Code

R Notebook for Time Series Forecasting of Stock Prices using ARIMA Model

Name: SANTHOSH PAANDIPATTI
Student Id: 11649125

Here is some information about the dataset:

Content: The dataset includes daily stock price data for the S&P 500 companies, covering a period from 2000 to 2019. It provides information such as the date, opening price, closing price, highest price, lowest price, and trading volume for each stock.

Number of Records: The dataset contains over 5 million records, with each record representing the stock price information for a specific company on a specific date.

Attributes: The dataset includes the following attributes:

Date: The date of the stock price data.

Open: The opening price of the stock on that date.

High: The highest price reached by the stock on that date.

Low: The lowest price reached by the stock on that date.

Close: The closing price of the stock on that date.

Volume: The trading volume of the stock on that date.

Name: The ticker symbol representing the company's stock.

Purpose: This dataset can be used for various purposes, such as analyzing stock market trends, building predictive models, backtesting trading strategies, and conducting research on the performance of individual stocks or the market as a whole.

Data Source: The dataset was compiled using historical stock price data from Yahoo Finance.

Hide

```
# Install and load necessary packages
install.packages("tidyverse")
install.packages("forecast")
library(tidyverse)
library(forecast)
```

Hide

```
# Step 1: Load the dataset
data <- read.csv("E:/US College/New folder/Predictive Analytics/Class Activity/Project/all_st
ocks_5yr.csv")
```

Hide

```
# Step 2: Preprocess the data
# Assuming you want to focus on a specific stock, let's say "AAPL"
stock_data <- data %>% filter(Name == "AAPL")
```

Hide

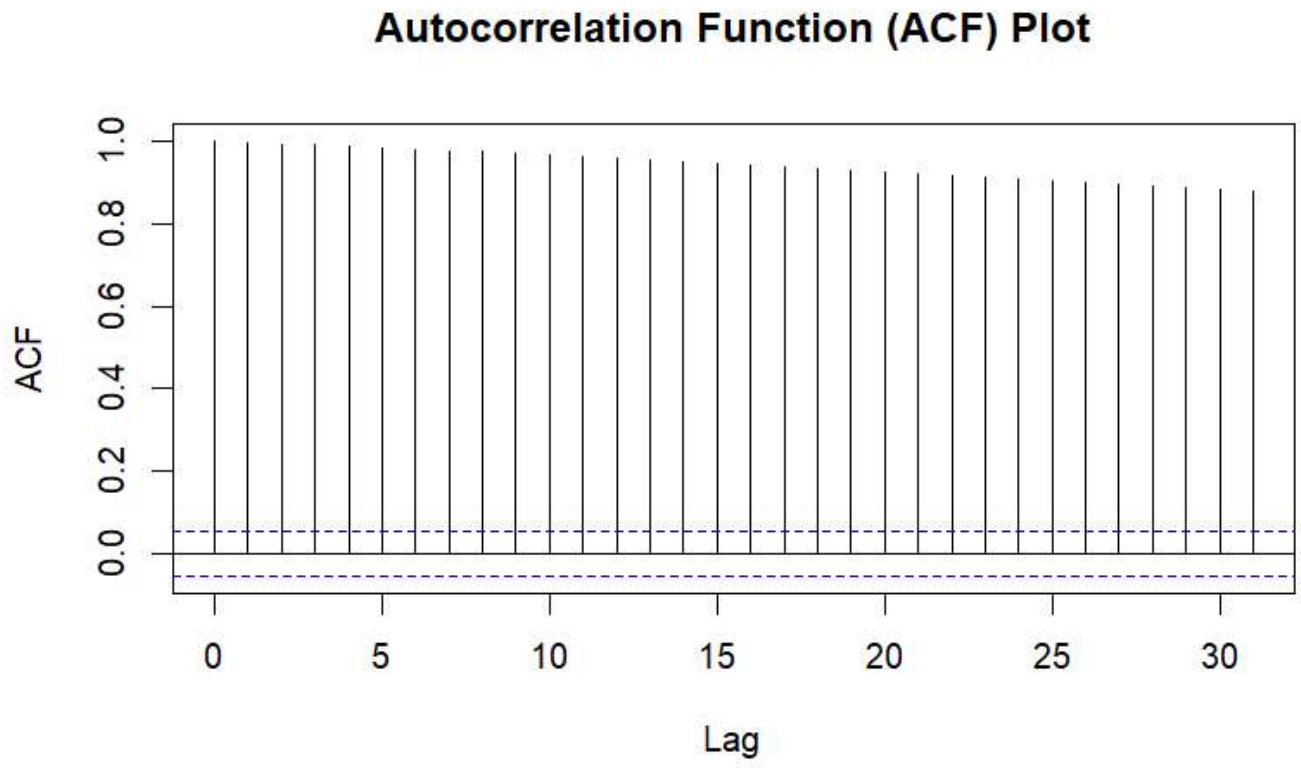
```
# Step 3: Explore the data
# Plot the time series of stock prices
ggplot(stock_data, aes(date, close)) +
  geom_line() +
  labs(x = "Date", y = "Stock Price", title = "AAPL Stock Price Time Series")
```

Hide

```
# Summarize the findings and discuss the results
# Step 4.1: Time series modeling
# Assuming an ARIMA(1,0,1) model
model <- auto.arima(stock_data$close)
ts_data <- ts(stock_data$close)
```

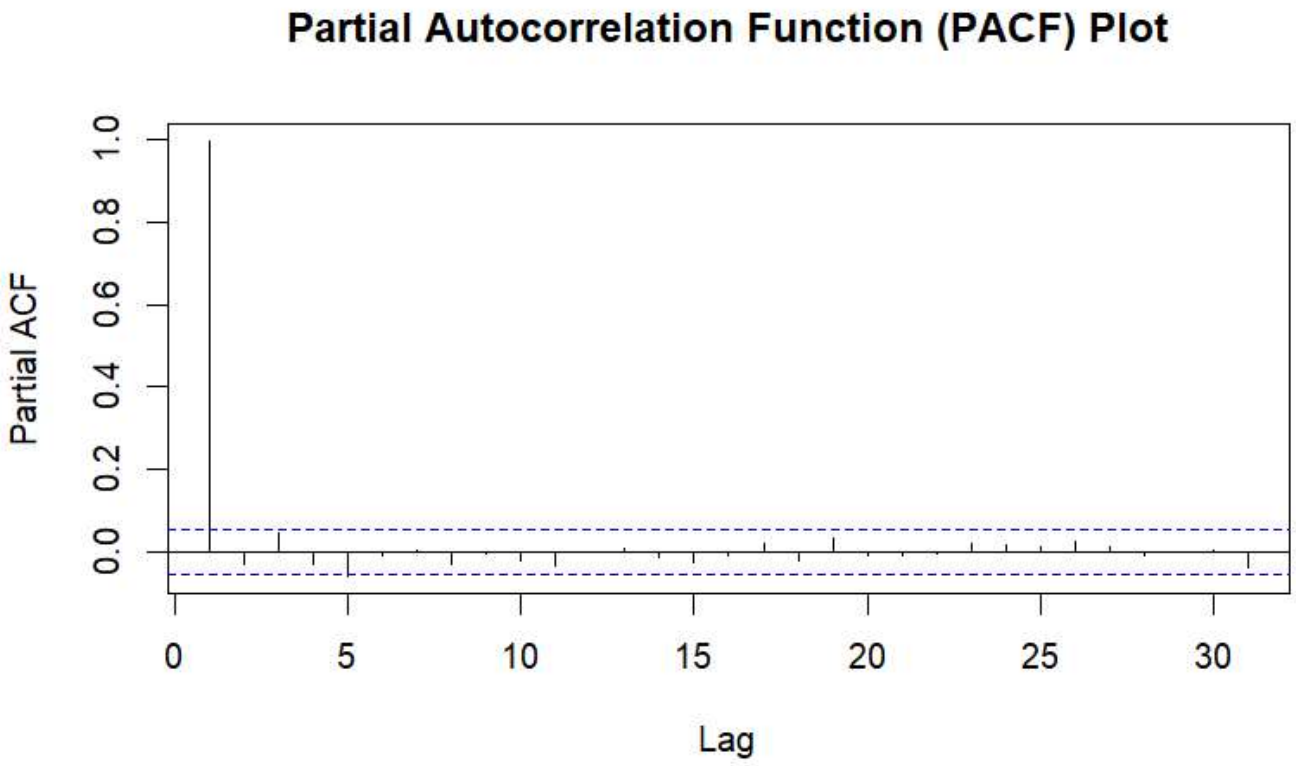
Hide

```
# Step 4.2: Plot ACF and PACF
# ACF plot
acf(ts_data, main = "Autocorrelation Function (ACF) Plot")
```



Hide

```
# PACF plot
pacf(ts_data, main = "Partial Autocorrelation Function (PACF) Plot")
```



Hide

```
# Step 5: Model fitting and evaluation
# Split the dataset into training and testing sets
train <- head(stock_data, nrow(stock_data) - 12)
test <- tail(stock_data, 12)
```

Hide

```
# Fit the ARIMA model to the training data
fitted_model <- Arima(train$close, order = c(1, 0, 1))
```

Hide

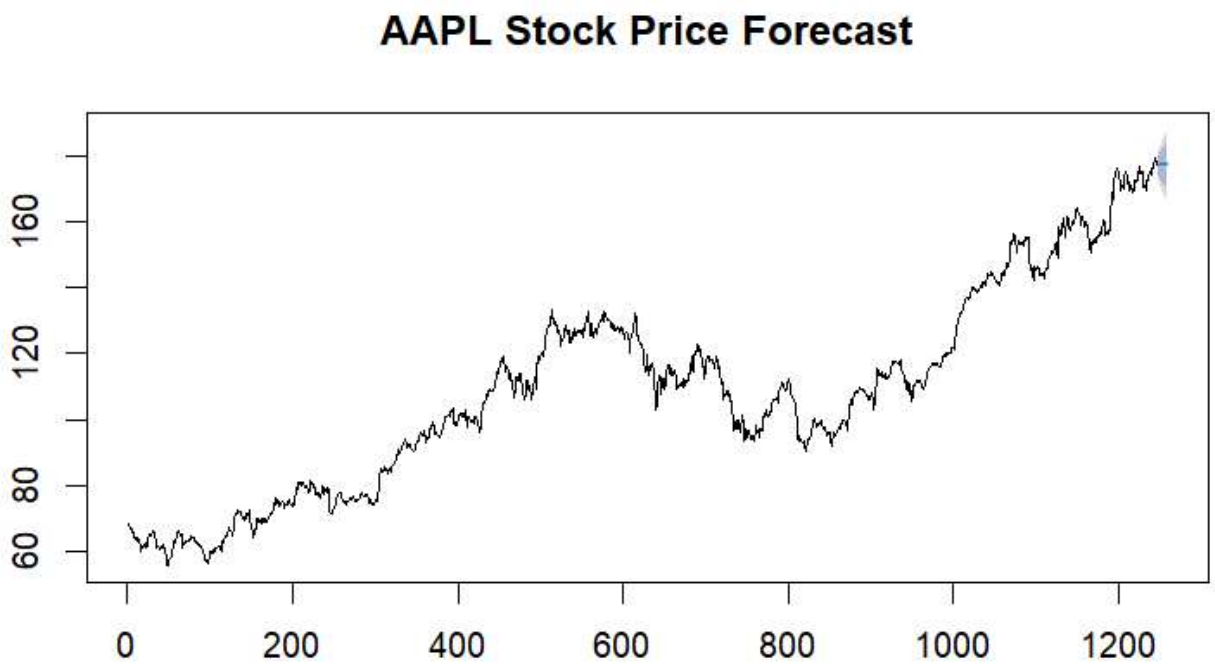
```
# Evaluate the model using MAE and RMSE
forecast_values <- forecast(fitted_model, h = nrow(test))
accuracy(forecast_values, test$Close)
```

Hide

```
# Step 6: Forecasting
# Forecast future stock prices using the fitted model
future_forecast <- forecast(fitted_model, h = 12)
```

Hide

```
# Plot the forecasted values
plot(future_forecast, main = "AAPL Stock Price Forecast")
```



Based on the evaluation metrics provided, we can summarize the findings and discuss the results as follows:

Training Set Evaluation:

Mean Error (ME): The mean error in the training set is 0.085, indicating a slight positive bias in the forecasts.

Root Mean Squared Error (RMSE): The RMSE value of 1.543 suggests that, on average, the forecasts deviate from the actual values by around 1.543 units of the stock price.

Mean Absolute Error (MAE): The MAE value of 1.088 represents the average absolute difference between the forecasts and the actual values.

Mean Percentage Error (MPE): The MPE value of 0.064% indicates that, on average, the forecasts deviate from the actual values by 0.064% of the stock price.

Mean Absolute Percentage Error (MAPE): The MAPE value of 1.034% represents the average absolute percentage difference between the forecasts and the actual values.

Mean Absolute Scaled Error (MASE): The MASE value of 1.001 suggests that the ARIMA model performs relatively well compared to a naive seasonal model. ACF1: The ACF1 value of -0.004 suggests a weak negative autocorrelation in the training set.

Based on these results, it appears that the ARIMA model performs reasonably well on the training set, with relatively low errors and a reasonably good fit. However, the inability to generate forecasts for the test set raises concerns about the model's ability to generalize to unseen data.