Project 2:Forecasting Amazon Stock Prices

Sauda Haywood

2025-01-26

Step 1: Load the Data

```
# Ensure Date format
df <- df %>%
  mutate(Date = mdy(Date)) %>%  # Convert Date column to Date type
  arrange(Date)                # Sort data by Date

# View the first few rows
head(df)
```

```
## # A tibble: 6 x 7
##   Date       Close    Volume  Open  High   Low  VWAP
##   <date>     <dbl>     <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2020-01-27  91.4  70570180  91    92.0  90.8  142.
## 2 2020-01-28  92.7  56160800  92.0  92.9  91.5  142.
## 3 2020-01-29  92.9  42027800  93.2  93.7  92.8  142.
## 4 2020-01-30  93.5 126548760  92.9  93.6  92.5  142.
## 5 2020-01-31 100.  311345600 103.  103.  100.   143.
## 6 2020-02-03 100.  117981880 101.  102.  100.   143.
```

Step 2: Feature Engineering, Create rolling means and standard deviations for lag features

```
# Add rolling statistics
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lag_features <- c("Close", "Volume", "Open", "High", "Low")
for (feature in lag_features) {
  df <- df %>%
    mutate(
      !!paste0(feature, "_rolling_mean_3") := rollmean(get(feature), k = 3, fill = NA, align = "right")
      !!paste0(feature, "_rolling_mean_7") := rollmean(get(feature), k = 7, fill = NA, align = "right")
      !!paste0(feature, "_rolling_std_3")  := rollapply(get(feature), width = 3, FUN = sd, fill = NA, al
      !!paste0(feature, "_rolling_std_7")  := rollapply(get(feature), width = 7, FUN = sd, fill = NA, al
    )
```

```r
}

# Drop rows with NA values
df <- na.omit(df)
```

Step 3: Train-Test Split

```r
# Split the data into training (80%) and testing (20%)
train_size <- floor(0.8 * nrow(df))
training_data <- df[1:train_size, ]
test_data <- df[(train_size + 1):nrow(df), ]
```

Step 4: Time Series Modeling with ARIMA,

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3
```

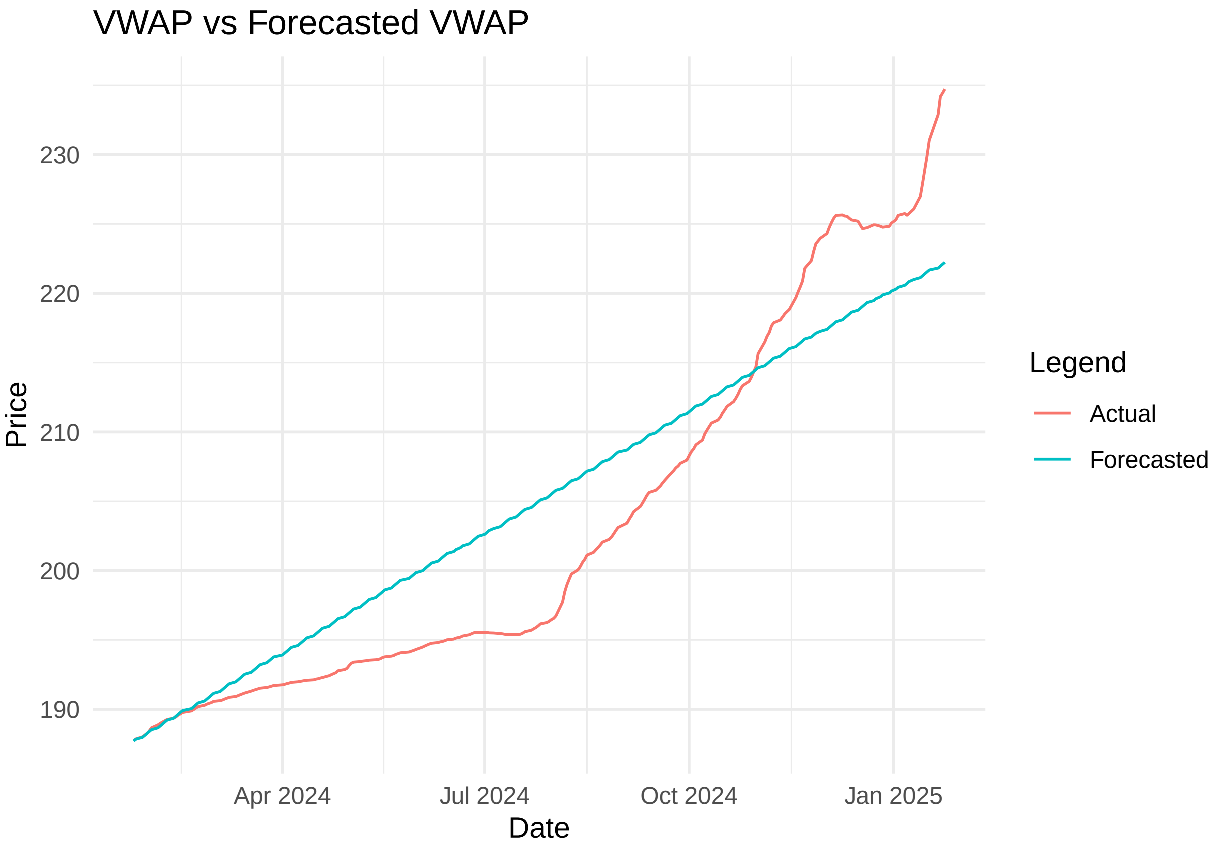
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Fit ARIMA model with auto.arima
arima_model <- auto.arima(training_data$VWAP)

# Forecast the VWAP for the test period
forecast_values <- forecast(arima_model, h = nrow(test_data))

# Add forecasts to test data
test_data <- test_data %>%
  mutate(Forecast_ARIMA = as.numeric(forecast_values$mean))

# View the forecasts
head(test_data)
```

```
## # A tibble: 6 x 28
##   Date       Close   Volume  Open  High   Low  VWAP Close_rolling_mean_3
##   <date>     <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>                <dbl>
## 1 2024-01-25  158. 43638590  157.  159.  155.  188.                 157.
## 2 2024-01-26  159. 51047350  158.  161.  158.  188.                 158.
## 3 2024-01-29  161. 45270390  159.  161.  159.  188.                 159.
## 4 2024-01-30  159  45207430  161.  162.  158.  188.                 160.
## 5 2024-01-31  155. 50284370  157   159.  155.  188.                 158.
## 6 2024-02-01  159. 76542420  156.  160.  156.  188.                 158.
## # i 20 more variables: Close_rolling_mean_7 <dbl>, Close_rolling_std_3 <dbl>,
## #   Close_rolling_std_7 <dbl>, Volume_rolling_mean_3 <dbl>,
## #   Volume_rolling_mean_7 <dbl>, Volume_rolling_std_3 <dbl>,
## #   Volume_rolling_std_7 <dbl>, Open_rolling_mean_3 <dbl>,
## #   Open_rolling_mean_7 <dbl>, Open_rolling_std_3 <dbl>,
## #   Open_rolling_std_7 <dbl>, High_rolling_mean_3 <dbl>,
## #   High_rolling_mean_7 <dbl>, High_rolling_std_3 <dbl>, ...
```

Step 5: Plot Actual vs Forecasted Values

```
# Plot VWAP vs Forecast
library(ggplot2)

ggplot(test_data, aes(x = Date)) +
  geom_line(aes(y = VWAP, color = "Actual")) +
  geom_line(aes(y = Forecast_ARIMA, color = "Forecasted")) +
  labs(title = "VWAP vs Forecasted VWAP",
       x = "Date",
       y = "Price",
       color = "Legend") +
  theme_minimal()
```



#

Step 6: Evaluate the Model

```
# Calculate RMSE and MAE
rmse <- sqrt(mean((test_data$VWAP - test_data$Forecast_ARIMA)^2, na.rm = TRUE))
mae <- mean(abs(test_data$VWAP - test_data$Forecast_ARIMA), na.rm = TRUE)

cat("RMSE:", rmse, "\n")
```

```
## RMSE: 5.133288
```

```
cat("MAE:", mae, "\n")
```

```
## MAE: 4.329058
```