

USD INDEX TRADE RECOMMENDER SYSTEM

KEVIN GLYNN (MENTOR)

SRINIVAS KRISHNASWAMY (STUDENT)

WHAT IS THE DOLLAR INDEX ?

IT IS A WEIGHTED GEOMETRIC MEAN OF THE UNITED STATES DOLLAR (USD)'S VALUE RELATIVE TO FOLLOWING SELECT CURRENCIES

- EURO (EUR), 57.6%
- JAPANESE YEN (JPY), 13.6%
- POUND STERLING (GBP), 11.9%
- CANADIAN DOLLAR (CAD), 9.1%
- SWEDISH KRONA (SEK), 4.2%
- SWISS FRANC (CHF) 3.6%

Established in 1973 with a value of 100. If it is above 100, USD is stronger than 1973, if below 100 then weaker than 1973 !

FACTORS INFLUENCING THE U.S. DOLLAR

1. DAILY PRICES OF THE U.S.DOLLAR INDEX
2. FEDERAL RESERVE FUND RATE
3. OIL PRICES
4. PERSONAL CONSUMPTION
5. CONSUMER PRICES
6. HOME PRICES
7. INFLATION
8. STOCK MARKET PRICES
9. SOCIAL SECURITY
10. EMPLOYMENT NUMBERS
11. PAYROLL DATA
12. EMPLOYMENT GROWTH
13. FEDERAL RETIREMENT BENEFITS
14. SAVING RATE OF THE RESIDENTS OF THE U.S.
15. HOME CONSTRUCTION
16. RETURN ON INVESTMENT (ROI) ON THE SNP

AND MANY, MANY MORE NOT LISTED HERE

WHO CARES AND WHY ?

TRADERS WHO TRADE IN BOND, CURRENCY AND GOLD MARKETS WITHIN THE WORLD'S CAPITAL MARKET FINANCIAL SYSTEM.

THE US DOLLAR IS THE WORLD'S RESERVE CURRENCY AND AS SUCH ANY CHANGE IN IT'S VALUE CAN AFFECT HUMANS WORLDWIDE.

HOW CAN MACHINE LEARNING BE USED ?

IF WE CAN BUILD A SYSTEM TO PREDICT THE NEXT DAY'S CLOSING PRICE CORRECTLY, WE CAN MAKE A TRADE - I.E. BUY / SELL THE DOLLAR INDEX IF PREDICTION SAYS IT IS GOING UP (SELL / BUY IF THE DOLLAR INDEX GOES DOWN)

SO LET'S BUILD ONE ...

HERE ARE THE STEPS –

1. ACQUIRE DATA FROM WEBSITES THAT PROVIDE FINANCIAL DATA – QUANDL AND INVESTING.COM
2. SPLIT THE DATA INTO TRAIN/TEST
3. TRAIN MODEL ON THE TRAINING DATASET
4. TEST THE MODEL ON THE TESTING DATASET
5. EVALUATE THE TEST DATASET
6. OPERATIONALIZE THE ML PIPELINE, EITHER ON A LOCAL COMPUTER OR ON THE CLOUD
7. USAGE -
 - PASSING IN THE INPUT DATASET (FEATURES) SHOULD RETURN THE OUTPUT PRICE (LABEL). THE LABEL IN OUR CASE IS THE PERCENTAGE GAIN OR LOSS
 - IF THE PERCENTAGE GAIN IS WITHIN 1.5% +/-, THEN THE USER PLACES A TRADE FOR THE NEXT DAY.
 - CLOSE OUT THE POSITION BY THE CLOSE OF BUSINESS THE NEXT DAY (SINCE WE ARE USING DAILY CLOSE PRICES AS THE LABEL)

LOCAL OPERATIONALIZATION

TURI CREATE



COURSERA STUDENTS TURI CREATE

Simple development of custom machine learning models

Build intelligent applications end-to-end in Python

```
1 import graphlab
2
3 #1. Load and explore data
4 sf = graphlab.SFrame('https://s3.amazonaws.com/datasets/reviews')
5
6 # 2. Train recommender model
7 model = graphlab.toolkits.recommender.create(data, item_id = 'name')
8 model.get_similar_items(['Coffee House'], k=5)
9
10 # 3. Deploy model as an intelligent service
11 my_deployment = graphlab.deploy.predictive_service.load('s3://rec_ps')
12 my_deployment.show()
13
14 def get_outreach_recs(rest_name, num_recs = 5):
15     return model.get_similar_items([rest_name], k=num_recs)
16
17 my_deployment.add('get_or_recs', get_outreach_recs)
18 my_deployment.apply_changes()
19 my_deployment.query('get_or_recs', rest_name='Best Coffee Shop')
```

CLOUD
OPERATIONALIZATION

MICROSOFT
AZURE ML



WHICH ML MODEL SHOULD WE PICK ?

TURI CREATE ANALYZES THE DATASET AND PICKS THE BEST MODEL – IN THIS CASE GRADIENT BOOSTED DECISION TREES (GBMs).

IN AZURE ML 3 MODELS WERE USED, BOOSTED TREES, NEURAL NETS AND LINEAR REGRESSION. THE RESULTS WERE THEN COMPARED.

The screenshot shows the Turi Create interface on a laptop screen. The title bar says "turi create intelligence". Below it, "COURSERA STUDENTS" and "TURI CREATE" are visible. The main area has a teal header with the text "Simple development of custom machine learning models" and a white footer with the text "Build intelligent applications end-to-end in Python". A code editor window displays the following Python script:

```
1 import graphlab
2
3 # 1. Load and explore data
4 df = graphlab.SFrame('https://s3.amazonaws.com/datasets/reviews')
5
6 # 2. Train recommender model
7 model = graphlab.toolkits.recommender.create(data, item_id = 'name')
8 model.get_similar_items("Coffee House", k=5)
9
10 # 3. Deploy model as an intelligent service
11 my_deployment = graphlab.deploy.predictive_service.load("s3://reco/pw")
12 my_deployment.show()
13
14 def get_outreach_recs(item_name, num_recs = 5):
15     return model.get_similar_items(item_name, k=num_recs)
16
17 my_deployment.add('get_outreach', get_outreach_recs)
18 my_deployment.apply_changes()
19 my_deployment.query('get_outreach', item_name='Best Coffee Shop')
```

Below the code editor is a large blue flask icon containing a green liquid with three bubbles, symbolizing data or a process.

LOCAL OPERATIONALIZATION

TURI CREATE (GRAPH LAB)

Building a Machine Learning Pipeline in GraphLab

```
In [41]: def trade_workflow(path):

    # Get Data
    data = create_trade_sFrame(path)

    # Make a train-test split
    train_data, test_data = data.random_split(0.8)

    # Train model.
    model = train_trade_model(train_data)

    # Make recommendations
    results, sfTrade = recommend_trades(model, test_data)

    print(results)

    # Return the SFrame of recommendations.

    return sfTrade
```

Creating and starting a job in GraphLab (on the Local machine)

```
In [42]: import graphlab as gl

# Deploy the job locally
job_local = gl.deploy.job.create(trade_workflow, path = 'D:/SpringBoard/Capstone2/Data/US Dollar Index Historical Data_17Yr.csv')

[INFO] graphlab.deploy.job: Validating job.
[INFO] graphlab.deploy.job: Creating a LocalAsync environment called 'async'.
[INFO] graphlab.deploy.job: Validation complete. Job: 'trade_workflow-Sep-16-2018-11-39-07' ready for execution.
[INFO] graphlab.deploy.job: Job: 'trade_workflow-Sep-16-2018-11-39-07' scheduled.
```

Getting the status of a job in GraphLab (on the Local machine)

```
In [46]: # get status immediately after creating this job.
job_local.get_status()

Out[46]: u'Completed'
```

```
In [29]: model
```

```
Out[29]: Class : BoostedTreesRegression
```

Schema

```
-----  
Number of examples : 2585  
Number of feature columns : 55  
Number of unpacked features : 55
```

Settings

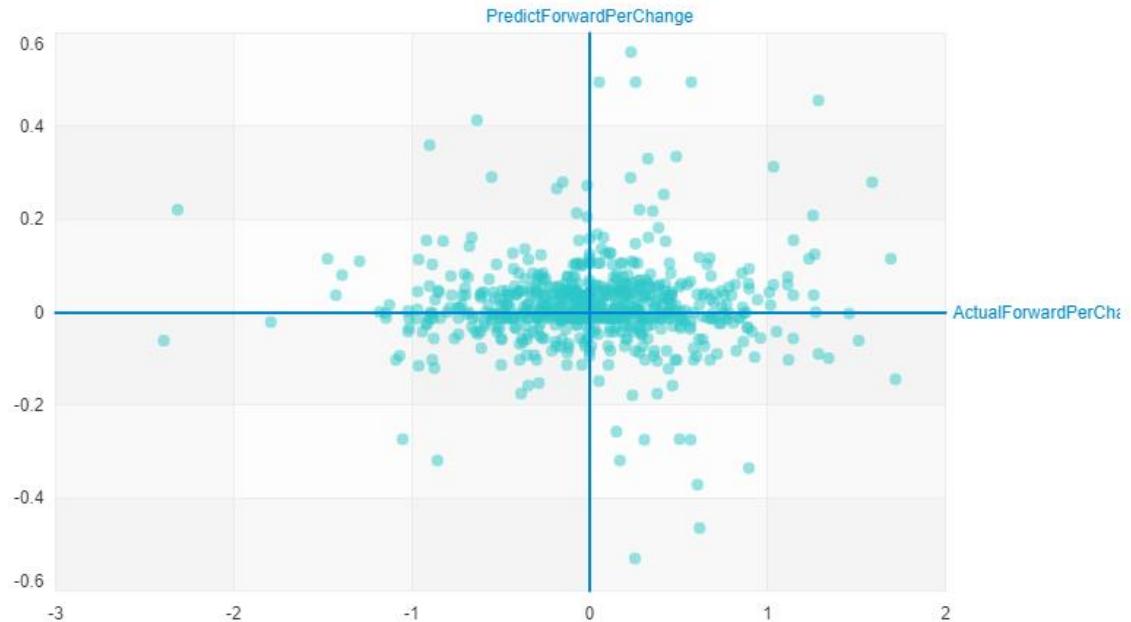
```
-----  
Number of trees : 10  
Max tree depth : 6  
Training time (sec) : 0.0469  
Training rmse : 0.5994  
Validation rmse : 0.4986  
Training max_error : 19.7859  
Validation max_error : 2.2403
```

Make predictions and evaluate the model ...

```
In [30]: #Make predictions and evaluate results.  
predictions = model.predict(test_data)  
results = model.evaluate(test_data)  
print(results)
```

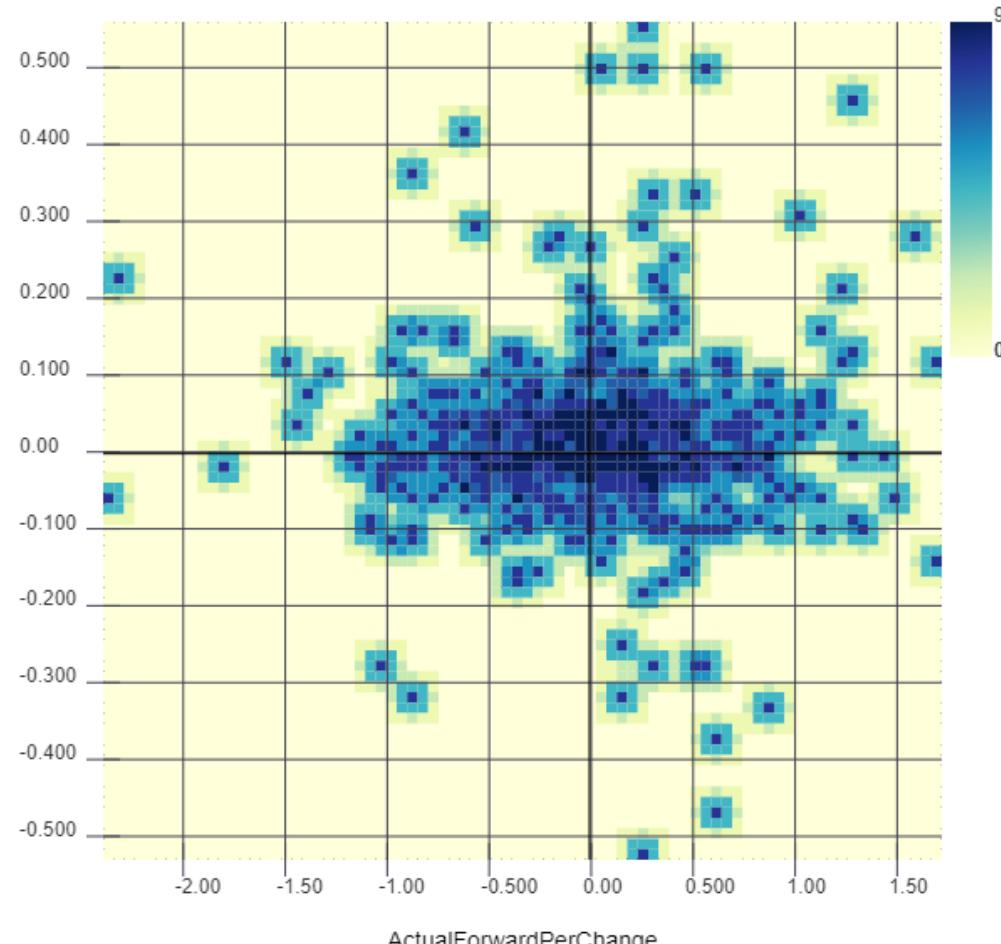
```
{'max_error': 2.534599542617798, 'rmse': 0.5210959975858873}
```

TURI CREATE



PredictForwardPerChange

point density



Heat Scaling Function

Logarithmic ▾

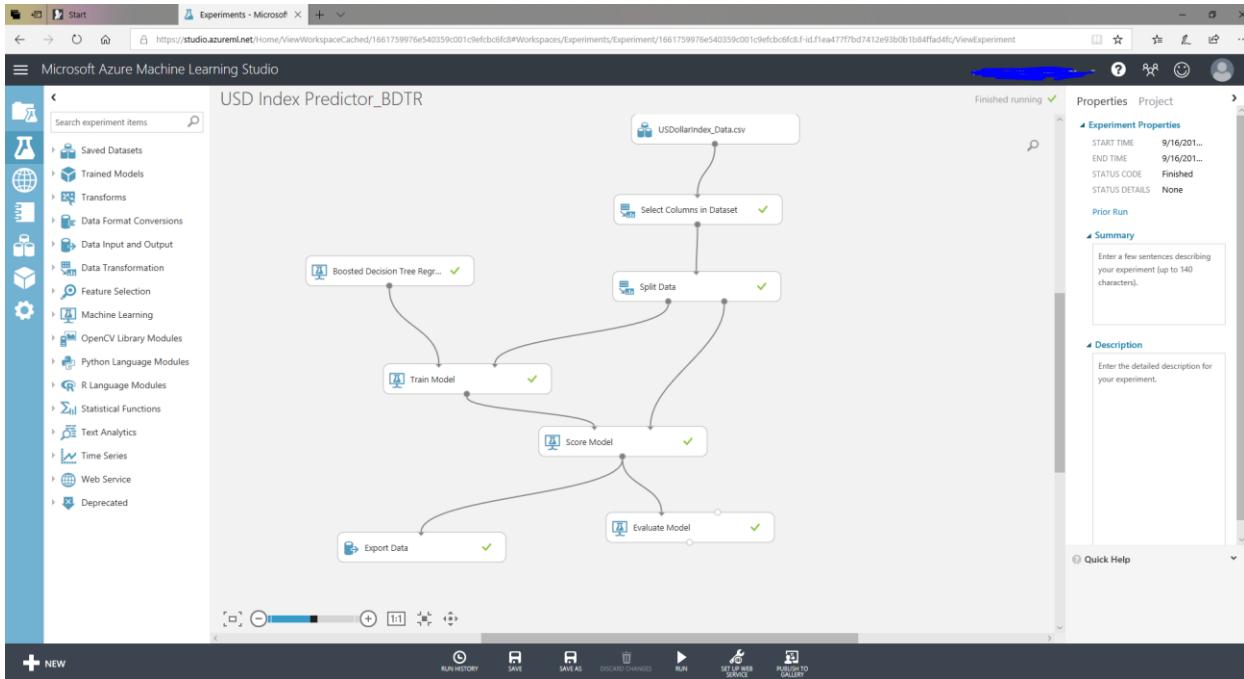
Hide Gridlines

TURI CREATE

BLUR 0

1

100



CLOUD OPERATIONALIZATION

AZURE ML – BOOSTED DECISION TREES REGRESSION

Microsoft Azure Machine Learning Studio

USD Index Predictor_BDTR

Finished running ✓ Properties Project

Search experiments

Saved

Trained

Transformed

Data In

Data Out

Data Flow

Features

Machines

Open

Python

R Language

Statistics

Text Analytics

Time Series

Web Services

Deployment

USD Index Predictor_BDTR > Evaluate Model > Evaluation results

Metrics

Mean Absolute Error	0.422229
Root Mean Squared Error	0.677442
Relative Absolute Error	1.164972
Relative Squared Error	1.949284
Coefficient of Determination	-0.949284

Error Histogram

frequency

Error

0.0035 0.68 1.3 2.0 2.1 3.4 4.0 4.1 5.4 6.0 6.1

Run History

Save

Save As

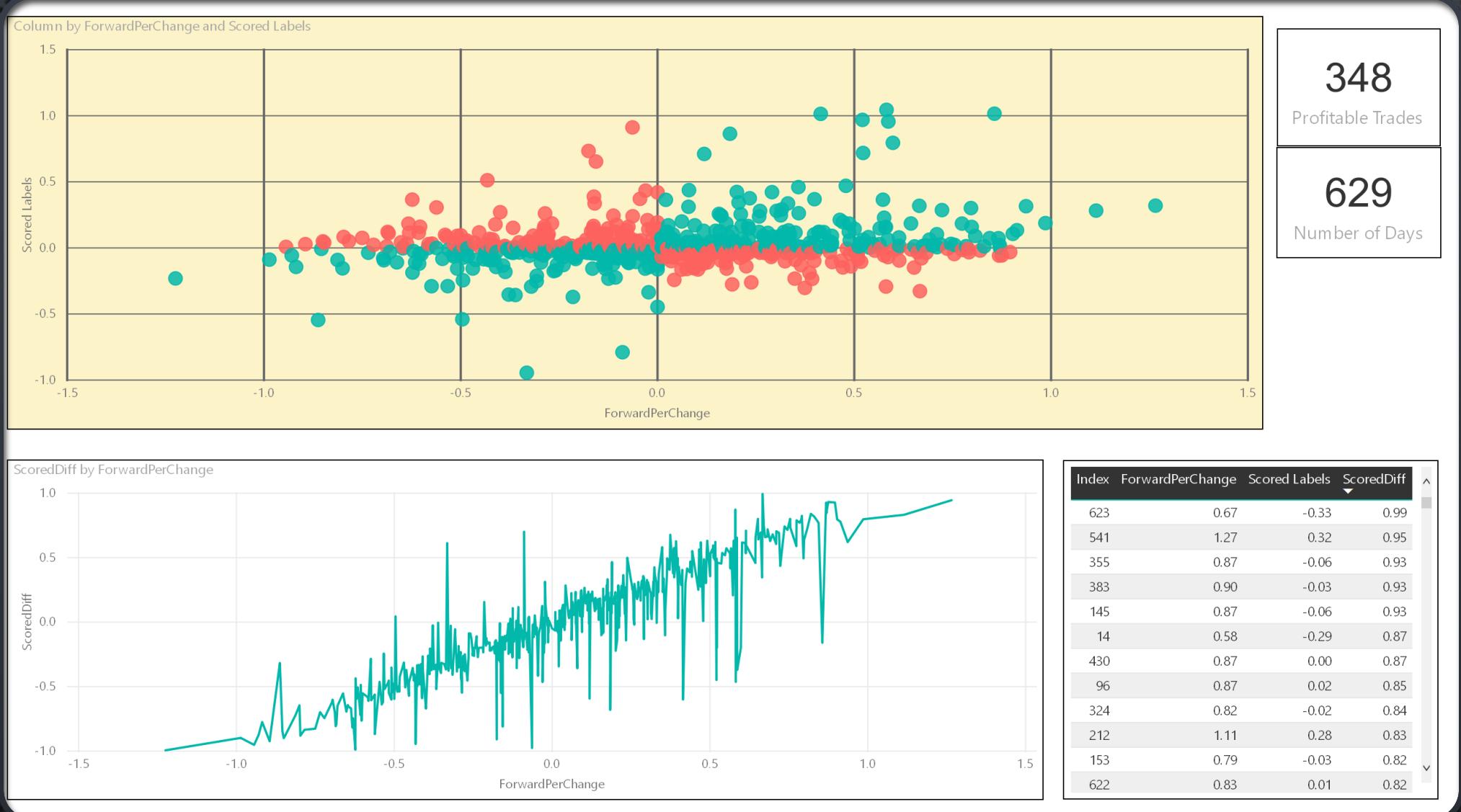
Discard Changes

Run

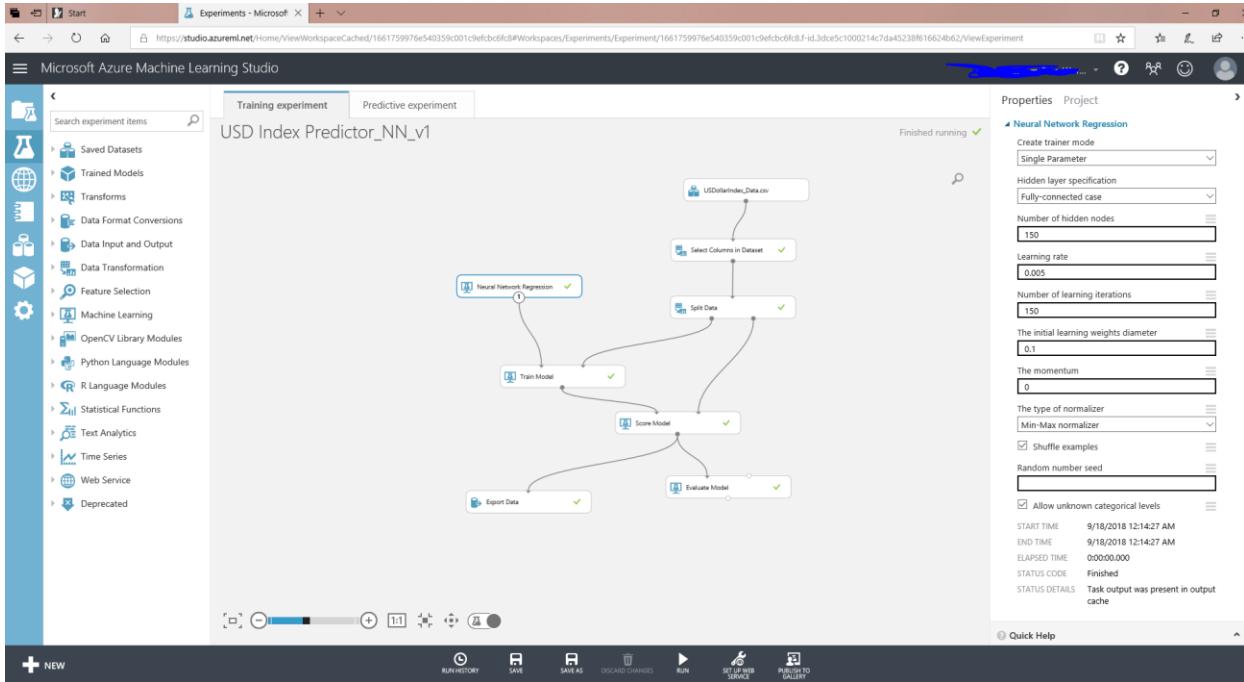
Set Up Web Service

Publish to Gallery

NEW



BOOSTED DECISION
TREES REGRESSION



CLOUD OPERATIONALIZATION

AZURE ML – NEURAL NETWORK REGRESSION

Experiments - Microsoft

https://studio.azureml.net/Home/ViewWorkspaceCached/1661759976e540359c001c9efcbc6fc8#Workspaces/Experiments/Experiment/1661759976e540359c001c9efcbc6fc8.f-id.3dce5c1000214c7da45238f616624b62/ViewExperiment

Microsoft Azure Machine Learning Studio

Training experiment Predictive experiment Properties Project

USD Index Predictor_NN_v1 > Evaluate Model > Evaluation results

Metrics

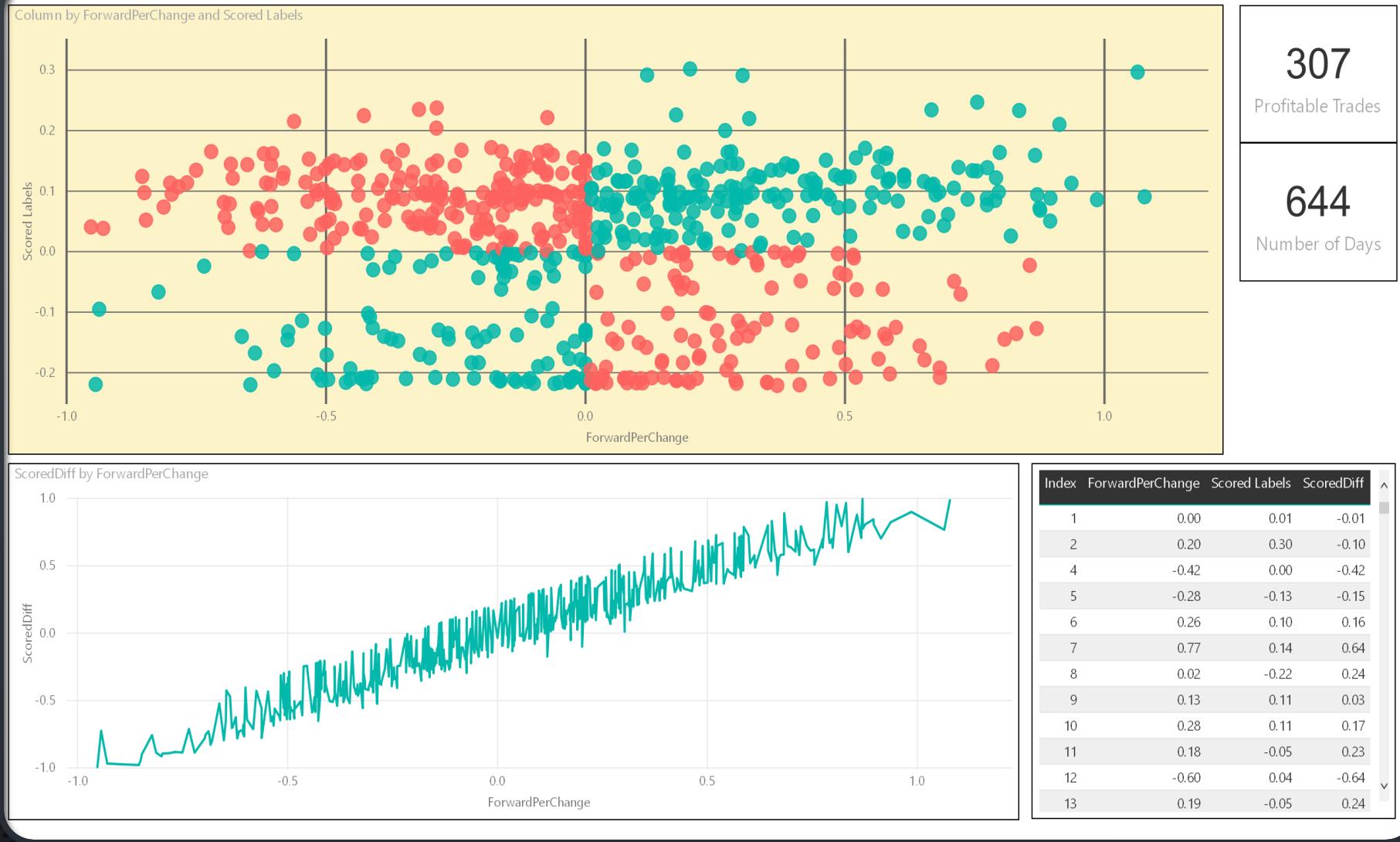
Mean Absolute Error	0.381917
Root Mean Squared Error	0.502182
Relative Absolute Error	1.053748
Relative Squared Error	1.071158
Coefficient of Determination	-0.071158

Error Histogram

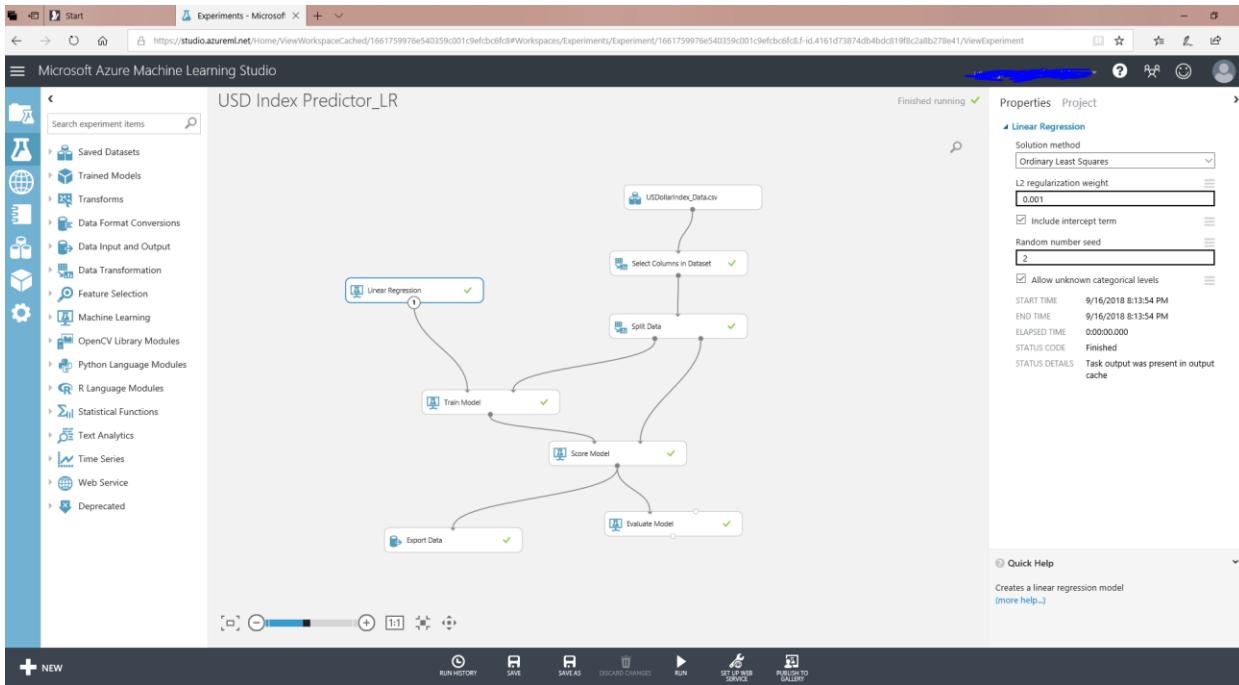
The histogram displays the distribution of errors. The x-axis is labeled 'Error' and ranges from 0.000016 to 2.1. The y-axis is labeled 'frequency' and ranges from 0 to 240. The distribution is highly right-skewed, with the highest frequency in the first bin (0.000016 to 0.21), followed by the second bin (0.21 to 0.42).

Error Range	Frequency
0.000016 - 0.21	240
0.21 - 0.42	200
0.42 - 0.63	110
0.63 - 0.84	70
0.84 - 1.05	30
1.05 - 1.26	20
1.26 - 1.47	10
1.47 - 1.68	5
1.68 - 1.89	3
1.89 - 2.1	2

Run History Save Save As Discard Changes Run Set Up Web Service Publish To Gallery



NEURAL NETS REGRESSION



CLOUD OPERATIONALIZATION

AZURE ML – LINEAR REGRESSION

Start Experiments - Microsoft | +

https://studio.azureml.net/Home/ViewWorkspaceCached/1661759976e540359c001c9efcbc6fc8#Workspaces/Experiments/Experiment/1661759976e540359c001c9efcbc6fc8-f-id.4161d73874db4bd819f8c2a8b278e41/ViewExperiment

Microsoft Azure Machine Learning Studio

USD Index Predictor_LR

Finished running ✓ Properties Project

Search experiments

Metrics

Mean Absolute Error	0.399964
Root Mean Squared Error	0.518255
Relative Absolute Error	1.103543
Relative Squared Error	1.140823
Coefficient of Determination	-0.140823

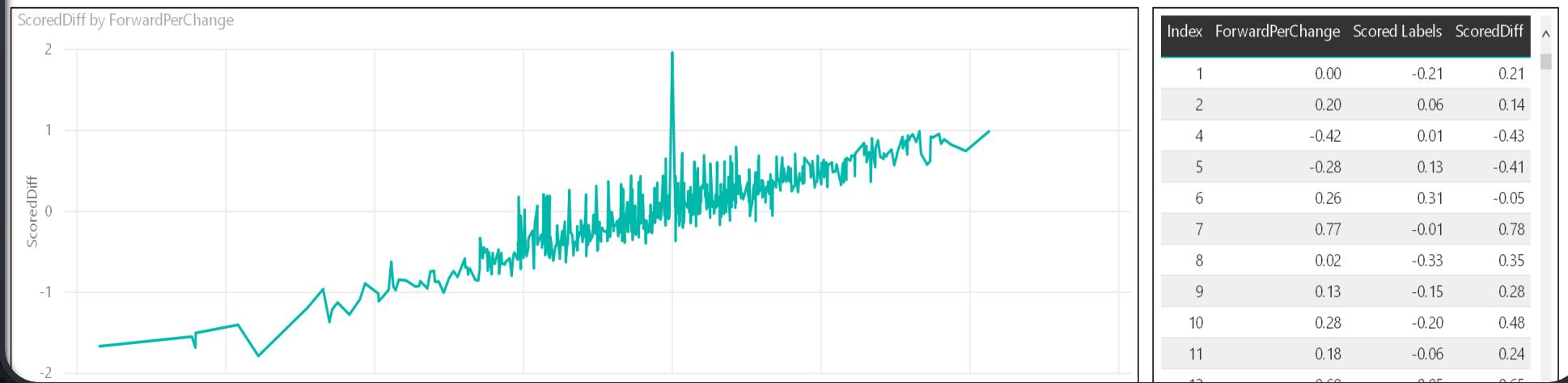
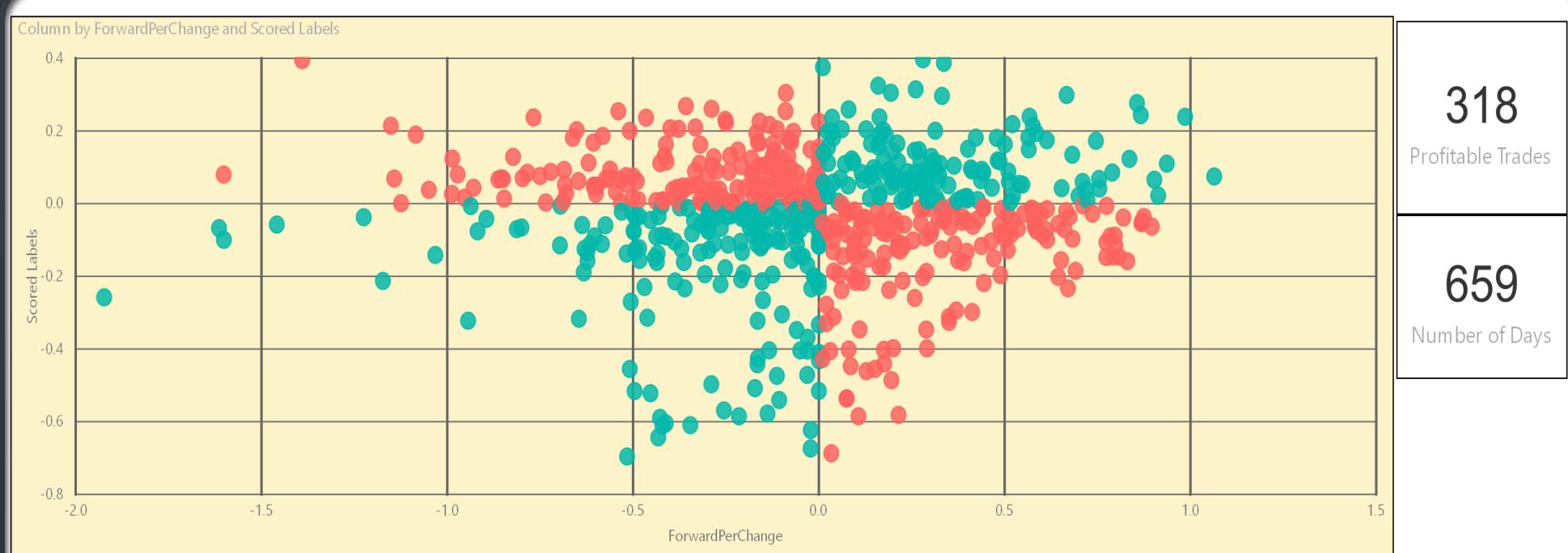
Error Histogram

frequency

Error

0.00083 0.21 0.43 0.64 0.85 1.1 1.3 1.5 1.7 1.9 2.1

Run History Save Discard Changes Run Set Up Web Service Publish to Gallery



LINEAR REGRESSION

Microsoft Azure Machine Learning Studio

Experiments - Microsoft

https://studio.azureml.net/Home/ViewWorkspaceCached/1661759976e540359c001c9efcbc6fc8#Workspaces/Experiments/Experiment/1661759976e540359c001c9efcbc6fc8.f-id.7e84ee647f424f98b14ca0553dc083f6/ViewExperiment

Training experiment Predictive experiment

USD Index Predictor_NN_v1 [Predictive Exp.]

Finished running ✓
Draft saved at 11:40:55 PM

Properties Project

Experiment Properties

- START TIME 9/21/2018 11:40:13 PM
- END TIME 9/21/2018 11:40:14 PM
- STATUS CODE Finished
- STATUS DETAILS None

Go to web service
Prior Run

Summary

Description

Quick Help

Search experiment items

Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Time Series
Web Service
Deprecated

```
graph TD; A[USDollarindex_Data.csv] --> B[Select Columns in Dataset]; B --> C[Score Model]; C --> D[Web service input]; C --> E[Web service output]
```

USING THE TRAINED MODEL TO CREATE AND DEPLOY AN AZURE ML WEB SERVICE

Microsoft Azure Machine Learning Web Services

Quickstart Dashboard Batch Request Log Configure Consume **Test** Swagger API

← USD Index Predictor_NN_v1 [Predictive Exp.]

default

View in Studio

Request-Response Batch

input1 output1

	index	Price	Open	High	Low	Vol	PercentChange	Number_SS_Recipients	AmountUSD	Number_Male_SS_Recipients
index	20050131T000000	86	83.6	83.84	83.39	0	0.14	30100000	956.52	15500000
Date	20050131T000000									
Price	86									
Open	83.6									
High	83.84									
Low	83.39									
Vol	0									
PercentChange	0.14									
Number_SS_Recipients	30100000									

Microsoft Azure Machine Learning Web Services

Quickstart Dashboard Batch Request Log Configure Consume **Test** Swagger API

1000_Plus 18438 Goods producing 22002.4

Total private 111162 Service providing 89159.3

Goods producing 22002.4 FundsPaidInMillionsUSD 23125

Service providing 89159.3 SnPROI 0.0297533

FundsPaidInMillionsUSD 23125 CostIndex 80.773

SnPROI 0.0297533 PopulationInMillions 293.262

CostIndex 80.773 LongRate 4.15

PopulationInMillions 293.262 ForwardPerChange -0.17

LongRate 4.15 Scored Labels 0.164874196052551

RowNum 0

ForwardPerChange -0.17

ForwardPrice 83.45

ForwardPerChangeDirection 0

Test Request-Response

TESTING THE WEB SERVICE
MANUALLY.

CONCLUSIONS

- BOTH TURI – CREATE (GRAPHLAB) AND AZURE ML GAVE THE BEST RESULTS FOR BOOSTED DECISION TREE MODELS
- IF TRADES WERE MADE BASED ON THIS SYSTEM, IT WOULD BE MAKE A PROFIT SLIGHTLY MORE THAN 50% OF THE TIME (PROFITABLE TRADES ARE DEFINED AS THOSE TRADES IN WHICH THE SIGN +/- OF THE ACTUAL LABEL IS THE SAME AS THE PREDICTED LABEL). THIS MEANS THAT THE SYSTEM IS ONLY SLIGHTLY BETTER THAN A COIN TOSS !

LINKS

TURI CREATE (GRAPHLAB) JUPYTER NOTEBOOK AND SOURCE DATA

[HTTPS://GITHUB.COM/SRNIKRIS/SPRINGBOARD_CAPSTONE_2](https://github.com/srinikris/Springboard_Capstone_2)

AZURE ML LINK

[HTTPS://GALLERY.CORTANAINTELLIGENCE.COM/EXPERIMENT/USD-INDEX-PREDICTOR-NN-v1-PREDICTIVE-EXP](https://gallery.cortanaintelligence.com/Experiment/USD-Index-Predictor-NN-v1-Predictive-Exp)