# Predicting the future

“The quickest way to double your money is to fold it over and put it back in your pocket” - Will Rogers

**Ryan, Murthy and Francene**

# Contents



# Summary and Motivation

## The First Thing.

Our client Ralph Williams has decided to begin self-investing part of his portfolio. He's asked us to create a dashboard so that he may make informed trading decisions.

## The Second Thing.

We advised Ralph that direct-investing with limited knowledge of investments is risky, and we wouldn't recommend he self-invest, but you know Ralph, and if you forgot, here's Ralph.

## The Third Thing.

We sourced our data for the top 10 market-cap stocks from Alpaca and the top 10 crypto's from Shrimpy. We used Random Forest a supervised learning model to analyze the data and inform our dashboard.

# Model Summary

| | |
|---|---|
| Model Type | **Random Forrest.** |
| What is Random Forrest? | **Random forests are an ensemble learning method for classification.** |
| What's Classification? | **Classification is a form of supervised learning that is intended for predicting variables. In our case future stock and crypto performance.** |

# Data Cleanup

and Model Training

#1

FETCHED DATA
Alpaca and Shrimpy(
CryptoData.ipynb

#2

DATA CLEANING
Dropped nulls etc.
CryptoCleaning.ipynb

#3

ANALYSIS AND TESTING
CryptoAnalysis.ipynb
Google Colab
LSTM

#4

MODEL BUILDING
ModelBuilding.ipynb

# Model Evaluation

## Technical Indicators

Exponential moving Averages in Feature 1.

Exponential Moving Average of Daily Return Volatility in Feature 2.

Bolligner Bands in Feature 3.

## Model

Supervised Machine Learning Model used was Random Forest based on Three Signals:

crossover_signal.

vol_trend_signal.

bollinger_signal.

One year of historical data used.

## Model Continued

Set the dependent variable based on the logic where if daily return is greater than 0, then 1, else then 0.

Two sets of data constructed the X_train and y_train.

## Fit

SKlearn linear regression

Confusion matrix

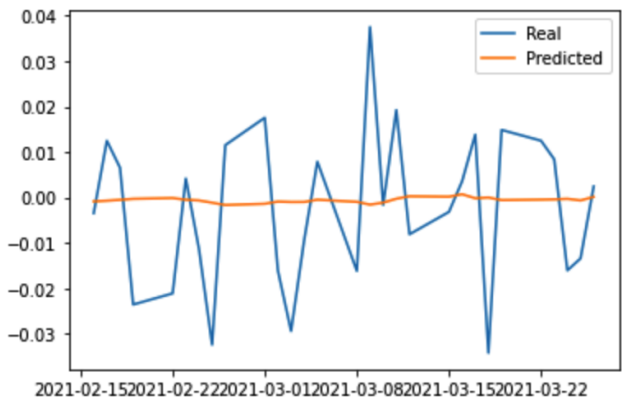
Accuracy score

# Top 10 Market Cap NYSE

## Alpaca Plots

```
# Plot the real vs predicted prices as a line chart
stocks.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0267088ed0>
```




```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 5, 5)              140
_________________________________________________________________
dropout (Dropout)            (None, 5, 5)              0
_________________________________________________________________
lstm_1 (LSTM)                (None, 5, 5)              220
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 5)              0
_________________________________________________________________
lstm_2 (LSTM)                (None, 5)                 220
_________________________________________________________________
dropout_2 (Dropout)          (None, 5)                 0
_________________________________________________________________
dense (Dense)                (None, 1)                 6
=================================================================
Total params: 586
Trainable params: 586
Non-trainable params: 0
_________________________________________________________________
```

```
# Train the model
model.fit(X_train, y_train, epochs=10, shuffle=False, batch_size=1, verbose=1)
```

```
Epoch 1/10
65/65 [==============================] - 4s 5ms/step - loss: 0.1649
Epoch 2/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0479
Epoch 3/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0505
Epoch 4/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0537
Epoch 5/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0592
Epoch 6/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0599
Epoch 7/10
65/65 [==============================] - 0s 4ms/step - loss: 0.0576
Epoch 8/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0362
Epoch 9/10
65/65 [==============================] - 0s 5ms/step - loss: 0.0520
Epoch 10/10
65/65 [==============================] - 0s 4ms/step - loss: 0.0490
<tensorflow.python.keras.callbacks.History at 0x7f026b4ccf10>
```

## Evaluate the Model

It's time to evaluate our model to assess its performance. We will use the `evaluate` method using the testing data.

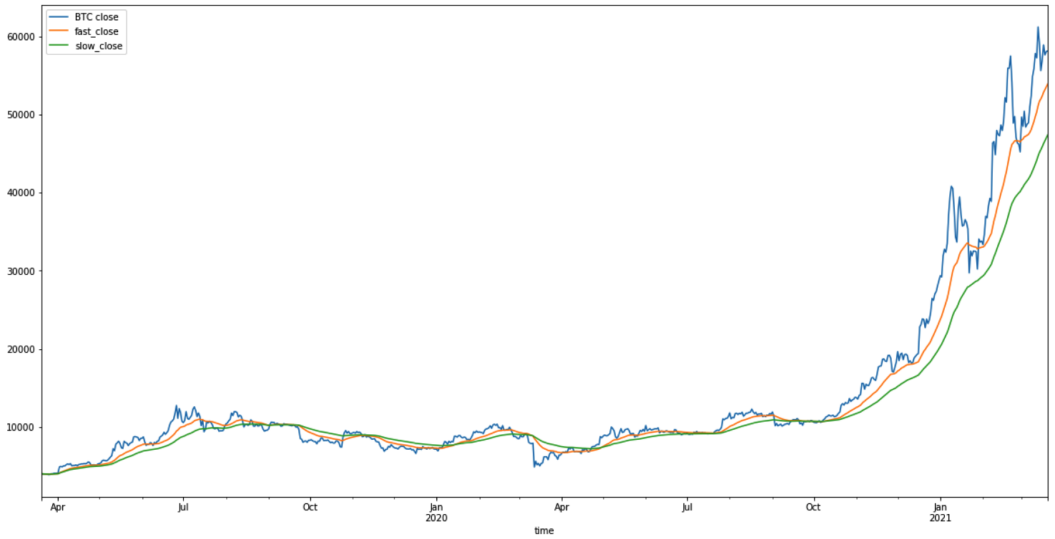
```
[40]: # Evaluate the model
      model.evaluate(X_test, y_test)
```

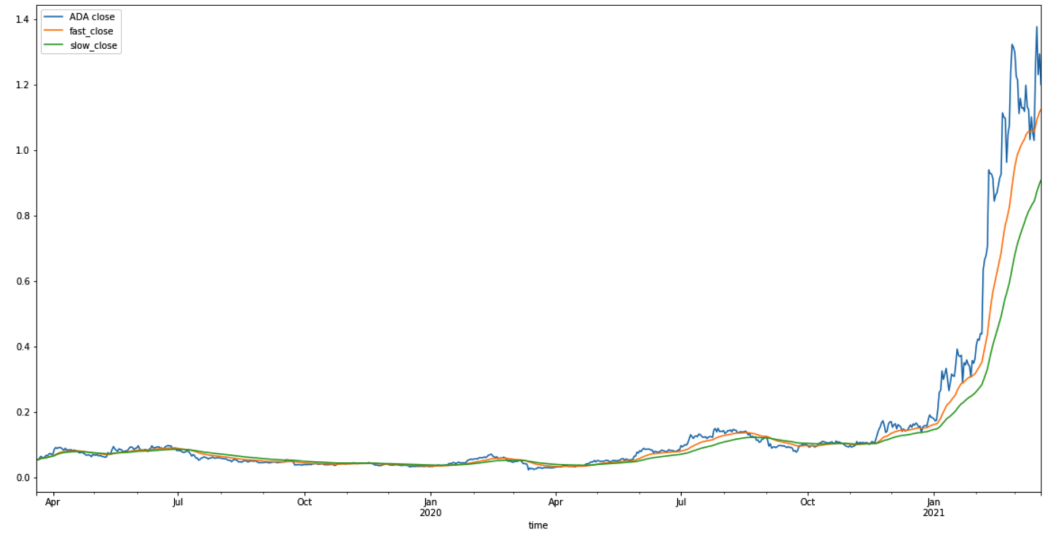
```
1/1 [==============================] - 1s 1s/step - loss: 0.0457
[40]: 0.045650918036699295
```

fig. 1 Stocks Real vs Predicted Prices, fig 2 Model, Fig 3 Fit & Train, fig 4 Tory, Fig 5 Evaluation

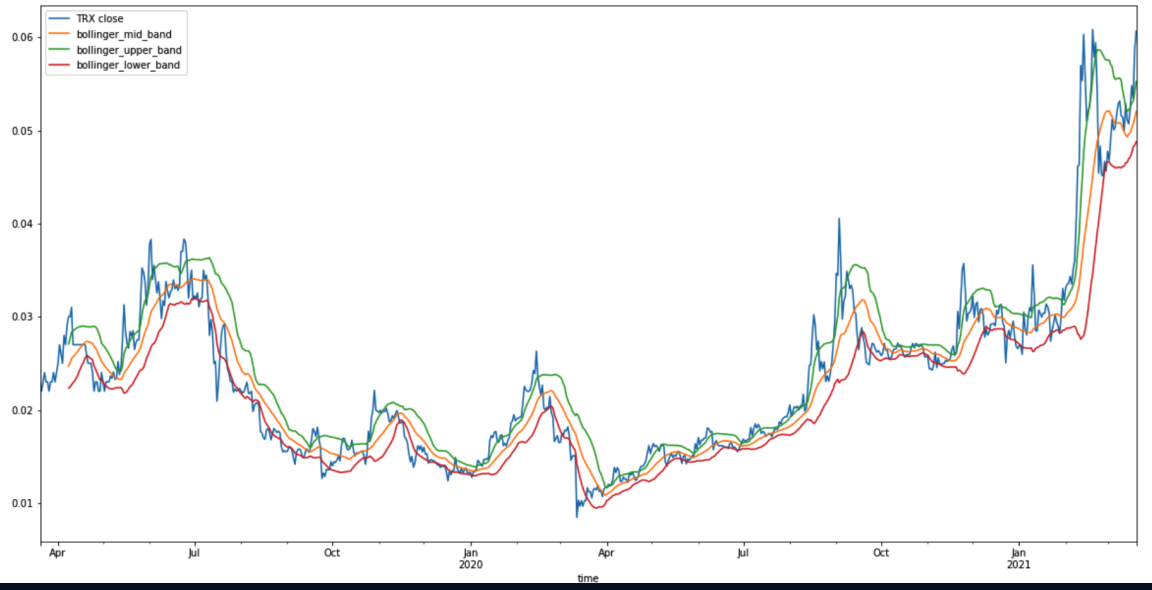
# Top 10 Crypto's

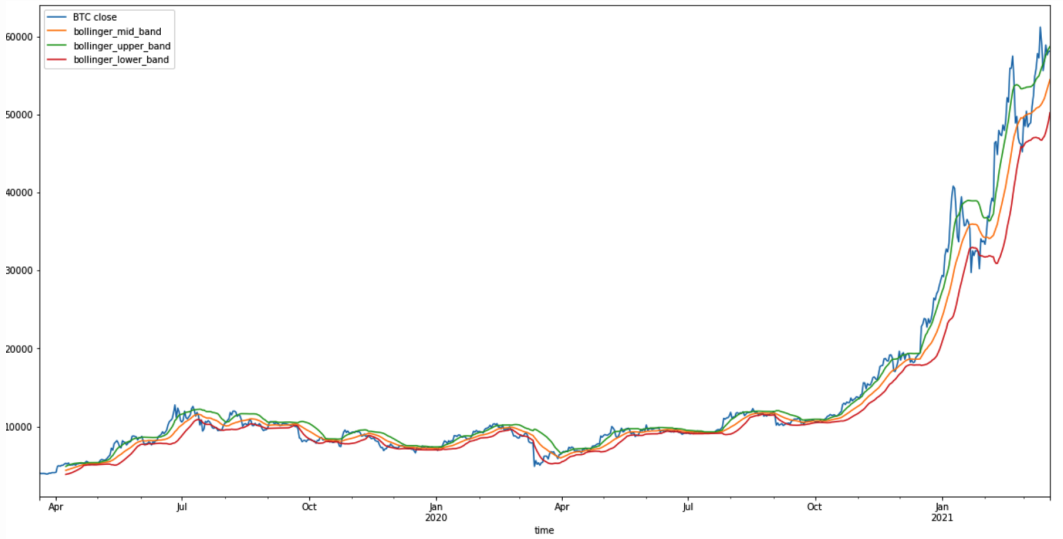
## Shrimpy Plots




sample AMA's (automates analytical model)




Sample Bollinger Bands

# Dashboard Demonstration

# Discussion

## 1/ What are our findings?

Reasons for traveling include recreation, or waging or fleeing war or for the enjoyment of traveling.


## 2/ Was the model good enough?

Travelers may use human-powered transport such as walking or bicycling; or vehicles, such as automobiles, trains and airplanes.

## 3/ What are the conclusions?

However, The French revolution brought with it the end of the Grand Tour.


# Postmortem

1 **What difficulties arose?**

Mainly dashboard issues. We used several iterations and ultimately ended up using Tableau.

2 **Additional questions or problems?**

No.

3 **What would we research next?**

We would expand our dashboard.


Questions?