УДК 004.852
DOI 10.22213/2410-9304-2018-1-15-25

К. С. Пивкин, аспирант
Удмуртский государственный университет

РЕАЛИЗАЦИЯ РЕГРЕССИОННЫХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ ТОВАРНОГО СПРОСА С ПОМОЩЬЮ ЯЗЫКА R

Рассматривается регрессионный анализ как ключевой метод прогнозирования величины товарного спроса. Приводится список методов, являющихся наиболее эффективными для расчета оценки прогноза: линейная регрессия с регуляризацией, регрессия на основе опорных векторов, метод случайного леса. Необходимые расчеты реализуются на языке программирования R с использованием как базового функционала, так и расширений, которые обеспечивают возможность использования рассматриваемых методов. В качестве входящих данных используются показатели работы магазина и товарные характеристики. Определяется метрика качества результата работы алгоритмов – среднеквадратическое отклонение ошибки. Проводится разделение выборки данных на обучающую и тестовую, последовательно рассчитываются результаты для каждого приведенного алгоритма. Делаются выводы о том, что для рассматриваемой выборки наилучший результат дает алгоритм случайного леса. Выводится степень корреляционной связи между прогнозами по разным алгоритмам, на основе которой высказывается предположение о возможном совместном использовании прогнозов. Исходя из этого строится простейшая комбинация алгоритмов – арифметическое среднее. Данный ансамбль алгоритмов оказался лучше всех рассмотренных методов прогнозирования, взятых по отдельности. Определяется план дальнейшего исследования по созданию комитета алгоритмов на основании методов бэггинга, бустинга или стэкинга.

Ключевые слова: товарный спрос, регрессионный анализ, язык R, машина опорных векторов, случайный лес, регуляризация, кросс-валидация, ансамбль алгоритмов.

Регрессионные методы прогнозирования товарного спроса: описание подхода

В рамках задачи прогнозирования товарного спроса рассматривается большое количество математических методов и моделей. Так как в случае с прогнозированием товарного спроса речь идет о прогнозировании величины, которая будет возникать в будущем с учетом фактора времени, то наиболее популярными методами прогнозирования в экономических системах, исходя из работы С. Г. Светунькова и И. С. Светунькова [1, 2], являются инструменты анализа временных рядов: модели скользящего среднего, авторегрессии, ARIMA, тренд-сезонные модели, например, модель Хольта – Винтерса и многие другие. Подходы, используемые в данном моделировании, исследователю важно понимать и использовать. Но еще важнее тот факт, что данным инструментарием круг рассматриваемых методик не ограничен.

В статье будут рассмотрены модели регрессионного анализа, с помощью которых можно обрабатывать данные как панельные, т. е. те данные, которые содержат сведения об одном и том же множестве объектов за ряд последовательных периодов времени [3]. Объектами в контексте данной задачи являются товары, по которым измеряется спрос. Ввиду подобного представления данных при их обработке реализуется системный подход к товарному подмножеству всех реализуемых в торговой сети позиций, объединенных сходными характеристиками, – товарной группе. Суть системного подхода заключается в том, что при моделировании на панельных данных исследователь учитывает ряд показателей и параметров (возможно, созданных искусственно), которые характеризуют специфику магазина, товарной группы и отдельных товарных кластеров внутри этой группы.

Предварительно изучив работы Стрижова [4] и Куна [5] по регрессионному анализу, в данной научной статье приводится ряд методов регрессионного анализа для решения задачи прогнозирования товарного спроса, такие как:

- линейная регрессия;

• линейная регрессия с регуляризацией: гребневая и лассо;

• регрессия, основанная на машине опорных векторов;

• случайный лес.

Основным результатом статьи может считаться настройка параметров модели для каждого метода в соответствии с принципами статистического обучения, сравнение результатов моделирования по заданному критерию качества и анализ возможности совместного использования регрессионных методов в моделировании товарного спроса.

Основные расчеты проводились с помощью языка программирования R и таких дополнительных пакетов, как:

• *dplyr* – для работы с данными;

• *caret* – для предобработки данных и реализации методов машинного обучения;

• *glmnet* – для реализации множественной линейной регрессии с регуляризацией;

• *e1071* – для реализации регрессии на машине опорных векторов (*SVR* – Support Vector Regression);

• *randomForest* – для реализации метода «случайный лес»;

• *parallel, doParallel, foreach* – для создания параллельных процедур вычислений параметров моделей.

Для начала приводится список всех переменных, используемых при моделировании целевой переменной «Спрос». Под спросом здесь понимается тот объем товаров в натуральном выражении, который реализуется в заданной точке продаж за единицу времени. В данном случае единица времени определяется как день. Ниже приведена таблица со всеми независимыми переменными, участвующими в моделировании (табл. 1).

Таблица 1. **Независимые переменные для моделирования товарного спроса**

Наименование переменной	Название переменной в скрипте R	Тип переменной	Содержательный смысл
Код товара	*code*	Категориальная	Идентификатор товарной позиции. Не участвует в обучении модели, но позволяет идентифицировать товар для оценки результата прогноза по товарам группы
Товарный кластер (товарная подгруппа)	*cluster*	Категориальная	Определенная подгруппа основной товарной группы, которая объединяет схожие по следующим характеристикам товары:
Лаговая переменная спроса	*newsalesL1, …, newsalesL7*	Количественная	Лаговые переменные спроса, сдвинутые от 1 до 7
Цена	*price*	Количественная	Стоимость товара за единицу продукции
Наличие акции	*isAction*	Категориальная	Категориальная переменная, которая определяет наличие ценовой акции на указанный момент времени по данному товару
Наличие акции лаг	*isActionL1*	Категориальная	Переменная, аналогичная показателю «Наличие акции», с лагом, равным 1
Уровень скидки	*levelDiscount*	Количественная	Показатель, которые отражает уровень скидки на товар в момент промоакции (величина по модулю)
Температура воздуха	*tempMean*	Количественная	Средняя температура воздуха в географической зоне точки продаж
Порядковый день года	*YOD*	Количественная	Порядковый номер дня в году
День недели	*weekday*	Категориальная	Показатель, характеризующий день недели, в который была совершена продажа товара
Наличие праздничного периода	*holiday*	Категориальная	Наличие праздничного периода. Определяются все календарные праздничные периоды (Рождество, новогодние каникулы и т. п.)

Окончание табл. 1

Наименование переменной	Название переменной в скрипте R	Тип переменной	Содержательный смысл
Порядковый день в праздничном периоде	*NDoH*	Количественная	Номер дня в праздничном периоде
Страна	*country*	Категориальная	Страна-производитель товара. Определяется исходя из информационной базы розничного предприятия. Также задаваемое количество стран ограничено исходя наблюдаемой дисперсии признака
Производитель	*makerDescr*	Категориальная	Наименование производителя товара. Определяется исходя из информационной базы розничного предприятия. Также задаваемое количество стран ограничено исходя из наблюдаемой дисперсии признака
Вес (емкость) товара	*weight*	Количественная	Показатель, который определяет физическую размерность товара. Является важным для выбора покупателя
Количество чеков	*chqNumber*	Количественная	Количество чеков, отбитых в торговой точке, за определенный день

Перед тем как проводятся основные этапы моделирования целевой переменой, даются основные характеристики выборки, по которым будет решаться искомая задача:

1. Исходная выборка формируется по одной товарной группе: в ее составе находится зависимая переменная «Спрос» и независимые переменные, указанные в табл. 1.

2. В исходной выборке сформировано 351 245 строк – продажи товаров по каждому SKU (от англ. *Stock Keeping Unit* – идентификатор товарной позиции) за период от 01.10.2013 по 30.09.2016.

Для дальнейшего моделирования разделим исходную выборку на обучающую и тестовую. Все рассматриваемые операции приводятся на языке программирования R для возможного воспроизведения схожих результатов (здесь и далее вставки со скриптом на языке R выделены курсивом):

```
library(dplyr)
training <- filter(dfBeerModeling, Date < "2016-02-01")
testing <- filter(dfBeerModeling, Date >= "2016-02-01")
```

Обучающая и тестовая выборки в противовес классическому принципу разделяются не случайно, а по определенной метке времени, по которой имеют объем 80 и 20 % соответственно.

Также в статье дается определение метрики качества (функционала) рассчитываемых моделей:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2,$$

где y_i – фактические значения целевой переменной; $\hat{y}_i$ – оценка прогнозной величины, n – количество элементов в выборке. MSE является среднеквадратическим отклонением ошибки модели и рассчитывается строго на результатах тестовой выборки.

Модели прогнозирования спроса

Линейная регрессия

В ходе анализа переменных и выявления определенных связей в них [6] было решено провести оценку модели линейной регрессии с нелинейной комбинацией переменных:

$$y = \beta_0 + \left(\alpha_1 y_{t-1} + \alpha_2 y_{t-2} + ... + \alpha_k y_{t-k} \right) \times$$
$$\times \left(\beta_1 x_1 + \beta_2 x_2 + ... + \beta_{m-l} x_{m-l} \right) + ... + \beta_m x_m + \varepsilon,$$

где y – целевая (зависимая) переменная; $y_{t-1}, y_{t-2}, ..., y_{t-k}$ – лагированные значения ряда; $\alpha_1, \alpha_2, ..., \alpha_k$ – авторегрессионные коэффициенты модели; m – количество независимых переменных; $x_1, x_2, ..., x_m$ – незави-

симые переменные и $\beta_1, \beta_2, ..., \beta_m$ – коэффициенты при зависимых переменных, рассчитанные методом наименьших квадратов; β_0 – свободный коэффициент модели; ε – случайная ошибка модели.

Необходимо заранее обозначить, что интерпретация тех или иных коэффициентов выходит за рамки данной статьи. Здесь оценивается качество модели с точки зрения выбранного функционала.

Далее приводится скрипт на R с оценкой созданной модели линейной регрессии:

```
model.base <- lm(newsales ~ (newsalesL1 + newsalesL2 + newsalesL3 + newsalesL5 + newsalesL6 + newsalesL7):(cluster + newprice*isAction + weekday + YOD + numberPositionAction + isActionL1) + tempMean + chqNumber + levelDiscount + holiday*NDoH + country + maker.Descr + weight, data = training)
summary(model.base)
## Часть вывода результата пропущена
## Residual standard error: 3.409 on 276687 degrees of freedom
## Multiple R-squared: 0.8039, Adjusted R-squared: 0.8038
## F-statistic: 6034 on 188 and 276687 DF, p-value: < 2.2e-16
```

Объясненный R^2 равен 0.8038, что говорит о достаточно высокой адаптации модели к обучающимся данным. Тем не менее также необходимо учитывать возможные эффекты переобучения модели. Для оценки переобучения необходимо обратить внимание на коэффициент *RMSE* (*Residual standard error)*, который для обучающейся выборки равен 3.409. Способ, который обычно используют для оценки линейной регрессии на предмет переобученности, называется перекрестная проверка (кросс-валидация) [7].

В данном случае будет проводиться перекрестная проверка с разбиением обучающей выборки на 5 частей. В ходе кросс-валидации каждая из выделенных частей будет выступать как тестовая, остальные – предназначены для обучения модели. Данная процедура будет повторяться для 5 разных вариантов:

```
library(caret)
trControl = trainControl(method = "repeatedcv", number = 5, repeats = 5)
modelLm <- train(newsales ~ ... + weight, data = training, method = "lm", trControl = trControl) ### формула аналогична предыдущей
modelLm
## Linear Regression
## Часть вывода результата пропущена

## RMSE Rsquared
## 3.453482 0.7984404
```

Сравнивая полученную величину *RMSE* с полученной на всей выборке, можно сделать вывод, что наблюдается эффект незначительный переобучения модели. Рассчитанный MSE_{lm} на тестовой выборке равен 9.0779. Для того чтобы сопоставить необходимые результаты, производится расчет $RMSE_{lm} = \sqrt{MSE_{lm}} = 3.0129$. Следовательно, по линейной модели получен достаточно устойчивый результат даже по сравнению с кросс-валидацией.

Для избавления от возможного эффекта переобучения обычно для линейной регрессии используется регуляризация коэффициентов. Далее прорабатывается этот метод, но перед этим будет проведена стандартизация переменных модели, а также будут пересчитаны оценки на тестовой выборке для сравнения качества линейных моделей с нестандартизированными и стандартизированными коэффициентами. Обучающиеся данные обрабатываются с помощью преобразования Бокса – Кокса, затем стандартизуются, перед этим выделяя целевые переменные в отдельные объекты [8]:

```
y.train = training$newsales
y.test = testing$newsales

trans = preProcess(training, method = c("BoxCox", "center", "scale"))
trainingStand <- predict(trans, training)
```

Построим на преобразованных данных линейную модель:

```
model.baseSt <- lm(y.train ~ ... + weight, data = trainingStand)
```

Необходимо проверить значение MSE_{lm} на тестовой выборке для линейной модели с преобразованными данными:

```
testStand <- predict(trans, testing)
base.predSt <- predict(model.baseSt, testStand)

mean((base.predSt - y.test)^2)
## 9.3245
```

Видно, что $MSE_{lm} = 9.3245$ для стандартизированных данных, в то время как $MSE_{lm} = 9.0779$ – для нестандартизированных. Соответственно, следует использовать

результат по линейной модели с исходными данными.

Линейные модели с регуляризацией

Разработанная модель множественной линейной регрессии имеет большую размерность: всего для прогнозирования одной целевой переменной рассчитывается 194 коэффициента. Чтобы снизить размерность модели и устранить эффект переобучения, применяют метод сжатия коэффициентов модели к нулю – регуляризацию. Наиболее распространенными методами, в основе которых лежит процедура регуляризации, являются гребневая и лассо-регрессии [9]. При нахождении коэффициентов для гребневой или лассо-регрессии минимизируют следующий функционал

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{m}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{m}K=$$

$$=RSS+\lambda\sum_{j=1}^{m}K\to\min,$$

$K=\beta_j^2$, если оценивается гребневая регрессия,

$K=\left|\beta_j\right|$, если оценивается лассо-регрессия,

где K – это тип оценки коэффициентов, который выбирается в зависимости от метода; λ – это гиперпараметр метода, с помощью которого выбирается сила сжатия коэффициентов модели.

С помощью реализации десятикратной перекрестной проверки выбирается λ для модели гребневой (*ridge*) регрессии. Перед этим происходит подготовка данных для обработки в функции *glmnet*:

```
library(glmnet)
x.train = model.matrix(newsales ~ ... + weight, data = training)[,-1]
x.train = as.data.frame(x.train)[,-1]

set.seed(123)
CV.ridge = cv.glmnet(x.train, y.train, alpha = 0)
bestlam = CV.ridge$lambda.min
bestlam
## [1] 0.6501895
```

Видно, что гиперпараметр λ=0.6501895, что достаточно близко к 0. Это значит, что метод близок к обычной оценке коэффициентов модели по методу наименьших квадратов, который и был использован ранее для построения линейной регрессии. Тем не менее некоторое сжатие происходит, поэтому далее необходимо рассчитать параметры гребневой регрессии и значение *MSE* на тестовой выборке:

```
model.ridge <- glmnet(x.train, y.train, alpha = 0, lambda = bestlam)
x.test = model.matrix(newsales ~ (newsalesL1 + newsalesL2 + newsalesL3 + newsalesL5 + newsalesL6 + newsalesL7):(cluster + newprice*isAction + weekday + YOD + numberPositionAction + isActionL1) + tempMean + chqNumber + levelDiscount + holiday*NDoH + country + maker.Descr + weight, data = testing)[,-1]
ridge.pred <- predict(model.ridge, x.test)
mean((ridge.pred - y.test)^2)
## [1] 9.005262
```

Видно, что после регуляризации коэффициентов по методу гребневой регрессии качество модели на тестовой выборке увеличилось: MSE_{lm} на тестовой выборке составляет 9.0052 против 9.0779 при простом методе наименьших квадратов.

Далее проводятся аналогичные расчеты для метода лассо-регрессии и выводится оценка *MSE*:

```
mean((lasso.pred - y.test)^2)
## [1] 9.038254
```

Как и в случае с гребневой регрессией по результатам перекрестной проверки значение параметра λ является низким – даже ближе к 0. При этом значение MSE_{lm} = 9.0383, что ниже, чем аналогичное значение, рассчитанное по методу наименьших квадратов, но выше, чем при гребневой регрессии.

Регрессия на основе опорных векторов

Одним из распространенных методов классификации на данный момент является машина опорных векторов. Метод основан на представлении разделимости классов с помощью гиперплоскости, находящейся в $N-1$ пространстве. При этом для расширения пространства предикторов используются так называемые ядерные функции.

Существует также расширение машины опорных векторов для задач регрессии – метод регрессии на основе опорных векторов. Производится расчет регрессионной модели с радиальным ядром. Для ускорения расчетов используется:

- часть данных из обучающей выборки и часть из тестовой. Было принято решение

выбрать данные по всем субботам из исходной выборки, так как целевая переменная в эти дни обладает максимальными средними значениями, дисперсией и стандартным отклонением;

• параллелизация расчетов в R при помощи пакета *parallel*, что облегчает использование кросс-валидации и подбор параметров модели.

Далее приводится расчет для 10 состояний модели с 10 парами случайно отобранных параметров. Рассматривается изменение параметра *C* (регулируется для оптимизации проблемы переобучения) и уровня допускаемой ошибки метода. При выборе параметров используется 10-кратная кросс-валидация:

```
library(e1071)
library(parallel)
training6 <- filter(training, weekday == "суббота")
testing6 <- filter(testing, weekday == "суббота")

set.seed(123) # для воспроизводимости расчетов
epsilon <- sample(seq(0,1,0.1), 10)
cost <- sample(c(0.001, 0.01, 0.1, 1, 10, 50, 100,
250, 500),10, replace = T)
parms = cbind(epsilon, cost)

cl <- makeCluster(getOption("cl.cores", detect-
Cores()-2))
clusterExport(cl,c("training6","parms"))
clusterEvalQ(cl,library(e1071))

tuneResult =
  parApply(cl, parms, 1, function(i) {
    tune(svm, newsales ~ newsalesL1 + newsalesL2 +
newsalesL3 + newsalesL5 + newsalesL6 +
newsalesL7 + cluster + newprice + isAction + week-
day + YOD + numberPositionAction + isActionL1 +
tempMean + chqNumber + levelDiscount + holiday
+ NDoH + country + maker.Descr + weight, data =
training6, ranges = i
    )
  }
)
stopCluster(cl)
```

Результат оптимизации гиперпараметров можно представить в табл. 2 (показатель «Представление» является оценкой ошибки метода; в данном случае – *MSE*).

Видно, что лучшим представлением обладают гиперпараметры с уровнем ошибки от 0,1 до 0,2 и значением *C,* близким от 0 до 50. Далее, на базе ошибки уровня от 0,1 до 0,2 формируется еще один расчет оптимальных гиперпараметров для срезов *C*, равных 10, 25, 50, 100, 250 и 500 (в том числе и для исключения максимальных значений). Расчет аналогичен предыдущему, поэтому приводится только результат поиска гиперпараметров и оценка модели (табл. 3).

Таблица 2

Уровень ошибки (epsilon)	*C* – cost	Представление
0,2	0,100	27,28656
0,9	50,000	27,35101
0,1	0,100	27,44505
0,3	500,000	27,45248
0,7	10,000	27,46631
1,0	100,000	27,47229
0,6	0,001	27,49667
0,0	500,000	27,58455
0,8	0,001	27,59547
0,5	500,000	27,60613

Таблица 3

Уровень ошибки	*C* – cost	Представление
0,12	50	27,22596
0,16	250	27,25693
0,17	10	27,26441
0,19	50	27,27378
0,16	100	27,29423
0,13	25	27,36361
0,13	500	27,3795
0,13	100	27,38807
0,20	25	27,4231
0,14	50	27,44286
0,14	500	27,45856
0,13	250	27,48779
0,10	100	27,49689
0,20	25	27,50112
0,15	25	27,57781

Видно, что высокие значения для *C* приводят к неудовлетворительному результату по общему представлению модели. Далее проводится итоговый расчет для более низких уровней ошибки (от 0,025 до 0,06) и штрафного коэффициента *C*. Расчет проводится аналогично, как и для табл. 4 и 5 с тем отличием, что используется механизм 5-й перекрестной проверки вместо 10-й для ускорения подбора гиперпараметров. Результат представлен в табл. 4.

Таблица 4

Уровень ошибки	*C* – cost	Представление
0,035	5,00	27,60794
0,035	10,00	27,65244
0,050	0,50	27,66187
0,040	20,00	27,66956
0,035	15,00	27,7037
0,060	20,00	27,76763
0,025	5,00	27,77034
0,025	10,00	27,78503
0,040	10,00	27,80793
0,055	1,00	27,85994
0,035	5,00	27,87249
0,060	15,00	27,90332
0,040	1,00	27,93776
0,030	1,00	27,96066
0,025	1,00	27,99153
0,035	15,00	28,06269
0,040	5,00	28,0715
0,050	20,00	28,11136
0,045	1,00	28,15191
0,050	15,00	28,17575

Далее производится оценка *MSE* на обучающейся и тестовой выборках для лучших 3 пар гиперпараметров для табл. 4–6: $C = 0.1$ и epsilon = 0.2, $C = 50$ и epsilon = 0.12, $C = 5$ и epsilon = 0.035. Лучший результат выглядит следующим образом при параметрах *cost = 5, epsilon = 0.035*:

```
mean((tr.pr.svr - training6$newsales)^2)
## [1] 15.2969
mean((ts.pr.svr - testing6$newsales)^2)
## [1] 14.52964
```

Видно, что для данных параметров переобучение модели незначительно. Следовательно, последний вариант гиперпараметров будет приниматься за базовый для расчетов для всей обучающейся выборки.

Ниже оценивается общая модель регрессии на опорных векторах для всей обучающейся выборки и рассчитывается *MSE* для тестовой (для cost = 5, epsilon = 0.035). Используется параллельное вычисление с разбиением выборки на 7 наборов данных в разрезе дней недели:

```
training1 <- filter(training, weekday == "понедельник")
… ### скрипт для training2…6 аналогичен
training7 <- filter(training, weekday == "воскресенье")
train.list = list(training1, training2, training3, training4, training5, training6, training7)
cl <- makeCluster(getOption("cl.cores", detectCores()))
clusterExport(cl,"train.list")
clusterEvalQ(cl,library(e1071))
svm.modelGeneral =
 parLapply(cl, train.list, function(i) {
  svm(newsales ~ newsalesL1 + … + weight, cost = 2.5, epsilon = 0.035, data = i)
 })
stopCluster(cl)

testing$svm.predict=0
testing$svm.predict[testing$weekday == "понедельник"] =
 predict(svm.modelGeneral[[1]], testing[testing$weekday == "понедельник",])
…
testing$svm.predict[testing$weekday == "воскресенье"] =
 predict(svm.modelGeneral[[7]], testing[testing$weekday == "воскресенье",])

mean((testing$svm.predict - testing$newsales)^2)
    ## [1] 9.277606
```

Очевидно, итоговый результат, полученный при оценке SVR, несколько хуже, чем при оценке ридж-регрессии. Следует оценить регрессию на опорных векторах также для более низких значений штрафного коэффициента (при той же заданной ошибке) (табл. 5).

Таблица 5

C – cost	*MSE* на тестовой
5,00	9,277606
2,50	9,169015
1,00	9,248483

Как видно из таблицы, лучшей парой гиперпараметров для оценки регрессии на опорных векторах на всей обучающейся выборке является $C = 2.5$ и epsilon = 0.035. Построенная модель обладает прогностическим качеством, несколько уступающим моделям линейной регрессии (МНК и с регуляризацией). Тем не менее модель не обладает столь сложным функционалом по сравнению с моделями линейной регрессии, позволяя вместо этого использовать ядерные функции разнообразной формы. Основным минусом модели является использование метода случайного поиска гиперпараметров модели, продолжительность которого возрастает на больших объемах выборки.

Метод «случайный лес»

Метод «случайный лес» (random forest) основан на простом предположении о том, что некий средний результат по значительному количеству простых моделей дает лучшую аппроксимацию целевой переменной, чем каждая модель в отдельности. При этом в основе выстраиваемых моделей находится метод построения дерева решений, широко используемый в интеллектуальном анализе данных [10].

Для эффективной работы алгоритма также важны гиперпараметры определенного вида:

- количество строящихся деревьев решений в составе случайного леса;
- количество случайно отобранных признаков для построения каждого дерева в составе случайного леса – *M*.

Для поиска оптимального значения количества деревьев обычно анализируют ошибку *out-of-bag (OBB)*, которая, по сути, является среднеквадратической ошибкой деревьев на той доле от обучающей выборки, которая не была задействована при их формировании [11]. Было принято решение об использовании параметра в 700 деревьев. Подобное решение вполне понятно из-за большого размера используемых деревьев.

Выбор параметра *M* определяется с помощью алгоритма *tuneRF,* заложенного в пакете *randomForest.* Алгоритм является итерационным и также основан на минимизации ошибки OBB. Стартовое значением для *M* (*mtry*) определяется как деление количества признаков на 3, затем алгоритм корректирует значение *mtry*, двигаясь вверх или вниз в соответствии с указанным шагом. Алгоритм был реализован в 2 вариантах для поиска достоверного результата:

```
tuneI <-
tuneRF(x = training6[,-c(1,2,4,17,23)], y = training6$newsales,
stepFactor=1.5, ntreeTry = 200, trace = F)
tuneII <-
tuneRF(x = training6[,-c(1,2,4,17,23)], y = training6$newsales,
stepFactor=2.5, ntreeTry = 200, trace = F)
```

Исходя из результата работы алгоритмов минимальным OBB обладает случайный лес со значение $M = 5$.

Для выбранных параметров реализуется алгоритм случайного леса при использовании пакета *randomForest* и параллельного вычисления в пакете *parallel*:

```
train.list = list(training1, training2, training3, training4, training5, training6, training7)

cl <- makeCluster(getOption("cl.cores", detectCores() - 1))
clusterExport(cl,"train.list")
clusterEvalQ(cl,library(randomForest))

mod.forest <-
      parLapply(cl, train.list, function(i) {
      randomForest(newsales ~ newsalesL1 + ... +
      weight, ntree = 700, mtry = 5, data = i)
})
stopCluster(cl)

testing$rf.predict=0
testing$rf.predict[testing$weekday == "понедельник"] =
 predict(mod.forest[[1]], testing[testing$weekday == "понедельник",])
...
testing$rf.predict[testing$weekday == "воскресенье"] =
 predict(mod.forest[[7]], testing[testing$weekday == "воскресенье",])

mean((testing$rf.predict - testing$newsales)^2)
# 8.96718
```

Значение *MSE* на тестовой выборке равно 8.96718, что является минимальным значением в сравнении со всеми использованными алгоритмами.

Корреляция результатов и выводы по моделированию

По результатам моделирования прогноза покупательского спроса на товар выводятся следующие характеристики *MSE* на тестовой выборке:

Таблица 6

Метод прогнозирования	*MSE* на тестовой выборке	*RMSE* на тестовой выборке
Линейная регрессия	9.077873	3.012951
Гребневая регрессия (ridge)	9.005262	3.000877
Лассо-регрессия	9.038254	3.006369
Регрессия на опорных векторах	9.169015	3.028038
Случайный лес	8.96718	2.994525

Очевидно, что метод случайного леса имеет преимущество перед другими мето-

дами. Тем не менее нельзя сказать, что стоит использовать только его при построении модели прогнозирования. Имеет смысл скомбинировать использованные подходы и выйти на более качественный результат. Для проверки этого предположения следует вывести матрицу корреляций результатов прогнозирования на тестовой выборке (табл. 7).

Таблица 7

Матрица	Спрос (Y)	Прогноз				
		RF	SVR	LM	Ridge	Lasso
Спрос	1.0000000	0.8792364	0.8783794	0.8771240	0.8781998	0.8776937
RF	0.8792364	1.0000000	0.9843456	0.9809727	0.9843101	0.9821861
SVR	0.8783794	0.9843456	1.0000000	0.9725575	0.9750715	0.9737348
LM	0.8771240	0.9809727	0.9725575	1.0000000	0.9973992	0.9995951
Ridge	0.8781998	0.9843101	0.9750715	0.9973992	1.0000000	0.9985077
Lasso	0.8776937	0.9821861	0.9737348	0.9995951	0.9985077	1.0000000

Видна высокая степень связи между результатами разных методов. Также очевидно, что методы, основанные на линейной регрессии, сильно коррелируют друг с другом. Поэтому для дальнейших вычислительных экспериментов используется только лучший из примененных линейных методов – гребневая регрессия.

Далее следует провести простое усреднение результатов прогнозирования по 3 методам: гребневой регрессии, регрессии на опорных векторах и случайного леса:

$$\hat{y}_c = \frac{\hat{y}_1 + \hat{y}_2 + \hat{y}_3}{3},$$

где $\hat{y}_c$ – скомбинированная оценка прогноза.

По рассчитанной оценке рассчитаем стандартные метрики: $MSE = 8.634084$ и $RMSE = 2.938381$. Метрики показывают лучший результат даже по сравнению с отдельным методом случайного леса на этой же тестовой выборке. Следовательно, делается вывод, что возможны дальнейшие преобразования с рассчитанным результатом по моделям прогнозирования с применением более эффективных техник объединения результатов.

Результаты в приведенном исследовании позволяют сделать следующие выводы:

1. Примененные методы прогнозирования позволяют выявлять нелинейность в отношениях между переменными и их влиянием на целевой признак без применения специального экономического анализа. Это позволяет экономить время и получать лучший результат, но только в том случае, если необходима высокая точность прогноза и необязательна интерпретация модели.

2. Разные методы прогнозирования – линейная регрессия, регрессия на опорных векторах, случайный лес – дают схожий, удовлетворительный, но не одинаковый результат при реализации их обучения. Это позволяет сделать вывод о возможности их использования как «кирпичиков» в единой системе прогнозирования.

3. Дальнейшее улучшение качества прогнозирования возможно при создании определенного комитета (ансамбля) примененных моделей прогнозирования. Наиболее распространенными методами по созданию ансамблей являются бустинг (*boosting*), бэггинг (сокр. от *bootstrap aggregating*) и стэкинг (*stacking*) [12, 13]. Применимость данных техник и результат от них является дальнейшим предметом исследования научной работы.

Библиографические ссылки

1. *Светуньков С. Г., Светуньков И. С.* Методы социально-экономического прогнозирования: учебник для вузов. Т. I. СПб. : Изд-во СПбГУЭФ, 2009. 147 с.

2. *Светуньков С. Г., Светуньков И. С.* Методы социально-экономического прогнозирования: учебник для вузов. Т. II. СПб. : Изд-во СПбГУЭФ, 2010. 103 с.

3. *Адамадзиев К. Р., Касимова Т. М.* Применение панельного метода при исследовании эффективности производства зерна в Республике Дагестан // Фундаментальные исследования. 2012. № 6-3. С. 699–701.

4. *Стрижов В. В., Крымова Е. А.* Методы выбора регрессионных моделей. М. : ВЦ РАН, 2010. 60 с.

5. *Kuhn M., Johnson K.* Applied Predictive Modeling. Springer, 2013. 600 p. 203 illus., 153 illus. in color.

6. *Пивкин К. С.* Корреляционный анализ факторов влияния на покупательский спрос розничного магазина как этап формирования модели прогнозирования и управления запасами // Вестник УдГУ. Серия: Экономика. 2016. № 3. С. 40–50.

7. *Огурцов А. В.* Настройка гиперпараметров и оценка качества прогностической модели: пример с использованием языка R и пакета caret // Математика, статистика и информационные технологии в экономике, управлении и образовании : сборник трудов V Международной научно-практической конференции. Тверь. 2016. С. 73–78.

8. *Мастицкий С. Э.* Подготовка данных для создания предсказательных моделей: трансформация предикторов. Блог «R: Анализ и визуализация данных» [Электронный ресурс]. URL: http://r-analytics.blogspot.ru/2015/07/blog-post_19.html#.WFZ55fmLTIV.

9. *Джеймс Г., Уиттон Д., Хасти Т., Тибширани Р.* Введение в статистическое обучение с примерами на языке R / пер. с англ. С. Э. Мастицкий. М. : ДМК-Пресс, 2016. 450 с.

10. *Паклин Н. Б., Орешков В. И.* Бизнес-аналитика: от данных к знаниям (+CD) : учеб. пособие. 2-е изд., испр. СПб. : Питер, 2013. 704 с. : ил.

11. *Джеймс Г., Уиттон Д., Хасти Т., Тибширани Р.* Введение в статистическое обучение с примерами на языке R.

12. *Паклин Н. Б., Орешков В. И.* Бизнес-аналитика: от данных к знаниям.

13. Business Data Analytics: Ансамбли моделей. [Электронный ресурс]. URL: http:// businessdataanalytics.ru/ModelEnsembles.htm.

References

1. Svetun'kov S. G., Svetun'kov I. S. (2009). *Metody sotsial'no-ekonomicheskogo prognozirovaniya* [Methods of socio-economic forecasting]. Vol. 1. St. Petersburg: Izd-vo SPbGUEF, 147 p. (in Russ.).

2. Svetun'kov S. G., Svetun'kov I. S. (2010). *Metody sotsial'no-ekonomicheskogo prognozirovaniya* [Methods of socio-economic forecasting]. Vol. 2. St. Petersburg: Izd-vo SPbGUEF, 147 p. (in Russ.).

3. Adamadziev K. R., Kasimova T. M. (2012). *Fundamental'nye issledovaniya* [Basic research], no. 6-3. pp. 699-701 (in Russ.).

4. Strizhov V. V., Krymova E. A. (2010). *Metody vybora regressionnykh modelei* [Methods for selecting regression models]. Moscow: VTs RAN, 60 p. (in Russ.).

5. *Kuhn M., Johnson K.* Applied Predictive Modeling. Springer, 2013. 600 p. 203 illus., 153 illus. in color.

6. Pivkin K. S. (2016). *Vestnik UdGU. Seriya: Ekonomika* [Bulletin of UdSU. Series: The Economy], no. 3, pp. 40-50 (in Russ.).

7. Ogurtsov A. V. (2016). *Nastroika giperparametrov i otsenka kachestva prognosticheskoi modeli: primer s ispol'zovaniem yazyka R i paketa caret* [Adjusting hyperparameters and assessing the quality of the predictive model: an example using the R language and the caret package]. Proceedings of *Matematika, statistika i informatsionnye tekhnologii v ekonomike, upravlenii i obrazovanii*, Tver, pp. 73-78 (in Russ.).

8. Mastitskii S. E. *Podgotovka dannykh dlya sozdaniya predskazatel'nykh modelei: transformatsiya prediktorov. Blog «R: Analiz i vizualizatsiya dannykh»* [Prepare data for creating predictive models: transform predictors. Blog "R: Data analysis and visualization"]. Available at http://r-analytics.blogspot.ru/2015/07/blog-post_19.html#.WFZ55fmLTIV (in Russ.).

9. Dzheims G., Uitton D., Khasti T., Tibshirani R. (2016). *Vvedenie v statisticheskoe obuchenie s primerami na yazyke R* [Introduction to statistical learning with examples in R]. Moscow: DMK-Press, 450 p. (in Russ.).

10. Paklin N. B., Oreshkov V. I. (2013). *Biznes-analitika: ot dannykh k znaniyam* [Business Intelligence: from data to knowledge]. St. Petersburg: Piter, 704 p. (in Russ.).

11. Dzheims G., Uitton D., Khasti T., Tibshirani R. (2016). *Vvedenie v statisticheskoe obuchenie s primerami na yazyke R* [Introduction to statistical learning with examples in R].

12. Paklin N. B., Oreshkov V. I. (2013). *Biznes-analitika: ot dannykh k znaniyam* [Business Intelligence: from data to knowledge].

13. Business Data Analytics: Ансамбли моделей. [Электронный ресурс]. URL: http://business-dataanalytics.ru/ ModelEnsembles.htm.

* * *

K. S. Pivkin, Post-graduate, Udmurt State University

Realization of Regression Methods of Demand Forecasting Using the R Language

Regression analysis is considered as a key method for forecasting the magnitude of demand of goods. The list of methods that are the most effective for calculating the forecast estimate is presented: linear regression with regularization, regression on the basis of support vectors, random forest method. Necessary calculations are implemented in the programming language R, using both the basic functional and additional packages, which make it possible to use the methods in question. As input data, the store performance and product characteristics are used. The metric of the quality of the result of the operation of the algorithms is determined, i.e., the mean squared error. The sample of data is divided into training and test data, the results for each of the above algorithms are calculated in sequence. Conclusions are drawn that for the sample under consideration, the random forest algorithm yields the best result. The degree of correlation between forecasts for different algorithms is derived, on the basis of which an assumption is made about possible joint use of forecasts. Proceeding from this, the simplest combination of algorithms is constructed, i.e., the arithmetic mean. This ensemble of algorithms turned out to be better than all the considered methods of forecasting, taken separately. A plan for further research on the creation of a committee of algorithms based on methods of bugging, boosting or stacking is determined.

Keywords: demand of good, regression analysis, R language, support vector machine, random forest, regularization, cross-validation, ensemble of algorithms.

Получено: 01.12.17