ARIMA VS FBProphet

Objective

Challenge our existing model (fbprophet) and determine if ARIMA is more accurate

FBProphet Recap

What is FBProphet?

- Utilizes the additive regression model

$$y(t) = \underbrace{\boxed{g(t)}}_{\text{Trend}} + \underbrace{\boxed{s(t)}}_{\text{Seasonality}} + \underbrace{\boxed{h(t)}}_{\text{Holidays}} + \epsilon_t$$

- Trend g(t): models non-periodic changes.
- Seasonality s(t): represents periodic changes.
- Holidays component h(t): contributes information about holidays and events.

ARIMA

What is ARIMA?

- Short for ‘AutoRegressive Integrated Moving Average’
- Forecasting algorithm based on the idea that the information in the past values of the time series can alone be used to predict the future values.

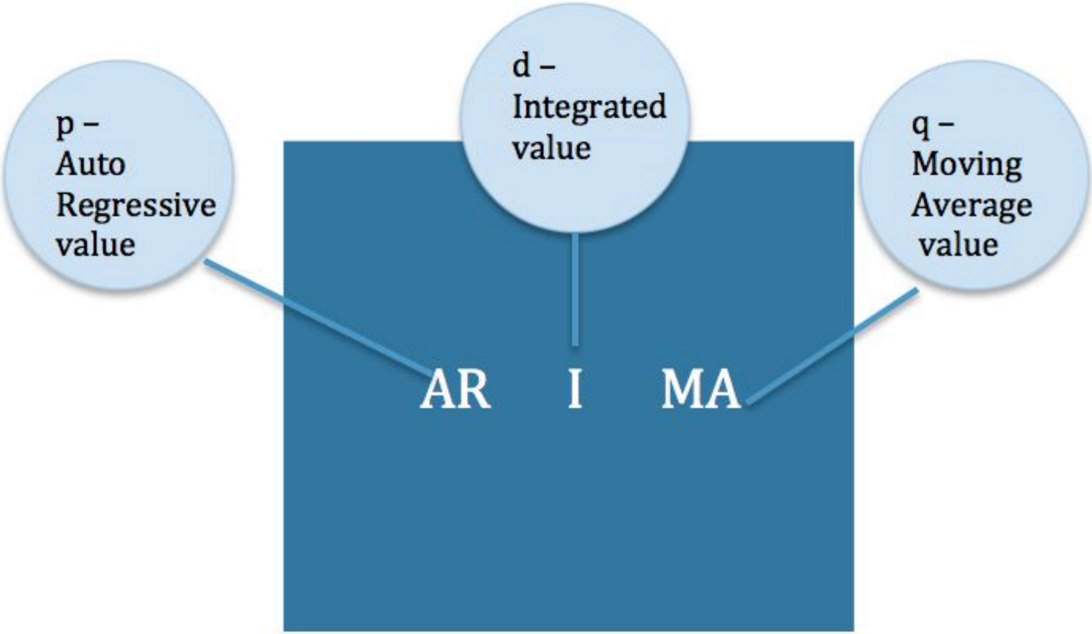
$$y'_t = \underset{\text{intercept}}{\boxed{c}} + \underset{\substack{\text{lags}\\ \text{(AR)}}}{\boxed{\varphi_1 y'_{t-1} + \dots + \varphi_p y'_{t-p}}} + \underset{\substack{\text{errors}\\ \text{(MA)}}}{\boxed{\theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t}}$$

Predicted Yt = Intercept + Lagged Values + Lagged Errors

- Lag features are target values from previous periods

ARIMA(p,d,q)

- p is the order of the AR term
- q is the order of the MA term
- d is the number of differencing required to make the time series stationary



PROPHET

“Change Points”

VS

Future values vs past values

METHODS

Step 1 — **Data Collection** — YFinance - public, updated daily
- Stock: MSFT
- Data length = 1 year

Step 2 — **Data Preparation** — Prophet - Drop, rename columns

ARIMA

Step 3 — **Predictions Model** — Prophet

ARIMA (q, d, p)

Step 4 — **Analysis** — Outcomes graphs, predicted/actual

Step 5 — **Model Accuracy** — Mean Absolute Percentage Score (Prophet vs ARIMA)

RESULTS

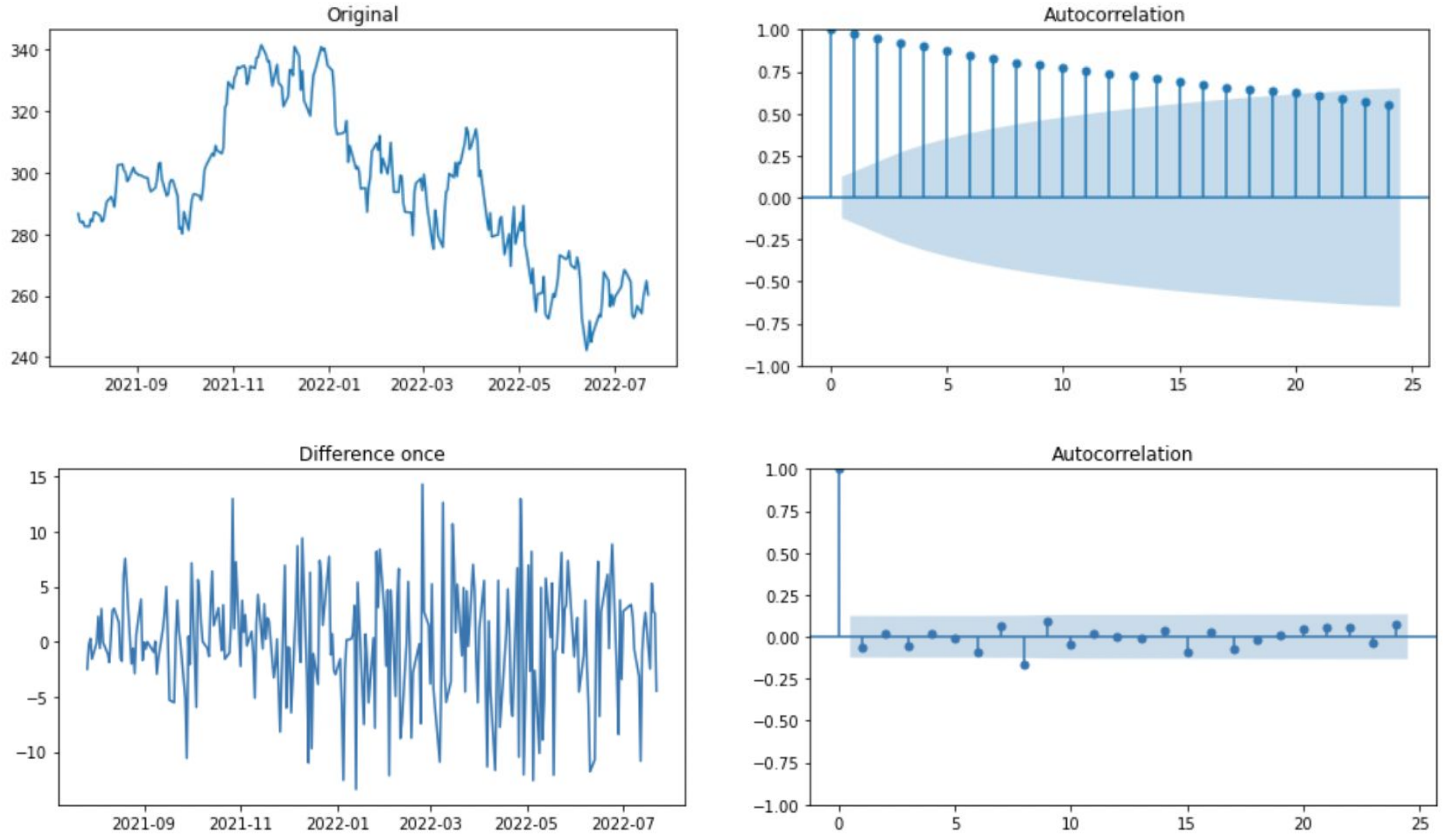
Is our data Stationary?
Augmented Dickey Fuller Test (ADF Test)

- If the p-value < 0.05 (stationary)
- If the p-value > 0.05 (non-stationary)

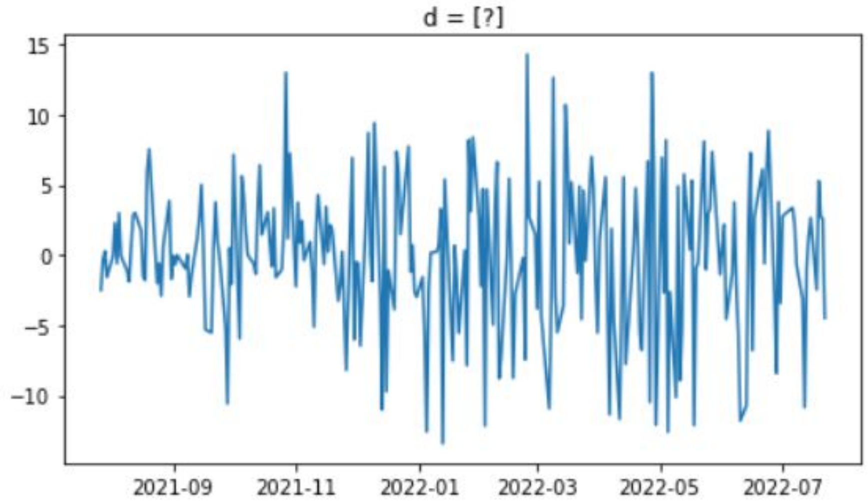
```
ADF Statistic: -1.4443417714750664
p-value: 0.5608433462424031
```

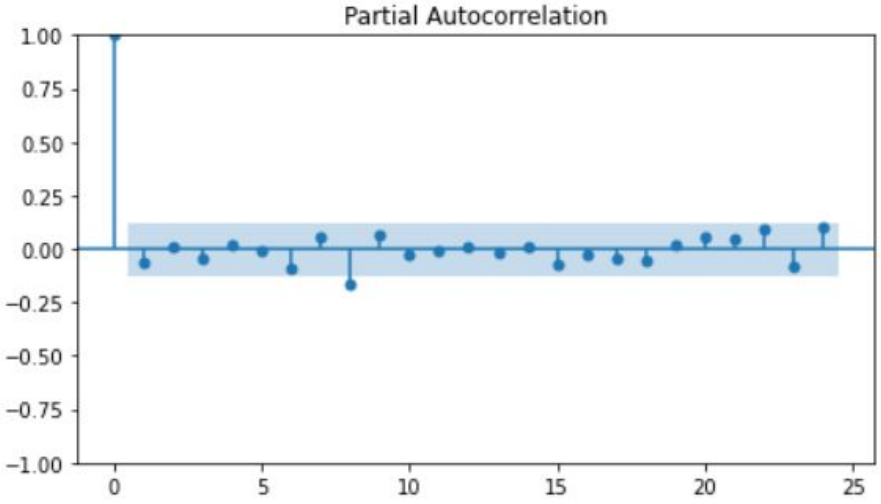
Non-Stationary > Find order of differencing (d)

d - number of differencing required to make the time series stationary

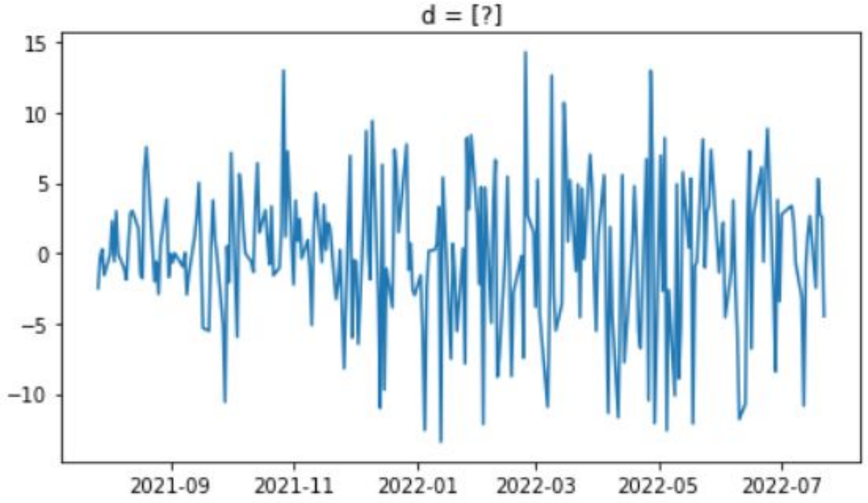


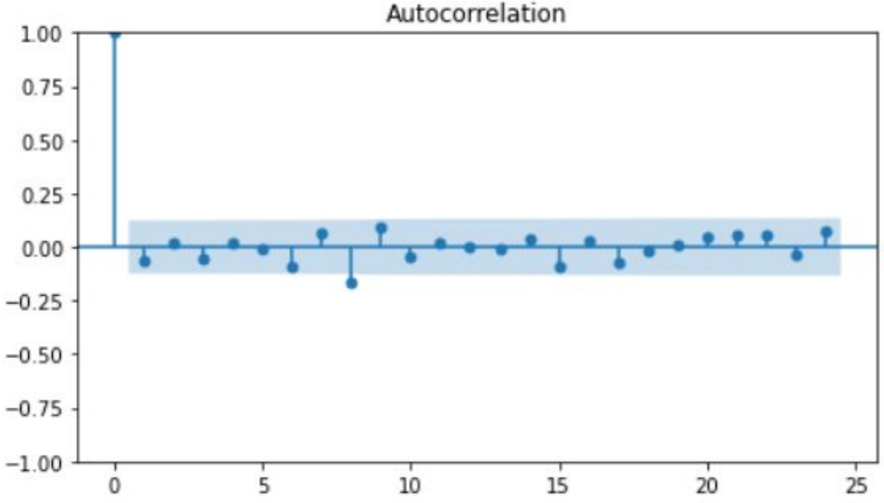
p - the order of the Autoregressive (AR) term





q -the order of the Moving Average (MA) term

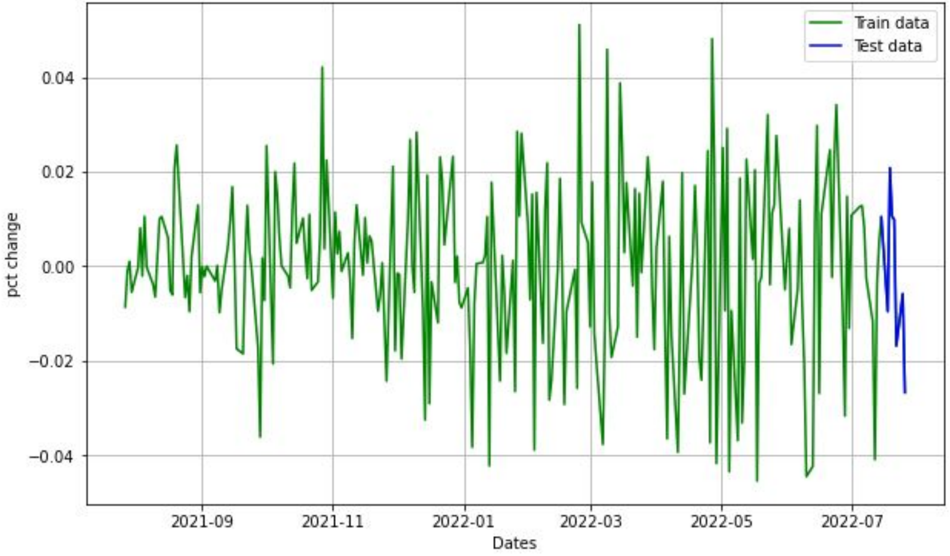




TESTING THE MODEL - Splitting test / train data

```
train_data, test_data = stock_df_arima_pctchange[3:int(len(stock_df_arima_pctchange)-days_test_data)], stock_df_arima_pctchange[int(len(stock_df_arima_pctchange)-days_test_data):]
```

7 days



1 year

TESTING THE MODEL - Applying ARIMA

```
model = ARIMA(train_data, order=(p, d, q))
result = model.fit()
result.summary()
```

0, 1, 0

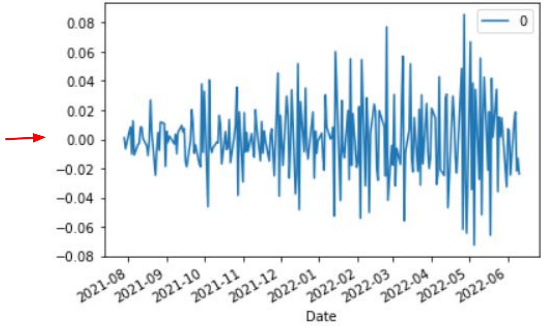
SARIMAX Results

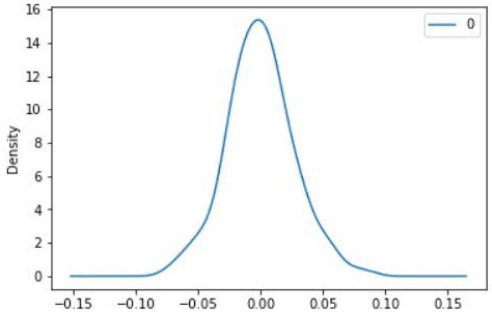
Dep. Variable:	pct change	**No. Observations:**	220
Model:	ARIMA(0, 1, 0)	**Log Likelihood**	481.908
Date:	Tue, 26 Jul 2022	**AIC**	-961.817
Time:	21:12:55	**BIC**	-958.428
Sample:	0	**HQIC**	-960.448
	- 220		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	0.0007	6.25e-05	11.486	0.000	0.001	0.001

Ljung-Box (L1) (Q):	69.53	**Jarque-Bera (JB):**	2.66
Prob(Q):	0.00	**Prob(JB):**	0.26
Heteroskedasticity (H):	4.66	**Skew:**	0.17
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.41

```
residuals = pd.DataFrame(result.resid)
residuals.plot()
residuals.plot(kind='kde')
```





TESTING THE MODEL - Organizing data

```
forecast = result.get_forecast(days_test_data)
```

test_data_close

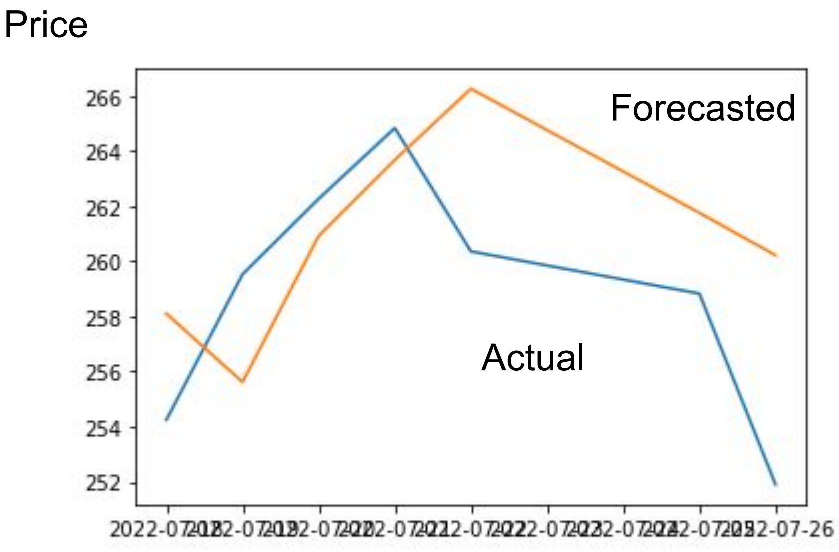
```
ci = forecast.conf_int()
```

date	predicted_mean	ower pct change	upper pct change	value	test data close	Lower CI	Upper CI
2022-06-13	-0.044564	-0.097086	0.007959	242.259995	NaN	NaN	NaN
2022-06-14	-0.044564	-0.118841	0.029714	244.490005	231.464011	213.469494	249.458527
2022-06-15	-0.044564	-0.135535	0.046408	251.759995	233.594644	211.353086	255.836202
2022-06-16	-0.044564	-0.149608	0.060481	244.970001	240.540656	214.094644	266.986668
2022-06-17	-0.044564	-0.162007	0.072880	247.649994	234.053250	205.283149	262.823350

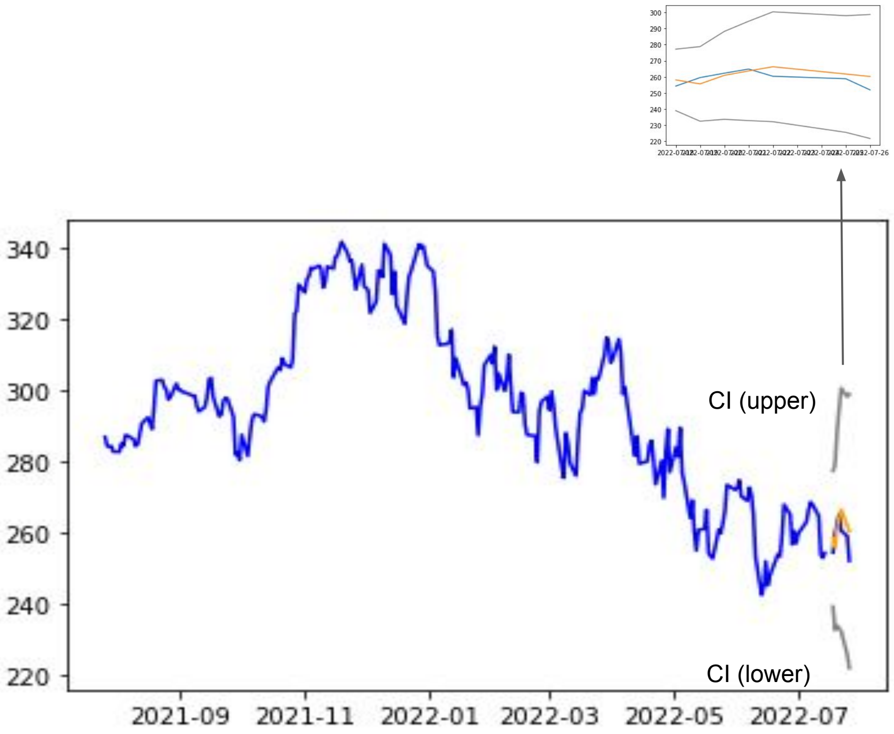
```
# Convert pct_change back to closing prices using formula:
# (1+pctchange) * prev closing price
```

TESTING THE MODEL -

Actual vs Forecasted prices [MSFT]



7days



1 year

Prophet vs ARIMA
- Accuracy Score Comparison
Past 7 days test data (7/18/21-7/26/21)

	ARIMA	PROPHET
mse	20.859511	64.606851
mae	3.917781	7.064014
rmse	4.567221	8.037839
mape	0.015252	0.028046

ARIMA model shows better performance!

CONCLUSION

Limitations

ARIMA	
conceptual limitations:	practical limitations:
data must be stationary	extra preprocessing steps
univariate	stock prices are not continuous - indices would convert to ints
continuous data	extra postprocessing steps
structural changes	
explainability	

Limitations

FBProphet Conceptual Limitations ● Weak/simple underlying assumptions ● Designed to deal with the types of problems FB faces ● External head-to-head studies show that FBP underperforms ARIMA, even when given more data.	Practical Limitations ● Fewer than ARIMA ● Weaker accuracy

NEXT STEPS

- Accuracy for different time horizons 1 week vs 1 month
- Feed model with more granular trading data like hourly or by minute
- Consider precision vs accuracy
- Perform future testing and compare models
- Consider other models to compare (neural network etc)
- Consider adding other factors - social media mentions, economy, sentiment, etc
- Consider comparing with seasoned trader
- Make program user friendly
  - Build AWS bot for better user experience

Contributors

Sungmoo Ban

Morgan Blackmore

Ryan Johnson

Jung Kim

Jennifer Taylor

Danica Valera

Questions?