

# Jupyter Notebooks: Interactive Visualization Approaches

**QCon 2018  
San Francisco  
November 07, 2018**

**Chakri Cherukuri  
Senior Researcher  
Quantitative Financial Research Group**

**TechAtBloomberg.com**

**Bloomberg**

# Outline

- Introduction
- Overview of interactive widgets
- Example with code walk-through
  - Normal Distribution
- Case studies
  - Server log dashboard
  - Twitter sentiment analysis
  - Tools for deep learning
- Q & A

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Interactive widgets

## Python Object

```
IntSlider(description='slider,  
value=50)
```

## Visual Representation



- Interface the user sees
- Its attributes (*traits*) can send events when updated

- Graphics, Interaction
- JavaScript implementation hidden from the user

# Interactive widgets

- **ipywidgets**: core UI controls (text boxes, sliders, button etc.)
- **bqplot**: 2D plotting widgets (built on top of the ipywidgets framework)
- **Key takeaways:**
  - **ipywidgets** and **bqplot** can be combined to build interactive visualizations
  - Attributes of widgets (traits) can be linked using callbacks

# Server Logs Dashboard

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

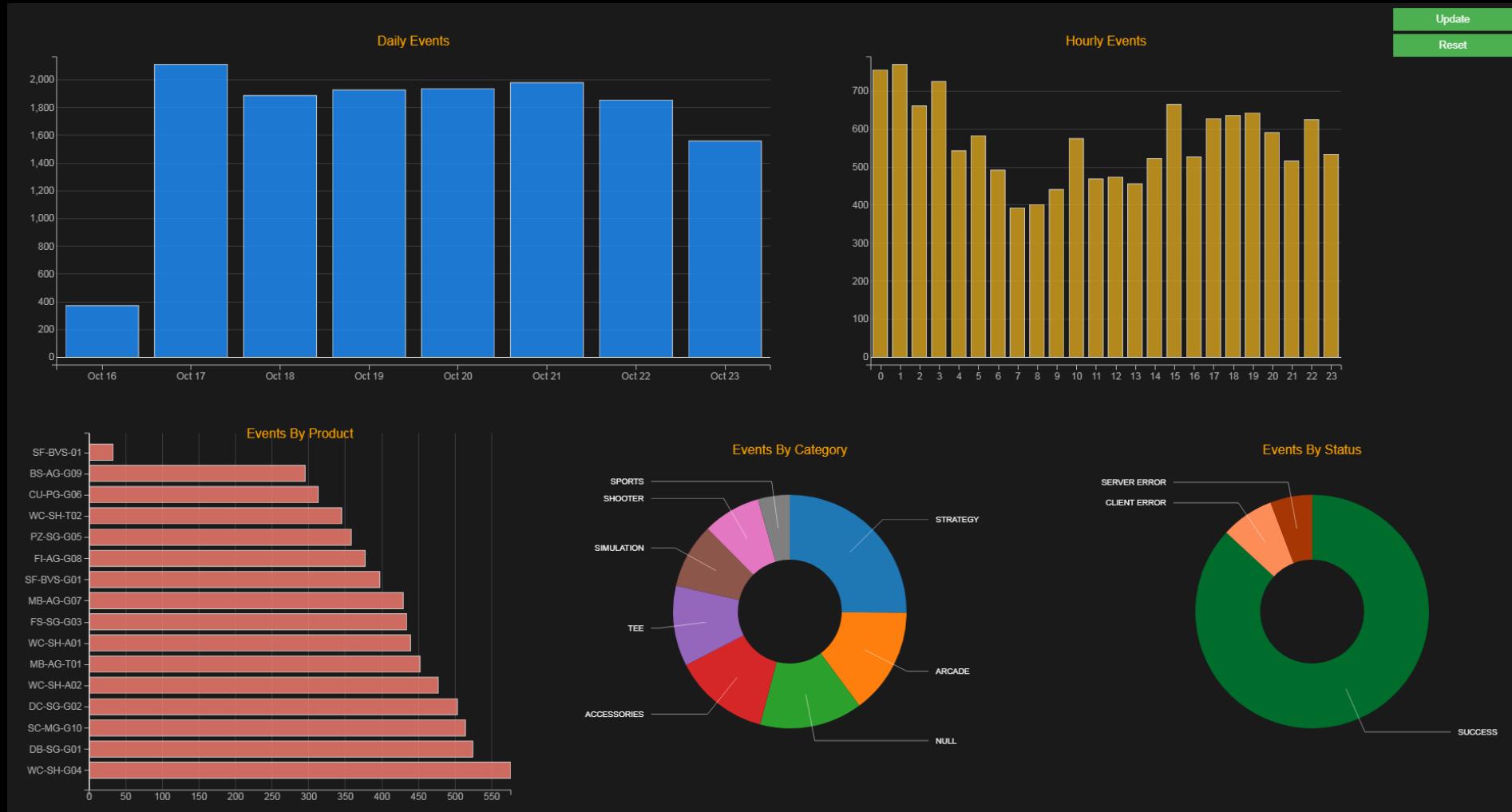
# Server Logs

- Information we can glean:
  - Timestamp of requests/events
  - IP address
  - HTTP status codes
  - Agent
  - URL
- URLs can be parsed to obtain search parameters:
  - Product IDs
  - Category IDs

# Dashboard Components

- Time series plots of events
  - Trends
  - Seasonality
  - Outliers
- Daily/hourly aggregation of events
- Plots of events broken down by
  - Products
  - Categories
  - Status codes

# Server log dashboard



TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Performance/Scaling

- Data ingestion from streaming sources like **Kafka**
- Notebook servers running on spark/hadoop clusters
- Streaming data frames
  - dask dataframes
  - pandas with chunking

# Twitter Sentiment Analysis

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# News/Twitter Sentiment

- Social sentiment from the raw news story or tweet. Data is:
  - Unstructured
  - Highly time-sensitive
- Story-level sentiment
- Company-level sentiment
- Sentiment score can be used as a trading signal

# Twitter Sentiment Classification

## Problem statement:

Predict the sentiment (negative, neutral, positive) of a tweet for a company

Ex: “\$CTIC Rated strong buy by three WS analysts. Increased target from \$5 to \$8.” = Positive

Three way classification problem

- Input: raw tweets
- Output: sentiment label  $\Sigma$  {negative, neutral, positive}

# Methodology

- We are given **labeled** training and test data sets
- Train classifier on training data set
- Predict labels on test data and evaluate performance

# One vs. rest Logistic Regression

- Train three binary classifiers for each label
  - Model 1: Negative vs. Not Negative
  - Model 2: Positive vs. Not Positive
  - Model 3: Neutral vs. Not Neutral
- Get probabilities (measures of confidence) for each label
- Output the label associated with the highest probability

# Classifier performance analysis

- Look at misclassifications
  - Confusion Matrix
- Understand model predicted probabilities
  - Triangle visualization
- Fix data issues

# Confusion Matrix

K x K matrix where K = number of classes

Cell[i, j] = number of samples whose:

Actual label = i

Predicted label = j

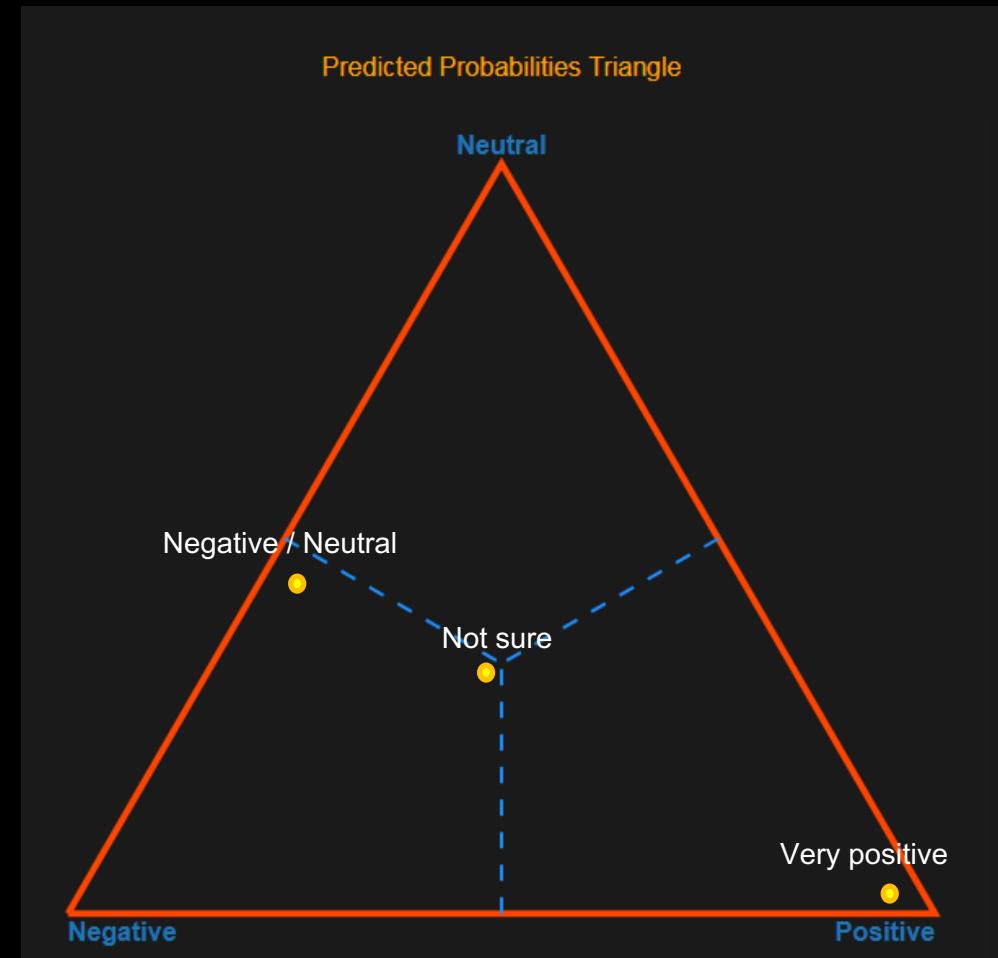
Diagonal entries: correct predictions

Off diagonal entries: misclassifications

Confusion Matrix			
Actual	Predicted		
	Negative	Neutral	Positive
Negative	2,064	628	103
Neutral	686	11,095	766
Positive	95	777	2,183

# Triangle Visualization

- Model returns 3 probabilities (which sum to 1)
- How can we visualize these 3 numbers?
  - Points inside an equilateral triangle



# Performance analysis dashboard

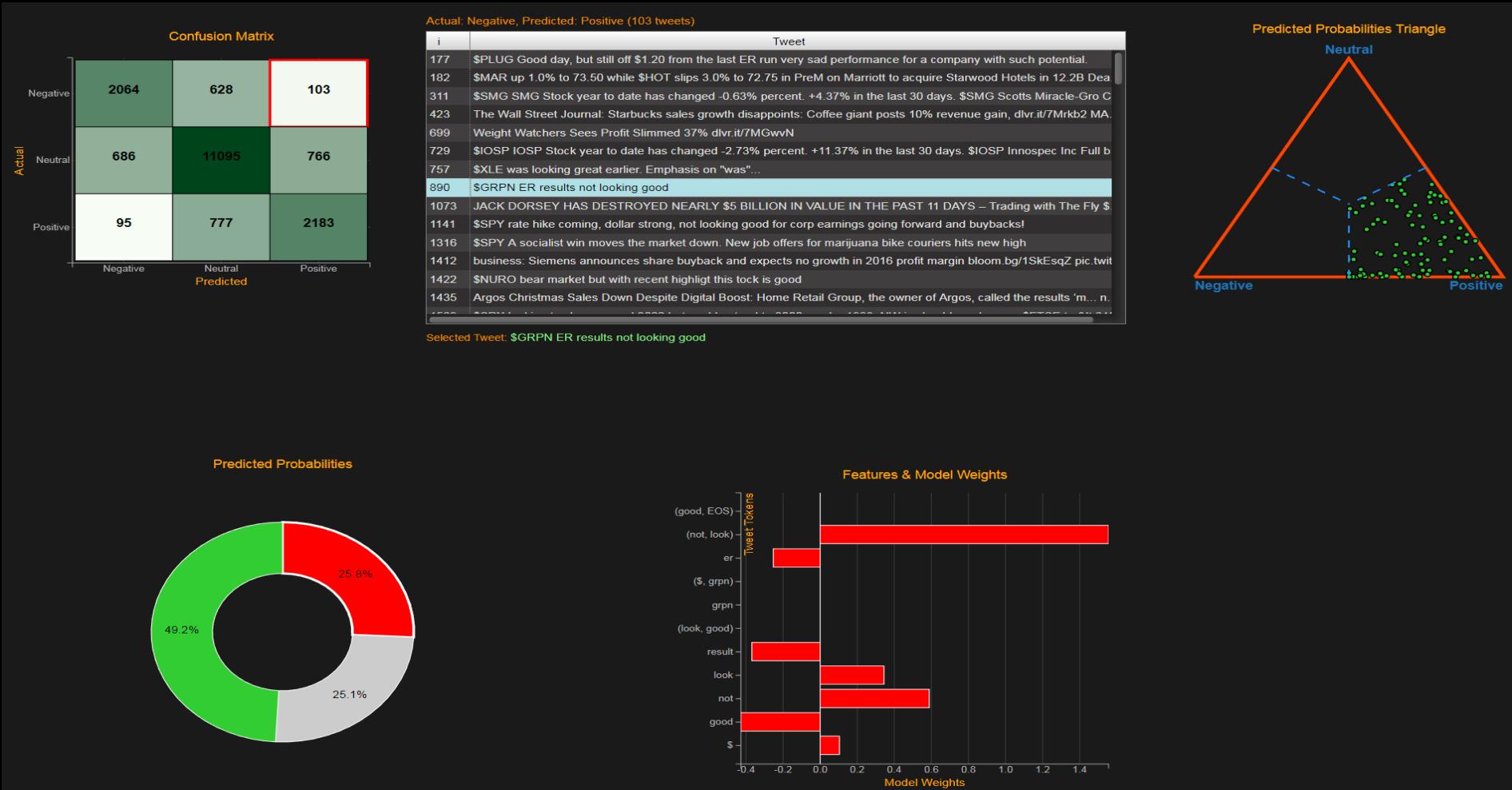
Use the dashboard to:

- Analyze misclassifications (using confusion matrix)
- Improve model by adding more features (by looking at model coefficients)
- Fix data issues (using triangle and lasso)

# Analyze Misclassifications



# Analyze Misclassifications

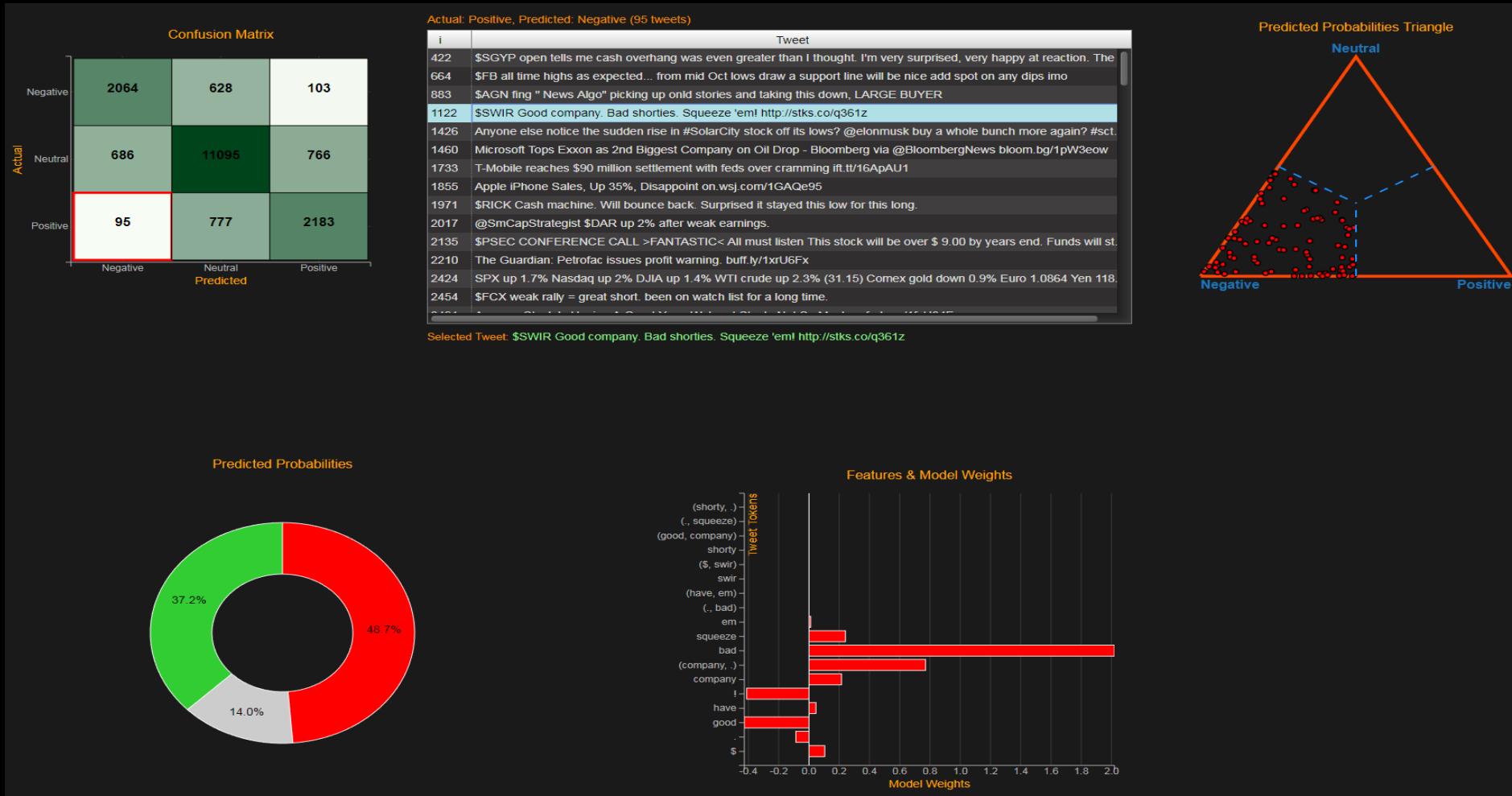


TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Analyze Misclassifications



TechAtBloomberg.com

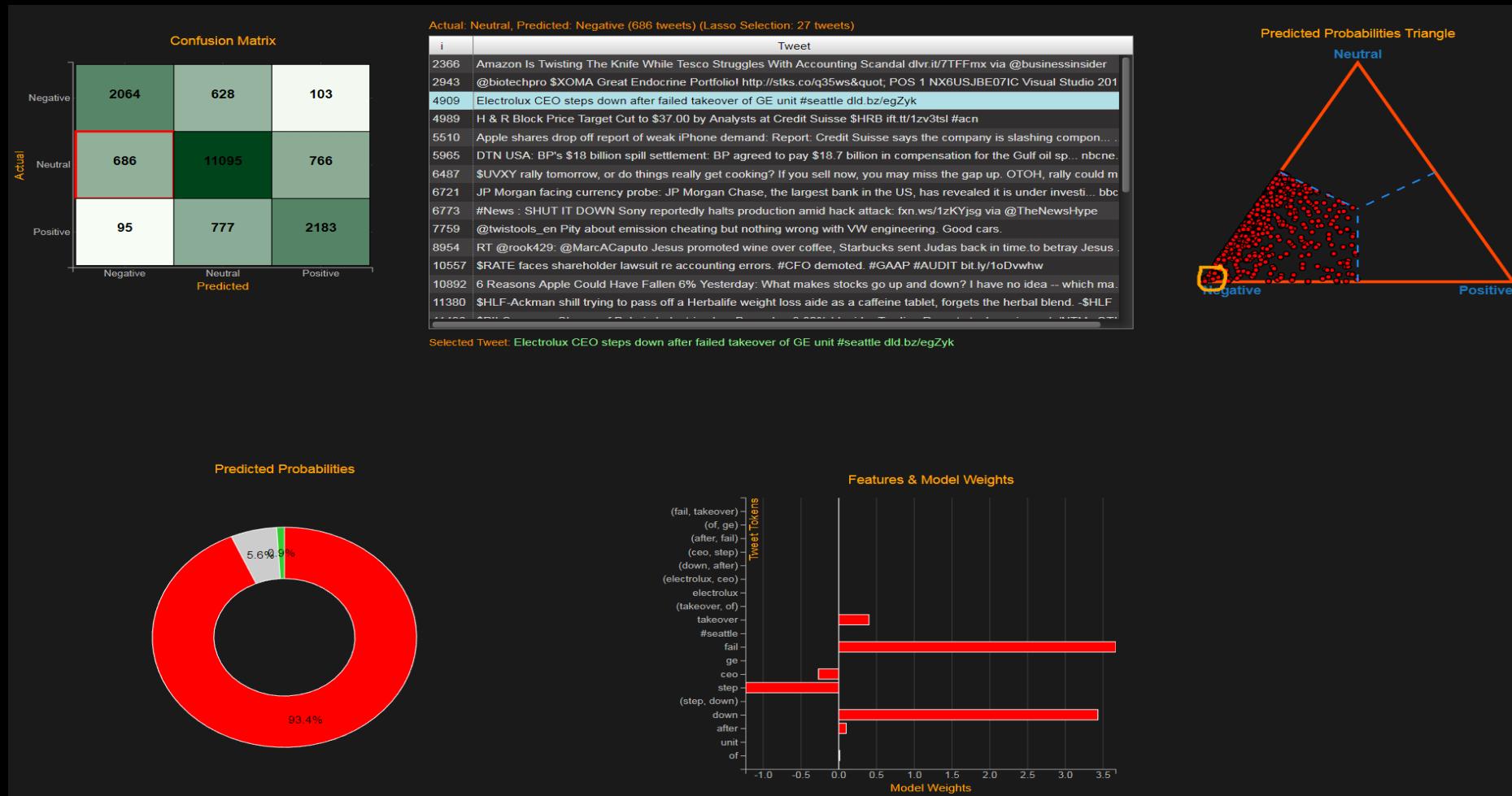
© 2018 Bloomberg Finance L.P. All rights reserved.

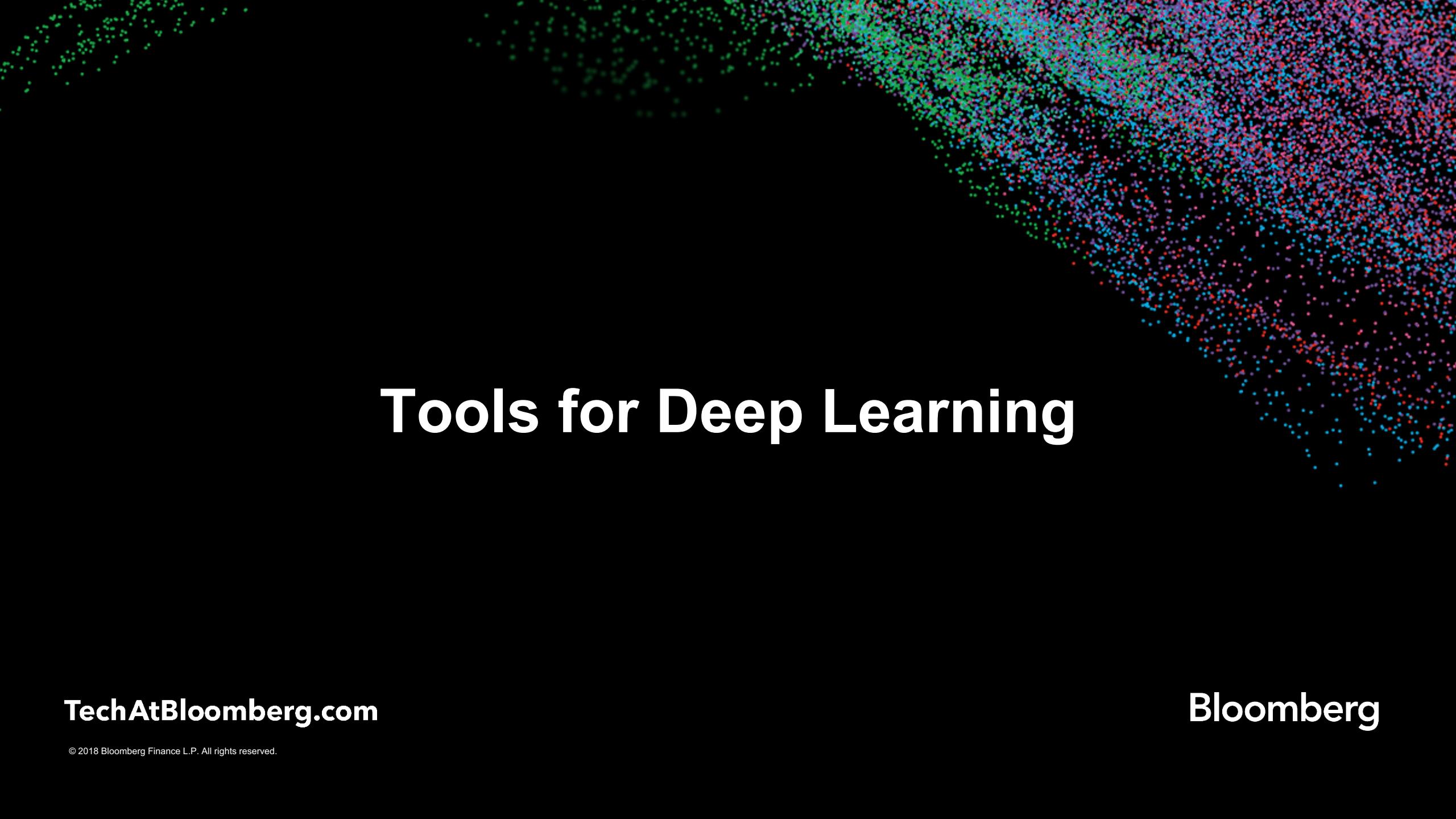
Bloomberg

# Use Lasso To Find Data Issues



# Use Lasso To Find Data Issues





# Tools for Deep Learning

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Deep Learning Tools

## Graphical Wizard

- Select network parameters
- Build network architecture
- Training plots
  - Real time loss/accuracy curves
  - Distribution of weights/biases/activations
- Diagnostic plots
  - Residual vs. Predicted Values
  - Confusion Matrix

# Network Parameters

Network Parameters    Architecture    Training    Diagnostics

Epochs: 100  
Batch Size: 64  
Loss: mse  
Optimizer: adam

optimizer params

lr	0.001
beta_1	0.9
beta_2	0.999
epsilon	1e-8
decay	0

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Network Architecture

Network Parameters		Architecture		Training		Diagnostics	
<b>Hidden Layers</b> <span style="color: green;">+</span> <span style="color: red;">-</span>							
	Inputs	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Outputs	
<b>Nodes</b>	4	100	60	30	10	1	
<b>Activation</b>	relu	relu	relu	relu	relu	linear	
<b>Batch Norm</b>	■	■	■	■	■		
<b>Dropout prob</b>	0	0	0	0	0		

TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Loss/Accuracy Curves



TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Distributions of Weights/Biases/Activations

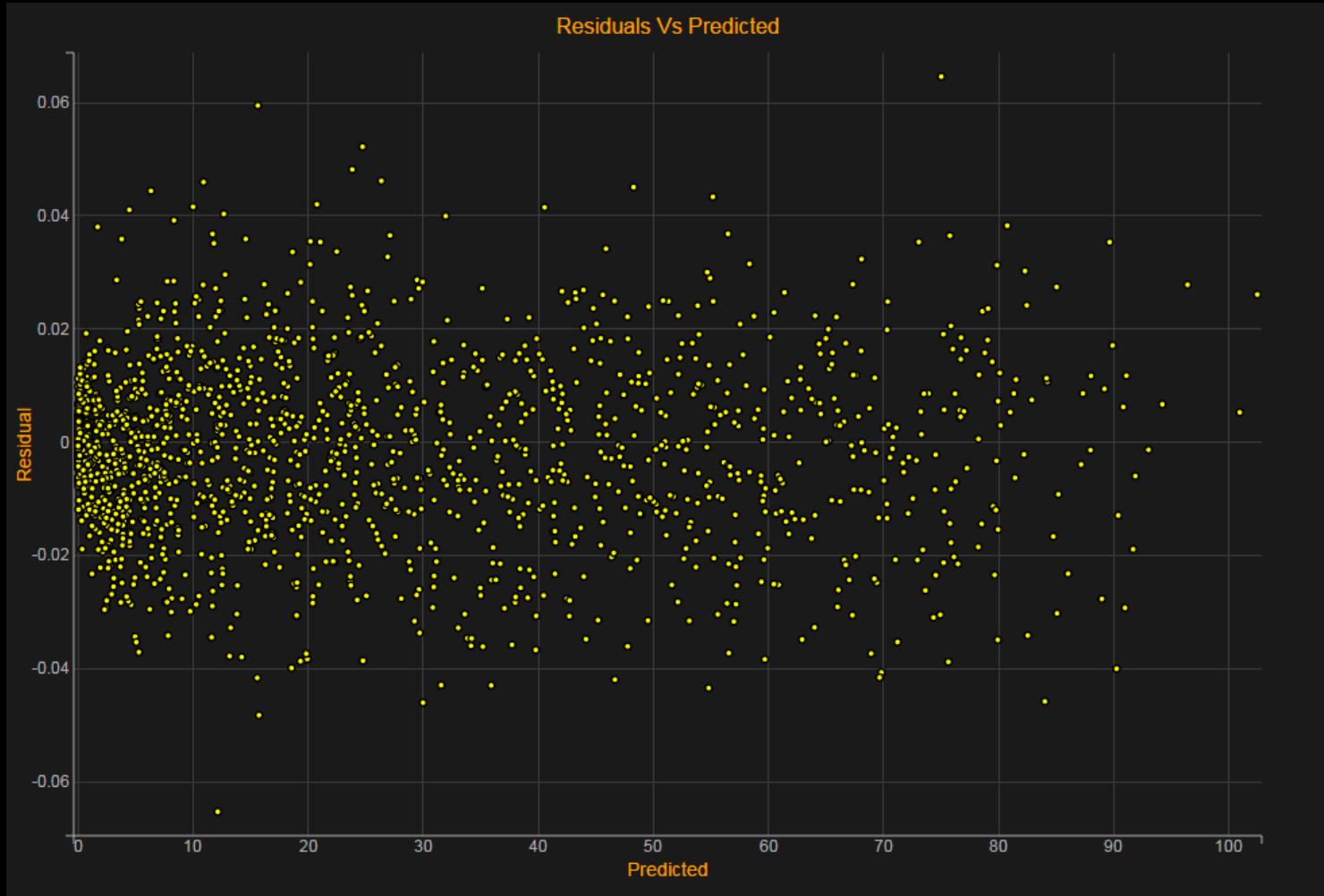


TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Diagnostic Plots



TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.

Bloomberg

# Resources

- Widget libraries used to build the applications:  
**ipywidgets**: <https://github.com/jupyter-widgets/ipywidgets>  
**bqplot**: <https://github.com/bloomberg/bqplot>
- Machine Learning libraries  
**scikit-learn**: <https://http://scikit-learn.org>  
**tensorflow**: <https://www.tensorflow.org>  
**keras**: <https://keras.io>
- Link to the notebooks/code: [https://github.com/ChakriCherukuri/qcon\\_2018](https://github.com/ChakriCherukuri/qcon_2018)
- Tech At Bloomberg blog: [www.TechAtBloomberg.com](http://www.TechAtBloomberg.com)