

technical_trading_strategy_r

2025-12-01

Implementing a dual moving average crossover strategy on “Netflix” stock data, calculate key portfolio performance metrics, and compare the results to those of a simple buy-and-hold approach. This comparison will help assess the strategy’s effectiveness in timing market entry and exit points, as well as its potential for managing risk.

1. Technical Implementation and Visualization

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

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

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

```
## Loading required package: TTR
```

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

```
library(PerformanceAnalytics)
```

```
##  
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':  
##  
##      legend
```

```
#1. selecting the NETFLIX Stock:  
ticker <- "NFLX"  
  
#2. date period - 1 jan 2019 to 31 dec 2024  
start_date <- "2019-01-01"  
end_date   <- "2024-12-31"  
getSymbols(ticker, src = "yahoo", from = start_date, to = end_date)
```

```
## [1] "NFLX"
```

```
data <- get(ticker)

# Adjusted closing prices
price <- Ad(data)

#3. fast(SMA 20) and slow (SMA 100) sma computation:
SMA20 <- SMA(price, n = 20)      # Fast SMA
SMA100 <- SMA(price, n = 100)    # Slow SMA

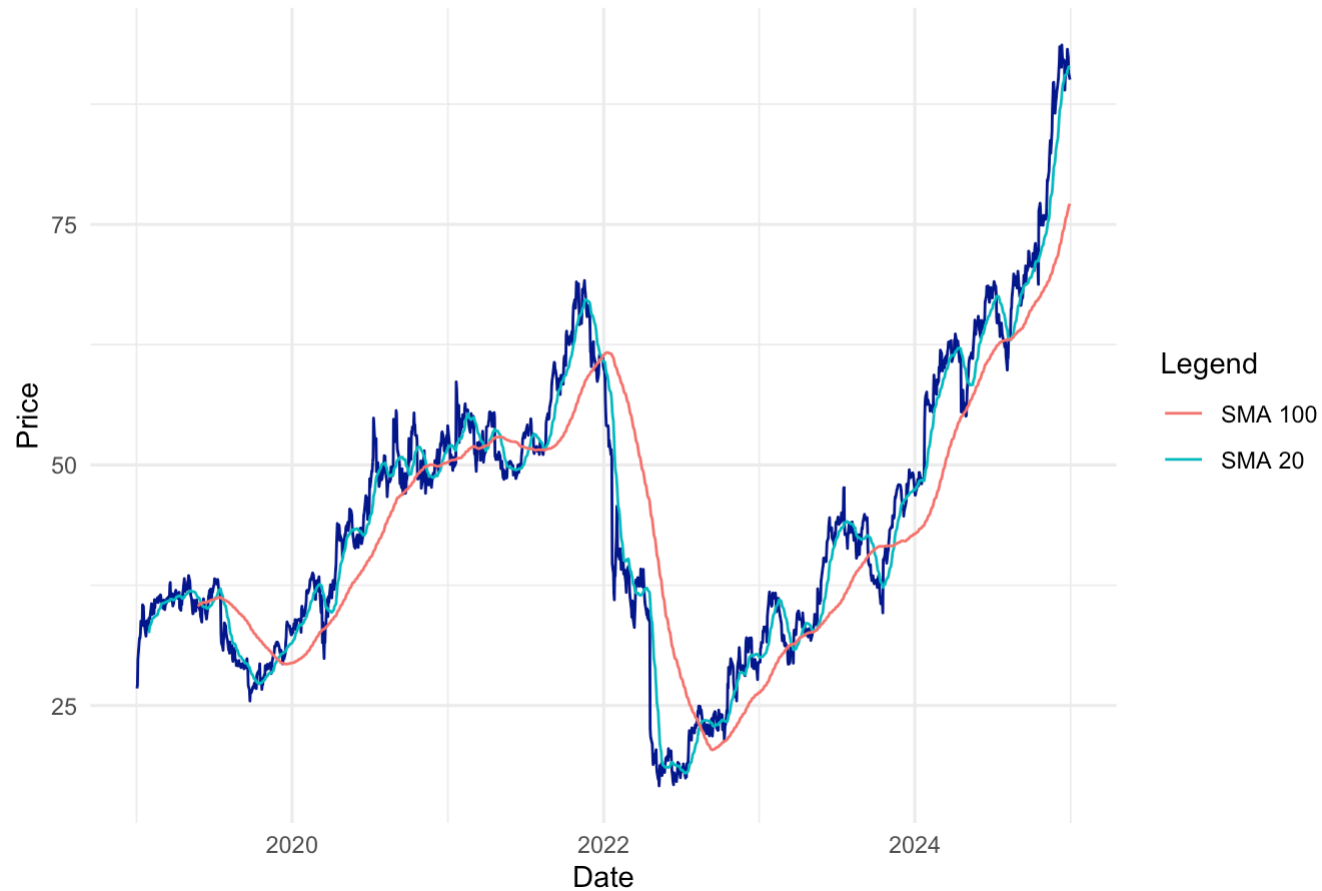
#4. Plotting of NFLX closing price, the SMA 20, and the SMA 100
library(ggplot2)
df <- data.frame(
  date = index(price),
  price = as.numeric(price),
  SMA20 = as.numeric(SMA20),
  SMA100 = as.numeric(SMA100)
)

ggplot(df, aes(x = date)) +
  geom_line(aes(y = price), color = "darkblue") +
  geom_line(aes(y = SMA20, color = "SMA 20")) +
  geom_line(aes(y = SMA100, color = "SMA 100")) +
  labs(title = paste(ticker, "Closing Price with SMA20 and SMA100"),
       y = "Price", x = "Date", color = "Legend") +
  theme_minimal()
```

```
## Warning: Removed 19 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 99 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

NFLX Closing Price with SMA20 and SMA100



Interpretation:

The graph shows Netflix's closing price with the 20-day (fast) and 100-day (slow) moving averages. The fast SMA follows the price more closely, while the slow SMA captures the broader trend. Buy and sell signals occur when the fast SMA crosses above or below the slow SMA. In early 2022, the fast SMA crossed below the slow SMA, signaling a sell and avoiding losses, while in 2023–2024 it crossed above, signaling a buy during the uptrend.

2. Strategy Signal Generation and Portfolio Return:

```
#5. Buy/Long Position (Invested = 1): SMA 20 crosses above SMA 100.
```

```
# trading signal
```

```
# Invested = 1 when SMA20 > SMA100
```

```
signal <- ifelse(SMA20 > SMA100, 1, 0)
```

```
# Lag by 1 day
```

```
signal_lag <- Lag(signal, 1)
```

```
# First day = 0 (in cash)
```

```
signal_lag[is.na(signal_lag)] <- 0
```

```
#6. daily portfolio returns
```

```
ret <- dailyReturn(price)
```

```
# Portfolio returns = stock returns * trading position
```

```
port_ret <- ret * signal_lag
```

```
#7. cumulative portfolio value - starting $1
```

```
# Strategy cumulative returns
```

```
cum_port <- cumprod(1 + port_ret)
```

```
# Buy & Hold cumulative returns
```

```
cum_buy <- cumprod(1 + ret)
```

3. Performance Evaluation

```
#8. annualized Sharpe Ratio, at Risk-Free Rate of 0.02.
```

```
rf <- 0.02
```

```
# converting to daily rate
```

```
rf_daily <- rf / 252
```

```
# Annualized Sharpe Ratio
```

```
sharpe_port <- SharpeRatio.annualized(port_ret, Rf = rf_daily)
```

```
sharpe_bh <- SharpeRatio.annualized(ret, Rf = rf_daily)
```

```
#9. Maximum Drawdown (Max DD)
```

```
mdd_port <- maxDrawdown(port_ret)
```

```
mdd_bh <- maxDrawdown(ret)
```

```
cat("Final Value (Dual SMA Strategy):", round(last(cum_port), 4), "\n")
```

```
## Final Value (Dual SMA Strategy): 3.1273
```

```
cat("Final Value (Buy & Hold):", round(last(cum_buy), 4), "\n\n")
```

```
## Final Value (Buy & Hold): 3.3641
```

```
cat("Annualized Sharpe Ratio (Dual SMA):", round(sharpe_port, 4), "\n")
```

```
## Annualized Sharpe Ratio (Dual SMA): 0.5723
```

```
cat("Annualized Sharpe Ratio (Buy & Hold):", round(sharpe_bh, 4), "\n\n")
```

```
## Annualized Sharpe Ratio (Buy & Hold): 0.4565
```

```
cat("Max Drawdown (Dual SMA):", round(mdd_port, 4), "\n")
```

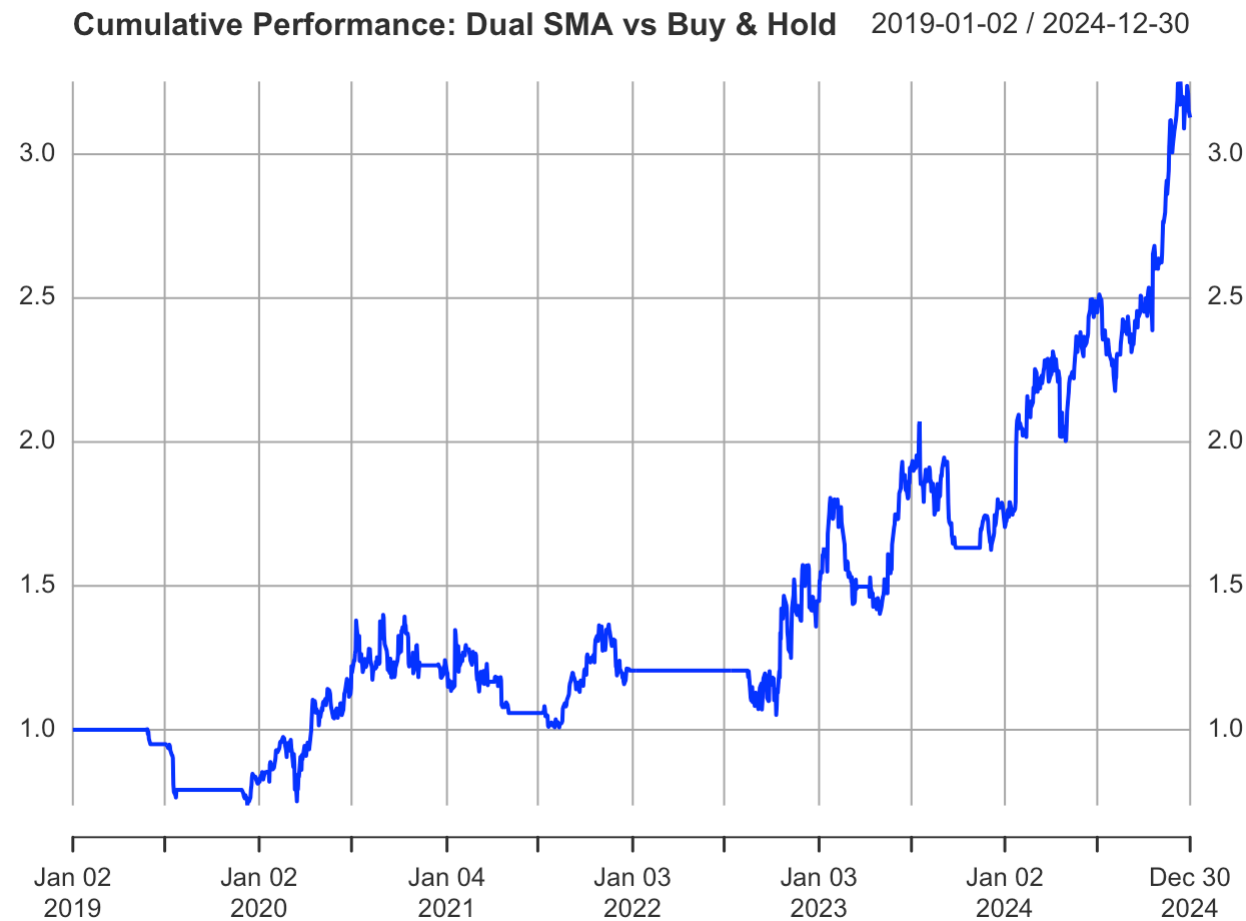
```
## Max Drawdown (Dual SMA): 0.2794
```

```
cat("Max Drawdown (Buy & Hold):", round(mdd_bh, 4), "\n")
```

```
## Max Drawdown (Buy & Hold): 0.7595
```

```
#10. PLOT TO VISUALIZE
```

```
plot(cum_port, main = "Cumulative Performance: Dual SMA vs Buy & Hold",  
     col = "blue", lwd = 2)
```



```
lines(cum_buy, col = "black", lwd = 2)
legend("topleft",
      legend = c("Dual SMA Portfolio", "Buy & Hold"),
      col = c("blue", "black"), lwd = 2)
```

Cumulative Performance: Dual SMA vs Buy & Hold 2019-01-02 / 2024-12-30



Interpretation:

This graph compares the cumulative investment growth for the Dual SMA strategy and a simple buy-and-hold approach. While the buy-and-hold strategy achieves a slightly higher final value, it experiences significantly larger drawdowns, particularly during the 2022 market drop. In contrast, the Dual SMA strategy shows smoother growth with smaller losses during downturns, reflecting lower risk and better drawdown management.

The visual trade-off between total return and risk aligns with the calculated Sharpe ratios and maximum drawdowns, confirming that the Dual SMA strategy provides more stable, risk-adjusted performance.

4. Analysis

The Dual SMA strategy invests in the stock when the 20-day SMA crosses above the 100-day SMA and remains invested until the signal reverses. Over the selected period, the strategy achieved a final cumulative value of 3.1273, compared to 3.3641 for a simple Buy & Hold strategy.

In terms of risk-adjusted performance, the annualized Sharpe ratio for the Dual SMA strategy is 0.5723, higher than the Buy & Hold Sharpe ratio of 0.4565, indicating that the strategy provided better risk-adjusted returns despite generating slightly lower absolute returns. Additionally, the maximum drawdown of the Dual SMA strategy is 0.2794, substantially lower than the Buy & Hold drawdown of 0.7595, suggesting that the SMA strategy successfully limited large losses during market downturns.

Hence, it can be said that while the Buy & Hold strategy delivered slightly higher total returns, the Dual SMA Crossover strategy improved risk management and provided more stable returns. This shows that the strategy was effective in reducing downside risk while capturing a reasonable portion of the stock's gains over the period.

KNN

Constructing a trading strategy using the same stock and time period, applying the k-Nearest Neighbors algorithm or regression with k-fold cross-validation, and evaluating its performance.

```
library(caret)
```

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

```
library(PerformanceAnalytics)  
library(quantmod)  
library(dplyr)
```

```
##
## ##### Warning from 'xts' package #####
## #
## # The dplyr lag() function breaks how base R's lag() function is supposed to #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or #
## # source() into this session won't work correctly. #
## #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #
## # dplyr from breaking base R's lag() function. #
## #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
## #
## #####
```

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

```
## The following objects are masked from 'package:xts':
##
## first, last
```

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

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

#1. Feature Engineering

```
ret_1 <- Lag(ret, 1)
ret_2 <- Lag(ret, 2)
sma_diff <- SMA20 - SMA100
next_dir <- ifelse(dplyr::lead(ret, 1) > 0, 1, 0)
```

Combining into xts

```
df_xts <- cbind(ret_1, ret_2, sma_diff, next_dir)
colnames(df_xts) <- c("ret_1", "ret_2", "sma_diff", "next_dir")
```

Removing missing values

```
df_xts <- na.omit(df_xts)
```

```
df <- data.frame(
  ret_1 = coredata(df_xts[, "ret_1"]),
  ret_2 = coredata(df_xts[, "ret_2"]),
  sma_diff = coredata(df_xts[, "sma_diff"]),
  next_dir = factor(coredata(df_xts[, "next_dir"]), levels = c(0,1), labels = c("Down","Up"))
)
```

#2. Train-Test Split performance

```
n <- nrow(df)
train_index <- 1:floor(0.7*n)
train_df <- df[train_index, , drop = FALSE]
test_df <- df[(max(train_index)+1):n, , drop = FALSE]
```

#3. Training KNN with 5-Fold CV

```
train_control <- trainControl(method = "cv", number = 5)
set.seed(123)
```

```
knn_fit <- train(
  next_dir ~ ret_1 + ret_2 + sma_diff,
  data = train_df,
  method = "knn",
  trControl = train_control,
  tuneGrid = data.frame(k = c(3,5,7))
)
```

```
#4. Prediction on the test data
pred <- predict(knn_fit, newdata = test_df)
pred_num <- ifelse(pred == "Up", 1, 0)

#5. Portfolio Returns
# Slice returns for test period and make xts
test_dates <- index(df_xts)[(max(train_index)+1):nrow(df_xts)]
test_ret_xts <- xts(coredata(ret[(length(ret) - nrow(test_df) + 1):length(ret)]), order.by = test_dates)

#6. Portfolio returns: investment only if predicted Up
port_ret_knn_xts <- xts(pred_num, order.by = test_dates) * test_ret_xts
cum_port_knn <- cumprod(1 + port_ret_knn_xts)
cum_buy_test <- cumprod(1 + test_ret_xts) # Buy & Hold same period

#7. Evaluation
rf_daily <- 0.02 / 252
sharpe_knn <- SharpeRatio.annualized(port_ret_knn_xts, Rf = rf_daily)
mdd_knn <- maxDrawdown(port_ret_knn_xts)

cat("Final Value (kNN Strategy):", round(last(cum_port_knn), 4), "\n")
```

```
## Final Value (kNN Strategy): 1.287
```

```
cat("Final Value (Buy & Hold):", round(last(cum_buy_test), 4), "\n\n")
```

```
## Final Value (Buy & Hold): 2.7916
```

```
cat("Annualized Sharpe Ratio (kNN):", round(sharpe_knn, 4), "\n")
```

```
## Annualized Sharpe Ratio (kNN): 0.6479
```

```
cat("Max Drawdown (kNN):", round(mdd_knn, 4), "\n")
```

```
## Max Drawdown (kNN): 0.196
```

Results -

- The KNN strategy ended with a final value of 1.287 while Buy & Hold reached 2.7916, indicating lower absolute returns.
- The Sharpe ratio (0.6479) is higher than Buy & Hold (0.4565), showing better risk-adjusted performance.
- Max drawdown (0.196) is much lower than Buy & Hold (0.7595), showing the strategy reduced downside risk significantly.

The KNN trading strategy predicted the next-day stock direction using past returns and SMA differences. During the test period, the portfolio grew to 1.287, while simply buying and holding the stock would have grown it to 2.792. However, the strategy achieved a higher Sharpe ratio (0.648 vs 0.457) and a lower maximum drawdown (0.196 vs 0.760), showing better risk-adjusted performance and smaller potential losses.