NatGas Example with Chan's approach

Alexander Arzt

11-3-2021

Chan's approach to meta labeling

In this short example we will look at Ernie Chan's approach to meta labeling. It is different to de Prado's approach in the way that it simply uses past trades of a strategy and meta labels them (so no triple barrier method used).

Let's prepare the data first:

```
features <- readxl::read_xlsx("features_diff_ng.xlsx")
features <- features[,-1]

result <- read.csv("set1ng.csv", sep = ";")
result <- result[,3]

df <- cbind(features, result)
df <- df[,c(1,6,8)]


n <- dim(df)[1]
df$result <- ifelse(df$result ==1, 1, 0)
df$result <- as.factor(df$result)


df2 = read.csv("ngeod.csv", sep = ";")
df2 = df2[,c(2,3)]
df2$TradeReturn <- df2$TradeReturn+1
df2$TradeReturn <- log(df2$TradeReturn)
```

So this is what the data looks like: two features and the column containing the meta labels:

```
head(df)
```

```
##   Parkvol absmomo60 result
## 1 2.20082     0.082      1
## 2 2.09844     0.045      0
## 3 2.00474     0.054      1
## 4 1.90653     0.089      0
## 5 1.90920     0.058      1
## 6 2.00814     0.129      1
```

Fit secondary model

The results of our primary model (our trading strategy) are meta labeled in the results column of the df data frame. Now we will fit a KNN classifier to it and do out of sample predictions in walk forward manner.

```
steps <- 300
library(kknn)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
preds <- rep(NA,steps)

for (i in 1:steps) {

  window <- (899+i):((n-steps)-1+i)
  test_thresh <- ((n-steps)+i)
  train_df <- df[window,]
  test_df <- df[test_thresh,]

  fit <- kknn(result~., train=train_df, test = test_df, k=7)
  preds[i] <- predict(fit)
}

preds <- as.factor(preds-1)


true <- df$result[((n-steps)+1):n]

result <- confusionMatrix(true,preds)
result
```

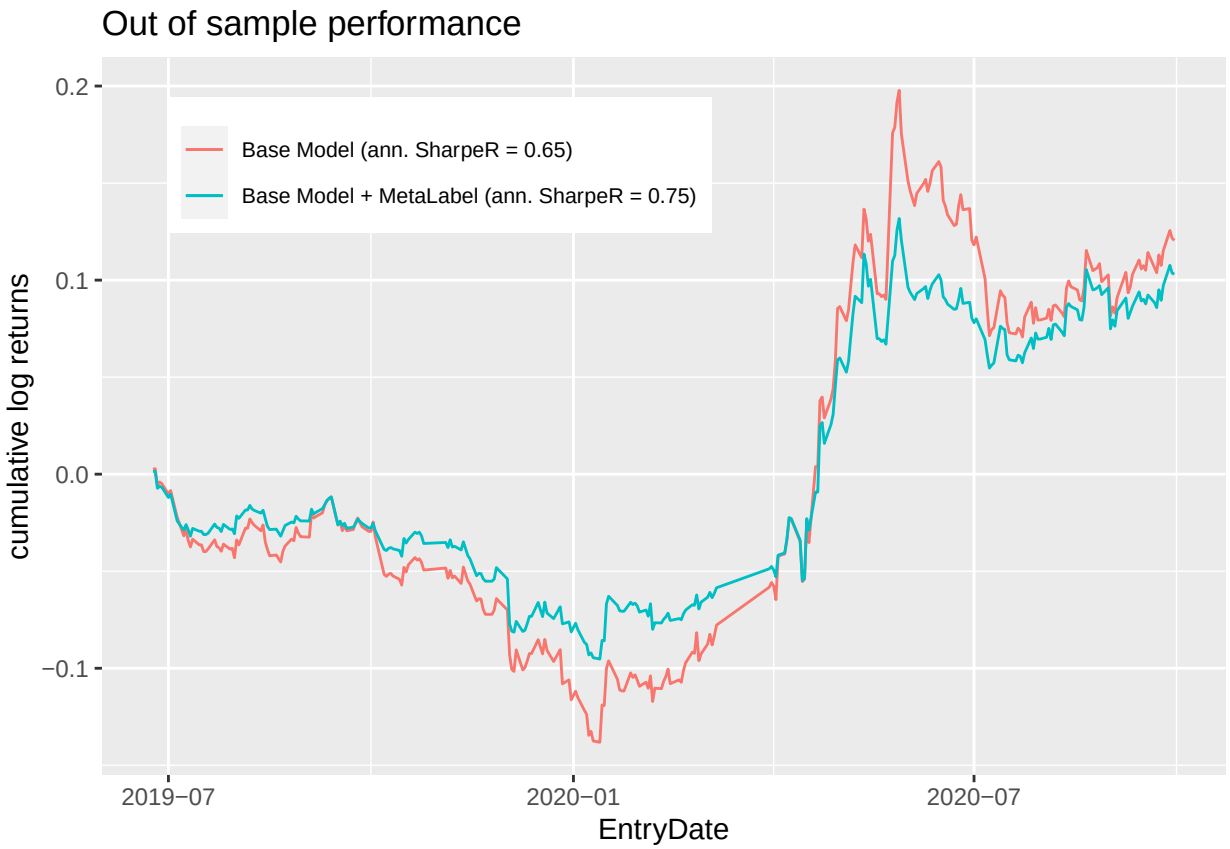
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 85 71
##          1 64 80
##
##                Accuracy : 0.55
##                  95% CI : (0.4918, 0.6072)
##     No Information Rate : 0.5033
##     P-Value [Acc > NIR] : 0.05942
##
##                   Kappa : 0.1002
##
##  Mcnemar's Test P-Value : 0.60558
##
##             Sensitivity : 0.5705
##             Specificity : 0.5298
##          Pos Pred Value : 0.5449
##          Neg Pred Value : 0.5556
##              Prevalence : 0.4967
##          Detection Rate : 0.2833
##    Detection Prevalence : 0.5200
##       Balanced Accuracy : 0.5501
##
##        'Positive' Class : 0
##
```

```
result$byClass[7]
```

```
##       F1
## 0.557377
```

Comparision: primary model vs. primary model + secondary model

```
library(dplyr)

df2$EntryDate <- as.character(df2$EntryDate)
df2$EntryDate <- as.Date(df2$EntryDate, "%Y%m%d")
df2 <- df2[seq(n,1),]
df2 <- df2[1490:n, ]
df2 <- df2 %>% mutate(cumsum = cumsum(TradeReturn))

# With Meta model
df3 <- cbind(df2,preds)
df3$size <- ifelse(df3$preds == 1, 1, 0.5)
df3$new_ret <- df3$TradeReturn*df3$size
df3 <- df3 %>% mutate(new_cumsum = cumsum(new_ret))

df2$new_cums <- df3$new_cumsum

oldSharpe = (mean(df3$TradeReturn)/sd(df3$TradeReturn))*sqrt(252)
newSharpe = (mean(df3$new_ret)/sd(df3$new_ret))*sqrt(252)

ggplot(data = df2, aes(x=EntryDate)) +
  geom_line(aes(y=cumsum, color="Base Model (ann. SharpeR = 0.65) ")) +
  geom_line(aes(y=new_cums, color="Base Model + MetaLabel (ann. SharpeR = 0.75)")) +
  theme(legend.position = c(.3, .85))+
  theme(legend.title = element_blank())+
  labs(y = "cumulative log returns")+
  theme(legend.text = element_text(size = 8))+
  ggtitle("Out of sample performance")
```


Out of sample performance
Base Model (ann. SharpeR = 0.65)
Base Model + MetaLabel (ann. SharpeR = 0.75)
cumulative log returns
0.2
0.1
0.0
−0.1
2019−07
2020−01
2020−07
EntryDate