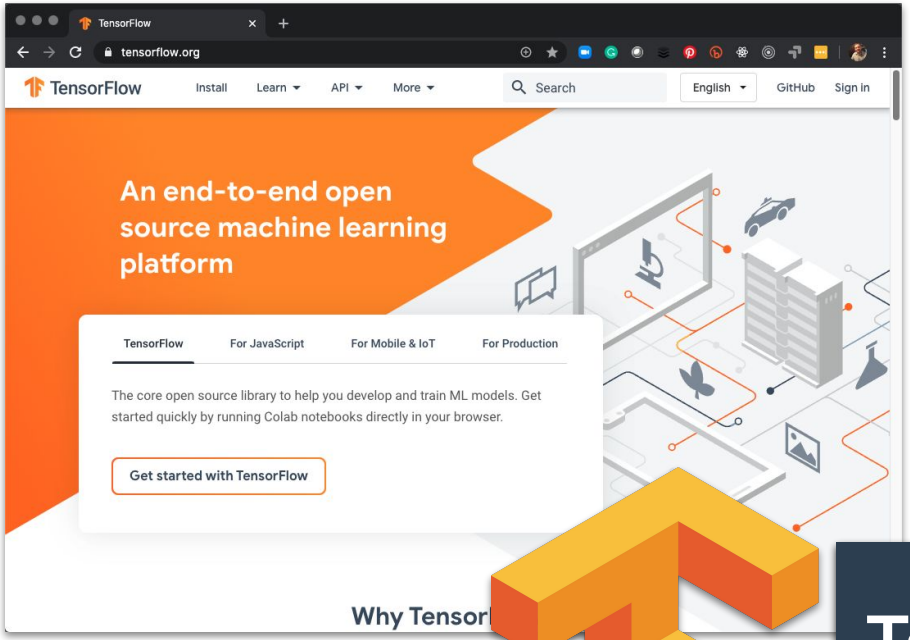
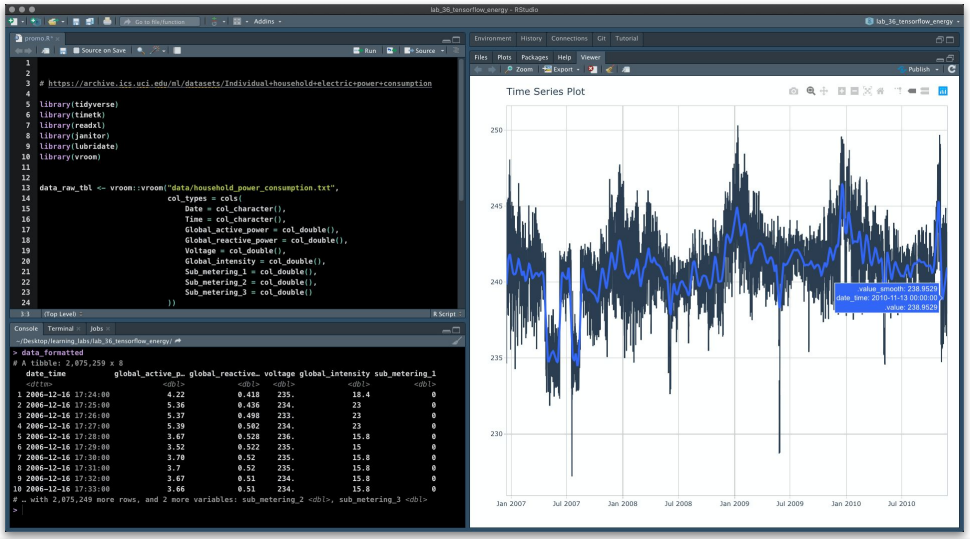



# TensorFlow for Multivariate

## Forecasting Energy Data [Time Series]

Matt Dancho & David Curry

***Business Science Learning Lab***

# Learning Lab Structure

- **Presentation**
  (20 min)
- **Demo's**
  (30 min)
- **Pro-Tips**
  (15 mins)

## Matt Dancho

Founder of Business Science, Matt designs and executes educational courses and workshops that deliver immediate value to organizations. His passion is **up-leveling future data scientists** coming from **untraditional backgrounds**.

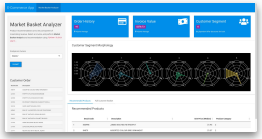
## David Curry

Founder of Sure Optimize, David works with businesses to help improve website performance and SEO using data science. His passion is **ethical Machine Learning initiatives**.

# Python & R Series

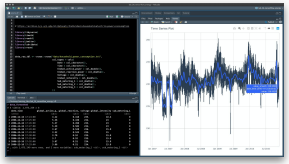

- **Lab 33 - Shiny HR Attrition App**
  - Attrition Clustering w/ **Scikit Learn**

- **Lab 34 - Shiny Ecommerce App**
  - Machine Learning w/ **Scikit Learn**



- **Lab 35 - Shiny Finance App**
  - Univariate Forecasting w/ **TensorFlow**

- **Lab 36 - Energy Forecasting**
  - Multivariate Forecasting w/ **TensorFlow**



- **Lab 37 - Shiny Twitter App**
  - Text Classification

# Learning Labs PRO

1.5-Hour Courses

Recordings, Code, Bonuses, & Slack

- **$39/month for monthly subscription**
- **$29/month for annual subscription**

**Get Labs *2X per Month* on Advanced Topics**

*university.business-science.io*

## LL PRO Course List

***Coming Next! Python & R Series.***

### Python & R Series, 5-Part Series

- **Lab 34 [Part 2]: Advanced Customer Segmentation & Market Basket Analyzer App** (E-Commerce)
- **Lab 33 [Part 1]: Employee Segmentation with Python & R** (HR Analytics)

### Shiny API, 5-Part Series

- **Lab 32 [Part 5]: Text Mining Tweets with Twitter & Tidytext**
- **Lab 31 [Part 4]:** Forecasting Google Analytics with Facebook Prophet & Shiny
- **Lab 30 [Part 3]:** Shiny Financial Analysis with Tidyquant API (Finance)
- **Lab 29 [Part 2]:** Shiny Crude Oil Forecast (Multivariate ARIMA) with Quandl API & Fable
- **Lab 28 [Part 1]:** Shiny Real Estate App with Zillow API

### Marketing Analytics, 4-Part Series

- **Lab 27 [Part 4]: Google Trends Automation with Shiny**
- **Lab 26 [Part 3]:** Machine Learning for Customer Journey
- **Lab 25 [Part 2]:** Marketing Multi-Channel Attribution with ChannelAttribution
- **Lab 24 [Part 1]:** A/B Testing for Website Optimization with Infer & Google Optimize

### SQL for Data Scientists, 3-Part Series

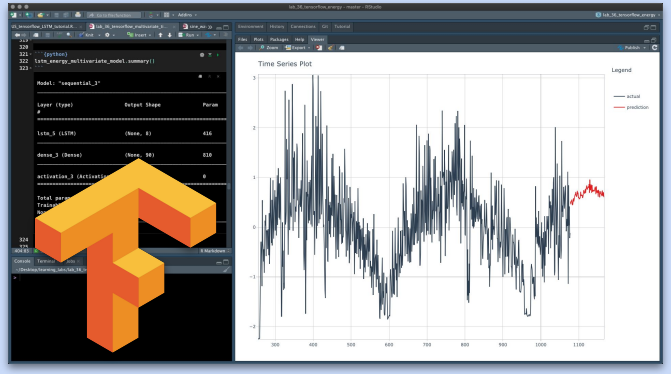
- **Lab 23 [Part 3]:** Google Analytics & BigQuery (SQL) - Conversion Funnel Analysis
- **Lab 22 [Part 2]: SQL for Time Series - Mortgage Loan Delinquency**
- **Lab 21 [Part 1]:** SQL for Data Science - Home Loan Applications & Default

### Plus 20 More Labs:

## Continuous Learning

Data Science

# Agenda




- **Business Case**
  - Forecasting Applications
  - Shiny App from Lab 35
- **TensorFlow**
  - **Expands on Lab 35 (TF for Finance)**
  - **Univariate, Single Step**
  - **Univariate, Multi-Step**
  - **Multivariate, Single Step**
  - **Multivariate, Multi-Step**
- **30-Min Demo**
  - 600 Lines of Code
  - **Enhanced TF Tutorial w/ Univariate, Multivariate, Single Step, & Multi-Step Explained [LL PRO]**
- **Career Acceleration**
  - **Shiny capabilities**
  - **Build your career**

# Bonuses

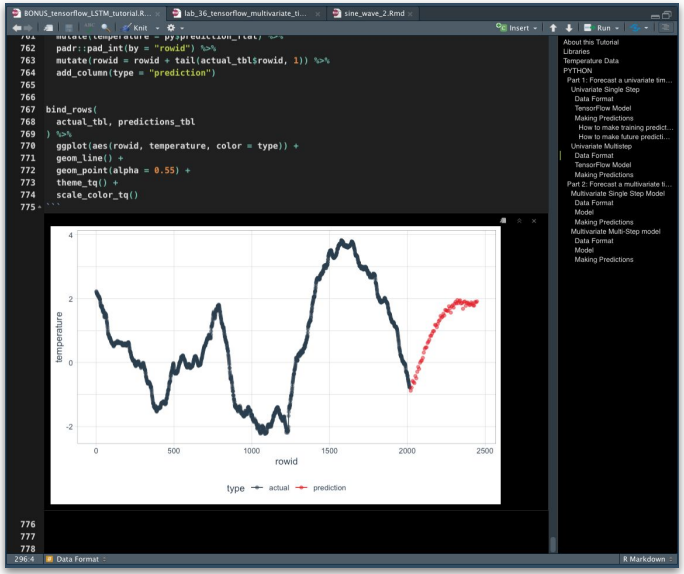
for LL PRO Members Today

# $29/mo

**When you purchase an annual plan**

## Bonus #1 - Enhanced TensorFlow Training

Uses TensorFlow LSTMs in the 4 Most Common Situations




## Bonus #2 - Lab 35 LSTM Explorer

Integrate TensorFlow into your **Shiny Apps!**

# Business Case Study

## Forecasting

# Demo - Become a LL PRO Member, & get this App!

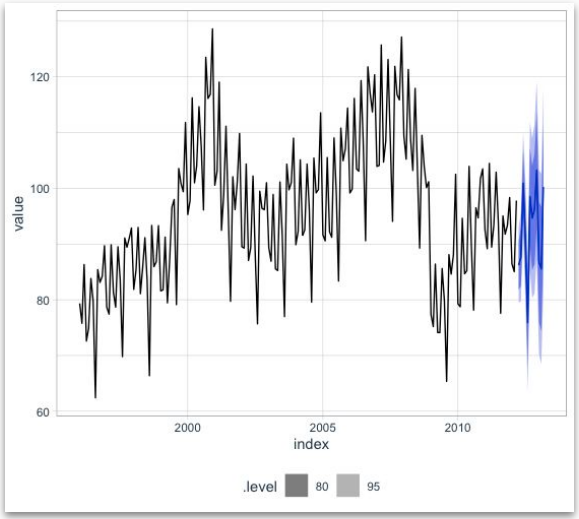

LSTM

## Gold Price Prediction

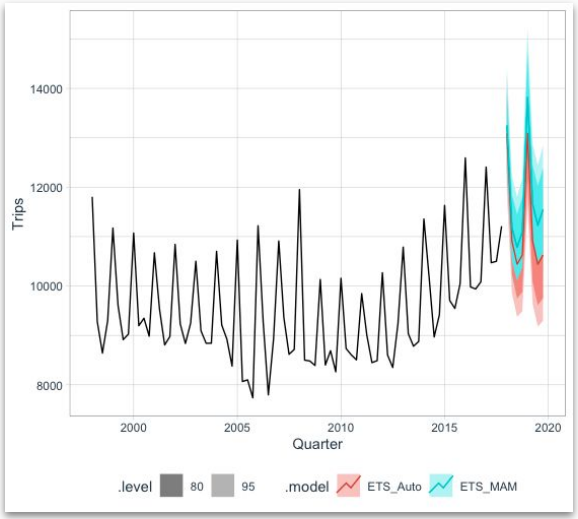
Using Deep Learning

# Classical Methods

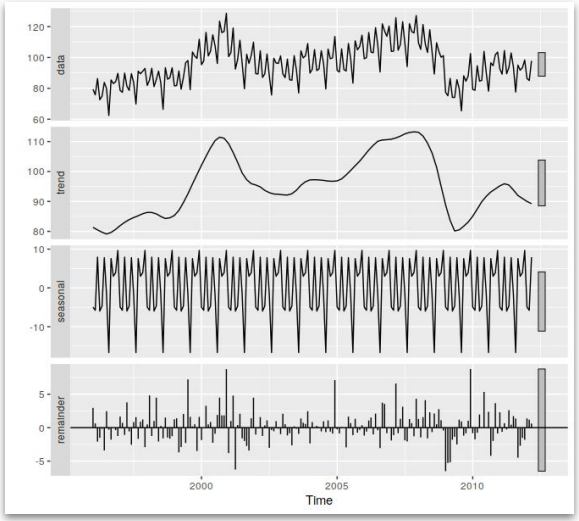
## ARIMA




## ETS



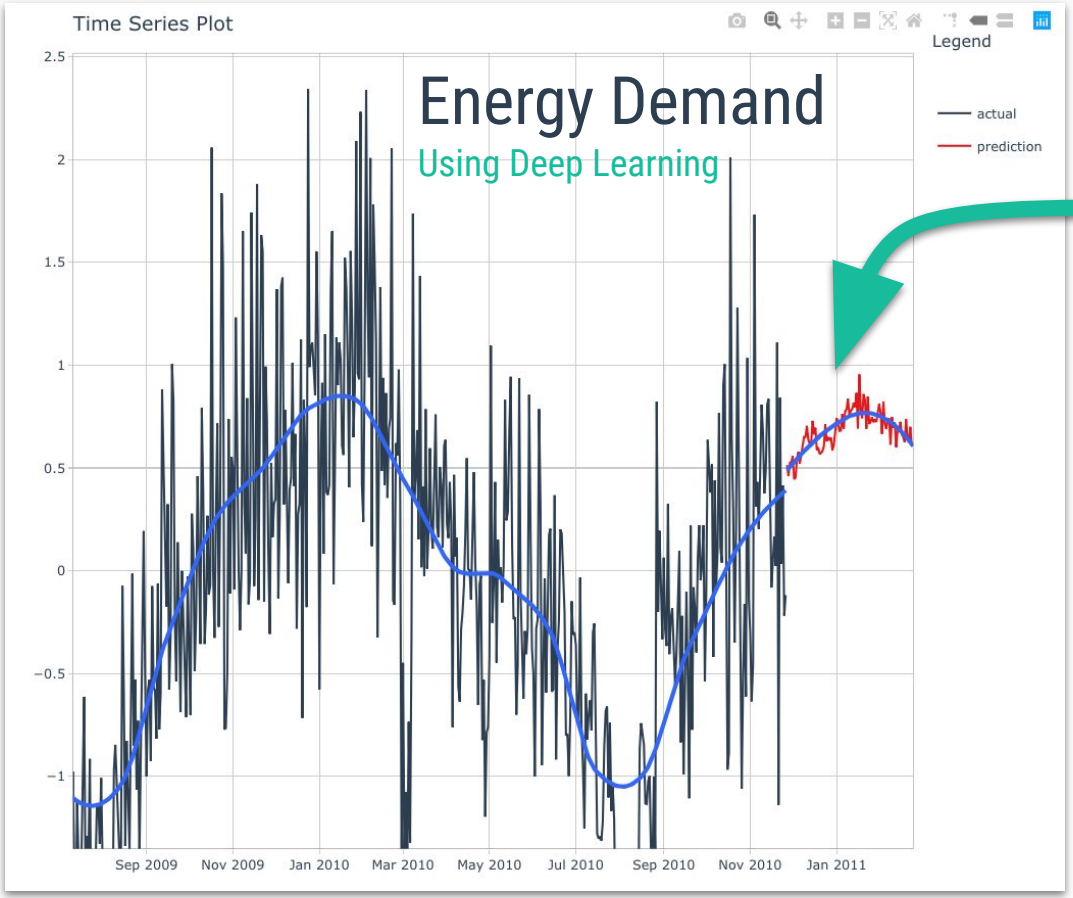

## STL Decomposition



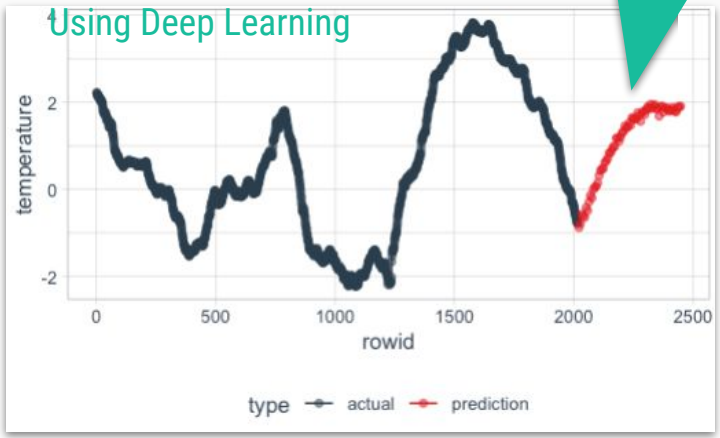

## Classical Methods

Are simple.
Designed for problems with strong autocorrelation, simple trends, & seasonalities.

# Energy Prices & Weather Data




LSTM



# What is an LSTM?

An LSTM is a model that can be used to **model sequences**.

## Comparison to ARIMA

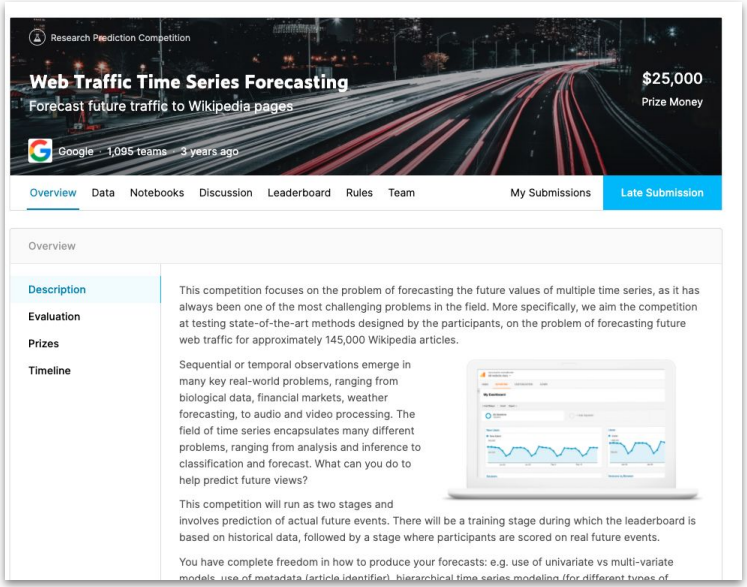
**ARIMA** is a time series model that predicts the future using Autocorrelation with the past.

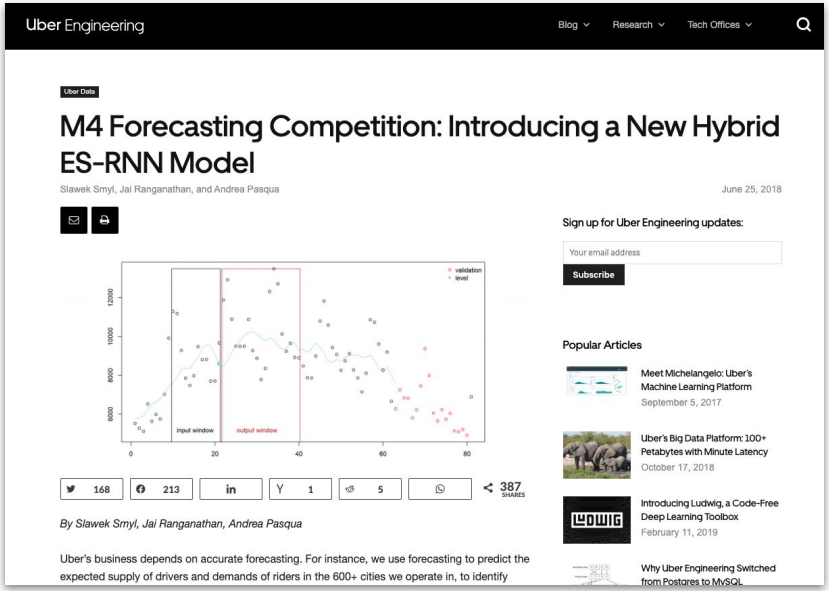
**LSTM** does the same thing with added benefits.

## LSTM Benefits

- LSTM is powerful **enough to learn without tweaking parameters** (just let it run long enough to build relationships)
- Multivariate regressors can easily be incorporated into LSTM (ARIMA requires modeling errors separately)

# Recurrent Neural Networks (RNNs)




## Competition Winners

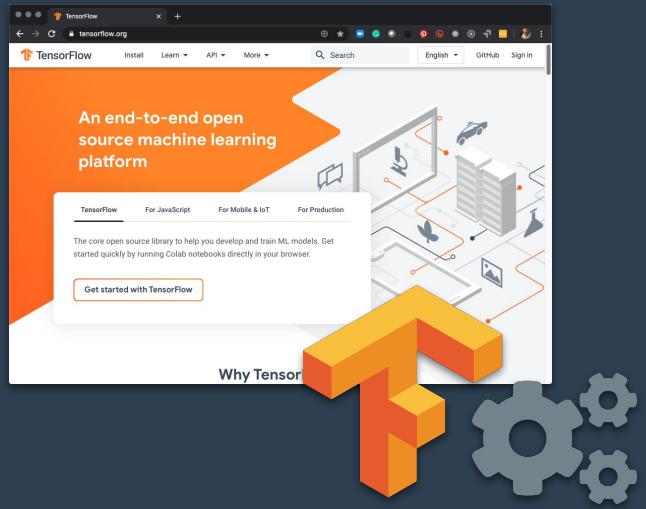
**1st Place** - Kaggle Web Traffic Time Series Forecasting
**1st Place** - M4 Forecasting Competition

Source:

1. https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/43795
2. https://eng.uber.com/m4-forecasting-competition/

# Workflow

From zero to
Automated LSTM Forecasting



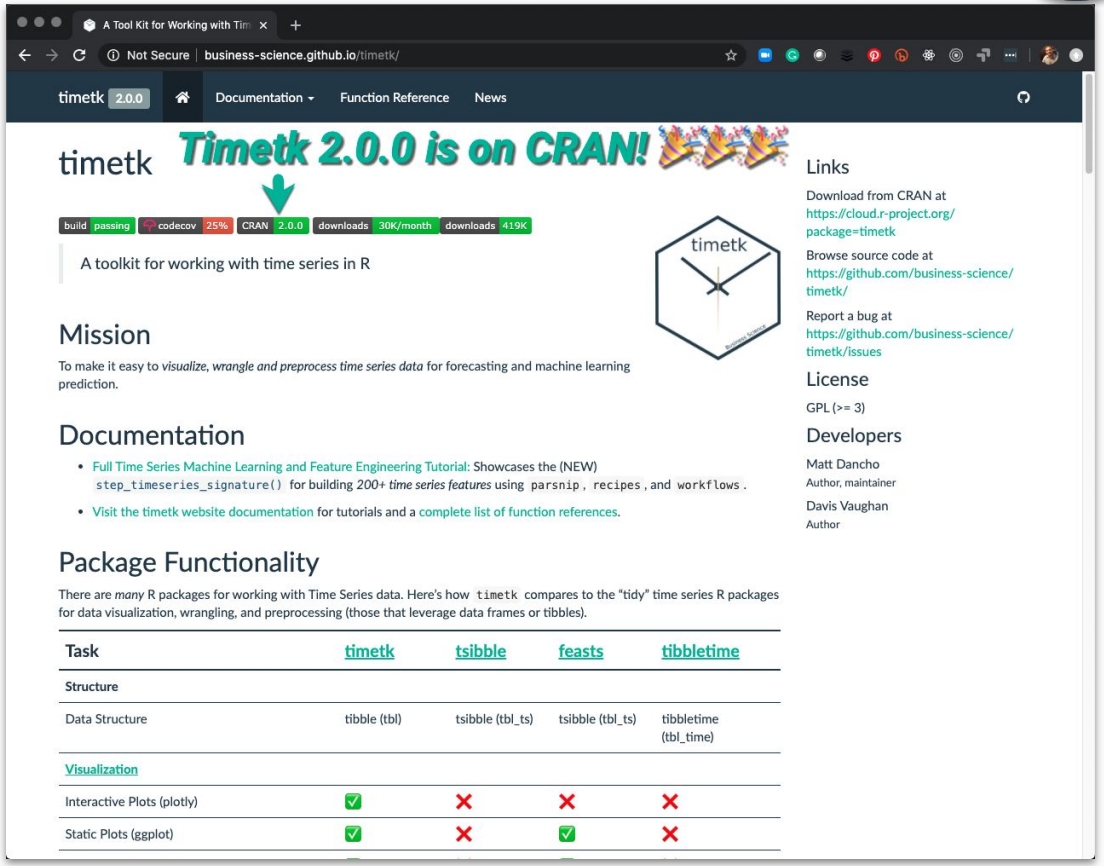
# Announcement: Timetk 2.0.0

Mission:

To make it easy to **visualize, wrangle and preprocess time series data** for forecasting and machine learning prediction.

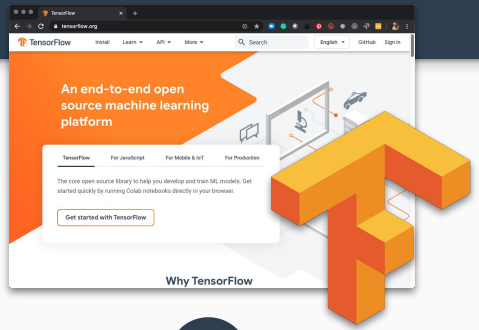
Source:
https://business-science.github.io/timetk/

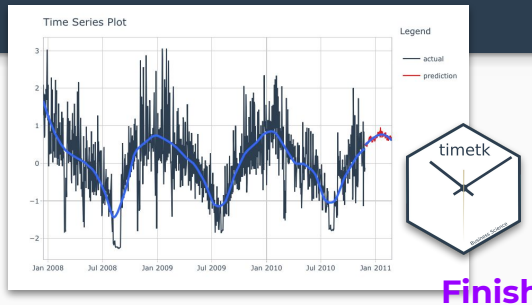
***More announcements coming shortly... Stay tuned!***



# Workflow

## Step-By-Step


Start

1

2

3

Finish

### Data Wrangling & Preprocessing

Formatting the data for analysis

### TensorFlow

LSTM
Deep Learning

### Visualization

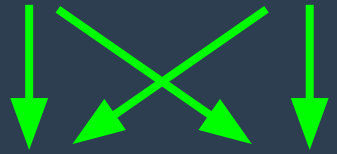
Interactive Plot w/ timetk

# Goals today

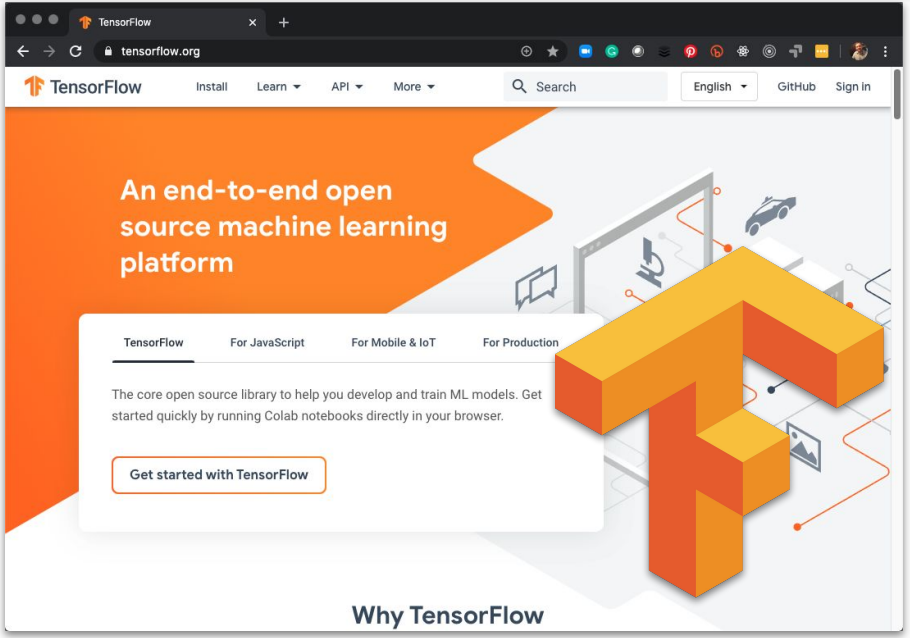
## Minimal Theory, Maximal Results

4 Types of Problems:

1. Univariate vs Multivariate

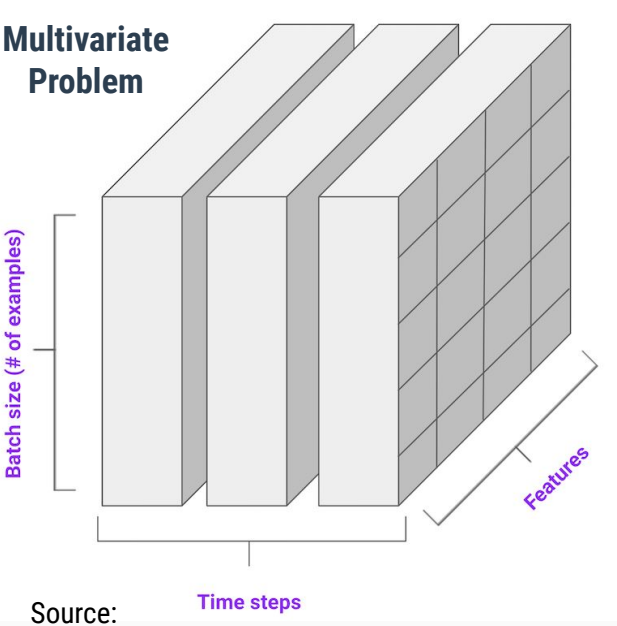


2. Single Step vs Multi-Step

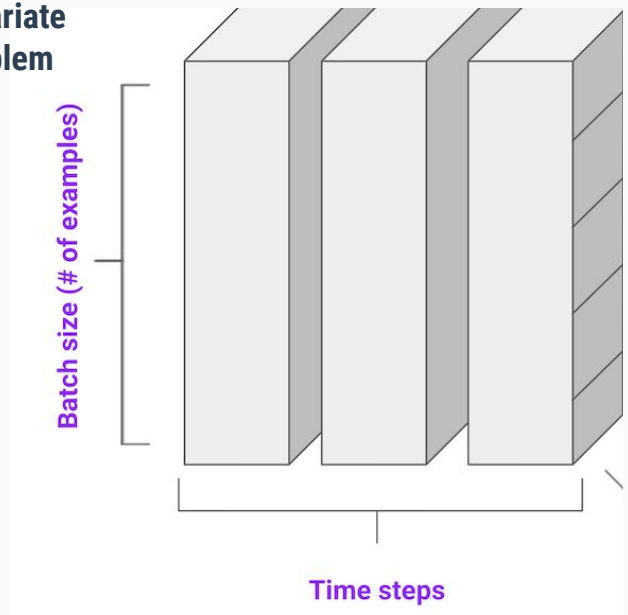



Source: https://www.tensorflow.org/

# LSTM Implementation - Formatting Predictors




Source:
https://www.tensorflow.org/tutorials/structured_data/time_series#top_of_page




## Step 1 - Prepare the Data

Shaping the problem as a **3D Tensor**

# LSTM Implementation

```
299 ## LSTM Model
300
301 ```{python}
302 model = Sequential()
303
304 model.add(Input(shape=(12,1)))
305 model.add(LSTM(64, return_sequences=True, activation='tanh'))
306 model.add(LSTM(12, return_sequences=False, activation='relu'))
307
308 model.add(Dense(1))
309
310 model.add(Activation(activation='linear'))
311
312 model.compile(optimizer="rmsprop", loss="mae")
313
314 ```
```

Lookback

Single Step vs Multi-Step

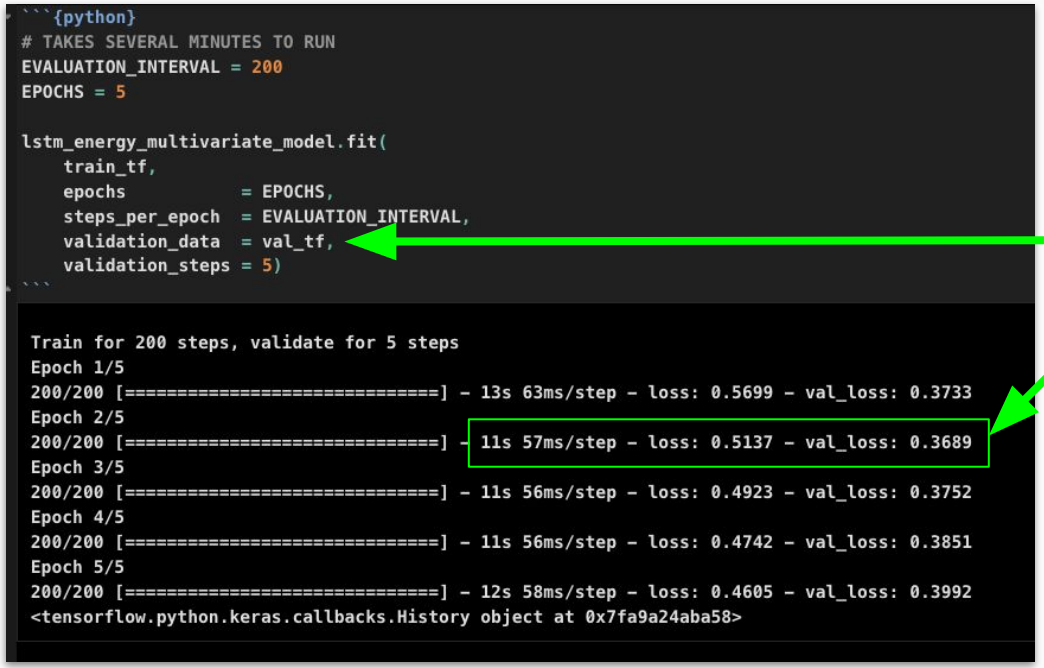
```
316 ```{python}
317 model.summary()
318 ```

Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_26 (LSTM)               (None, 12, 64)            16896
_________________________________________________________________
lstm_27 (LSTM)               (None, 12)                3696
_________________________________________________________________
dense_13 (Dense)             (None, 1)                 13
_________________________________________________________________
activation_13 (Activation)   (None, 1)                 0
=================================================================
Total params: 20,605
Trainable params: 20,605
Non-trainable params: 0
_________________________________________________________________
```

## Step 2 - Design a Model

Set of instructions for TensorFlow's Model API

# LSTM Implementation



```{python}
# TAKES SEVERAL MINUTES TO RUN
EVALUATION_INTERVAL = 200
EPOCHS = 5

lstm_energy_multivariate_model.fit(
    train_tf,
    epochs           = EPOCHS,
    steps_per_epoch  = EVALUATION_INTERVAL,
    validation_data  = val_tf,
    validation_steps = 5)
```

```
Train for 200 steps, validate for 5 steps
Epoch 1/5
200/200 [==============================] - 13s 63ms/step - loss: 0.5699 - val_loss: 0.3733
Epoch 2/5
200/200 [==============================] - 11s 57ms/step - loss: 0.5137 - val_loss: 0.3689
Epoch 3/5
200/200 [==============================] - 11s 56ms/step - loss: 0.4923 - val_loss: 0.3752
Epoch 4/5
200/200 [==============================] - 11s 56ms/step - loss: 0.4742 - val_loss: 0.3851
Epoch 5/5
200/200 [==============================] - 12s 58ms/step - loss: 0.4605 - val_loss: 0.3992
<tensorflow.python.keras.callbacks.History object at 0x7fa9a24aba58>
```

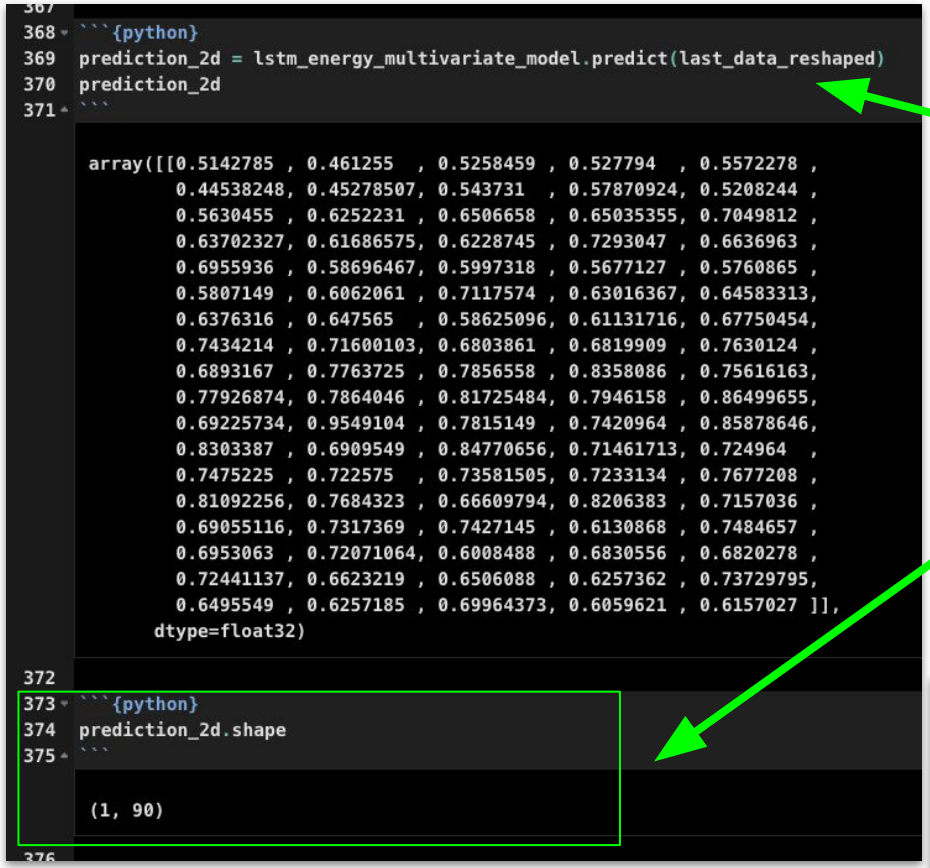
**Critical Point:**
We're implementing Train/Test Validation this Week

**Validation** helps calibrate our model to avoid overfitting

## Step 3 - Fit the Model

Takes a few minutes - Optimizes using Gradient Descent

# LSTM Implementation



```{python}
prediction_2d = lstm_energy_multivariate_model.predict(last_data_reshaped)
prediction_2d
```

```
array([[0.5142785 , 0.461255  , 0.5258459 , 0.527794  , 0.5572278 ,
        0.44538248, 0.45278507, 0.543731  , 0.57870924, 0.5208244 ,
        0.5630455 , 0.6252231 , 0.6506658 , 0.65035355, 0.7049812 ,
        0.63702327, 0.61686575, 0.6228745 , 0.7293047 , 0.6636963 ,
        0.6955936 , 0.58696467, 0.5997318 , 0.5677127 , 0.5760865 ,
        0.5807149 , 0.6062061 , 0.7117574 , 0.63016367, 0.64583313,
        0.6376316 , 0.647565  , 0.58625096, 0.61131716, 0.67750454,
        0.7434214 , 0.71600103, 0.6803861 , 0.6819909 , 0.7630124 ,
        0.6893167 , 0.7763725 , 0.7856558 , 0.8358086 , 0.75616163,
        0.77926874, 0.7864046 , 0.81725484, 0.7946158 , 0.86499655,
        0.69225734, 0.9549104 , 0.7815149 , 0.7420964 , 0.85878646,
        0.8303387 , 0.6909549 , 0.84770656, 0.71461713, 0.724964  ,
        0.7475225 , 0.722575  , 0.73581505, 0.7233134 , 0.7677208 ,
        0.81092256, 0.7684323 , 0.66609794, 0.8206383 , 0.7157036 ,
        0.69055116, 0.7317369 , 0.7427145 , 0.6130868 , 0.7484657 ,
        0.6953063 , 0.72071064, 0.6008488 , 0.6830556 , 0.6820278 ,
        0.72441137, 0.6623219 , 0.6506088 , 0.6257362 , 0.73729795,
        0.6495549 , 0.6257185 , 0.69964373, 0.6059621 , 0.6157027 ]],
      dtype=float32)
```

```{python}
prediction_2d.shape
```

```
(1, 90)
```

**3D Arrays**
LSTM's work with 3D sequences (cubes)

**2D Arrays**
LSTM's return predictions as 2D arrays

## Step 4 - Predict

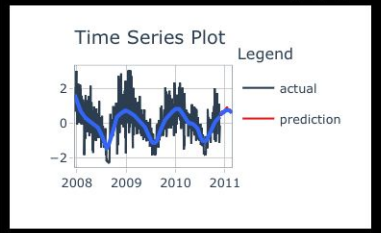
Reshaping the data becomes very critical w/ multivariate

# Automate with Shiny

# 30-Min Demo

lab_36_tensorflow_energy - master - RStudio

```
403   plot_time_series(date_time,
      global_active_power, .color_var = type,
      .smooth_period = "180 days")
404 ```
```




```
405
406
407 # BONUSES
408
409 TensorFlow Enhanced LSTM Tutorial:
410
411     * __Univariate Single-Step Prediction:__
    Where will your forecast be 2 months in the
    future?
412     * __Univariate Multi-Step Prediction:__
    What are all of the future values expected
```

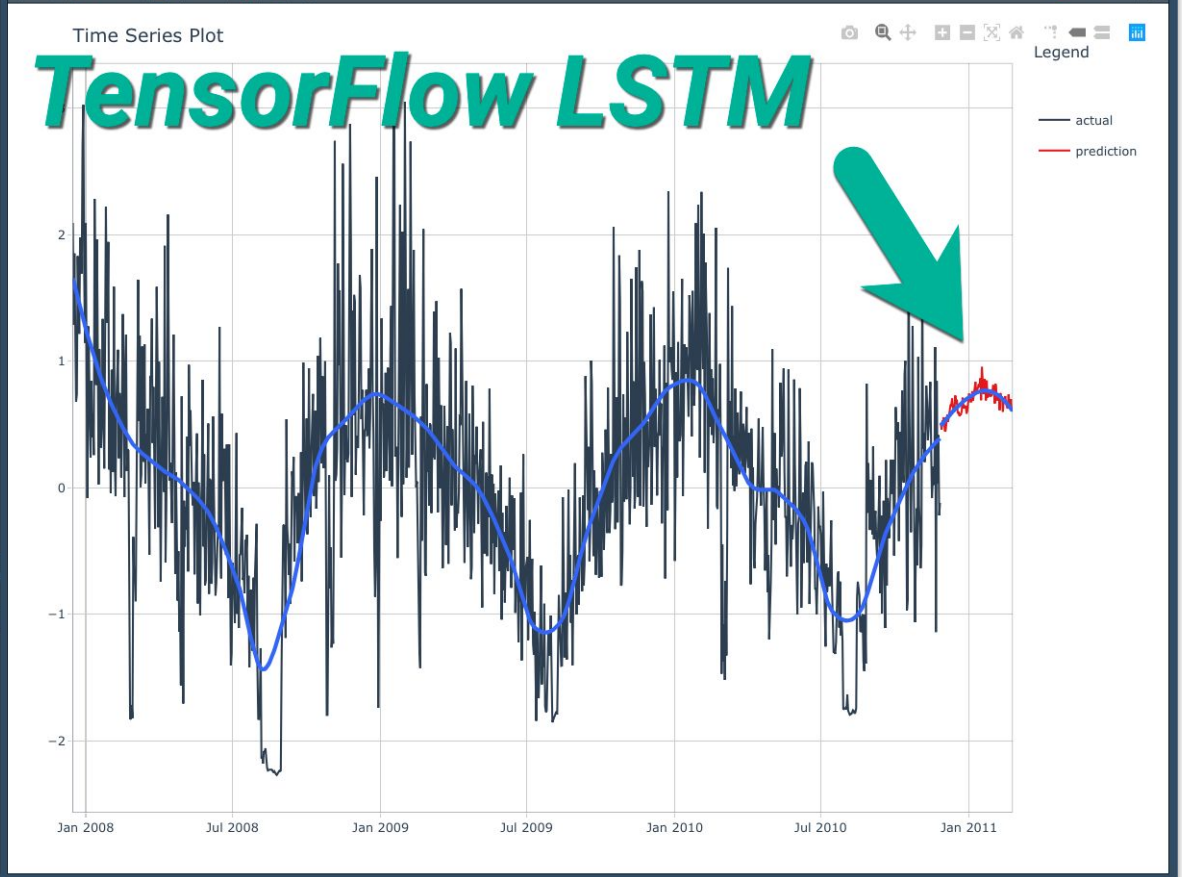
Libraries
R (DATA PREP)
Data
Visualization
Data Wrangling
Autocorrelation (Features)
Preprocessing
PYTHON
TensorFlow Modeling
Packages
Setup
Data Formatting
Model
Making Predictions
BONUSES

Console | Terminal | Jobs

~/Desktop/learning_labs/lab_36_tensorflow_energy/

**180-Day Forecast**
**Energy Data**
**Crushes Simple Methods**

# TensorFlow LSTM



# Career Acceleration

## And how Shiny Apps help

# Automate with Shiny

# Career Tip #1: Give Businesses Apps

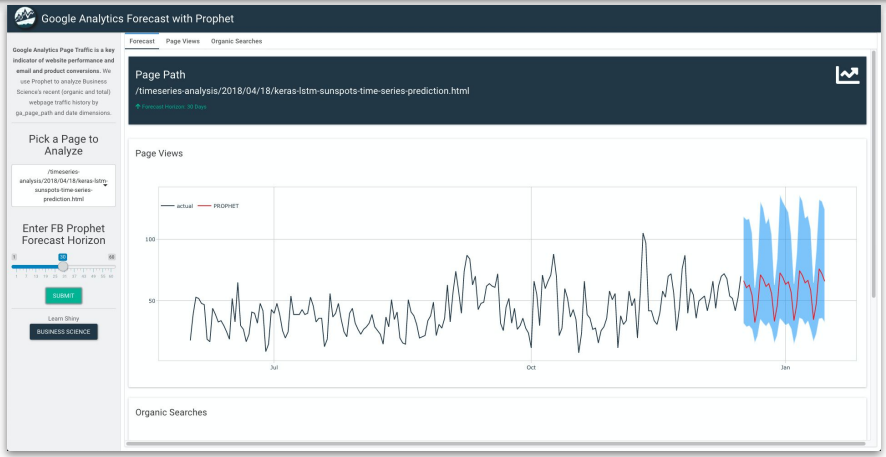
## Apps are what businesses need

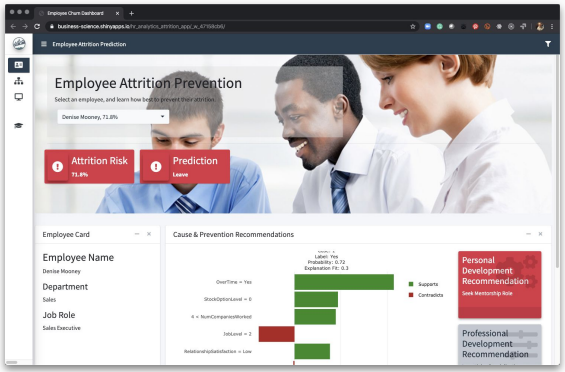
Businesses **can't** scale reports

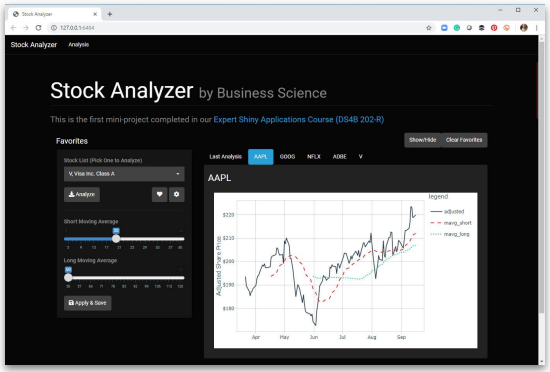
Businesses **have** cloud & servers (AWS / Azure)

Businesses **need** your analysis in the right form

**You can** solve these problems with shiny apps

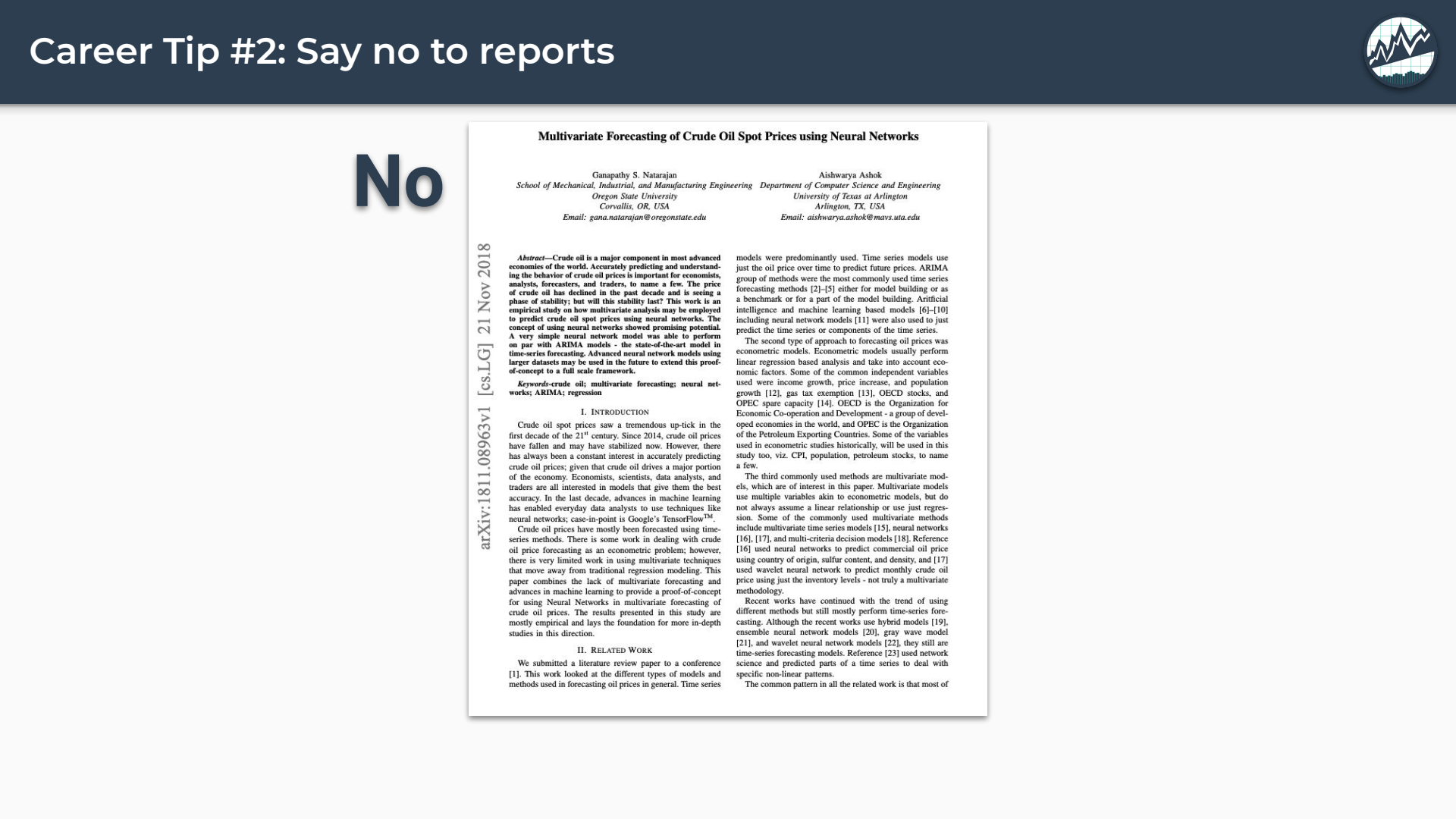



**Apps solve these challenges**

# Career Tip #2: Say no to reports



No



## Multivariate Forecasting of Crude Oil Spot Prices using Neural Networks

Ganapathy S. Natarajan
*School of Mechanical, Industrial, and Manufacturing Engineering*
*Oregon State University*
*Corvallis, OR, USA*
*Email: gana.natarajan@oregonstate.edu*

Aishwarya Ashok
*Department of Computer Science and Engineering*
*University of Texas at Arlington*
*Arlington, TX, USA*
*Email: aishwarya.ashok@mavs.uta.edu*

***Abstract*—Crude oil is a major component in most advanced economies of the world. Accurately predicting and understanding the behavior of crude oil prices is important for economists, analysts, forecasters, and traders, to name a few. The price of crude oil has declined in the past decade and is seeing a phase of stability; but will this stability last? This work is an empirical study on how multivariate analysis may be employed to predict crude oil spot prices using neural networks. The concept of using neural networks showed promising potential. A very simple neural network model was able to perform on par with ARIMA models - the state-of-the-art model in time-series forecasting. Advanced neural network models using larger datasets may be used in the future to extend this proof-of-concept to a full scale framework.**

***Keywords*-crude oil; multivariate forecasting; neural networks; ARIMA; regression**

### I. Introduction

Crude oil spot prices saw a tremendous up-tick in the first decade of the 21st century. Since 2014, crude oil prices have fallen and may have stabilized now. However, there has always been a constant interest in accurately predicting crude oil prices; given that crude oil drives a major portion of the economy. Economists, scientists, data analysts, and traders are all interested in models that give them the best accuracy. In the last decade, advances in machine learning has enabled everyday data analysts to use techniques like neural networks; case-in-point is Google's TensorFlow™.

Crude oil prices have mostly been forecasted using time-series methods. There is some work in dealing with crude oil price forecasting as an econometric problem; however, there is very limited work in using multivariate techniques that move away from traditional regression modeling. This paper combines the lack of multivariate forecasting and advances in machine learning to provide a proof-of-concept for using Neural Networks in multivariate forecasting of crude oil prices. The results presented in this study are mostly empirical and lays the foundation for more in-depth studies in this direction.

### II. Related Work

We submitted a literature review paper to a conference [1]. This work looked at the different types of models and methods used in forecasting oil prices in general. Time series models were predominantly used. Time series models use just the oil price over time to predict future prices. ARIMA group of methods were the most commonly used time series forecasting methods [2]–[5] either for model building or as a benchmark or for a part of the model building. Aritficial intelligence and machine learning based models [6]–[10] including neural network models [11] were also used to just predict the time series or components of the time series.

The second type of approach to forecasting oil prices was econometric models. Econometric models usually perform linear regression based analysis and take into account economic factors. Some of the common independent variables used were income growth, price increase, and population growth [12], gas tax exemption [13], OECD stocks, and OPEC spare capacity [14]. OECD is the Organization for Economic Co-operation and Development - a group of developed economies in the world, and OPEC is the Organization of the Petroleum Exporting Countries. Some of the variables used in econometric studies historically, will be used in this study too, viz. CPI, population, petroleum stocks, to name a few.

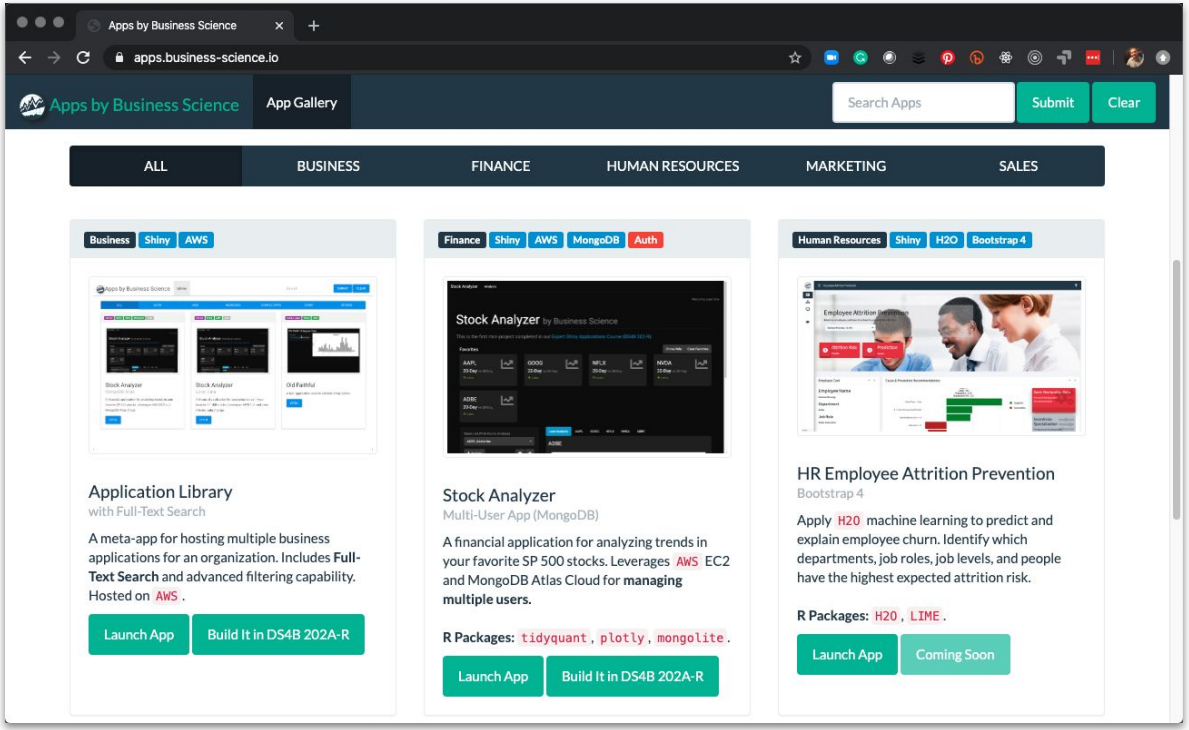
The third commonly used methods are multivariate models, which are of interest in this paper. Multivariate models use multiple variables akin to econometric models, but do not always assume a linear relationship or use just regression. Some of the commonly used multivariate methods include multivariate time series models [15], neural networks [16], [17], and multi-criteria decision models [18]. Reference [16] used neural networks to predict commercial oil price using country of origin, sulfur content, and density, and [17] used wavelet neural network to predict monthly crude oil price using just the inventory levels - not truly a multivariate methodology.

Recent works have continued with the trend of using different methods but still mostly perform time-series forecasting. Although the recent works use hybrid models [19], ensemble neural network models [20], gray wave model [21], and wavelet neural network models [22], they still are time-series forecasting models. Reference [23] used network science and predicted parts of a time series to deal with specific non-linear patterns.

The common pattern in all the related work is that most of

# Career Tip #3: Say YES to apps.

## Yes




apps.business-science.io

# 4-Course R-Track System

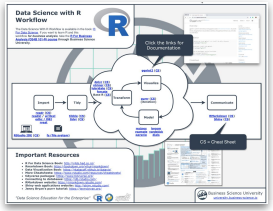
## Business Analysis with R
(DS4B 101-R)

## Data Science For Business with R
(DS4B 201-R)

## Web Apps & Shiny Developer
(DS4B 102-R + DS4B 202A-R)

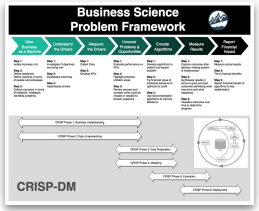
**Project-Based Courses with Business Application**

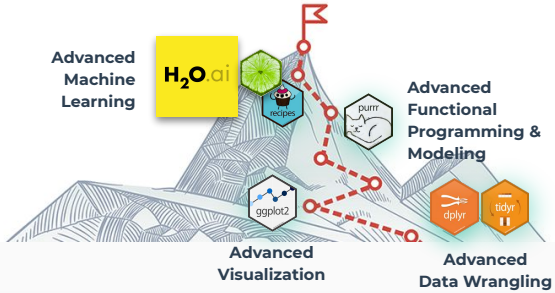
### Data Science Foundations
**7 Weeks**



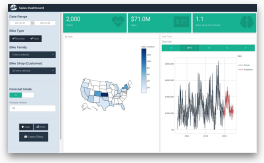


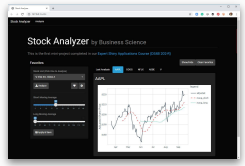
### Machine Learning & Business Consulting
**10 Weeks**




### Web Application Development
**12 Weeks**






# Key Benefits

- Fundamentals - Weeks 1-5 (25 hours of Video Lessons)
  - Data Manipulation (dplyr)
  - Time series (lubridate)
  - Text (stringr)
  - Categorical (forcats)
  - Visualization (ggplot2)
  - Programming & Iteration (purrr)
  - 3 Challenges

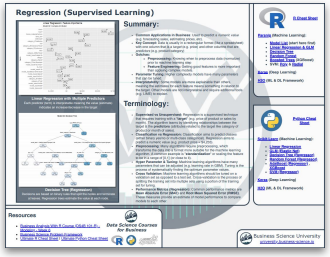
- **Machine Learning - Week 6 (8 hours of Video Lessons)**
  - **Clustering (3 hours)**
  - **Regression (5 hours)**
  - **2 Challenges**

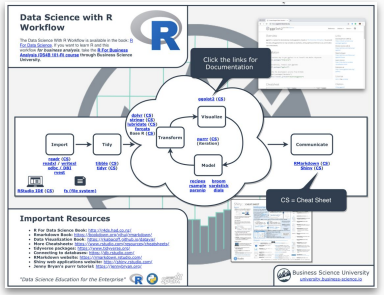
- Learn Business Reporting - Week 7
  - RMarkdown & plotly
  - 2 Project Reports:
    1. Product Pricing Algo
    2. Customer Segmentation

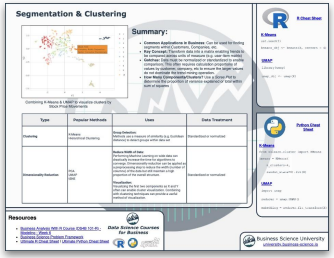
# Business Analysis with R (DS4B 101-R)

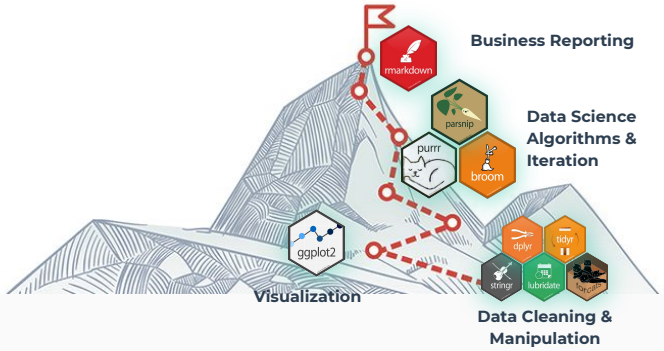
Data Science Foundations

**7 Weeks**




# Key Benefits

## End-to-End Churn Project

Understanding the Problem & Preparing Data - Weeks 1-4

- Project Setup & Framework
- Business Understanding / Sizing Problem
- Tidy Evaluation - rlang
- EDA - Exploring Data -GGally, skimr
- Data Preparation - recipes
- Correlation Analysis
- 3 Challenges

**Machine Learning - Weeks 5, 6, 7**

- **H2O AutoML - Modeling Churn**
- **ML Performance**
- **LIME Feature Explanation**

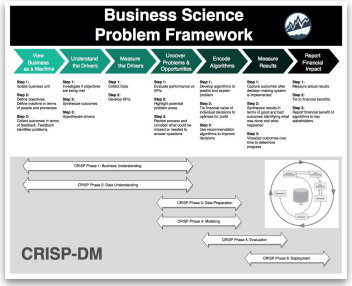
**Return-On-Investment - Weeks 7, 8, 9**

- **Expected Value Framework**
- **Threshold Optimization**
- **Sensitivity Analysis**
- **Recommendation Algorithm**

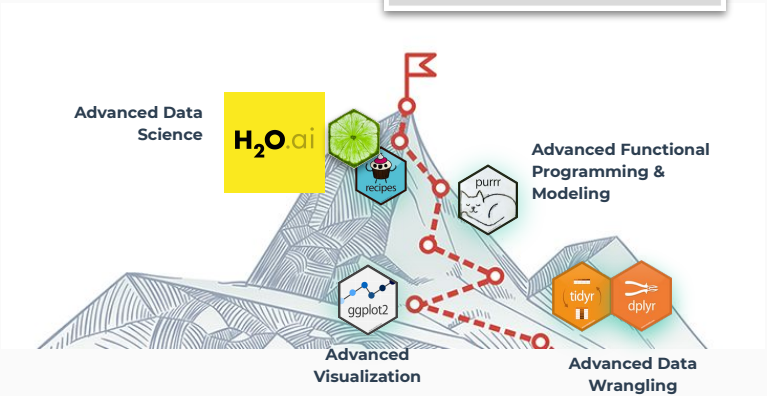
# Data Science For Business (DS4B 201-R)

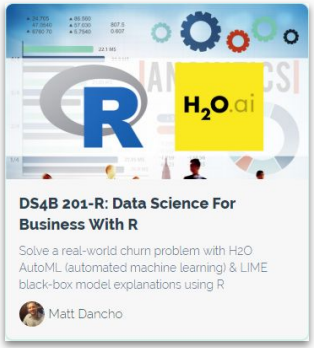
Machine Learning & Business Consulting

**10 Weeks**





# Key Benefits

Learn Shiny & Flexdashboard
- Build Applications
- Learn Reactive Programming
- Integrate Machine Learning

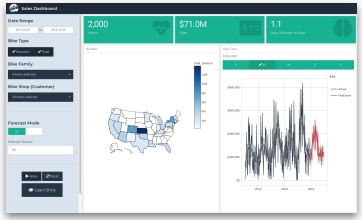
**App #1: Predictive Pricing App**
- Model Product Portfolio
- XGBoost Pricing Prediction
- Generate new products instantly

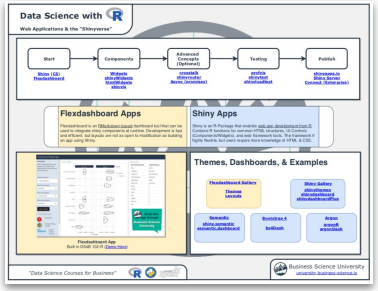
**App #2: Sales Dashboard with Demand Forecasting**
- Model Demand History
- Segment Forecasts by Product & Customer
- XGBoost Time Series Forecast
- Generate new forecasts instantly

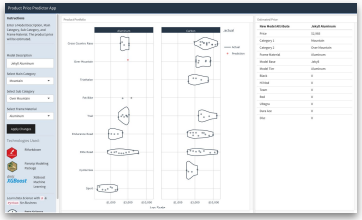
# Shiny Apps for Business (DS4B 102-R)

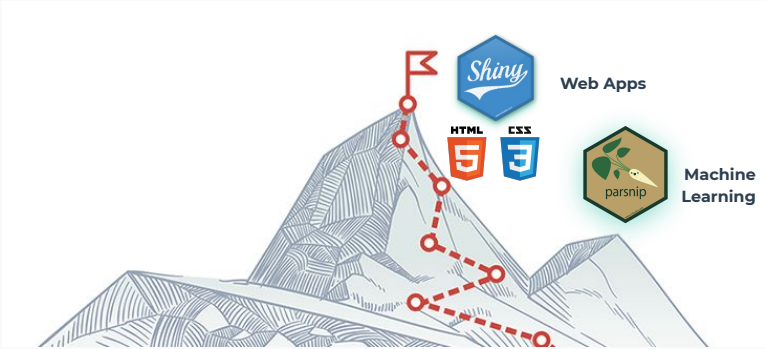
Web Application Development

**4 Weeks**






# Key Benefits

Frontend + Backend + Production Deployment

**Frontend for Shiny**
- Bootstrap

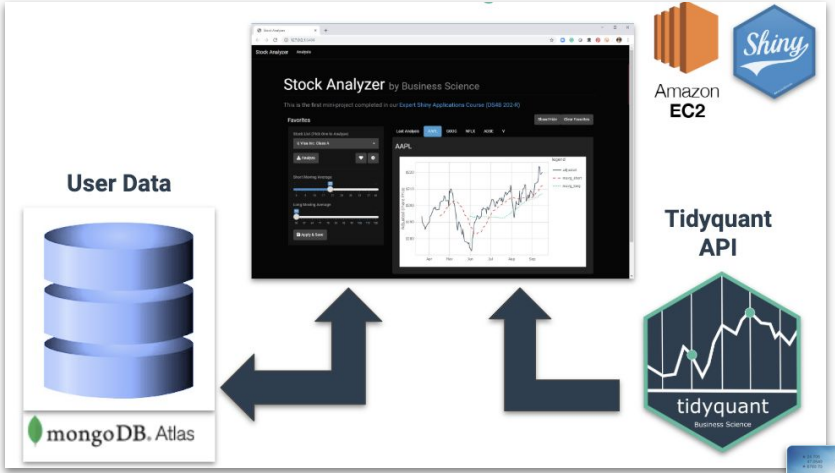
**Backend for Shiny**
- MongoDB Atlas Cloud
- Dynamic UI
- User Authentication & Security

**Production Deployment**
- AWS
- EC2 Server
- SSL & HTTPS Encryption

# Shiny Apps for Business (DS4B 202A-R)

Web Application Development

**6 Weeks**




aws

docker git Bootstrap mongoDB.

# 15% OFF PROMO Code: learninglabs

## R-TRACK BUNDLE


**4-Course Bundle - Machine Learning + Expert Web Applications (R-Track)**

Go from Beginner to Expert Data Scientist & Shiny Developer in Under 6-Months

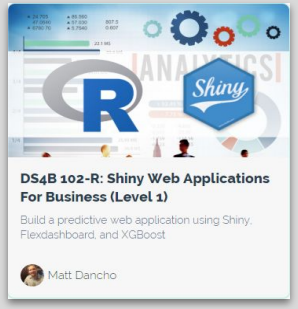
4 Course Bundle ~~$1,596~~

**$127/mo**
**Limited Time**


**DS4B 101-R: Business Analysis With R**

Your Data Science Journey Starts Now! Learn the fundamentals of data science for business with the tidyverse.

Matt Dancho




**DS4B 102-R: Shiny Web Applications For Business (Level 1)**

Build a predictive web application using Shiny, Flexdashboard, and XGBoost

Matt Dancho


**DS4B 201-R: Data Science For Business With R**

Solve a real-world churn problem with H2O AutoML (automated machine learning) & LIME black-box model explanations using R

Matt Dancho


**DS4B 202A-R: Expert Shiny Developer with AWS**

Learn how to build Scalable Data Science Applications using R, Shiny, and AWS Cloud Technology.

Matt Dancho

| | | |
|---|---|---|
| ○ | **Paid Course** 15% COUPON DISCOUNT | ~~$1,596~~ **$1,356.60** |
| ◉ | **12 Low Monthly Payments** 15% COUPON DISCOUNT<br>12X Payment Plan | ~~12 payments of $149/m~~ **12 payments of $126.65/m** |

# Career acceleration awaits

university.business-science.io