Predictive Models for Interest Rate

Abstract

This study investigates the relationship between the Federal Reserve's interest rate and inflation, primarily focusing on the Consumer Price Index (CPI). We analyzed historical data from 1990 to assess the correlation between the FED's interest rate and CPI, finding that the relationship is weak at the monthly level but more pronounced annually. A Granger causality test confirmed that the FED's interest rate causes changes in CPI, suggesting a significant impact of monetary policy on inflation. Further analysis of the lags in the FED's interest rate revealed a negative correlation, indicating that interest rate increases tend to coincide with a gradual decrease in inflation. Additionally, models such as regression and Random Forest were employed to predict core CPI, showing that while monetary policy has a delayed effect, it significantly influences inflation. Incorporating exchange rate data, we observed that the FED's rate has an immediate effect on foreign currencies like the Yuan and the Yen, contrasting with the delayed impact on inflation. This study concludes that while the FED's interest rate affects inflation with long-term, gradual effects, its influence on exchange rates is immediate and direct.

Motivation

The motivation for this project arises from the imminent change in the U.S. government, where discussions have begun regarding the implementation of new import tariffs and how these could affect domestic prices. Given the various opinions about this impact, the focus has shifted to the role of the Federal Reserve (FED) and its influence not only on the U.S. economy but also on other countries' economies. Building on this debate, the project aims to understand the relationship between the FED's interest rate and inflation through the analysis of historical data and the evaluation of predictive model performance.

Library Imports

The project begins by importing the necessary libraries for financial analysis, time series manipulation, data transformation, modeling, and visualization.

```
# Financial analysis
library(quantmod)

# Time series analysis
library(tseries)

# Data transformation
library(dplyr)
library(tidyr)
library(reshape2)
library(kableExtra)
library(purrr)

# Data modeling
library(lmtest)
library(randomForest)
library(caret)
```

```
# Visualization
library(ggplot2)
```

These libraries help manage financial data, manipulate time series data, build and evaluate models, and create visualizations.

---

1 **Data Loading**

In this step, we fetch historical data for the Federal Reserve's interest rate and the Consumer Price Index (CPI) for inflation. The data is obtained from FRED, starting from January 1990. Two separate data frames are created for each dataset.

```
first_date <- "1990-01-01"

# Get data for the Federal Reserve interest rate and inflation (CPI)
getSymbols("FEDFUNDS", src = "FRED", from = first_date, to = Sys.Date())
getSymbols("CPIAUCNS", src = "FRED", from = first_date, to = Sys.Date())

# Convert to data frame and rename columns
fed_df <- data.frame(
  Date = index(FEDFUNDS),
  FED = as.numeric(FEDFUNDS[, 1])
)

ipc_df <- data.frame(
  Date = index(CPIAUCNS),
  IPC = as.numeric(CPIAUCNS[, 1])
)
```

1.1 Monthly Inflation Calculation

The inflation data from FRED is cumulative, but we are interested in the monthly change in inflation. To calculate this, we create a new column that represents the difference between consecutive values.

```
# Calculate monthly change in CPI
ipc_df$IPC_Monthly <- c(NA, diff(ipc_df$IPC))
```

This new column, `IPC_Monthly`, will allow us to analyze inflation on a month-to-month basis.

1.2 Visualization of Data

We then visualize the data to observe the trends over time. The first graph shows the evolution of the Federal Reserve's interest rate, and the second graph shows the monthly inflation index.

```
# Plot Federal Reserve Interest Rate
ggplot(fed_df, aes(x = Date, y = FED)) +
  geom_line() +
  ggtitle("Federal Reserve Interest Rate") +
  xlab("Date") +
  ylab("Rate (%)") +
  theme_minimal()

# Plot Inflation (CPI)
ggplot(ipc_df, aes(x = Date, y = IPC_Monthly)) +
  geom_line() +
  ggtitle("Consumer Price Index (Inflation)") +
  xlab("Date") +
  ylab("Index") +
  theme_minimal()
```

These graphs help visualize the fluctuations in the interest rate and inflation over time.

---

2 **Relationship Between FED & CPI**

2.1 Correlation

The first step in analyzing the relationship between the Federal Reserve's interest rate (FED) and the Consumer Price Index (CPI) is to check for correlation between the monthly interest rate and inflation.

```
# Correlation between the FED interest rate and monthly inflation
merged_df <- merge(fed_df, ipc_df, by = "Date")
correlacion <- cor(merged_df$FED, merged_df$IPC_Monthly, use = "complete.obs")
print(correlacion)
```

If the correlation value is close to zero, it indicates that there is no strong correlation between the two variables. Since the relationship may not be evident month-to-month, we also examine the data on an annual basis.

Annual Correlation

We calculate the annual averages for both the FED interest rate and CPI to explore the correlation over the long term.

```
# Annual average of the FED rate
fed_df$Year <- format(fed_df$Date, "%Y")
fed_annual <- aggregate(FED ~ Year, data = fed_df, FUN = mean)

# Annual average of CPI
ipc_df$Year <- format(ipc_df$Date, "%Y")
ipc_annual <- aggregate(IPC ~ Year, data = ipc_df, FUN = mean)
ipc_annual$IPC_Annualy <- c(NA, diff(ipc_annual$IPC))

merged_annual <- merge(fed_annual, ipc_annual, by = "Year")
merged_annual <- na.omit(merged_annual)
correlacio_annual <- cor(merged_annual$FED, merged_annual$IPC_Annualy, use =
"complete.obs")
print(correlacio_annual)
```

The correlation value is still close to zero, suggesting no significant correlation between the FED rate and inflation on an annual basis. We visualize these results in a graph showing the evolution of both variables over time.

```
ggplot(merged_df, aes(x = Date)) +
  geom_line(aes(y = FED, color = "FED")) +
  geom_line(aes(y = IPC_Monthly * 2, color = "IPC")) +
  scale_y_continuous(
    name = "FED Rate (%)",
    sec.axis = sec_axis(~ . / 2, name = "Monthly Inflation (CPI)")
  ) +
  labs(
    title = "FED Rate and Monthly Inflation Evolution",
    x = "Date",
    color = "Variable"
  ) +
  theme_minimal()
```

The graph above shows the evolution of the FED interest rate and monthly inflation (CPI), which provides a visual representation of their movements over time.

2.2 Granger Causality Test

The Granger Causality Test evaluates whether one time series, X, can help predict another time series, Y. The test is based on the idea that if knowing past values of X improves the prediction of Y, then X is said to "cause" Y in the Granger sense.

In this case, we test two hypotheses:

1. FED → CPI (Does the FED interest rate affect inflation?)
2. CPI → FED (Does inflation affect the FED interest rate?)

We create two time series, one for the FED and one for CPI, and use them to analyze the impact of previous months on the current variable.

```
# Time series
merged_df <- na.omit(merged_df)
fed_ts <- ts(merged_df$FED, start = c(2000, 1), frequency = 12)
ipc_ts <- ts(merged_df$IPC_Monthly, start = c(2000, 1), frequency = 12)

# Lags to use: 12 months
granger_test <- grangertest(ipc_ts ~ fed_ts, order = 12) # FED → CPI
print(granger_test)
```

FED → CPI Test

The test compares two regression models:

- Model 1:
  [
  IPC_t = \alpha + \sum_{i=1}^{12} \beta_i IPC_{t-i} + \sum_{i=1}^{12} \gamma_i FED_{t-i} + \varepsilon_t
  ]
  This model uses the past values of both IPC and FED to predict the current CPI.
- Model 2:
  [
  IPC_t = \alpha + \sum_{i=1}^{12} \beta_i IPC_{t-i} + \varepsilon_t
  ]
  This model uses only the past values of IPC, excluding the FED.

Interpretation of Results

Column	Meaning
Res.Df	Residual degrees of freedom after model adjustment
Df	Difference in degrees of freedom between the models
F	F-statistic of the Granger test
Pr(>F)	p-value to assess significance

1. Res.Df (Residual Degrees of Freedom)

Represents the degrees of freedom of the model after adjusting the parameters. It is the result of the difference between the number of observations and the number of parameters estimated.

2. Df (Degrees of Freedom)

Shows the difference in degrees of freedom between the two models:

[
\text{Df} = \text{Res.Df Model 2} - \text{Res.Df Model 1}
]

The Df value is -12 because the full model has 12 additional parameters due to the lags of the second variable.

3. F (F-Statistic)

The F-statistic tests whether the complete model (with both time series) explains significantly more variability than the reduced model. It is calculated as:

[
F = \frac{(\text{Sum of Squares Explained by FED}) / 12}{(\text{Sum of Squares of Residuals}) / \text{Res.Df Model 1}}
]

A large F value means adding the FED improves the prediction. A small F value indicates that the additional variable does not provide relevant information.

4. Pr(>F) (p-value)

This is the probability of obtaining an F-value this high if the series had no effect.

- If p < 0.05, we reject the null hypothesis and conclude that there is Granger causality.
- If p > 0.05, there is insufficient evidence to claim that one variable causes the other.

Results

Since the p-value obtained is less than 0.05, we reject the null hypothesis and conclude that FED causes CPI in the Granger sense (with 95% confidence). This indicates that the FED interest rate has a significant impact on inflation (CPI).

2.3 Lags and Future Values

Based on the results from the Granger Causality Test, we aim to explore whether the lags of the FED interest rate affect the current CPI. To do this, we assign previous values (lags) of the FED to each corresponding date and then calculate the correlation between the lags of the FED and inflation.

```
# Create the lags for the FED

months <- 36

for (i in 1:months) {
  merged_df <- merged_df %>%
    mutate(!!paste("FED_lag", i, sep = "_") := lag(FED, i))
}

# Calculate the correlation between IPC_Monthly and the lags
correlations_1 <- sapply(1:months, function(i) cor(merged_df$IPC_Monthly,
merged_df[[paste("FED_lag", i, sep = "_")]], use = "complete.obs"))

# Create a dataframe for the table
correlation_table_1 <- data.frame(
  Lag = paste("FED_lag", 1:months, sep = "_"),
  Correlation = correlations_1
)

correlation_table_1

# Correlation plot

# X-axis labels
correlation_table_1$Lag <- as.numeric(gsub("FED_lag_", "", correlation_table_1$Lag))
correlation_table_1$Lag <- factor(correlation_table_1$Lag, levels =
sort(unique(correlation_table_1$Lag)))

ggplot(correlation_table_1, aes(x = Lag, y = Correlation, fill = Correlation)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient2(low = "#1ac5ca", high = "#f87970", mid = "#ffffff", midpoint =
0, name = "Correlation", limits = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Correlation between CPI and FED interest rate lags",
       x = "FED Lags",
       y = "Correlation") +
  scale_y_reverse()
```

2.4 Interpretation of the Correlations

Upon observing the correlation values, we note the following:

1. Negative Correlation: All lags show a negative correlation, suggesting that an increase in the FED interest rate tends to be associated with a decrease in the CPI, although the magnitude of this relationship is small.

2. Weak Correlation: The correlation values are small, indicating that the relationship is not strong.

3. Stability across Lags: The correlation remains relatively stable across all lags, suggesting that the effects of FED's monetary policy on inflation may not be immediate and do not diminish abruptly over time.

4. Slightly Decreasing Trend: The correlation appears to become slightly more negative as the lags increase, reaching its lowest value beyond the first 12 months. This behavior might suggest that the impact of FED's decisions on inflation has a gradual effect that persists over several months.

Although the results show that there is a relationship between the lags of the FED interest rate and inflation, this relationship is weak, and it does not seem to be immediate or strong.

---

3 **Other Inflation Indicators**

Given that no clear and relevant data was observed, it may be helpful to consider additional factors for a more accurate understanding of the impact of monetary policy on inflation. For this, we incorporated other variants of the CPI, such as the Core CPI, which excludes seasonal and temporary price changes, and inflation expectations according to the University of Michigan. We examine the correlation for each of these variables.

```
# Core CPI
getSymbols("CPILFESL", src = "FRED", from = first_date, to = Sys.Date())
core_ipc_df <- data.frame(
  Date = index(CPILFESL),
  IPC_core = as.numeric(CPILFESL[, 1])
)

core_ipc_df$IPC_core_Monthly <- c(NA, diff(core_ipc_df$IPC_core))
core_ipc_df$IPC_core_Monthly <- as.numeric(core_ipc_df$IPC_core_Monthly)

# Inflation Expectations from the University of Michigan
getSymbols("MICH", src = "FRED", from = first_date, to = Sys.Date())
expectations_df <- data.frame(
  Date = index(MICH),
  IPC_exp = as.numeric(MICH[, 1])
)

# Incorporating the new variables
merged_df <- merge(merged_df, core_ipc_df, by = "Date", all.x = TRUE)
merged_df <- merge(merged_df, expectations_df, by = "Date", all.x = TRUE)
merged_df <- na.omit(merged_df)

# Correlation

# Core IPC
correlations_core_1 <- sapply(1:months, function(i) cor(merged_df$IPC_core_Monthly,
merged_df[[paste("FED_lag", i, sep = "_")]], use = "complete.obs"))

# Inflation Expectations
correlations_exp_1 <- sapply(1:months, function(i) cor(merged_df$IPC_exp,
merged_df[[paste("FED_lag", i, sep = "_")]], use = "complete.obs"))

# Correlation Table
```

```
correlation_table_2 <- data.frame(
  Lag = 1:months,
  IPC_Monthly = correlations_1,
  IPC_core = correlations_core_1,
  IPC_exp = correlations_exp_1
)

correlation_table_2

# Table with the three correlations
correlation_graph_2 <- data.frame(
  Lag = rep(1:months, 3),
  IPC = c(correlations_1, correlations_core_1, correlations_exp_1),
  Type = rep(c("IPC", "Core IPC", "Inflation Expectations"), each = months)
)

# Correlation plot

# X-axis labels
correlation_graph_2$Lag <- as.factor(correlation_table_2$Lag)
correlation_graph_2$Lag <- as.numeric(as.character(correlation_graph_2$Lag)) # Lag to
numeric

ggplot(correlation_graph_2, aes(x = Lag, y = IPC, fill = IPC)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_gradient2(low = "#1ac5ca", high = "#f87970", mid = "#ffffff", midpoint =
0, name = "Correlation", limits = c(-1, 1)) +
  scale_y_reverse(limits = c(0, min(correlation_graph_2$IPC))) +  # Invert the Y-
axis, starting at 0
  scale_x_continuous(breaks = seq(0, max(correlation_graph_2$Lag, na.rm = TRUE), by =
6)) +  # Multiples of 6
  theme_minimal() +
  labs(title = "Correlation between CPI and FED interest rate lags",
       x = "FED Lags",
       y = "Correlation") +
  facet_wrap(~Type, scales = "fixed")  # Keep the same axis across all plots
```

The relationship between the lags and the various CPI variants is consistently negative. While inflation expectations show somewhat higher correlation values, this becomes more evident in the Core CPI. Let's now explore how a regression model might predict this variable.

```
# Regression
fed_lags <- paste("FED_lag", 1:months, sep = "_", collapse = " + ")
model <- lm(paste("IPC_core_Monthly ~", fed_lags), data = merged_df)
summary(model)
```

The model explains about 50% of the variability in the Core CPI, with a moderate amount of residual error, indicating that a significant portion of this variability is not explained by the FED lags. We expect better results when performing a more time-limited analysis.

---

4 **Recent Years**

We filter the data to include values from 2020 to the present. Then, we revisit the correlation between the variables.

```
# Filter the data by date
start_date <- as.Date("2000-01-01")
filtered_df <- merged_df[merged_df$Date >= start_date, ]

# Calculate correlations for the three types of CPI (CPI, Core CPI, Inflation
Expectations)
correlations_2 <- sapply(1:months, function(lag) cor(filtered_df$IPC,
filtered_df[[paste0("FED_lag_", lag)]]))
correlations_core_2 <- sapply(1:months, function(lag) cor(filtered_df$IPC_core,
filtered_df[[paste0("FED_lag_", lag)]]))
correlations_exp_2 <- sapply(1:months, function(lag) cor(filtered_df$IPC_exp,
filtered_df[[paste0("FED_lag_", lag)]]))

# Correlation table
correlation_table_3 <- data.frame(
  Lag = 1:months,
  IPC_Monthly = correlations_2,
  IPC_core = correlations_core_2,
  IPC_exp = correlations_exp_2
)

correlation_table_3

# Table with the three correlations
correlation_graph_3 <- data.frame(
  Lag = rep(1:months, 3),
  IPC = c(correlations_2, correlations_core_2, correlations_exp_2),
  Type = rep(c("CPI", "Core CPI", "Inflation Expectations"), each = months)
)

# X-axis labels
correlation_graph_3$Lag <- as.factor(correlation_table_3$Lag)
correlation_graph_3$Lag <- as.numeric(as.character(correlation_graph_3$Lag)) # Lag to
numeric

ggplot(correlation_graph_3, aes(x = Lag, y = IPC, fill = IPC)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_gradient2(low = "#1ac5ca", high = "#f87970", mid = "#ffffff", midpoint =
0, name = "Correlation", limits = c(-1, 1)) +
  scale_y_reverse(limits = c(0, min(correlation_graph_3$IPC))) +  # Reverse the Y-
axis, starting at 0
  scale_x_continuous(breaks = seq(0, max(correlation_graph_3$Lag, na.rm = TRUE), by =
6)) +  # Multiples of 6
  theme_minimal() +
  labs(title = "Correlation between CPI and FED interest rate lags",
       x = "FED Lags",
       y = "Correlation") +
  facet_wrap(~Type, scales = "fixed")  # Keep the same axis across all plots
```

Both in CPI and Core CPI, there is an initial misalignment with the interest rate, with a negative correlation that reaches its most significant point beyond month 30. After this point, both variables begin to gradually align with the FED interest rate. In contrast, Inflation Expectations show a much earlier alignment, which could be explained by the speed at which economic agents incorporate changes in the interest rate into their inflation expectations.

---

5 **Model**

To assess the relationship between the FED interest rate and core inflation (Core CPI), we trained a Random Forest model using different lags of the interest rate as predictor variables.

```
# Normalize the data (except 'Date')
filtered_df_norm <- filtered_df %>%
  mutate(across(-Date, ~ as.numeric(scale(.))))

# Define predictor variables
fed_lags <- names(filtered_df_norm)[grepl("FED_lag_", names(filtered_df_norm))]

# Create dynamic formula for the model
formula_rf <- as.formula(paste("IPC_core_Monthly ~", paste(fed_lags, collapse = " +
")))

# Split into 'train' and 'test'
set.seed(54)  # Reproducibility
trainIndex <- createDataPartition(filtered_df_norm$IPC_core_Monthly, p = 0.8, list =
FALSE)
trainData <- filtered_df_norm[trainIndex, ]
testData <- filtered_df_norm[-trainIndex, ]

# Train Random Forest model
model_rf <- randomForest(formula_rf, data = trainData, ntree = 500)

# Predictions
predictions <- predict(model_rf, newdata = testData)

# Metrics: Mean Squared Error & R^2
mse <- mean((predictions - testData$IPC_core_Monthly)^2)
rsq <- 1 - sum((predictions - testData$IPC_core_Monthly)^2) /
sum((testData$IPC_core_Monthly - mean(testData$IPC_core_Monthly))^2)

print(paste("MSE: ", mse))
print(paste("R-squared: ", rsq))

# Variable importance
importance(model_rf)
```

The Random Forest model suggests that approximately half of the variability in core inflation can be explained by the lags of the FED interest rate. This indicates that monetary policy has a significant impact, though it is not the only factor at play.

Regarding the importance of different lags, the most relevant effects occur around the one-year lag, which is consistent with the idea that changes in interest rates take time to affect inflation. However, longer lags also emerge, suggesting that monetary policy can still have an impact even several years after its implementation.

These results reinforce the notion that the relationship between interest rates and inflation is not immediate, and that there are delayed effects to consider when analyzing the effectiveness of economic policy decisions.

---

6 Hyperparameters

We introduced certain hyperparameters to fine-tune the model.

- mtry: The number of variables considered at each split of the tree.
- ntree: The total number of trees in the forest.
- nodesize: The maximum number of observations in each terminal node.

We aimed to determine the best value for mtry to optimize the model's performance through an out-of-bag (OOB) error evaluation.

The out-of-bag (OOB) error is a performance measure in Random Forest, calculated using the data not used in the construction of each tree. These data, which are left out in each iteration due to bootstrap sampling, are then used to evaluate the model, providing an accurate estimate without needing a validation set.

```
set.seed(54)

# Define hyperparameters to test
tuneGrid <- expand.grid(mtry = seq(2, ncol(trainData) - 1, by = 2))

# Store OOB errors for different mtry values
mtry_values <- 1:(ncol(trainData) - 1)
oob_errors <- numeric(length(mtry_values))

for (i in mtry_values) {
  rf_model <- randomForest(IPC_core_Monthly ~ ., data = trainData, mtry = i, ntree =
500)
  oob_errors[i] <- rf_model$mse[length(rf_model$mse)]  # Last OOB error value
}

# Plot OOB error vs mtry
df <- data.frame(mtry = mtry_values, oob_error = oob_errors)
ggplot(df, aes(x = mtry, y = oob_error)) +
  geom_point(size = 2) +
  geom_line() +
  ggtitle("OOB Error vs. mtry") +
  xlab("Number of variables selected at each split (mtry)") +
  ylab("OOB Error") +
  theme_minimal()
```

6.1 Cross-validation

We performed cross-validation to find the best mtry value and train a more robust Random Forest model. The model's performance was evaluated using Mean Squared Error (MSE) and R-squared.

```
# Find the best mtry
rf_tuned <- train(IPC_core_Monthly ~ ., data = trainData,
                  method = "rf",
                  tuneGrid = tuneGrid,
                  trControl = trainControl(method = "cv", number = 5),  # 5-fold CV
                  ntree = 500)

# Train model with best mtry
model_rf <- randomForest(IPC_core_Monthly ~ .,
                         data = trainData,
                         mtry = rf_tuned$bestTune$mtry,  # Best mtry
                         ntree = 500,
                         nodesize = 5)  # You can adjust `nodesize`

# Make predictions
predictions <- predict(model_rf, newdata = testData)

# Calculate MSE and R-squared using IPC_core_Monthly as the actual variable
mse <- mean((predictions - testData$IPC_core_Monthly)^2)
rsq <- 1 - sum((predictions - testData$IPC_core_Monthly)^2) /
sum((testData$IPC_core_Monthly - mean(testData$IPC_core_Monthly))^2)

# Print results
print(paste("MSE: ", mse))
print(paste("R-squared: ", rsq))
```

6.2 Other Hyperparameters

- ntree (Number of trees): Controls the number of trees in the forest. A higher number of trees usually results in a more robust and accurate model, but it comes at the cost of longer computation time.
- nodesize (Node size): Sets the minimum number of observations required in a terminal node. Lower values allow greater flexibility in the split, leading to more complex models that may be prone to overfitting. Higher values make the trees more general and less prone to overfitting, but the models may be too simple.

```
# Define values for ntree and nodesize to test
ntree_values <- c(100, 200, 500, 1000)
nodesize_values <- c(1, 5, 10, 20)

# Create an empty data frame to store results
results <- data.frame(mtry = integer(), ntree = integer(), nodesize = integer(), MSE
= numeric(), Rsq = numeric())

# Loop over ntree and nodesize values
for (ntree_value in ntree_values) {
  for (nodesize_value in nodesize_values) {

    # Define tuneGrid with only mtry
    tuneGrid <- expand.grid(mtry = seq(2, ncol(trainData) - 1, by = 2))

    # Train model with current ntree and nodesize values
    rf_tuned <- train(IPC_core_Monthly ~ ., data = trainData,
                      method = "rf",
                      tuneGrid = tuneGrid,
                      trControl = trainControl(method = "cv", number = 5),  # 5-fold
CV
                      ntree = ntree_value,
                      nodesize = nodesize_value)

    # Get the best mtry
    best_mtry <- rf_tuned$bestTune$mtry

    # Train model with the best values for the three hyperparameters
    model_rf <- randomForest(IPC_core_Monthly ~ .,
                             data = trainData,
                             mtry = best_mtry,
                             ntree = ntree_value,
                             nodesize = nodesize_value)

    # Make predictions
    predictions <- predict(model_rf, newdata = testData)

    # Calculate MSE and R-squared
    mse <- mean((predictions - testData$IPC_core_Monthly)^2)
    rsq <- 1 - sum((predictions - testData$IPC_core_Monthly)^2) /
sum((testData$IPC_core_Monthly - mean(testData$IPC_core_Monthly))^2)

    # Store the results
    results <- rbind(results, data.frame(mtry = best_mtry, ntree = ntree_value,
nodesize = nodesize_value, MSE = mse, Rsq = rsq))
  }
}

# Print the results
print(results)
```

No significant differences were observed in the performance when varying the mtry, ntree, and nodesize hyperparameters, indicating that a moderate adjustment is sufficient to achieve good results. The variations in MSE and R-squared are minimal, suggesting that no drastic changes to the hyperparameters are necessary. A more efficient configuration can be chosen without compromising performance, prioritizing overall results over extreme parameter optimization.

---

7 **Global Impact**

We incorporate exchange rate data for the following currencies to understand the global effect of the FED's interest rate:

1. Euro, U.S. Dollars to One Euro
2. Yuan, Chinese Yuan Renminbi to One U.S. Dollar
3. Yen, Japanese Yen to One U.S. Dollar
4. Real, Brazilian Reals to One U.S. Dollar

The following R code is used to retrieve the data:

```
# Obtain exchange rate data
start_currency_date <- as.Date("2000-01-01")
```

```
# Euro to Dollar Exchange Rate (Dollar to Euro)
getSymbols("DEXUSEU", src = "FRED", from = start_currency_date, to = Sys.Date())
dolar_euro_df <- data.frame(
  Date = index(DEXUSEU),
  EUR_USD = 1 / as.numeric(DEXUSEU[, 1])  # Invert exchange rate
)

# Dollar to Chinese Yuan (Yuan to Dollar)
getSymbols("DEXCHUS", src = "FRED", from = start_currency_date, to = Sys.Date())
yuan_df <- data.frame(
  Date = index(DEXCHUS),
  USD_CNY = as.numeric(DEXCHUS[, 1])
)

# Dollar to Japanese Yen (Yen to Dollar)
getSymbols("DEXJPUS", src = "FRED", from = start_currency_date, to = Sys.Date())
yen_df <- data.frame(
  Date = index(DEXJPUS),
  USD_JPY = as.numeric(DEXJPUS[, 1])
)

# Dollar to Brazilian Real (Real to Dollar)
getSymbols("DEXBZUS", src = "FRED", from = start_currency_date, to = Sys.Date())
real_df <- data.frame(
  Date = index(DEXBZUS),
  USD_BRL = as.numeric(DEXBZUS[, 1])
)
```

Next, we create a function `get_monthly_avg` that calculates the monthly average of these exchange rates, which allows comparison with the monthly interest rate changes from the FED.

```
get_monthly_avg <- function(df, currency_name) {
  df$Date <- as.Date(df$Date)
  df$YearMonth <- format(df$Date, "%Y-%m")

  df_monthly_avg <- df %>%
    group_by(YearMonth) %>%
    summarise(across(where(is.numeric), mean, na.rm = TRUE))

  df_monthly_avg <- df_monthly_avg %>%
    rename(!!currency_name := !!sym(currency_name), Date = YearMonth)

  return(df_monthly_avg)
}
```

The function is applied to generate monthly averages for the exchange rates and then merge them with the FED interest rate data. A heatmap is then created to visualize the correlation between currency behavior and FED rate changes.

```
# Monthly averages
euro_df_monthly <- get_monthly_avg(dolar_euro_df, "EUR_USD")
yuan_df_monthly <- get_monthly_avg(yuan_df, "USD_CNY")
yen_df_monthly <- get_monthly_avg(yen_df, "USD_JPY")
real_df_monthly <- get_monthly_avg(real_df, "USD_BRL")
```

```
# Merge dataframes
currency_df <- merge(euro_df_monthly, yuan_df_monthly, by = "Date", all = TRUE)
currency_df <- merge(currency_df, yen_df_monthly, by = "Date", all = TRUE)
currency_df <- merge(currency_df, real_df_monthly, by = "Date", all = TRUE)

fed_df <- fed_df %>%
  mutate(Date = format(as.Date(Date), "%Y-%m"))

currency_df <- merge(currency_df, fed_df, by = "Date", all = TRUE)
currency_df <- na.omit(currency_df)

currency_numeric_df <- currency_df %>% select_if(is.numeric)
colnames(currency_numeric_df) <- gsub("USD|_", "", colnames(currency_numeric_df))

# Correlation matrix
correlation_matrix <- cor(currency_numeric_df)

# Heatmap visualization
correlation_melted <- melt(correlation_matrix)

ggplot(correlation_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "#1ac5ca", high = "#f87970", mid = "#ffffff", midpoint =
0) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation between currencies and FED rate", x = NULL, y = NULL)
```

The heatmap reveals a strong correlation between the rise in the FED's interest rates and the depreciation of currencies, particularly the Chinese Yuan and Japanese Yen. This confirms the global impact of the FED's decisions.

Finally, we explore the lag effect of FED interest rates on currency exchange rates:

```
# FED rate lags
merged_df_lags <- merged_df %>%
  select(Date, starts_with("FED_Lag"))
merged_df_lags <- merged_df_lags %>%
  mutate(Date = format(as.Date(Date), "%Y-%m"))

currency_df <- merge(currency_df, merged_df_lags, by = "Date", all = TRUE)

# Model creation
model_data <- currency_df %>%
  select(USD_CNY, starts_with("FED")) %>%
  na.omit()  # Remove rows with NA

set.seed(54)  # For reproducibility
train_index <- sample(1:nrow(model_data), 0.8 * nrow(model_data))
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]

lm_model <- lm(USD_CNY ~ ., data = train_data)
summary(lm_model)
```

```
# Model evaluation
predictions <- predict(lm_model, newdata = test_data)

mse <- mean((test_data$USD_CNY - predictions)^2)
r2 <- summary(lm_model)$r.squared

cat("MSE:", mse, "\nR²:", r2, "\n")
```

The R^2 value of the model indicates that about 70% of the variability in the USD/CNY exchange rate can be explained by the FED rate. Interestingly, unlike inflation, lag effects do not appear to be significant in the model, suggesting that the impact is immediate.

---

8 **Conclusions**

We studied the relationship between the FED's interest rate and inflation, focusing primarily on the evolution of the Consumer Price Index (CPI). We began by obtaining historical data for both the FED's interest rate and the CPI since 1990, and analyzed the correlation between the two variables. The initial analysis showed that the relationship is not strong on a monthly basis, but becomes more evident when observed on an annual level.

We performed a Granger causality test to understand if the FED's interest rate affects inflation (and vice versa). The results confirmed that the FED's rate causes the CPI, indicating that monetary policy decisions have a significant impact on inflation.

When analyzing the lags of the FED's interest rate, we found a negative correlation, suggesting that increases in the interest rate tend to be associated with a decrease in inflation, although this relationship is weak and gradual. In other words, the effect of monetary policy is not immediate and persists for several months. Despite this weak relationship, the results are consistent with the idea that the FED's decisions have a long-term effect.

We then incorporated other CPI variables, such as core CPI and inflation expectations, to see if they could provide a more accurate perspective. By training a regression model and a Random Forest model to predict core CPI, we found that, although the model explains about 50% of the variability in core CPI, the relationship with the lags of the FED's rate is not complete. However, the Random Forest model showed that monetary policy has a significant impact, but this effect takes time to manifest.

When adjusting the model's hyperparameters, we observed that a moderate adjustment is sufficient to achieve good results, without the need for extreme optimization. This was validated with out-of-bag (OOB) error and the coefficient of determination (R^2).

Finally, we incorporated exchange rate data to observe the global impact of the FED's interest rate. We observed that the impact of the FED's rate on foreign currencies, such as the Yuan and the Yen, is immediate, with lags having no significance in the model. This reinforces the idea that, unlike inflation, the effect of the FED's monetary policy on exchange rates is immediate and direct.

In summary, while the FED's interest rate has a significant impact on inflation with gradual and long-term effects, its influence on exchange rates is immediate, especially in currencies like the Yuan and the Yen.