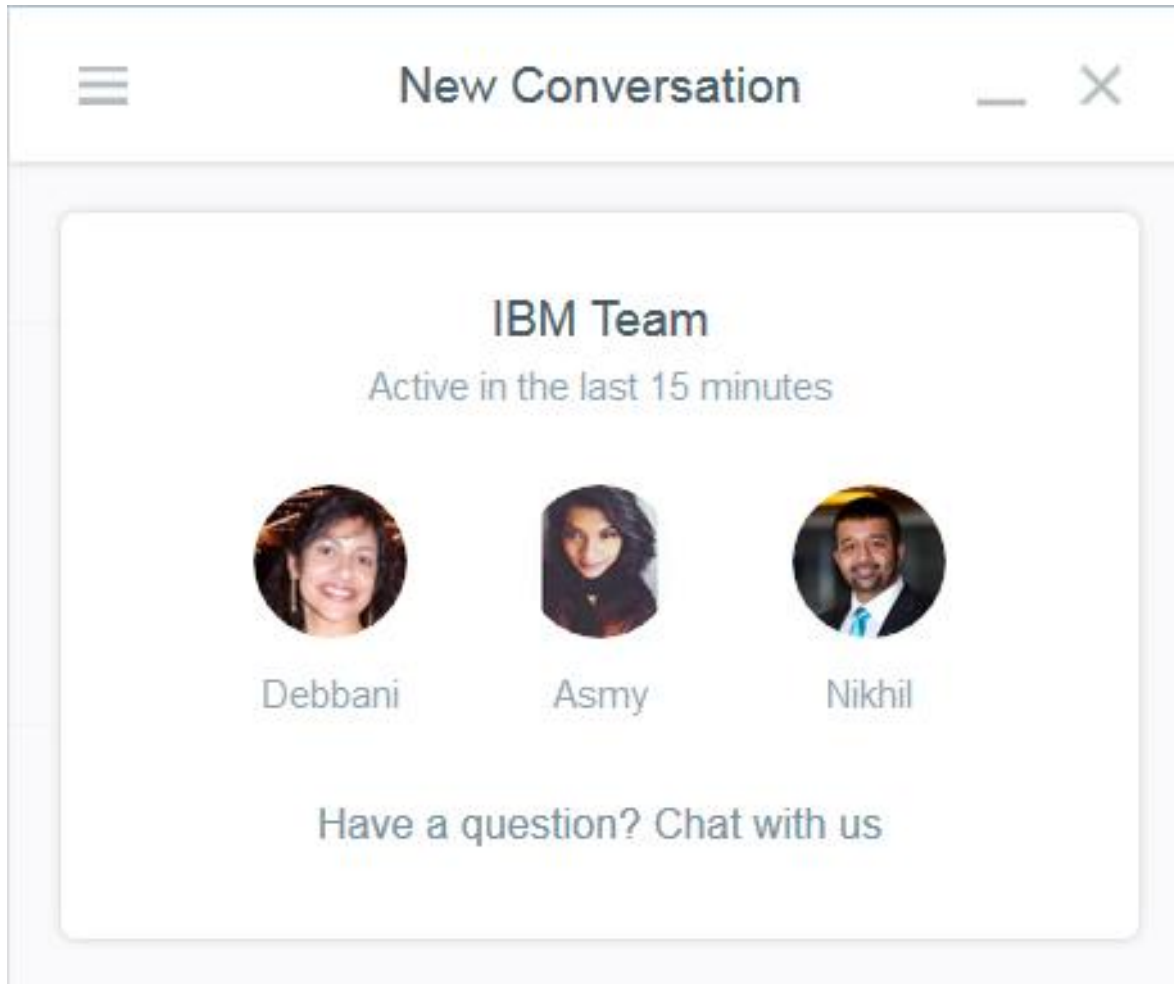
40

Clear

Save

After the access token is saved, a GitHub repository can be connected to a project on the project's Settings page.

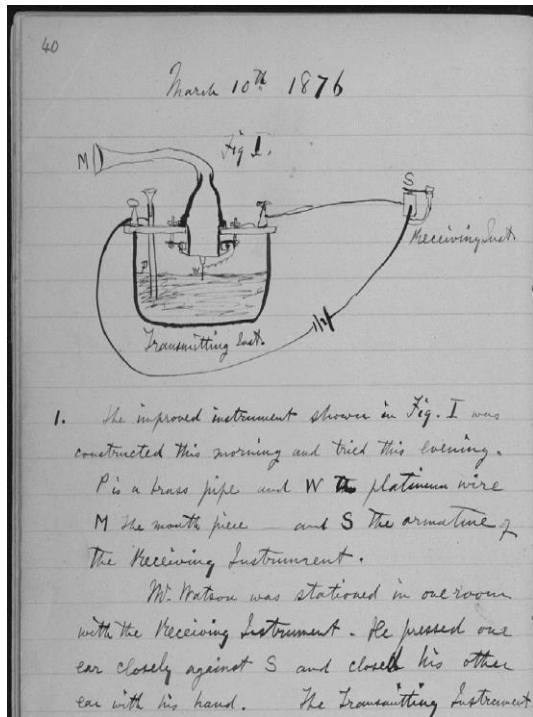
Live chat on Intercom for support from the IBM team and to provide your feedback on how we can improve



What is a “Notebook”?

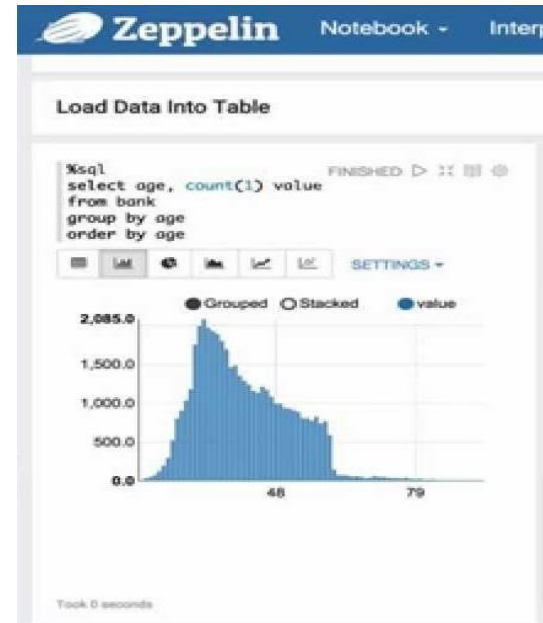
Pen and Paper

- Pen and paper has long provided the rich experience that scientists need to document progress through notes and drawings:
 - Expressive
 - Cumulative
 - Collaborative

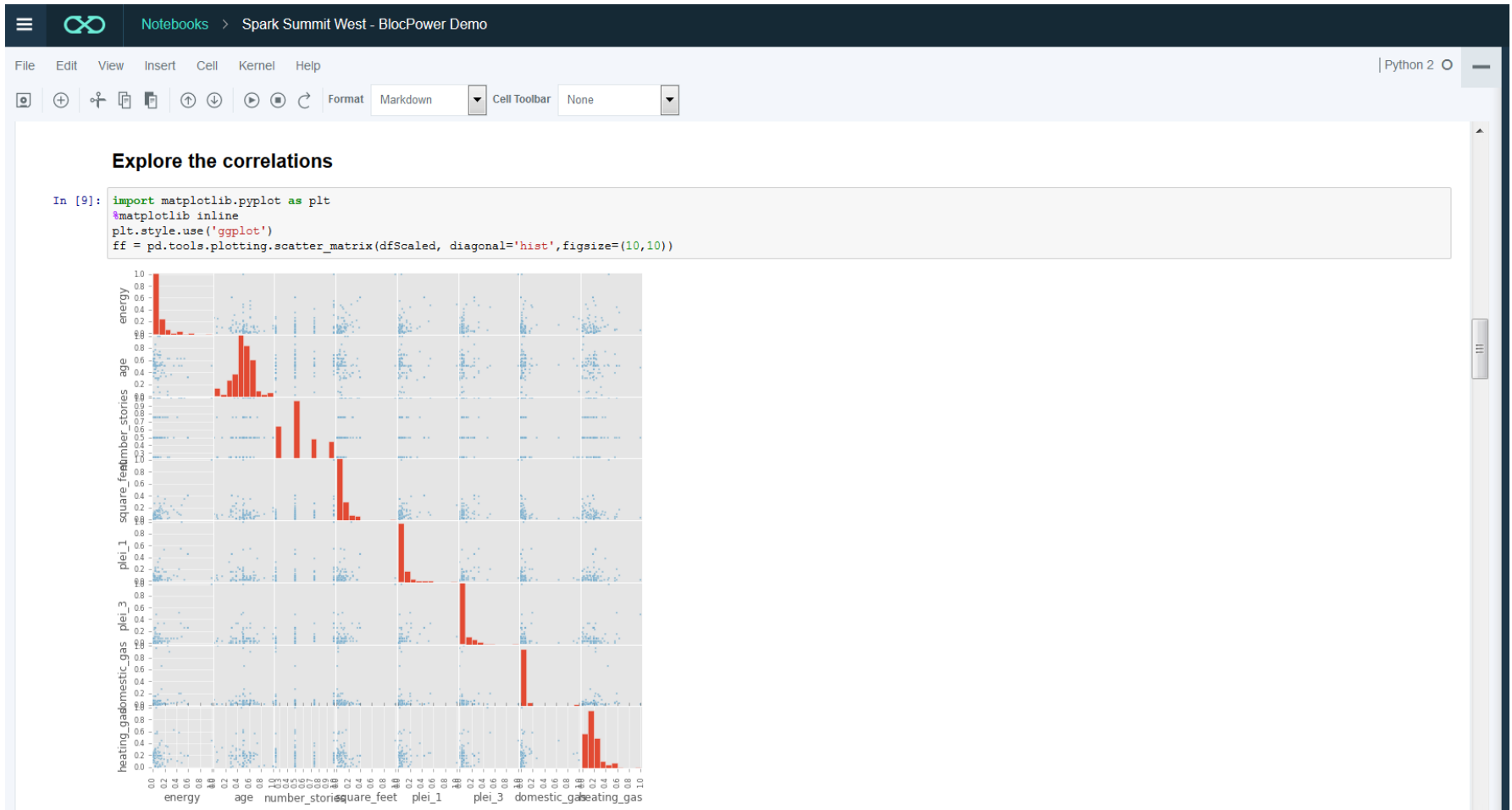


Notebooks

- Notebooks are the digital equivalent of the “pen and paper” lab notebook, enabling data scientists to document reproducible analysis:
 - Markdown and visualization
 - Iterative exploration
 - Easy to share



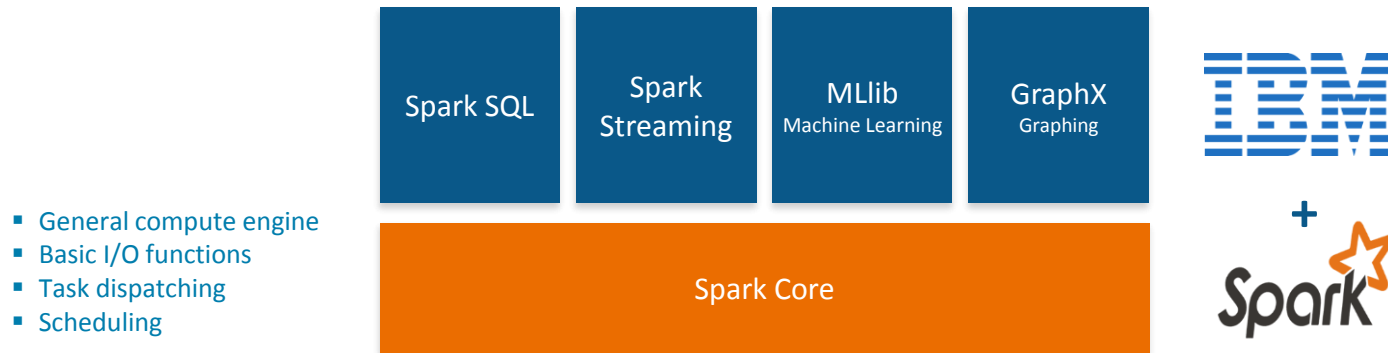
Integrated Jupyter Notebooks for interactive and collaborative development - seamless execution on Spark



From a Notebook in DSX you can use IBM's managed Spark Service to blend multiple data types, sources, and workloads



Benefits of Spark for Data Science



- Allows Data Scientists to code at scale
 - In-Memory processing that scales in a distributed architecture
- Supports multiple programming interfaces (Scala, Python, Java and R)
- Provides unified APIs (SQL, Streaming, Machine Learning, etc.)

The Spark service uses Bluemix Object Storage as its preferred data store for building performant applications



- Object storage provides **inexpensive, scalable and self-healing** retention of massive amounts of unstructured data
- Every object exists at the same level in a **flat address space**
- Bluemix Object Storage has a **drag-and-drop** upload and **Swift API** for programmatic access













Object Storage
IBM

Supported Data Sources via on-premises and cloud Connectors

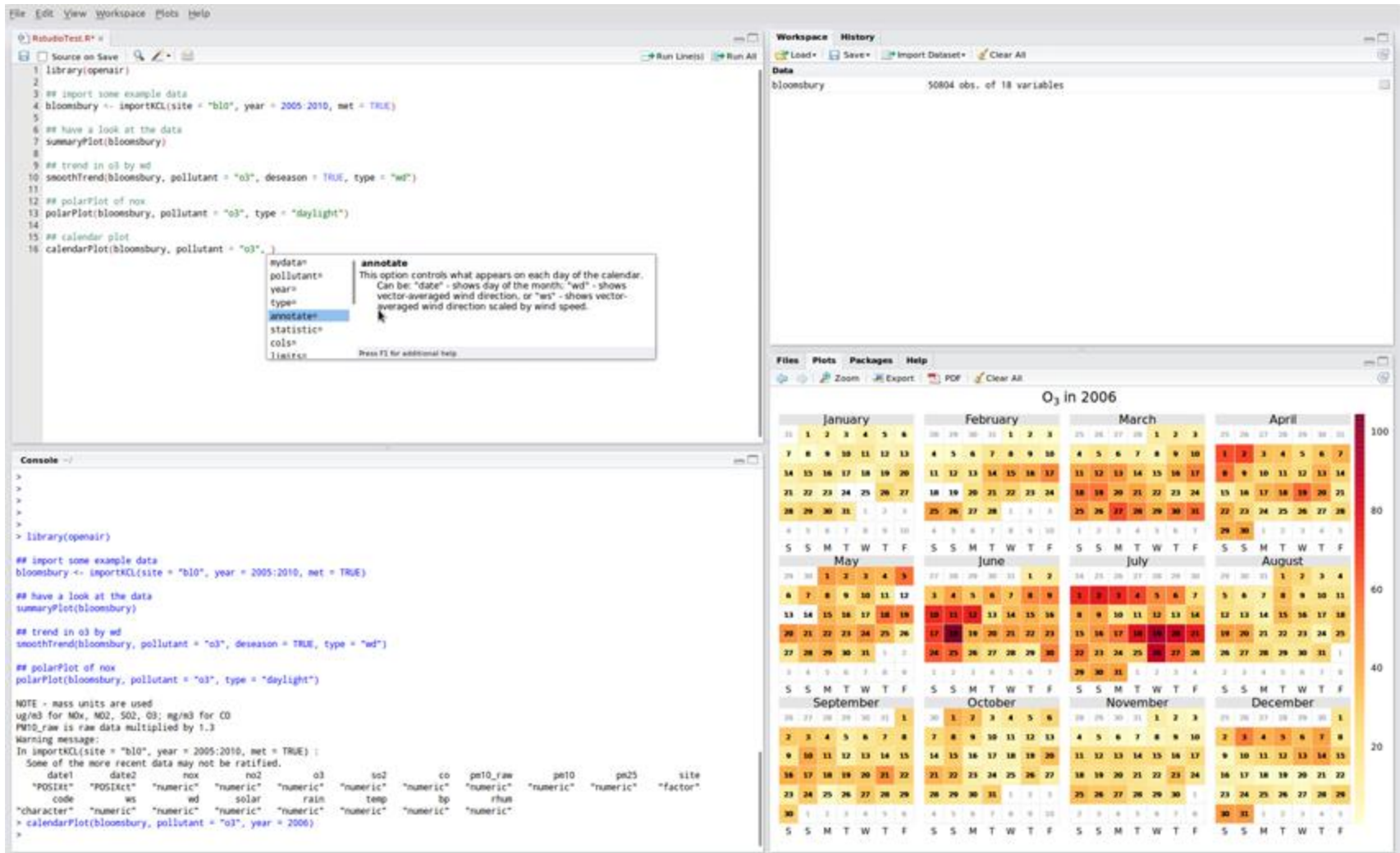
IBM services in IBM Cloud

 IBM Informix	 PostgreSQL on Compose	 MySQL on Compose	 Cloud Object Storage
 IBM Db2 for i	 IBM Cloudant	 Cloud Object Storage (IaaS)	 IBM Db2 on Cloud
 Object Storage OpenStack Swift for IBM Cloud	 IBM Db2	 IBM BigInsights HDFS	 IBM Db2 Hosted
 Object Storage OpenStack Swift (IaaS)	 IBM PureData for Analytics	 IBM Db2 for z/OS	 IBM Db2 Warehouse on Cloud

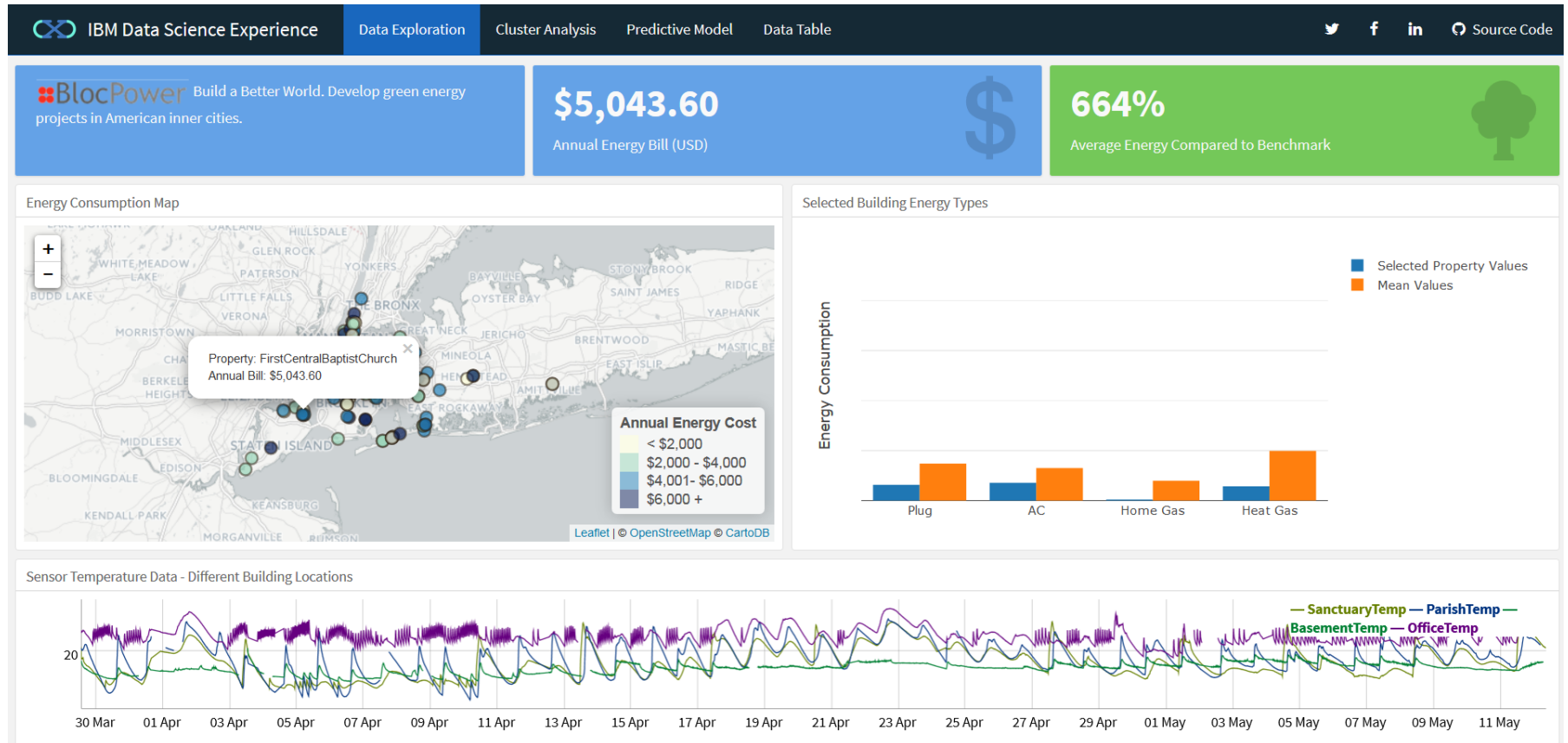
Third-party services

 Cloudera Impala	 Salesforce.com	 Apache Hive	 Amazon Redshift
 Microsoft SQL Server	 Sybase IQ	 Sybase	 Oracle
 Amazon S3	 MySQL	 Hortonworks HDFS	 PostgreSQL
 Pivotal Greenplum	 Microsoft Azure SQL Database		

DSX has RStudio built into the experience thanks to our strategic partnership



With RStudio you can create Shiny web applications to make your analysis accessible to the business



Operationalize insights with IBM Machine Learning

IBM Machine Learning



Data Access:

- Easily connect to Behind-the-Firewall and Public Cloud Data
- Catalogued and Governed Controls through Watson Data Platform

Creating Models:

- Single UI and API for creating ML Models on various Runtimes
- Auto-Modeling and Hyperparameter Optimization

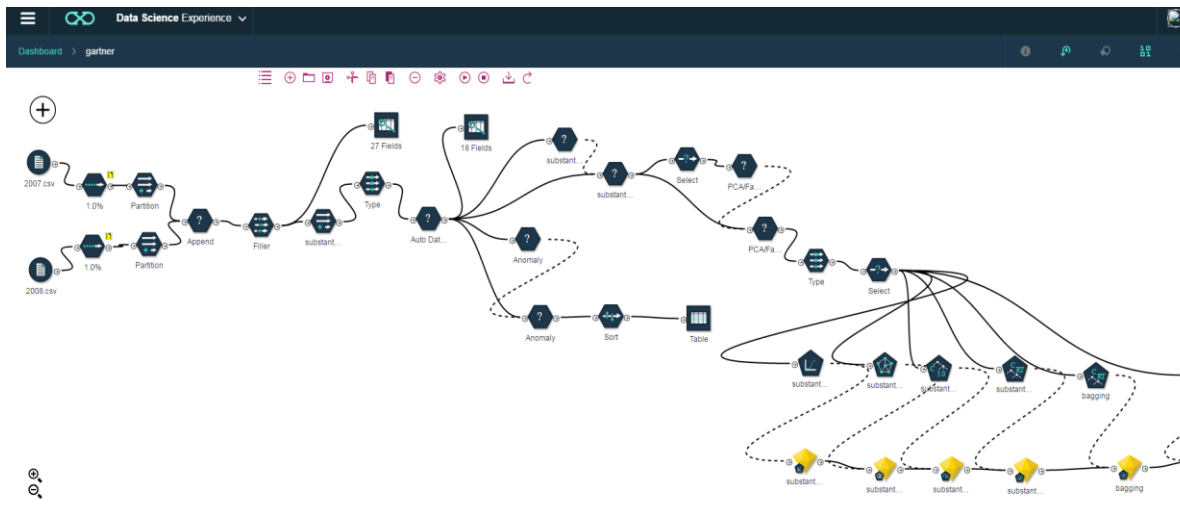
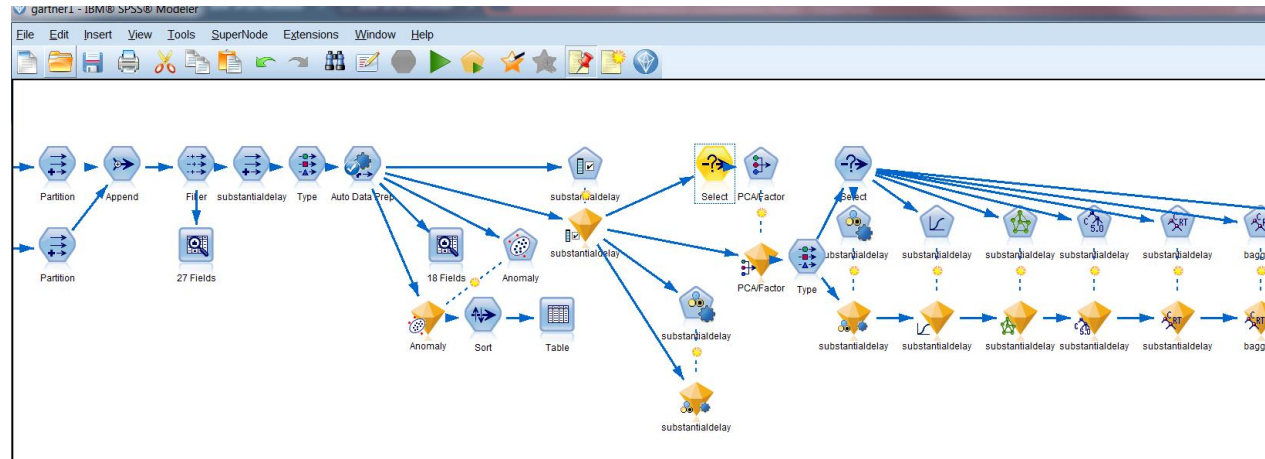
Web Service:

- Real-time, Streaming, and Batch Deployment
- Continuous Monitoring and Feedback Loop

Intelligent Apps:

- Integrate ML models with apps, websites, etc.
- Continuously Improve and Adapt with Self-Learning

- DSX Canvas will have compatibility with legacy SPSS Modeler streams
- Multiple execution runtimes: SPSS Modeler, SparkML
- Planned support for R/Python/SQL code



- Pipeline deployment from DSX Canvas (left) via IBM Machine Learning

Stream Designer – Open Beta

- Characteristics of Stream Processing
 - Continuous processing
 - Multiple varied data sources
 - High data rates/ data volumens
 - Near-real time action
- DSX Stream Designer
 - Design stream flow with new Stream Designer
 - Executes in Streaming Analytics Service (based on IBM Streams)
 - Can invoke stream within Jupyter notebooks using Stream API



DSX Local

- **Very similar to the public cloud version of DSX**
- **Runs on hardware that is provided by the customer**
 - The DSX Local software and hardware are managed by the customer
- **DSX Local comes with all the software it needs to run, although it can integrate with existing customer systems such as**
 - Databases and HDFS storage
 - LDAP servers for authentication



IBM Data Science Experience
<https://www.youtube.com/watch?v=1HjzkLRdP5k&t=29s>

Labs

Lab Overview

Use IBM's Data Science Experience (DSX) and IBM cloud services to create a working cloud-based application from start to finish. Participants will be led through a series of four labs. The first three build upon one another so it is important that they are completed in order.

- Lab 1 - The first lab will begin with loading raw delimited data into DB2 Warehouse for Cloud and interacting with that data from a Jupyter notebook in DSX with python.
- Lab 2 - The second lab will leverage Spark machine learning (SparkML) on the loaded data to create categorical predictions using pyspark and a supervised learning model and store the results back to the database.
- Lab 3 - The third lab will guide participants in creating an R notebook and Shiny UI in DSX using RStudio.

Choose one of the two below:

- Lab-4a - This lab will use the Watson Machine Learning capability to create a machine learning model based on the Titanic data set. The model will be deployed in the IBM Cloud, and an application will be built that uses the deployed machine learning model to predict survivability given passenger characteristics.
- Lab 4b - The second lab will use the SPSS Modeler Flow designer to cleanse and prepare the Titanic data set for modeling. A Logistic Regression model will be trained and evaluated to predict survivability given passenger characteristics.

Lab 1

This lab will begin with loading raw delimited data into DB2 Warehouse for Cloud and interacting with that data from a Jupyter notebook in DSX with Python.

Objectives:

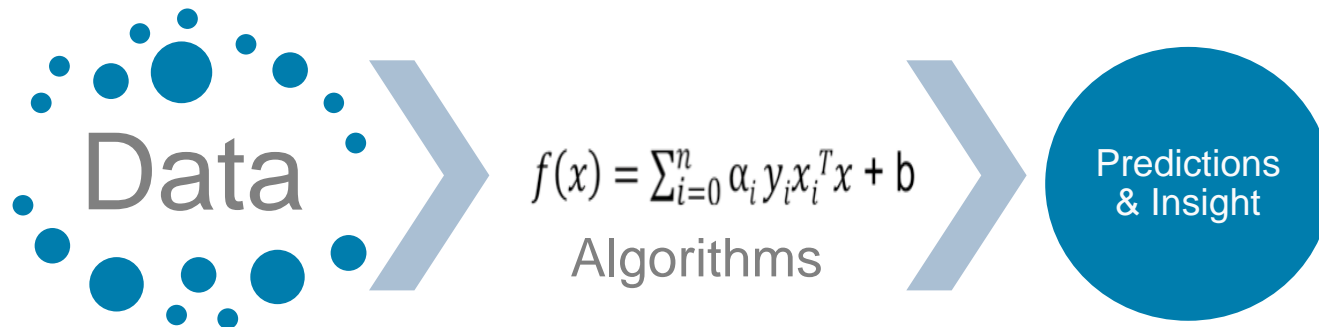
- Upon completing the lab, you will know how to:
 - Create a Jupyter IPython notebook from a URL
 - Establish a connection to DB2 Warehouse on Cloud
 - Use a dataframe to read and manipulate tables
 - Use SQL to query the database
 - Explore the data using techniques from earlier in the lab
 - Close the database connection

Lab 2

In this lab, you will use SparkML in IBM Data Science Experience to run generated travel data through a machine learning algorithm, automatically tune the algorithm, and load the data into a DB2 Warehouse database.

What is Machine Learning?

*“Computers that learn without being **explicitly programmed**”*
*“Using **algorithms** to understand patterns in data”*



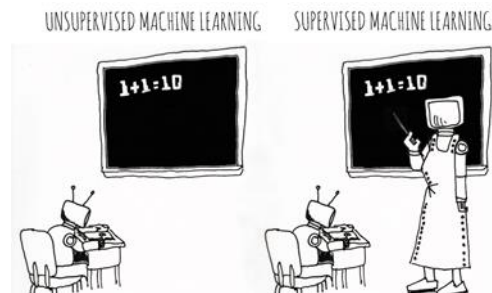
Categories of Machine Learning

■ Supervised learning

- The program is “trained” on a pre-defined set of “training examples”, which then facilitate its ability to reach an accurate conclusion when given new data
- The algorithm is presented with example inputs and their outcomes (labels)
- The goal is to learn a general rule that maps inputs to outputs

■ Unsupervised learning

- No labels are given to the learning algorithm, leaving it on its own to find structure (patterns and relationships) in its input



Categories of Machine Learning

Technique	Usage	Algorithms
Classification (or prediction)	<ul style="list-style-type: none">• Used to predict group membership (e.g., will this employee leave?) or a number (e.g., how many widgets will I sell?)	<ul style="list-style-type: none">• Decision Trees• Logistic Regression• Random Forests• Naïve Bayes• Linear Regression• Lasso Regressionetc
Segmentation	<ul style="list-style-type: none">• Used to classify data points into groups that are internally homogenous and externally heterogeneous.• Identify cases that are unusual	<ul style="list-style-type: none">• K-means• Gaussian Mixture• Latent Dirichlet allocationetc
Association	<ul style="list-style-type: none">• Used to find events that occur together or in a sequence (e.g., market basket)	<ul style="list-style-type: none">• FP Growth

Known as:

- Scale variables:

- ### Categorical variables:

- Known as:

- © 2017 IBM Corporation

Training, testing, & validation sets

- **During the model development process, supervised learning techniques employ training and testing sets and sometimes a validation set.**
 - Historical data with known outcome
 - Data is randomly split into training, testing, and/or validation sets (mutually exclusive records)
- **Why?**
 - Training set
 - Build the model
 - Tune the parameters
 - Testing set
 - Assess model quality during training/tuning process
 - Avoid overfitting the model to the training set
 - Validation set
 - Estimate accuracy or error rate of model after tuning
 - Used to compare multiple models

Spark ML

- **Spark ML is Spark's machine learning (ML) library**
- **Goal is to make machine learning scalable and easy**
 - No need to understand the detailed math!
- **Divides into two packages:**
 - spark.mllib contains the original API built on top of RDDs
 - spark.ml provides higher-level API built on top of DataFrames for constructing ML pipelines
 - A pipeline is a series of stages where each stage either transforms, or runs through a machine learning algorithm.
- **Using spark.ml is recommended because with DataFrames the API is more versatile and flexible**
 - spark.mllib will continue to be supported

Spark ML Pipeline Terminology

Spark ML standardizes APIs for machine learning algorithms to make it easier to combine multiple algorithms into a single pipeline, or workflow

- **DataFrame**: Spark ML uses DataFrame from Spark SQL as an ML dataset, which can hold a variety of data types
- **Transformer**: A Transformer is an algorithm which can transform one DataFrame into another DataFrame
- **Estimator**: An Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer
- **Pipeline**: A Pipeline chains multiple Transformers and Estimators together in a sequence to specify an ML workflow
- **Parameter**: All Transformers and Estimators share a common API for specifying parameters

Lab 2 – Female Human Trafficking


▪ Input


- Generated fake travel records based on incoming custom forms.
- Subset of records were vetted as “high”, “medium”, or “low” risk for Female Human Trafficking by an analyst.

- **Goal is to train a model on the vetted data to be able to score the unvetted travel records into high, medium, or low categories.**

Lab 2 Data

Field	Description
UUID	Hash-based unique identifier
VETTING_LEVEL	Analyst vetting status : 100- PENDING, 10 – HIGH, 20 – MED, 10 - LOW
NAME	Person name
GENDER	Person Gender
AGE	Person age at time of travel
BIRTH_DATE	Person birth date
BIRTH_COUNTRY	Person full birth country
BIRTH_COUNTRY_CODE	Person ISO 2 country
OCCUPATION	Person occupation as declared on form
ADDRESS	Person US address
SSN	Person Social Security Number
PASSPORT_NUMBER	Person Passport Number
PASSPORT_COUNTRY	Person Passport Issuing Country
PASSPORT_COUNTRY_CODE	Person Passport Issuing Country ISO 2 Code
COUNTRYIES_VISITED	The countries visited as declared on form
COUNTRIES_VISITED_COUNT	The number of countries visited as declared on form
ARRIVAL_AIRPORT_COUNTRY_CODE	ARRIVAL Airport country code ISO2
AIRPORT_ARRIVAL_IATA	ARRIVAL Airport 3 character code
AIRPORT_ARRIVAL_MUNICIPALITY	ARRIVAL Airport Municipality Derived from Code
ARRIVAL_AIRPORT_REGION	ARRIVAL Airport Region Derived from Code
DEPARTURE_AIRPORT_COUNTRY_CODE	DEPARTURE Airport Country code ISO2
DEPARTURE_AIRPORT_IATA	DEPARTURE Airport 3 character code
DEPARTURE_AIRPORT_MUNICIPALITY	DEPARTURE Airport Municipality Derived from Code.

 Target

 Features

Lab 2 Flow

- **Read in dataset as a DataFrame from dashDB**
 - Connect to dashDB
 - Read in the data
- **Identify Labels**
 - Label the data (“VETTING_LEVEL”)
 - Select features
- **Feature Engineering (Transformation)**
 - StringIndexer (occupation, country, gender, birth year variables)
 - VectorAssembler
 - Normalizer
- **Define Model and Setup Pipeline**
 - Naïve Bayes
- **Train the Model**
 - Split input data into Training (70%) and Test (30%) DataFrames
 - Cache the resulting DataFrames
 - Fit the Pipeline to the Training data set



Lab 2 Flow (continued)

- **Evaluate the resulting predictions**
 - Area under the ROC curve

- **Tune the model (hyperparameters)**
 - Build Parameter Grid
 - Cross-evaluate to find the best model

- **Score the unvetted records**
 - Use Best Model to Score unvetted records (VETTING LEVEL == 100)
 - Write results into DashDB table

Classification - Naïve Bayes

- **Two or more outcomes.**
- **Assumes independence among explanatory variables, which is rarely true (thus “naïve”).**
- **Despite its simplicity, often performs very well... widely used.**
- **Significant use cases:**
 - Text categorization (spam vs. legitimate, sports or politics, etc.) using word frequencies as the features
 - Medical diagnosis (e.g., automatic screening)

Lab 3

In this lab, you will learn some of the fundamentals of using RStudio and Shiny in DSX to work and interact with data in DB2 Warehouse and then to create a fully operational "reactive" web application that you can enhance further.

Objectives:

- Upon completing the lab, you will:
 - Create an RStudio project from a Git repository
 - Establish a connection to DB2 Warehouse using an ancillary file
 - Query, join, explore and visualize data in an R notebook
 - Derive categorical names from numerical levels in an R dataframe
 - Use ggplot2 to create bar plots of several of the columns in an R dataframe
 - Use a logarithmic scale when creating bar plots
 - Leverage shiny to create and run a web application
 - Interact with the shiny web application by running it externally

Lab 4a – Watson Machine Learning

In this lab, you will use IBM's Watson Machine Learning GUI to train, evaluate, and deploy a Watson Machine Learning model based on the Titanic dataset.

Objectives:

- Upon completing the lab, you will:
 - Become familiar with the Watson Machine Learning GUI.
 - Train/Evaluate a machine learning model
 - Deploy a machine learning model.
 - Deploy an application that invokes the machine learning model service.

Lab 4b – DSX SPSS Modeler

In this lab, you will use the Data Science Experience SPSS Modeler capability to explore, prepare, and model passenger data from the Titanic. The SPSS Modeler is a drag and drop capability to build machine learning pipelines.

Objectives:

- Upon completing the lab, you will:
 - Become familiar with the DSX SPSS Modeler capability
 - Profile the Titanic data set
 - Explore the Titanic data set with visualizations
 - Cleanse and Transform the data
 - Train/Evaluate a machine learning mode.

Demo Data - Titanic



Variable Descriptions:

survival	Survival (0 = No; 1 = Yes)
pclass	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
name	Name
sex	Sex
age	Age
sibsp	Number of Siblings/Spouses Aboard
parch	Number of Parents/Children Aboard
ticket	Ticket Number
fare	Passenger Fare
cabin	Cabin
embarked	Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C