Introduction to Watson Studio

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

Bernie Beekman
James Parry
Prithvi Rao

Outline

- Watson Studio Overview

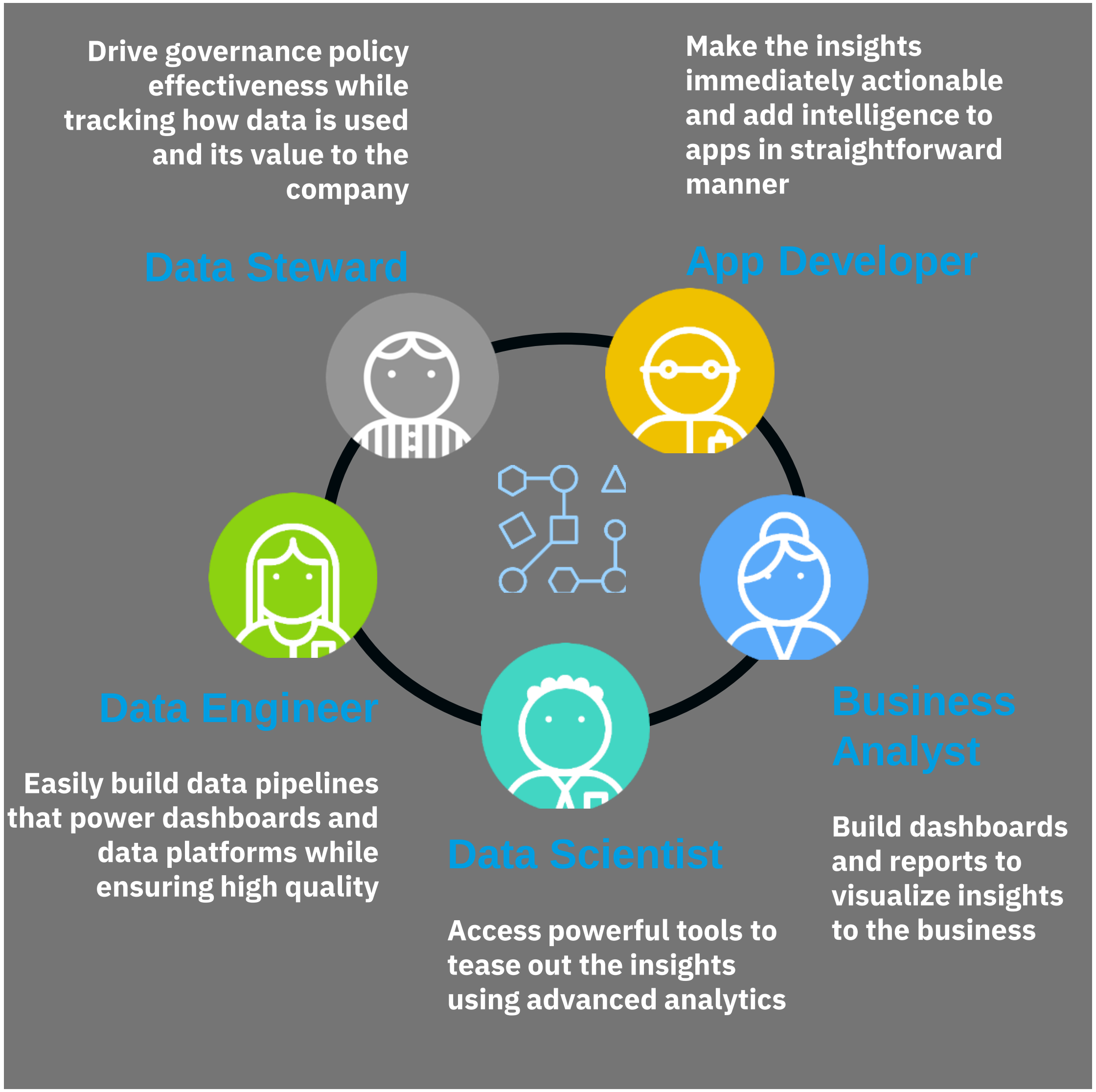
- Lab Overview

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

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.



Drive governance policy effectiveness while tracking how data is used and its value to the company

Data Steward

Make the insights immediately actionable and add intelligence to apps in straightforward manner

App Developer

Business Analyst

Build dashboards and reports to visualize insights to the business

Data Engineer

Easily build data pipelines that power dashboards and data platforms while ensuring high quality

Data Scientist

Access powerful tools to tease out the insights using advanced analytics

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

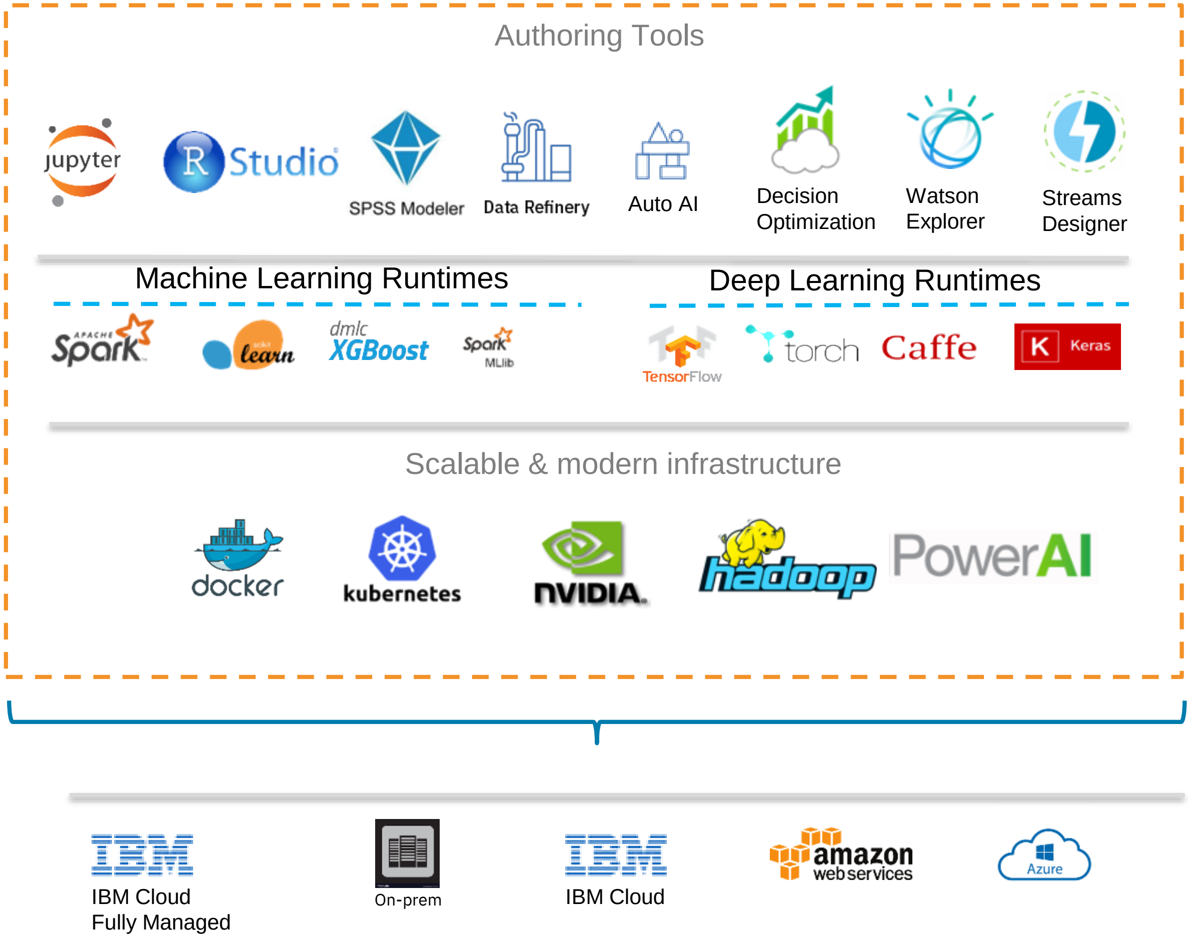
Watson Studio Tools

- 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

Build and train at scale



Authoring Tools

SPSS Modeler

Data Refinery

Auto AI

Decision Optimization

Watson Explorer

Streams Designer

Machine Learning Runtimes

Deep Learning Runtimes

Scalable & modern infrastructure

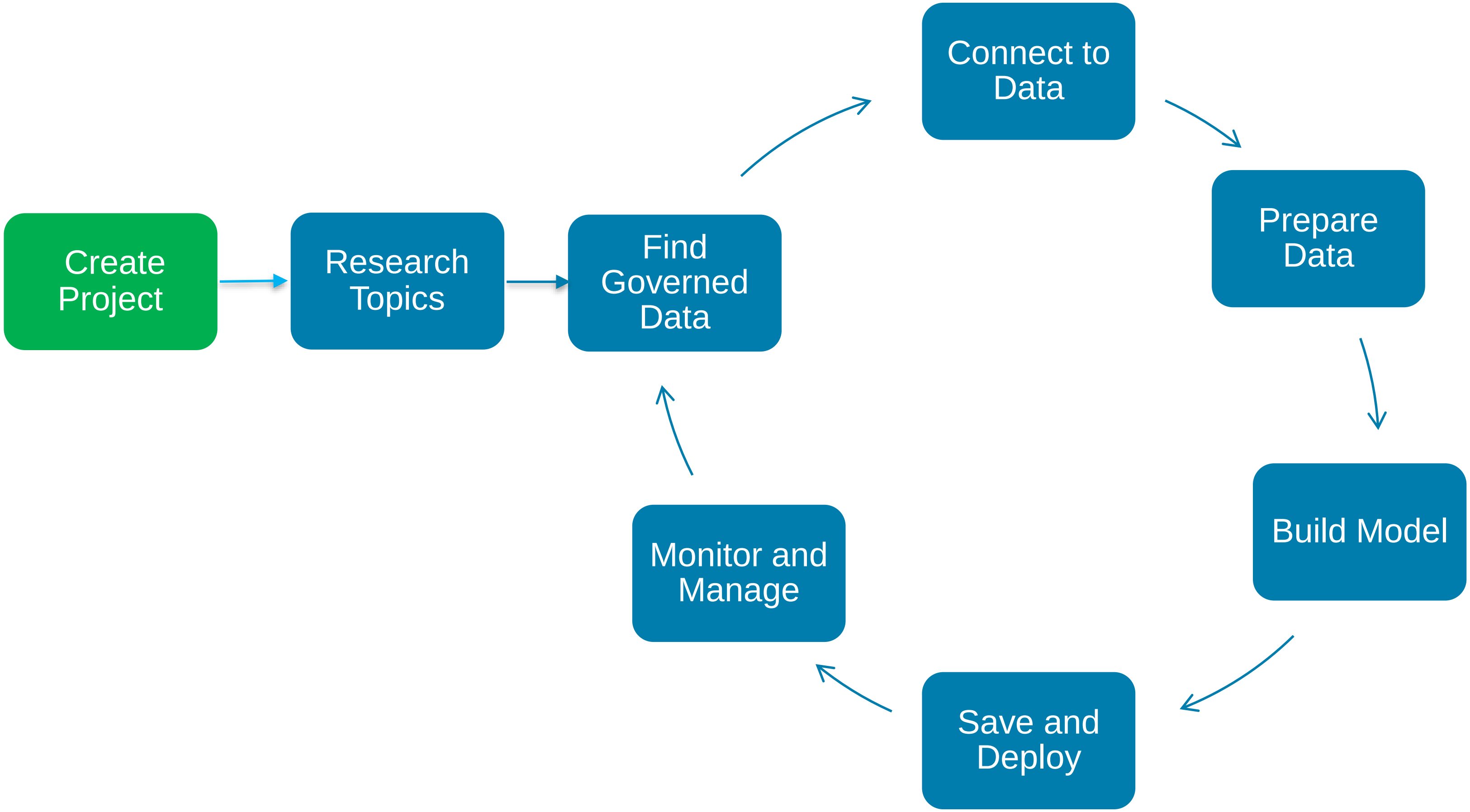
IBM Cloud Fully Managed

On-prem

IBM Cloud

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

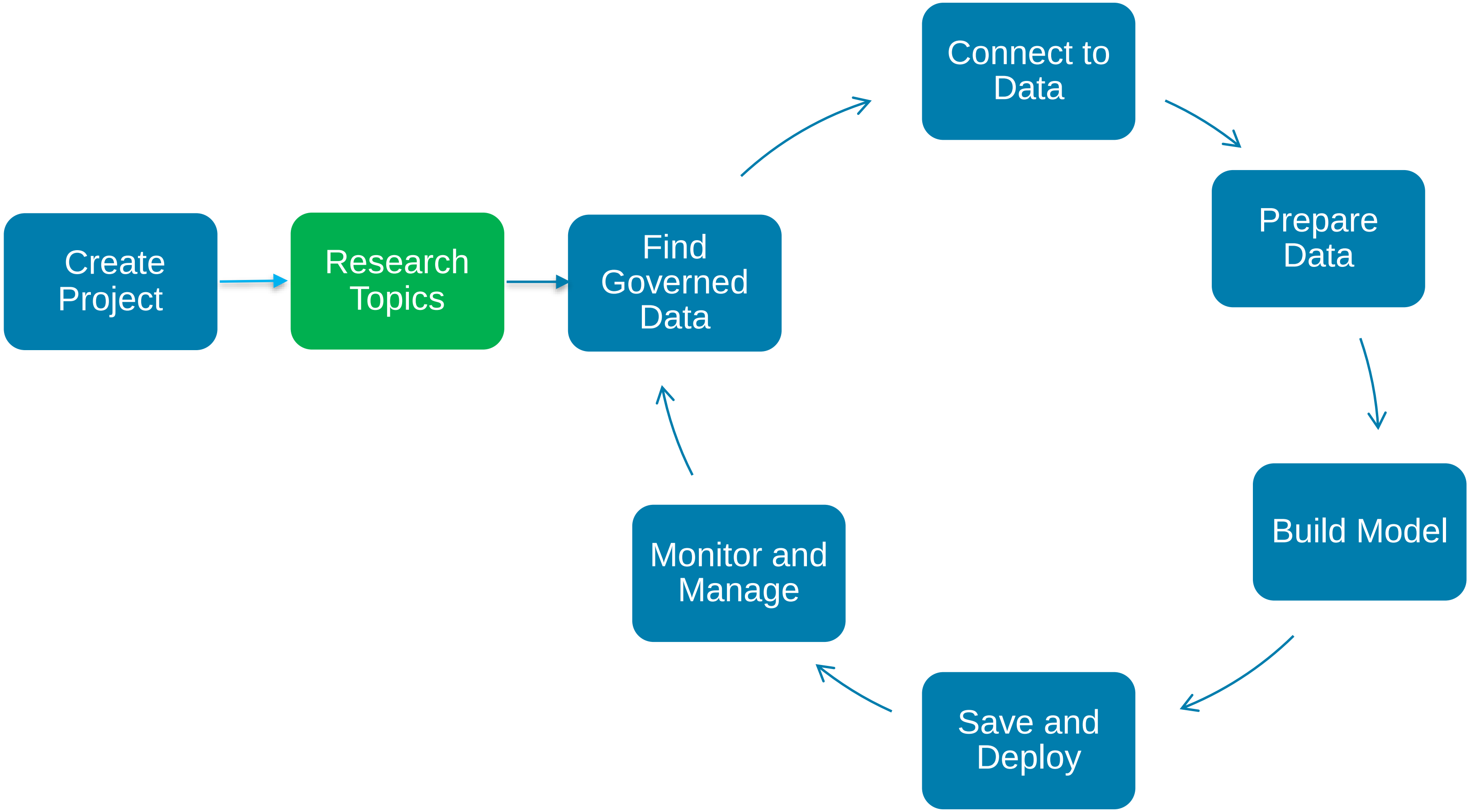
Making Data Science a Team Sport

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.



Research Topics

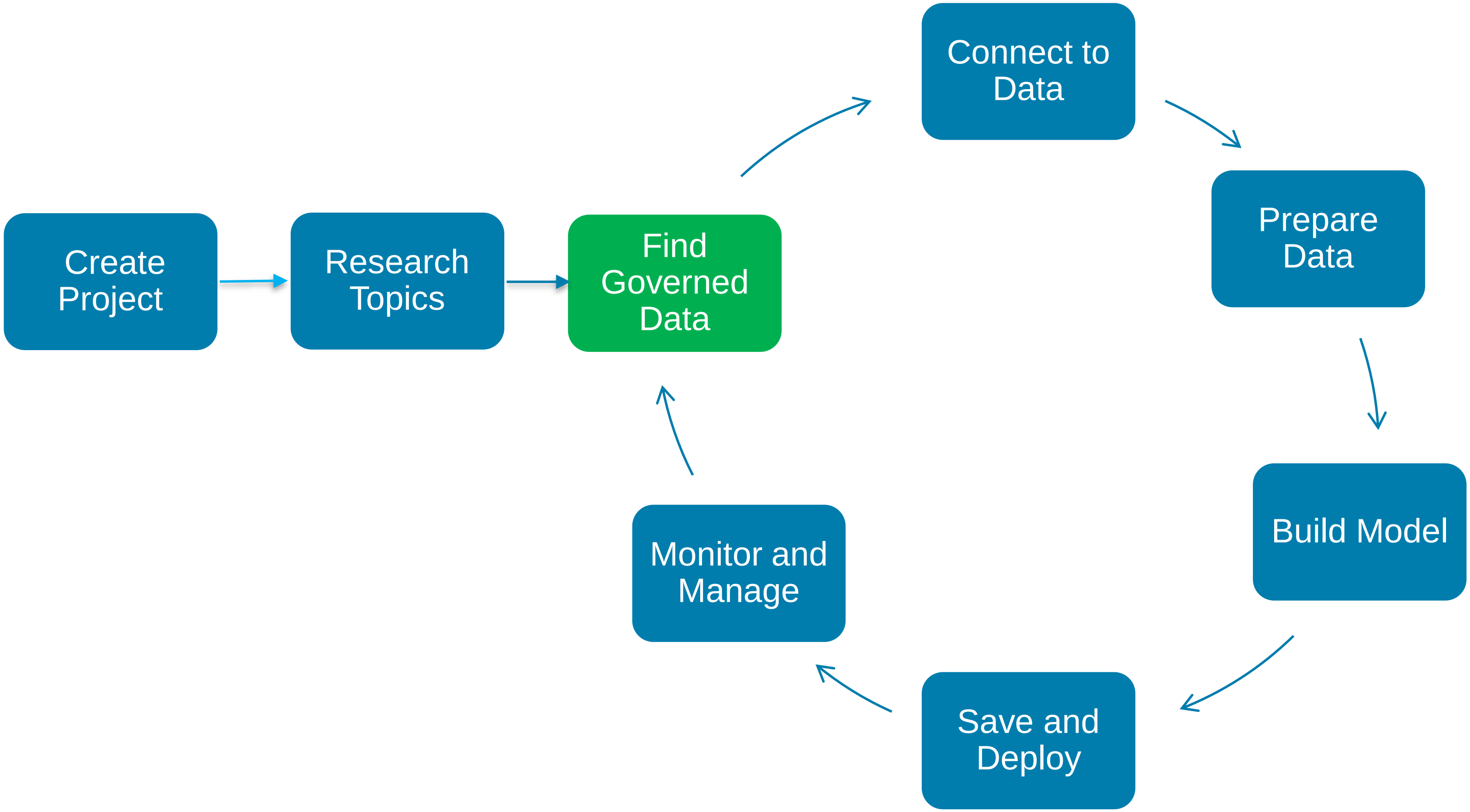
Watson Studio Gallery

Built-in learning to get started

- The Gallery includes notebooks, and data sets
- 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.



Create Project

Research Topics

Find Governed Data

Connect to Data

Prepare Data

Build Model

Save and Deploy

Monitor and Manage

Find Governed Data

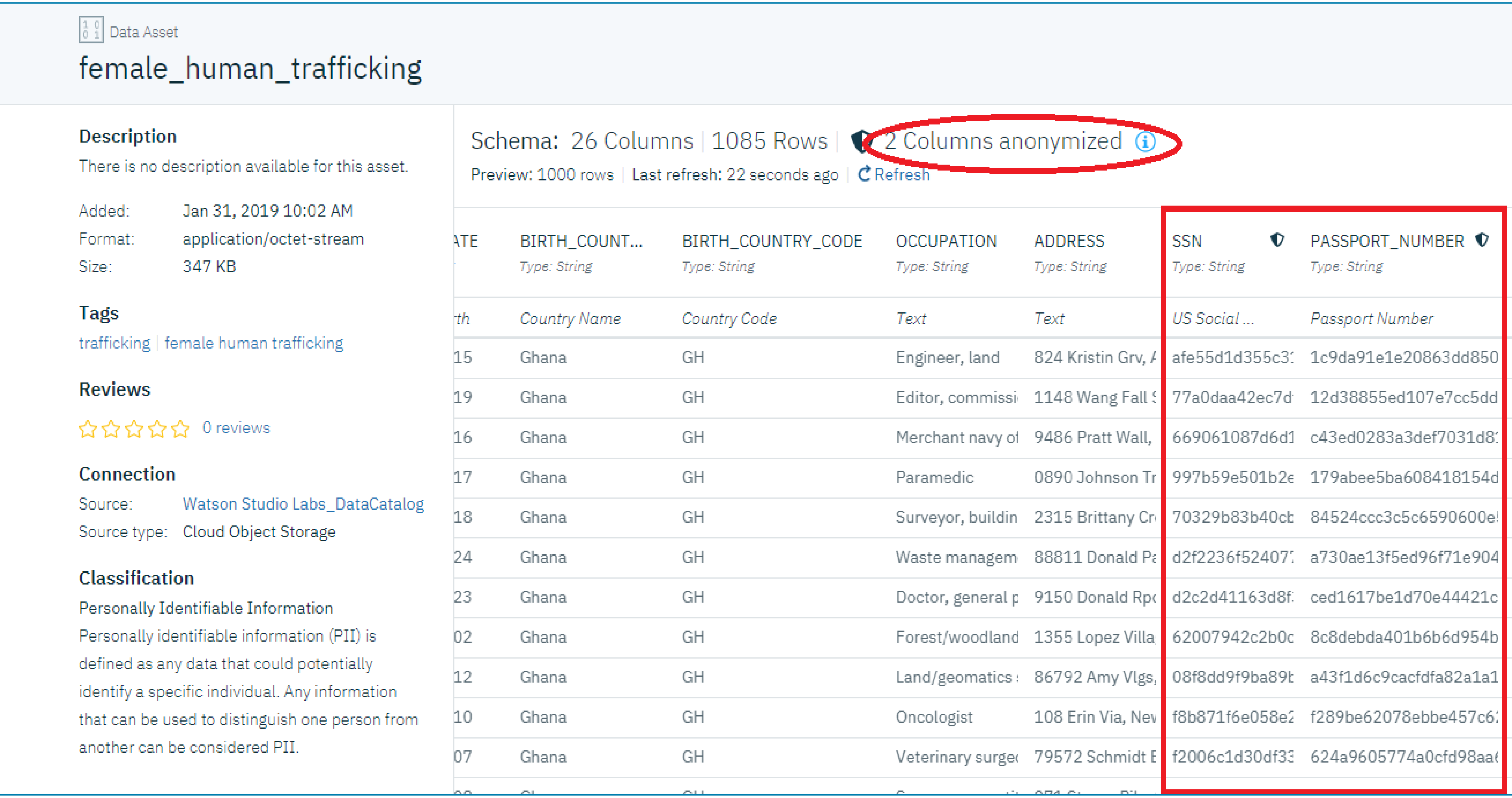
Watson Knowledge Catalog Features

Unlock tribal knowledge and unleash knowledge workers

- **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

Find Governed Data



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.

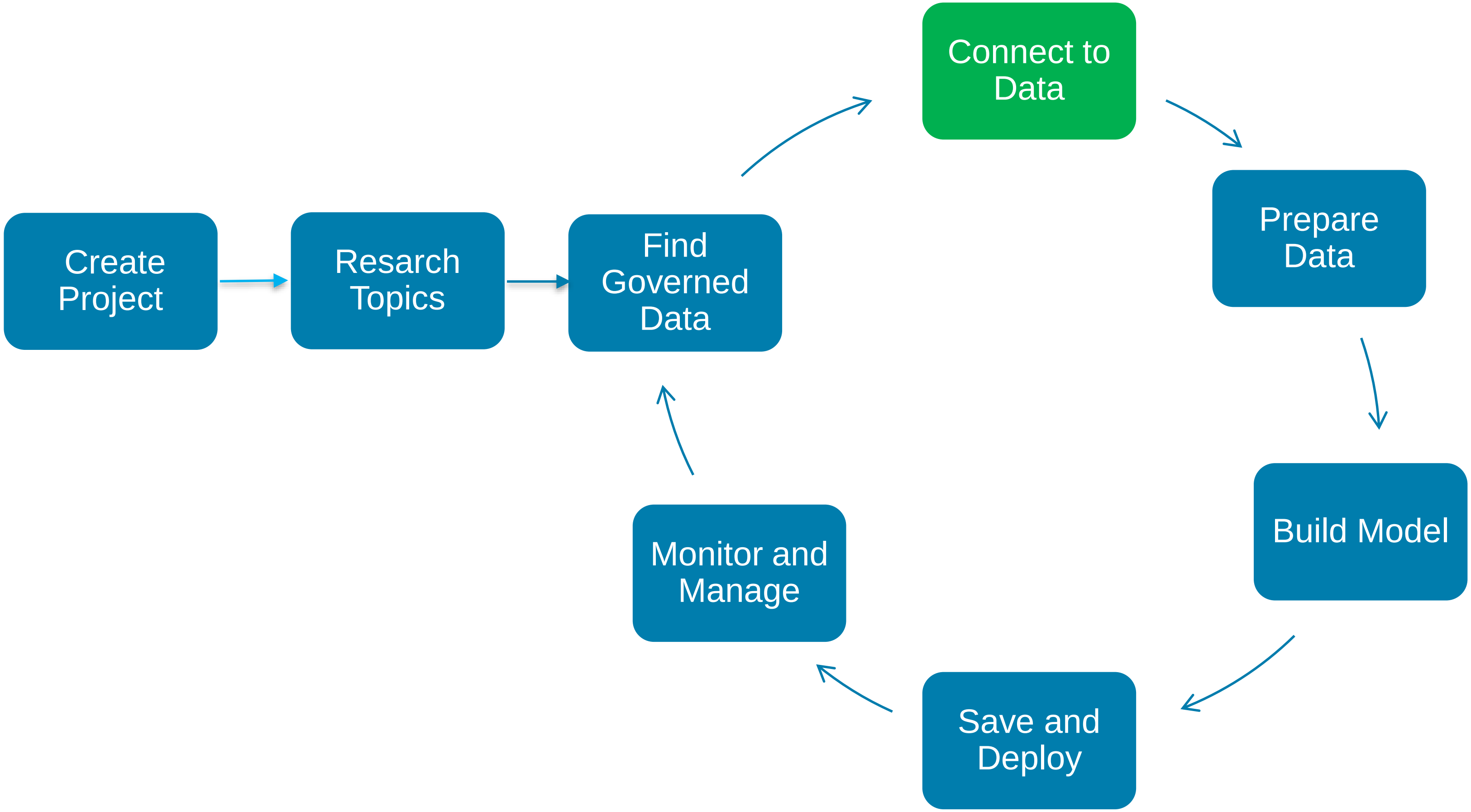
Schema: 26 Columns | 1085 Rows | 2 Columns anonymized ⓘ

Preview: 1000 rows | Last refresh: 22 seconds ago | ↻ Refresh

ATE	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 o	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 Rpc	d2c2d41163d8f	ced1617be1d70e44421c
02	Ghana	GH	Forest/woodland	1355 Lopez Villa	62007942c2b0c	8c8debda401b6b6d954b
12	Ghana	GH	Land/geomatics	86792 Amy Vlgs,	08f8dd9f9ba89b	a43f1d6c9cacfdfa82a1a1
10	Ghana	GH	Oncologist	108 Erin Via, New	f8b871f6e058e2	f289be62078ebbe457c6
07	Ghana	GH	Veterinary surge	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.

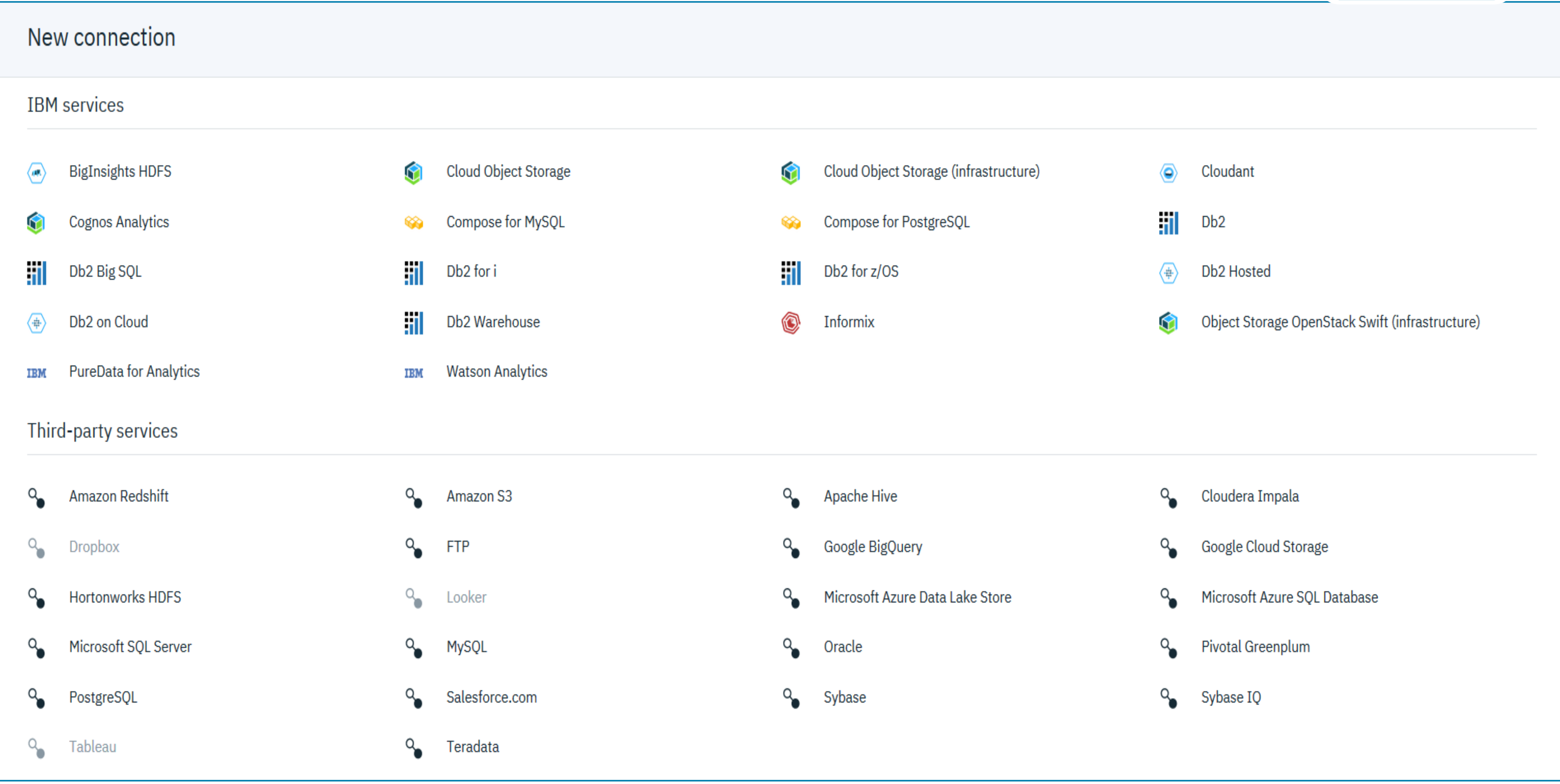


Watson Studio Connection Features

- 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

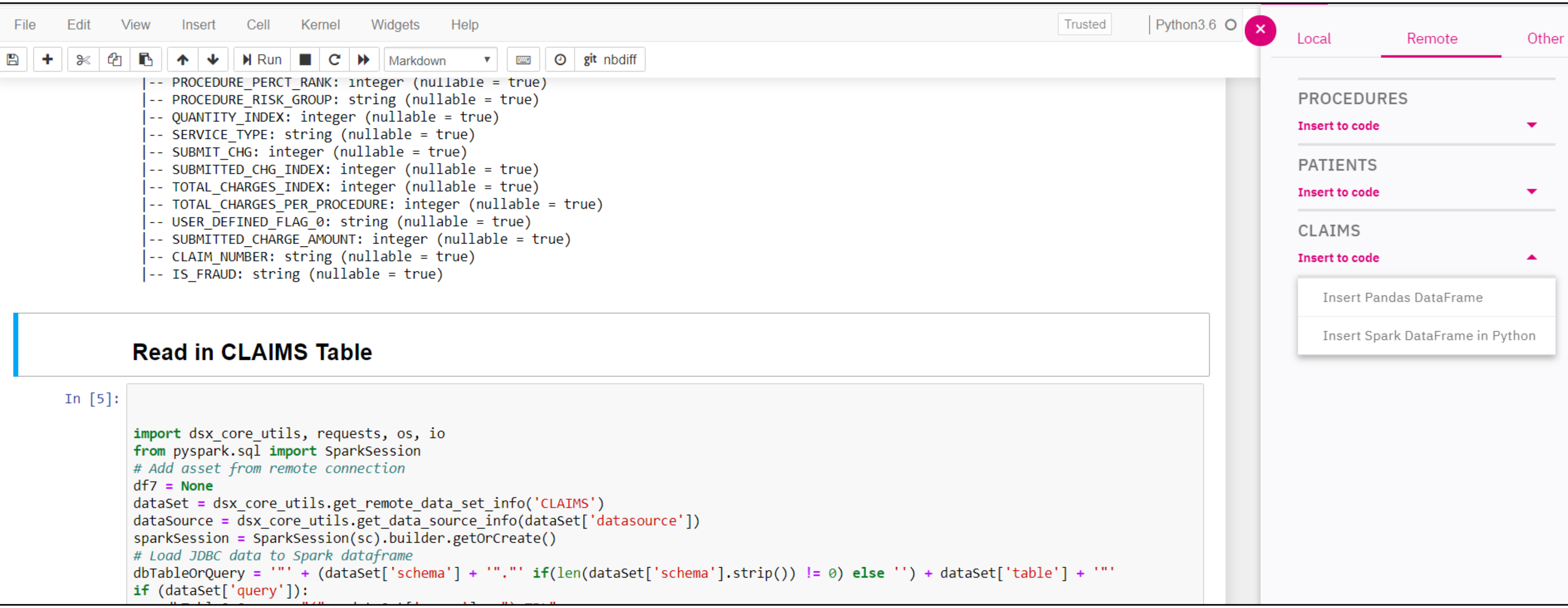
- BigInsights HDFS
- Cloud Object Storage
- Cloud Object Storage (infrastructure)
- Cloudant
- Cognos Analytics
- Compose for MySQL
- Compose for PostgreSQL
- Db2
- Db2 Big SQL
- Db2 for i
- Db2 for z/OS
- Db2 Hosted
- Db2 on Cloud
- Db2 Warehouse
- Informix
- Object Storage OpenStack Swift (infrastructure)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



```
 |-- 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)
```

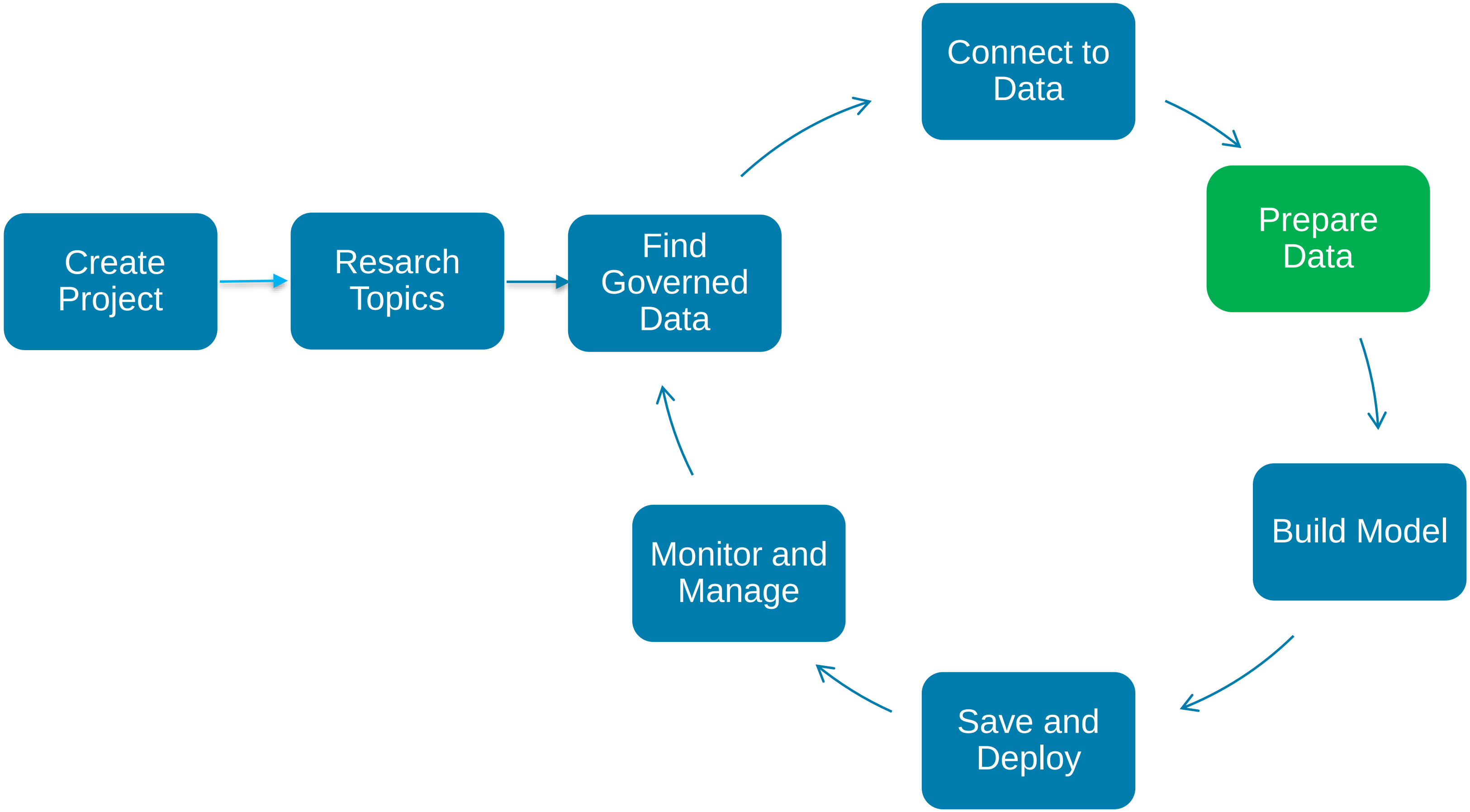
Read in CLAIMS Table

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'] + '"."' 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

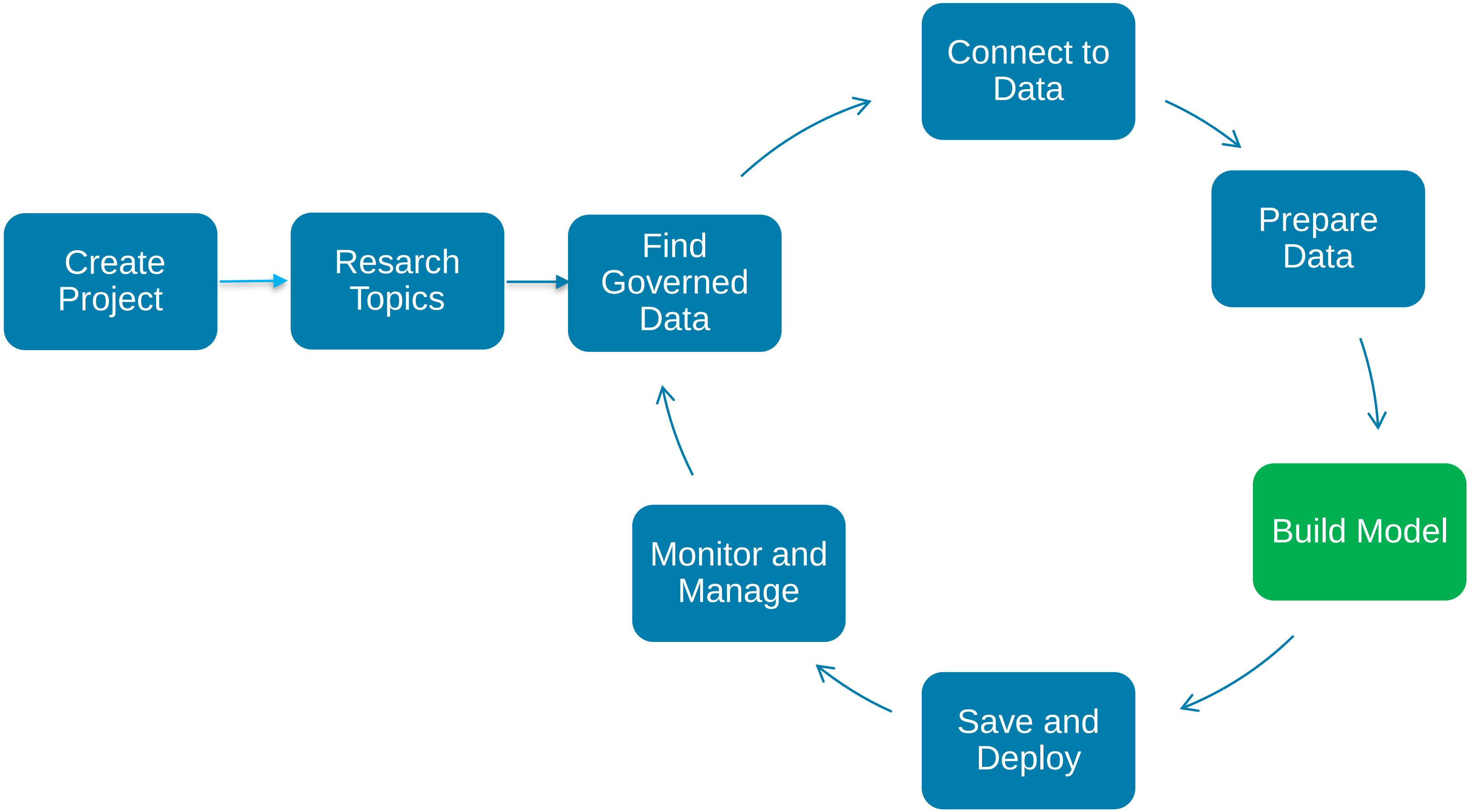
Making Data fit for use

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

Build Model

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

Open Source Tools

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

IBM Tools

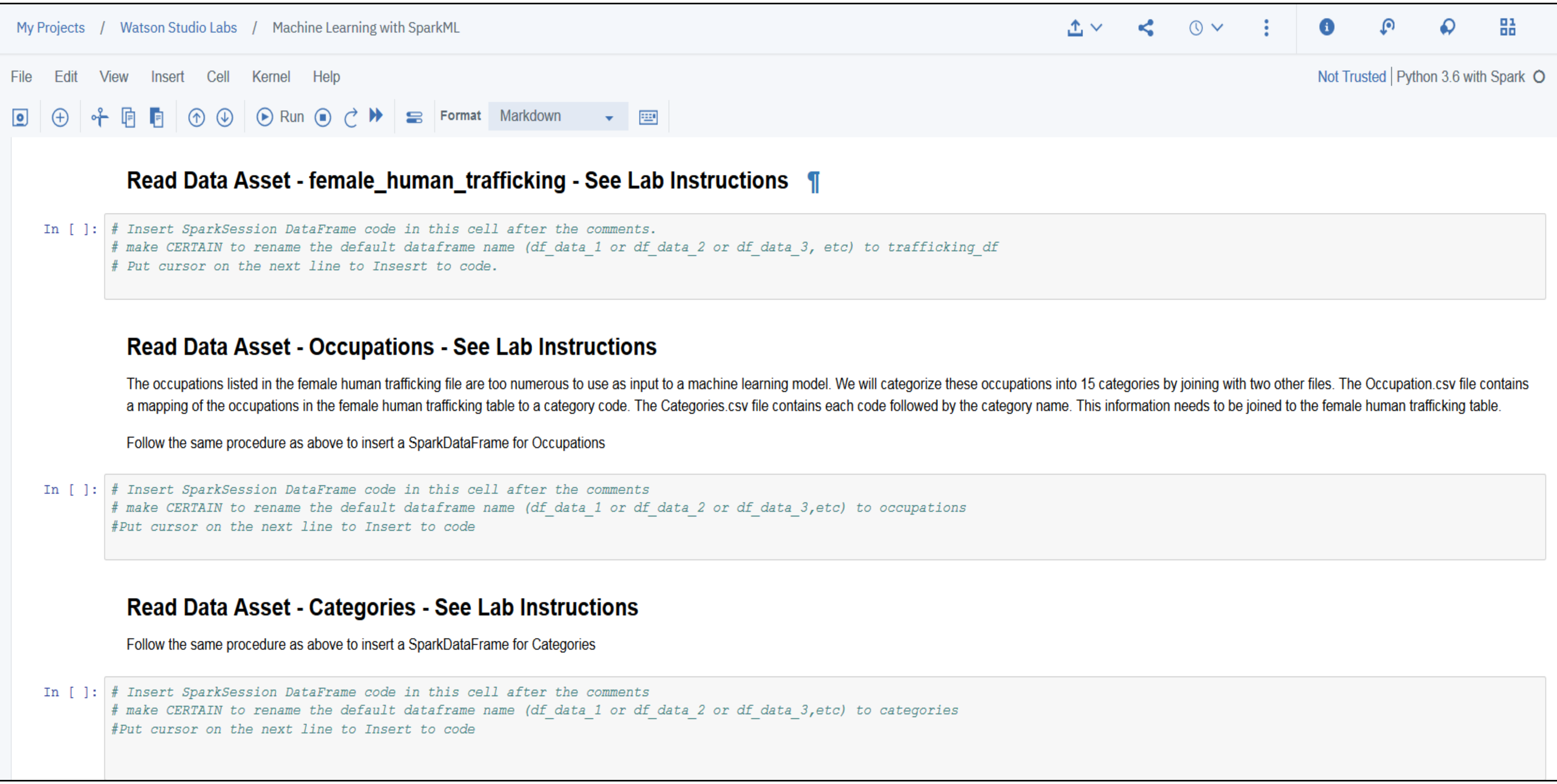
- AutoAI **
- SPSS Modeler**
- Neural Network Modeler**
- Experiment Builder**
- Natural Language Classifier Model
- Visual Recognition Model

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
Run
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 Insesrt 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
```

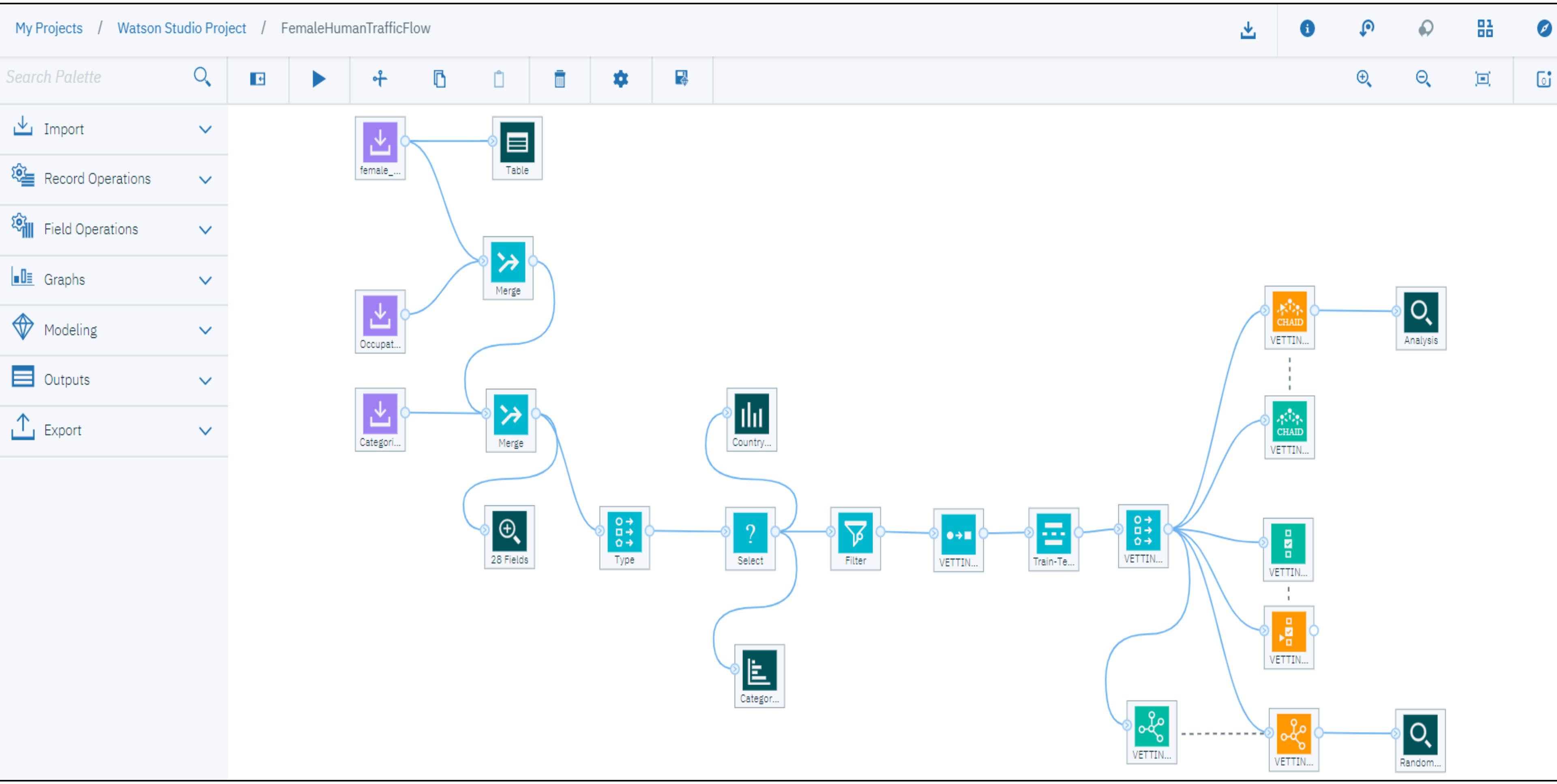
Read Data Asset - Categories - See Lab Instructions

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
```

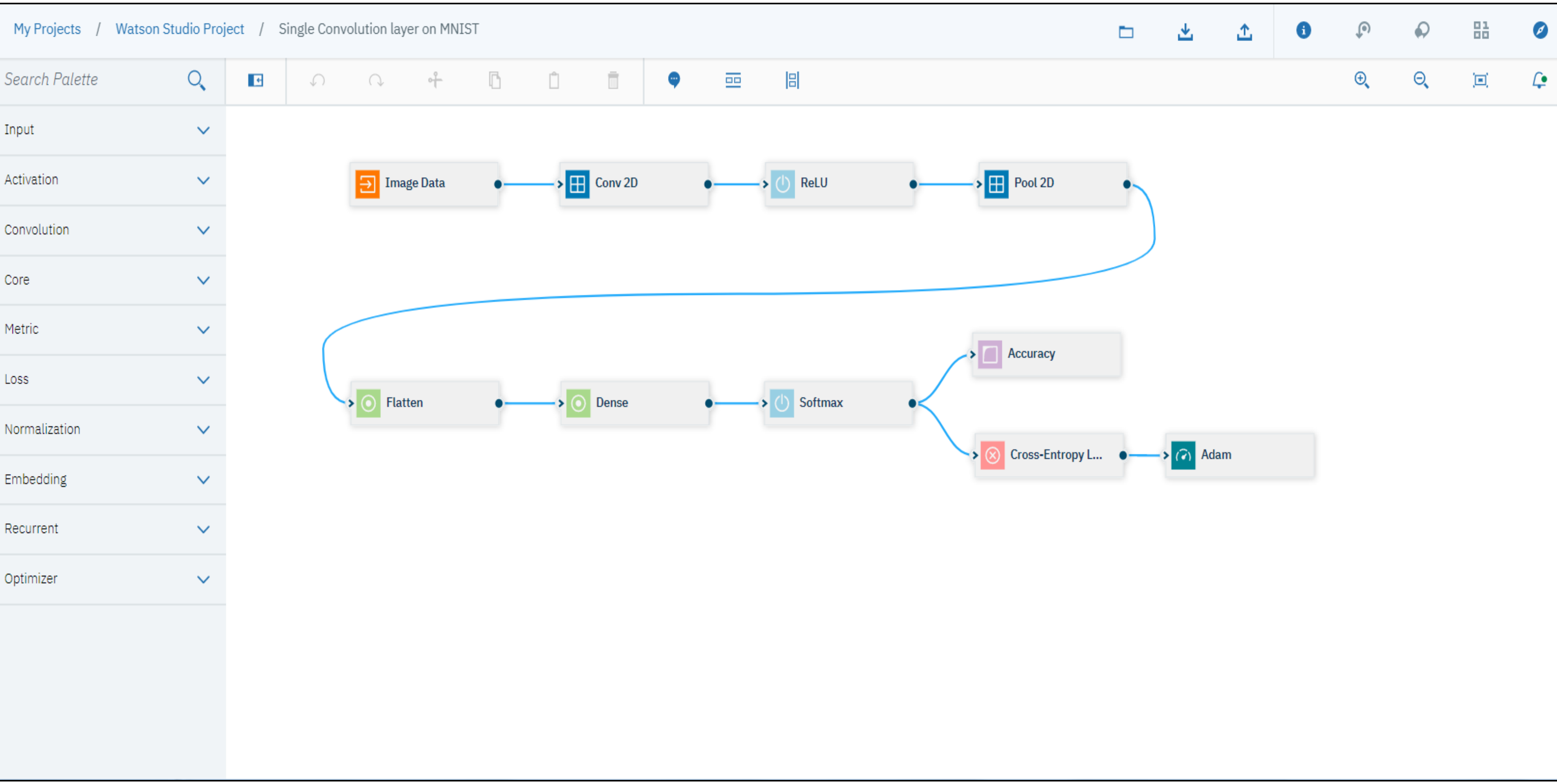
SPSS Modeler

Build Model



Neural Network Modeler

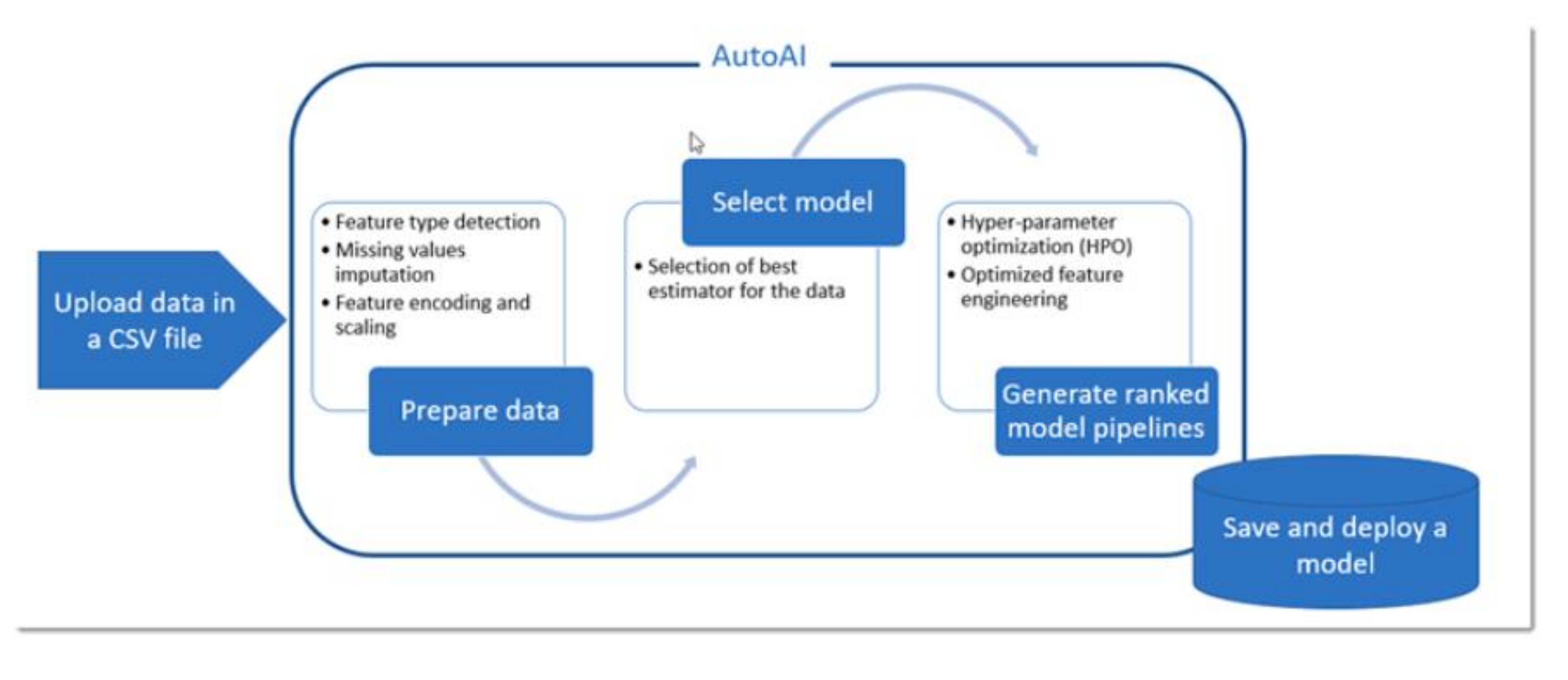
Build Model



My Projects / Watson Studio Project / Single Convolution layer on MNIST

AutoAI

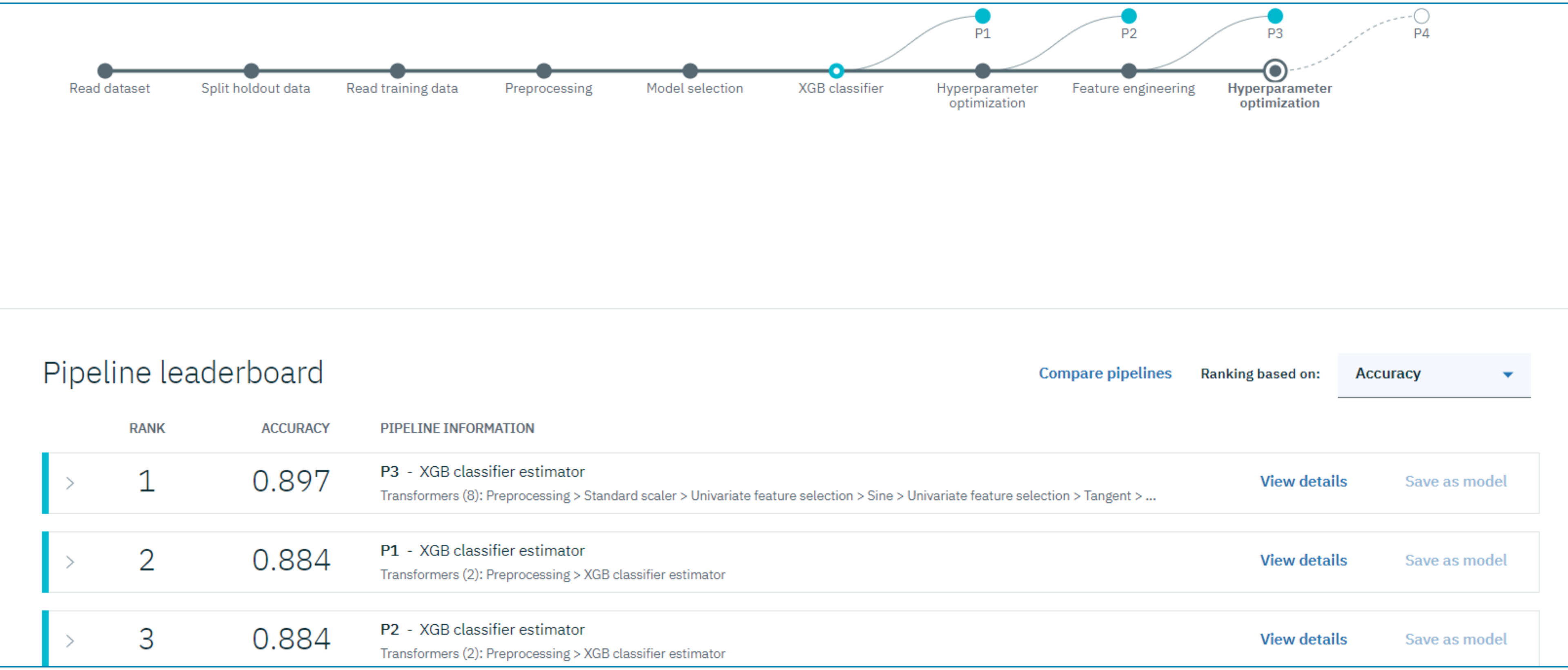
Build Model



AutoAI
Upload data in a CSV file
Feature type detection
Missing values imputation
Feature encoding and scaling
Prepare data
Select model
Selection of best estimator for the data
Hyper-parameter optimization (HPO)
Optimized feature engineering
Generate ranked model pipelines
Save and deploy a model

AutoAI

Build Model



Pipeline leaderboard

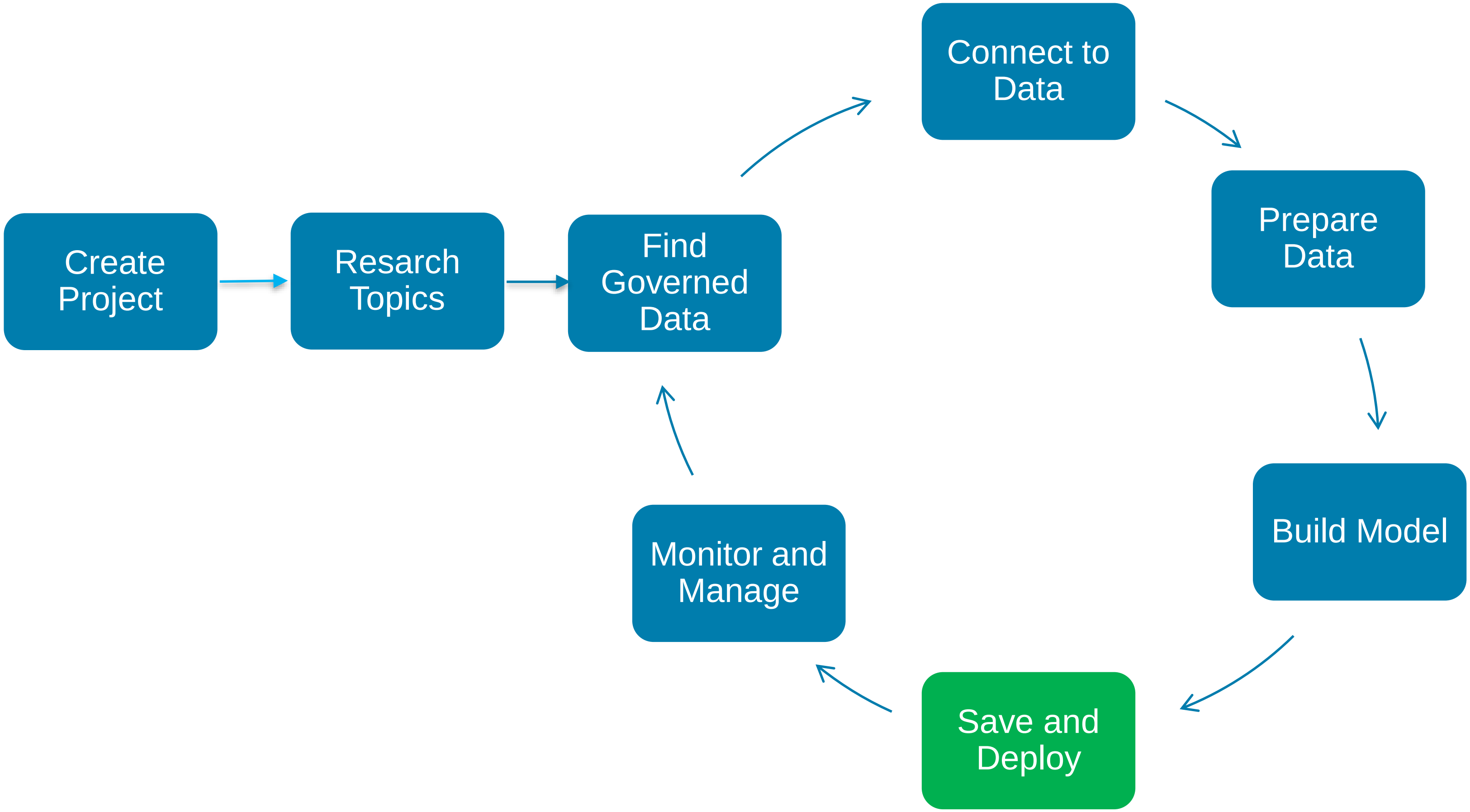
Compare pipelines

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

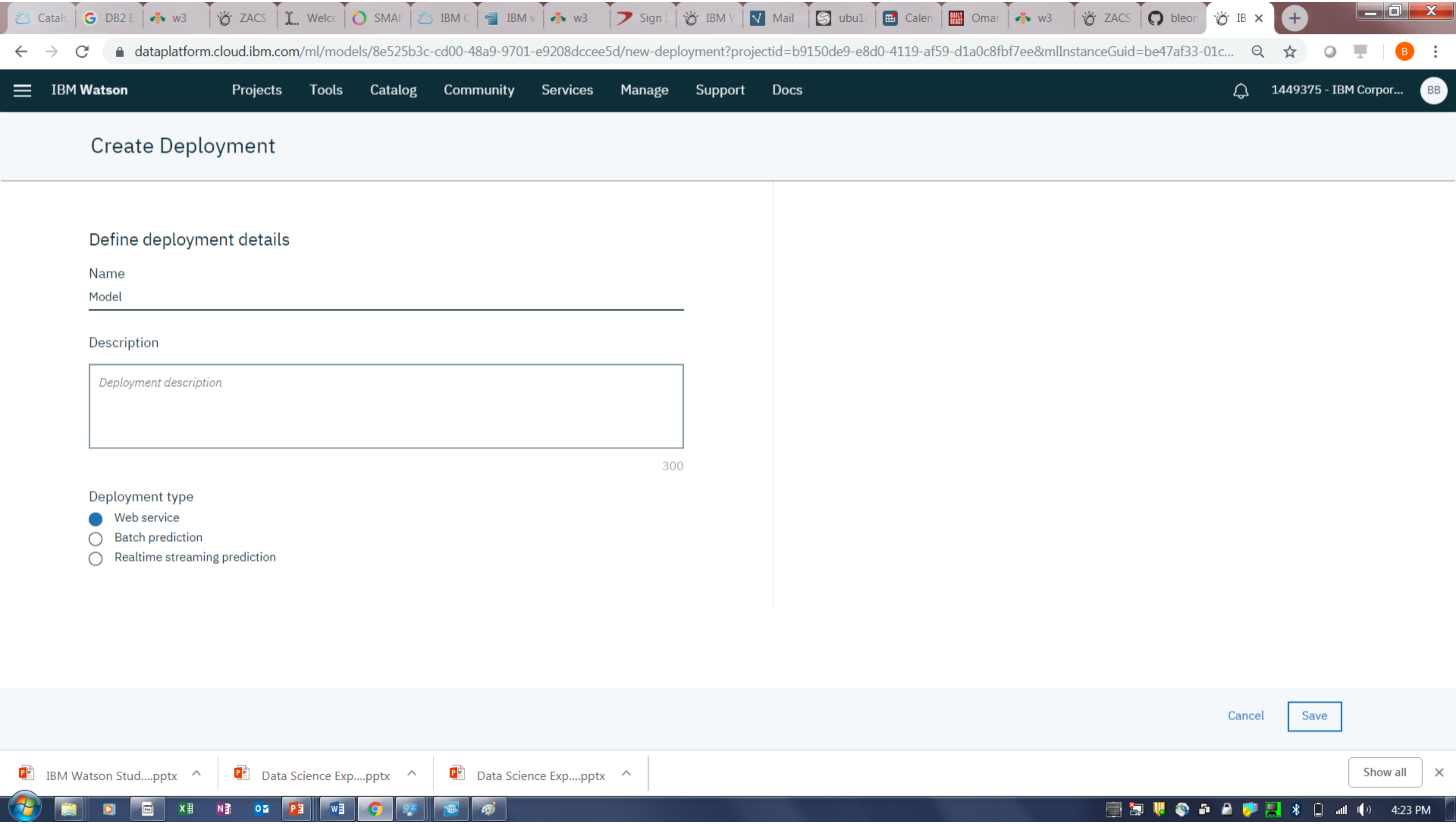
Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.



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...

Create Deployment

Define deployment details

Name

Model

Description

Deployment description

300

Deployment type

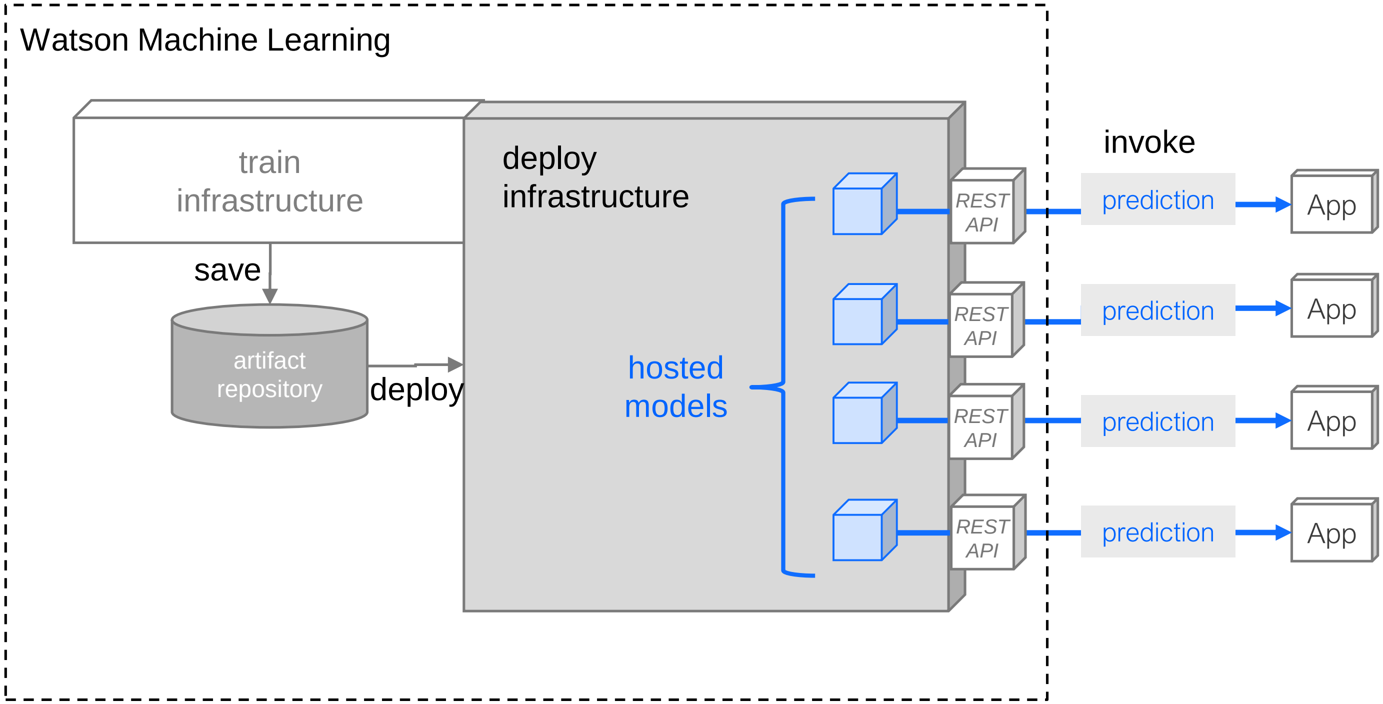
● Web service

○ Batch prediction

○ Realtime streaming prediction

Watson Studio Save and Deploy Trained Models

Save and Deploy Models with Watson Machine Learning



Watson Studio Save and Deploy Features

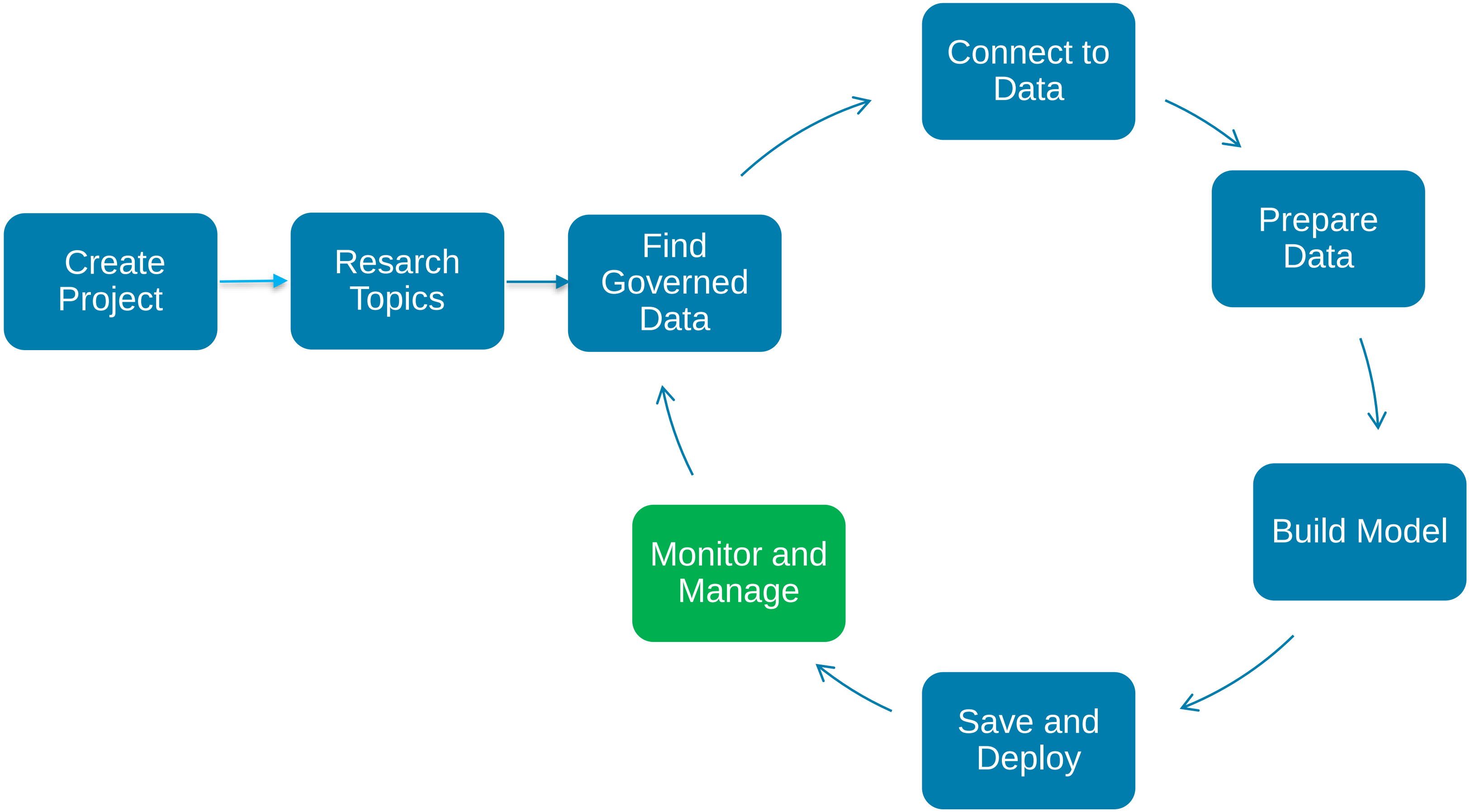
Save and Deploy

Save and Deploy Models with Watson Machine Learning

- 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.



Create Project

Resarch Topics

Find Governed Data

Connect to Data

Prepare Data

Build Model

Save and Deploy

Monitor and Manage

Monitor and Manage

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

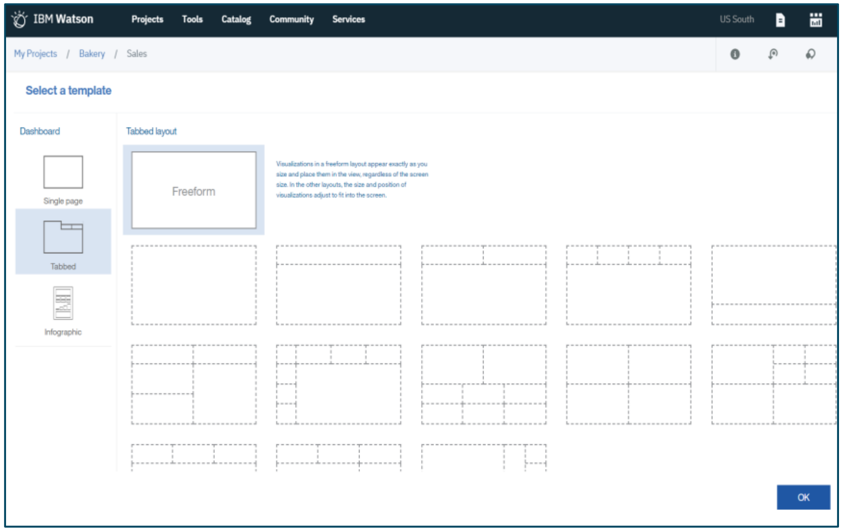
Open By Design

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

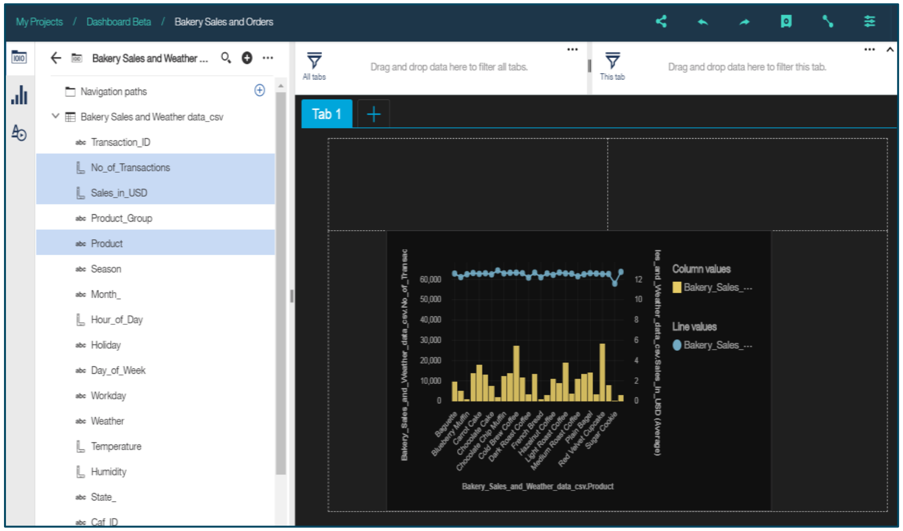
* https://arxiv.org/abs/1901.06261

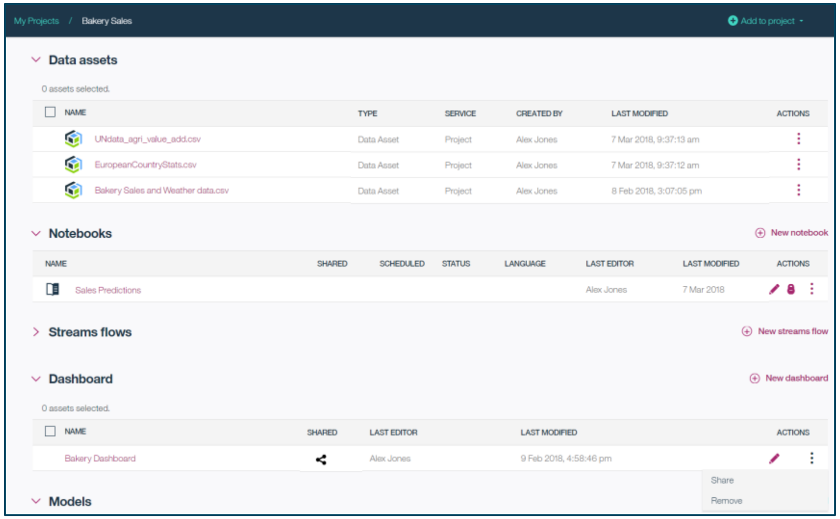
Watson Studio Dynamic Dashboards

Making insights available to all



Visualizations in a freeform layout appear exactly as you size and place them in the view, regardless of the screen size. In the other layouts, the size and position of visualizations adjust to fit into the screen.





Data assets

0 assets selected.

☐ 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

⊕ New notebook

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

Streams flows

⊕ New streams flow

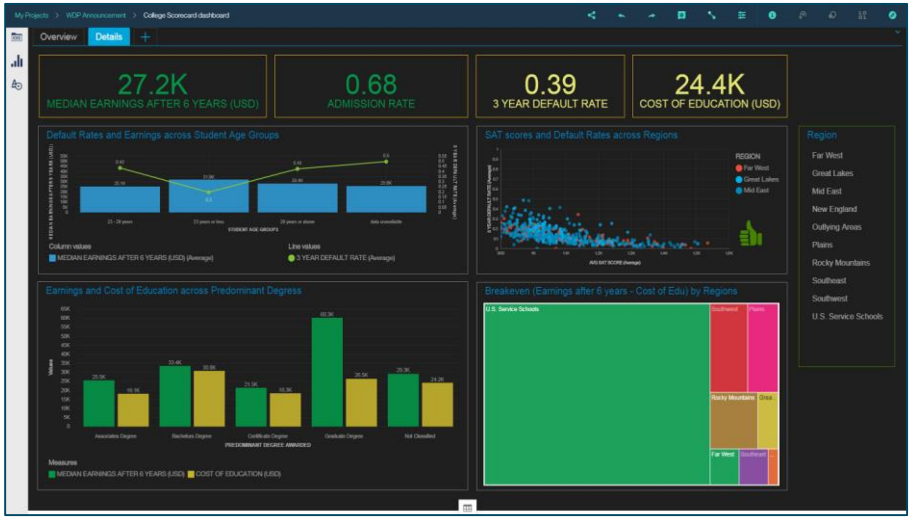
Dashboard

⊕ New dashboard

0 assets selected.

☐ 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.

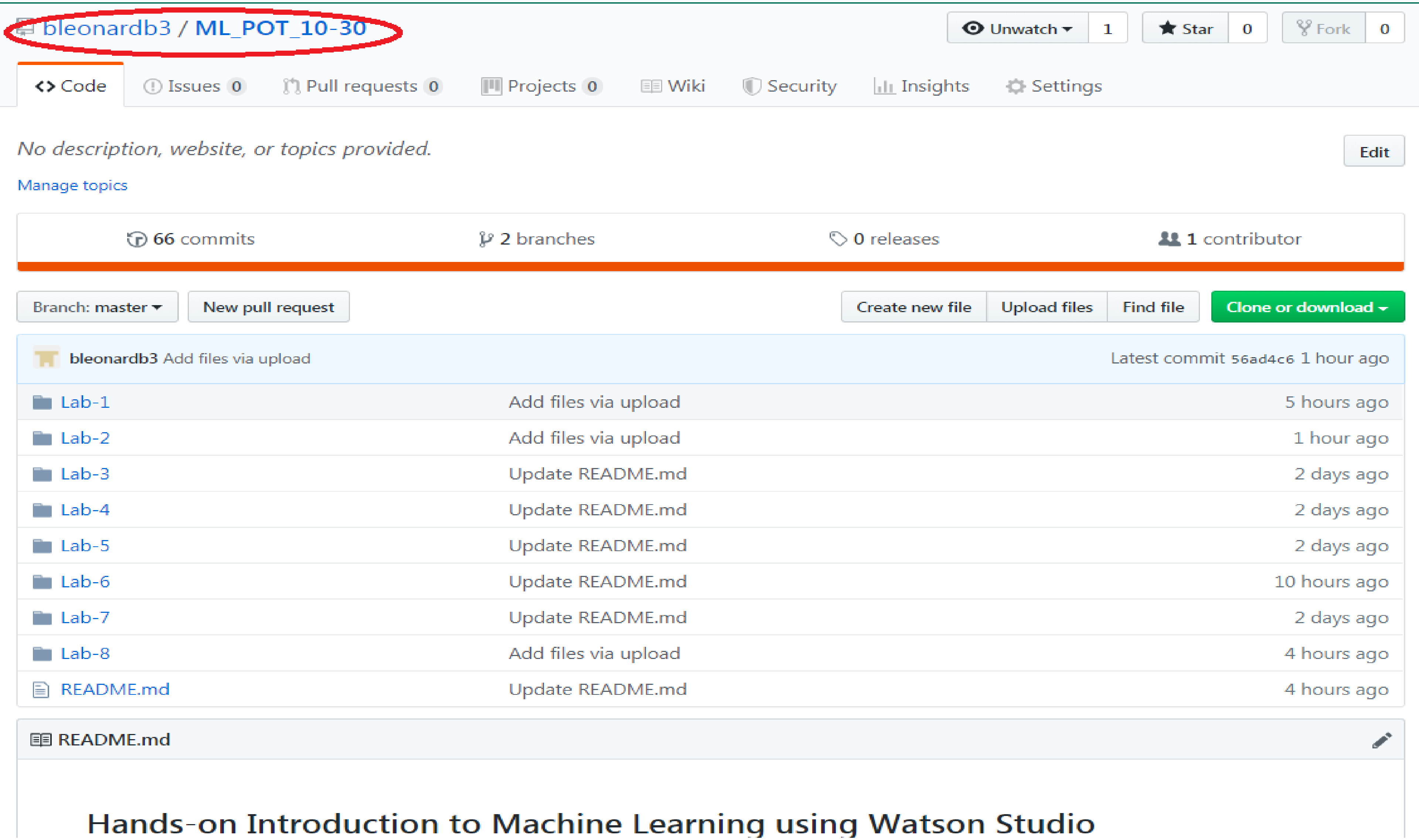
Integrated with Knowledge Catalog

- 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

- Watson Studio Overview
- Lab Overview

Github Repository



No description, website, or topics provided.

66 commits · 2 branches · 0 releases · 1 contributor

bleonardb3 Add files via upload — Latest commit 56ad4c6 1 hour ago

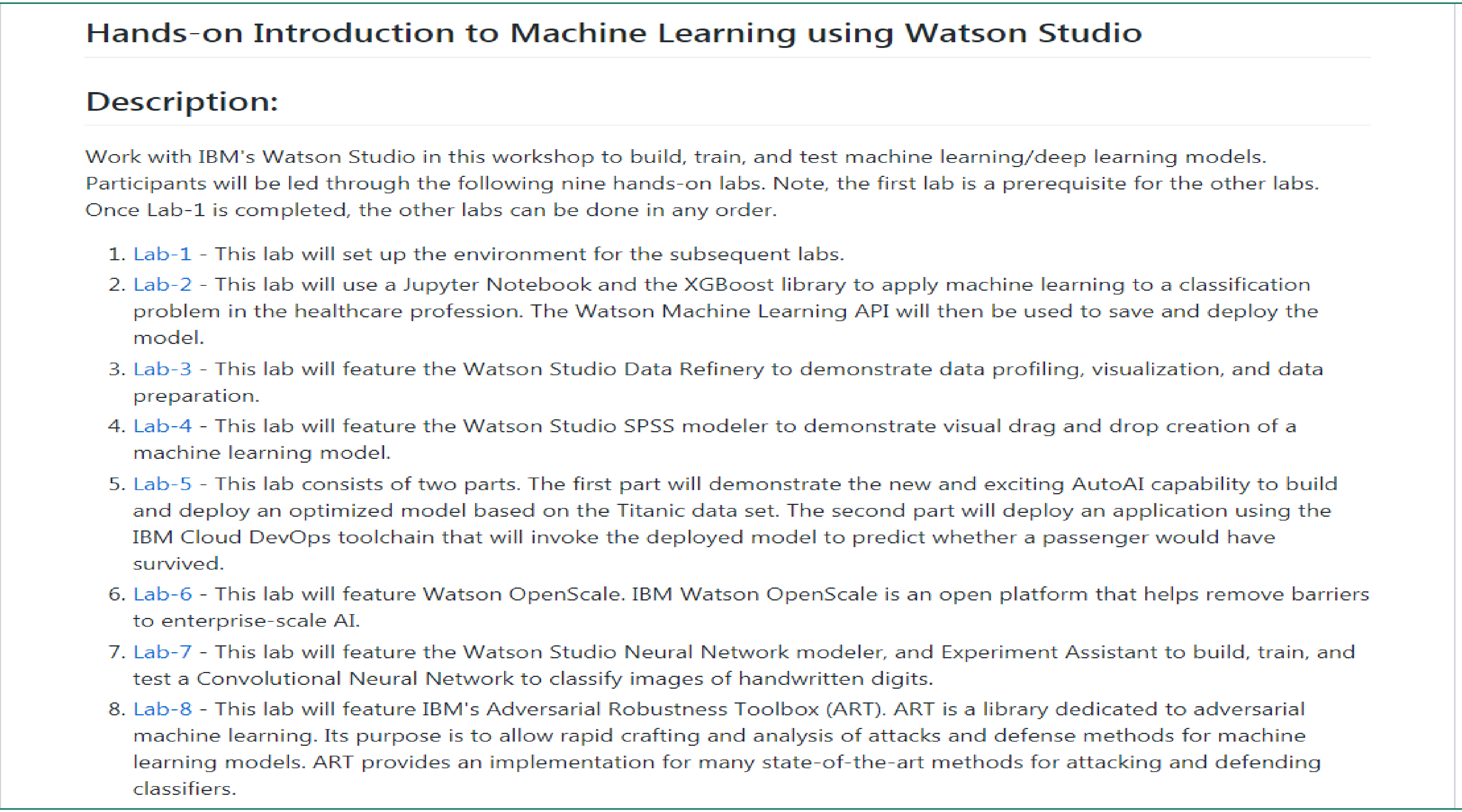
Lab-1	Add files via upload	5 hours ago
Lab-2	Add files via upload	1 hour ago
Lab-3	Update README.md	2 days ago
Lab-4	Update README.md	2 days ago
Lab-5	Update README.md	2 days ago
Lab-6	Update README.md	10 hours ago
Lab-7	Update README.md	2 days ago
Lab-8	Add files via upload	4 hours ago
README.md	Update README.md	4 hours ago

README.md

Hands-on Introduction to Machine Learning using Watson Studio

Github Repository

Readme



Hands-on Introduction to Machine Learning using Watson Studio

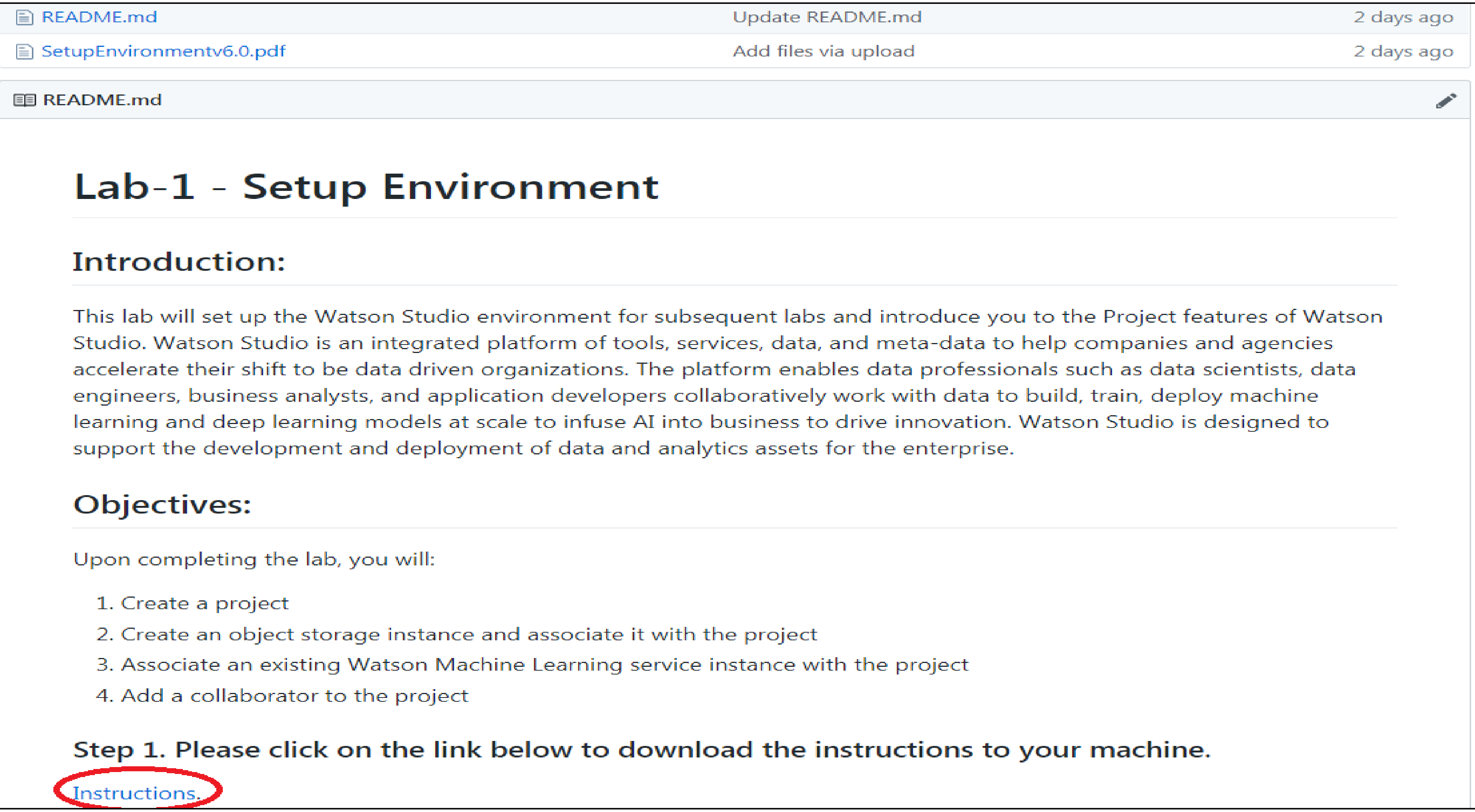
Description:

Work with IBM's Watson Studio in this workshop to build, train, and test machine learning/deep learning models. Participants will be led through the following nine hands-on labs. Note, the first lab is a prerequisite for the other labs. Once Lab-1 is completed, the other labs can be done in any order.

1. Lab-1 - This lab will set up the environment for the subsequent labs.
2. Lab-2 - This lab will use a Jupyter Notebook and the XGBoost library to apply machine learning to a classification problem in the healthcare profession. The Watson Machine Learning API will then be used to save and deploy the model.
3. Lab-3 - This lab will feature the Watson Studio Data Refinery to demonstrate data profiling, visualization, and data preparation.
4. Lab-4 - This lab will feature the Watson Studio SPSS modeler to demonstrate visual drag and drop creation of a machine learning model.
5. Lab-5 - 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 Titanic data set. The second part will deploy an application using the IBM Cloud DevOps toolchain that will invoke the deployed model to predict whether a passenger would have survived.
6. Lab-6 - This lab will feature Watson OpenScale. IBM Watson OpenScale is an open platform that helps remove barriers to enterprise-scale AI.
7. Lab-7 - This lab will feature the Watson Studio Neural Network modeler, and Experiment Assistant to build, train, and test a Convolutional Neural Network to classify images of handwritten digits.
8. Lab-8 - This lab will feature IBM's Adversarial Robustness Toolbox (ART). ART is a library dedicated to adversarial machine learning. Its purpose is to allow rapid crafting and analysis of attacks and defense methods for machine learning models. ART provides an implementation for many state-of-the-art methods for attacking and defending classifiers.

Github Repository

Lab-1 Readme



README.md	Update README.md	2 days ago
SetupEnvironmentv6.0.pdf	Add files via upload	2 days ago

README.md

Lab-1 - Setup Environment

Introduction:

This lab will set up the Watson Studio environment for subsequent labs and introduce you to the Project 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. Associate an existing Watson Machine Learning service instance with the project
4. Add a collaborator to the project

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

Instructions.

Lab Tips

- Watson Studio url: datascience.ibm.com
- Labs are in www.github.com/bleonardb3/ML_POT_10-30 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-1 make sure that you uncheck the "restrict who can be a collaborator" checkbox when creating the project.
- For Lab-7- use Firefox. Make sure you switch Region to us-geo.

Lab-1: Set up Environment

Introduction:

This lab will set up the Watson Studio environment for subsequent labs and introduce you to the Project 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
- Associate an existing Watson Machine Learning instance with the project
- Add a collaborator to the project

Lab-2: Build a Heart Disease Prediction Model

Introduction:

In this lab, you will use a Jupyter Notebook to train a model using the XGBoost library to classify whether a person has heart disease or not. In addition to training a model, the notebook also explains how to persist a trained model to the IBM Watson Machine Learning repository, and deploy the model as a REST service.

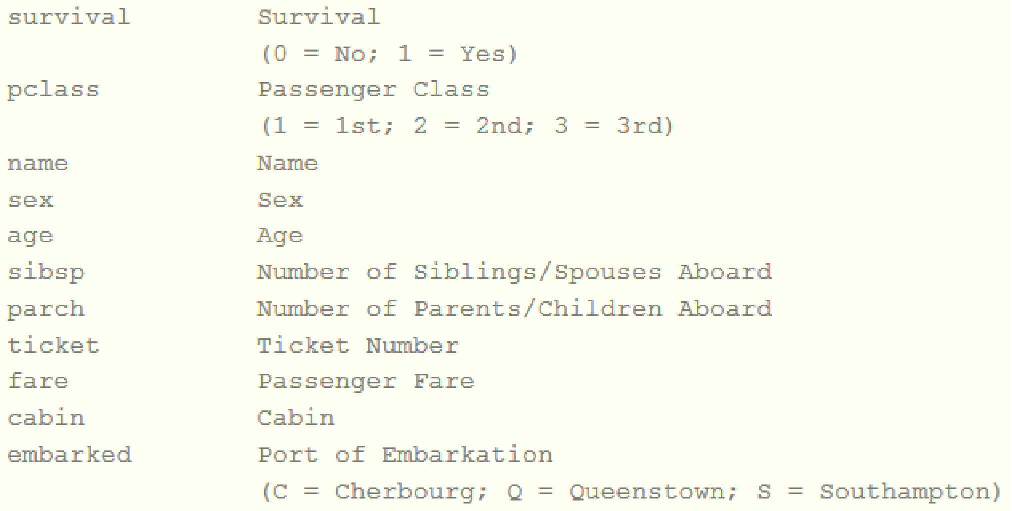
Objectives:

Upon completing the lab, you will know how to:

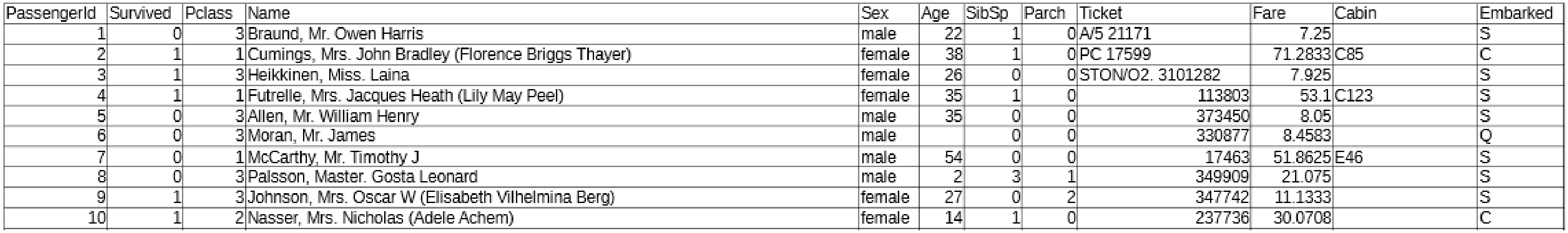
- Load a CSV file into Pandas DataFrame.
- Prepare data for training and evaluation.
- Create, train, and evaluate a XGBoost model.
- Visualize the importance of features that were used to train the model.
- Use cross validation to select optimal model hyperparameters based on a parameter grid
- Persist best model in Watson Machine Learning repository using Python client library.
- Deploy the model for online scoring using the Watson Machine Learning's REST APIs

Labs: 3,4,5 Titanic Data

- **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

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.

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 this lab, you will have:

- Become familiar with the Watson Studio SPSS Modeler capability
- Profiled the data set
- Explored the data set with visualizations
- Transformed the data
- Trained/Evaluated a machine learning mode.

Lab-5: AutoAI + DevOps

Introduction:

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. You will then deploy a web application that calls the Watson Machine Learning model.

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 survivability of a Titanic passenger.

Lab 6: Watson OpenScale

Introduction:

IBM Watson OpenScale is an open platform that helps remove barriers to enterprise-scale AI. In this lab you will configure Watson OpenScale to monitor quality, fairness,and drift and to provide the factors that explain a deployed model’s classification.

Objectives:

Upon completing the lab, you will

- Import and Deploy a machine learning model
- Provision a Watson OpenScale service
- Configure Watson OpenScale for Payload Logging, Quality, Fairness, and Drift.
- Submit Feedback and View Quality Metrics
- Score Data and View Fairness Metrics
- Explain a Transaction.

Lab-7: Recognizing Handwritten Digits

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

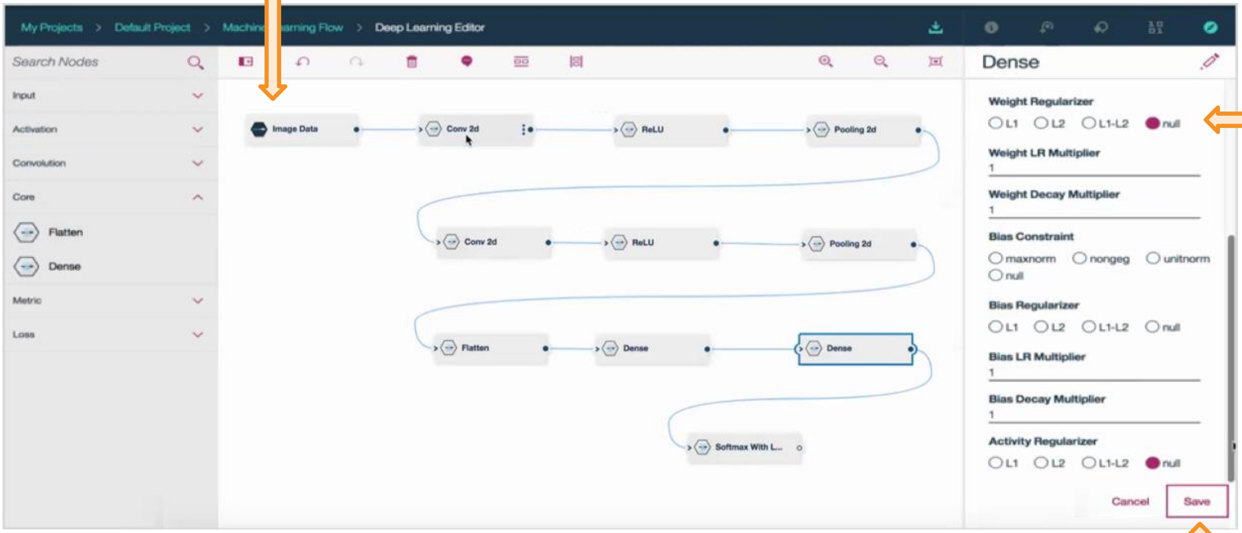
Neural Network Modeler

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

TensorFlow™ PYTORCH
Caffe Keras

Real-time validation of network flow



Dense

Weight Regularizer
○ L1 ○ L2 ○ L1-L2 ● null

Weight LR Multiplier
1

Weight Decay Multiplier
1

Bias Constraint
○ maxnorm ○ nonneg ○ unitnorm ○ null

Bias Regularizer
○ L1 ○ L2 ○ L1-L2 ○ null

Bias LR Multiplier
1

Bias Decay Multiplier
1

Activity Regularizer
○ L1 ○ L2 ○ L1-L2 ● null

Cancel Save

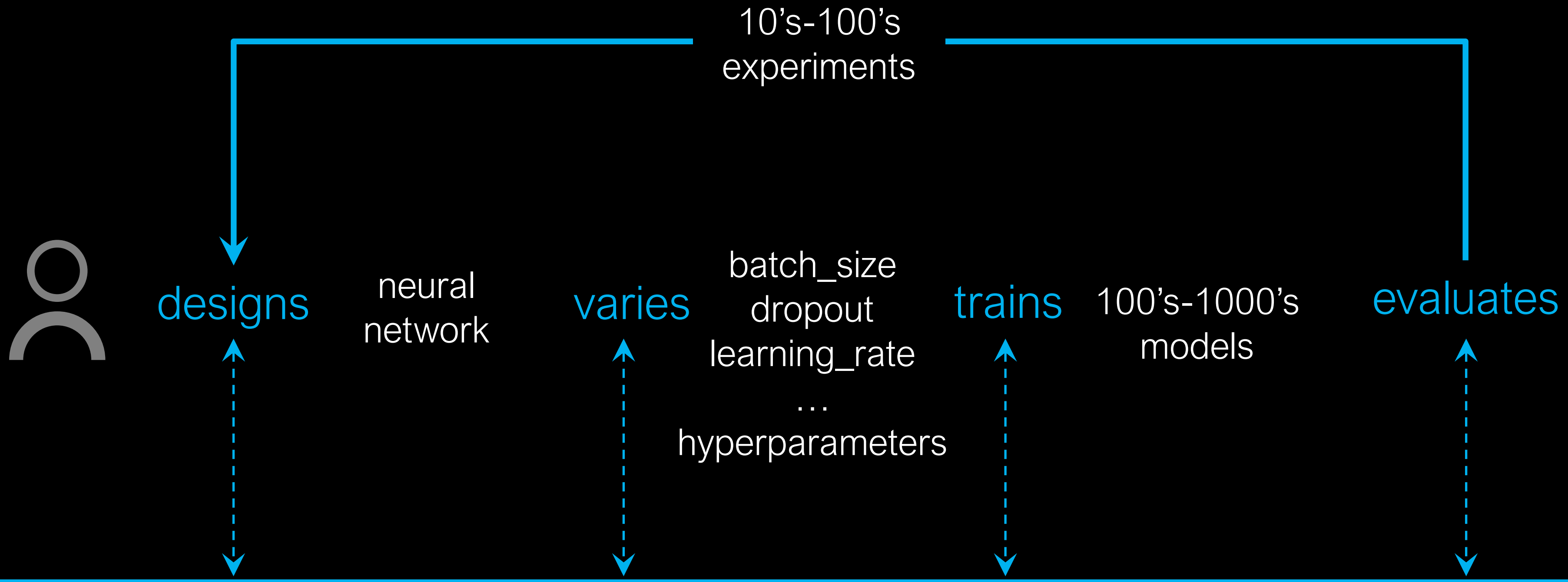
Drag-and-drop network layers

- 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

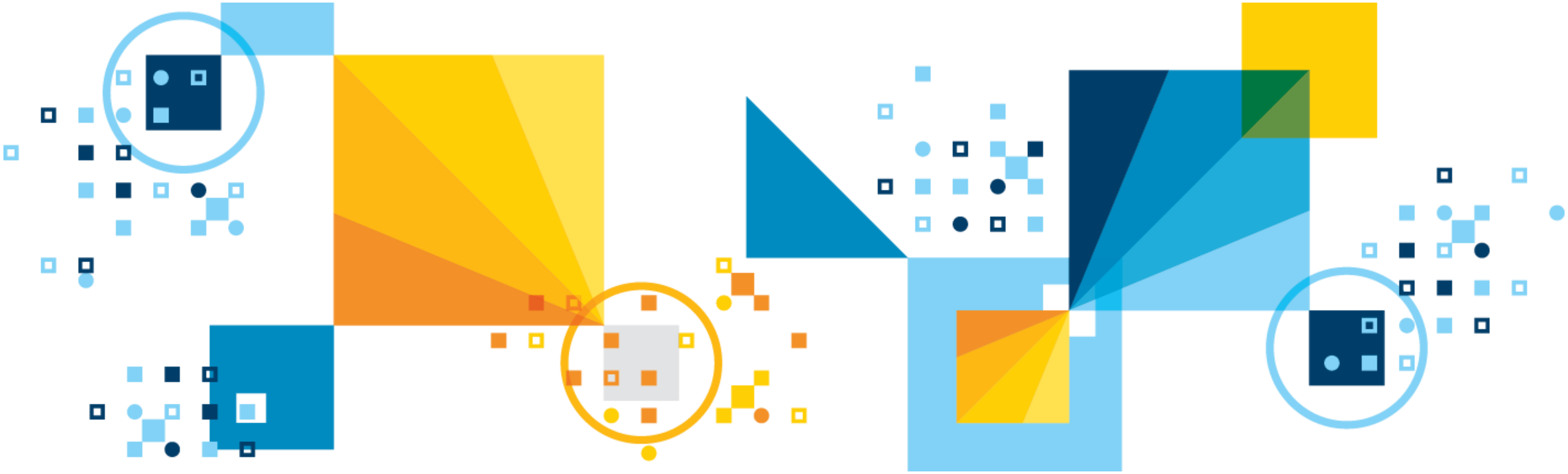
Experiment Builder



Experiment Builder
supports the end-to-end workflow

Introduction to Adversarial Robustness Toolbox

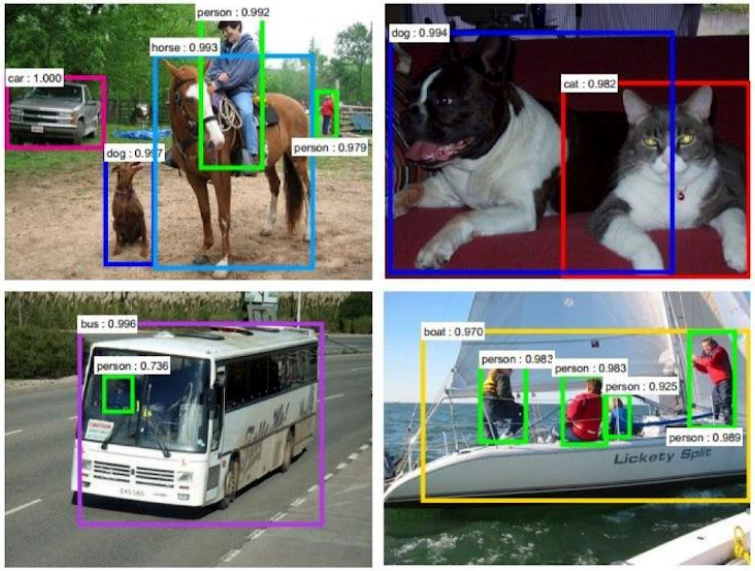
Zoya Yeprem



DNN in Visual Recognition

- **We use DNNs for lots of things:**
  - Facial recognition → iPhone FaceID
  - Text recognition → Mobile Check Deposit
  - Self driving cars → help detect signs, pedestrians, traffic lights, etc.

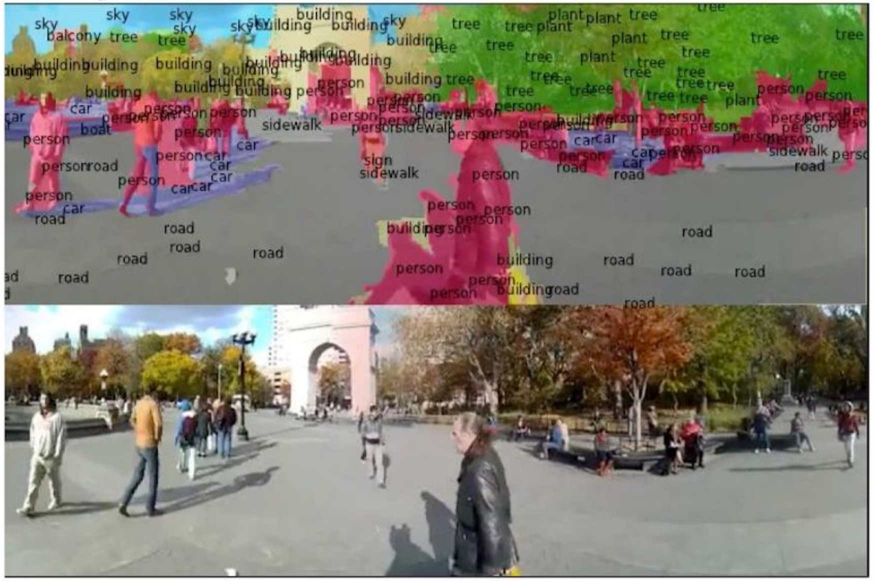
Detection



Figures copyright Shaoqing Ren, Kaiming He, Ross Girschick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



Figures copyright Clement Farabet, 2012. Reproduced with permission.

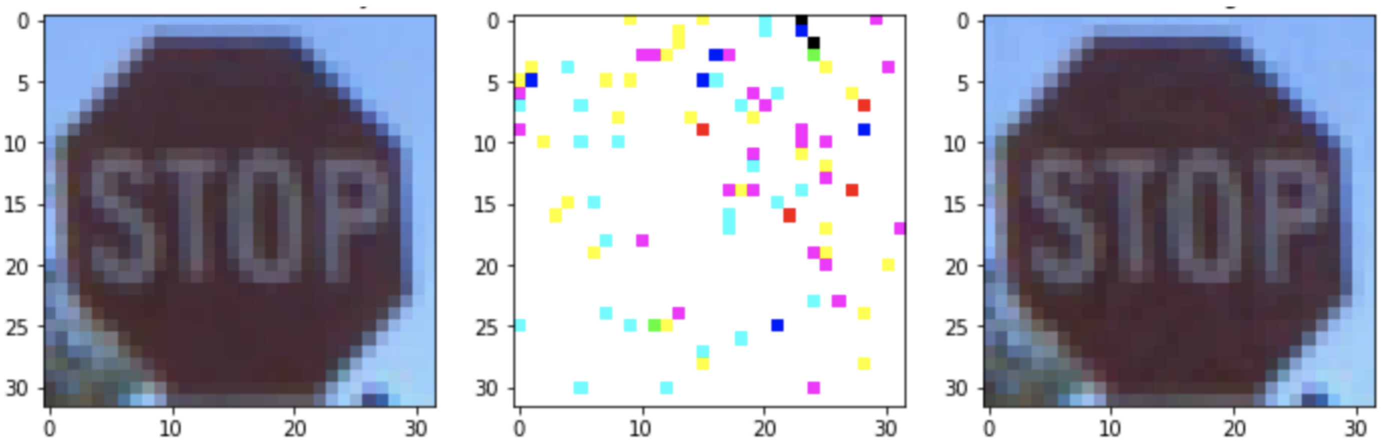
[Farabet et al., 2012]

What are Adversarial Images?

- Adversarial examples are inputs (say, images) which have deliberately been modified to produce a desired response by a DNN.

- Often, the target of adversarial examples is **misclassification** or a **specific incorrect prediction** which would benefit an attacker.

“Stop Sign“ + Noise = “Ahead Only” Sign



Adversarial Attacks

- Carefully crafted noise added to the input
- This noise gets amplified when ran through the layers of DNN
- The goal is to generate adversaries visually similar to original image

Stronger
BUT
More
Expensive

- FGSM
- Random + FGSM
- Projected Gradient Descent *
- DeepFool
- JSMA
- C&W

Threat Model

Black Box

- Attackers can only observe the outputs of a model. E.g. Attacking a model via an API
  - The adversary has no knowledge of the training algorithm or hyperparameters.

➢ Examples:
  - ➢ Boundary Attack
  - ➢ Substitute Blackbox Attack
  - ➢ Etc.

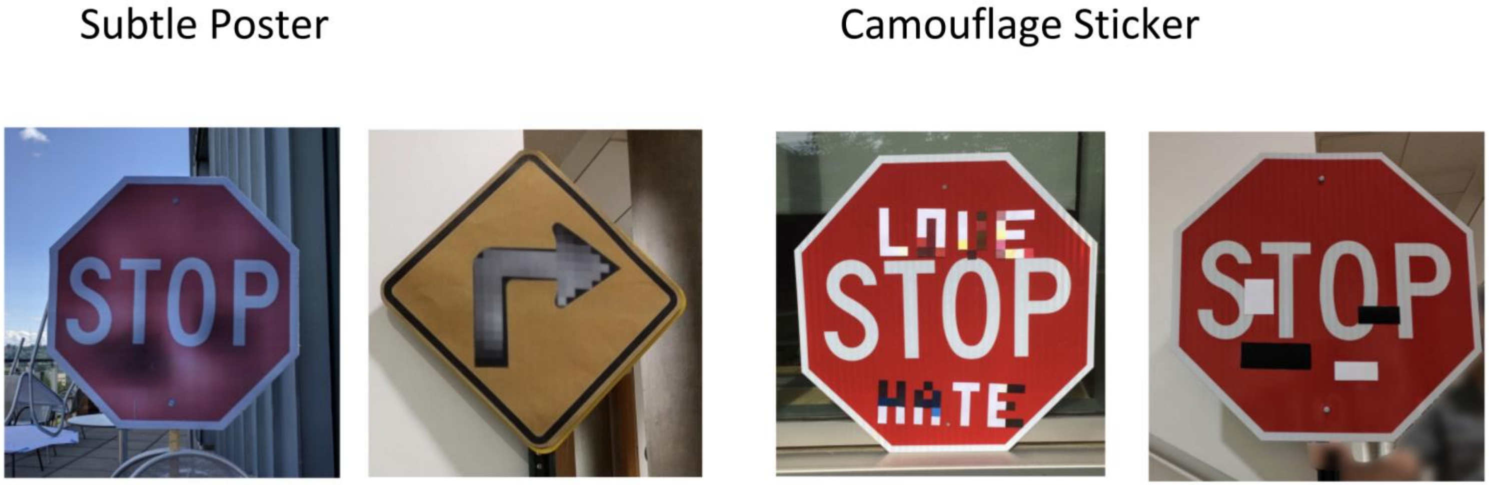
White Box

- attackers have complete access to the model that they want to attack.
- These are most effective attacks

➢ Examples:
  - ➢ Fast Gradient Sign Method (FGSM)
  - ➢ Random + FGSM
  - ➢ Projected Gradient Descent
  - ➢ Etc.

Why are they dangerous?

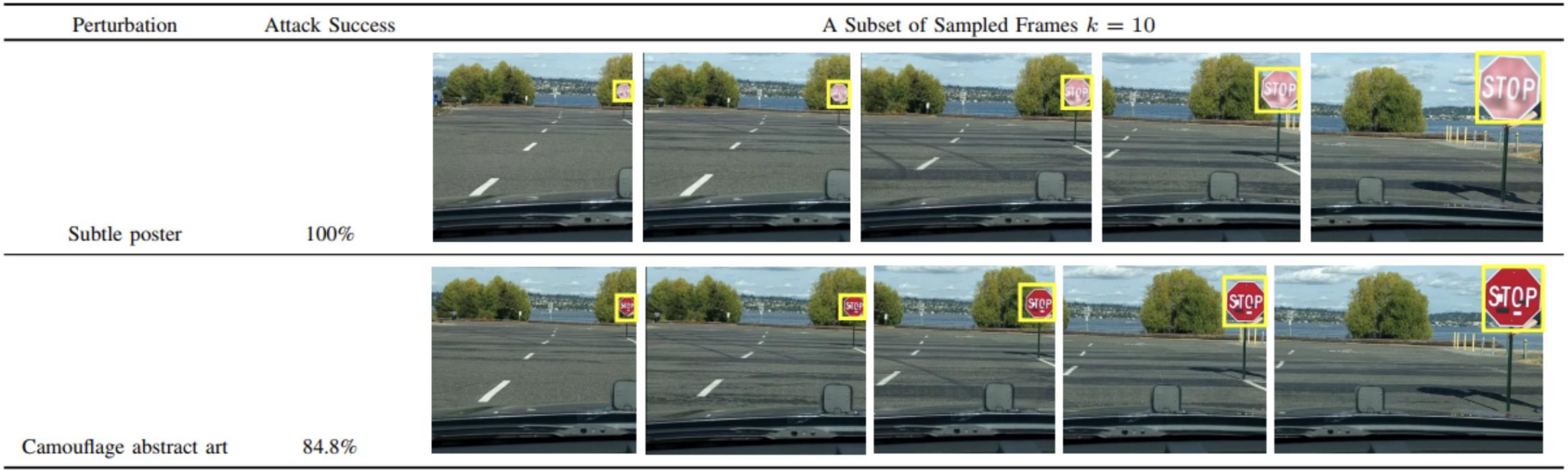
- **Can be crafted even if the attacker doesn’t have exact knowledge of the architecture of the DNN**
- **Adversarial attacks can be launched in the physical world**
  - adversaries could evade face recognition systems by wearing specially designed glasses
  - defeat visual recognition systems in autonomous vehicles by sticking patches to traffic signs



* Pictures from paper: Kevin Eykholt, et al. “Robust Physical-World Attacks on Deep Learning Visual Classification”

Are they effective?

- **Researchers proved that these attacks are successful! [1]**



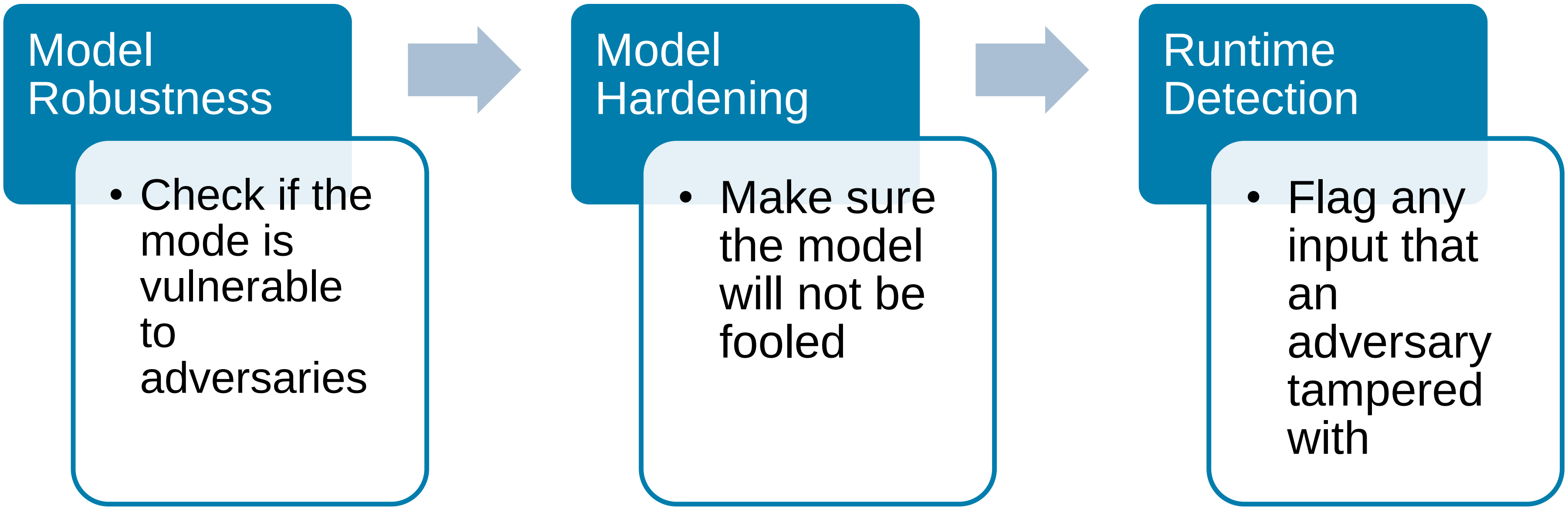
Perturbation	Attack Success	A Subset of Sampled Frames $k = 10$
Subtle poster	100%	
Camouflage abstract art	84.8%	

[1] Kevin Eykholt, et al. “Robust Physical-World Attacks on Deep Learning Visual Classification”

Adversarial Robustness Toolbox (ART)

- IBM Research team in Ireland developed the toolkit to help defend DNNs against adversarial attacks
- Open-source software library
- Written in python
- Supports most deep learning frameworks : TensorFlow, Keras, PyTorch, etc.
- It creates adversarial examples AND provides methods for defending DNNs against those.

How can ART help?



Model Robustness

- Check if the mode is vulnerable to adversaries

Model Hardening

- Make sure the model will not be fooled

Runtime Detection

- Flag any input that an adversary tampered with

Conclusions

- **Adversarial attacks are real threats**
  - Self-driving cars
  - Healthcare
  - Financial institutions
  - Insurance companies
  - ...
- **It's important to**
  - Realize there are vulnerabilities
  - Have means to protect ourselves

Lab 8: ART in Action

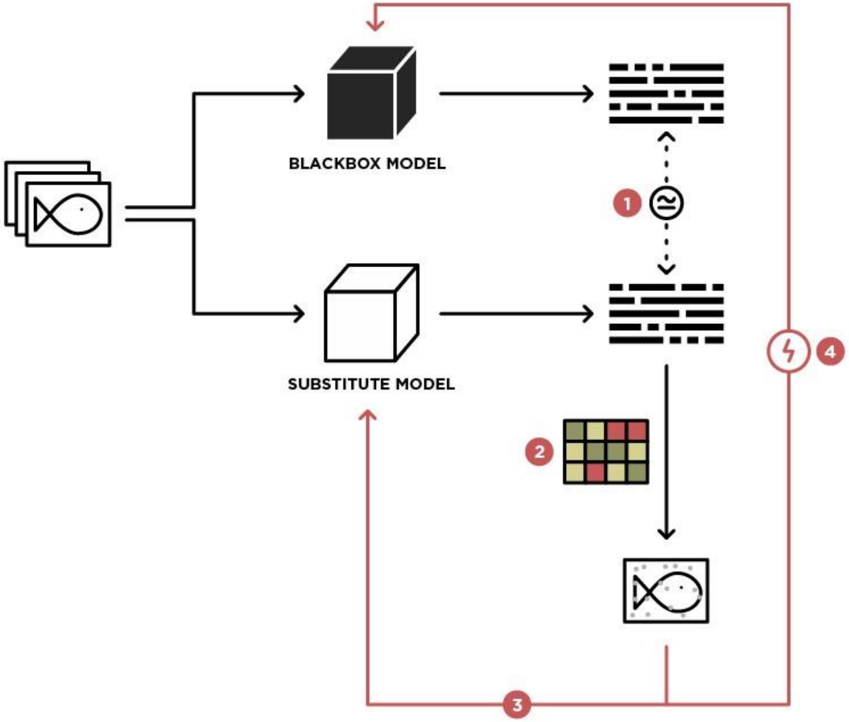
- **Create a Notebook in Watson Studio**
- **Upload the Lab-8 Notebook file using provided URL**
- **Run through cells**

- **Overview**
  - Load a Tensorflow trained model
  - Create an ART classifier object using the loaded model
  - Perform an adversarial attack
  - Perform a defense to make sure manipulated images can still be classified correctly

Bonus Slides

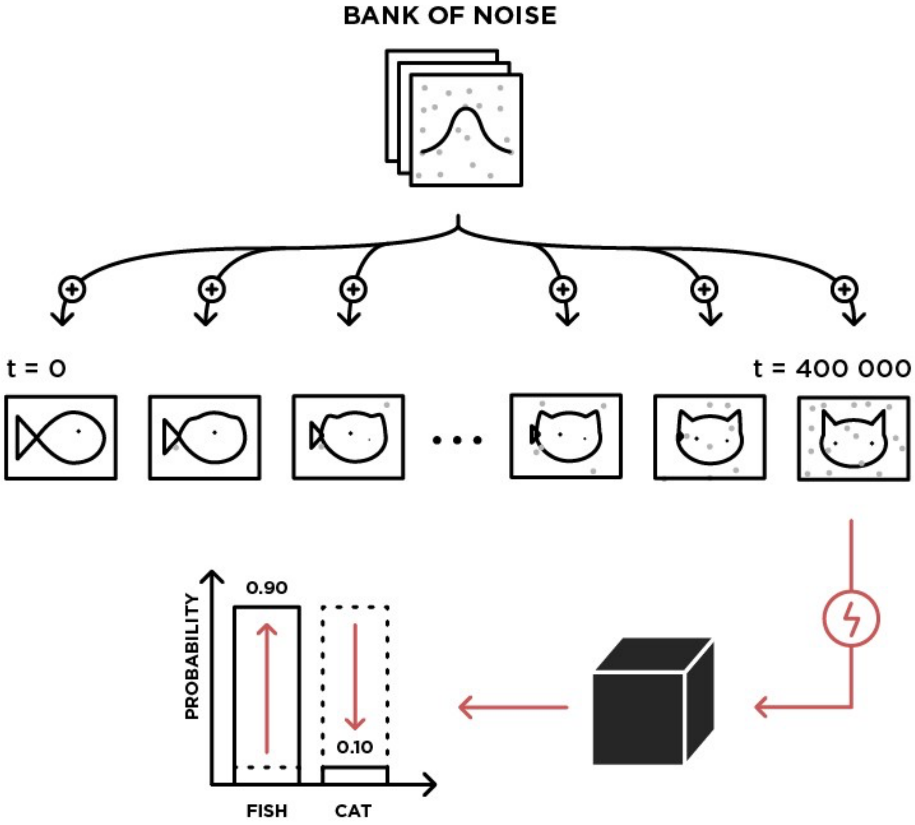
Substitute Blackbox Attack

- **Approximate the decision boundary of the Blackbox model that we want to attack**.
  - Train a **substitute model** on a synthetic dataset that is similar to the dataset that the Blackbox model is trained on.
  - The trick here is that the **label for the synthetic dataset should come from the Blackbox model's prediction**.



Boundary Attack

1. Evaluates a sequence of perturbed images through the model
   - In **non-targeted attack**, the starting image can be sampled from uniform noise.
   - In **targeted attack**, the starting image is an example from the target misclassification class.
2. The method **modifies the image iteratively** to look more like an example from another class while continuing to preserve its adversarial nature


BANK OF NOISE
t = 0
t = 400 000
PROBABILITY
0.90
0.10
FISH
CAT