

End-to-End Data Science using IBM's Watson Studio



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

Bernie Beekman
Michael Cronk

Agenda

Time	Description
9:00 AM – 10:00 AM	Overview of Watson Studio Lab Orientation
10:00 AM – 10:15 AM	Break
10:15 AM – 12:00 PM	Lab 1-3 – Set up Environment, Watson Knowledge Catalog, Data Refinery
12:00 PM – 12:30 PM	Lunch
12:30 PM – 02:30 PM	Lab Orientation Lab 4,5,6 – SPSS Modeler, SparkML Notebook, AutoAI+DevOps
02:30 PM – 03:45 PM	Lab Orientation Lab 7 – Neural Network
03:45 PM – 04:30 PM	Lab 8 - Rstudio + Shiny Wrap Up

Participant Background

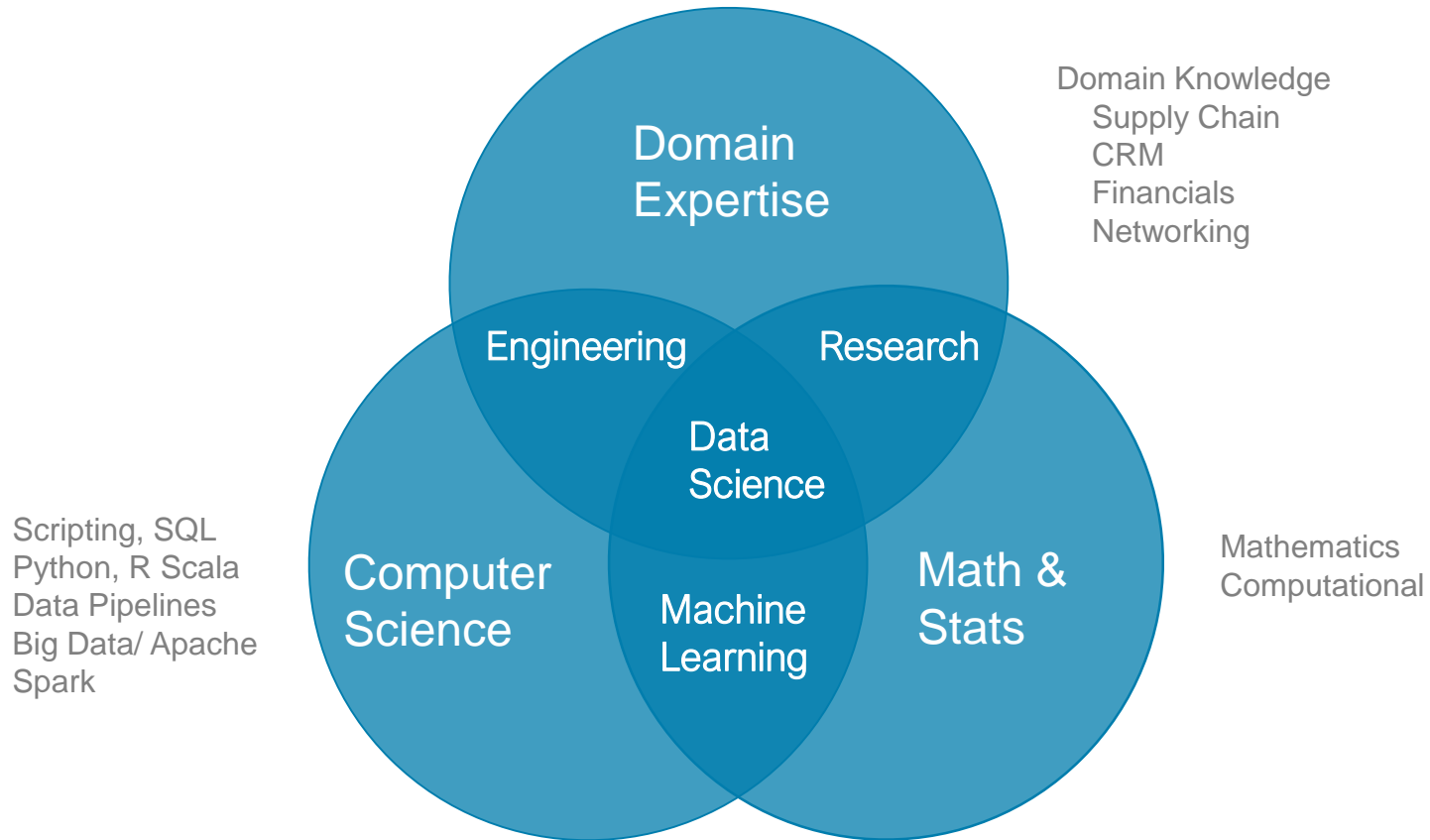
- R/Python/Scala
- Jupyter Notebook
- Machine Learning/Deep Learning
- Keras
- Spark
- Shiny
- IBM Cloud

Outline

- **Data Science Overview**
- **Watson Studio Overview**
- **Lab Overview**



What is Data Science?

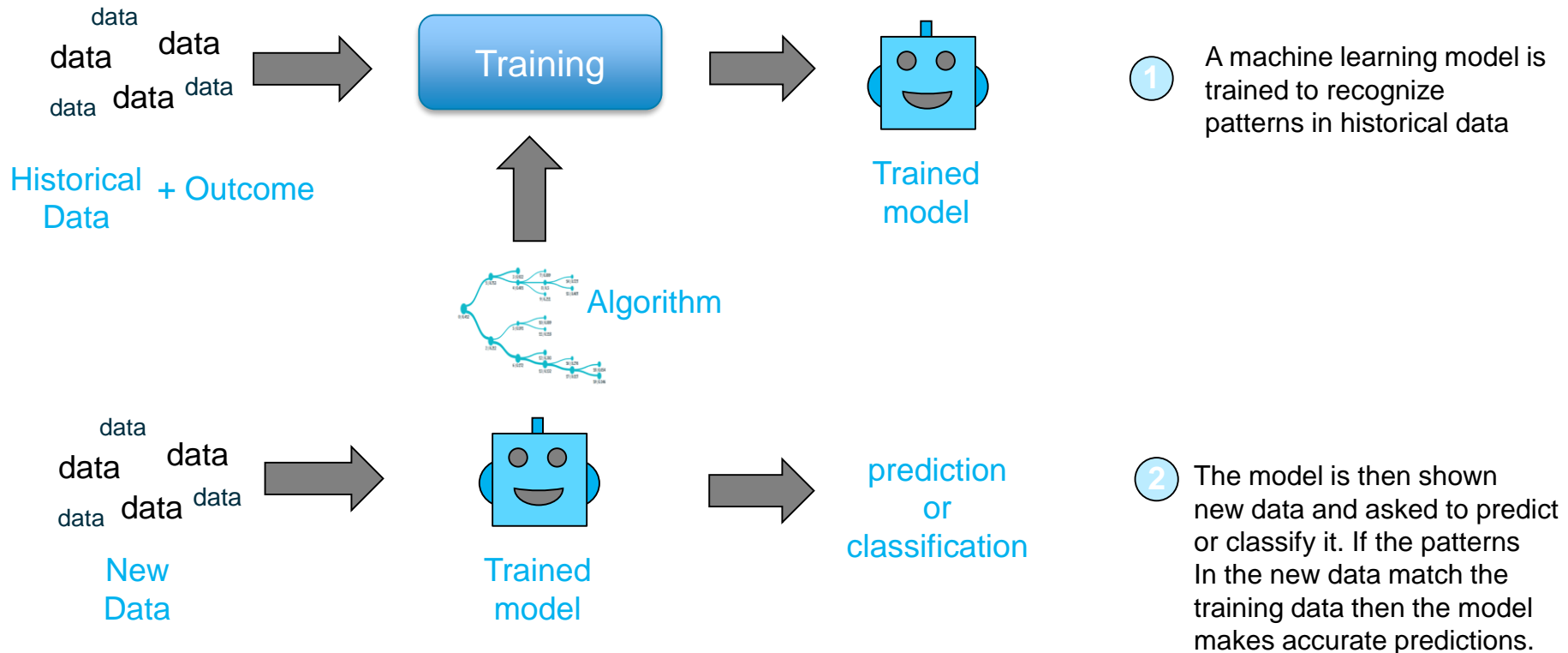


Data Science Projects Require Multiple Skills

Modified from Drew Conway's Venn Diagram

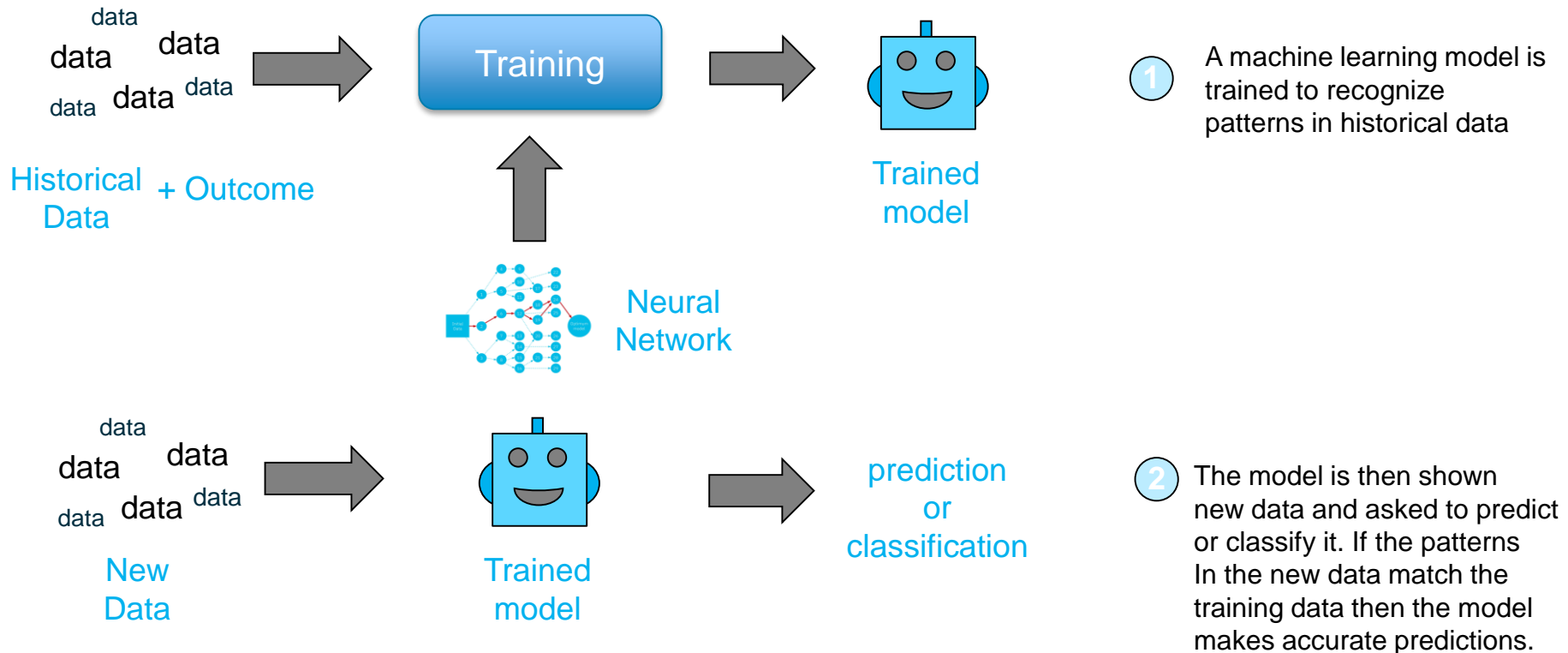
What is Machine Learning?

*“Computers that learn without being **explicitly programmed**”*



What is Deep Learning?

*“Computers that learn without being **explicitly programmed**”*

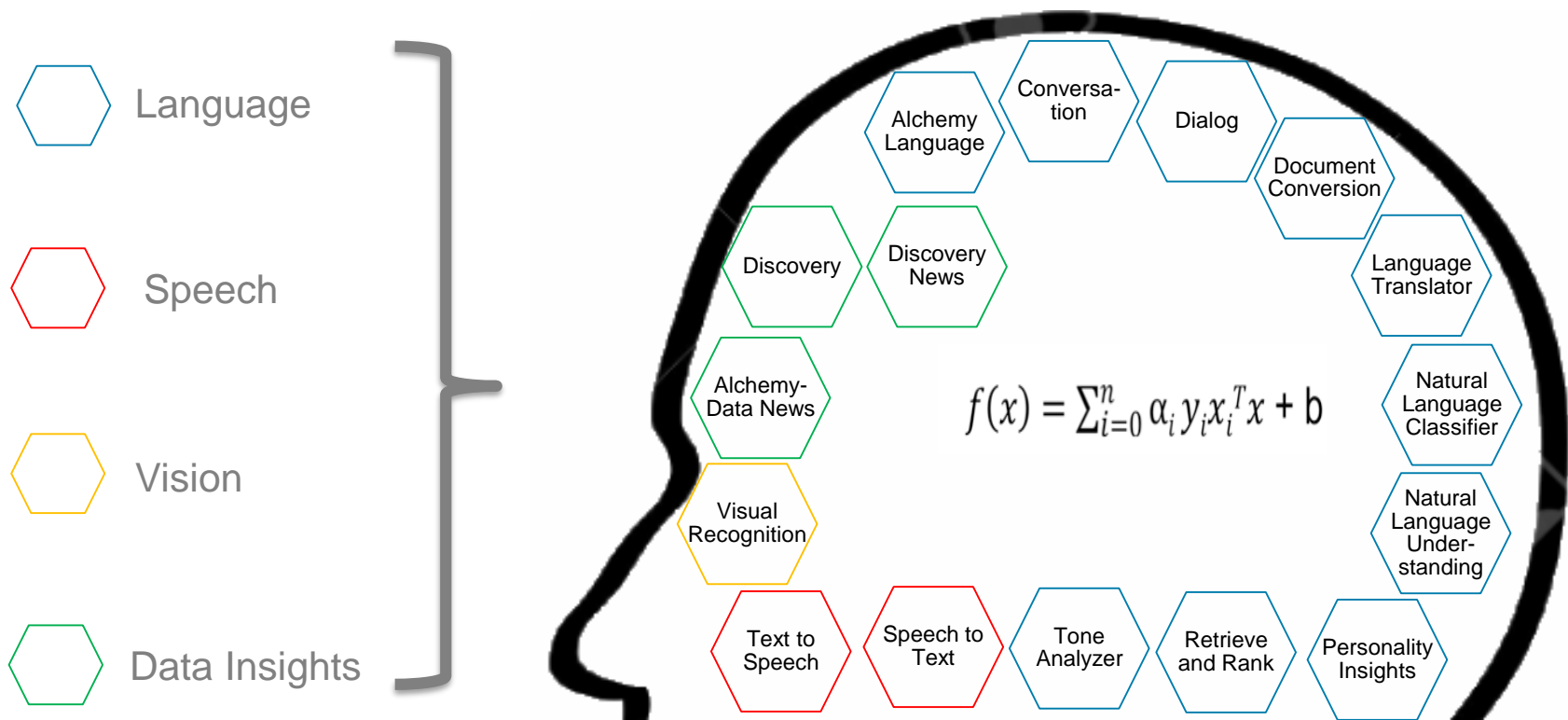


What is Artificial Intelligence?

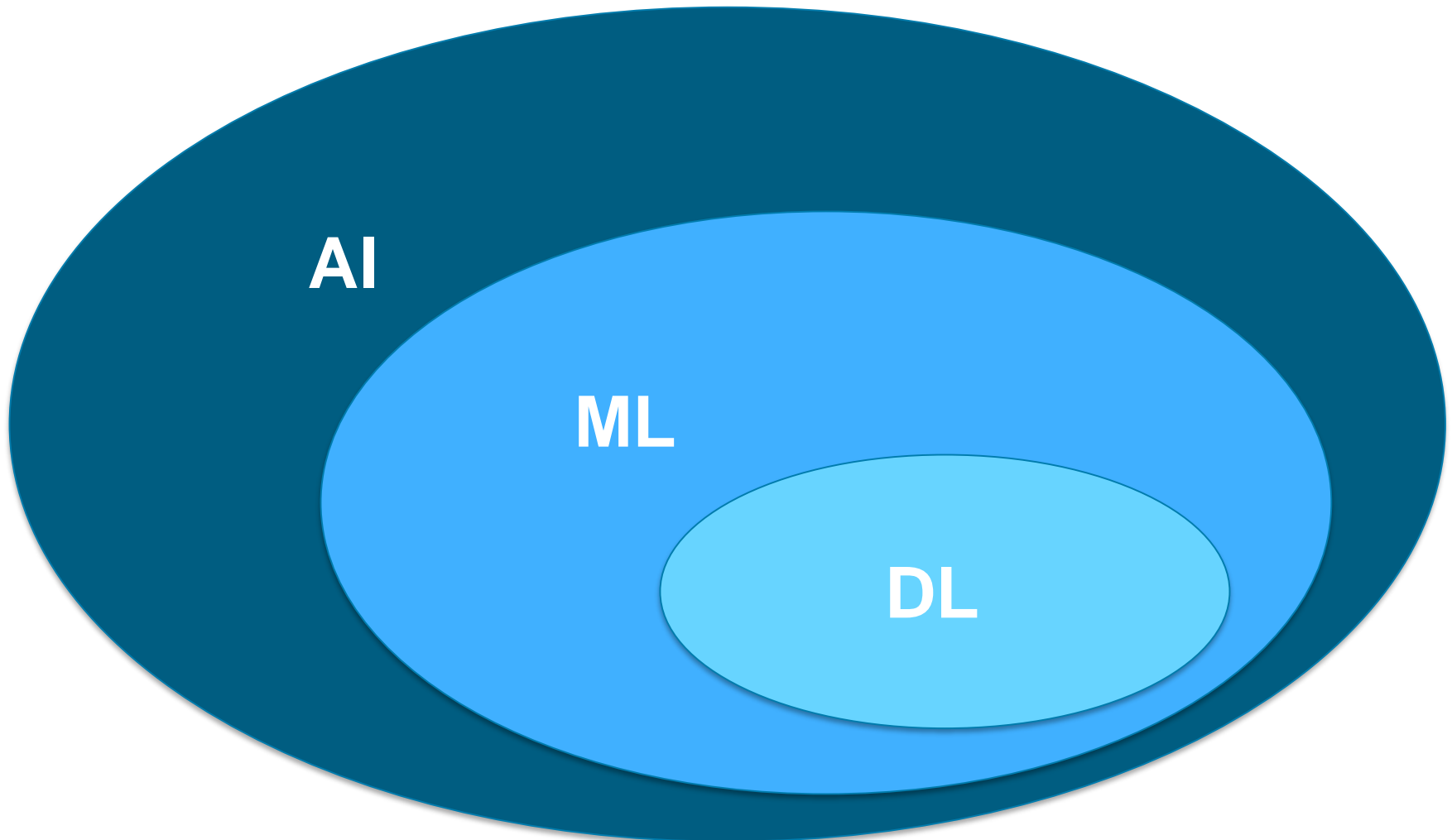
A theory and development of **computer** systems able to perform tasks that normally require **human intelligence**, such as **visual** perception, **speech** recognition, **decision-making**, and translation between **languages**..

Artificial Intelligence = Machine Learning ???

Data + Algorithms = Scored AI Models



Understanding AI, ML & DL Relationship...



IBM takes an Enterprise Approach to Data Science

- Freedom of Choice
 - Choose programming languages, open source libraries, IBM value-add capabilities
 - Code/Click
 - Machine Learning/Deep Learning/Decision Optimization.
- Operationalize Machine Learning
 - Manage complete ML lifecycle – Build, Deploy, Manage, Scale, Monitor, Re-train
- Hybrid ML
 - Build where you want, deploy where you want
- Governance
 - Ensure that right people get access to the right data

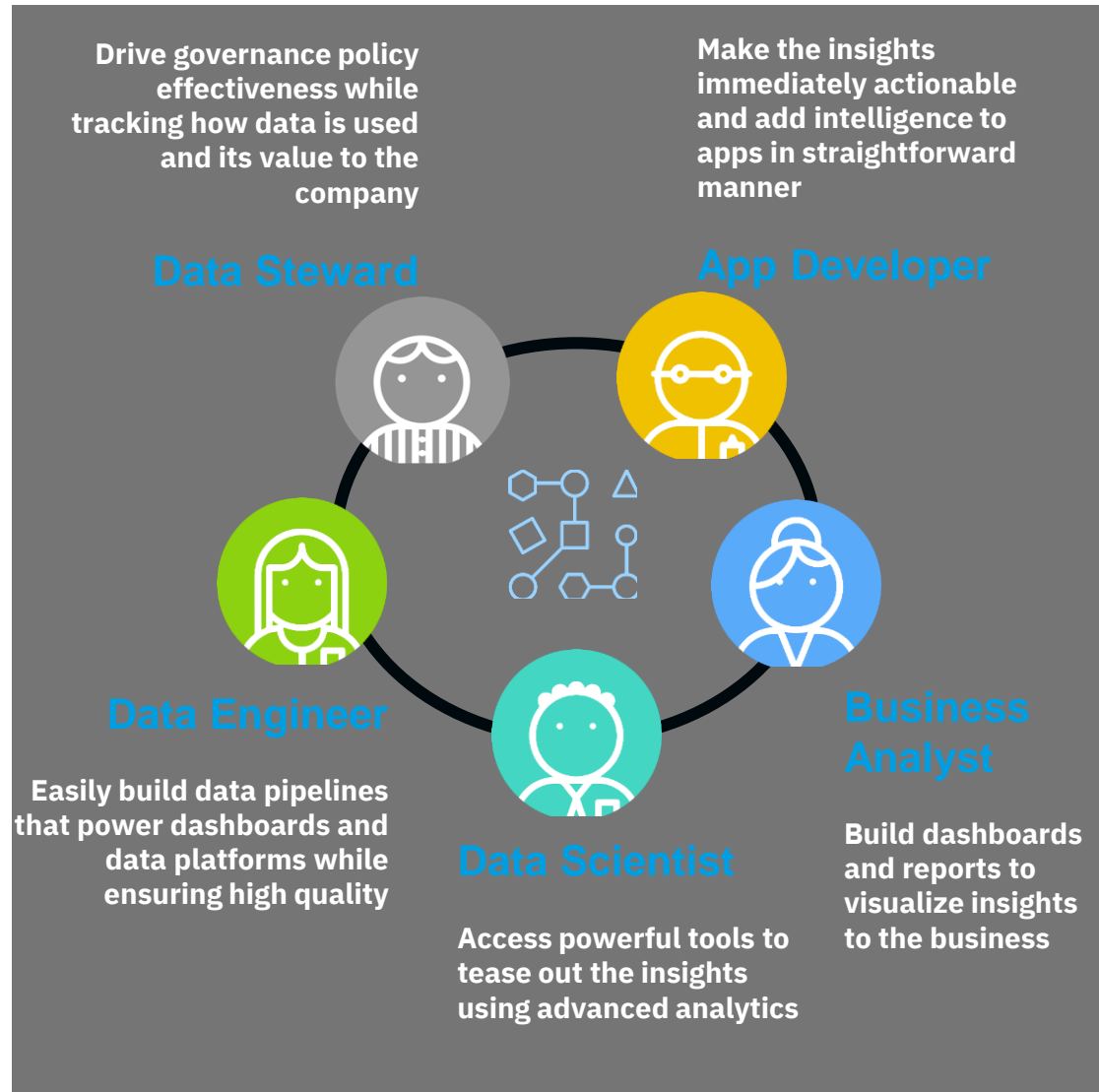
Outline

- Data Science Overview
- Watson Studio Overview
- Lab Overview



IBM Watson Studio Platform

An integrated platform of tools, services, data, and metadata that help companies or agencies accelerate their shift to be data-driven organizations.



Watson Studio Deployment Options

- Watson Studio on IBM Cloud
 - Managed offering provided by IBM

- Watson Studio Local
 - On-premise – Private Cloud
 - IBM Cloud, AWS, Azure

- Watson Studio Desktop

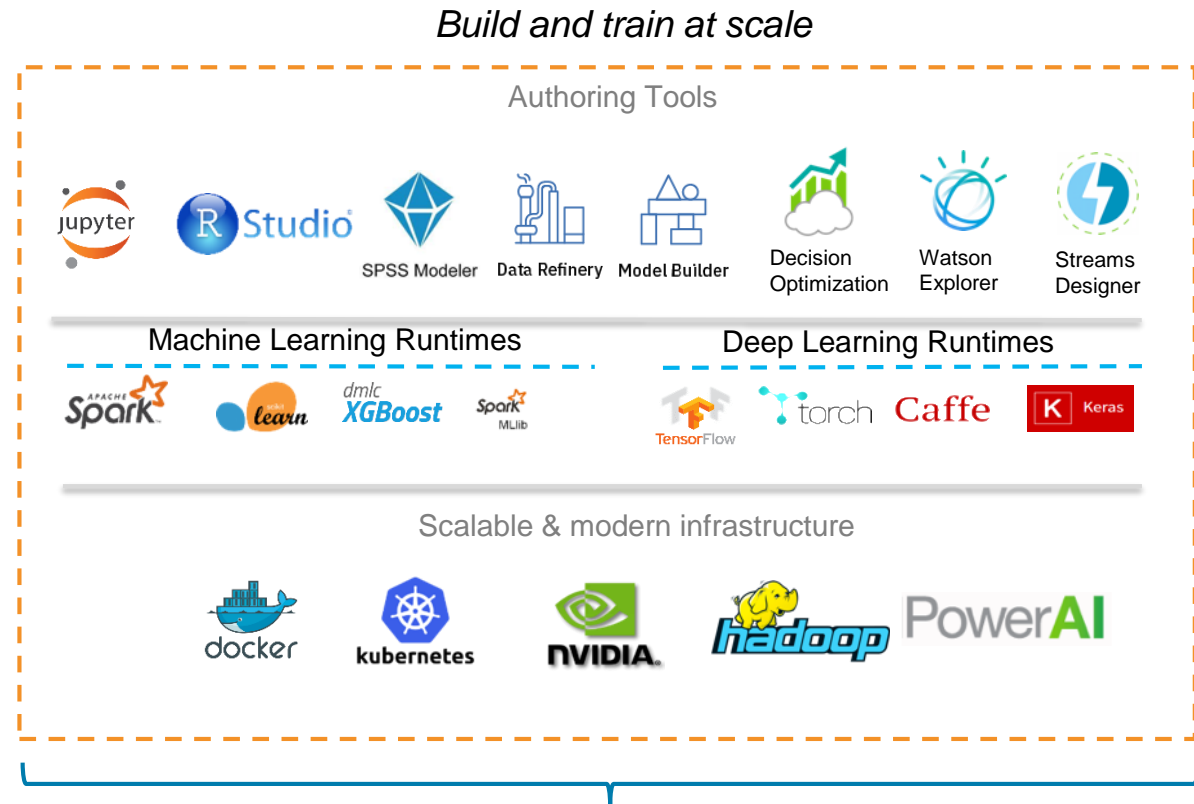
- IBM Cloud Private for Data
 - Watson Studio Local

Watson Studio Tools

- Using best of breed - Open source & IBM tools
- Code (R, Python or Scala) and no-code/visual modeling tools

- Container-based resource management
- Elastic cpu/gpu power
- Run on x86, Power, zLinux
- Integrate with Cloudera and HDP

- Train and deploy where your data lives



IBM
IBM Cloud
Fully Managed


On-prem

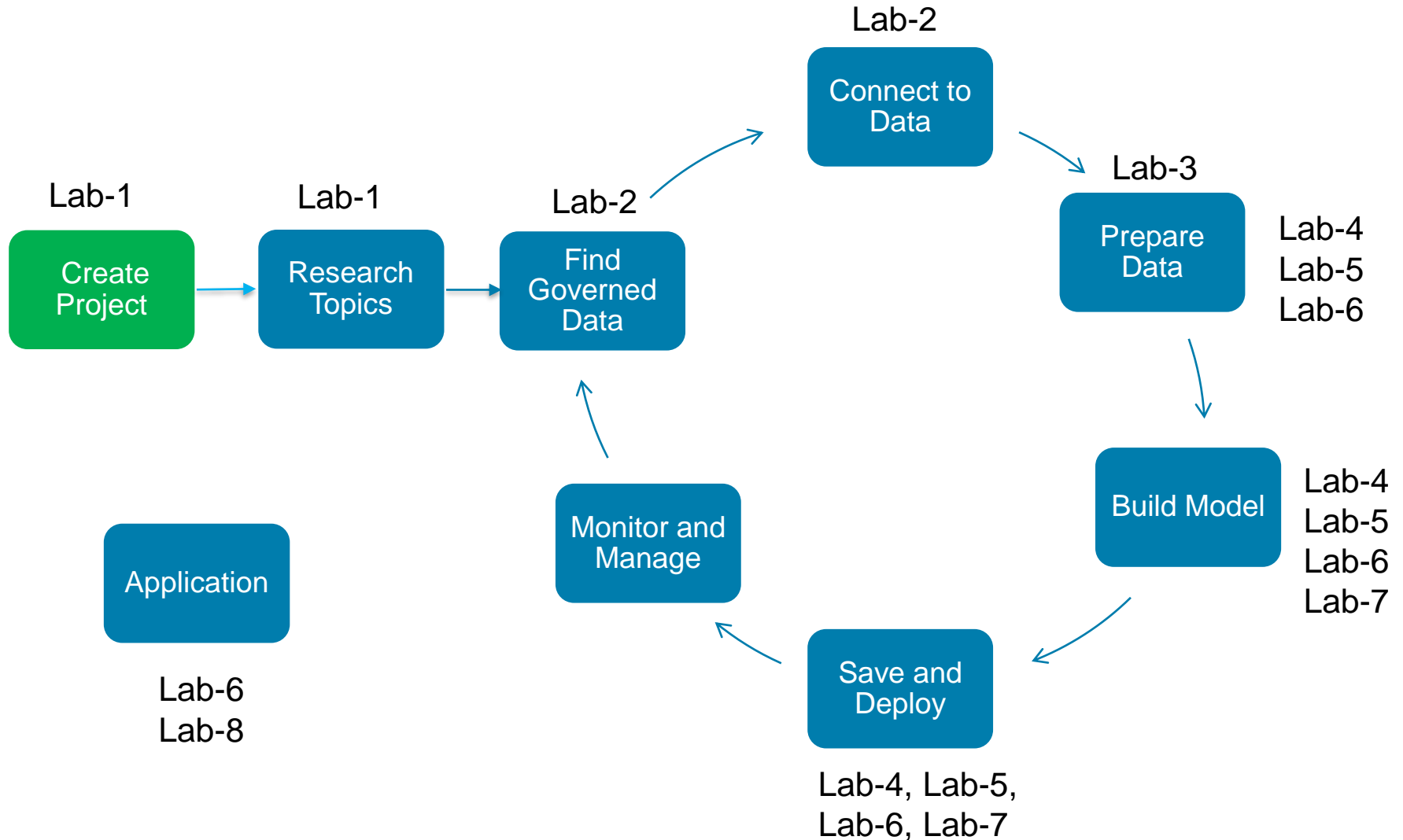
IBM
IBM Cloud

 **amazon**
web services

 **Azure**

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Studio Project Features

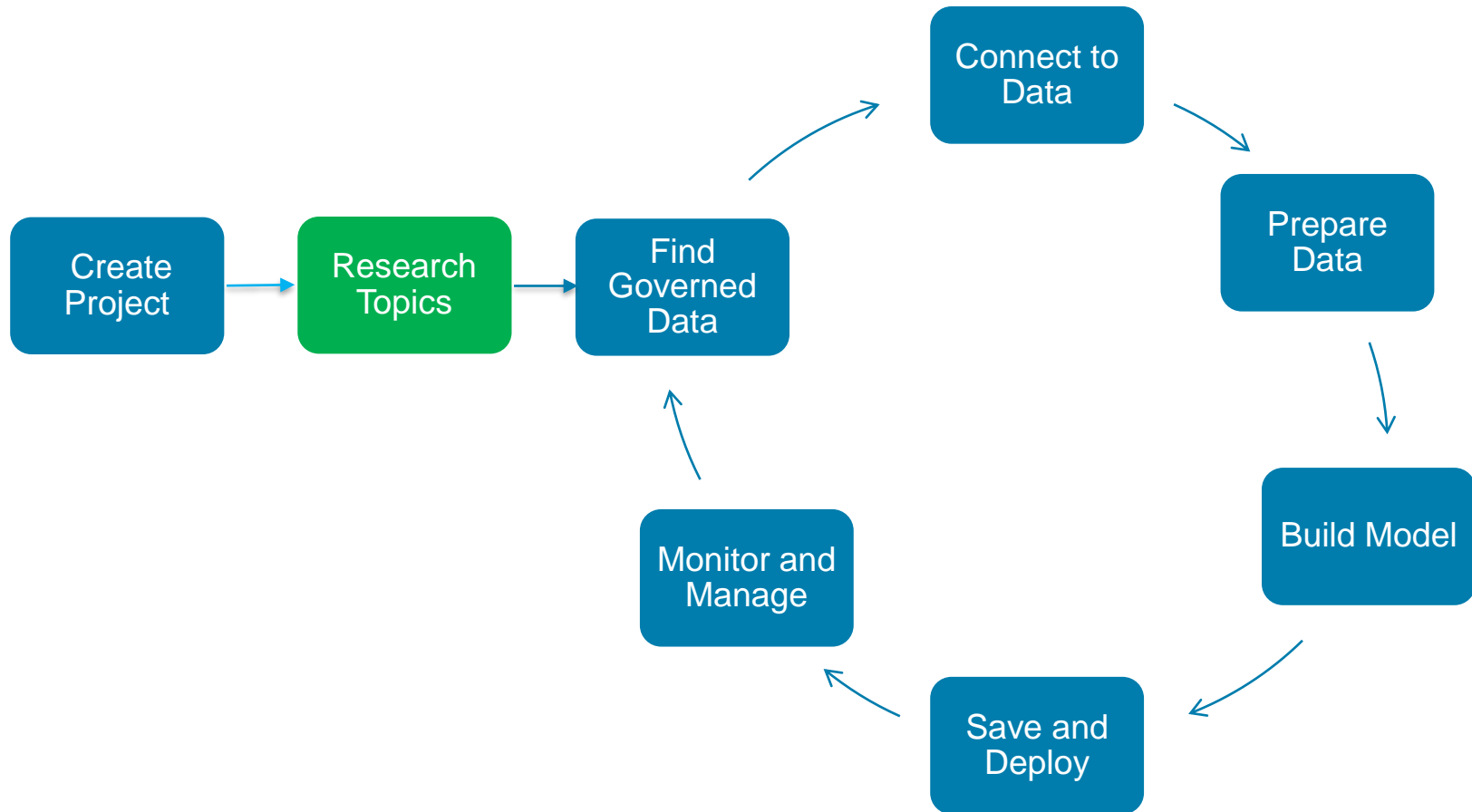
Making Data Science a Team Sport

Create
Project

- Organizes resources to achieve a particular data analysis goal
- Support role-based collaboration (Admin, Editor, Viewer)
- Assets from all IDEs can be included in one Watson Studio project: notebooks, data sources, flows, models, etc.
- Export/Import Projects

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



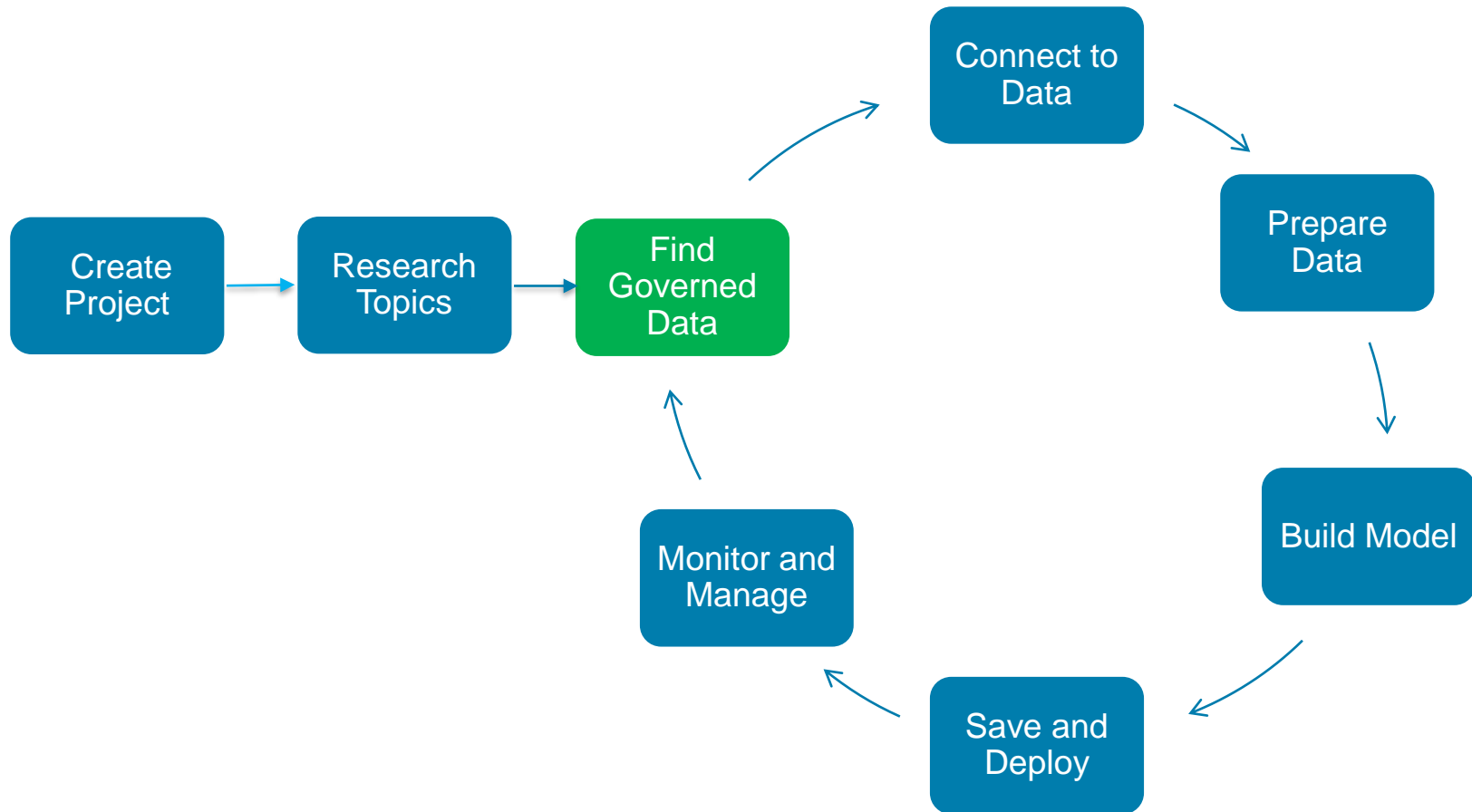
Watson Studio Community Card Features

Built-in learning to get started

- Community Card Feature includes curated articles, tutorials, notebooks, data sets, and papers
- Bookmark in Projects
- Copy notebooks or Data Sets into projects
- Continuously updated in IBM's managed service

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Knowledge Catalog Features

Unlock tribal knowledge and unleash knowledge workers

Find
Governed
Data

- **Find** data (structured, unstructured) and AI assets (e.g., ML/DL models, notebooks, Watson Data Kits) in the **Knowledge Catalog** with intelligent search and giving the right access to the right users.
- Discover assets, profiling, classification
- Policy, rule authoring
- Policy, rule enforcement
- Asset Usage Statistics

Watson Knowledge Catalog Features

Connect to
Data

10 Data Asset

female_human_trafficking

Description

There is no description available for this asset.

Added: Jan 31, 2019 10:02 AM
Format: application/octet-stream
Size: 347 KB

Tags

trafficking | female human trafficking

Reviews

☆☆☆☆☆ 0 reviews

Connection

Source: [Watson Studio Labs_DataCatalog](#)
Source type: Cloud Object Storage

Classification

Personally Identifiable Information

Personally identifiable information (PII) is defined as any data that could potentially identify a specific individual. Any information that can be used to distinguish one person from another can be considered PII.

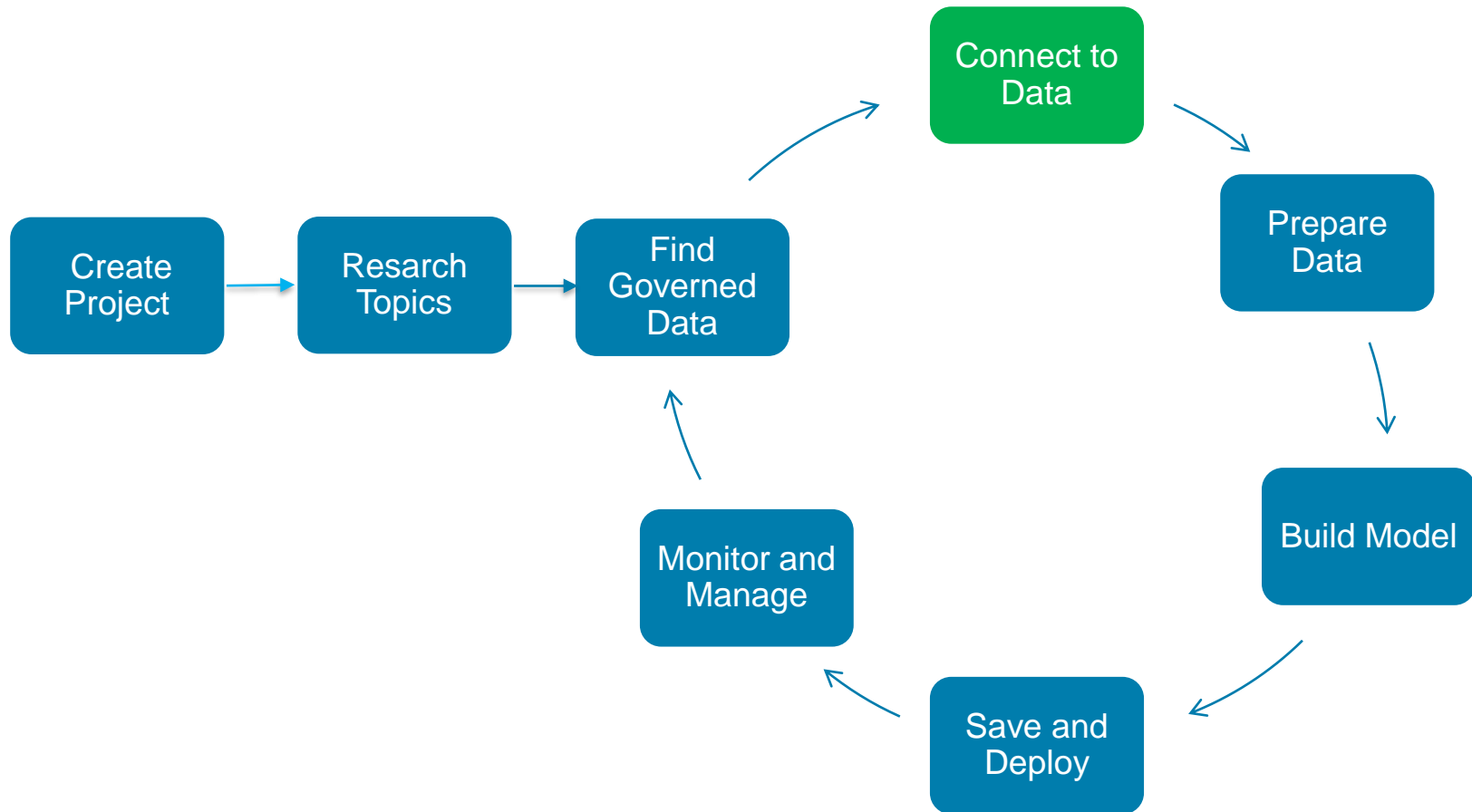
Schema: 26 Columns | 1085 Rows | 2 Columns anonymized ⓘ

Preview: 1000 rows | Last refresh: 22 seconds ago | [Refresh](#)

DATE	BIRTH_COUNT...	BIRTH_COUNTRY_CODE	OCCUPATION	ADDRESS	SSN	PASSPORT_NUMBER
	Type: String	Type: String	Type: String	Type: String	Type: String	Type: String
th	Country Name	Country Code	Text	Text	US Social ...	Passport Number
15	Ghana	GH	Engineer, land	824 Kristin Grv, A	afe55d1d355c3:	1c9da91e1e20863dd850
19	Ghana	GH	Editor, commissi	1148 Wang Fall S	77a0daa42ec7d	12d38855ed107e7cc5dd
16	Ghana	GH	Merchant navy of	9486 Pratt Wall,	669061087d6d1	c43ed0283a3def7031d8:
17	Ghana	GH	Paramedic	0890 Johnson Tr	997b59e501b2e	179abee5ba608418154d
18	Ghana	GH	Surveyor, buildin	2315 Brittany Cr	70329b83b40cb	84524ccc3c5c6590600e:
24	Ghana	GH	Waste managem	88811 Donald Pa	d2f2236f52407:	a730ae13f5ed96f71e904
23	Ghana	GH	Doctor, general p	9150 Donald Rpo	d2c2d41163d8f:	ced1617be1d70e44421c
02	Ghana	GH	Forest/woodland	1355 Lopez Villa	62007942c2b0c	8c8debda401b6b6d954b
12	Ghana	GH	Land/geomatics :	86792 Amy Vlgs,	08f8dd9f9ba89t	a43f1d6c9cacfdfa82a1a1
10	Ghana	GH	Oncologist	108 Erin Via, Nev	f8b871f6e058e2	f289be62078ebbe457c6:
07	Ghana	GH	Veterinary surger	79572 Schmidt E	f2006c1d30df33	624a9605774a0cfd98aa

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Studio Connection Features

Connect to
Data



















- Upload files
- Connectors to Structured and Unstructured, On-prem and Cloud data sources.
- Wizard based connection definition and code generation

Connection Options























Connect to
Data

New connection

IBM services

 BigInsights HDFS	 Cloud Object Storage	 Cloud Object Storage (infrastructure)	 Cloudant
 Cognos Analytics	 Compose for MySQL	 Compose for PostgreSQL	 Db2
 Db2 Big SQL	 Db2 for i	 Db2 for z/OS	 Db2 Hosted
 Db2 on Cloud	 Db2 Warehouse	 Informix	 Object Storage OpenStack Swift (infrastructure)
 PureData for Analytics	 Watson Analytics		

Third-party services

 Amazon Redshift	 Amazon S3	 Apache Hive	 Cloudera Impala
 Dropbox	 FTP	 Google BigQuery	 Google Cloud Storage
 Hortonworks HDFS	 Looker	 Microsoft Azure Data Lake Store	 Microsoft Azure SQL Database
 Microsoft SQL Server	 MySQL	 Oracle	 Pivotal Greenplum
 PostgreSQL	 Salesforce.com	 Sybase	 Sybase IQ
 Tableau	 Teradata		

Notebook Screenshot

[Connect to Data](#)

The screenshot shows the IBM Analytics Notebook interface. The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The right side of the top bar shows 'Trusted', 'Python3.6', and tabs for 'Local', 'Remote', and 'Other'. Below the menu is a toolbar with icons for file operations, running, and markdown. The main area contains two code cells. The first cell displays a schema for a table with various integer and string fields. The second cell, labeled 'In [5]:', contains Python code to connect to a remote data source and load a table into a Spark DataFrame. On the right side, there is a sidebar with a list of tables: PROCEDURES, PATIENTS, and CLAIMS. Each table has an 'Insert to code' button. A dropdown menu is open for the 'CLAIMS' table, showing options to 'Insert Pandas DataFrame' and 'Insert Spark DataFrame in Python'.

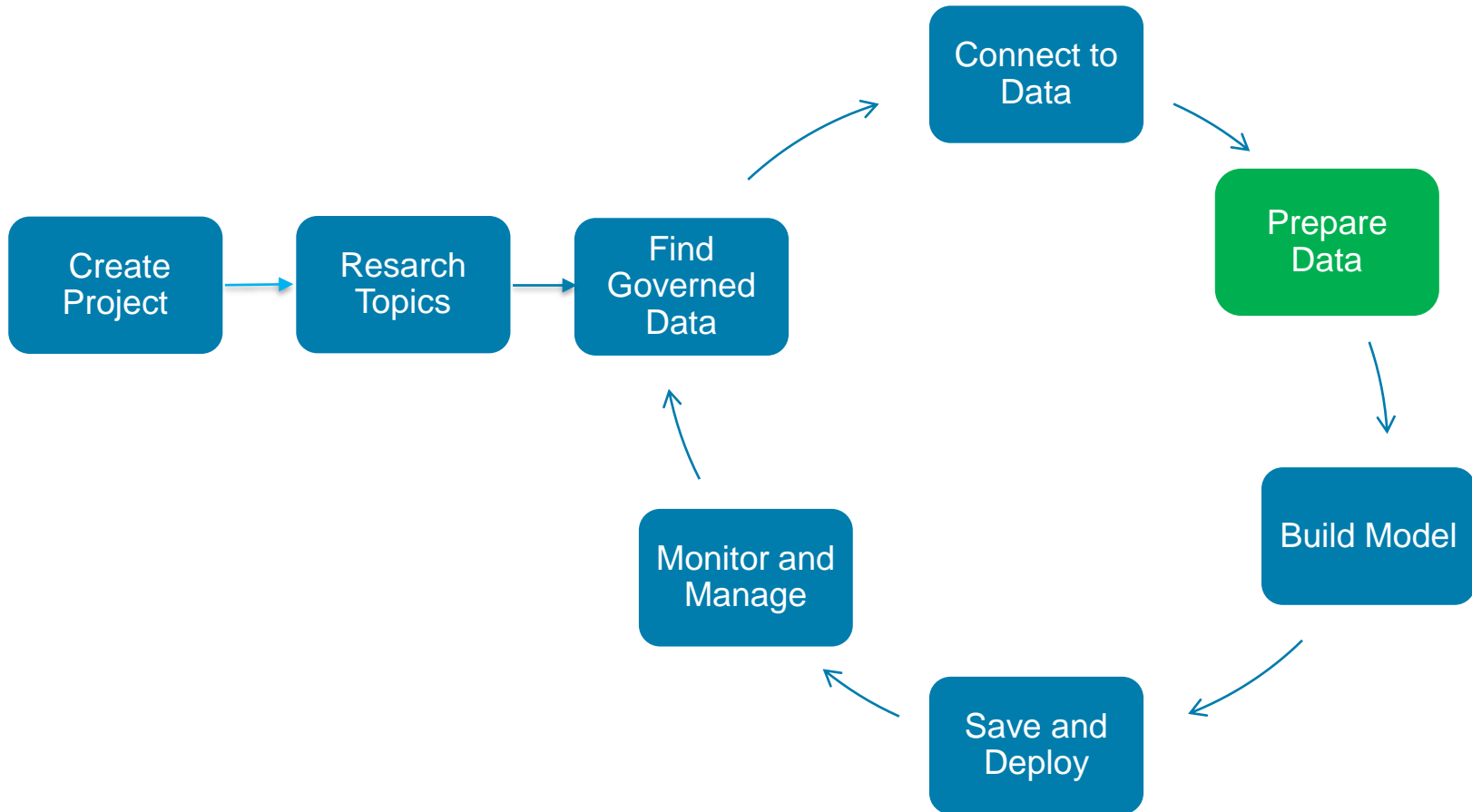
```
-- PROCEDURE_PERCT_RANK: integer (nullable = true)
-- PROCEDURE_RISK_GROUP: string (nullable = true)
-- QUANTITY_INDEX: integer (nullable = true)
-- SERVICE_TYPE: string (nullable = true)
-- SUBMIT_CHG: integer (nullable = true)
-- SUBMITTED_CHG_INDEX: integer (nullable = true)
-- TOTAL_CHARGES_INDEX: integer (nullable = true)
-- TOTAL_CHARGES_PER_PROCEDURE: integer (nullable = true)
-- USER_DEFINED_FLAG_0: string (nullable = true)
-- SUBMITTED_CHARGE_AMOUNT: integer (nullable = true)
-- CLAIM_NUMBER: string (nullable = true)
-- IS_FRAUD: string (nullable = true)
```

Read in CLAIMS Table

```
In [5]:
import dsx_core_utils, requests, os, io
from pyspark.sql import SparkSession
# Add asset from remote connection
df7 = None
dataSet = dsx_core_utils.get_remote_data_set_info('CLAIMS')
dataSource = dsx_core_utils.get_data_source_info(dataSet['datasource'])
sparkSession = SparkSession(sc).builder.getOrCreate()
# Load JDBC data to Spark dataframe
dbTableOrQuery = ''' + (dataSet['schema'] + '.' + dataSet['table'] if len(dataSet['schema'].strip()) != 0 else '') + dataSet['table'] + '''
if (dataSet['query']):
```

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Studio Data Refinery Features

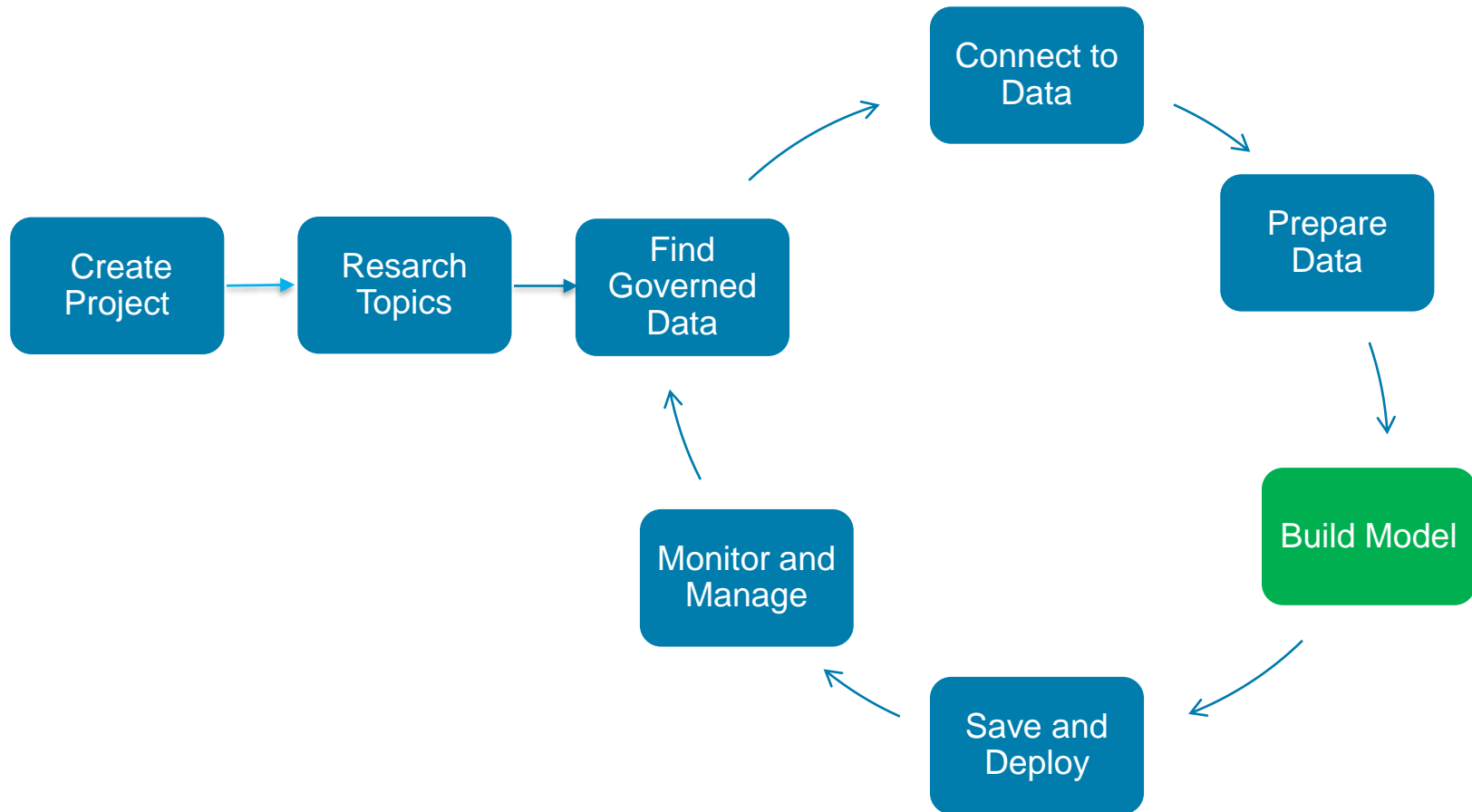
Making Data fit for use

Prepare
Data

- Data Refinery tool to profile, visualize, and shape data.
- Create data preparation pipelines via point and click capability on subset of data
 - ✓ Cleanse the data: fixing or removing data that is incorrect, incomplete, improperly formatted, or duplicated
 - ✓ Shape the data: customize data by filtering, sorting, combining, or removing columns, and performing operations
- Run the pipeline on all the data
 - Manually (on demand)
 - Automated (scheduled)

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Studio Model Building Features

The best of open source and IBM Watson tools to create start-of-the-art data products

Build Model

Open Source Tools

- Jupyter Notebooks**
- RStudio and Shiny**
- Libraries- scikit-learn, XGBoost, Spark**, TensorFlow**, Caffe, Keras, PyTorch

IBM Tools

- AutoAI **
- SPSS Modeler**
- Neural Network Modeler**
- Experiment Builder**
- Natural Language Classifier Model
- Visual Recognition Model

Train at scale on **GPUs** and **distributed** compute

** in hands-on labs


Jupyter Notebook

[Build Model](#)

My Projects / Watson Studio Labs / Machine Learning with SparkML

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.6 with Spark            Format Markdown 

Read Data Asset - female_human_trafficking - See Lab Instructions

```
In [ ]: # Insert SparkSession DataFrame code in this cell after the comments.  
# make CERTAIN to rename the default dataframe name (df_data_1 or df_data_2 or df_data_3, etc) to trafficking_df  
# Put cursor on the next line to Insert to code.
```

Read Data Asset - Occupations - See Lab Instructions

The occupations listed in the female human trafficking file are too numerous to use as input to a machine learning model. We will categorize these occupations into 15 categories by joining with two other files. The Occupation.csv file contains a mapping of the occupations in the female human trafficking table to a category code. The Categories.csv file contains each code followed by the category name. This information needs to be joined to the female human trafficking table.

Follow the same procedure as above to insert a SparkDataFrame for Occupations

```
In [ ]: # Insert SparkSession DataFrame code in this cell after the comments  
# make CERTAIN to rename the default dataframe name (df_data_1 or df_data_2 or df_data_3,etc) to occupations  
#Put cursor on the next line to Insert to code
```

Read Data Asset - Categories - See Lab Instructions

Follow the same procedure as above to insert a SparkDataFrame for Categories

```
In [ ]: # Insert SparkSession DataFrame code in this cell after the comments  
# make CERTAIN to rename the default dataframe name (df_data_1 or df_data_2 or df_data_3,etc) to categories  
#Put cursor on the next line to Insert to code
```

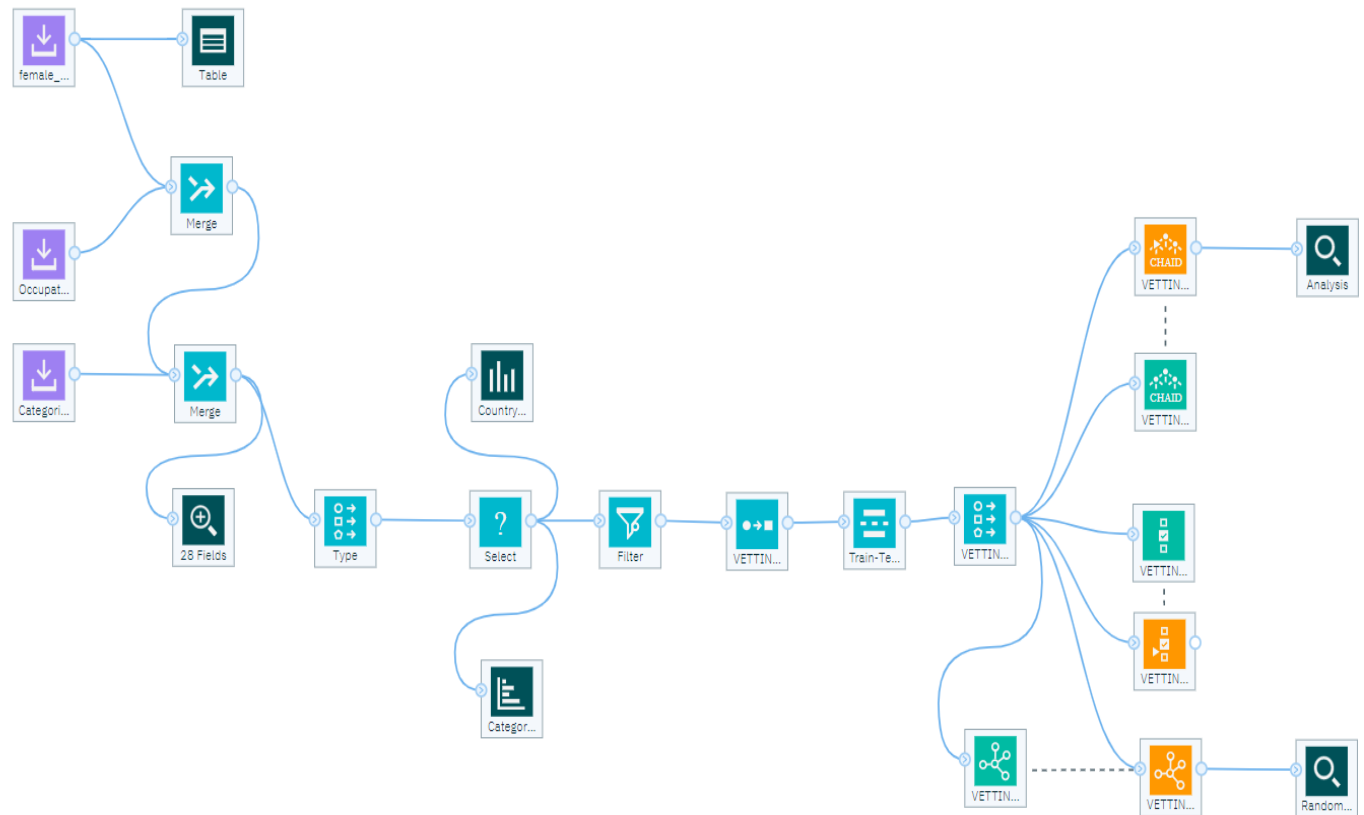
SPSS Modeler

Build Model

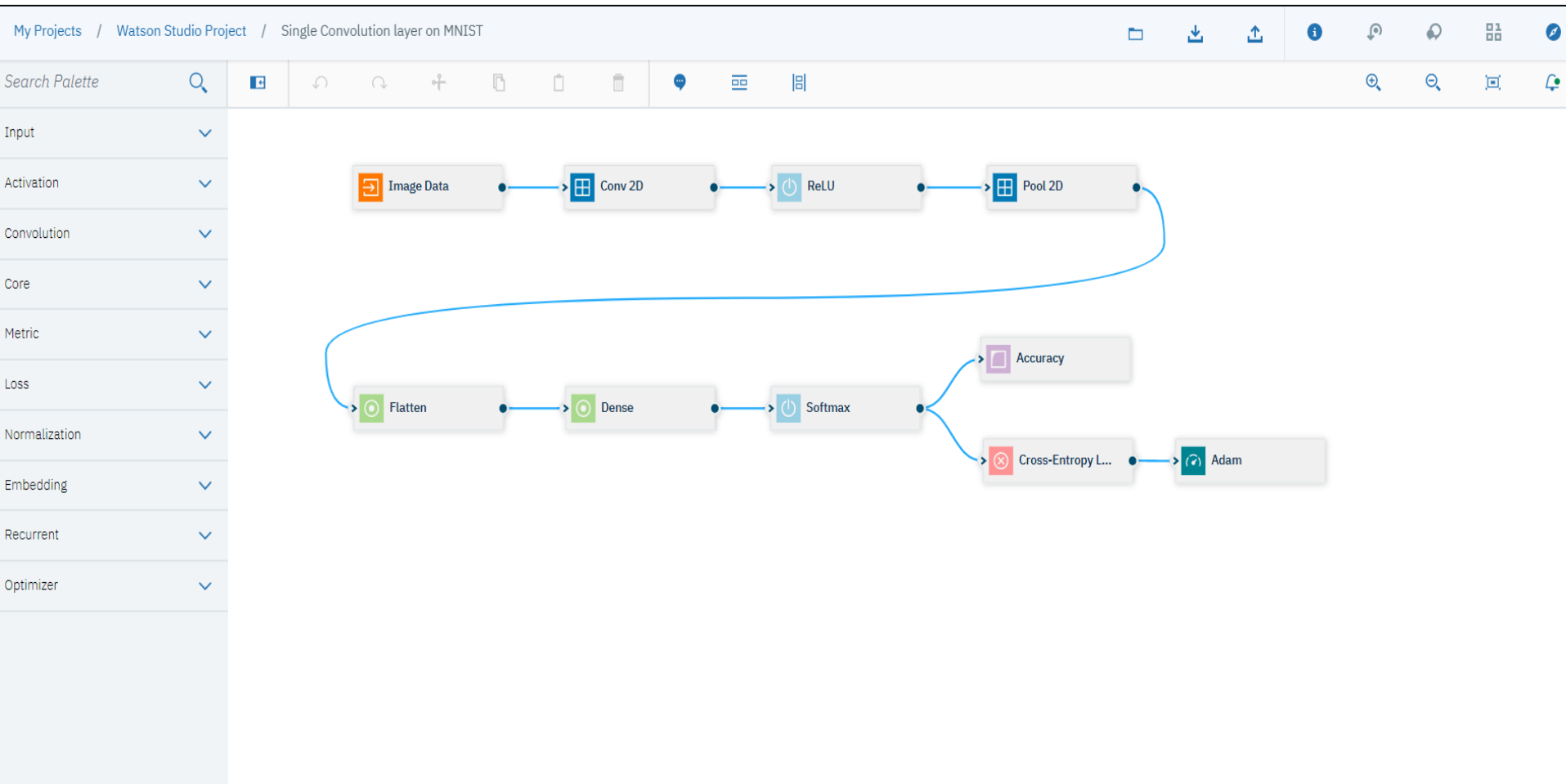
My Projects / Watson Studio Project / FemaleHumanTrafficFlow

Search Palette

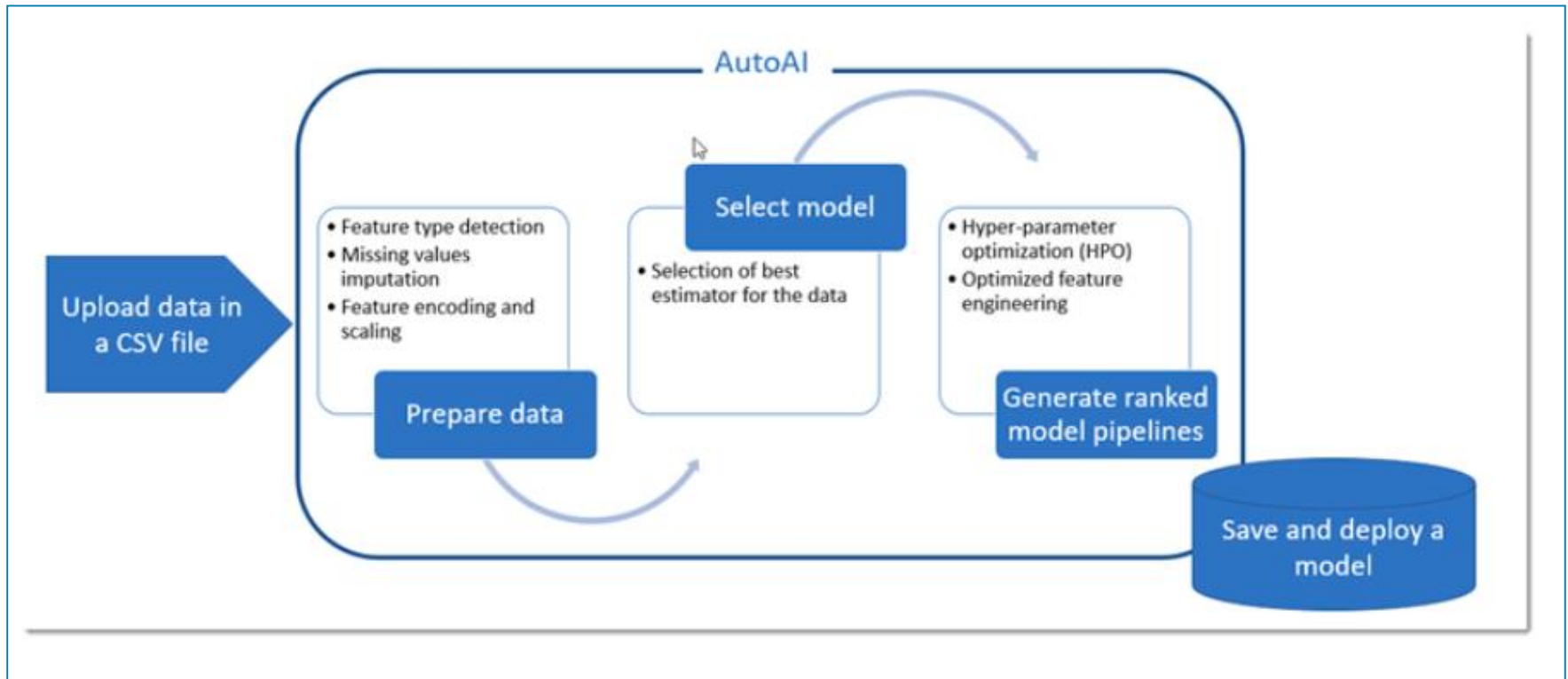
- Import
- Record Operations
- Field Operations
- Graphs
- Modeling
- Outputs
- Export



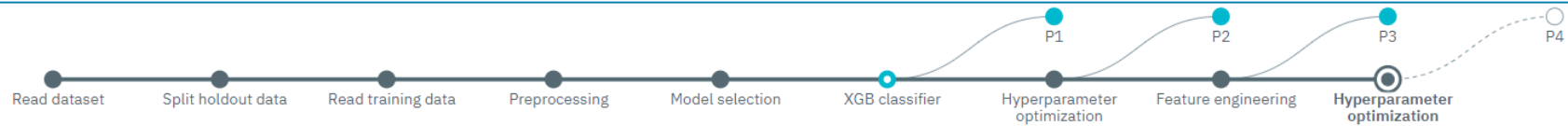
Neural Network Modeler

[Build Model](#)

AutoAI

Build Model


AutoAI

[Build Model](#)

Pipeline leaderboard

[Compare pipelines](#)

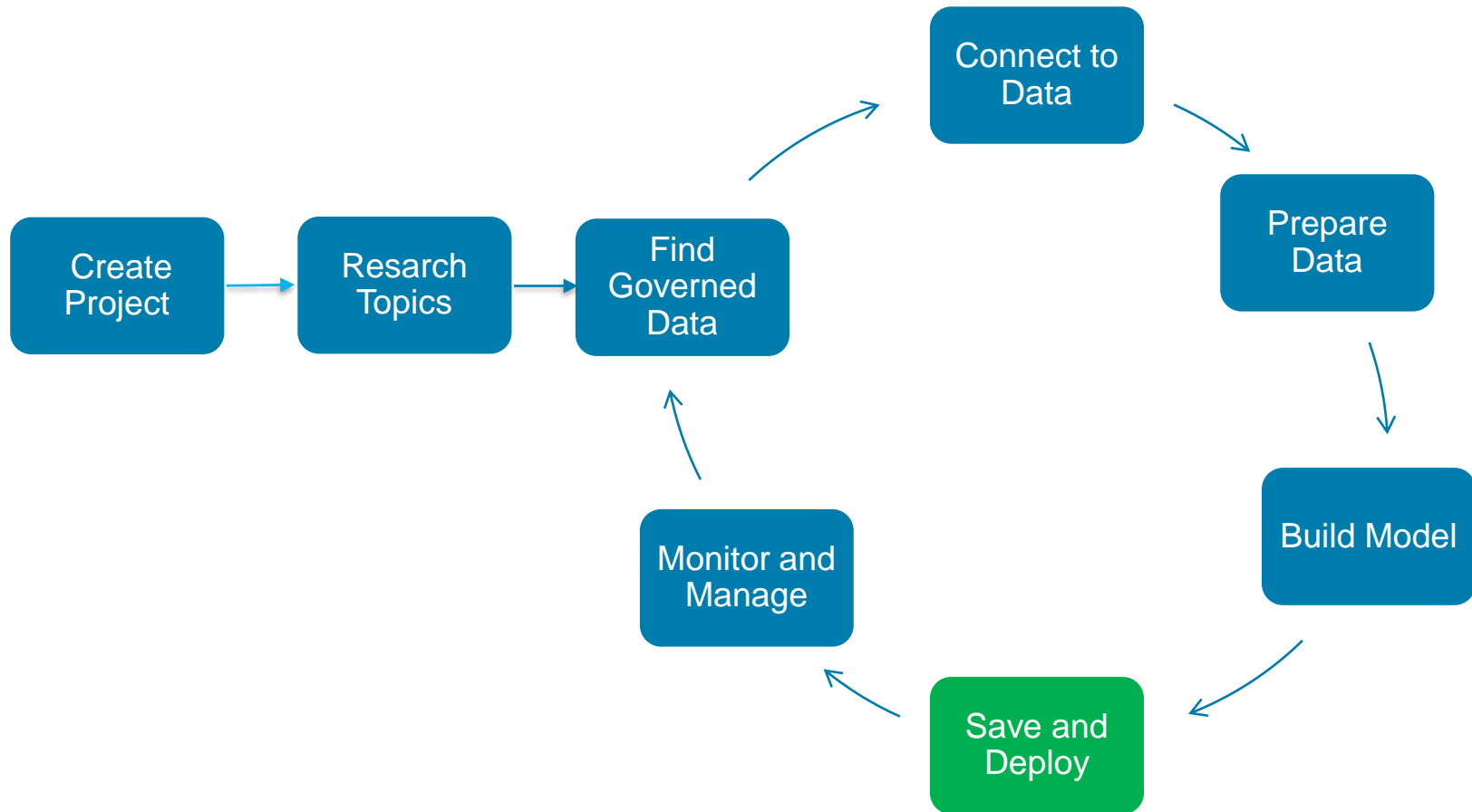
Ranking based on:

Accuracy 

	RANK	ACCURACY	PIPELINE INFORMATION		
>	1	0.897	P3 - XGB classifier estimator Transformers (8): Preprocessing > Standard scaler > Univariate feature selection > Sine > Univariate feature selection > Tangent > ...	View details	Save as model
>	2	0.884	P1 - XGB classifier estimator Transformers (2): Preprocessing > XGB classifier estimator	View details	Save as model
>	3	0.884	P2 - XGB classifier estimator Transformers (2): Preprocessing > XGB classifier estimator	View details	Save as model

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Studio Save and Deploy Models

Save and Deploy Models with Watson Machine Learning

Save and
Deploy



dataplatform.cloud.ibm.com/ml/models/8e525b3c-cd00-48a9-9701-e9208dccee5d/new-deployment?projectid=b9150de9-e8d0-4119-af59-d1a0c8fbf7ee&mlInstanceGuid=be47af33-01c...

IBM Watson Projects Tools Catalog Community Services Manage Support Docs 1449375 - IBM Corpor... BB

Create Deployment

Define deployment details

Name

Model

Description

Deployment description

300

Deployment type

- ☒ Web service
☐ Batch prediction
☐ Realtime streaming prediction

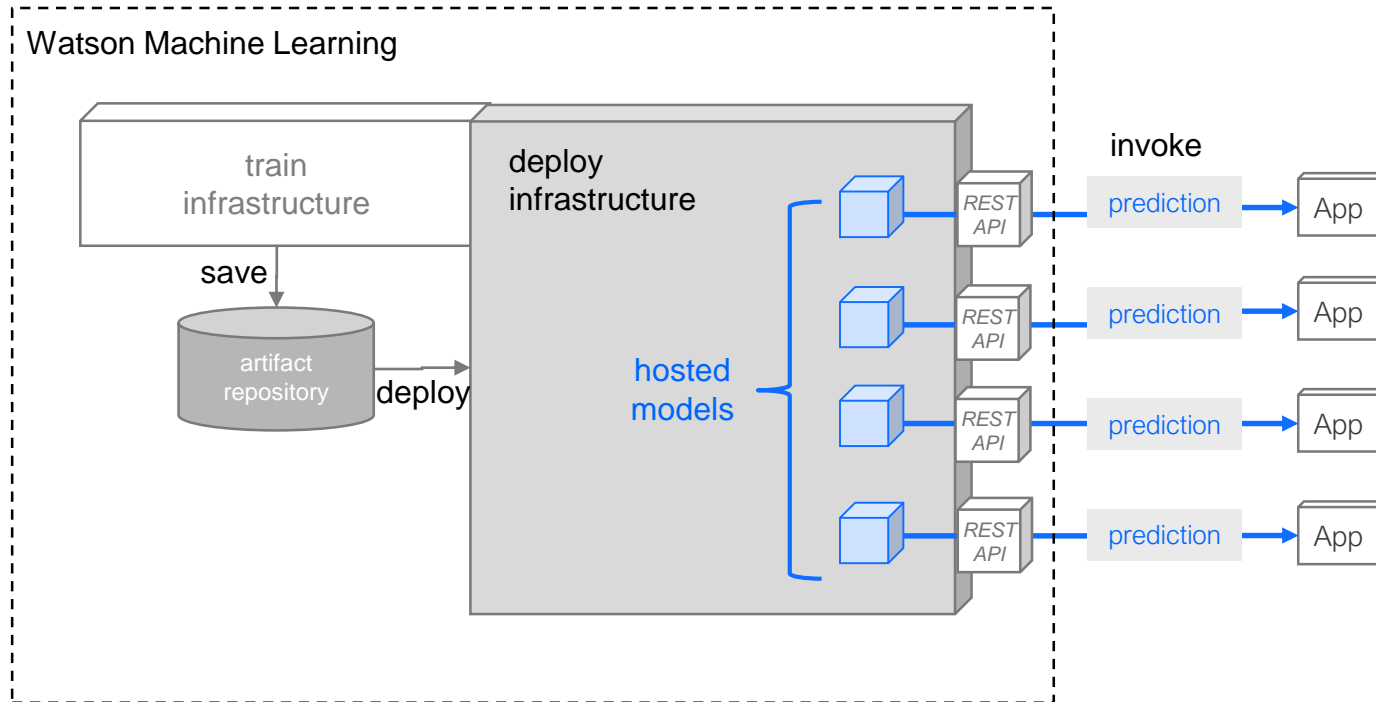
Cancel

Save



Watson Studio Save and Deploy Trained Models

Save and Deploy Models with Watson Machine Learning



Watson Studio Save and Deploy Features

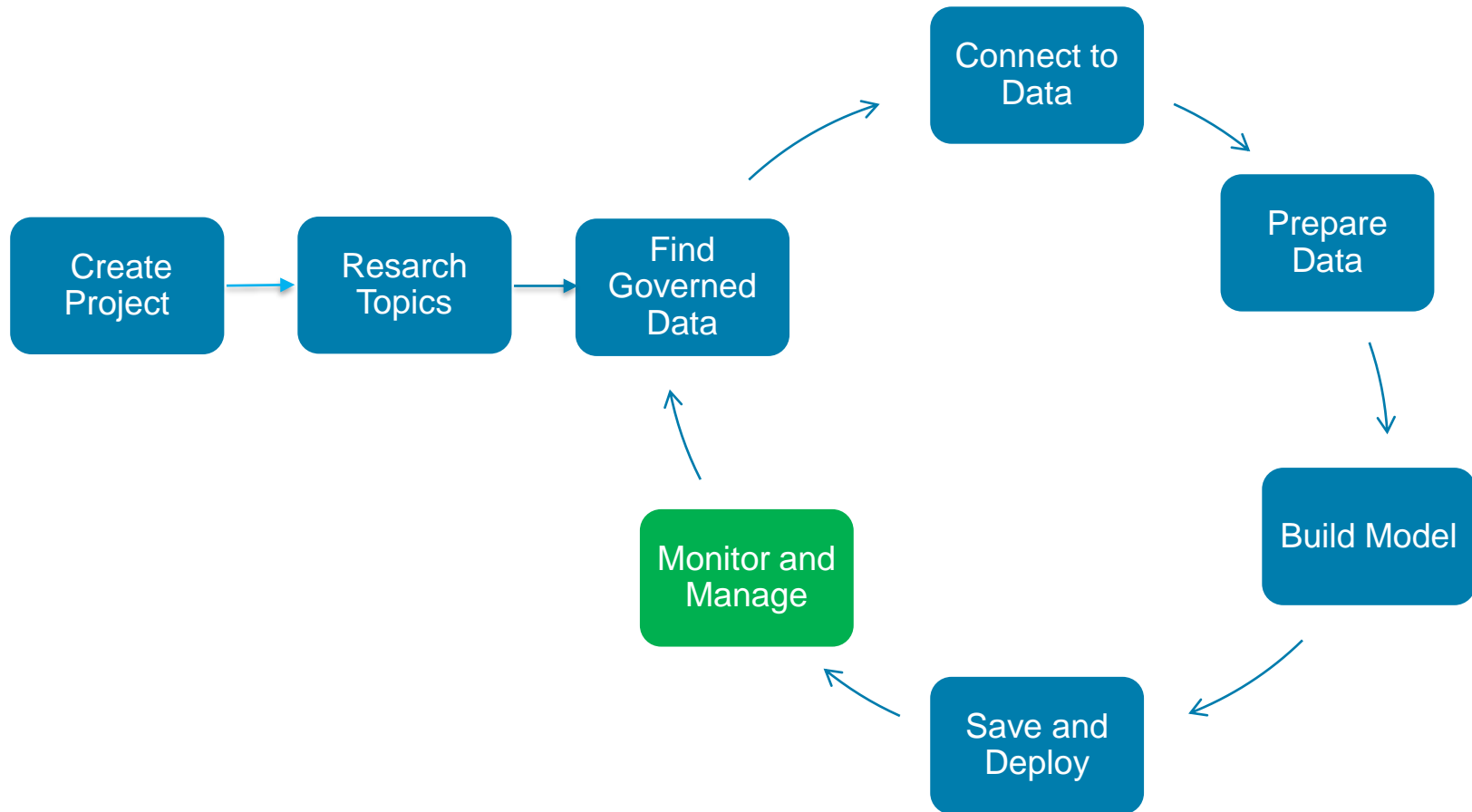
Save and Deploy Models with Watson Machine Learning

Save and
Deploy

- Watson Machine Learning API to save/load models to/from repository
- Watson Machine Learning API to deploy saved models easily and have them scale automatically.
- Watson Machine Learning API to invoke deployed models

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Watson Studio Monitor and Manage

Monitor and
Manage

Configure performance monitoring

Spark Service or Environment

Only Spark environments supporting Scala kernels can be used for continuous learning.


Spark 

Prediction type

binary 

Metric details (type / optional threshold)

areaUnderPR  0.8 

Feedback data connection (IBM Db2 Warehouse on Cloud - [Create new connection](#) )

dashdb: BLUDB / FeedbackBLB [Change feedback data reference](#)

Record count required for re-evaluation

500

Auto retrain

when model performance is below threshold 

Auto deploy

never 

Watson Studio - Monitor and Manage

Monitor and
Manage

Best Heart Drug Model

Overview

Evaluation

Deployments


Lineage

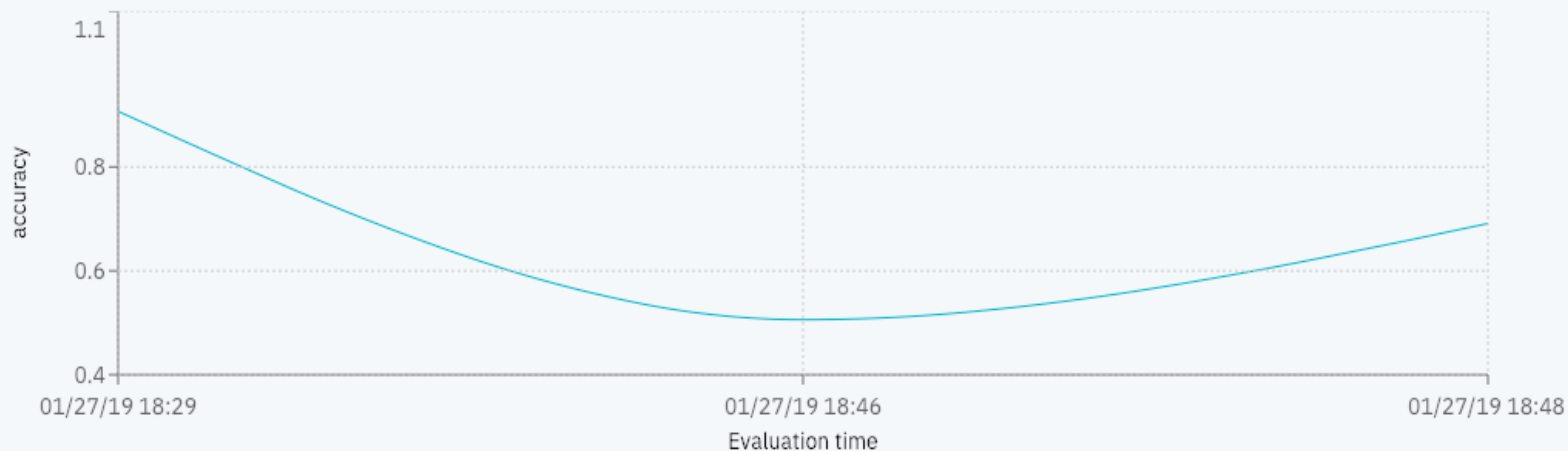
Evaluation Events

 Add feedback data  New evaluation

accuracy



 **Status:** Completed **Stage:** Redeploy **Completion Status:** New Version Not Deployed



Watson Studio Monitor and Manage Features

Monitor and
Manage

- Monitor the performance of the models in production and trigger automatic retraining and redeployment of models.

Our vision for Trusted AI

Pillars of trust, woven into the lifecycle of an AI application



**Is it
accurate?**



**Is it
fair?**



**Is it easy to
understand
?**



**Did anyone
tamper
with it?**

Watson OpenScale

Monitor and
Manage

Trust and Transparency

- Intelligently delivers bias mitigation help
- Provides traceability & auditability of AI predictions made in production applications
- Tracks AI accuracy in applications
- Explains an outcome in business terms

Automation

- Automatically detects and mitigates bias in model output, without affecting currently deployed model or outcomes
- *NeuNetS (beta) – automatically generate Neural Networks

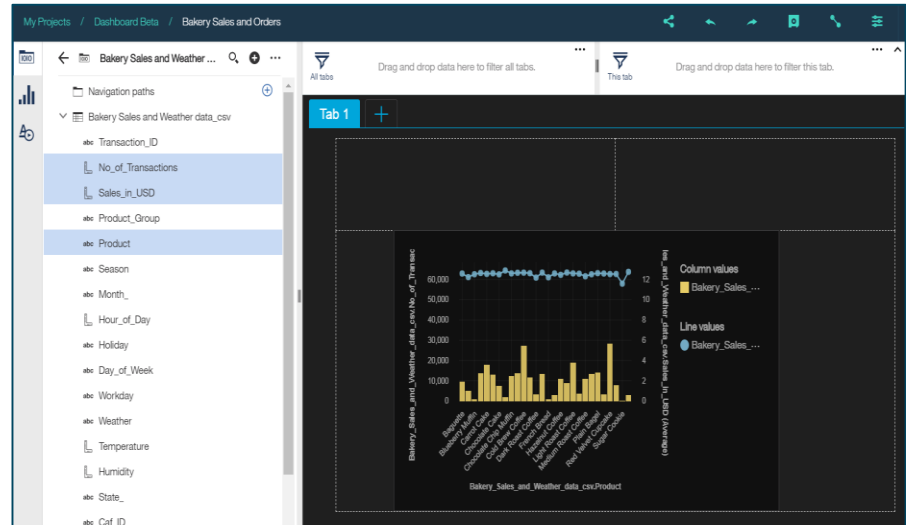
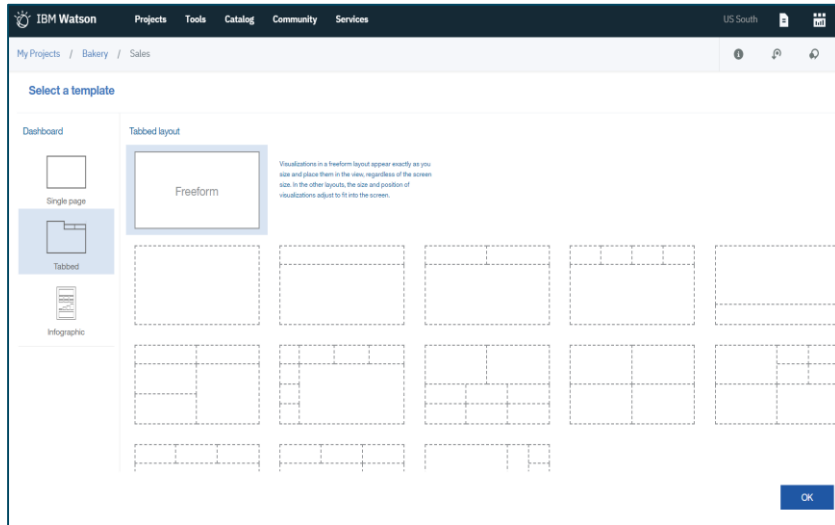
Open By Design

- Monitor models deployed on third party model server engines
- Deploy behind enterprise firewall or on IaaS provider

* <https://arxiv.org/abs/1901.06261>

Watson Studio Dynamic Dashboards

Making insights available to all



My Projects / Bakery Sales

0 assets selected.

Data assets

NAME	TYPE	SERVICE	CREATED BY	LAST MODIFIED	ACTIONS
UNdata_agri_value_add.csv	Data Asset	Project	Alex Jones	7 Mar 2018, 9:37:13 am	
EuropeanCountryStats.csv	Data Asset	Project	Alex Jones	7 Mar 2018, 9:37:12 am	
Bakery Sales and Weather data.csv	Data Asset	Project	Alex Jones	8 Feb 2018, 3:07:05 pm	

Notebooks

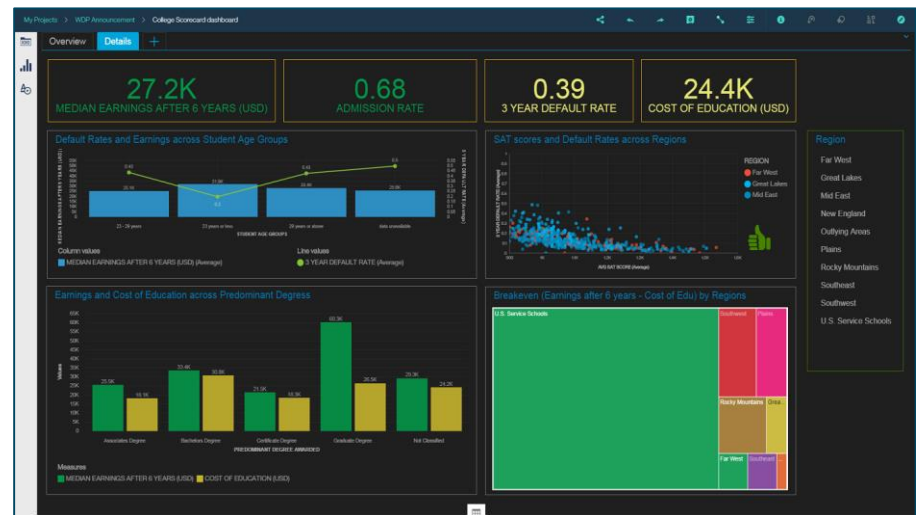
NAME	SHARED	SCHEDULED	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
Sales Predictions					Alex Jones	7 Mar 2018	

Streams flows

Dashboard

NAME	SHARED	LAST EDITOR	LAST MODIFIED	ACTIONS
Bakery Dashboard		Alex Jones	9 Feb 2018, 4:58:46 pm	

Models



Watson Studio Takeaways

Integrated Collaboration Environment

- Data Scientists, Subject Matter experts, Business Analysts & Developers all in one environment to accelerate innovation, collaboration and productivity
- Built-in learning to get started or go the distance with advanced tutorials

Choice of Tools for the full AI lifecycle

- Best in-breed open source and IBM tools that support the end-to-end AI lifecycle
- Choice of code or no-code tools to build and train your own ML/DL models or easily train and customize pre-trained Watson APIs

Support for all levels of expertise

- Use Watson smarts and recommendations for the best algorithms to use given your data, OR
- Use the rich capabilities and controls to fine tune your models

Multiple Deployment Options

- Watson Studio on IBM Cloud – Managed offering
- Watson Studio Local – Private Cloud, Public Cloud-(IBM, Azure, AWS)
- Watson Studio Desktop

Model lifecycle & management

- Deploy models into production then monitor them to evaluate performance.
- Capture new data for continuous learning and retrain models so they continually adapt to changing conditions.

Integrated with Knowledge Catalog

- Intelligent discovery of data and AI assets that enables reuse & improves productivity
- Seamlessly integrated for productive use with Machine Learning and Data science
- Powerful governance tools to control and protect access to data

Outline

- **Data Science Overview**
- **Watson Studio Overview**
- **Lab Overview**



Lab Use Case: 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 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 (SPSS Modeler)	Person age at time of travel
BIRTH_DATE (Notebook)	Person birth date
BIRTH_COUNTRY	Person full birth country
BIRTH_COUNTRY_CODE	Person ISO 2 country
OCCUPATION CATEGORY	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

Github Repository

[bleonardb3 / DS_POT_09-05](#)

Unwatch 1

★ Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

Edit

[Manage topics](#)

56 commits

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾



bleonardb3 Add files via upload

Latest commit ea655e5 16 minutes ago

Lab-1	Add files via upload	yesterday
Lab-2	Add files via upload	18 hours ago
Lab-3	Add files via upload	18 hours ago
Lab-4	Add files via upload	18 hours ago
Lab-5	Add files via upload	1 hour ago
Lab-6	Update README.md	18 hours ago
Lab-7	Add files via upload	18 hours ago
Lab-8	Add files via upload	16 minutes ago
README.md	Update README.md	yesterday

README.md



IBM Proof of Technology - End-to-End Data Science using

Github Repository

Readme

1. **Lab-1** - This lab will set up the Watson Studio environment for subsequent labs and introduce you to the Project and Community features of Watson Studio
2. **Lab-2** - This lab will introduce you to the features of IBM's Watson Knowledge Catalog. Watson Knowledge Catalog is a secure enterprise catalog to discover, catalog and govern your data and modeling assets with greater efficiency.
3. **Lab-3** - This lab will introduce the Data Refinery. Data Refinery is a self-service data preparation tool for data scientists, data engineers, and business analysts. Data Refinery provides profiling, visualization, and a robust set of transforms to prepare data for analytics purposes. We will continue to use the 3 Trafficking data sets in this lab to demonstrate data profiling, data visualization, and data preparation capabilities of the Data Refinery tool. Note the datasets use simulated data.
4. **Lab-4** - In this lab, you will use the Watson SPSS Modeler capability to explore, prepare, and model the trafficking data. The SPSS Modeler is a drag and drop capability to build machine learning pipelines.
5. **Lab-5** - In this lab, you will use SparkML in Watson Studio to run simulated travel data through a machine learning algorithm, automatically tune the algorithm, and load the data into a DB2 Warehouse database.
6. **Lab-6** - This lab consists of two parts. The first part will demonstrate the new and exciting AutoAI capability to build and deploy an optimized model based on the trafficking data sets. The second part will deploy an application using the IBM Cloud DevOps toolchain that will invoke the deployed model to predict the human trafficking risk.
7. **Lab-7** - This lab will use the MNIST computer vision data set to train a convolutional neural network (CNN) model to recognize handwritten digits. The Watson Studio neural network flow editor, Watson Studio experiment builder and the Watson Machine Learning component will be used to build, train, save, deploy, and test the model.
8. **Lab-8** - In this lab, you will learn some of the fundamentals of using RStudio and Shiny in Watson Studio to work and interact with data in a DB2 Warehouse on Cloud database and then to create a fully operational "reactive" web application that you can enhance further.

Github Repository

Lab-1 Readme

Lab-1 - Setup Environment

Introduction:

This lab will set up the Watson Studio environment for subsequent labs and introduce you to the Project and Community features of Watson Studio. Watson Studio is an integrated platform of tools, services, data, and meta-data to help companies and agencies accelerate their shift to be data driven organizations. The platform enables data professionals such as data scientists, data engineers, business analysts, and application developers collaboratively work with data to build, train, deploy machine learning and deep learning models at scale to infuse AI into business to drive innovation. Watson Studio is designed to support the development and deployment of data and analytics assets for the enterprise.

Objectives:

Upon completing the lab, you will:

1. Create a project
2. Create an object storage instance and associate it with the project
3. Create a Watson Machine Learning service instance and associate it with the project
4. Add a collaborator to the project
5. Research topics by searching the Community

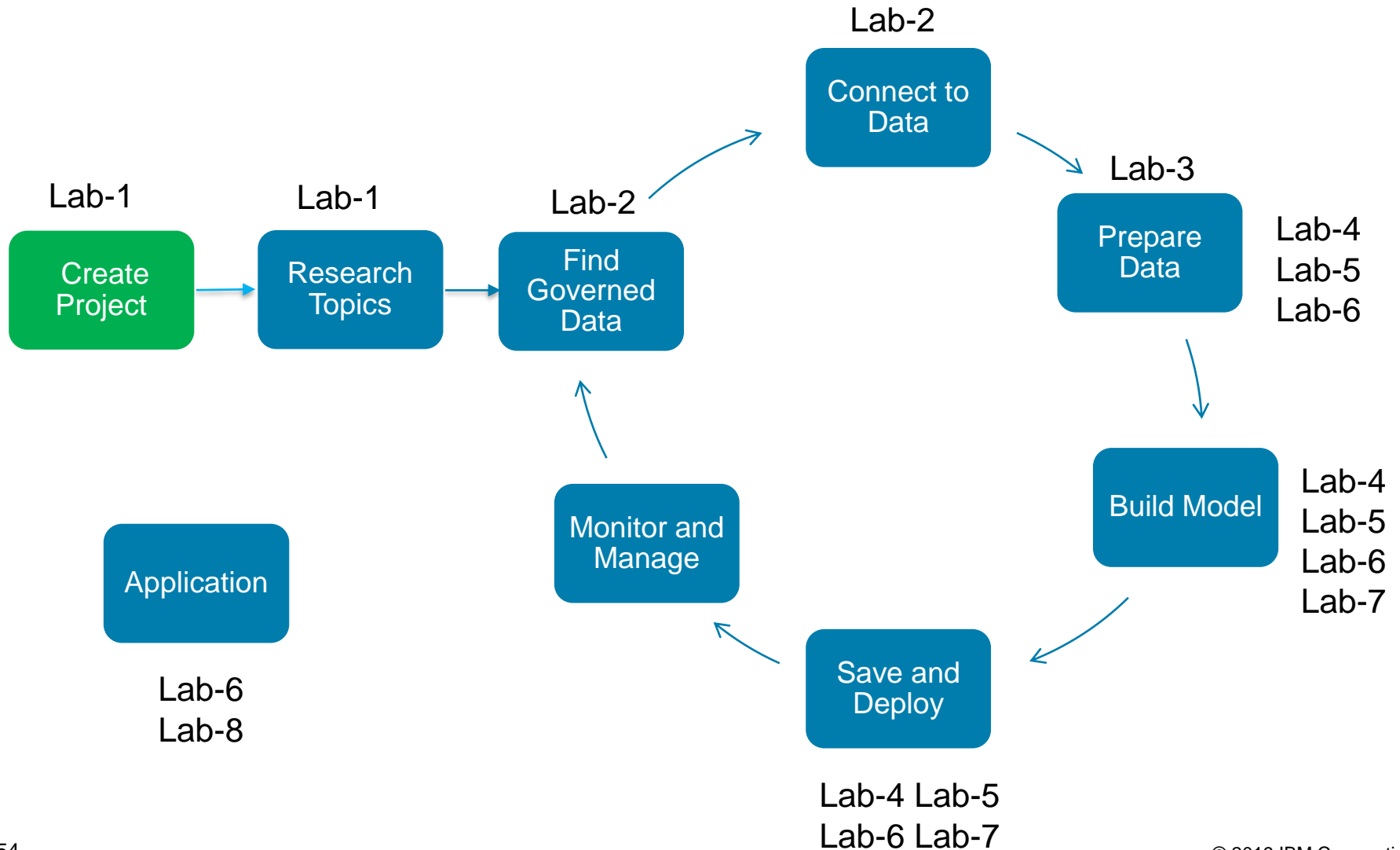
Instructions:

Step 1. Please click on the link below to download the instructions to your machine.

[Instructions.](#)

Watson Studio supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



Lab Tips

- Labs are in www.github.com/bleonardb3/DS_POT_09-05 repository.
- Instructions for each Lab are in the [README](#) file in the respective Lab folder.
- Cloud development enables making frequent improvements in the user interface. We reviewed the lab instructions and made screen updates so they should be pretty faithful to the user interface. Small differences may occur but shouldn't get in the way of successfully completing the labs.
- Do not use Internet Explorer or Edge as the browser. For Mac users do not use Safari.
- All of the Labs should be done in the project that you created when following the signup instructions.
- For Lab-5 make sure when you are creating the notebook that you switch the environment to the Python-Spark environment.
- For Lab-7 make sure that you change the bucket location from ap-geo to us-geo

Browser Tabs



Lab-1: Set up Environment

Introduction:

This lab will set up the Watson Studio environment for subsequent labs and introduce you to the Project and Community features of Watson Studio.

Objectives:

Upon completing this lab, you will know how to:

- Create a project
- Create an object storage instance and associate it with the project
- Create a Watson Machine Learning service instance and associate it with the project
- Add a collaborator to the project
- Research topics by searching the Community

Lab-2: Introduction to Watson Knowledge Catalog

Introduction:

This lab will introduce you to the features of IBM's Watson Knowledge Catalog. Watson Knowledge Catalog is a secure enterprise catalog to discover, catalog and govern your data and modeling assets with greater efficiency.

Objectives:

The goal of the lab is to gain familiarity with the features of the Watson Knowledge Catalog. Upon completing the lab, you will know how to:

- Create a governed catalog
- Add a member to the catalog
- Add Data Assets to the catalog
- Search the catalog
- Edit/Review/Profile a Data Asset
- Demonstrate access control features
- Create and enforce policy
- Push the Data Assets to a project.

Lab-3: Introduction to the Data Refinery

Introduction:

In this lab, you will use the Watson Studio Data Refinery to profile data, visualize data, and prepare data for modeling.

Objectives:

Upon completing the lab, you will know how to:

- Profile the data
- Visualize the data to gain a better understanding
- Prepare the data for modeling
- Run the sequence of data preparation operations on the entire data set.

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

Preprocessing: Matrix for Machine Learning

Known as:

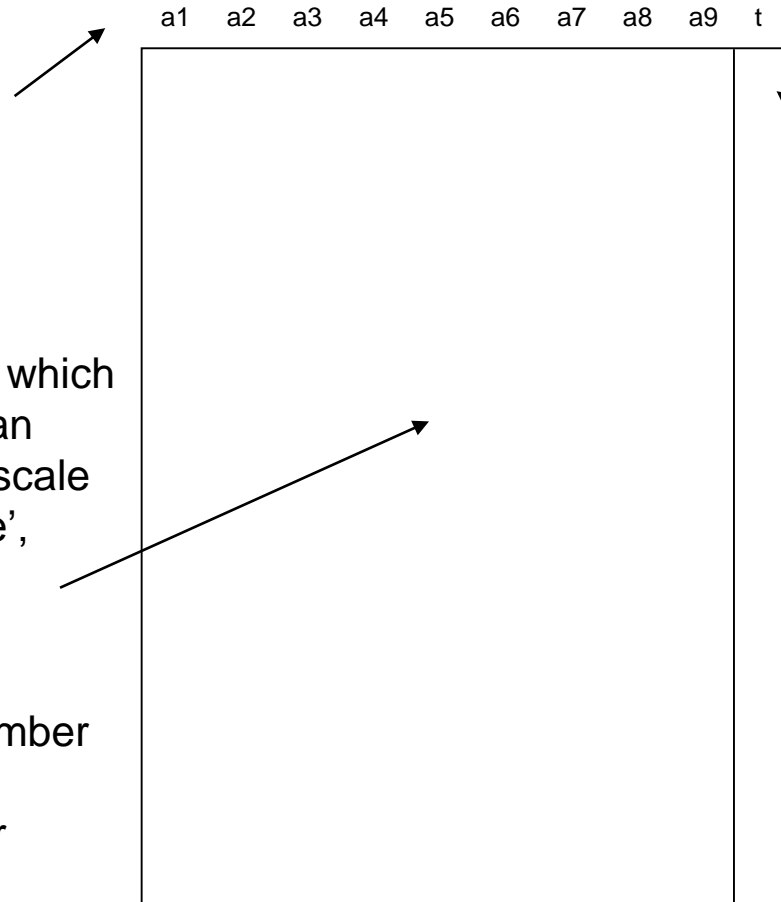
- Attributes
- Features
- Predictor variables
- Explanatory variables

Scale variables:

- Continuous variables, which can be measured on an interval scale or ratio scale
- 'Weight', 'Temperature', 'Salary', etc...

Categorical variables:

- Data with a limited number of distinct values or categories (nominal or ordinal)
- 'Hair color', 'Gender', 'Grape varieties', etc...



Known as:

- Label
 - Target variable
 - Dependent variable
- Scale or Categorical

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
- Validation set
 - Assess model quality during training/tuning process
 - Avoid overfitting the model to the training set
- Test set
 - Estimate accuracy or error rate of model after tuning
 - Used to compare multiple models

Lab-4: SPSS Modeler

Introduction:

In this lab, you will use the Watson Studio SPSS Modeler capability to explore, prepare, and model trafficking data. 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 Watson Studio SPSS Modeler capability
- Profile the data set
- Explore the data set with visualizations
- Transform the data
- Train/Evaluate a machine learning mode.

Lab-5: Machine Learning using SparkML

Introduction:

In this lab, you will use SparkML in Watson Studio to run generated travel data through a machine learning algorithm, automatically tune the algorithm, and load the prediction results into a DB2 on Cloud database.

Objectives:

Upon completing the lab, you will know how to use a Jupyter Notebook to:

- Connect to a cataloged assets to read in data used for machine learning.
- Select the target and features
- Transform data
- Declare a machine learning model.
- Setup up the data transform and modeling pipeline
- Train the model.
- Evaluate the model.
- Automatically tune the model.
- Score data and load into a new DB2 table.
- Save the trained model

Spark and Spark ML

Spark – why should I use it?

- Spark is a highly scalable runtime environment for analytics
- Provides the runtime engine and API
- Supports multiple languages: Python (PySpark), R (SparkR) and Scala

If you want to take advantage of Spark scalability and performance, you have to use Spark APIs

- Example (Python): Spark data frame vs. Pandas, Spark algorithms vs. scikit-learn
- It's possible to “mix and match” Spark and non-Spark code in a single notebook: the runtime environment will switch automatically
 - For example, use Python API for data understanding and SparkML for modeling

Spark Machine Learning API: <https://spark.apache.org/docs/latest/ml-guide.html>

Supported versions of Spark:

<https://www.ibm.com/software/reports/compatibility/clarity/prereqsForProduct.html>

Spark ML Pipeline Terminology

Spark ML standardizes APIs for distributed machine learning

- **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-5: Flow

Read in data from Cataloged Assets

- Join trafficking, job categories, occupations 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-5: 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 the Database

Save the model in the Model Repository

- Model properties can be saved as well (e.g Area under the ROC curve)

Lab-6: AutoAI + DevOps

Introduction:

This lab consists of two parts. The first part will demonstrate the new and exciting AutoAI capability to build and deploy an optimized model based on the trafficking data set. The second part will deploy a web application using the IBM Cloud DevOps toolchain that will invoke the deployed model to predict the trafficking risk.

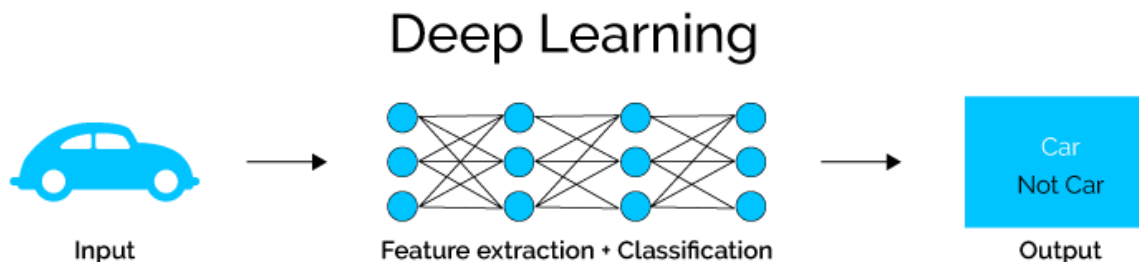
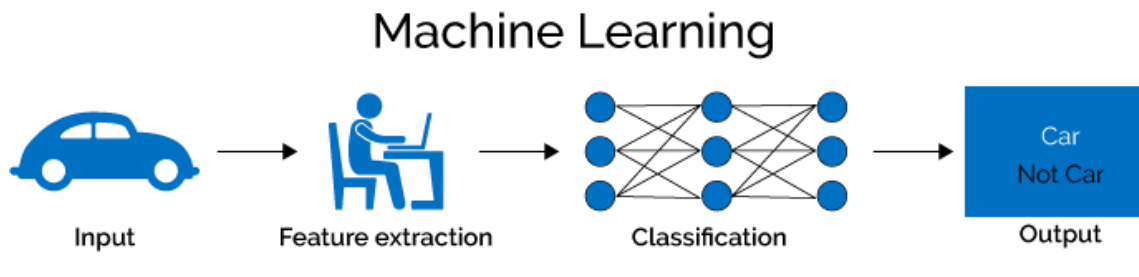
Objectives:

Upon completing the lab, you will:

- Become familiar with the AutoAI feature of Watson Studio.
- Train/Evaluate a machine learning model
- Deploy a machine learning model.
- Deploy a Python Flask web application that we will configure to "call" the deployed machine learning model.
- Configure the application to connect to the machine learning service.
- Update the code in the application to specify the endpoint of the deployed model, and use DevOps to build and re-deploy the application.
- Run the application to demonstrate the use of the deployed machine learning model to score the trafficking risk of a passenger.

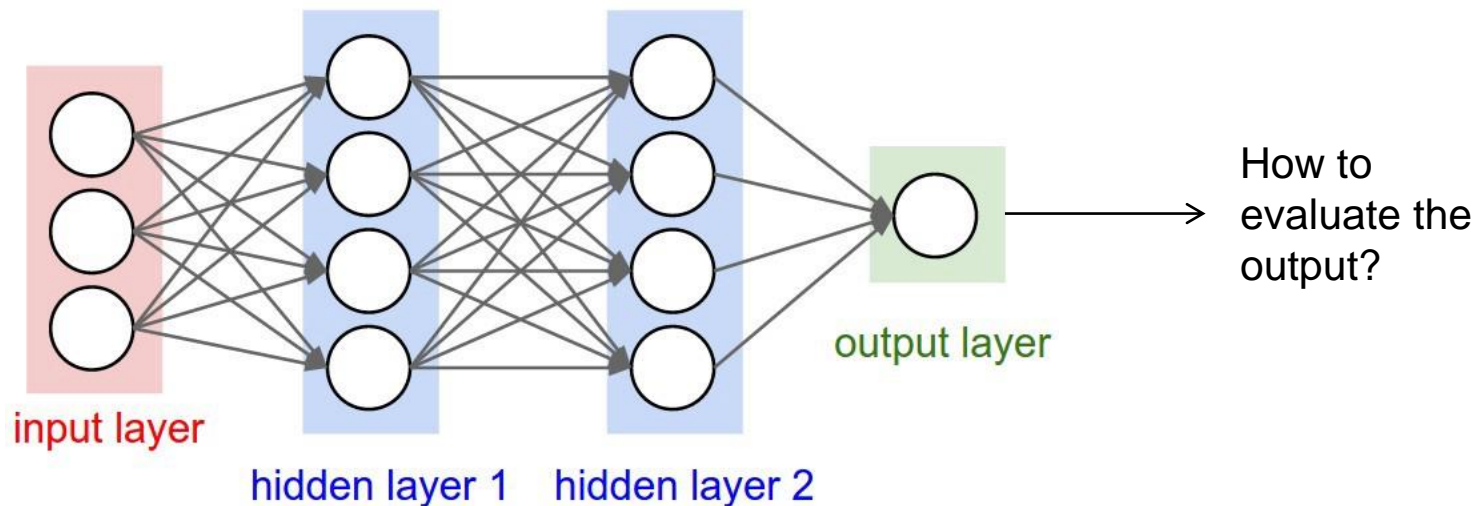
Deep Learning

- Deep Learning is a machine learning method.
- Could be supervised or unsupervised
- Originated in 1940s
- Became very popular this decade
 - Hardware Improvements/Cost – GPUs, Storage
 - Availability of Large Datasets for Training
 - Better performing algorithms.
- Especially good for human perception type task



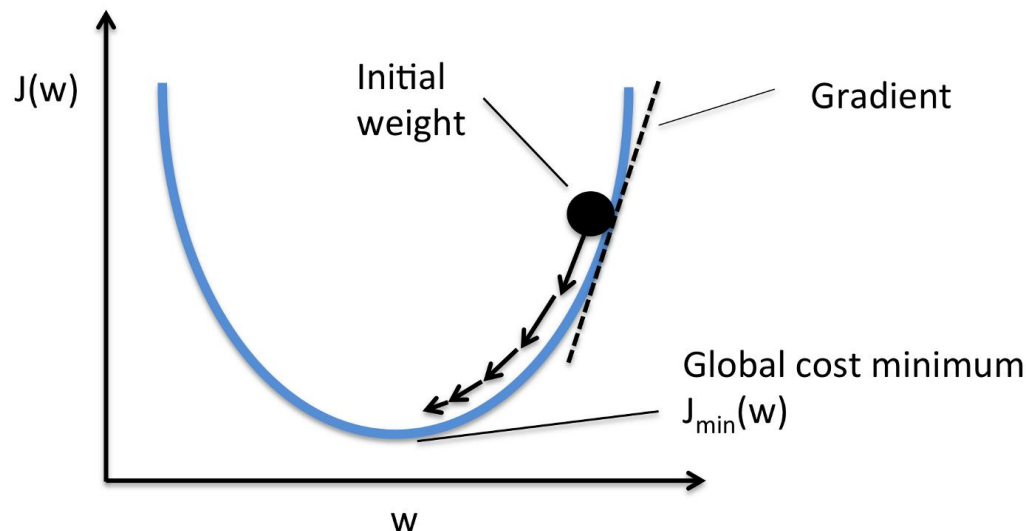
Deep Neural Networks (DNN)

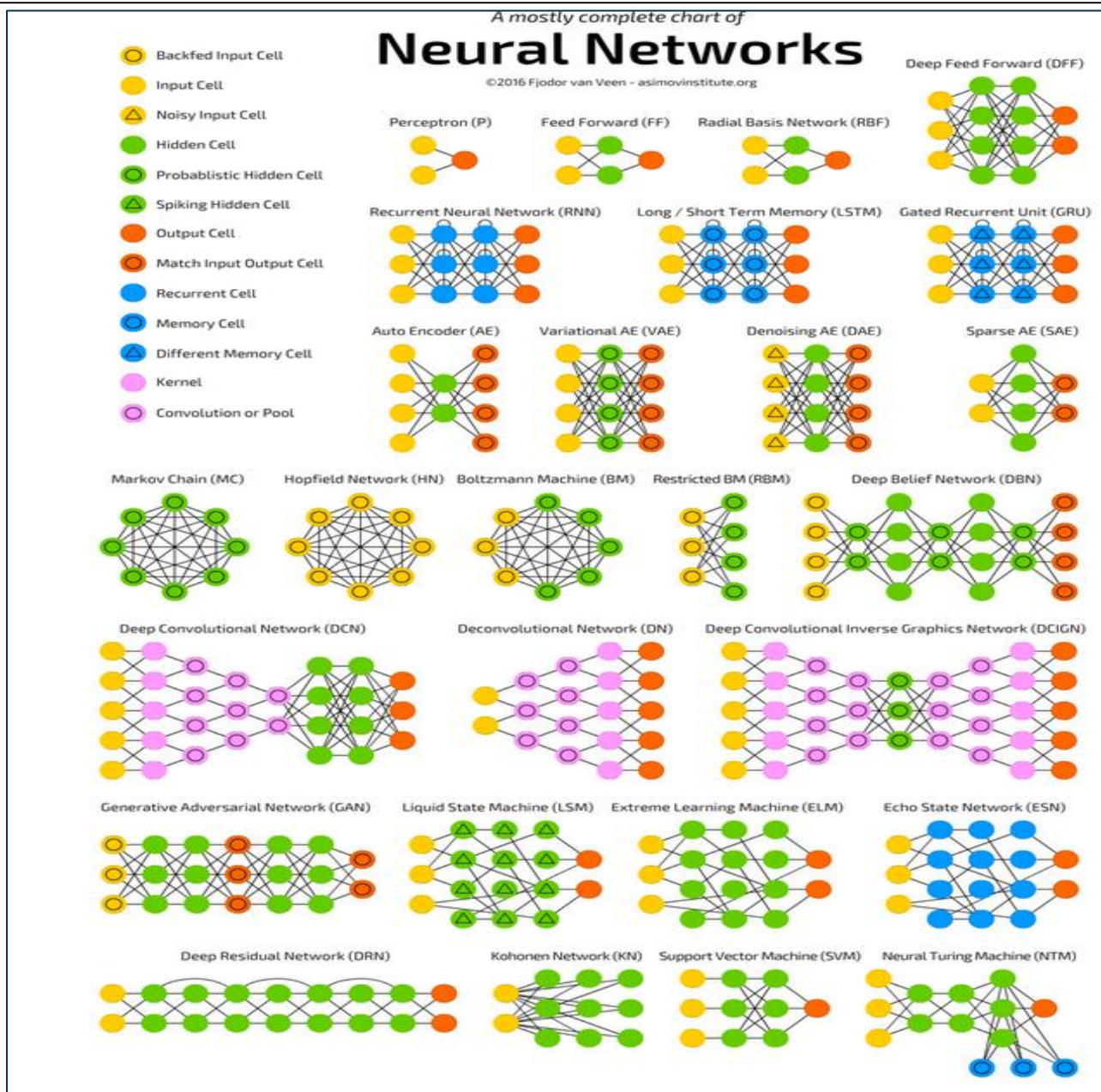
- A Neural Network with more than one hidden layers
 - The **input layer** receives input data.
 - The **hidden layers** perform mathematical computations on our inputs.
 - The **output layer** returns the output data.
- Training the AI is the hardest part of Deep Learning.
 - You need a large data set.
 - You need a large amount of computational power



Cost Function

- During training we need to know how our DNN is doing!
 - Compare it's predictions to dataset's output
 - Based on how far it is from actual value → update weights
- This function that does this is called "Cost Function"
- Ideally, we want our cost function to be zero.
 - Does not happen in real world
 - Instead use techniques like "Gradient Descent" → allows us to find the minimum of a function by iterating through dataset and updating the weights





Common Types of Deep Neural Networks

Convolutional Neural Networks

- Images classifications
- Objects detections
- Recognizing faces
- Natural language processing
- ..

Recurrent Neural Networks

- Speech Recognition
- Handwriting Recognition
- Machine Translation
- Sequence prediction
- ...

Lab-7: Neural Network Modeler

Introduction:

This lab will use the [MNIST](#) computer vision data set to train a convolutional neural network (CNN) model to recognize handwritten digits. The Watson Studio neural network flow editor, Watson Studio experiment builder and the Watson Machine Learning component will be used to build, train, and save the trained model.

Objectives:

Upon completing the lab, you will know how to:

- Create Cloud Object Storage buckets to contain the input and result files
- Create a neural network design from an example using the flow editor
- Use the experiment builder used to set up a training definition to train the neural network model
- Monitor the training progress and results.
- Save the trained model.
- Test the model

Neural Network Modeler

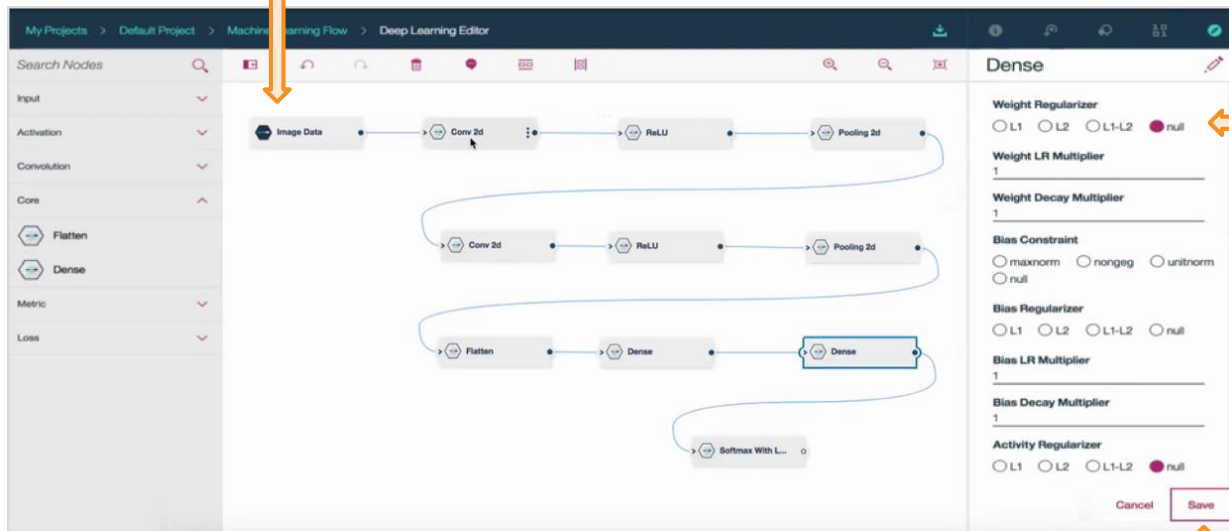
An intuitive drag-and-drop, no-code interface for designing neural network structures using the most popular deep learning frameworks. Quickly capture your network design then single click export for experimental optimization.

Supported Frameworks



Drag-and-drop
network layers

Real-time validation of network
flow

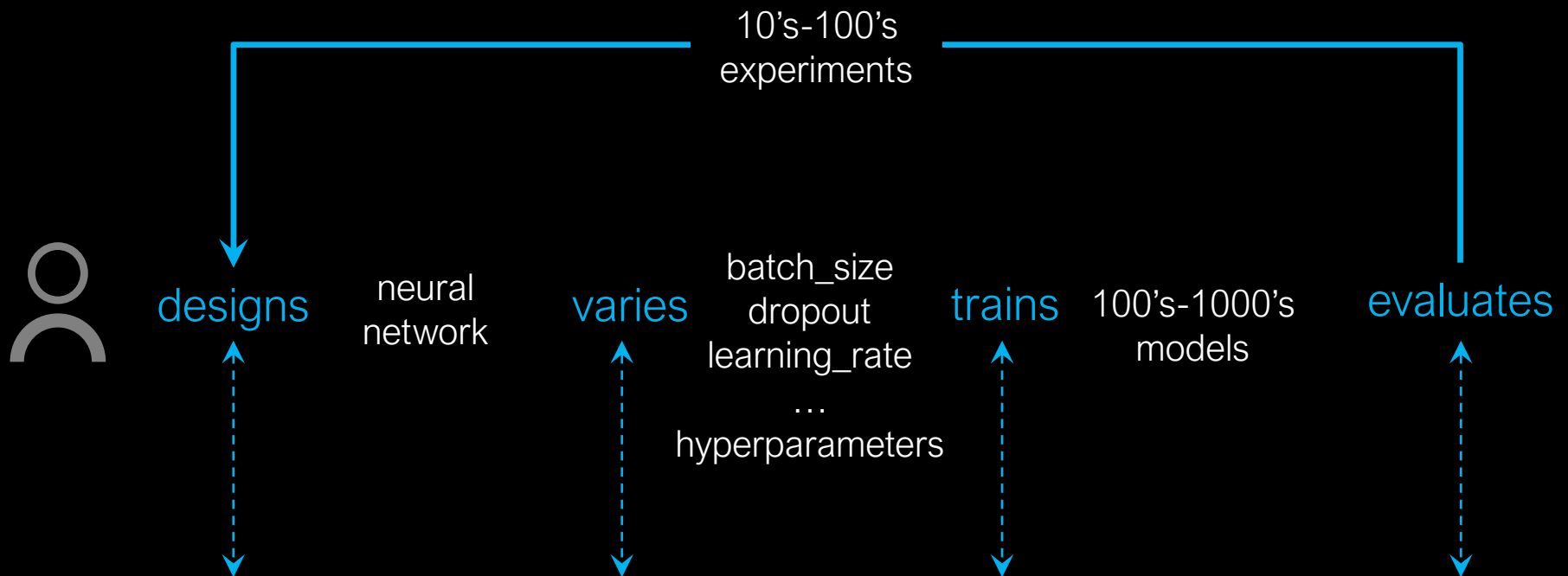


- Define layer configuration
- Choose optimizer params

- Generate CPU or GPU compatible code

- Save as popular framework code
- Export as a python notebook
- Execute as batch experiment

Experiment Builder



Experiment Builder
supports the end-to-end workflow

Lab-8: RStudio and Shiny

Introduction:

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

Objectives:

Upon completing the lab, you will know how to:

- Create an RStudio project from a Git repository
- Establish a connection to a DB2 on Cloud service using an ancillary file
- Query, join, explore and visualize data in a R notebook
- Derive categorical names from numerical levels in a R dataframe
- Use ggplot2 to create bar plots of several of the columns in a R dataframe
- Use a logarithmic scale when creating bar plots
- Close the database connection
- Leverage Shiny to create and run a web application
- Interact with the Shiny web application by running it externally
- Vet additional records in a DB2 database using the web application