Splunk .conf18 Splong!

Splunk, Pong...Splong! What Could Go Wrong?

Nicolas Stone, Abhijit Das

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

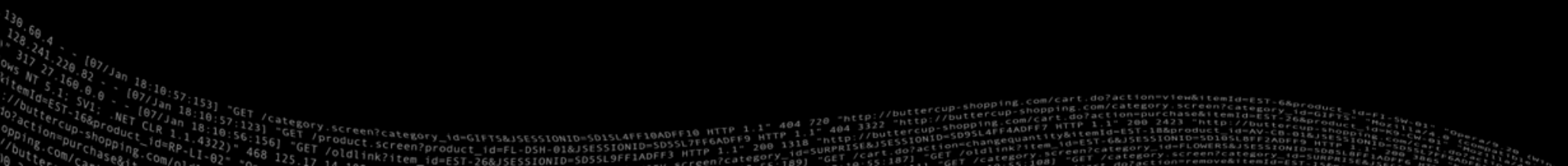
The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Presenters

- Nicolas Stone
  - Solutions Architect on Splunk's Partner Integration team
- Abhijit Das
  - Sr. Sales Engineer on Splunk's Global Strategic Alliances team

Collaborators

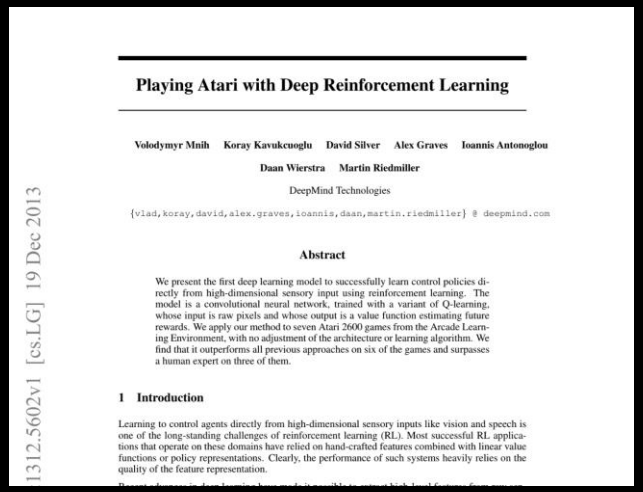
- Tom Heckman
  - SE Manager - Global Strategic Alliances
- Kyle Champlin
  - Sr. Product Manager - Security Markets



Splunk+Pong = Splong!

End-to-End Reinforcement Learning

- DeepMind (Google)
- Uses only screen images of the game
- Zero a priori information about the game
- Beats most classic Atari games
- Neural Networks with sophisticated Credit Assignment



Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou
Daan Wierstra Martin Riedmiller
DeepMind Technologies
{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

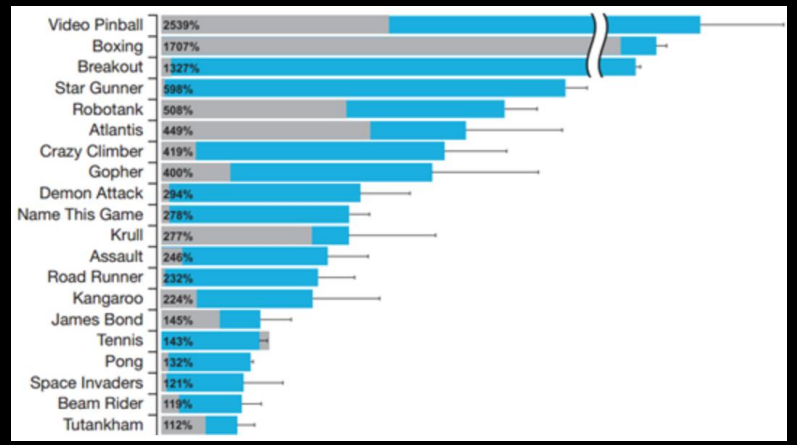
Abstract

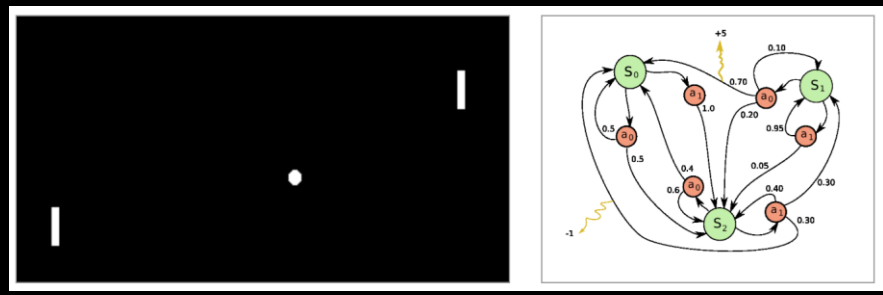
We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

1 Introduction

Learning to control agents directly from high-dimensional sensory inputs like vision and speech is one of the long-standing challenges of reinforcement learning (RL). Most successful RL applications that operate on these domains have relied on hand-crafted features combined with linear value functions or policy representations. Clearly, the performance of such systems heavily relies on the quality of the feature representation.

1312.5602v1 [cs.LG] 19 Dec 2013

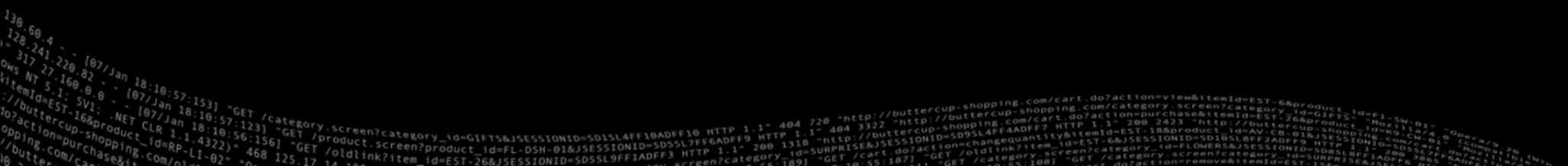




Splunk+Pong = Splong!

WAIT…We're not data scientists!

We just want to see Splunk play video games!



splunk> .conf18

Making machine data accessible, usable and valuable to everyone.

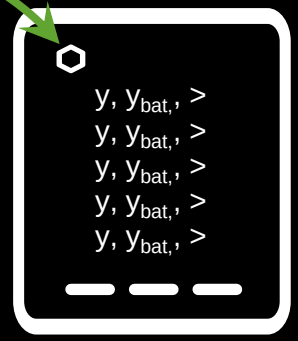
(Not just data scientists)

Splong Data Flow

1 Collect game metrics

```
<script src="{{SPLUNKWEB_URL_PREFIX}}/static/js/i18n.js"></script>
    <!-- Demo site CSS, not needed for PongGame library -->
    <link rel="stylesheet" href="{{SPLUNKWEB_URL_PREFIX}}/static/app/splong/pong-demo.css"> ...
    <!-- Bootstrap: Latest compiled and minified CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" ...
```

Pong.js inside a Splunk dashboard

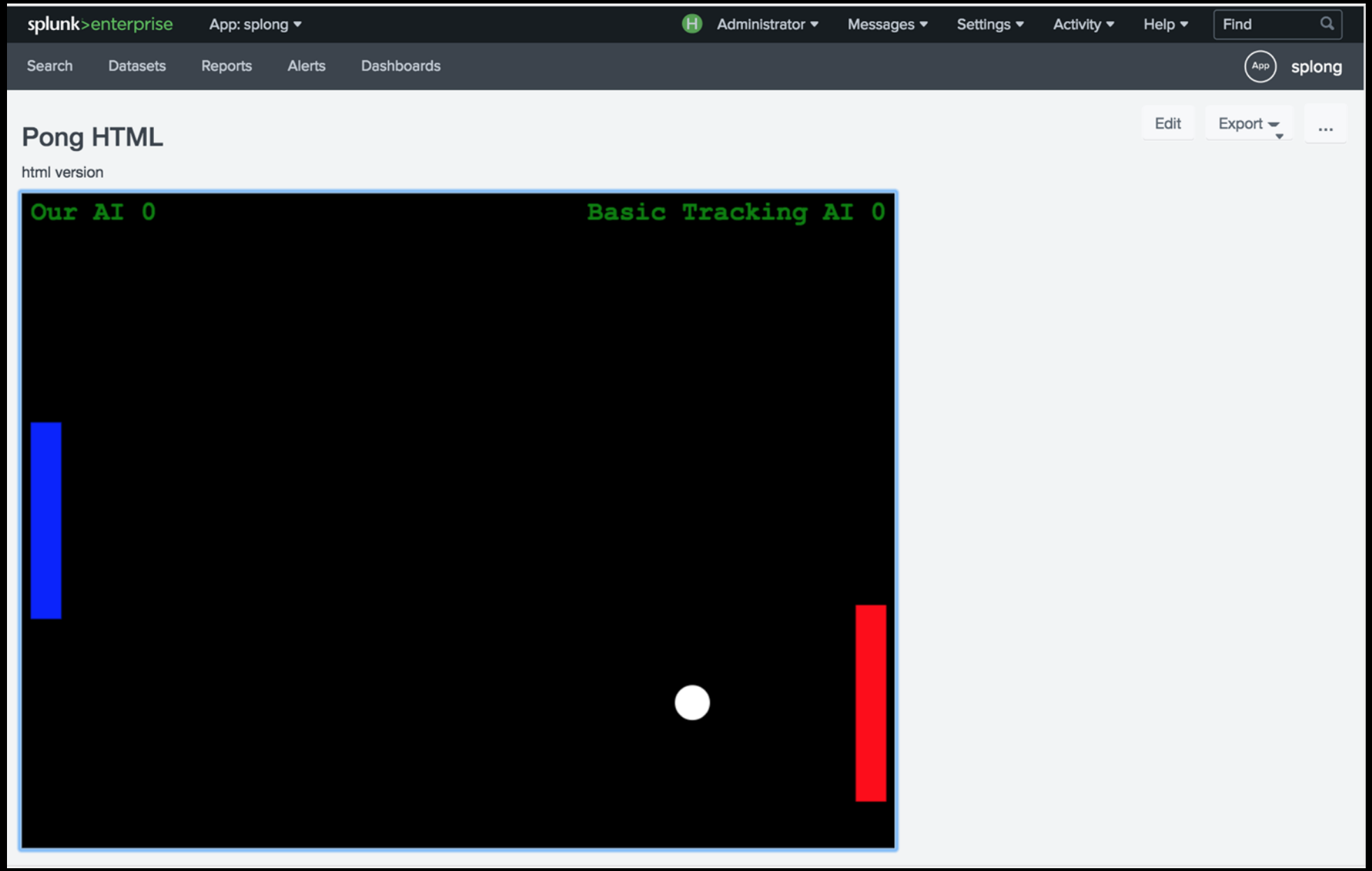


Index

2 Use MLTK to create Bat location predictions

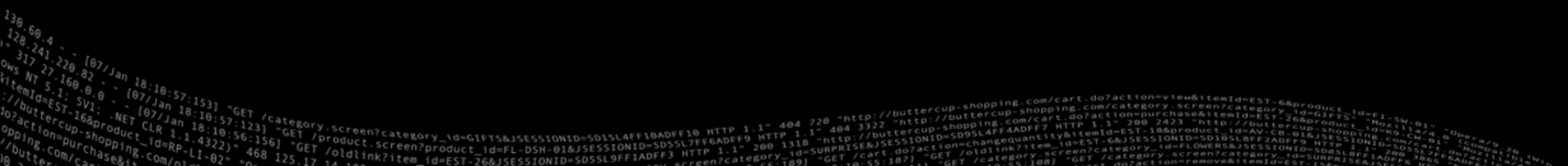
3 Update Pong.js data structures with predictions

splunk> .conf18

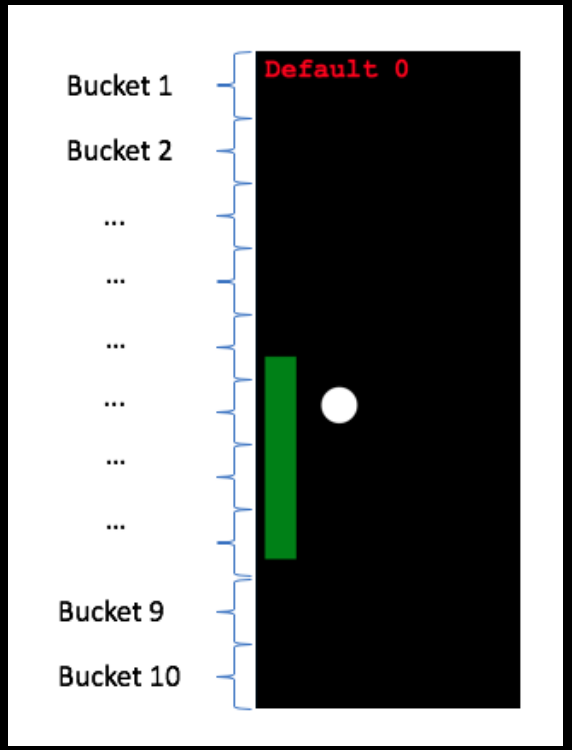


What to log?

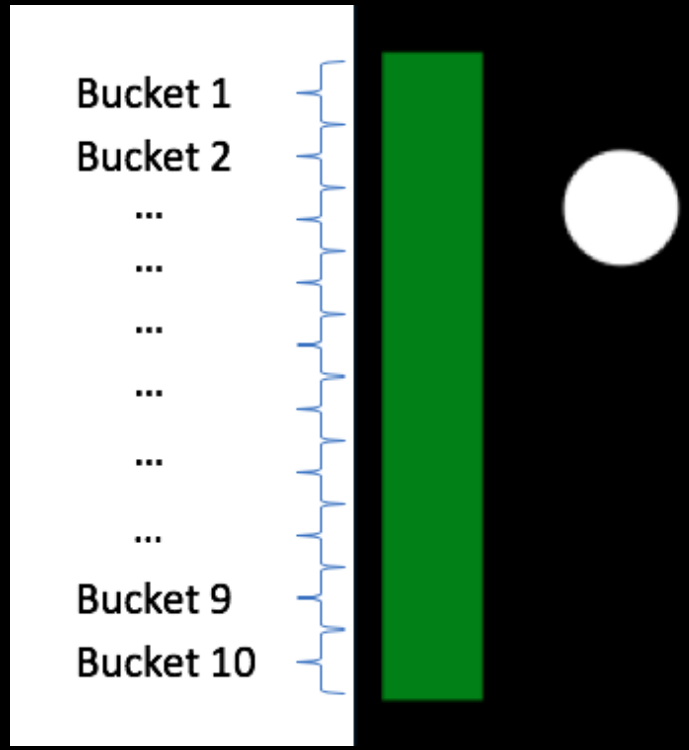
- The embedded Pong game comes with a basic AI that always tries to keep the ball in the middle of it's bat
  - We wanted to beat that…. at least most of the time
- We can beat the AI… Can we create a prediction model that plays like us?
- What do we know about the game of Pong that can make it easier to build this prediction model?



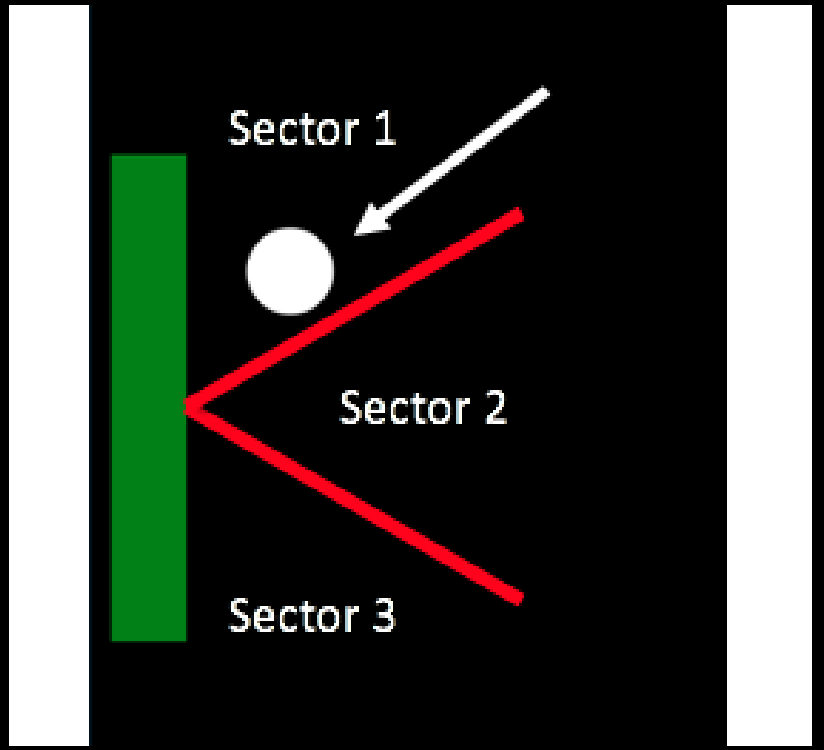
Bat Position



Ball on Bat Position



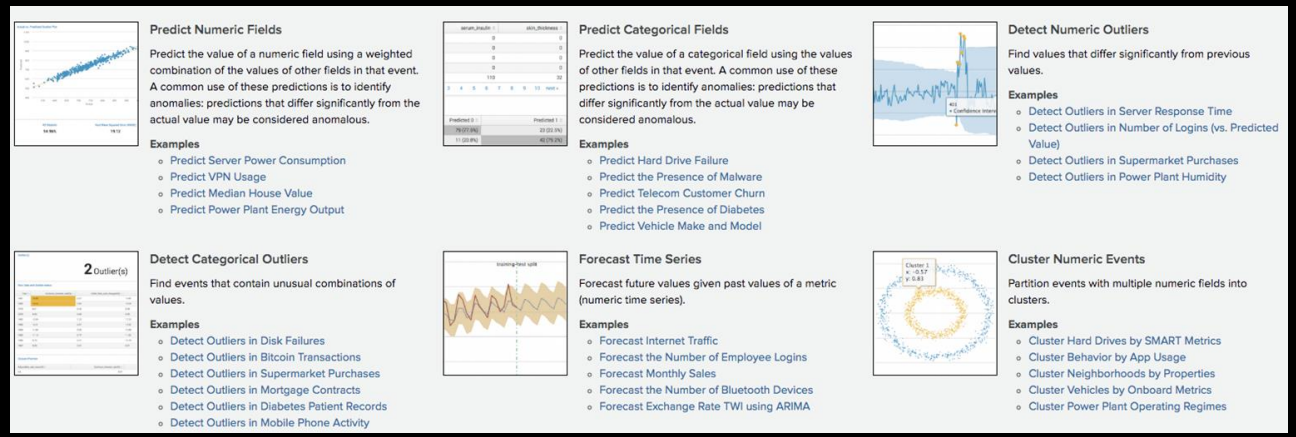
Ball incoming angle



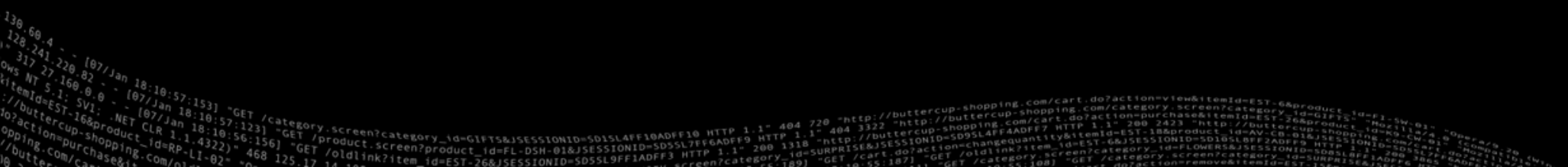
There are other values that matter! Ball and bat speed, etc.

Splunk Machine Learning Toolkit (MLTK)

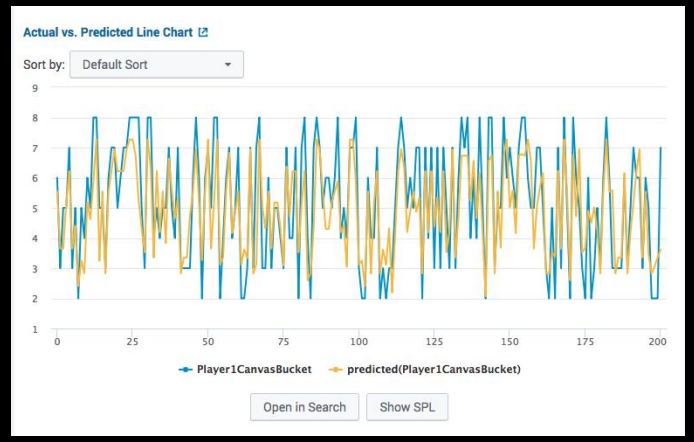
- Numeric fields, Clustering, Time series, etc.
- Example use cases for MLTK:
  - predict median house values
  - forecast monthly sales
  - predict customer churn
  - detect outliers in IT Ops data



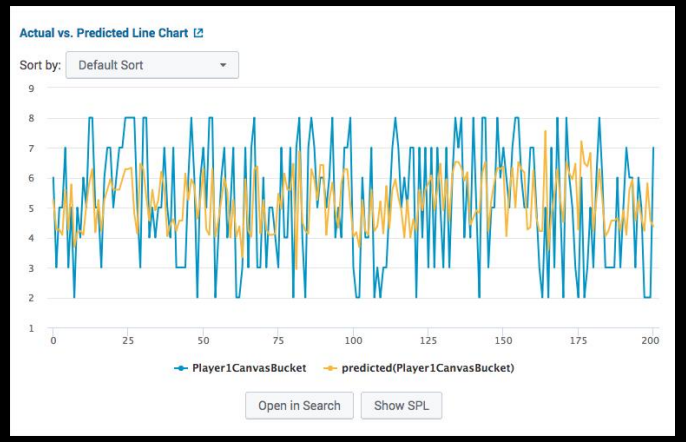
Quick MLTK Demo



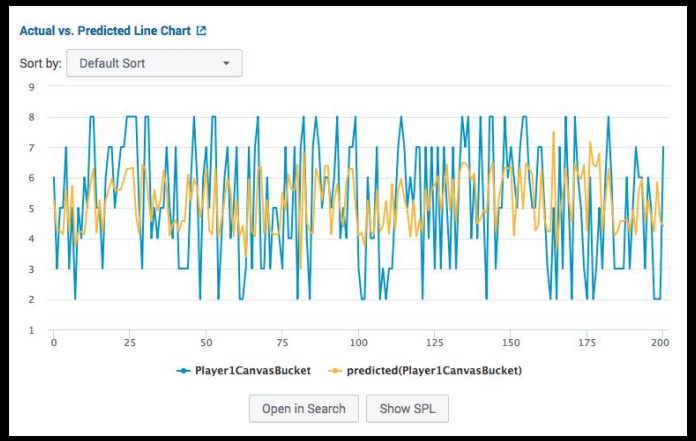
Which algorithm to select?



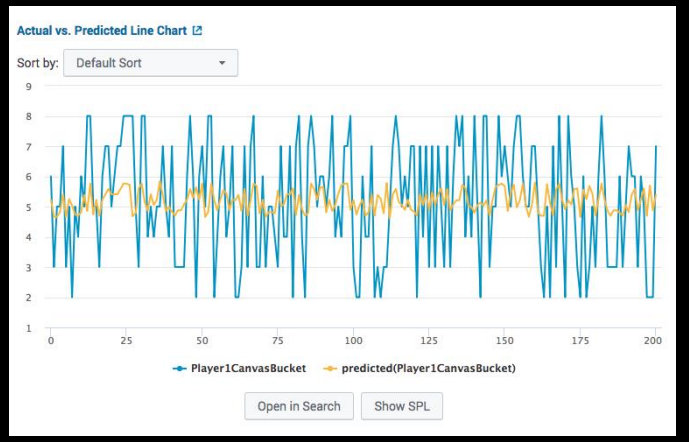
KernelRidge



Linear Regression

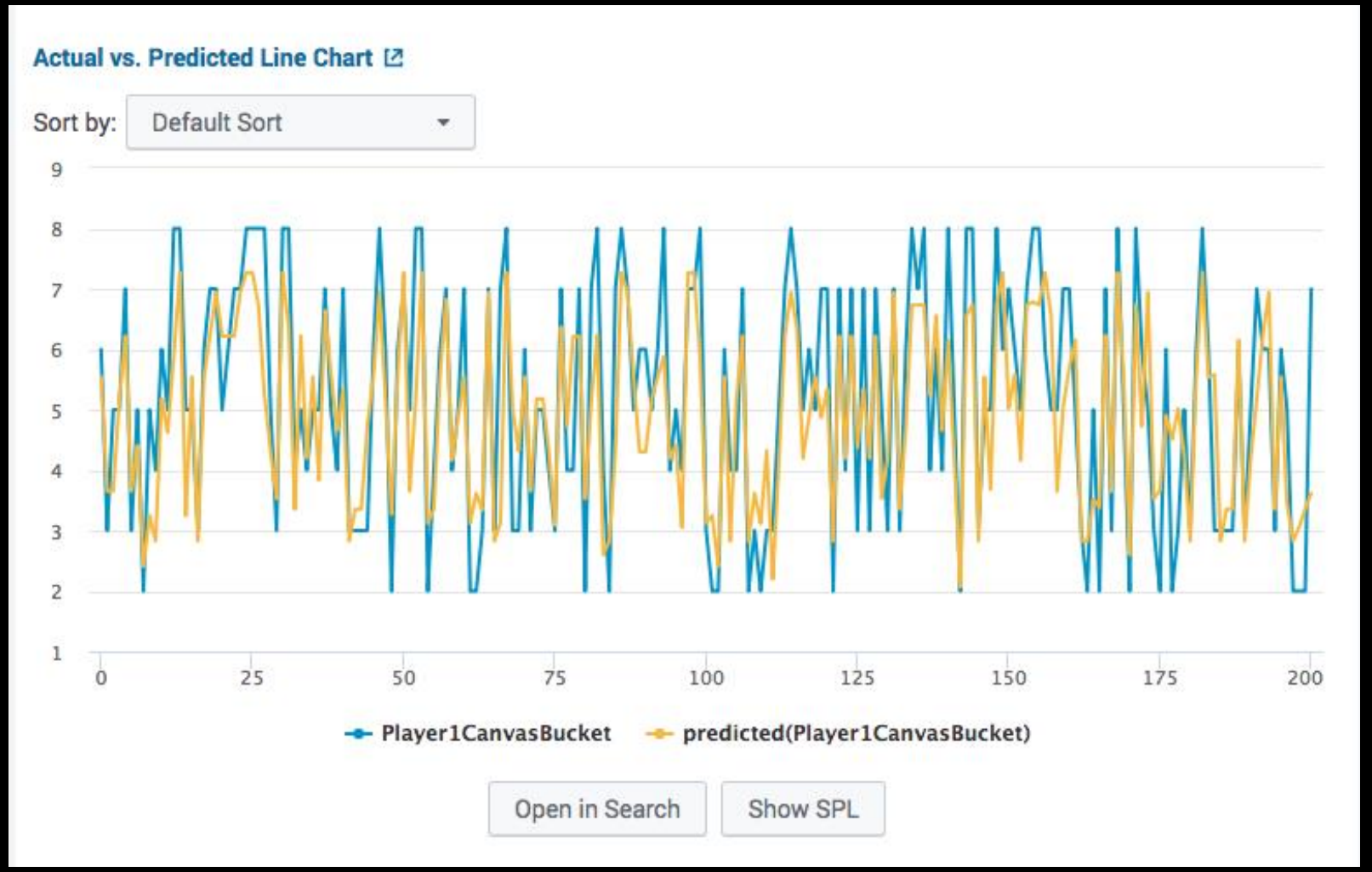


Ridge



ElasticNet

The Winner Is...KernelRidge!



Splong Internals

1

Collect:
- Ball Position (y)
- Ball on Bat Position (y_{bat})
- Ball Angle on Game Bat (Sector 1, 2 or 3)

```
1  <script src="{{SPLUNKWEB_URL_PREFIX}}/
   static/js/i18n.js"></script>
2      <!-- Demo site CSS, not needed for
       PongGame library -->
3      <link rel="stylesheet" href="
       {{SPLUNKWEB_URL_PREFIX}}/static/app/
       splong/pong-demo.css"> ...
4      <!-- Bootstrap: Latest compiled and
       minified CSS -->
5      <link rel="stylesheet" href="https://
       maxcdn.bootstrapcdn.com/
       bootstrap/3.3.6/css/bootstrap.min.css"
       ...
```

Dashbaord with Pong.js

Index

2

After manually playing games, use MLTK to predict bat positions

KernelRidge Predictions

3

Store model output in lookup

Lookup

4

Update data structures in Pong.js to drive

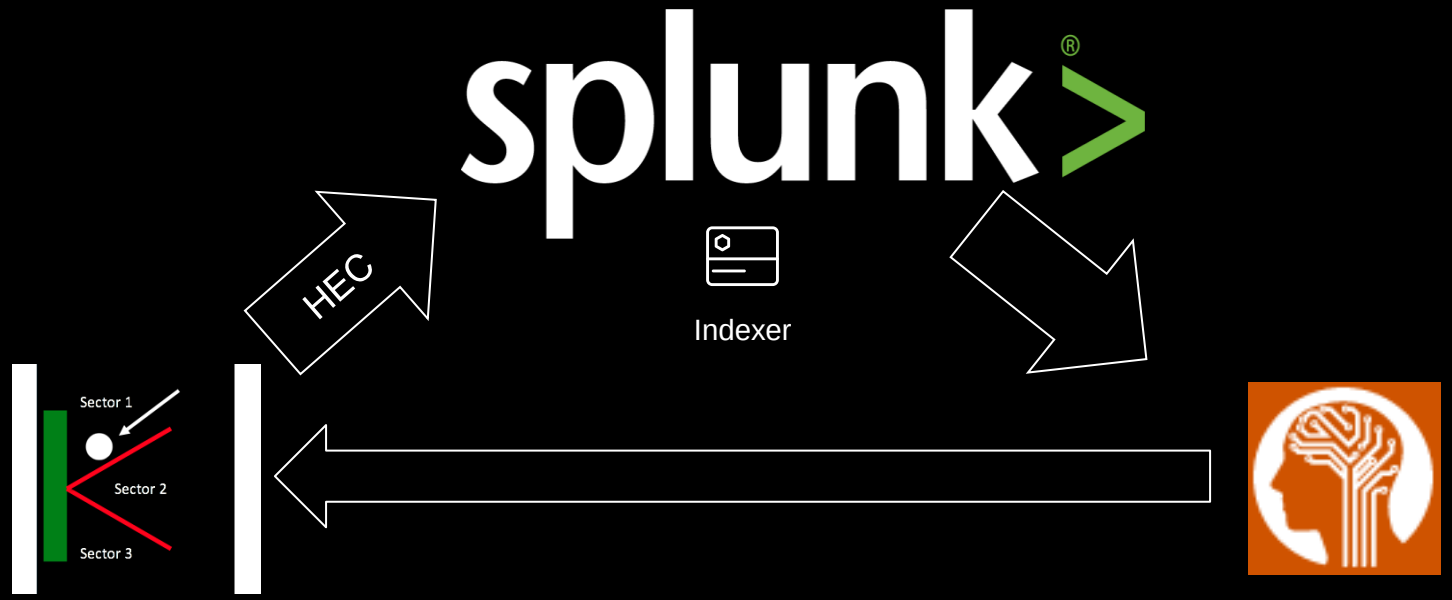
5

Splunk Plays Splong!

Splong! Demo

Processing the data models offline

- We process the data offline by using Node and Splunk's javascript SDK and update the data in the javascript



Processing the data models offline

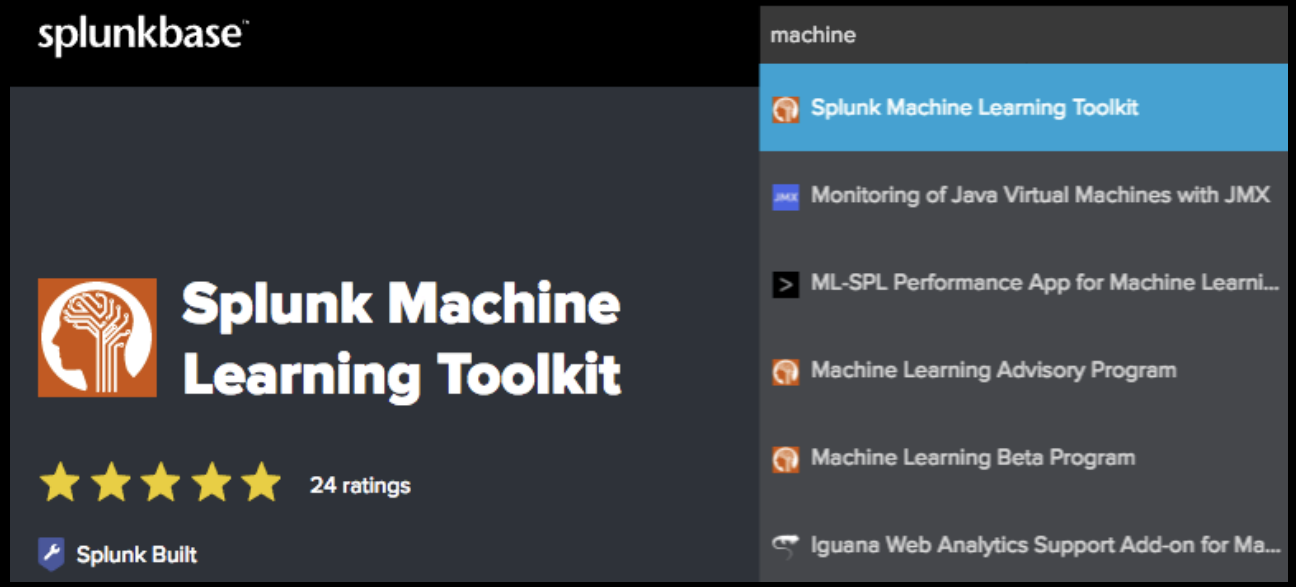
- Now, the variable in the javascript looks like:

```
    var positions = {"0": {"0": {"0": 1, "1": 1, "2": 1}, "1": {"0": 1, "1": 1, "2": 2}, "2": {"0": 1, "1": 1, "2": 2}, "3": {
"0": 1, "1": 1, "2": 1}, "4": {"0": 1, "1": 1, "2": 1}, "5": {"0": 1, "1": 1, "2": 1}, "6": {"0": 1, "1": 1, "2": 1}, "7": {"0
": 1, "1": 1, "2": 1}, "8": {"0": 1, "1": 1, "2": 1}, "9": {"0": 1, "1": 1, "2": 1}, "10": {"0": 1, "1": 1, "2": 1}}, "1": {"0
": {"0": 1, "1": 2, "2": 2}, "1": {"0": 1, "1": 3, "2": 4}, "2": {"0": 1, "1": 3, "2": 4}, "3": {"0": 1, "1": 2, "2": 3}, "4":
 {"0": 1, "1": 2, "2": 2}, "5": {"0": 1, "1": 2, "2": 2}, "6": {"0": 1, "1": 1, "2": 2}, "7": {"0": 1, "1": 1, "2": 1}, "8": {
```

Splunk Machine Learning Toolkit (MLTK)

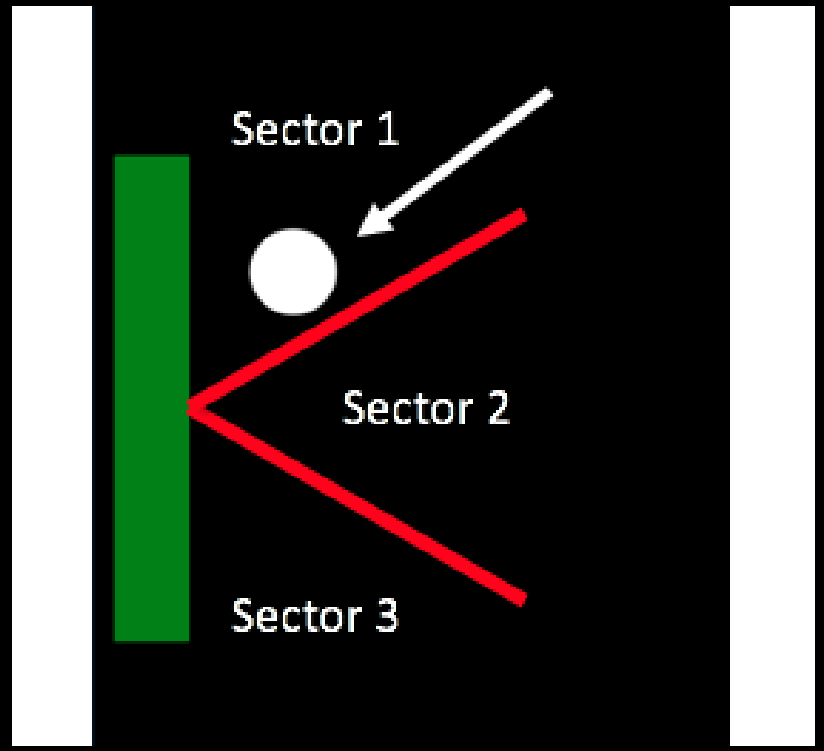
- MLTK is Splunk built and available for free at Splunkbase:

  https://splunkbase.splunk.com/



In Closing

- We wanted to showcase a simple approach to Machine Learning where data scientists are not needed and most importantly have fun!
- Using our own Splunked Pong data, we wanted to create a model in the Machine Learning Tool Kit to play (and hopefully win) against a rudimentary AI.



Q&A

Thank You

Don't forget to rate this session in the .conf18 mobile app

Pong2HEC

Sending events from the browser to Splunk

- Create an inputs.conf with a hardcoded token
  - allowQueryStringAuth!
    - Allows us to push events to a HEC endpoint using jQuery.post()

This is not a secure way of handling things...

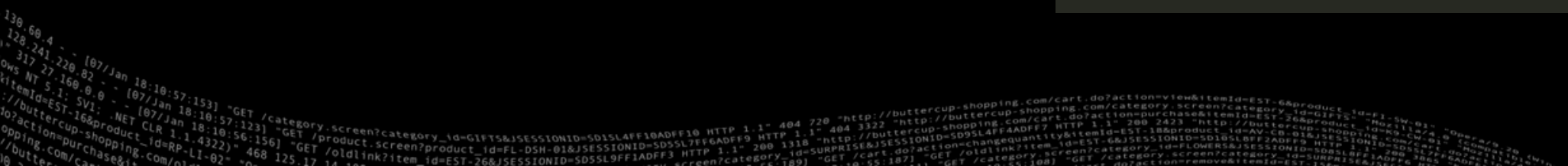
```
hec_token = '24b0e57d-15a4-4d7d-8a95-8606588fb4ca'
path = '/services/collector/event?token=' + hec_token;
hec_ep = "http://" + document.location.hostname + ':8088' + path;

currTime = (new Date).getTime() / 1000;
Ball["velocity"] = BallVelocity;
var game_state = {};
game_state["BatPosition"] = Player2Bat;
game_state["Ball"] = Ball;
game_state["RoundHash"] = RoundHash;
game_state["Player"]  = playerTurn;
var evt = {};
evt["sourcetype"] = "splong";
evt["event"] = game_state;
evt["time"] = currTime;

try {
    jQuery.post(hec_ep,JSON.stringify(evt));
}

catch(err){
    console.log(err);
}
```

```
[http://splong_hit_pair]
description = splong HEC token
disabled = 0
index = main
source = splong
sourcetype = splong_hit_pair
token = 24b0e57d-15a4-4d7d-8a95-8606588fb4ca
allowQueryStringAuth = true
```



splunk> .conf18

Processing the data models offline

- Final crontab script:

```
#!/bin/bash

# Run fit and apply

/usr/bin/node /home/splunk/splunk-sdk-javascript-1.8.4/examples/node/search.js --search 'search sourcetype=splong_hit_pair | w
here Player1CanvasBucket!="" AND Player2CanvasBucket!="" AND Player2BatBucket!="" AND Player2Sector!="" | fit KernelRidge  "Pl
ayer1CanvasBucket" from "Player2CanvasBucket" "Player2Sector" "Player2BatBucket" into "splong_ai"' --username 'admin' --passwo
rd 'admin123'

/usr/bin/node /home/splunk/splunk-sdk-javascript-1.8.4/examples/node/search.js --search '| inputlookup splong_lookup.csv | app
ly "splong_ai" | rename "predicted(Player1CanvasBucket)" AS predicted | eval predicted=ceiling(predicted) | table predicted, P
layer2CanvasBucket, Player2Sector, Player2BatBucket| sort by Player2CanvasBucket, Player2Sector, Player2BatBucket| outputlooku
p apply_results.csv'  --username 'admin' --password 'admin123'

/usr/bin/python /home/splunk/splunk-sdk-javascript-1.8.4/examples/node/dynamic_conversion.py /opt/splunk/etc/apps/search/looku
ps/apply_results.csv /mnt/data/splunk/etc/apps/splong/appserver/static/testinput_pong_ai.js /mnt/data/splunk/etc/apps/splong/a
ppserver/static/pong_ai.js
```

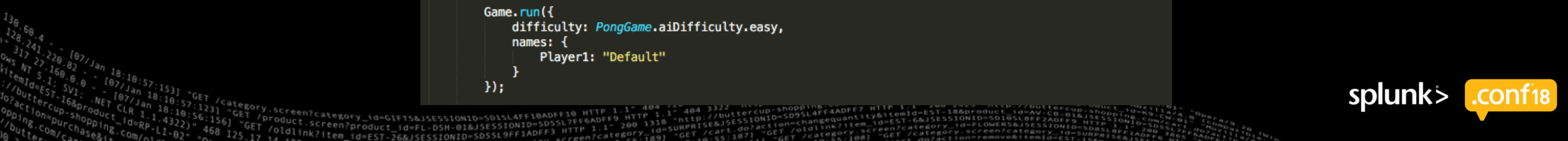
Embedding Pong into a Splunk dashboard

- We used an open source vanilla JS and HTML5 Canvas Pong implementation
  - https://github.com/SMenigat/html5-pong
    - First, convert a Splunk dashboard to HTML.
    - Then include the Pong JS library and embed the canvas.

```
<!-- PongGame: Latest JavaScript library -->
<script src="{{SPLUNKWEB_URL_PREFIX}}/static/app/splong/pong_manual.js" type="application/javascript"></script>
```

```
</script>
        <div class="row">
            <div class="col-md-8">
                <div id="canvasWrapper">
                    <div>
                        <canvas id="gameCanvas" tabindex="1"></canvas>
                    </div>
                </div>
            </div>
            <div class="col-md-4" id="optionWrapper">
            </div>
        </div>
        <hr />
    </div>
    <script>
        // initialize the game
        var Game = new PongGame('gameCanvas');

        Game.run({
            difficulty: PongGame.aiDifficulty.easy,
            names: {
                Player1: "Default"
            }
        });
```



splunk> .conf18