

Building an audio recognition

API

@chugths

Building an audio recognition API

- We will show you :
 - How we created a machine learning model to recognize bird songs;
 - How we exposed both the data preparation and predict steps as cloud based APIs;
 - How we used Node-RED to combine the audio recognition with Watson Visual Recognition to build a combined audio / visual bird recognition application.

Agenda

- Introduction
- Bird Song App
- The journey
- Putting it all together
- Education Resources



chugths top 6% this quarter

MQ Developer, Developer Experience Advocate and Hackathon mentor, with many years of programming experience including Assembler, C/C++, Java, Python, Go and Node.js.

1,814 REPUTATION

• 1 • 6 • 18

chugths

top 6% this quarter

MQ Developer, Developer Experience Advocate and Hackathon mentor, with many years of programming experience including Assembler, C/C++, Java, Python, Go and Node.js.

188

6

~19k

answers

questions

people reached



Hursley, United Kingdom



chugths



chugths



Member for 4 years, 3 months



244 profile views



Last seen 20 secs ago



Visited 742 days, 5 consecutive

By Day

<https://github.com/ibm-messaging/mq-dev-patterns>

A screenshot of a GitHub repository page. At the top, it shows the repository name 'ibm-messaging / mq-dev-patterns' with 7 stars, 26 forks, and 27 open issues. Below the header, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A note below the header states 'No description, website, or topics provided.' and has a 'Edit' button. The main content area shows a list of files and their details. It includes a branch dropdown set to 'master', a 'New pull request' button, and buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The files listed are: .gitignore, LICENSE, README.md, cddt.json, and env.json. Each file has a small icon, a name, a description, and a timestamp indicating when it was last modified.

<https://developer.ibm.com/messaging/learn-mq/mq-tutorials/>

MQ tutorials, taking you further

Every great achievement starts with a single step.
Here's a set of guided tutorials that provides you with the tools to master MQ.



Search by:

Skill level
 Any Skill Level
 Beginner
 Intermediate
 Advanced

Language

.NET
 C#
 Go
 Java
 Node.js
 Python

Search

App side auto-reconnect and CCDTs
A deep dive into connecting an application to multiple queue managers.

Securing IBM MQ with TLS
Developing with the IBM MQ MacOS Toolkit

Point to point in C# for .NET Framework
Write a C# application for .NET Framework that uses IBM MQ as a messaging provider. See how to use IBM MQ classes for XMS to put and get messages to and from a queue.

60 Minutes Start tutorial
30 Minutes Start tutorial

The Quarkus landing page features a dark blue background with a starry, nebula-like pattern. At the top right, there are links for 'GET STARTED', 'GUIDES', 'COMMUNITY', 'BLOG', and a prominent 'START CODING' button. The central text reads 'SUPersonic Subatomic Java' in large, bold, white letters. Below it, a subtitle says 'A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.' At the bottom, there's a blue call-to-action button labeled 'GET STARTED WITH QUARKUS'.

The GraalVM landing page has a dark teal background. At the top, there are links for 'Home', 'Docs', 'Downloads', 'Community', and a star count of '11,849'. The main title 'GraalVM™' is in large, bold, white letters. Below it, the tagline 'Run Programs Faster Anywhere' is displayed. To the right, there's a diagram showing a central 'GraalVM' box connected to various programming language icons like Java, Python, C/C++, and .NET. Below the diagram, there are two buttons: 'WHY GRAALVM' and 'GET STARTED'.

By Night

- <https://github.com/ibm-early-programs>
- <https://github.com/watson-developer-cloud/node-red-node-watson>

[README.md](#)

Node-RED Watson Nodes for IBM Cloud


New in version 0.9.2

- Assistant V2 - Fix bug session expiry bug.

New in version 0.9.1

- Assistant V2 - Allow flow to assign a string session id. The node maps this user specified session id to the real session id. Additional param option allow session id to be reset.

New in version 0.9.0

- Node-RED & IBM-Watson & Use of promises on API invocation & IAM URL construct migration & Removal of default endpoint of
 - Assistant V1
 - Assistant V2
- All Nodes now require Node-RED 1.0.x or above
- Remove watson-developer-cloud dependency
- Remove code for redundant nodes

Watson Nodes for Node-RED

A collection of nodes to interact with the IBM Watson services in [IBM Cloud](#).

 ibm-early-programs

[Repositories 20](#) [Packages](#) [People 12](#) [Teams 1](#) [Projects](#) [Settings](#)

Find a repository... Type: All Language: All Customize pins [New](#)

animal-sounds
Animal Sounds Machine Learning
JavaScript Apache-2.0 22 ⚡ 1 ① 2 0 Updated 15 days ago

node-red-contrib-watson-machine-learning
Writer Node for Facebook Messenger
HTML Apache-2.0 7 ⚡ 2 ① 3 0 Updated 23 days ago

node-red-contrib-facebook-messenger-writer
Writer Node for Facebook Messenger
HTML Apache-2.0 6 ⚡ 1 ① 0 0 Updated 29 days ago

node-red-contrib-pdf-hummus
JavaScript Apache-2.0 2 ⚡ 1 ① 0 0 Updated on 17 Jul 2019

node-red-contrib-browser-utils
Browser utilities for node-red
JavaScript 13 ⚡ 9 ① 0 0 Updated on 11 May 2019

watson-tv
Forked from annaet/watson-tv
CSS 15 ⚡ 3 ① 0 0 Updated on 18 Apr 2019

node-red-contrib-media-utils
A collection of Node-RED media nodes that can be used with IBM Watson services
JavaScript 6 ⚡ 5 ① 3 0 Updated on 21 Mar 2019

Top languages
JavaScript HTML CSS

People 12 >

[Invite someone](#)

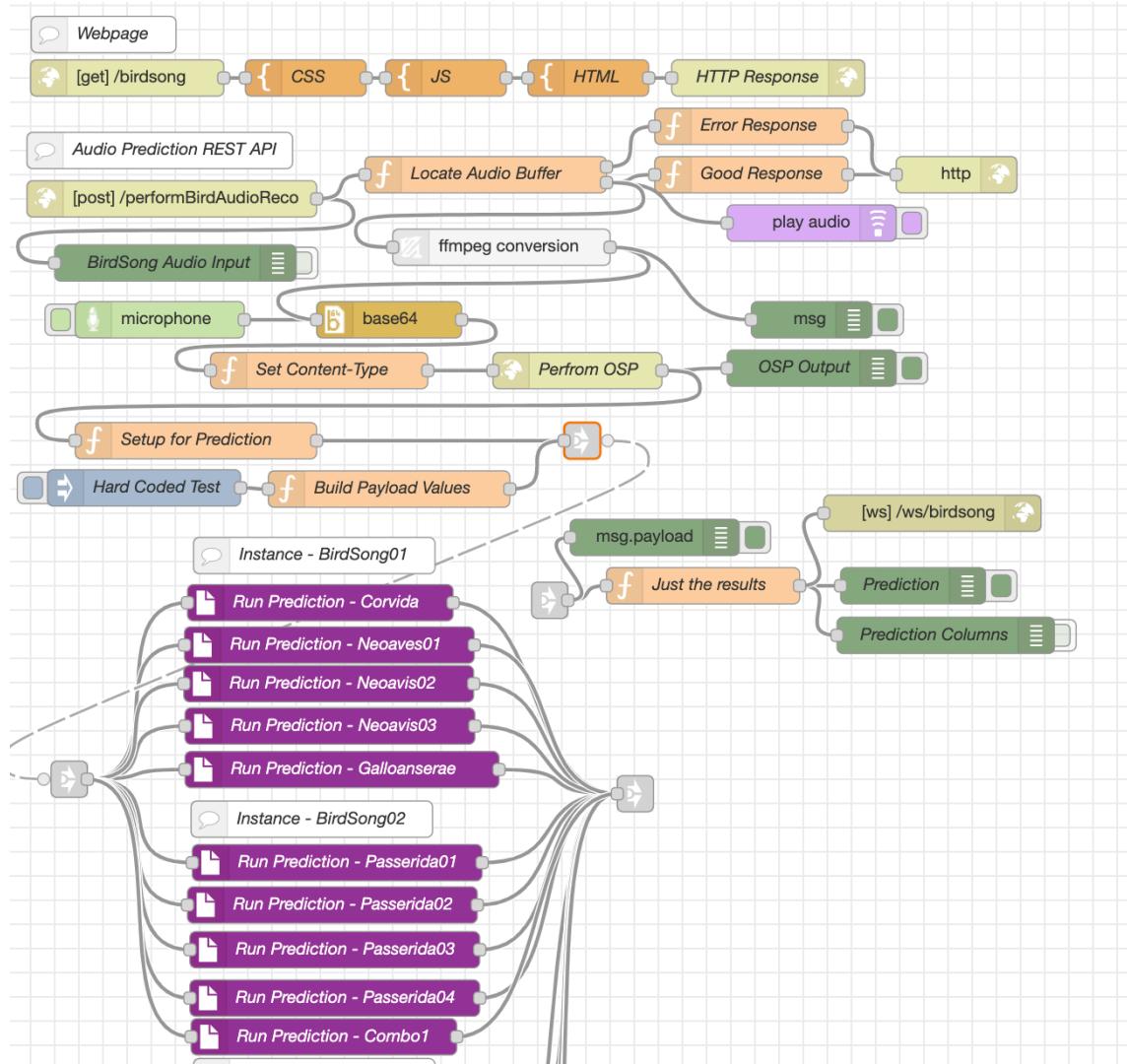
Node-RED

- <https://nodered.org>



The image shows the Node-RED runtime interface running on a Mac OS X desktop. The window title is "Node-RED Introduction". Inside, there's a flow diagram consisting of several nodes: an "http request" node, an "if/then" node, a "sentiment" node, and an "http response" node. The "if/then" node has two outputs: one leading to the "sentiment" node and another leading to the "http response" node. The "sentiment" node has an output leading to the "http response" node. On the left, a sidebar lists various node categories such as "http", "script", "math", "function", and "misc". On the right, there are several terminal-like panes showing log messages and configuration details. A play button icon is visible in the bottom right corner of the interface.

Bird-Song App



MACHINE LEARNING

Bird Type?

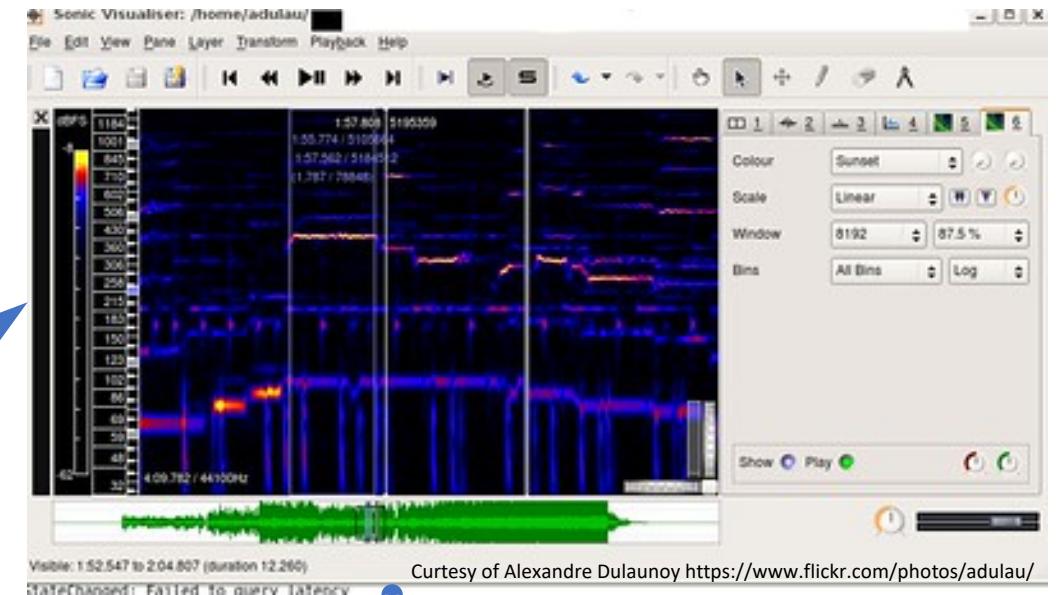
Select Audio File: Choose file BirdEurasianSkylark.wav Process Audio

House Sparrow	0.26025491078519614
Eurasian Blackcap	0.15588413459509004
Long-tailed Tit	0.11814913729487264
Common House Martin	0.10639800042789031
Eurasian Jay	0.5633333333333332
Rook	0.15151667200455501
Eurasian Treecreeper	0.22817530140110787
Common Linnet	0.21969696969696967
Grey Partridge	0.2971264229094923
Common Pheasant	0.25969822222878813
Black-headed Gull	0.15109126984126983
Eurasian Oystercatcher	0.14166666666666664
Grey Plover	0.1709510593146024
Wood Sandpiper	0.12616279069767442
Marsh Warbler	0.15316559615864306
Wood Warbler	0.1450336222398751
Dunlin	0.1470686040183597
Common Swift	0.2706315535327163
Northern Lapwing	0.18106026761840716

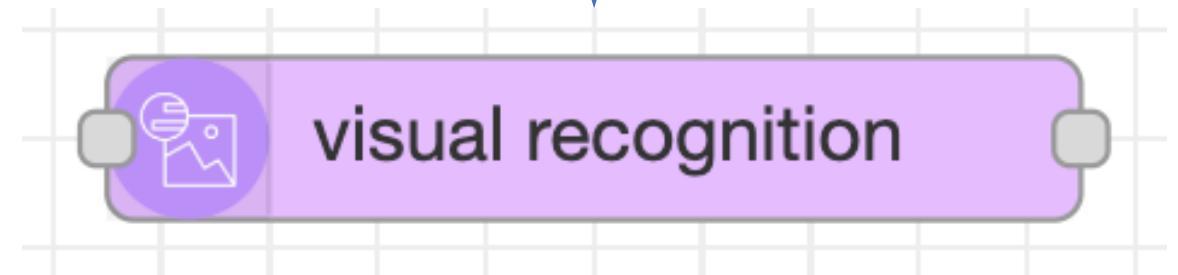
Why Audio



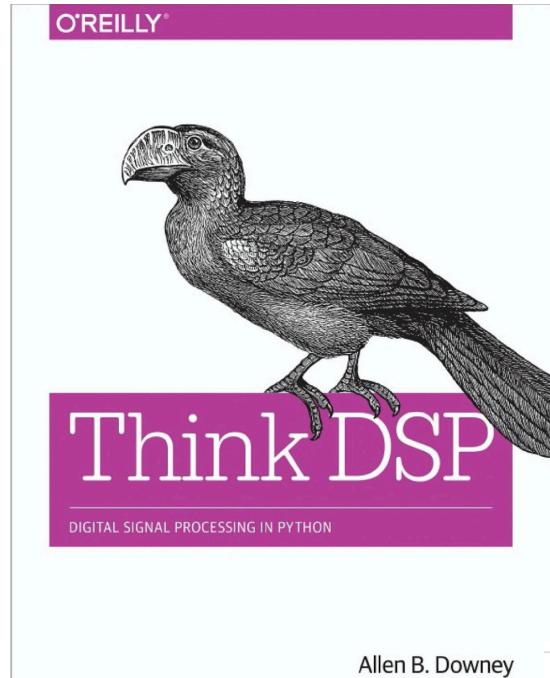
Courtesy of U.S. Department of Agriculture. <https://www.flickr.com/photos/usdagov/>



Courtesy of Alexandre Dulaunoy <https://www.flickr.com/photos/adulau/>



Digital Signal Processing



1	COLUMN1	COLUMN2	COLUMN3	COLUMN4	COLUMN5	COLUMN6	COLUMN7	COLUMN8	COLUMN9	COL
2	Common Redpoll	-93.5625537	158.1481422	-4.227614232	-9.234682697	3.402170101	-2.300618687	-43.1603349	13.49714183	53.5
3	Common Redpoll	-278.6487172	-41.90617055	5.877705015	6.392366763	6.608470871	31.32483828	6.421149731	19.57858368	-35.8
4	Common Redpoll	-0.010552397	-0.003655855	-0.043379321	-0.050190141	-0.000309817	-0.064935167	-0.040833887	-0.016352151	-0.03
5	Other	0.081999291	-0.379093308	-0.486601318	-0.416059985	0.310006568	0.882266991	-1.872927641	1.740553453	-2.06
6	Other	-5.787676968	-0.363697005	2.336301632	1.987178946	0.620883701	-0.053806095	-0.168789502	0.644747409	-2.41
-	-	-	-	-	-	-	-	-	-	-

Data Source

- <https://www.kaggle.com/rtatman/british-birdsong-dataset>

The screenshot shows a Kaggle dataset page for the "British Birdsong Dataset". The title "British Birdsong Dataset" is displayed prominently, along with the subtitle "264 recordings from 88 species". A profile picture of Rachael Tatman and a collaborator is shown, indicating the dataset was updated 2 years ago (Version 2). The main navigation tabs include Data, Tasks, Kernels (7), Discussion (2), Activity, and Metadata. A large download button offers "Download (636 MB)" and a "New Notebook" button. Below the main header, there are sections for Usability (7.6), License (Other (specified in description)), and Tags (music, animals, nature, ecology, acoustics). The "Description" section includes a "Context" paragraph about bird song usage and a "Content" paragraph detailing the dataset's origin from Xeno Canto and its balanced nature across 88 species.

Dataset

British Birdsong Dataset
264 recordings from 88 species

Rachael Tatman and 1 collaborator • updated 2 years ago (Version 2)

Data Tasks Kernels (7) Discussion (2) Activity Metadata Download (636 MB) New Notebook

Usability 7.6 License Other (specified in description) Tags music, animals, nature, ecology, acoustics

Description

Context:

Birds use songs and calls of varying length and complexity to attract mates, warn of nearby danger and mark their territory. This dataset contains a recordings of different birdsongs from bird species that can be found in Britain (although the recordings themselves are from many different locations).

Content:

This is a dataset of bird sound recordings, a specific subset gathered from the Xeno Canto collection to form a balanced dataset across 88 species commonly heard in the United Kingdom. It was originally compiled by Dan Stowell and shared on

Machine Learning

Welcome Soheel!

Watson Studio • Watson Knowledge Catalog

Start by creating a project

A project is how you organize your resources to work with data and collaborate with team members

Create a project
Create a project, then add the tools and assets you need.

Search a catalog
Find the assets you need in a catalog.

Prediction settings

PREDICTION COLUMN: COLUMN1 COLUMNS DATA TYPE: String DATA SOURCE: NormalizedNeoaves1.csv

Prediction type
Change the prediction type based on data in the prediction column. Changing the type changes other prediction settings.

Binary classification Classify data into categories. Choose this if your prediction column contains two distinct categories.	Multiclass classification Classify data into categories. Choose this if your prediction column contains multiple distinct categories.	Regression Predict values from a continuous set of values. Choose this if your prediction column contains a large number of values.
---	--	--

Optimized metric
Choose the metric to optimize for the experiment.

Accuracy (Recommended)	<input checked="" type="radio"/>
F1 Micro	<input type="radio"/>
F1 Macro	<input type="radio"/>

Experiment summary Pipeline comparison Rank by: Accuracy (Optimized) Score: Cross validation Holdout

Relationship map Prediction column: COLUMN1

FEATURE TRANSFORMERS PIPELINES TOP ALGORITHMS NormalizedNeoaves...

Progress map Swap view

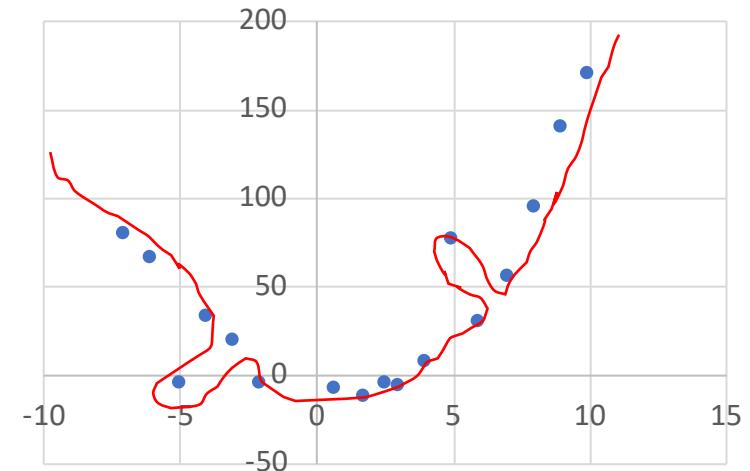
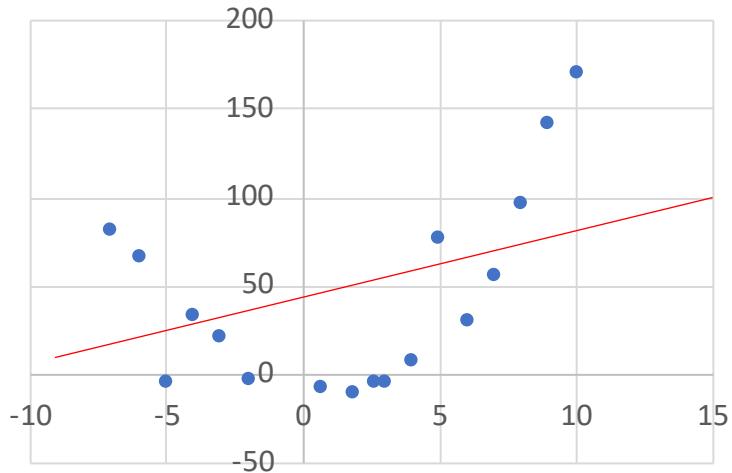
Experiment complete ✓
4 PIPELINES GENERATED
4 pipelines generated from algorithm. See pipeline leaderboard below for details.
Time elapsed: 94 minutes

View full log

Pipeline leaderboard

Rank	Name	Algorithm	Accuracy (Optimized)	Enhancements	Build time
1	Pipeline 4	LGBM classifier	0.723	HPO-1 FE HPO-2	00:28:26
2	Pipeline 3	LGBM classifier	0.715	HPO-1 FE	00:47:00
3	Pipeline 2	LGBM classifier	0.678	HPO-1	00:13:42
4	Pipeline 1	LGBM classifier	0.646	None	00:01:01

Data Science

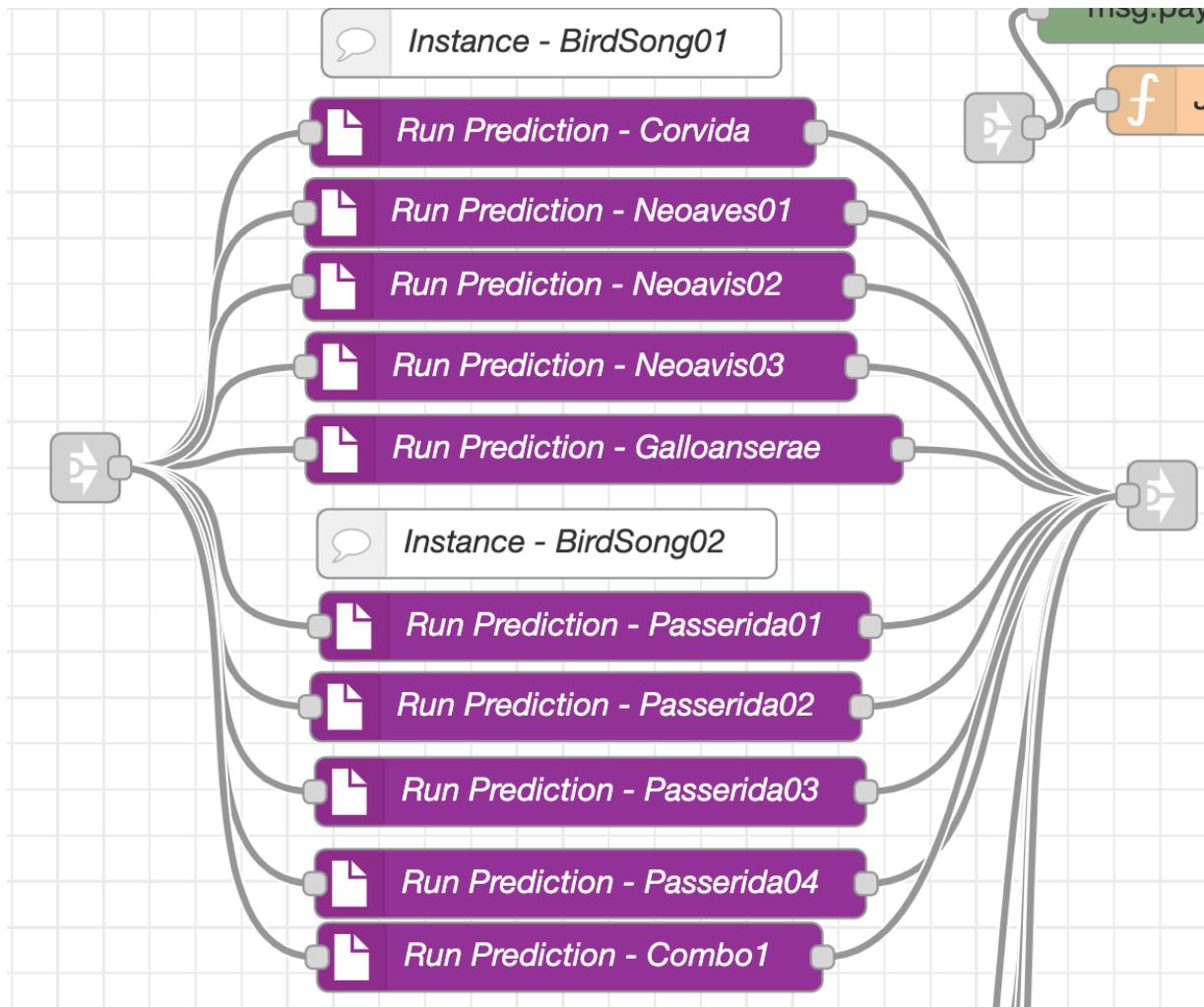


Min-Max
Feature
scaling

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Feature scaling is used to bring all values into the range [0,1]. This is also called unity-based normalization. This can be generalized to restrict the range of values in the dataset between any arbitrary points a and b , using for example $X' = a + \frac{(X - X_{\min})(b - a)}{X_{\max} - X_{\min}}$.

Why 9 Models?



API

- <https://watson-ml-v3-api.mybluemix.net>
- <https://watson-ml-v4-api.mybluemix.net>

Overview **Implementation** Test

Implementation

[View API Specification](#)

Scoring End-point	https://us-south.ml.cloud.ibm.com/v4/deployments/2c6c6ffb-4b5a-4c0e-a92a-bc69b1a8c82c/predictions
Authorization: Bearer <token>	Review the WML authentication documentation for details about generating IAM tokens.
ML-Instance-ID	The "ML-Instance-ID" HTTP header must be populated with the WML instance id, which can be obtained as described here .
Content-type: application/json	Required if the request body is sent in JSON format.

Code Snippets

[cURL](#) [Java](#) [JavaScript](#) [Python](#) [Scala](#)

```
# TODO: manually define and pass values to be scored below
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: Bearer $IAM_TOKEN" --header "ML-Instance-ID: $ML_INSTANCE_ID"
```

Node-RED Node

node-red-contrib-watson-machine-learning

0.1.0

Interact with Watson Machine learning

```
npm install node-red-contrib-watson-machine-learning
```

Node-RED This is a node-red wrapper for the GET and POST data retrieval and run prediction methods of the [Watson Machine Learning](#) service.

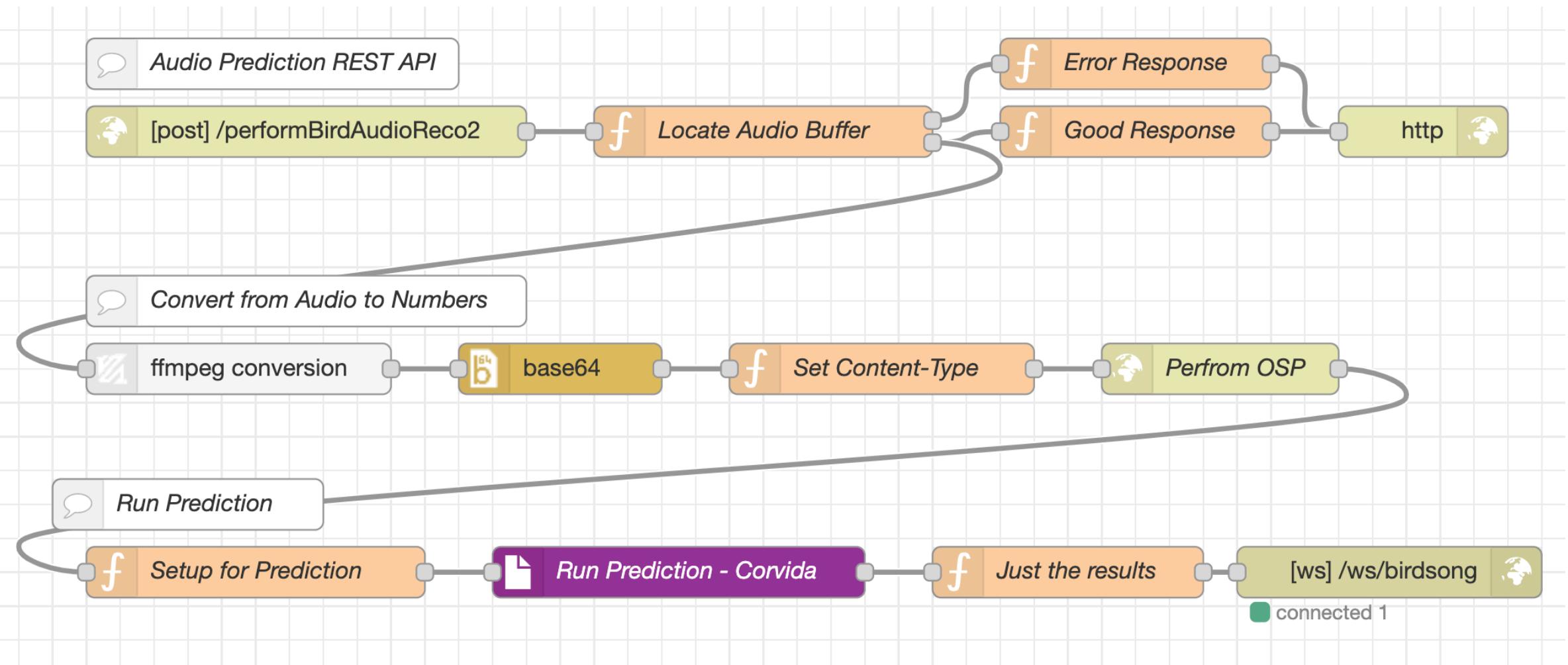
Install

Use the manage palette option to install this node

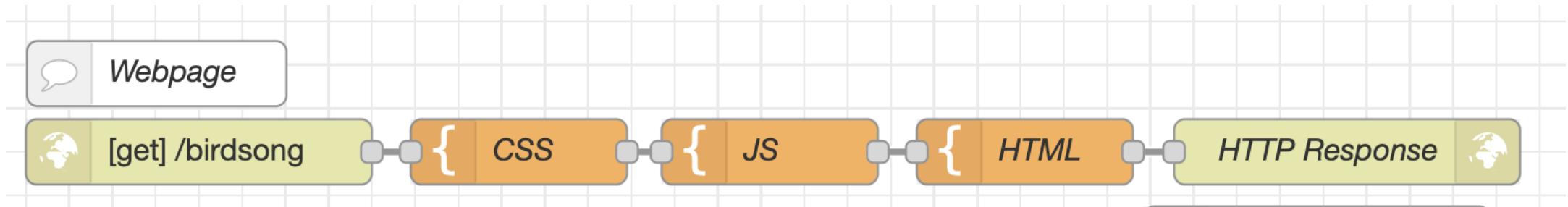
Usage

This node allows you to run predictions against a deployed machine learning model.

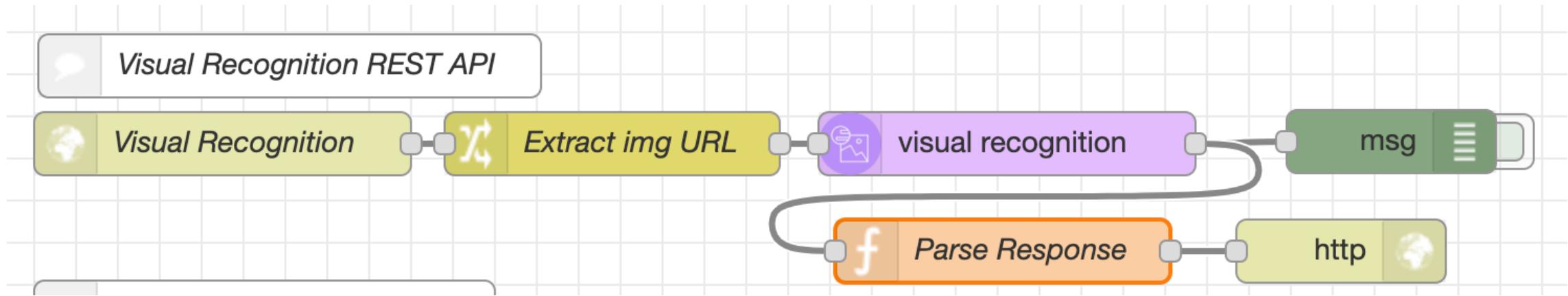
Putting it together



Icing on the cake



Combining with Visual Recognition



Courses

- <https://cognitiveclass.ai/courses/machine-learning-sound>



Introduction to Machine Learning with Sound

Earners of this badge know how to prepare data so that it can be consumed by machine learning models by using IBM Watson Studio, build a binary classification model that can predict which animal is making a sound, build multiclass classification models to detect whether a birdsong is from a bird from a specific order, make predictions on audio files by using a Node-RED app, and create a UI in Node-RED and integrate the Watson Visual Recognition service to identify images of animals.



Node-RED: basics to bots

This badge earner understands how to create simple to complex applications in the Node-RED flow editor. These applications include a Node-RED boilerplate that uses JavaScript and built in IBM Bluemix; a REST API; Watson cognitive applications that can translate text, analyze tone, replay audio and video, and send tweets to Twitter; and a Facebook Messenger bot that uses the Watson Conversation service.

- <https://cognitiveclass.ai/courses/node-red-basics-to-bots>