

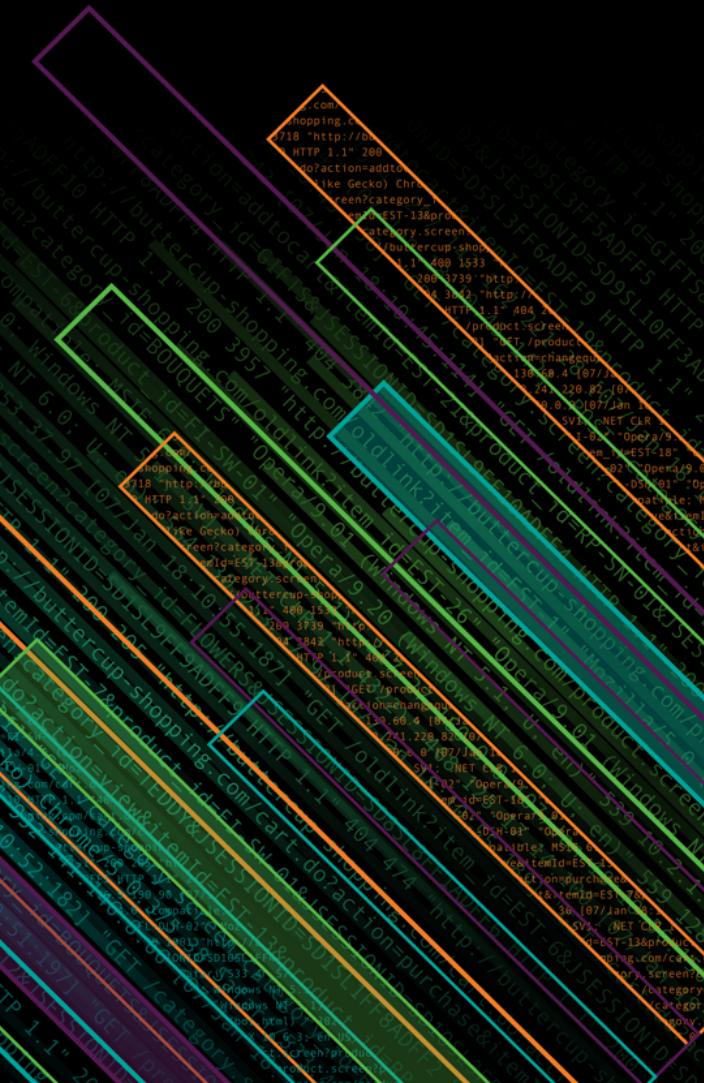


# Extending Splunk MLTK using GitHub Community

Gyanendra Rana  
Nathan Worsham

Senior Product Manager Machine Learning  
IS Security Administrator, [pinnacol.com](http://pinnacol.com)

October 2018



# 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.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2018 Splunk Inc. All rights reserved.

# Gyanendra Rana



- ▶ Splunk Senior Product Manager – Machine Learning
- ▶ 14 years of Software Engineering experience
- ▶ Responsible for Machine Learning Customer Advisory Program
- ▶ In my free time, I love to play with my 5 year old son

# Nathan Worsham



- ▶ Wear a lot of hats
  - IS Security Administrator
  - Splunk Admin and Advocate
  - GSEC, GCFA, GIAC Advisory Board
  - M.S. Data Science
- ▶ Unique view



# Agenda

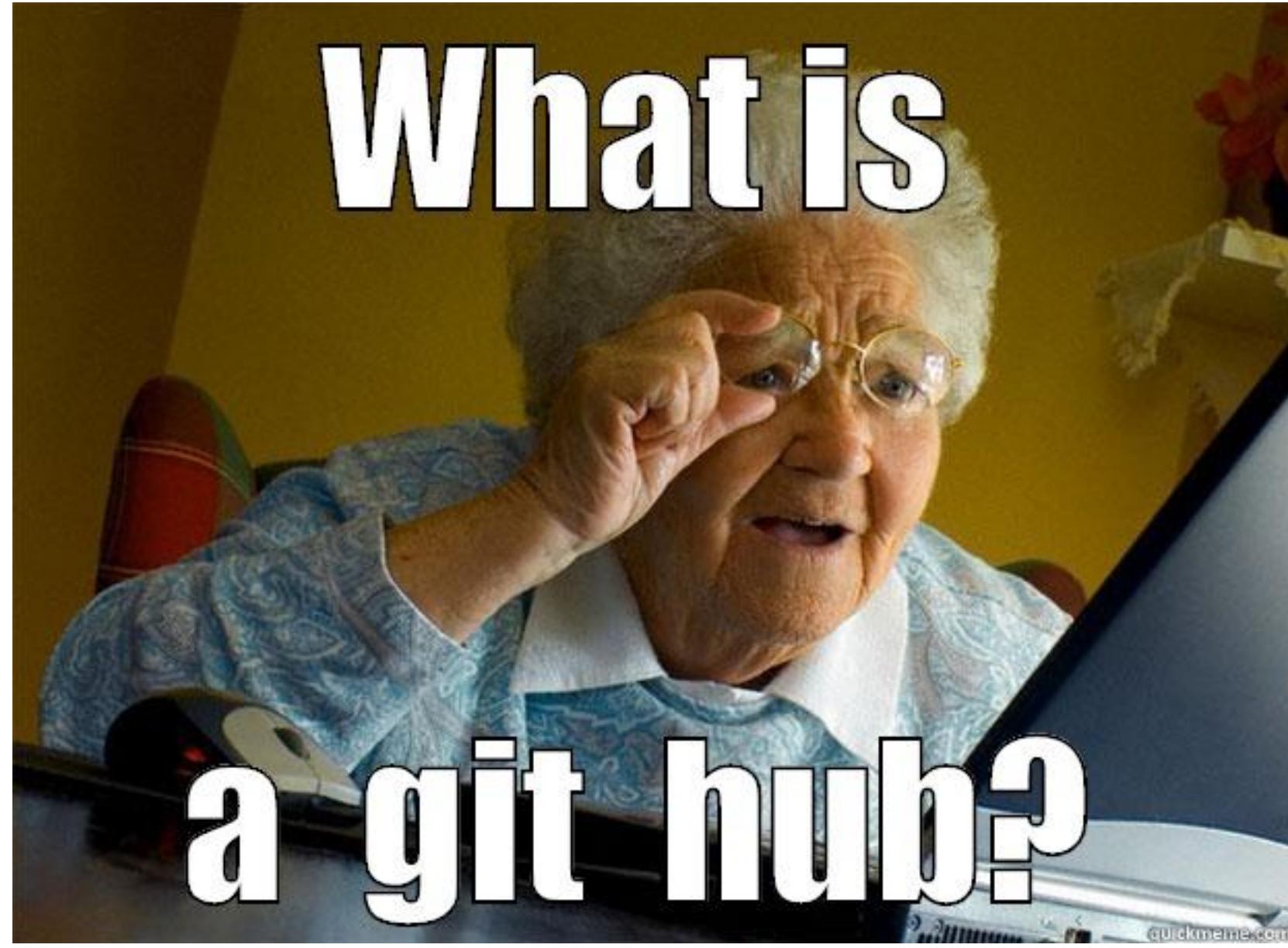
## ► Github Details

- What is Github?
- What are we Building?
- What Problem it will solve?
- Who will benefit from this capability?

## ► ML SPL API Recap

- ML-SPL Overview
- Algorithms
- Python Scientific computing Add-on
- ML-SPL Extensibility API
- Doc links on ML-SPL API
- Demo

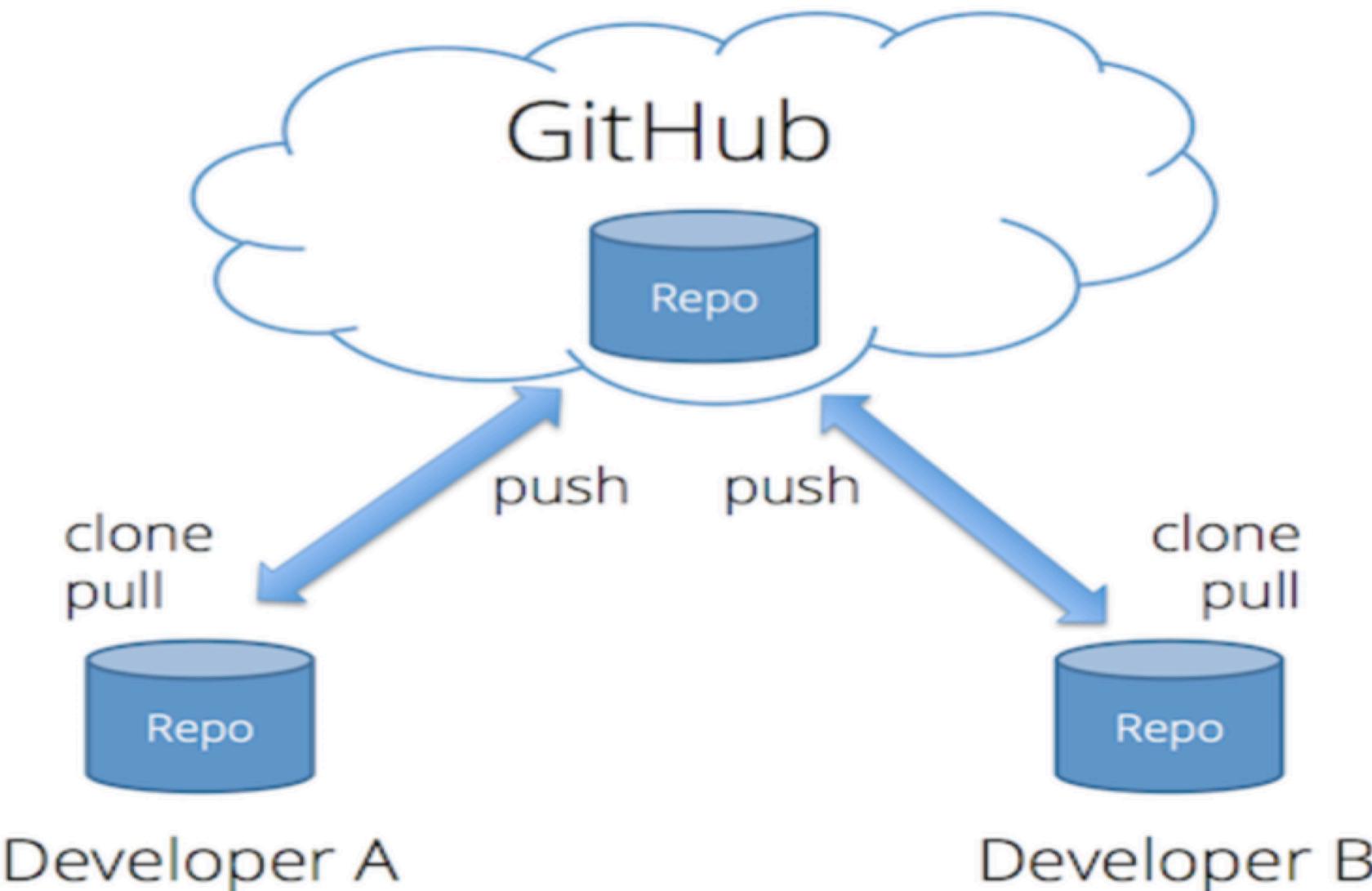
# GitHub Details



quickmeme.com

# What is a GitHub?

- ▶ GitHub is a web based repository and cloud-based service
- ▶ Online Code collaboration
- ▶ Version control helps developers track and manage changes to a software code
- ▶ Git is a distributed version control system.  
Entire codebase and history is available on every developer's computer for easy branching and merging
- ▶ GitHub has extra functionality on top of git  
UI, documentations, bug tracking, pull request and many more



338.60.4 - - [07/Jan 18:10:57:153] "GET /category.screen?category\_id=GIFTS&JSESSIONID=SD15LAFF10ADFFF0 HTTP/1.1" 404 720 "http://buttercup-shopping.com/cart.do?action=view&itemId=EST-6&product\_id=F1-SW-01" "Opera/9.80 (Windows NT 5.1; SV1; .NET CLR 1.1.4322)" 468 125.17 14 "http://buttercup-shopping.com/cart.do?action=update&oldlink?item\_id=EST-26&product\_id=EST-26&product\_name=GIFTS-F1-SW-01" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36" 200 2423 "http://buttercup-shopping.com/cart.do?action=changequantity&itemId=EST-18&productId=EST-6&SESSIONID=SD16SLBFF2ADEF0" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36 128.241.220.82 - - [07/Jan 18:10:57:153] "GET /category.screen?category\_id=GIFTS&JSESSIONID=SD15LAFF10ADFFF0 HTTP/1.1" 404 3322 "http://buttercup-shopping.com/cart.do?action=update&oldlink?item\_id=EST-26&product\_id=EST-26&product\_name=GIFTS-F1-SW-01" "Opera/9.80 (Windows NT 5.1; SV1; .NET CLR 1.1.4322)" 468 125.17 14 "http://buttercup-shopping.com/cart.do?action=changequantity&itemId=EST-18&productId=EST-6&SESSIONID=SD16SLBFF2ADEF0" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36 317.27.160.0.0 - - [07/Jan 18:10:56:156] "GET /oldlink?item\_id=EST-26&JSESSIONID=SD55L9FF1ADFF3 HTTP/1.1" 200 4318 "http://buttercup-shopping.com/cart.do?action=changequantity&itemId=EST-18&productId=EST-6&SESSIONID=SD16SLBFF2ADEF0" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36 128.241.220.82 - - [07/Jan 18:10:57:153] "GET /category.screen?category\_id=GIFTS&JSESSIONID=SD15LAFF10ADFFF0 HTTP/1.1" 404 3322 "http://buttercup-shopping.com/cart.do?action=update&oldlink?item\_id=EST-26&product\_id=EST-26&product\_name=GIFTS-F1-SW-01" "Opera/9.80 (Windows NT 5.1; SV1; .NET CLR 1.1.4322)" 468 125.17 14 "http://buttercup-shopping.com/cart.do?action=changequantity&itemId=EST-18&productId=EST-6&SESSIONID=SD16SLBFF2ADEF0" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36

# What are we building?

- ▶ Open Source Community
  - ▶ Platform for Sharing and reusing Algorithms



# What Problem it will solve?

- ▶ Learning and reusing machine learning algorithms from the Splunk open source community.
  - ▶ Raising issues as necessary
  - ▶ “Moar” Algorithms

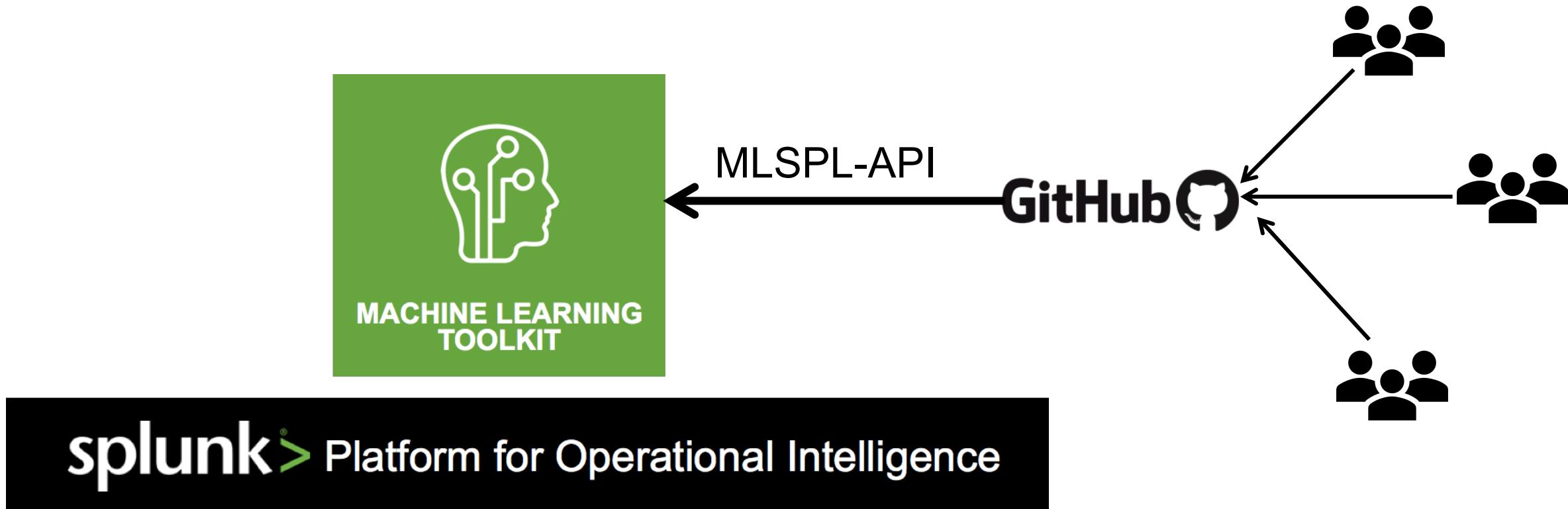


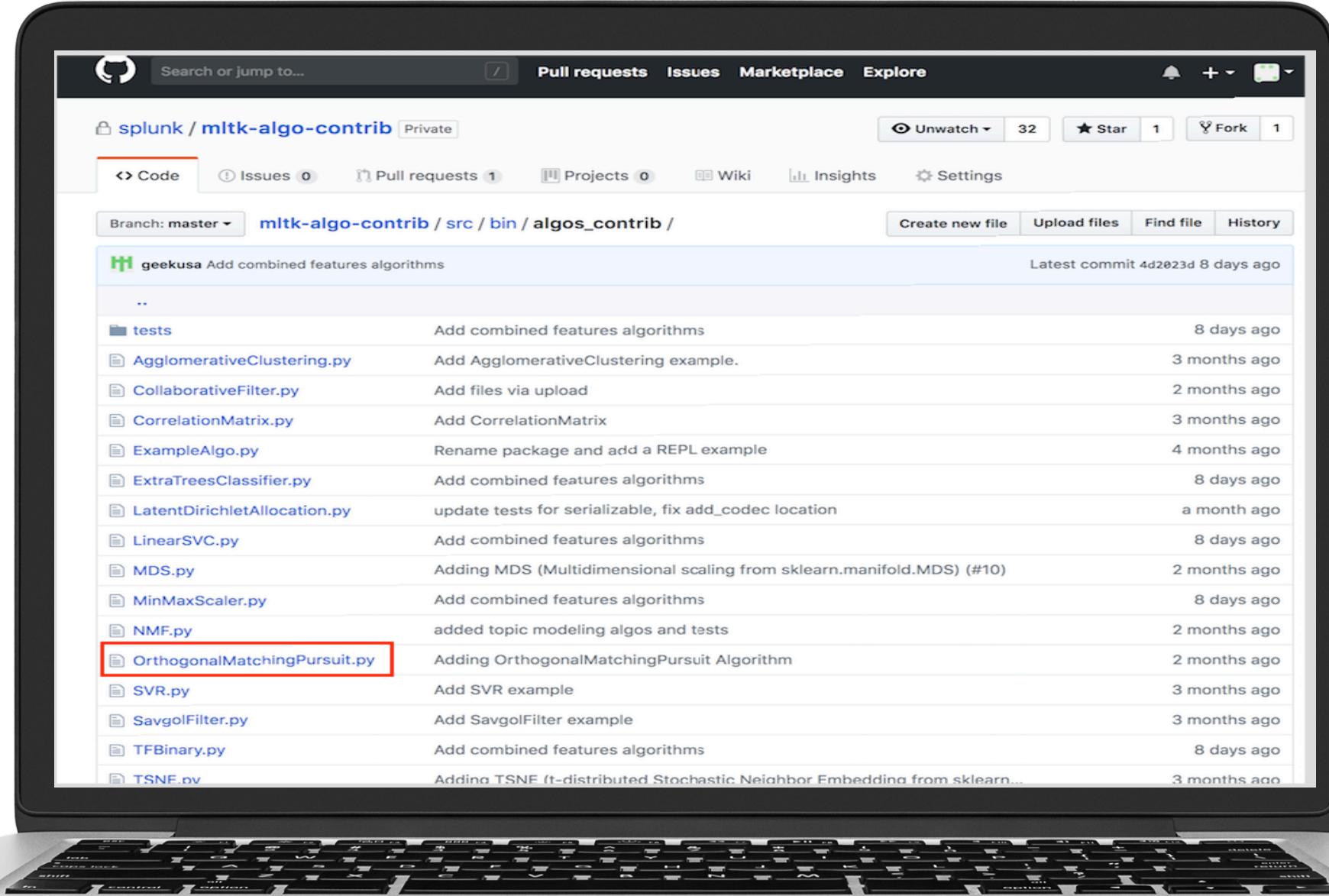
# Who will benefit from this initiative ?

- ▶ Existing or new users of the MLTK will have access to more algorithms, powering new and valuable use cases in Splunk
- ▶ Customers looking for solutions that are outside of MLTK immediate 30 odd algorithms.
- ▶ Consumers of the ML-SPL API will have opportunities of sharing their custom algorithms



# How does it work in Splunk?





The image shows a laptop screen with the Splunk Machine Learning Toolkit application open. The search bar at the top contains the following command:

```
| inputlookup server_power.csv | fit OrthogonalMatchingPursuit it_intercept=true "ac_power" n_nonzero_coefs=8 from "total-unhalted_core_cycles" "total-instructions_retired" "total-last_level_cache_references" "total-memory_bus_transactions" "total-cpu-utilization" "total-disk-accesses" "total-disk-blocks" "total-disk-utilization" into model|
```

The results table below the search bar shows 31,272 results (before 8/29/18 11:24:57.000 AM) with no event sampling. The table has columns for time, ac\_power, predicted(ac\_power), total-cpu-utilization, total-disk-accesses, total-disk-blocks, total-disk-utilization, total-instructions\_retired, total-last\_level\_cache\_references, total-memory\_bus\_transactions, and total-unhalted\_core\_cycles. The data spans from April 2008 to April 2009. The bottom right corner of the screen features the Splunk conf18 logo.

# ML-SPL API Recap

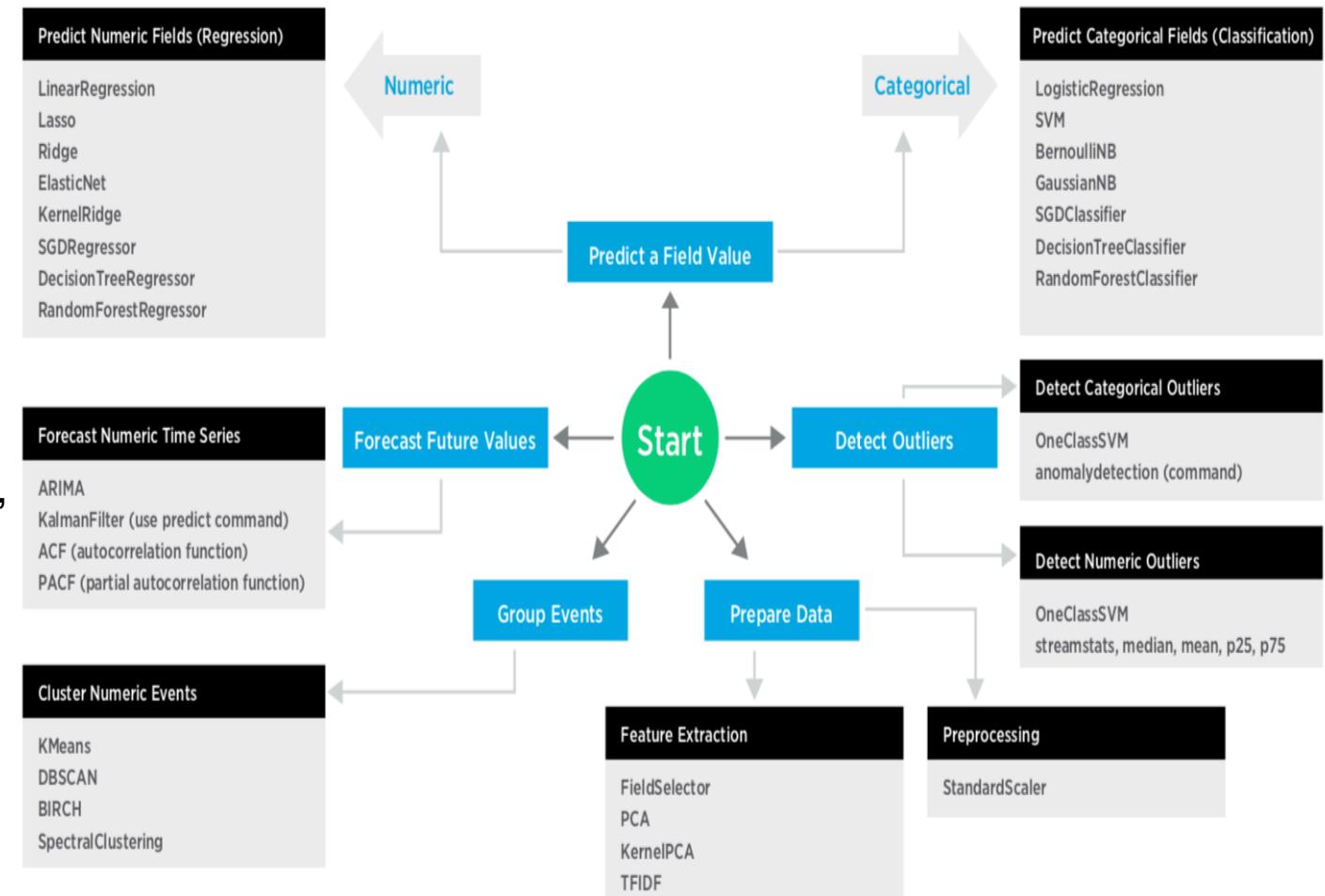


# ML-SPL Overview

- ▶ A suite of SPL search commands specifically for Machine Learning:
    - **fit**
    - **apply**
    - **summary**
    - **listmodels**
    - **deletemodel**
    - **sample**
  - ▶ Implemented using modules from the Python for Scientific Computing Add-on for Splunk:
    - **scikit-learn, numpy, pandas, statsmodels, scipy**

# Algorithms

- ▶ 30 packaged algorithms come with the MLTK
  - **Regressors** – predicting numeric output
  - **Classifiers** – predicting categorical output
  - **Clusterers** – grouping like with like
  - **Preprocessing**
  - **Time series analysis** – e.g. ARIMA, ACF, PACF
  - **Feature extraction** – e.g. PCA, TFIDF



# Python for Scientific Computing (PSC)

Free add-on available on Splunkbase

- ▶ Required dependency of the MLTK
- ▶ Provides needed libraries for ML
- ▶ Miniconda-based
- ▶ Most notable packages:
  - **scikit-learn**
  - **pandas**
  - **NumPy**
  - **SciPy**
  - **StatsModels**

# ML-SPL Extensibility API

- ▶ The ML-SPL Extensibility API allows one **to add custom algorithms** that can be used with the MLTK's search commands.
  - ▶ ML-SPL API: Similar to...
    - Python SDK for custom commands API
    - Custom Visualization API (a.k.a. “modviz”)
    - scikit-learn estimator API
  - ▶ Can be used in separate standalone apps too!
    - Still must have MI TK & PSC installed

# Documentation

- We have ML-SPL API documentation

<http://docs.splunk.com/Documentation/MLApp/latest/API/Introduction>

- Also, Checkout the last year conf presentation on ML-SPL Extensibility API :  
<https://conf.splunk.com/files/2017/slides/advanced-machine-learning-using-the-extensible-ml-api.pdf>

# Demo

# We Interrupt Your Regularly Scheduled Program

## My MLTK journey



# Final Fantasy

# **STOP the madness!**

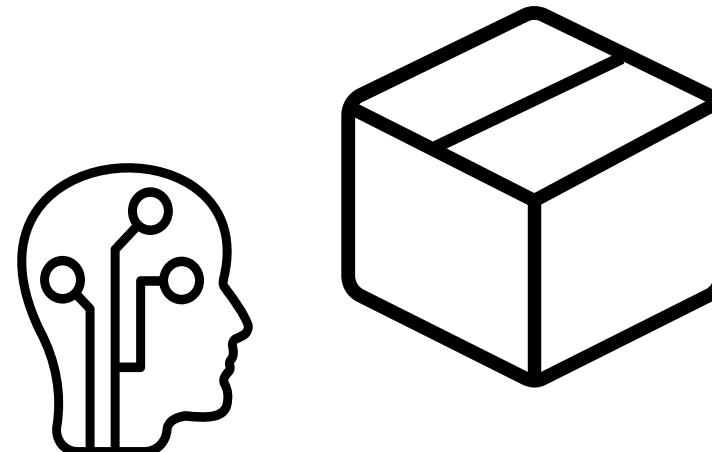
- ▶ Help our SOC only see emails of interest
  - ▶ Classic binary classification problem
  - ▶ But how to get there?



# Something, Something... Machine Learning

**NARRATOR:** “Staring at the black box, Nathan looked uncertain.”

- ▶ Need to get email into Splunk
    - Apps for that... right?
  - ▶ Process the emails for features
    - Some sort of SPL magic... right?
  - ▶ Finally MLTK make a model
    - Apply model on production emails



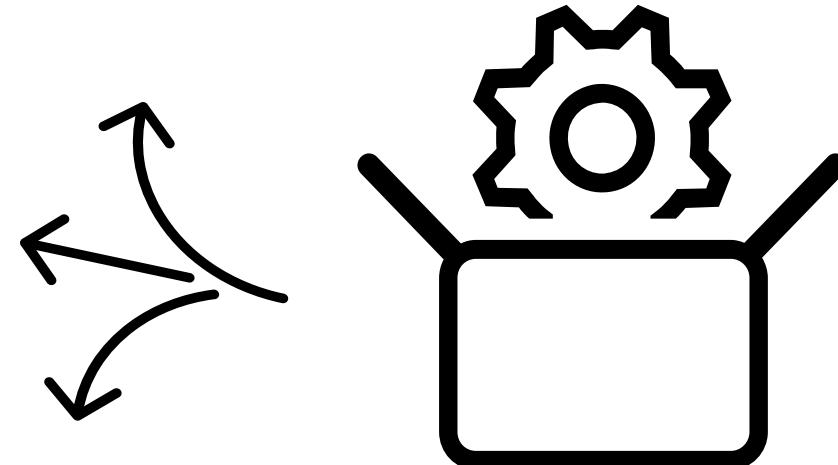
“If you want to make an apple pie from scratch, you must first create the universe.”

Carl Sagan

## Steps to Get There

**NARRATOR:** “Inside the box, a pair of baby shoes gleamed back at him.”

- ▶ Need to get email into Splunk
    - Tried TA-mailclient, need to fork
  - ▶ Process the emails for features
    - 2 new apps, for cleaning and extraction
  - ▶ Finally MLTK make a model
    - Use ML-SPL API to add algorithms



# A Tale of Two Feature Sets

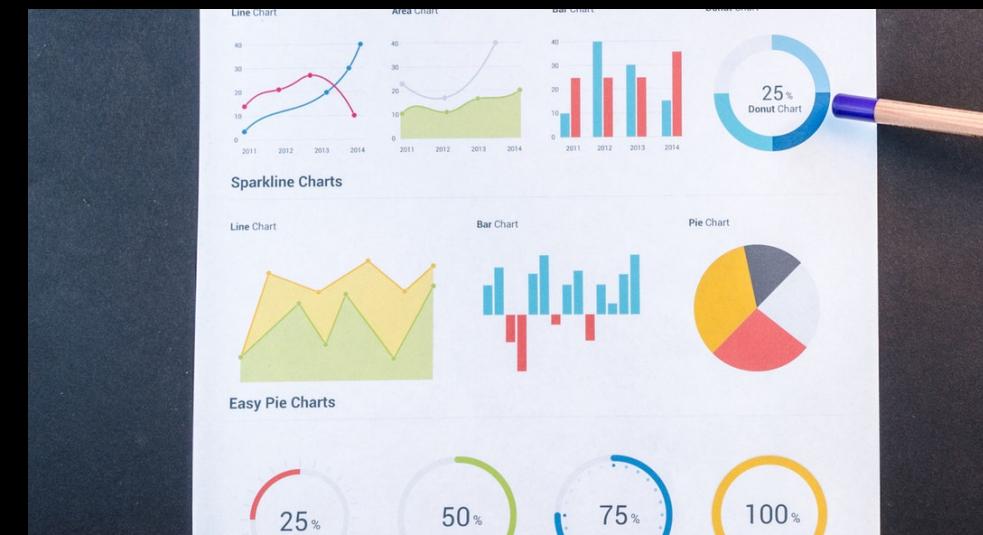
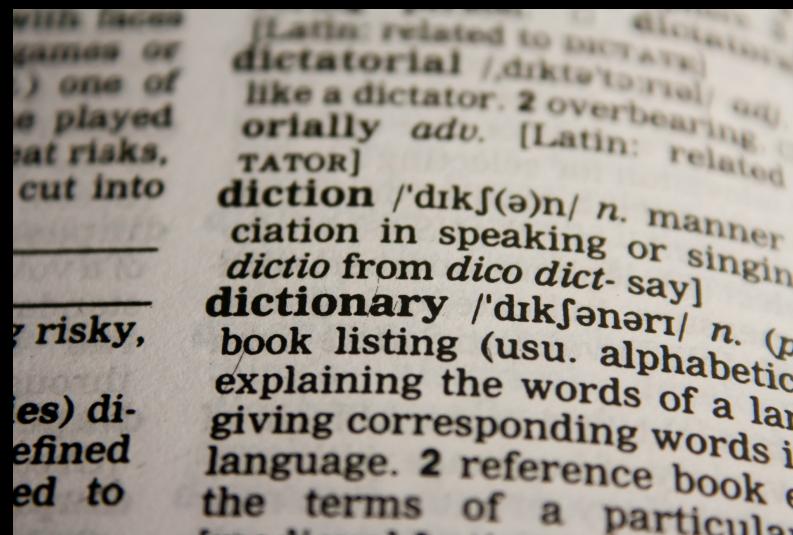
# Emails are more than just text

# ► Text Features

- Language is hard – especially for machines
  - Not literally solid... tough, see what I mean?

## ► Non-text features

- ML is used to these
  - Body length, has attachment, etc



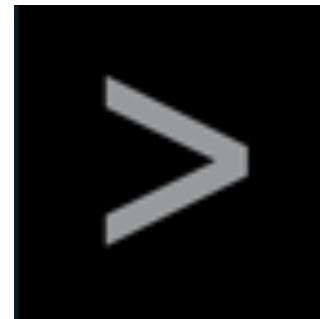
# There is One Born Every Minute

Open framework to the rescue

## ► Mail-Parser Plus App

- Custom command – mailparser
- ML friendly features

<https://splunkbase.splunk.com/app/4129>



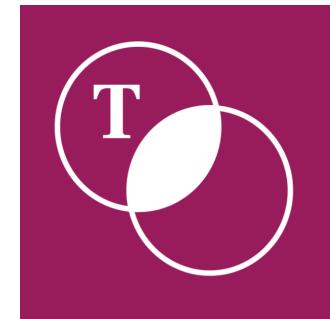
Blog:

<https://github.com/geekusa/nlp-text-analytics>

## ► NLP Text Analytics App

- Custom commands
- MLTK custom algos
- Dashboards

<https://splunkbase.splunk.com/app/4066/>



Blog:

[https://github.com/geekusa/nlp-text-analytics/blob/master/PROJECT\\_FILES/blog.md](https://github.com/geekusa/nlp-text-analytics/blob/master/PROJECT_FILES/blog.md)

# Talkin' Bout My App Creation

## NLP Text Analytics... so far

- ▶ 3 custom commands
    - cleantext, vader, bs4
    - NLTK – that's Natural Language Toolkit
    - BeautifulSoup4
  - ▶ 3 primary dashboards
    - Counts (EDA), Themes (ML- Unsupervised Learning), Sentiment
  - ▶ 7 Algorithms
    - Topic Modeling
    - Combined Features
- 

### ▶ ML-SPL API file hierarchy

nlp\_text\_analytics

└ bin

  └ topic\_modeling\_algos

    TruncatedSVD.py

...

  └ combined\_feats\_algos

    LinearSVC.py

...

└ default

  algos.conf



[LinearSVC]

package = combined\_feats\_algos

...

# NLP Example Algo

## LinearSVC.py

```
#!/usr/bin/env python
from sklearn.svm import LinearSVC as _LinearSVC
from codec import codecs_manager
from base import BaseAlgo, ClassifierMixin
from util.param_util import convert_params

class LinearSVC(ClassifierMixin, BaseAlgo):
    def __init__(self, options):
        self.handle_options(options)
        out_params = convert_params(
            options.get('params', {}),
            floats=['gamma', 'C', 'tol', 'intercept_scaling'],
            ints=['random_state', 'max_iter'],
            strs=['penalty', 'loss', 'multi_class'],
            bools=['dual', 'fit_intercept'])
        self.estimator = _LinearSVC(**out_params)

    @staticmethod
    def register_codecs():
        from codec.codecs import SimpleObjectCodec
        codecs_manager.add_codec('combined_feats_algos.LinearSVC', 'LinearSVC',
                                 SimpleObjectCodec)
        codecs_manager.add_codec('sklearn.svm.classes', 'LinearSVC', SimpleObjectCodec)
```

sklearn docs

<b>Parameters:</b>	<b>penalty</b> : string, 'l1' or 'l2'
	Specifies the regularization norm.
	The 'l1' leads to sparse solutions.
<b>loss</b> : string, 'hinge' or 'squared_hinge'	
	Specifies the loss function.
	While 'squared_hinge' is the standard loss function for SVC, 'hinge' is used by the linear SVM algorithm (LinearSVC).
<b>dual</b> : bool, (default: True)	
	Specifies whether to use the dual coordinate descent algorithm or the simple one-dimensional minimization algorithm.

# Custom Made

## NLP Example Custom Command

### ► cleantext command

```
| inputlookup peter_pan.csv | table sentence | cleantext  
textfield=sentence mv=f ngram_range=1-3 ngram_mix=t base_type=lemma_pos
```

### ► TFIDF ready

sentence	ngrams	pos_tag	pos_tuple
child except one grow	child except except one one grow child except one except one grow	NNS IN CD VB	["child", "NNS"] ["except", "IN"] ["one", "CD"] ["grow", "VB"]
soon know grow way wendy know	soon know know grow grow way way wendy wendy know soon know grow know grow way grow way wendy way wendy know	RB VBP VB NN NNP VBD	["soon", "RB"] ["know", "VBP"] ["grow", "VB"] ["way", "NN"] ["wendy", "NNP"] ["know", "VBD"]
one day two year old play garden pluck another flower ran	one day	CD NN CD NNS JJ VBG NN VBD	["one", "CD"]

**Counts**

Click in panels with (+) for further action

Time Range: Last 24 hours | Text Search (output must be tabular-i.e. ... | table text): inputlookup moby\_dick.csv

Text Field: sentence | Remove HTML:  Yes  No | Remove Stopwords (en):  Yes  No | Custom Stopwords (comma separated):

ngram Range: 2-3 | Using Grouping Field:  Yes  No | Grouping Field: chapter | Submit | Hide Filters

Total # Terms <b>102,984</b>	Total # Unique Terms <b>12,591</b>	Total # sentence <b>9,438</b>	Avg Terms Per sentence <b>10.9</b>
Total # ngrams <b>178,890</b>	Total # Unique ngrams <b>165,592</b>	Total # chapter <b>136</b>	Avg Terms Per chapter <b>757</b>

Top Parts-of-Speech Tags (+)

Top Terms (+)	Top ngrams
sentence	sperm whale
whale	white whale
one	old man
say	moby dick
like	captain ahab
upon	right whale
ship	captain peleg
man	aye aye
ahab	one hand
go	cry ahab
see	

Top Terms (proportion):

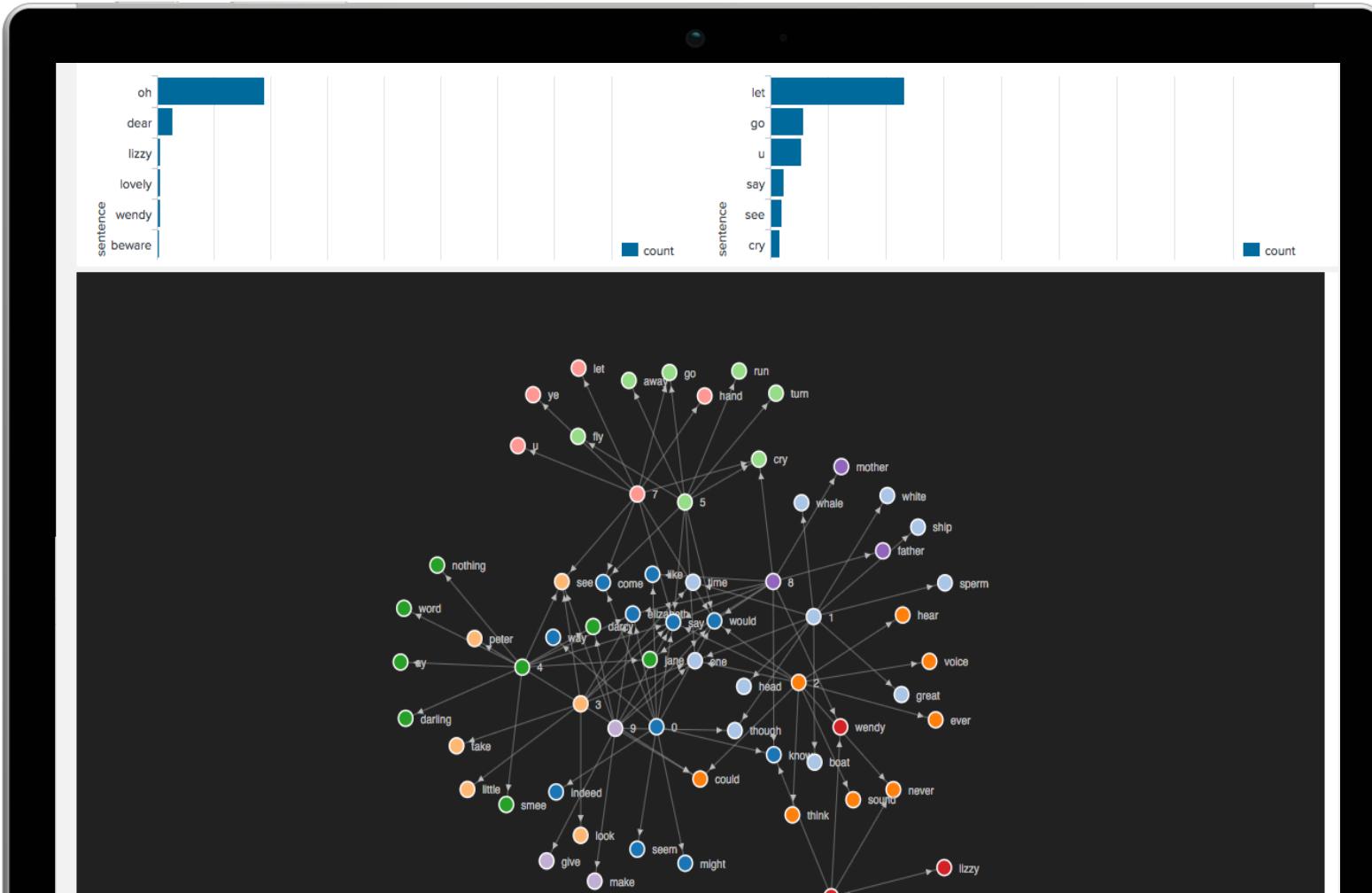
Term	Count	Proportion
sentence	1299	~0.0125
whale	918	~0.0087
one	618	~0.0058
say	577	~0.0054
like	560	~0.0052
upon	551	~0.0051
ship	507	~0.0048
man	506	~0.0048
ahab	480	~0.0045
go	460	~0.0043
see		

Top ngrams:

ngram	Count
sperm whale	187
white whale	107
old man	81
moby dick	80
captain ahab	64
right whale	51
captain peleg	33
aye aye	29
one hand	29
cry ahab	28

# Theme Park

## Themes Dashboard



► Unsupervised ML  
(Clustering like  
Kmeans)

- Can pass through topic modeling algos first

Welcome Counts Themes Sentiment Search

NLP Text Analytics

### Sentiment

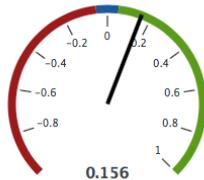
Time Range Text Search (output must be tabular-i.e ... | table text)

Last 24 hours | inputlookup pride...csv

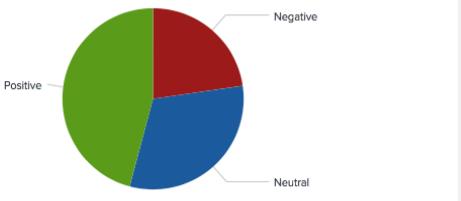
Text Field Neutral Zone Size

sentence Medium Submit Hide Filters

#### Average Sentiment

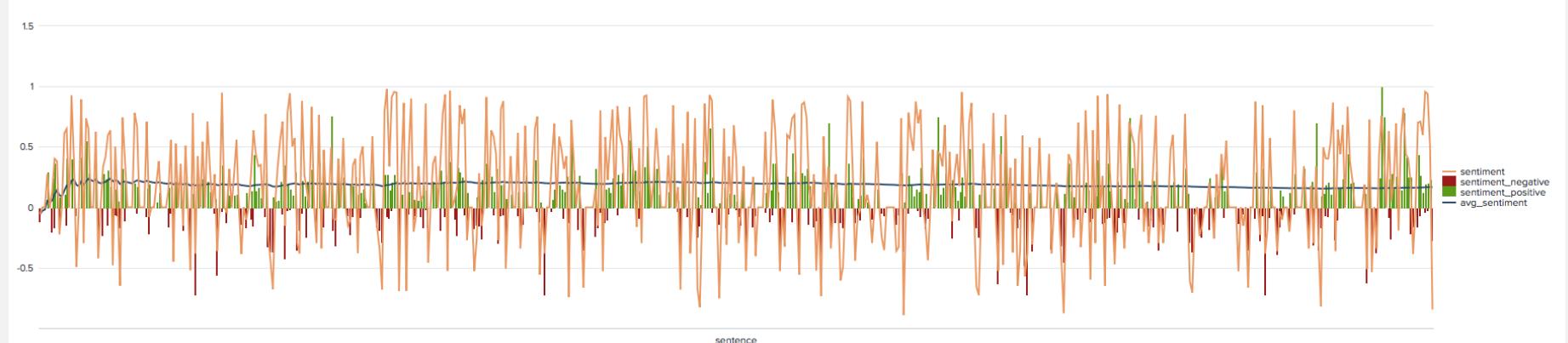


Positive  
0.156



#### Sentiment Scores Per sentence

Sample Ratio 0.1



# Don't Forget to Tip Your Waiter

Pro tip

- ▶ rejects or depends to hide options
- ▶ Use the `comment("")` macro to turn off portions of the search

```

<input id="horiz_two_radio_1" type="radio" token="html_dummy">
<label>Remove HTML</label>
<choice value="Yes">Yes</choice>
<choice value="No">No</choice>
<change>
  <condition value="Yes">
    <set token="html_begin"></set>
    <set token="html_end"></set>
  </condition>
  <condition value="No">
    <set token="html_begin">`comment("</set>
    <set token="html_end">")`</set>
  </condition>
</change>
<default>No</default>
<initialValue>No</initialValue>

```

```

<search id="base_search">
  <query>
    $master_search$
    $html_begin$1 bs4 textfield=$textfield$ $html_end$
```

# A Matter of Great Import

| fit <imported\_algo\_name\_here> target from features

- ▶ Algorithms imported by NLP
- ▶ Topic Modeling Algos
  - TruncatedSVD (aka LSA), LatentDirichletAllocation, NMF
- ▶ Combined Features Algos
  - MinMaxScaler, LinearSVC, ExtraTreesClassifier, TFBinary
- ▶ Shared with Splunk's MLTK GitHub Collaboration

# A Perfect Fit

## Binary model building

- ▶ Run search, then mailparser

```
<email search> | mailparser
```

- ▶ Under-sample majority

```
| sample 400 by target
```

- ▶ Clean the text

```
| cleantext  
textfield=mail_text mv=f
```

- ▶ Change continuous in to interval

```
| rangemap  
field=num_repeat_url  
qt01=0-1 qt02=2-3  
default=highest | rename  
range AS num_repeat_url
```

- ▶ One-hot encoding numerical categories

```
| foreach date-* [ eval  
<<MATCHSTR>>_{{<<FIELD>>}} = 1  
] | fields - date_hour  
date_wday | fillnull
```

- ▶ Binarize text

```
| fit TFBinary  
max_features=9000  
ngram_range=1-3 min_df=2  
into tfbin
```

- ▶ Scale and encode

```
| fit MinMaxScaler * into  
mms | fields - MMS_target*
```

- ▶ Split train/test sets (80/20)

```
| sample partitions=10  
seed=1234 | search  
partition_number < 8
```

- ▶ Fit to models

```
| fit LinearSVC target  
from MMS_* into svc  
| fit  
ExtraTreesClassifier  
target from MMS_* into et  
| fit MLPClassifier  
target from MMS_* into nn
```

# All In Favor?

# VotingClassifier

- ▶ Not everything importable
  - ▶ API Limitations

```
<repeat preprocessing> | apply tfbin  
| apply mms | sample partitions=10 seed=1234  
| search partition_number > 7  
| apply svc as v1 | apply et as v2 | apply nn  
as v3 | eval vote =  
case(v1=='v2',v1,v1=='v3',v1,true(),v2)
```



target	v1	v2	v3	vote
investigate	investigate	investigate	investigate	investigate
investigate	investigate	investigate	investigate	investigate
investigate	investigate	investigate	investigate	investigate
investigate	investigate	investigate	investigate	investigate
investigate	investigate	ignore	investigate	investigate
investigate	investigate	investigate	ignore	investigate

# Afraid of Commitment

## Me too

- ▶ Not a “real” developer?
    - Don’t let that stop you, it didn’t for me
  - ▶ Need help, just ask
    - Nerds like to be heros



<https://dont-be-afraid-to-commit.readthedocs.io/en/latest/git/git.html>

# If I Could Turn Back Time

# **Had the Collaboration Repo Existed**

- ▶ Less time struggling
  - ▶ More time doing



# I Hope You Brought Enough to Share

# MORE is BETTER!

- ▶ Stand on shoulders of giants
  - ▶ Single source for MLTK expansion
  - ▶ Pretty please



# Branching Out

# Tips for importing new algos

- ▶ README.md
  - ▶ Find similar algorithm
    - from sklearn.*class*
  - ▶ Codecs are tricky
    - Use Python's dir() function

```
$SPLUNK_HOME/bin/splunk cmd python  
> import sklearn.classname as lookhere  
> dir(lookhere)
```



# Q&A

# Thank You

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

