Построение прогнозирующей модели

Голубев А.А., группа ИУ9-11м
МГТУ им. Н.Э. Баумана

Москва — 2019

Постановка задачи

Ответить на вопрос:

можно ли прогнозировать рост цены акции по следующим величинам:

- история цен данной акции;
- некоторое множество биржевых индексов.

Выбор признаков для прогнозирования

- выбрана акция Сбербанка как одна из наиболее ликвидных;
- период: 2014 – 2019 гг. ежедневно с 11 до 19 часов;

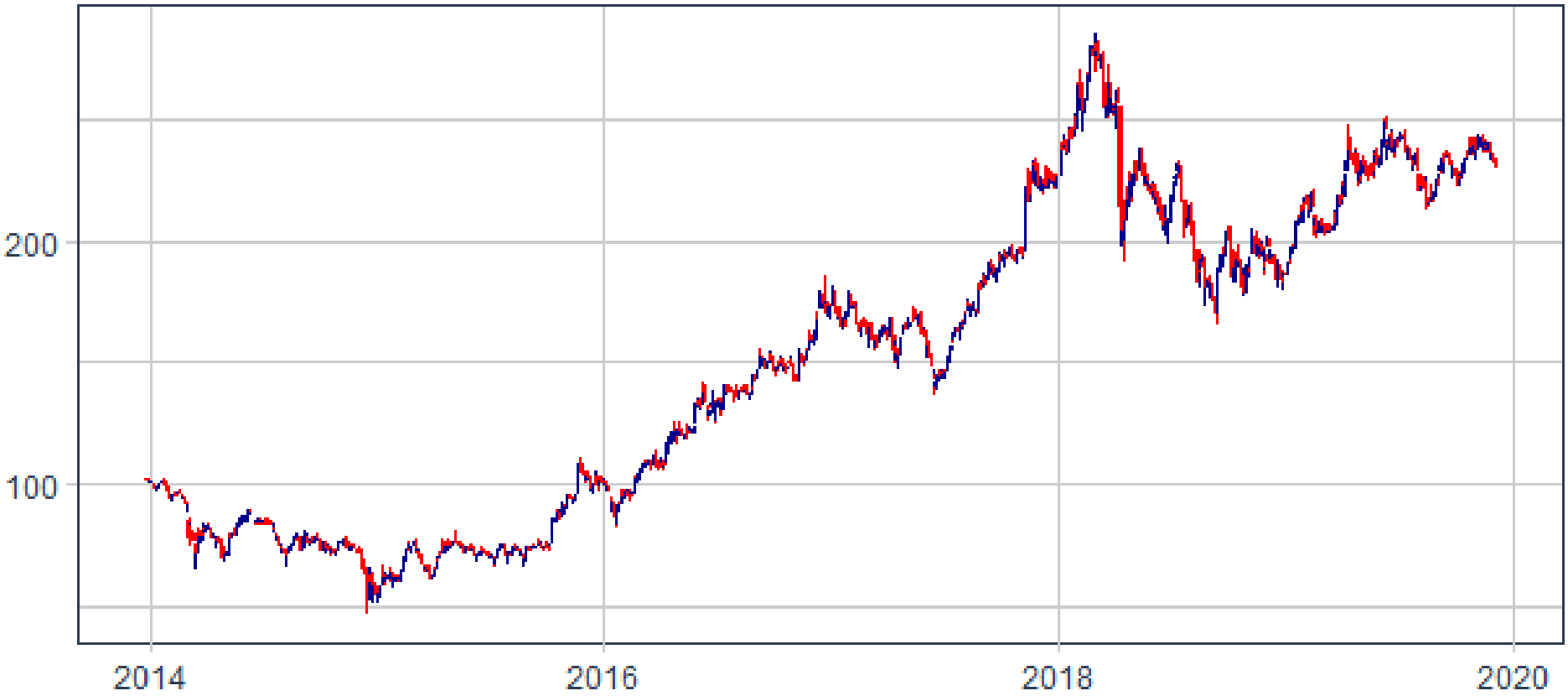
Признаки за каждый торговый день на периоде:

- направленный объем;
- OPEN, LOW, HIGH, CLOSE, Volume характеристики;
- биржевые индексы imoex, imoex10.

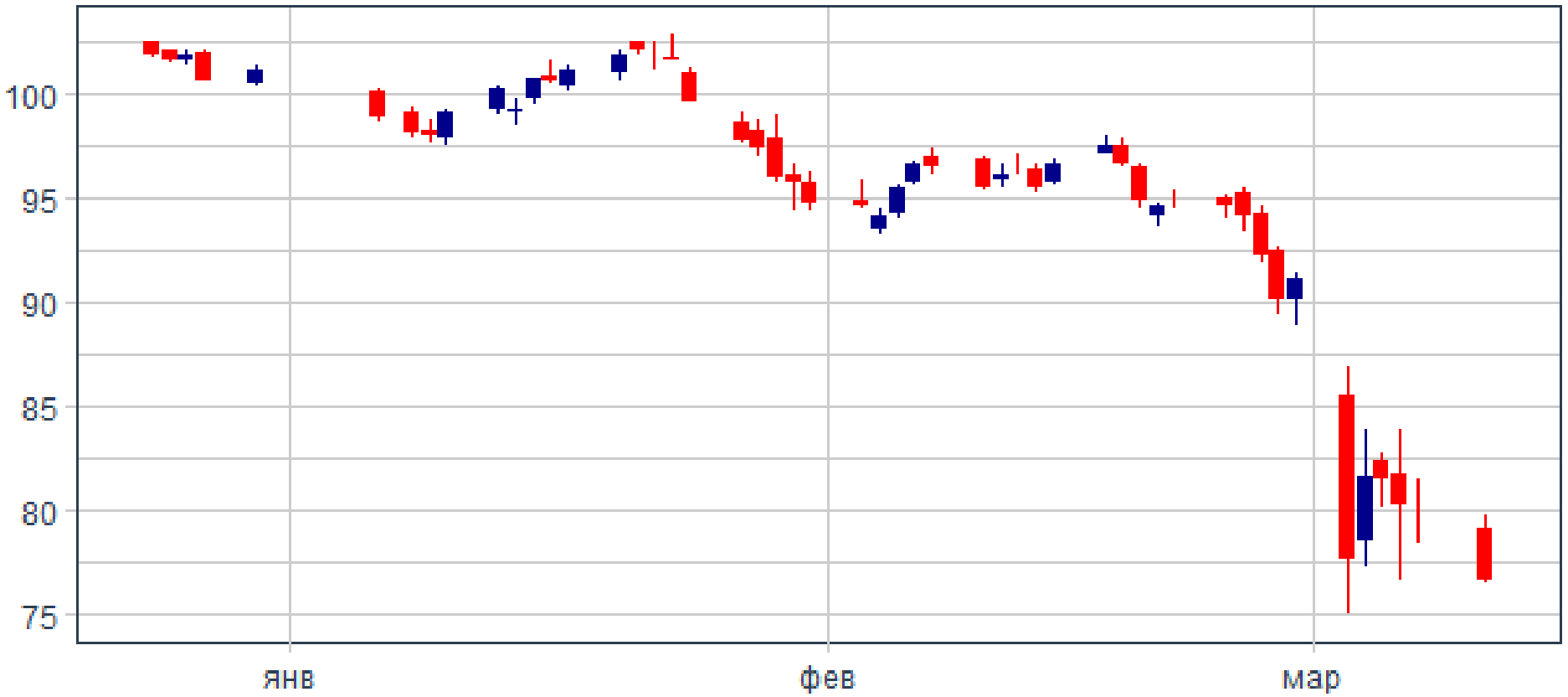
Вид данных

	A	B	C	D	E	F
1	time	OPEN	HIGH	LOW	CLOSE	VOL
2	24.12.2013	102.51	102.54	101.75	101.96	40138690
3	25.12.2013	102.2	102.22	101.53	101.62	22072070
4	26.12.2013	101.61	102.18	101.45	101.95	22183460
5	27.12.2013	102.09	102.21	100.68	100.68	36832780
6	30.12.2013	100.51	101.39	100.45	101.17	34094760
7	06.01.2014	100.2	100.31	98.62	98.91	31402070
8	08.01.2014	99.1	99.41	97.85	98.19	42304620
9	09.01.2014	98.24	98.77	97.69	98	45619030
10	10.01.2014	97.85	99.3	97.52	99.2	51181490
11	13.01.2014	99.3	100.35	99.04	100.25	61750730
12	14.01.2014	99.15	99.73	98.51	99.24	78195860
13	15.01.2014	99.8	100.79	99.56	100.77	77459850
14	16.01.2014	100.9	101.64	100.53	100.65	61767390
15	17.01.2014	100.36	101.45	100.1	101.17	68773290
16	20.01.2014	100.98	102.17	100.66	101.96	62337570
17	21.01.2014	102.51	102.57	101.9	102.2	61308500

Свечной график



Свечной график в масштабе



Предобработка данных

- расчет направленного объема по часовым дням;
  - за каждый день высчитываем объем продаж — сумма объемов часовых интервалов с 11 до 19 в пределах каждого дня в течение которых цена снизилась больше порогового уровня;
  - аналогично объем покупок — сумма объемов часовых интервалов с 11 до 19 в пределах каждого дня в течение которых цена возросла больше порогового уровня;
  - остальные дни — нейтральный объем.
- расчет бинарных меток ответов за каждый день (рост цены=1);
  - метка для текущего дня = 1, если с предыдущего дня цена выросла больше порогового значения (1-2 %);
- нормализация данных средстами R (функция rescale)
- разделение данных на тренировочную и тестовую выборки (70%/30%)

```
13  #load data
14  data <- read.csv('SBERBANK_Learning_Data_02_lookback_next.txt', sep = '\t')
15  data$time <- as.Date(data$time, "%d.%m.%Y")
16
17  # data normalization
18  for (i in colnames(data)){
19    data[[i]] <- rescale(data[[i]])
20  }
21
22  #train/test
23  data$CLASS <- factor(target_buy, levels = c(0,1))
24  train.idx<-seq(1,round(length(data$CLASS)*0.7),1)
25
26  train <- data[train.idx,]
27  test <- data[-train.idx,]
28
29  # targets
30  target_train=train$CLASS
31  target_test=test$CLASS
32
33  # избавляемся от меток в датафрейме
34  new_tr <- model.matrix(~.+0,data = train[,-c(length(colnames(data))[1])])
35  new_ts <- model.matrix(~.+0,data = test[,-c(length(colnames(data))[1])])
36
```

Постановка задачи классификации

- обучение с учителем;
- нужно предсказывать такие дни, на следующий день после которых цена возрастает более чем на заданный в % порог;
- только количественные признаки, категорийных нет;
- функция потерь = количество ошибок классификации.

- Статистическая модель, используемая для прогнозирования вероятности возникновения некоторого события путём подгонки данных к логистической кривой.

$$\varphi(\omega, x) = \frac{1}{1 + e^{-\omega^T x}},$$

- ω — обучаемый вектор весов;
- x — вектор признаков элемента выборки.
- значение $\varphi(\omega, x) > 0.5 \to 1$, иначе 0;
- при настройке алгоритма находятся веса, соответствующие минимуму логарифмической функции правдоподобия:

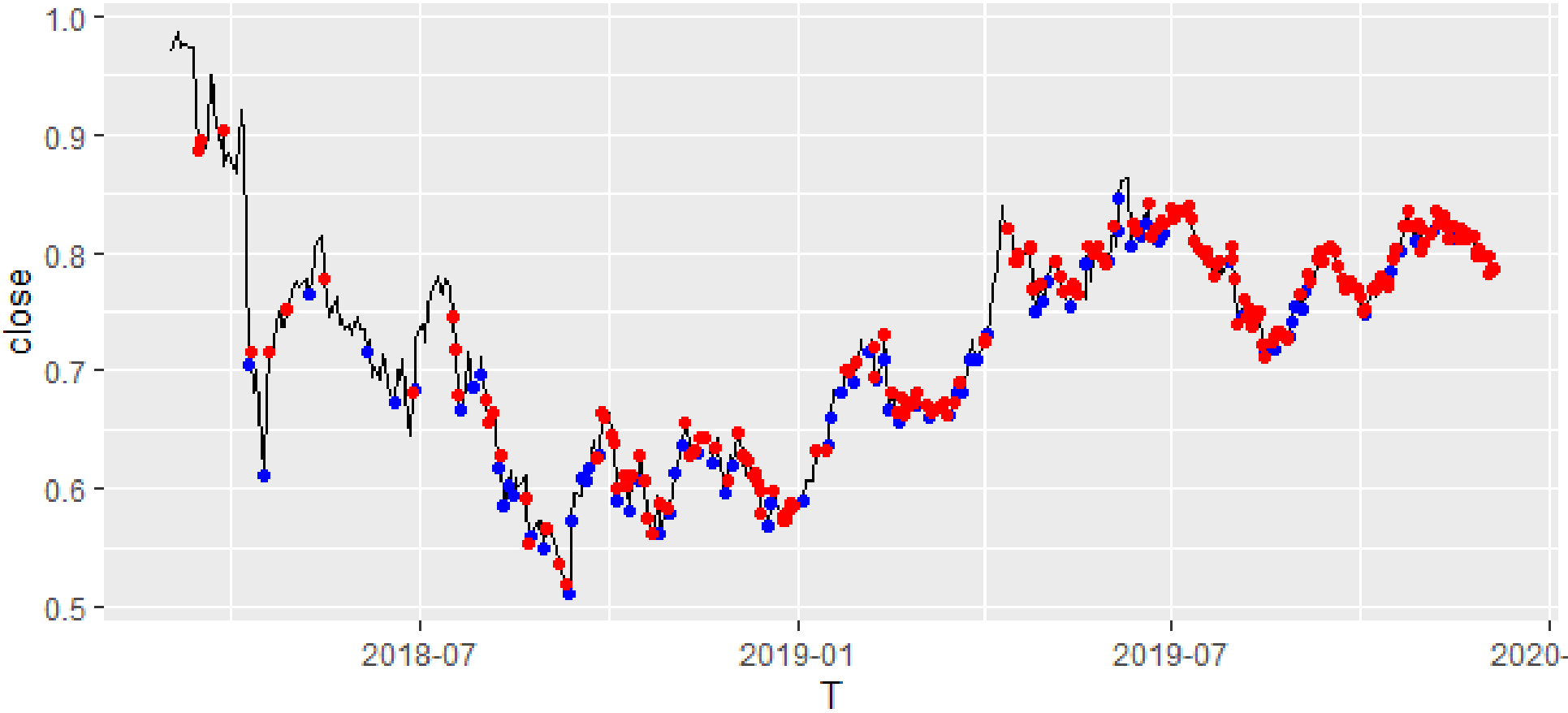
$$J(\omega) = \sum_{1}^{n} \left[- y^{(i)} \log \left(\varphi(\omega^T x^{(i)}) \right) - \left(1 - y^{(i)}\right) \log \left(1 - \varphi(\omega^T x^{(i)})\right) \right]$$

```
50  # fitting model
51  logit <- glm(CLASS~., data = train, family=binomial)
52  summary(logit)
53
54  decision <- predict(logit, newdata = test,type="response")
55  decision_flag <- ifelse(decision>= 0.5, 1, 0)
56  table(test$CLASS, decision_flag, dnn=c("Actual","Predicted"))
```

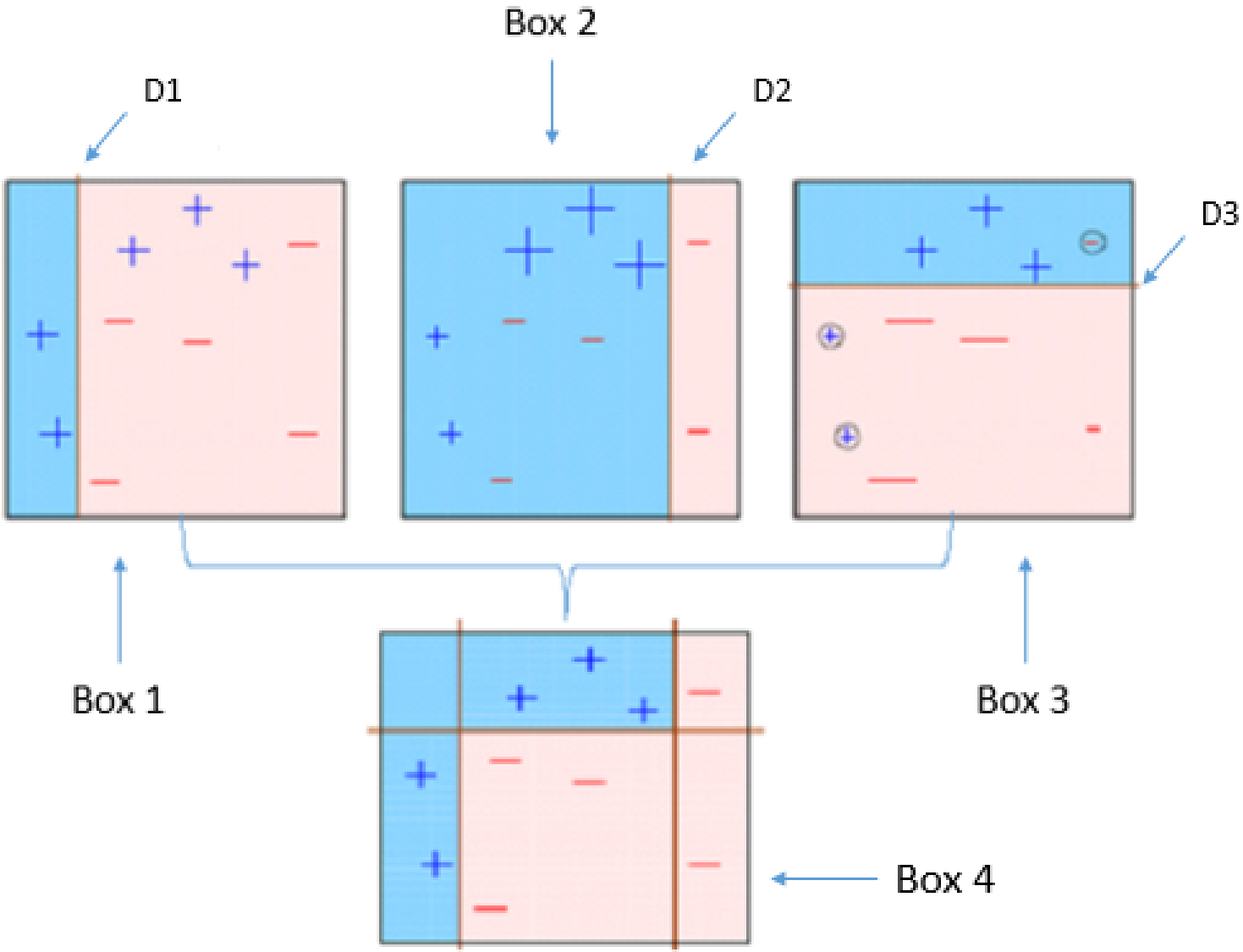
```
      Predicted
Actual   0   1
     0 194 138
     1  64  53
> summary(logit)

Call:
glm(formula = CLASS ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6760  -0.8304  -0.6813   1.2302   2.2481
```

Рис.: Матрица ошибок и распределение остатков, $acc = 0.57$

Результаты



- градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей;
- на каждой итерации вычисляются отклонения предсказаний уже обученного ансамбля на обучающей выборке.
- следующая модель, которая будет добавлена в ансамбль будет предсказывать эти отклонения;
- добавив предсказания нового дерева к предсказаниям обученного ансамбля мы можем уменьшить среднее отклонение модели, которое является целью оптимизационной задачи;
- новые деревья добавляются в ансамбль до тех пор, пока ошибка уменьшается, либо пока не выполняется одно из правил "ранней остановки".


Box 2
D1
D2
D3
Box 1
Box 3
Box 4

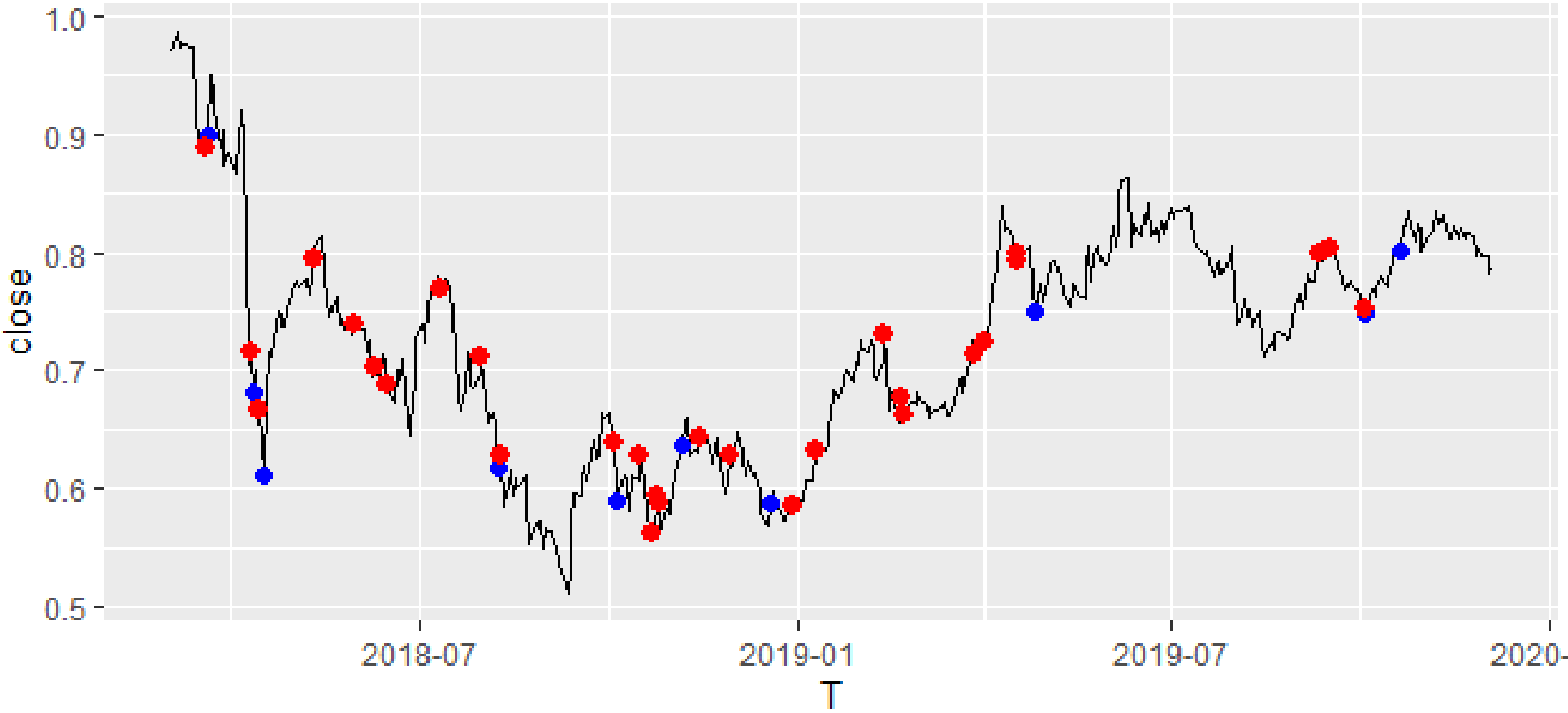
Настройка модели и результаты

```r
#preparing model
params <- list(booster = "gbtree", objective = "binary:logistic", eta=0.3,
               gamma=0, max_depth=6, min_child_weight=1, subsample=1, colsample_bytree=1)
xgbcv <- xgb.cv( params = params, data = dtrain, nrounds = 100, nfold = 5,
                 showsd = T, stratified = T, print_every_n = 10, early_stop_round = 20, ma
xgb1 <- xgb.train (params = params, data = dtrain, nrounds = 79,
                   watchlist = list(val=dtest,train=dtrain), print_every_n = 10, early_st

#model prediction
xgbpred <- predict (xgb1,dtest)
xgbpred <- as.numeric(ifelse (xgbpred > 0.4,1,0))
```

```
      Predicted
Actual   0   1
     0 268  64
     1  89  28
> summary(xgb1)
                Length Class              Mode
handle               1 xgb.Booster.handle externalptr
raw             134181 -none-             raw
niter                1 -none-             numeric
evaluation_log       3 data.table         list
call                 9 -none-             call
params              11 -none-             list
callbacks            2 -none-             list
feature_names       60 -none-             character
nfeatures            1 -none-             numeric
>
```

Рис.: Матрица ошибок и распределение остатков, $acc = 0.69$

Результаты



Выводы

- для данной задачи использованная модель не позволяет достаточно точно прогнозировать динамику цены;
- необходимо рассматривать более длительную историю, а также более сложные признаки и модели.