STOCK PRICE PREDICTION USING HISTORICAL DATA AND TWITTER SENTIMENTS

San Jose State University | Department of Computer Science
Data Mining | CMPE-255 | Professor David C. Anastasiu
Fall 2017

TEAM MEMBERS

MANIKA KAPOOR

MASI NAZARIAN

SOWMYA GOWRISHANKAR

OUTLINE

- Introduction and Motivation
- Datasets
- Data Preprocessing
- Algorithms Used
- Evaluation
- Conclusion | Demo

INTRODUCTION

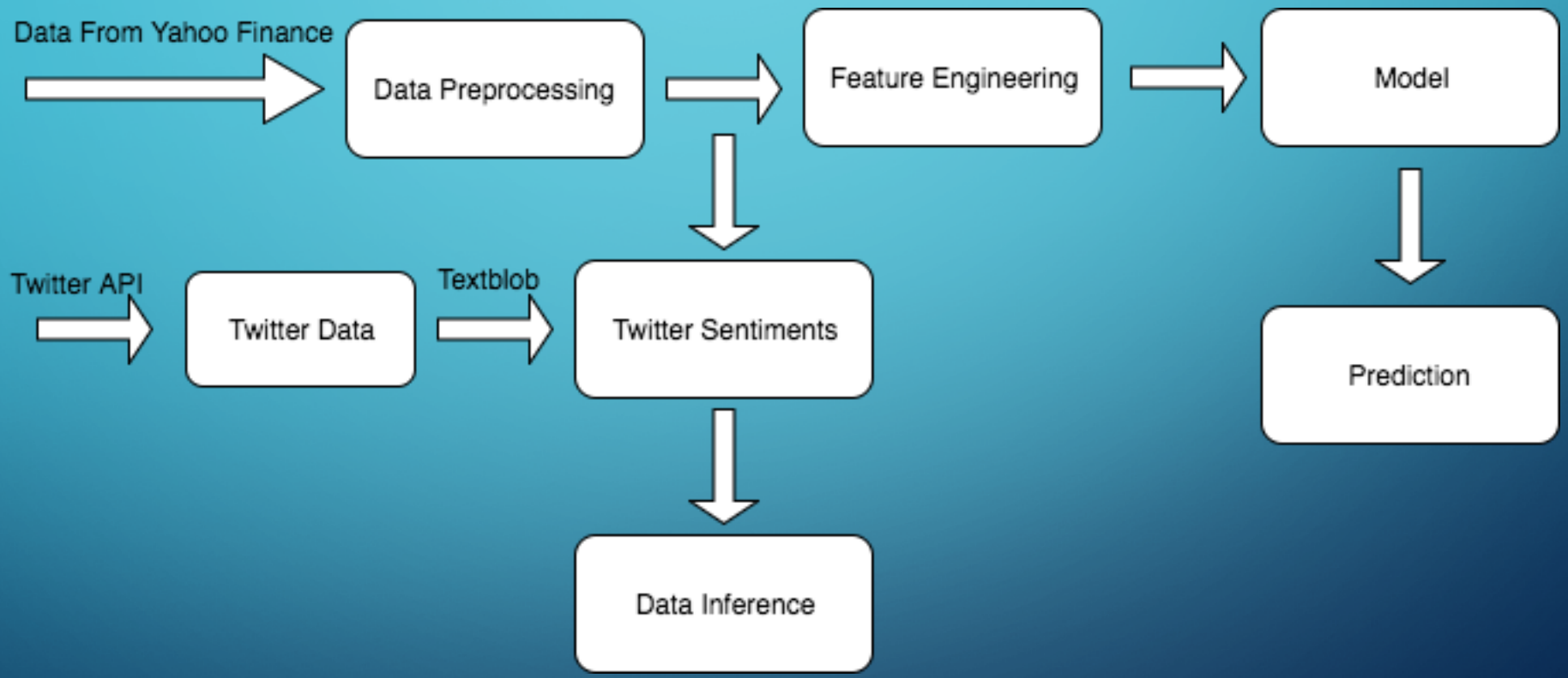
Stock Market Prediction

- Complex Problem involving seasonality/ trend in the data

Problems Addressed:

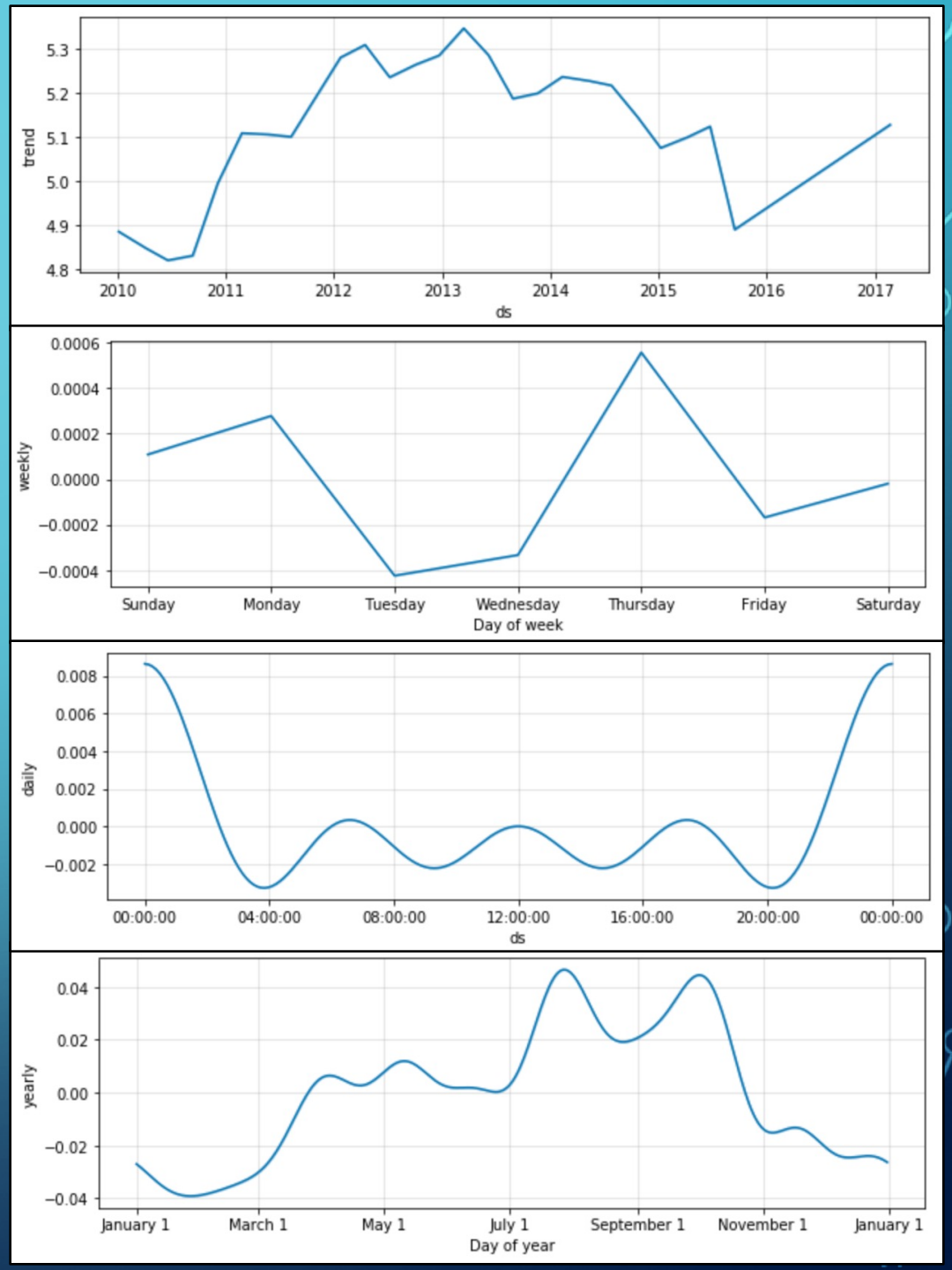
- Finding the best model to predict Stock Prices using historical price data
- Correlation between twitter sentiments and stock price fluctuation

FLOW DIAGRAM



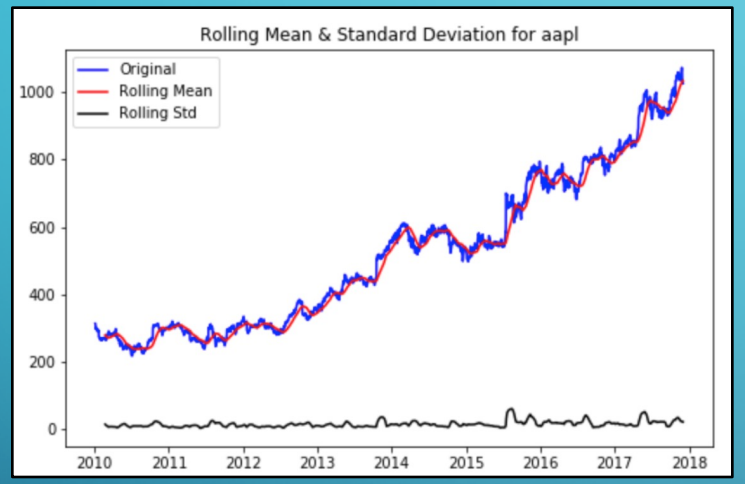
DATA EXPLORATION - EXTRACTING COMPONENTS

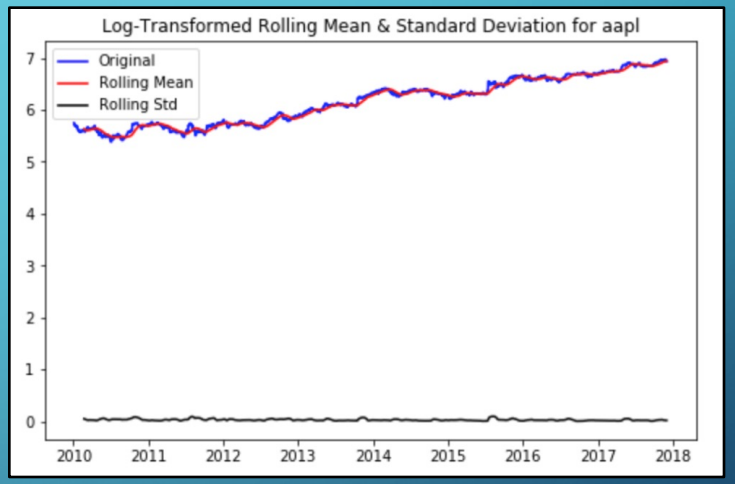
- Result from Facebook Prophet
- Time series components:
  - Overall Trend
  - Daily Seasonality
  - Weekly Seasonality
  - Yearly Seasonality

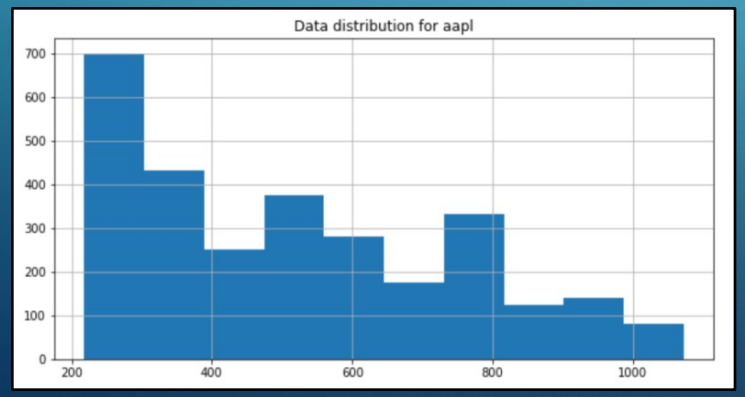


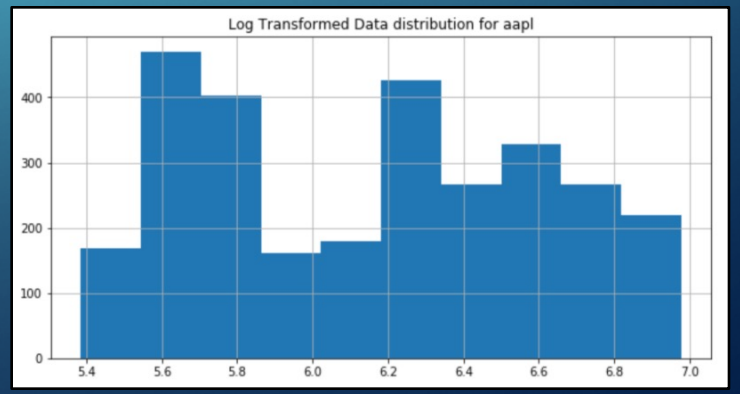
DATA EXPLORATION – CHECK NON-STATIONARITY

- Results from ARIMA
- Non-Stationarity Analysis







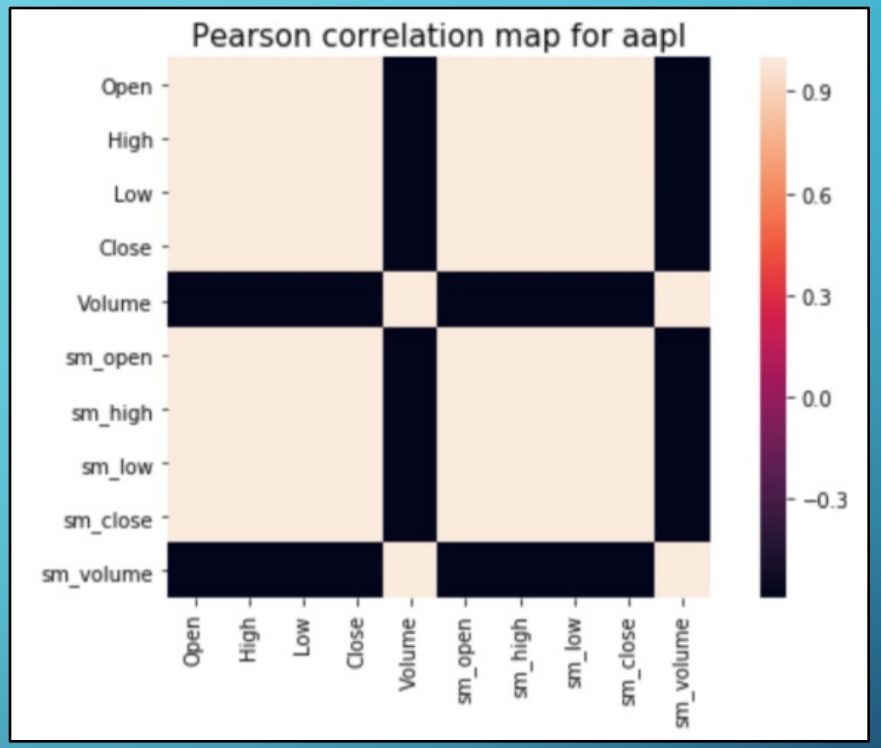


DATA PREPROCESSING

- Interpolating Missing Dates: Add rows and impute values for weekends
- Feature Selection: Remove Highly correlated Columns
- Feature Engineering: Add features derived from closing price
- Combine Datasets: Add features from S&P 500 to original stock data
- Non-Stationarity Removal: log transformation

FEATURE ENGINEERING

- Correlation matrix | heat map
- Open, High, Low, and Close are highly correlated
- Add derived features



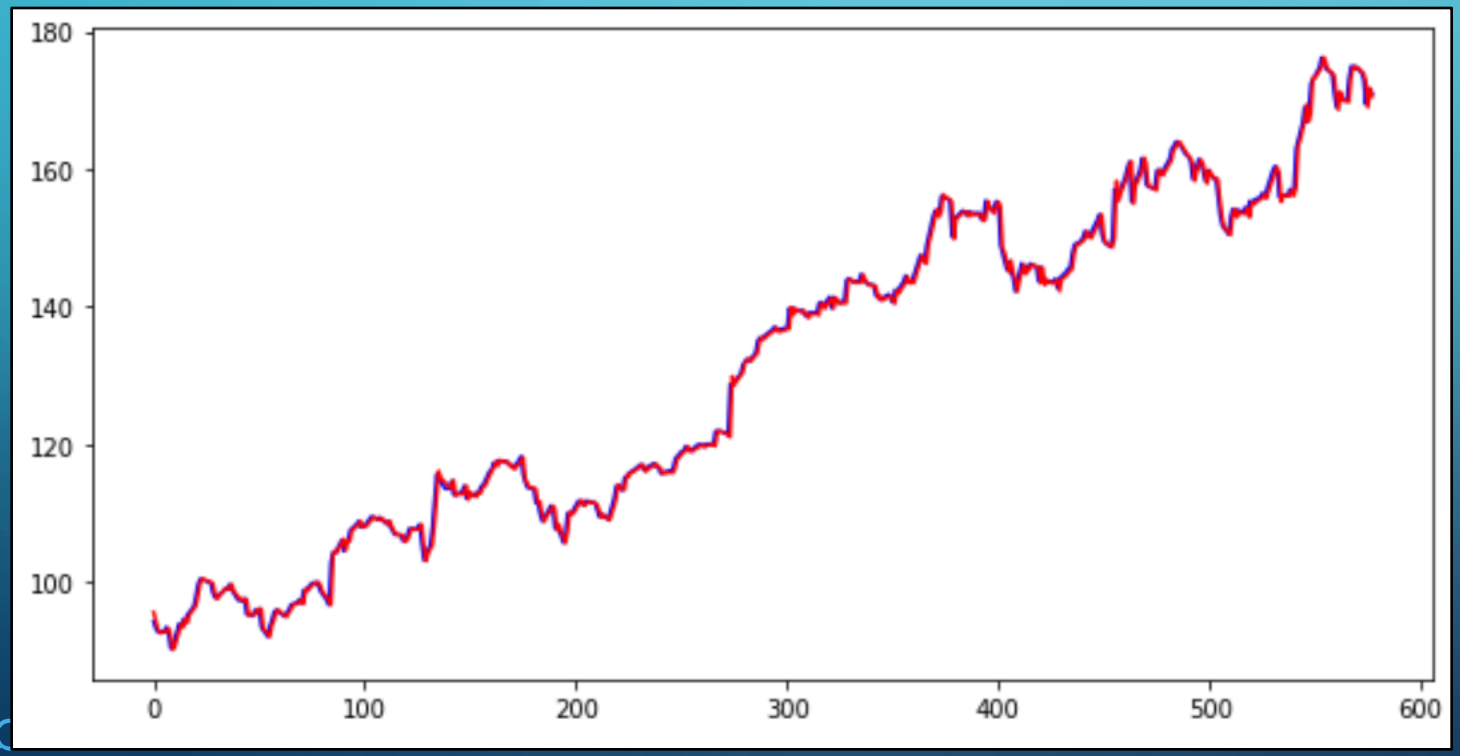
- Final Feature Set:

```
Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume', 'prev_diff',
       '50d', '10d_vol', 'sm_open', 'sm_high', 'sm_low', 'sm_close',
       'sm_adj_close', 'sm_volume', 'sm_prev_diff'],
      dtype='object')
```

LINEAR REGRESSION

```
linearReg = LinearRegression(normalize=False)
linearReg.fit(X_train_transform, y_train_transform)
predictions = linearReg.predict(X_test_transform)
```

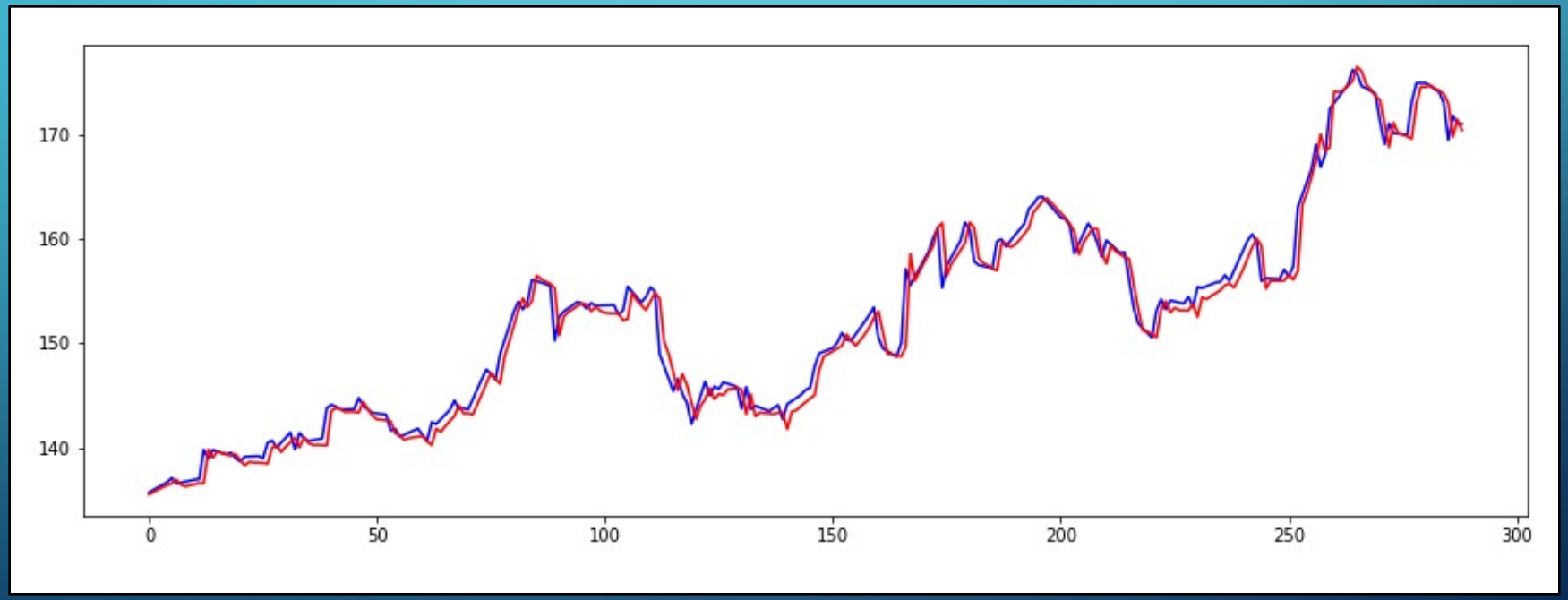
MSE: 1.430



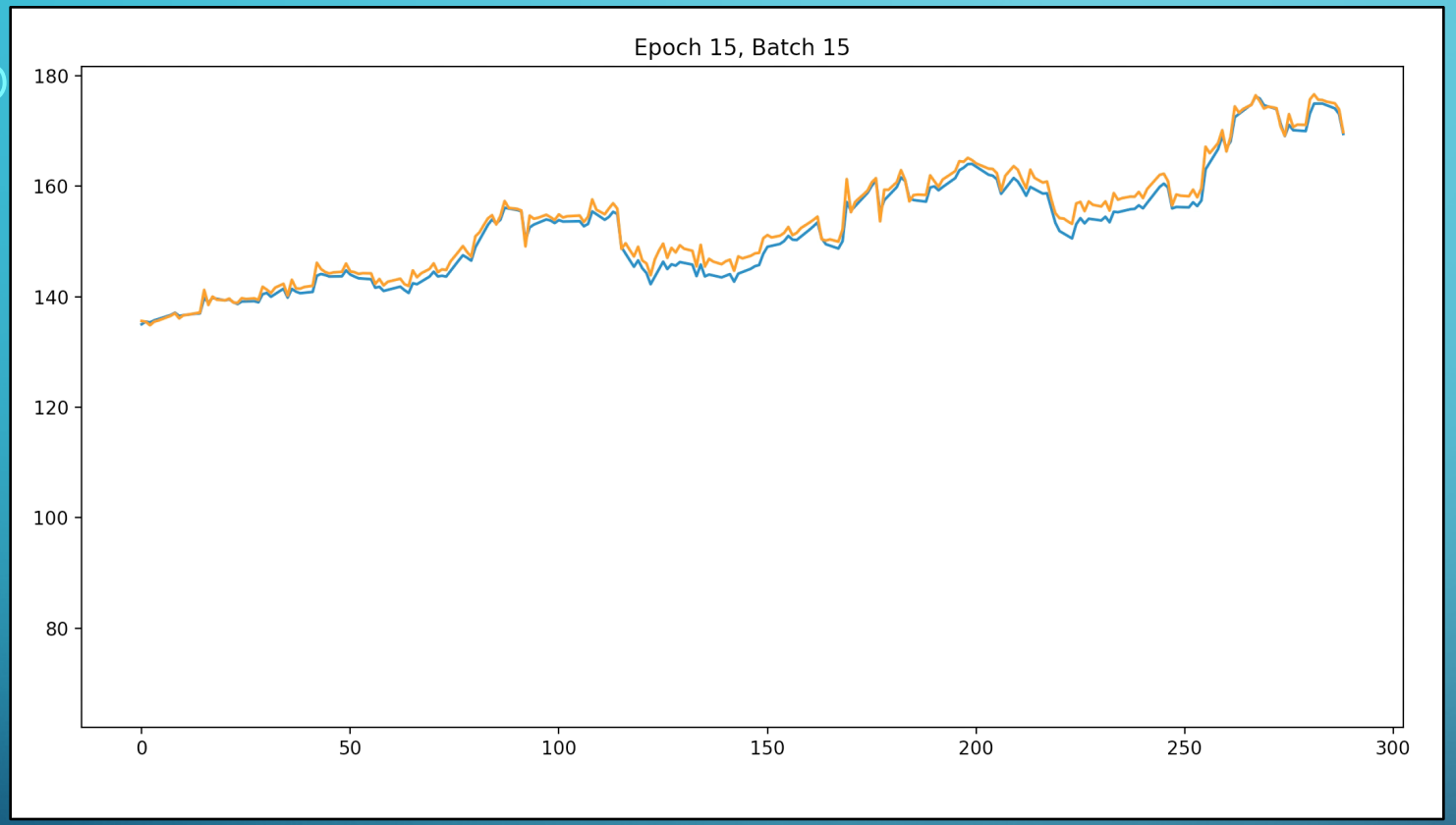
SUPPORT VECTOR REGRESSOR (SVR)

```
svm_linear = SVR(kernel='linear', C=2500)
```

MSE: 2.065



ARTIFICIAL NEURAL NETWORKS



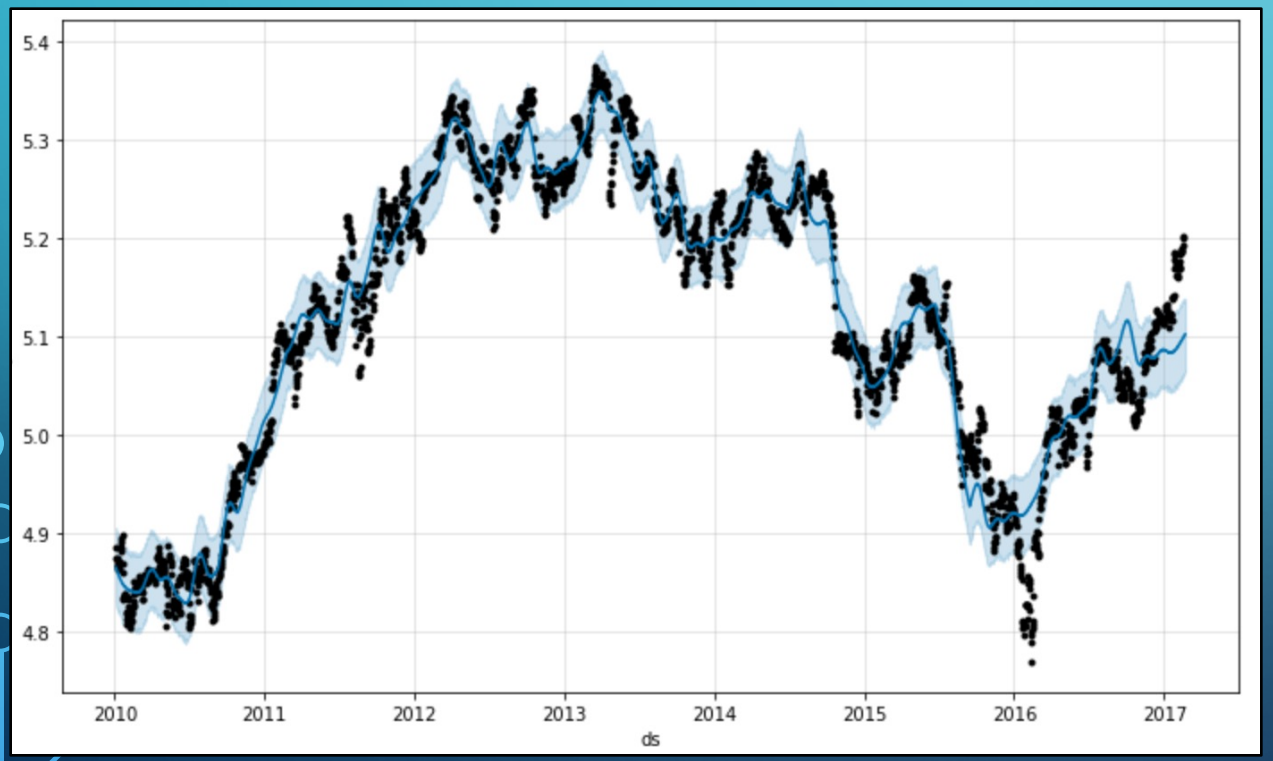
MSE: 0.638

```
# Model architecture parameters
numberOfFeatures = X_train.shape[1]
layer_nodes_1 = 1024
layer_nodes_2 = 512
layer_nodes_3 = 256
layer_nodes_4 = 128
layer_nodes_5 = 64
```

```
# Define all the hidden layers
layer_1 = tf.nn.relu(tf.add(tf.matmul(X, weight_layer_1), bias_layer_1))
layer_2 = tf.nn.relu(tf.add(tf.matmul(layer_1, weight_layer_2), bias_layer_2))
layer_3 = tf.nn.relu(tf.add(tf.matmul(layer_2, weight_layer_3), bias_layer_3))
layer_4 = tf.nn.relu(tf.add(tf.matmul(layer_3, weight_layer_4), bias_layer_4))
layer_5 = tf.nn.relu(tf.add(tf.matmul(layer_4, weight_layer_5), bias_layer_5))
```

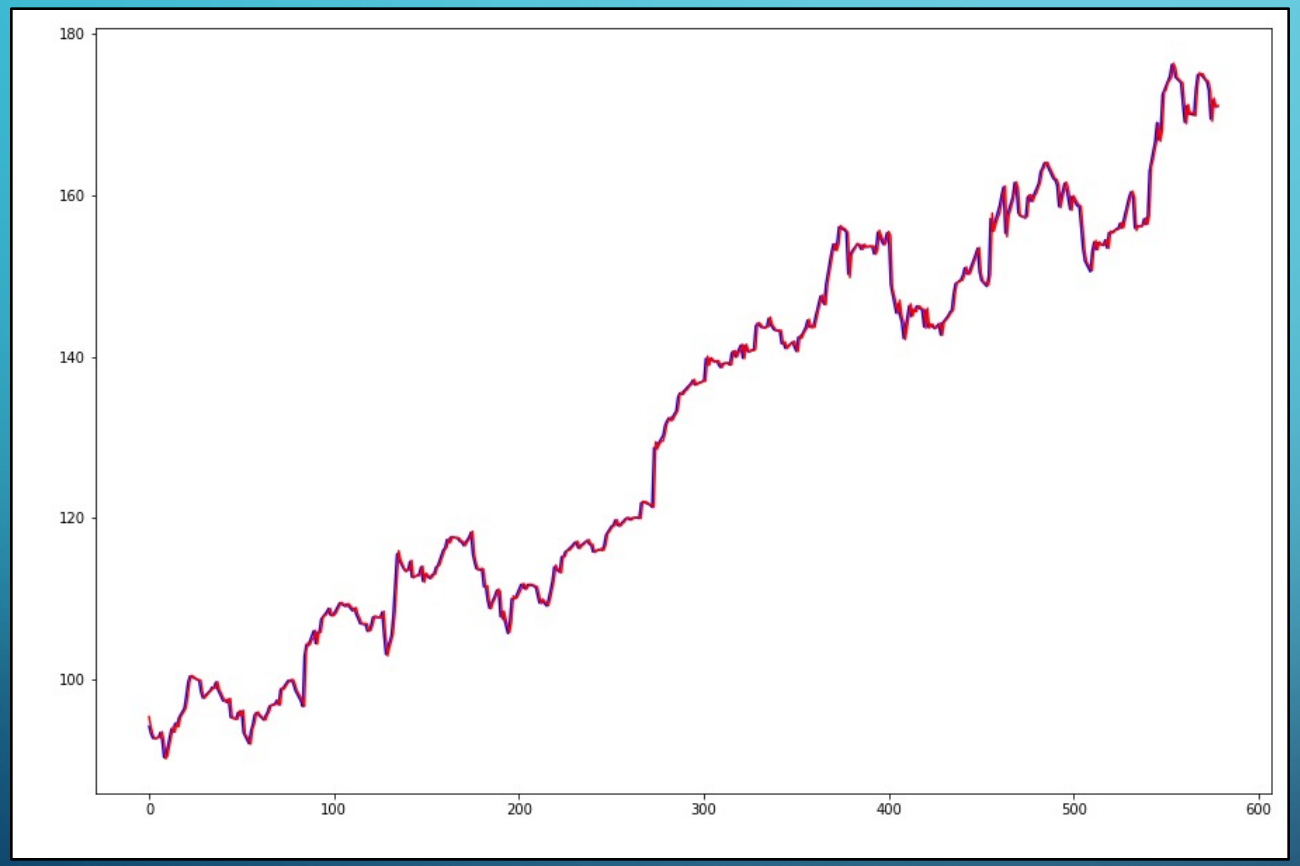
FACEBOOK PROPHET

```
dataProphetRed = dataProphet.rename(columns={"index": "ds", "Close": "y"})
dataProphetRed['y_orig'] = dataProphetRed['y']

#log transform y
dataProphetRed['y'] = np.log(dataProphetRed['y'])

splitIndex = int(np.floor(dataProphetRed.shape[0]*0.95))
X_train_prophet, X_test_prophet = dataProphetRed[:splitIndex], dataProphetRed[splitIndex:]

model=Prophet(daily_seasonality=True)
model.fit(X_train_prophet)

future_data = model.make_future_dataframe(periods=30)
forecast_data = model.predict(future_data)
```



MSE: 1639.284

ARIMA

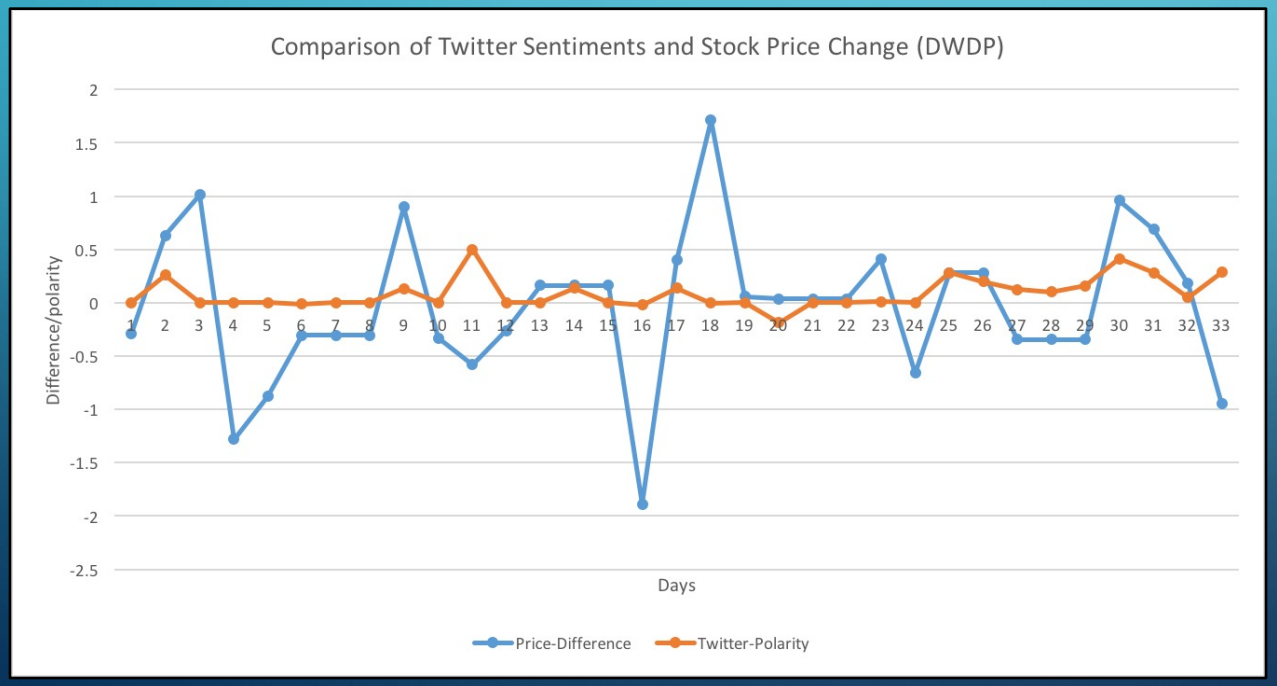


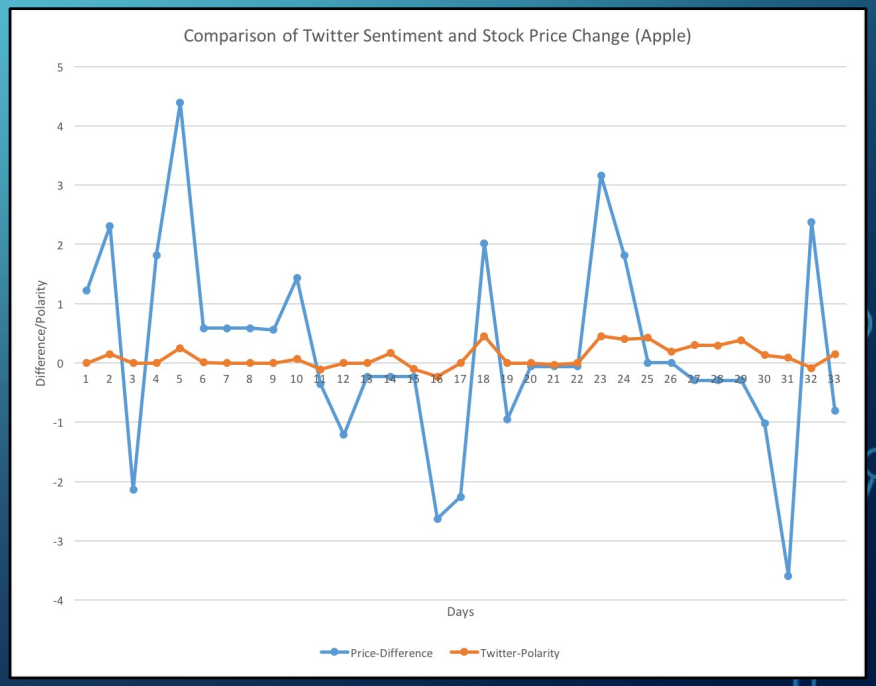
MSE: 1.493

```
model = ARIMA(history, order=(5,1,0))
model_fit = model.fit(disp=0)
output = model_fit.forecast()
```

INFERENCE – TWITTER SENTIMENT ANALYSIS

- Stock price time series showed considerable correlation with Twitter sentiments
- Twitter sentiments can be used as extra feature in future to improve performance





RESULTS

- ANN yielded the best performance, followed closely by ARIMA

	ARIMA	FB Prophet	ANN	SVR	LR
AAPL	1.493	1639.284	0.638	2.065	1.430
GOOGL	48.482	11647.023	43.5441	83.290	44.347
INTU	1.241	79.222	0.357	1.804	1.177
IBM	1.544	1143.301	3.77034	1.779	1.667
DWDP	0.265	19.465	0.160506	0.334	0.249
TOT	0.139	4.837	0.00597506	0.132	0.208

CONCLUSION

- Adding the new features i.e. 10 day volatility, 50 day moving average, difference helped
- Combining the features from S&P dataset improved performance
- Interpolation of data for weekends helped in maintaining time series data continuity
- Stock price time series showed considerable correlation with Twitter sentiments, so it can be used as a feature for Stock Price Prediction
- Twitter data was found to be insufficient to be used as a feature for price prediction and hence, was instead used for inference