Predicting the Stocks

By: Anna Kelley and Alejandro Campos

Topics

- The Problem
- The Goal
- Our Approach
- Conclusion

The Problem

- People lose lots of money in stocks
- One of the most difficult jobs is to analyze and predict the trend of a stock
- Market is extremely volatile, nearly everything online can affect its market price
- An algorithm can help people make passive income

The Goal

To program a software capable enough to take existing data from the market and utilize it to train itself and eventually predict future trends for any given stock

Our Approach

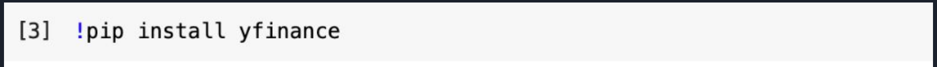
We’ll be utilizing Long Short-Term Memory method to train the data and accurately predict the future trends

LSTM is a a deep learning artificial recurrent neural network (RNN) architecture.

Unlike traditional feed-forward neural networks, LSTM has feedback connections. It can handle single data points (such as pictures) as well as full data sequences (such as speech or video).

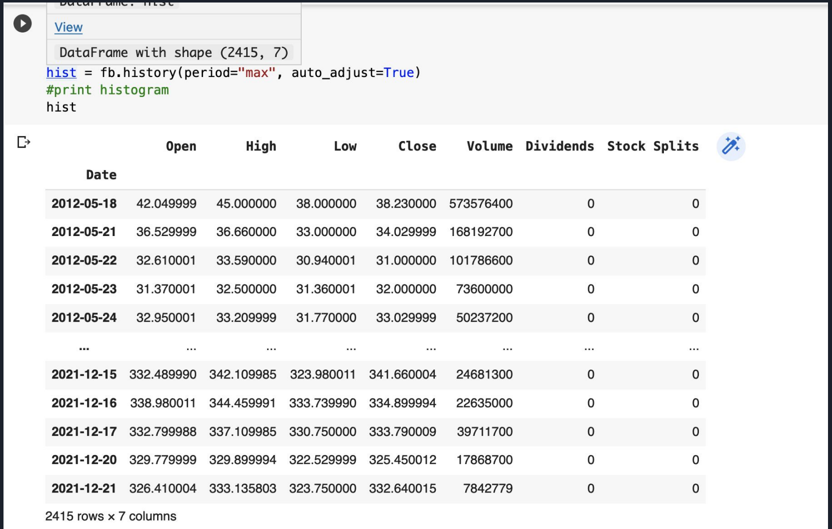
Prepare the data for analysis:

Download using yahoo finance:



```
[3]  !pip install yfinance
```

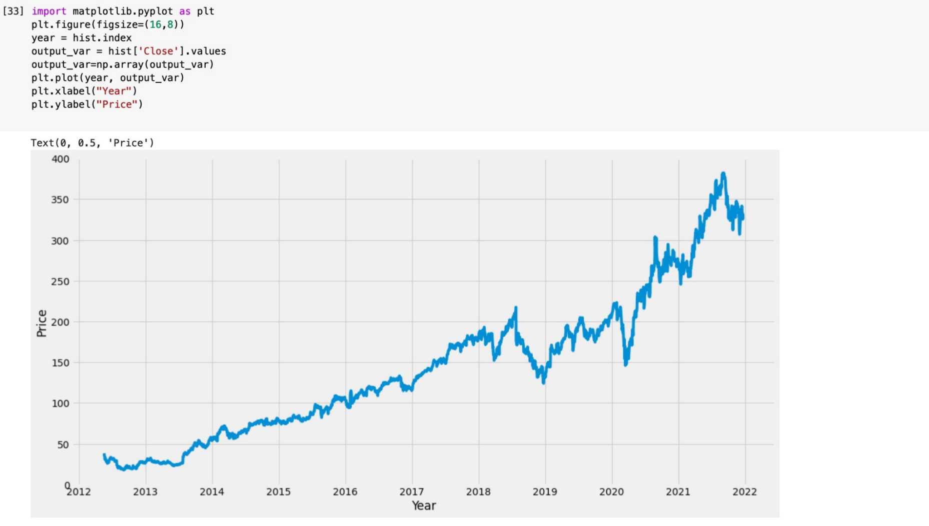
Raw Data:



```
DataFrame: hist
View
DataFrame with shape (2415, 7)
hist = fb.history(period="max", auto_adjust=True)
#print histogram
hist
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2012-05-18	42.049999	45.000000	38.000000	38.230000	573576400	0	0
2012-05-21	36.529999	36.660000	33.000000	34.029999	168192700	0	0
2012-05-22	32.610001	33.590000	30.940001	31.000000	101786600	0	0
2012-05-23	31.370001	32.500000	31.360001	32.000000	73600000	0	0
2012-05-24	32.950001	33.209999	31.770000	33.029999	50237200	0	0
...	...	...	...	...	...	...	...
2021-12-15	332.489990	342.109985	323.980011	341.660004	24681300	0	0
2021-12-16	338.980011	344.459991	333.739990	334.899994	22635000	0	0
2021-12-17	332.799988	337.109985	330.750000	333.790009	39711700	0	0
2021-12-20	329.779999	329.899994	322.529999	325.450012	17868700	0	0
2021-12-21	326.410004	333.135803	323.750000	332.640015	7842779	0	0

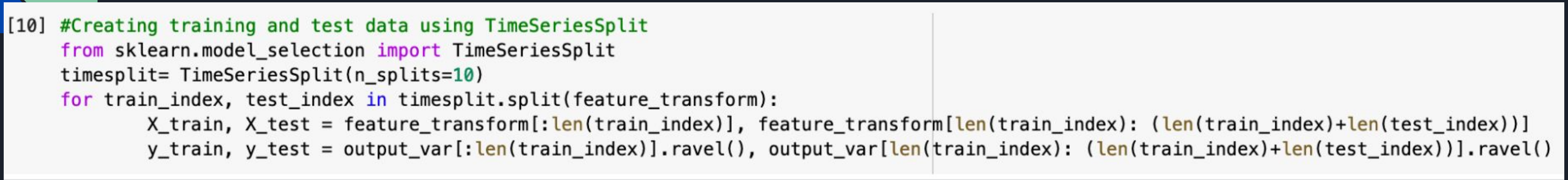
2415 rows × 7 columns



```
[33] import matplotlib.pyplot as plt
     plt.figure(figsize=(16,8))
     year = hist.index
     output_var = hist['Close'].values
     output_var=np.array(output_var)
     plt.plot(year, output_var)
     plt.xlabel("Year")
     plt.ylabel("Price")
```

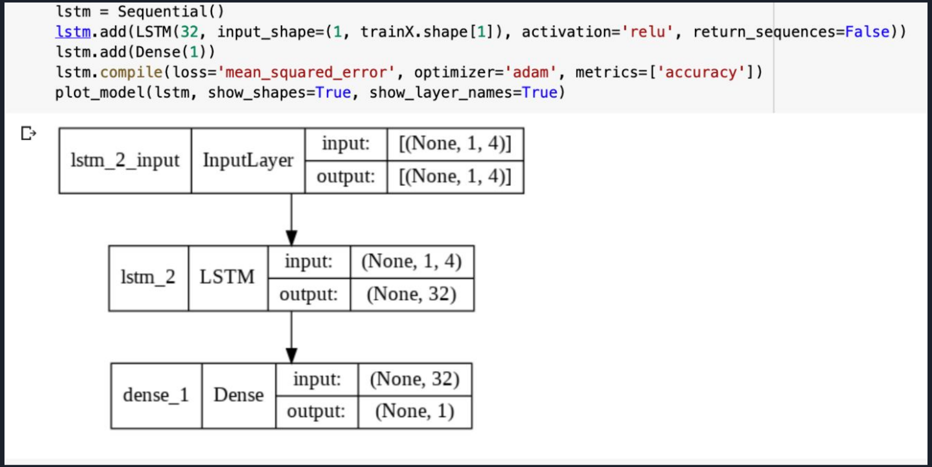
```
Text(0, 0.5, 'Price')
```

Split the data into training and test:



```
[10] #Creating training and test data using TimeSeriesSplit
     from sklearn.model_selection import TimeSeriesSplit
     timesplit= TimeSeriesSplit(n_splits=10)
     for train_index, test_index in timesplit.split(feature_transform):
             X_train, X_test = feature_transform[:len(train_index)], feature_transform[len(train_index): (len(train_index)+len(test_index))]
             y_train, y_test = output_var[:len(train_index)].ravel(), output_var[len(train_index): (len(train_index)+len(test_index))].ravel()
```

Building first LSTM model with one layer:



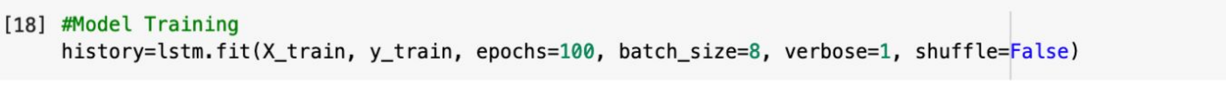
```
lstm = Sequential()
lstm.add(LSTM(32, input_shape=(1, trainX.shape[1]), activation='relu', return_sequences=False))
lstm.add(Dense(1))
lstm.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
plot_model(lstm, show_shapes=True, show_layer_names=True)
```

lstm_2_input | InputLayer | input: [(None, 1, 4)] | output: [(None, 1, 4)]

lstm_2 | LSTM | input: (None, 1, 4) | output: (None, 32)

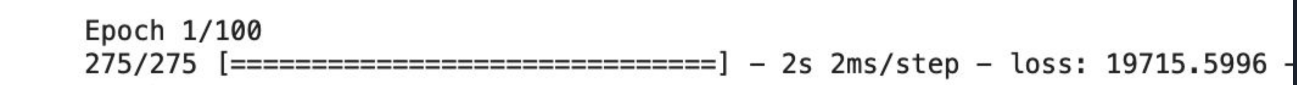
dense_1 | Dense | input: (None, 32) | output: (None, 1)

Training the data:

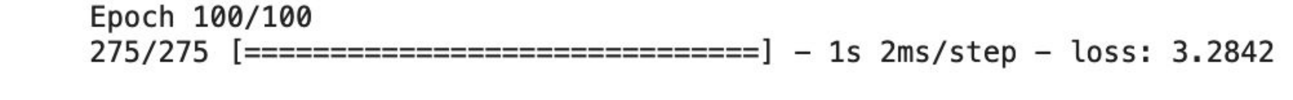


```
[18] #Model Training
     history=lstm.fit(X_train, y_train, epochs=100, batch_size=8, verbose=1, shuffle=False)
```

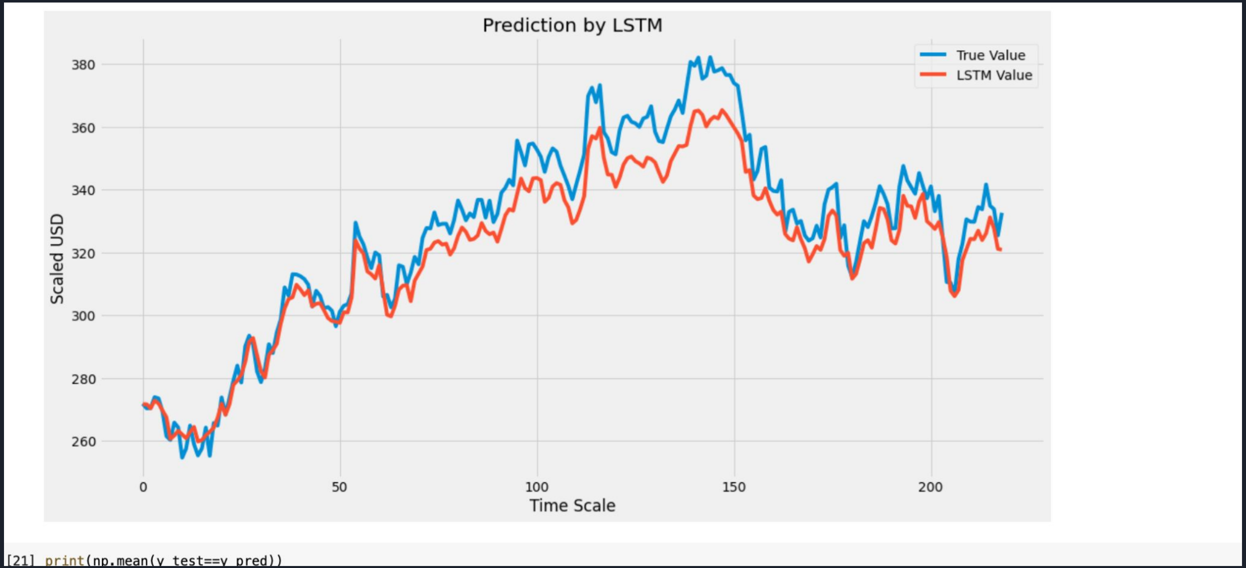
Loss



```
Epoch 1/100
275/275 [==============================] - 2s 2ms/step - loss: 19715.5996 -
```

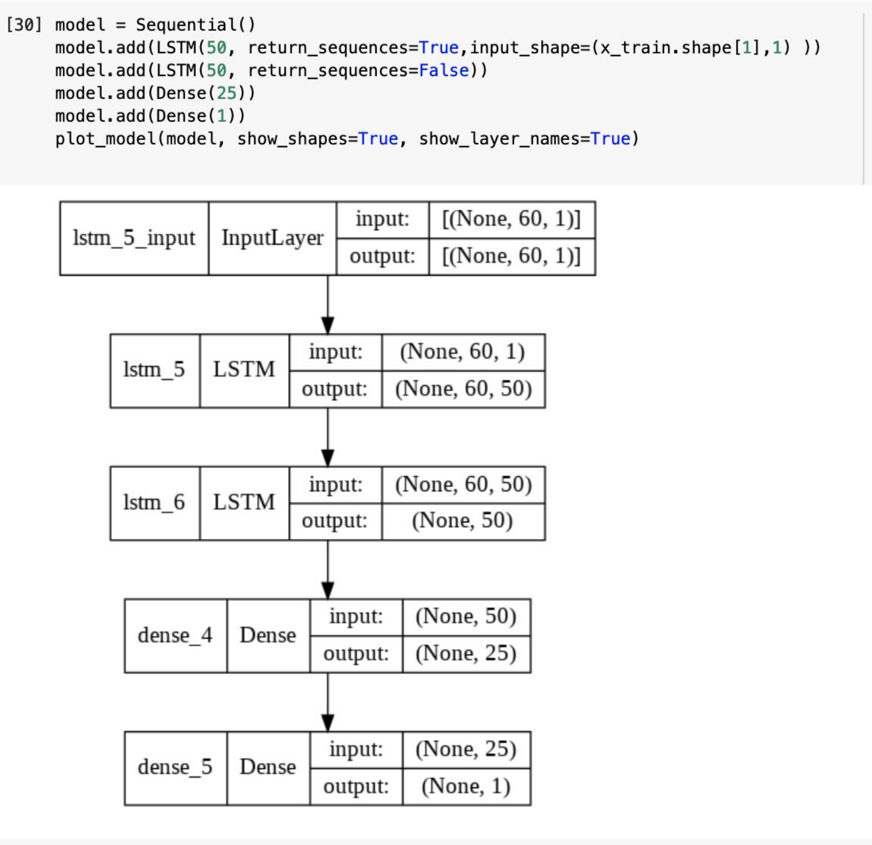


```
Epoch 100/100
275/275 [==============================] - 1s 2ms/step - loss: 3.2842
```



```
[21] print(np.mean(y_test==y_pred))
```

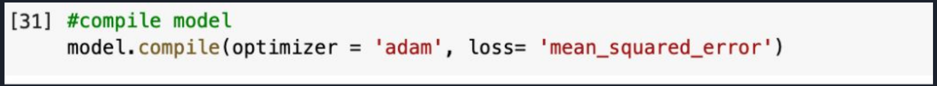
Building second LSTM model with multiple layers:



```
[30] model = Sequential()
     model.add(LSTM(50, return_sequences=True,input_shape=(x_train.shape[1],1) ))
     model.add(LSTM(50, return_sequences=False))
     model.add(Dense(25))
     model.add(Dense(1))
     plot_model(model, show_shapes=True, show_layer_names=True)
```

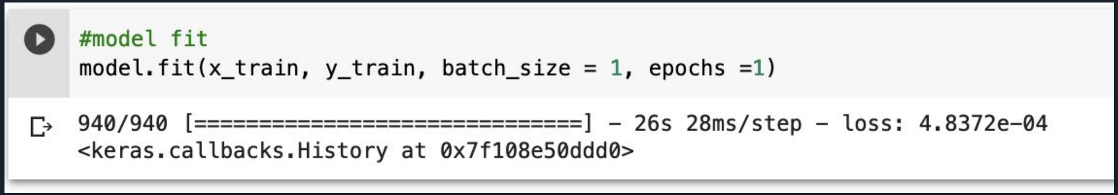
lstm_5_input	InputLayer	input:	[(None, 60, 1)]
		output:	[(None, 60, 1)]
lstm_5	LSTM	input:	(None, 60, 1)
		output:	(None, 60, 50)
lstm_6	LSTM	input:	(None, 60, 50)
		output:	(None, 50)
dense_4	Dense	input:	(None, 50)
		output:	(None, 25)
dense_5	Dense	input:	(None, 25)
		output:	(None, 1)

Training the data for second Model:



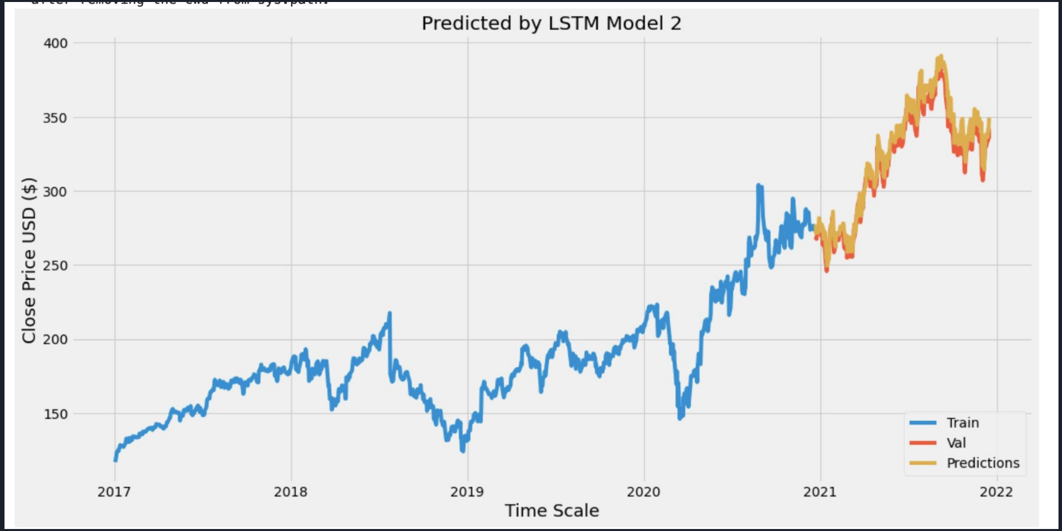
```
[31] #compile model
     model.compile(optimizer = 'adam', loss= 'mean_squared_error')
```

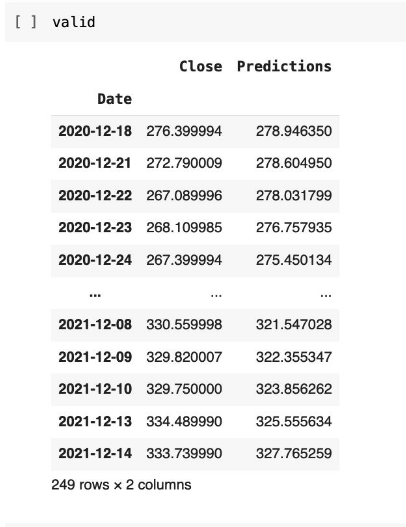
Smaller Loss



```
#model fit
model.fit(x_train, y_train, batch_size = 1, epochs =1)

940/940 [==============================] - 26s 28ms/step - loss: 4.8372e-04
<keras.callbacks.History at 0x7f108e50ddd0>
```





```
[ ] valid
```

	Close	Predictions
Date		
2020-12-18	276.399994	278.946350
2020-12-21	272.790009	278.604950
2020-12-22	267.089996	278.031799
2020-12-23	268.109985	276.757935
2020-12-24	267.399994	275.450134
...	...	...
2021-12-08	330.559998	321.547028
2021-12-09	329.820007	322.355347
2021-12-10	329.750000	323.856262
2021-12-13	334.489990	325.555634
2021-12-14	333.739990	327.765259

249 rows × 2 columns

Conclusion

- Over 50% of citizens in the US aren’t invested into the stock market
- All billionaires and other financially successful people are
- We want to change that = encourage others to invest in the stock market
- Easy-to-use software that can also help teach you about the market while making you passive income

References

1. https://www.analyticsvidhya.com/blog/2021/10/machine-learning-for-stock-market-prediction-with-step-by-step-implementation/
2. https://youtu.be/QIUxPv5PJOY
3. https://en.wikipedia.org/wiki/Long_short-term_memory
4. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6