

Data Science Workshop

Watson Machine Learning

Toulon, February 2019

Emmanuel Génard – genard@fr.ibm.com
Data Scientist & Cloud Developer Advocate Europe,
IBM Business Solution Center Nice, France

(Recap) Why use Machine Learning?

The ultimate goal of a machine learning system is to make a **prediction**.

Here are some examples you may be familiar with:

- Predict whether an image is a cat or dog
- Predict the value of a home
- Predict which products to recommend to a user
- Predict which users share the same interests
- Predict when to turn, accelerate, or apply the brakes in a self-driving car

What is Watson Machine Learning?

IBM Watson Machine Learning is a full-service IBM Cloud offering that makes it easy for **developers** and **data scientists** to work together to integrate predictive capabilities with their applications.

- The Machine Learning service is a set of **REST APIs** that you can call from any programming language to develop applications that make smarter decisions, solve tough problems, and improve user outcomes.



What is Watson Machine Learning?

IBM Watson™ Machine Learning enables users to perform two fundamental operations of machine learning: training and scoring.

1. **Training** is the process of refining an algorithm so that it can learn from a data set. The output of this operation is called a model. A model encompasses the learned coefficients of mathematical expressions.
2. **Scoring** is the operation of predicting an outcome by using a trained model. The output of the scoring operation is another data set containing predicted values.



Watson Machine Learning Deployment

Machine Learning features

Take advantage of machine learning **models management** (continuous learning system) and **deployment** (online, batch, streaming).

Select any of widely supported machine learning frameworks: TensorFlow, Keras, Caffe, PyTorch, Spark MLlib, scikit learn, xgboost and SPSS.



IBM Watson Studio

My Projects / ISEN

Launch IDE Add to project

Watson Machine Learning models

New Watson Machine Learning model

NAME	STATUS	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
Loan_Status	trained	spss-modeler-18.1	spss-modeler-18.1	18 Feb 2019	⋮
StockPredict	trained	wml-1.1	spark-2.1	31 May 2018	⋮
House Pricing First Run	trained	wml-1.1	spark-2.1	17 May 2018	⋮

Integration with Watson Studio

Create and train machine learning models with the best tools and the latest expertise in a social environment built by and for data scientists.

Wide choice of interfaces

Use the command line interface and Python client to manage your artifacts. Extend your application with artificial intelligence through the Watson Machine Learning REST API.

Code Snippets

cURL Java JavaScript Python Scala

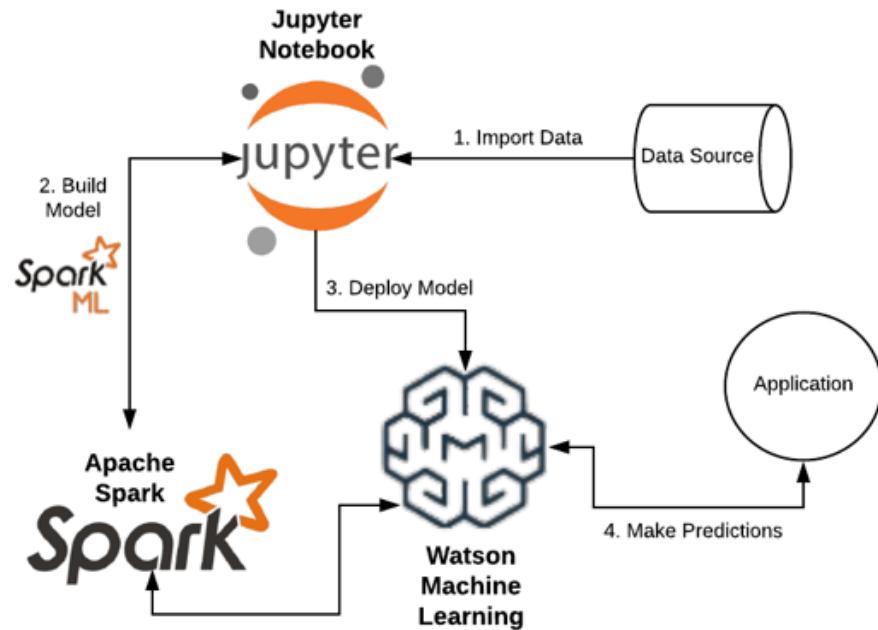
```
const XMLHttpRequest = require("xmlhttprequest").XMLHttpRequest;
const btoa = require("btoa");
const wml_credentials = new Map();

// NOTE: you must manually construct wml_credentials hash map below using information retrieved
// from your IBM Cloud Watson Machine Learning Service instance

wml_credentials.set("url", wml_service_credentials_url);
wml_credentials.set("username", wml_service_credentials_username);
wml_credentials.set("password", wml_service_credentials_password);
```

Watson Machine Learning Workflow

1. **Create** a Jupyter Notebook and import, clean, and analyze the data.
2. Use Apache Spark ML to **build and test** a machine learning model.
3. **Deploy** the model to Watson ML.
4. **Call** the Watson ML **scoring endpoint** (REST API) to make predictions from a client application or backend service.



Watson Machine Learning Model Builder

New Watson Machine Learning model 

Select model type

Model builder From file From sample

Select runtime

Only Spark environments supporting Scala kernels can be used for model builder creation.

Default Spark Scala 2.11

The selected runtime uses one driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM. This runtime consumes 1.5 capacity units per hour.

 Your Spark runtime will be automatically stopped when you save your model, or after 3 hours of inactivity. To avoid consuming extra capacity unit hours delete your model builder instance or [stop your runtime](#) when you are finished with it.

[Learn more](#) about capacity unit hours and Watson Studio pricing plans.

Automatic

Prepare my data and create a model automatically

Manual

Let me prepare my data and select which models to train

1. **Automatically** build a prediction model
2. **Select** the training data set
3. **Select** the target
4. **Select** the predictors
5. Model Builder **suggests** the best algorithms
6. **Configure** the estimators
7. **Train** several models at once
8. **Compare** the results
9. **Deploy**

Select a technique

Column value to predict (Label Col)

CHURN (String)

Feature columns

All (default)

 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



Scoring of deployed models, and continuous learning

IBM Watson Projects Tools Community Services US South ⌂ ⌃ ⌚ TH

My Projects / Demo Script / NYC Learning Multiple Attr ⌂ ⌃ ⌚ ⌚ ⌚

NYC Learning Multiple Attr

Overview Evaluation Deployments

Evaluation Events

+ Add feedback data + New evaluation

accuracy

The chart displays accuracy values over time. A red horizontal line represents a threshold at 0.6. A blue line shows the accuracy fluctuating between approximately 0.3 and 0.6. A tooltip provides specific details for a data point at 10/13/17 08:50.

accuracy: 0.571
threshold: 0.6
phase: training
evaluation time: 10/13/17 08:50

Evaluation time

Last Evaluation Result

Version	90697ee3-99cc-45b5-91f1-a54af74eb075
Phase	monitoring

8 Watson / Presentation Title / Date

Scoring of deployed models

Model Testing

KaggleHousePrice

Overview Implementation **Test**

Enter input data

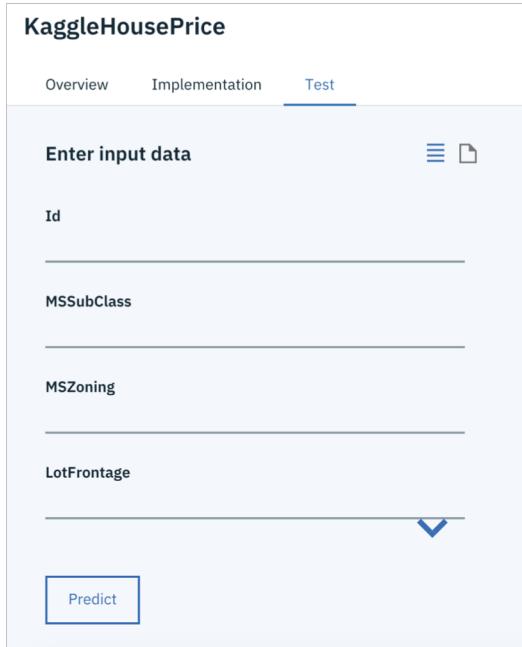
Id

MSSubClass

MSZoning

LotFrontage

Predict



Batch scoring

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bearer $WML_AUTH_TOKEN' -d '{"fields": ["ID", "Gender", "Status", "Children", "Est Income", "Car Owner", "Age", "LongDistance", "International", "Local", "Dropped", "Paymethod", "LocalBilltype", "LongDistanceBilltype", "Usage", "RatePlan"], "values": [$ARRAY_OF_VALUES_TO_BE_SCORED, $ANOTHER_ARRAY_OF_VALUES_TO_BE_SCORED]} }' <scoring endpoint>
```

Python SDK available

Watson Developer Cloud Python SDK

build passing slack 123/9472 pypi v2.8.0 CLAs signed 49

Python client library to quickly get started with the various [Watson APIs](#) services.

► [Table of Contents](#)

Before you begin

- You need an [IBM Cloud](#) account.

Installation

To install, use `pip` or `easy_install`:

```
pip install --upgrade watson-developer-cloud
```

or

```
easy_install --upgrade watson-developer-cloud
```

Thank You



Useful Links & Resources

External

Getting Started:

[Service Homepage](#)
[Feature Requests / Suggestions](#)

Case Studies:

[OmniEarth](#)
[Aerialtronics](#)
[BlueChasm](#)
[iTrend](#)

Tutorials & Best Practices:

[Training models with Watson Studio](#)
[Getting started with Watson + Core ML](#)
[Stacking Multiple Custom Models](#)
[Create a Calorie Counting App](#)
[Watson Visual Recognition & Twilio](#)
[Best Practices for Custom Models](#)

Code Patterns:

[Classify vehicle damage](#)
[Analyze industrial equipment for defects](#)
[Create an Android calorie-counter app](#)

External continued

Books:

[Redbook: Building Cognitive Application using IBM Watson Services vol3 – Watson Visual Recognition](#)

Blogs:

[IBM Watson on Medium](#)

Internal

[Slack Channel: #ibmvisual-recognition](#)
[Service Roadmap](#)
[IBMer key limit increase request form](#)
[ZACS portal](#)
[Digital Sales Play](#)
[Content Request & Feedback Form](#)