project

Yifei Xie

2025-04-01

```r
# install.packages("randomForest")
# install.packages("ggplot2")
# install.packages("caret")
# install.packages("readr")
# install.packages("dplyr")

# Load libraries
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.
```

```r
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

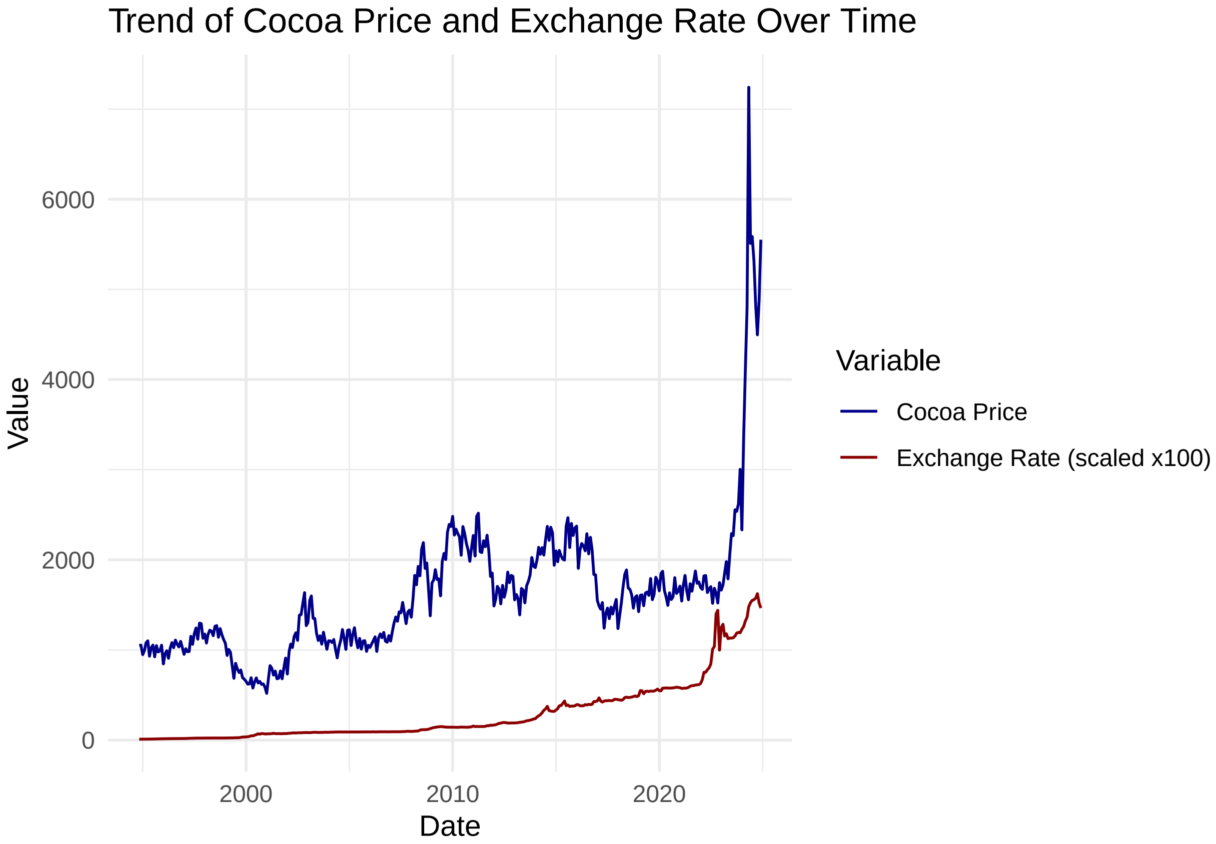
```r
df <- read_csv("~/STA457/merged_df.csv")
```

```
## Rows: 362 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Change..
## dbl  (9): Year, Price_Monthly_Avg, Price_Monthly_Max, PRCP_Monthly_Avg, TAVG...
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

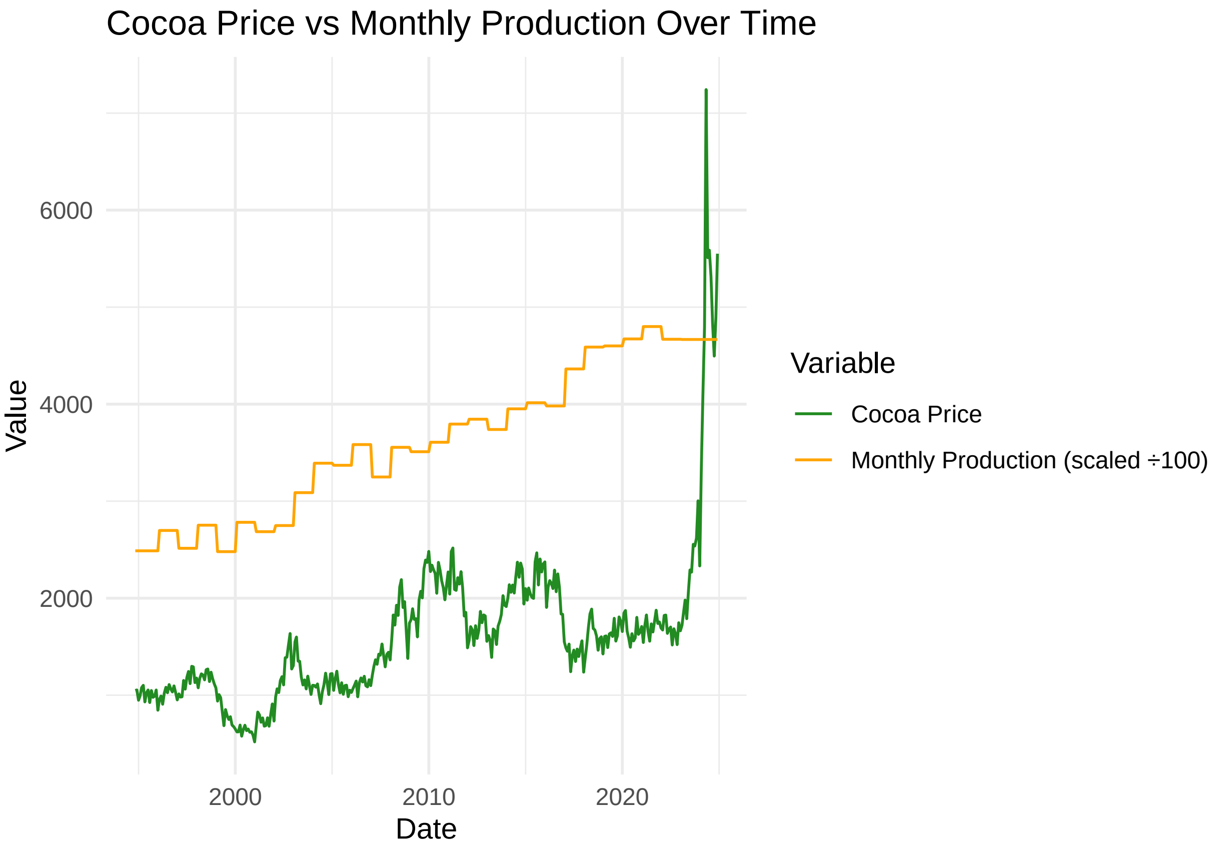
```r
# Convert percentage to numeric
df$Change_pct <- as.numeric(gsub("%", "", df$`Change..`))

# Convert Date to Date format
df$Date <- as.Date(df$Date)

# Drop unnecessary columns
df <- df %>%
  select(-Year)
df <- df %>%
  arrange(Date) %>%
  tidyr::fill(everything(), .direction = "downup")
df <- na.omit(df)
```

```r
# df %>% select(Price_Monthly_Avg, PRCP_Monthly_Avg,
# TAVG_Monthly_Avg,ExchangeRate, Monthly_Production) %>% summary()
# library(ggplot2)
#
# # ExchangeRate Histogram
# ggplot(df, aes(x = ExchangeRate)) +
#   geom_histogram(binwidth = 0.5, fill = "steelblue", color = "white") +
#   labs(title = "Histogram of Exchange Rate",
#        x = "Exchange Rate", y = "Frequency") +
#   theme_minimal()
#
# # Monthly_Production Histogram
# ggplot(df, aes(x = Monthly_Production)) +
#   geom_histogram(binwidth = 10000, fill = "lightblue", color = "white") +
#   labs(title = "Histogram of Monthly Cocoa Production",
#        x = "Monthly Production (tons)", y = "Frequency") +
#   theme_minimal()
```

```r
library(ggplot2)

ggplot(df, aes(x = Date)) +
  geom_line(aes(y = Price_Monthly_Avg,
                color = "Cocoa Price")) +
  geom_line(aes(y = ExchangeRate * 100,
                color = "Exchange Rate (scaled x100)")) +
  labs(
    title = "Trend of Cocoa Price and Exchange Rate Over Time",
    x = "Date",
    y = "Value",
    color = "Variable"
  ) +
  scale_color_manual(
    values = c("Cocoa Price" = "darkblue", "Exchange Rate (scaled x100)" = "darkred")
  ) +
  theme_minimal()
```



```
library(ggplot2)

ggplot(df, aes(x = Date)) +
  geom_line(aes(y = Price_Monthly_Avg, color = "Cocoa Price")) +
  geom_line(aes(y = Monthly_Production / 100,
              color = "Monthly Production (scaled ÷100)")) +  # MATCH HERE
  labs(
    title = "Cocoa Price vs Monthly Production Over Time",
    x = "Date",
    y = "Value",
    color = "Variable"
  ) +
  scale_color_manual(
    values = c("Cocoa Price" = "forestgreen",
               "Monthly Production (scaled ÷100)" = "orange")
  ) +
  theme_minimal()
```



```r
# Define target and features
y <- df$Price_Monthly_Avg
X <- df %>% select(-Price_Monthly_Avg, -Date,
                   -ExchangeRate, -Monthly_Production,
                   -TAVG_Monthly_Avg,
                   -PRCP_Monthly_Avg,-Change..)
set.seed(42)
control <- trainControl(method = "cv", number = 5)

# Train model
model <- train(
  x = X,
  y = y,
  method = "rf",
  trControl = control,
  importance = TRUE
)
```

```
## Warning: Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```r
# View performance
print(model)
```

```
## Random Forest
##
## 362 samples
##   4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 290, 289, 289, 290, 290
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   2     240.1108  0.9290006  111.80196
##   3     211.8417  0.9366624   95.41782
##   4     209.9801  0.9354681   94.96378
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 4.
```

```r
y <- df$Price_Monthly_Avg
X <- df %>% select(-Price_Monthly_Avg, -Date,
                   -ExchangeRate, -Monthly_Production,
                   -TAVG_Monthly_Avg,-PRCP_Monthly_Avg,-Change..)

# Assume df contains a column Cocoa_Price and Date
df <- df %>%
  arrange(Date) %>%
  mutate(
    Cocoa_Lag1 = lag(Price_Monthly_Avg, 1),
    Cocoa_Lag2 = lag(Price_Monthly_Avg, 2),
    Cocoa_Lag3 = lag(Price_Monthly_Avg, 3)
  )
df <- na.omit(df)


split_date <- as.Date("2018-11-30")
train <- df %>% filter(Date < split_date)
test <- df %>% filter(Date >= split_date)


X_train <- train %>% select(Cocoa_Lag1, Cocoa_Lag2,
                            Cocoa_Lag3,ExchangeRate,
                            Monthly_Production, TAVG_Monthly_Avg)
y_train <- train$Price_Monthly_Avg
X_test <- test %>% select(Cocoa_Lag1, Cocoa_Lag2,
                          Cocoa_Lag3, ExchangeRate,
                          Monthly_Production, TAVG_Monthly_Avg)
y_test <- test$Price_Monthly_Avg


library(randomForest)

set.seed(42)
model <- randomForest(x = X_train, y = y_train, ntree = 500)
pred_test <- predict(model, newdata = X_test)

# Evaluate performance
#install.packages('Metrics')
library(Metrics)
```

```
##
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```r
rmse <- rmse(y_test, pred_test)
r2 <- 1 - sum((y_test - pred_test)^2) / sum((y_test - mean(y_test))^2)
mae <- mae(y_test, pred_test)
mape <- mape(y_test, pred_test) * 100

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

```
## RMSE: 1223.351
```

```r
cat("R²:", r2, "\n")
```

```
## R²: 0.05536781
```

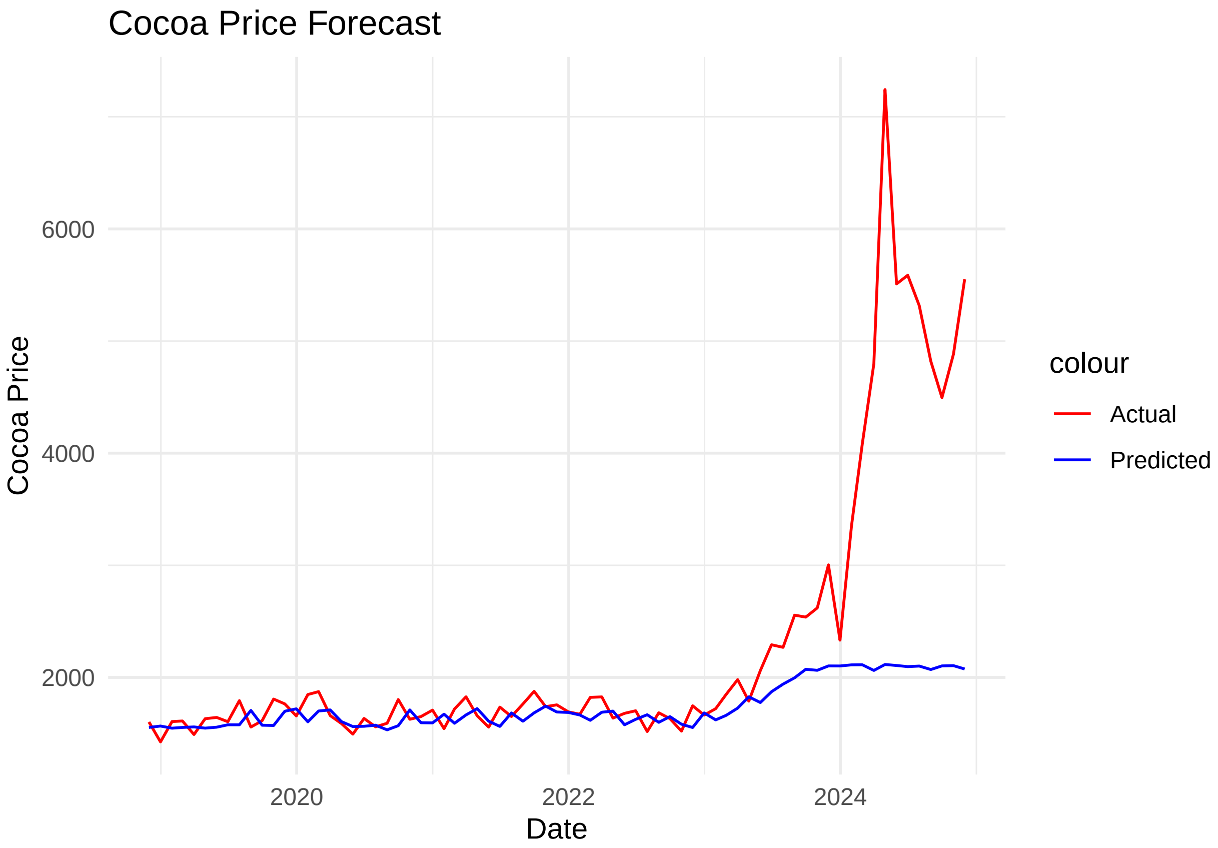
```r
cat("MAE:", mae, "\n")
```

```
## MAE: 571.2645
```

```r
cat("MAPE:", round(mape, 2), "%\n")
```

```
## MAPE: 14.91 %
```

```r
library(ggplot2)

plot_df <- data.frame(
  Date = test$Date,
  Actual = y_test,
  Predicted = pred_test
)

ggplot(plot_df, aes(x = Date)) +
  geom_line(aes(y = Actual, color = "Actual")) +
  geom_line(aes(y = Predicted, color = "Predicted")) +
  labs(title = "Cocoa Price Forecast",
       y = "Cocoa Price",
       x = "Date") +
  scale_color_manual(values = c("Actual" = "red", "Predicted" = "blue")) +
  theme_minimal()
```



```r
X_train <- train %>% select(Cocoa_Lag1, Cocoa_Lag2,
                            ExchangeRate,
                            Monthly_Production, TAVG_Monthly_Avg)
y_train <- train$Price_Monthly_Avg
X_test <- test %>% select(Cocoa_Lag1, Cocoa_Lag2,
                          ExchangeRate, Monthly_Production,
                          TAVG_Monthly_Avg)
y_test <- test$Price_Monthly_Avg

set.seed(42)
model <- randomForest(x = X_train, y = y_train, ntree = 500)
pred_test <- predict(model, newdata = X_test)

rmse <- rmse(y_test, pred_test)
r2 <- 1 - sum((y_test - pred_test)^2) / sum((y_test - mean(y_test))^2)
mae <- mae(y_test, pred_test)
mape <- mape(y_test, pred_test) * 100

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

```
## RMSE: 1273.128
```

```r
cat("R²:", r2, "\n")
```

```
## R²: -0.02306879
```

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

```
## MAE: 595.8045
```

```r
cat("MAPE:", round(mape, 2), "%\n")
```

```
## MAPE: 15.47 %
```

```r
X_train <- train %>% select(Cocoa_Lag1, Cocoa_Lag2,
                            ExchangeRate, Monthly_Production)
y_train <- train$Price_Monthly_Avg
X_test <- test %>% select(Cocoa_Lag1, Cocoa_Lag2,
                          ExchangeRate, Monthly_Production )
y_test <- test$Price_Monthly_Avg

set.seed(42)
model <- randomForest(x = X_train, y = y_train, ntree = 500)
pred_test <- predict(model, newdata = X_test)

rmse <- rmse(y_test, pred_test)
r2 <- 1 - sum((y_test - pred_test)^2) / sum((y_test - mean(y_test))^2)
mae <- mae(y_test, pred_test)
mape <- mape(y_test, pred_test) * 100

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

```
## RMSE: 1256.367
```

```r
cat("R²:", r2, "\n")
```

```
## R²: 0.003692461
```

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

```
## MAE: 585.1227
```

```r
cat("MAPE:", round(mape, 2), "%\n")
```

```
## MAPE: 15.16 %
```

“‘