Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РтФ

Школа бакалавриата

Оценка

Дата защиты

АНАЛИЗ ЗАВИСИМОСТИ СТОИМОСТИ АКЦИЙ КОМПАНИИ ОТ ДАТЫ

Пояснительная записка к проекту по модулю «Методы анализа Big Data»

090302 0000 713/013 ПЗ

Подпись Дата

Преподаватель С.Г. Мирвода

Студенты Н.Д. Матросова

А.Л. Озорнин

Группы РИ-450004, РИ-450005

**Содержание**

[Введение 3](#_Toc535467964)

[1 Краткое описание набора данных 4](#_Toc535467965)

[2 Краткий анализ данных 5](#_Toc535467966)

[3 Выбор и построение модели 7](#_Toc535467967)

[4 Проверка модели 10](#_Toc535467968)

[5 Разработка приложения 12](#_Toc535467969)

[Заключение 13](#_Toc535467970)

[Библиографический список 14](#_Toc535467971)

[ПРИЛОЖЕНИЕ А – ИСХОДНЫЙ КОД R 15](#_Toc535467972)

[ПРИЛОЖЕНИЕ Б – ИСХОДНЫЙ КОД RMD 17](#_Toc535467973)

# Введение

Увеличение глобального количества хранимых данных и вычислительных мощностей привело к развитию направления анализа больших данных. Данное направление позволяет обрабатывать большие наборы данных, находить зависимости, составлять различные модели, осуществлять прогнозы.

В данной работе необходимо выполнить анализ набора данных, подобрать подходящую модель, осуществить прогноз, проверить статистическую значимость модели, а также качество прогнозов.

В качестве набора данных для работы были выбраны данные фондовых индексов Dow Jones для крупнейшей нефтяной компании ExxonMobil (Exxon Mobile Corporation, XOM). Цель работы заключается в исследовании зависимости стоимости акций данной компании от времени.

# Краткое описание набора данных

Исследование экономических данных является одной из наиболее перспективных направлений Big Data, так как позволяет извлечь из исследований реальную прибыль.

Представленный набор данных содержит данные о котировках в период с 2014 по 2018 годы.

Таблица данных содержит следующие колонки:

* Дата (date)
* Стоимость акций при открытии биржи (open)
* Самая низкая стоимость за сутки (low)
* Самая высокая стоимость за сутки (high)
* Стоимость акций при закрытии биржи (close)
* Суточное изменение (change)
* Суточное изменение в процентах (changePercent)
* Изменение с течением времени (changeOverTime)

Биржи не работают в выходные дни, поэтому соответствующих записей нет в таблице.

# Краткий анализ данных

Данные хранились в файле CSV (comma separated values). С помощью языка R данный файл был прочитан, была составлена таблица данных (dataframe). Текстовые значения даты были преобразованы в стандартный для языка R тип Date. Была составлена сводка по набору данных.

Далее для представления стоимости акций в данный день будет использоваться стоимость при закрытии биржи (close). Стоимость указана в долларах США (USD).

Ниже представлены основные параметры стоимости:

|  |  |
| --- | --- |
| Среднее значение | 77.4452 |
| Медиана | 77.9298 |
| Дисперсия | 24.51068 |
| Стандартное отклонение | 4.950826 |
| Минимальное значение | 60.3055 |
| Максимальное значение | 88.0921 |

Был построен график стоимости акций:

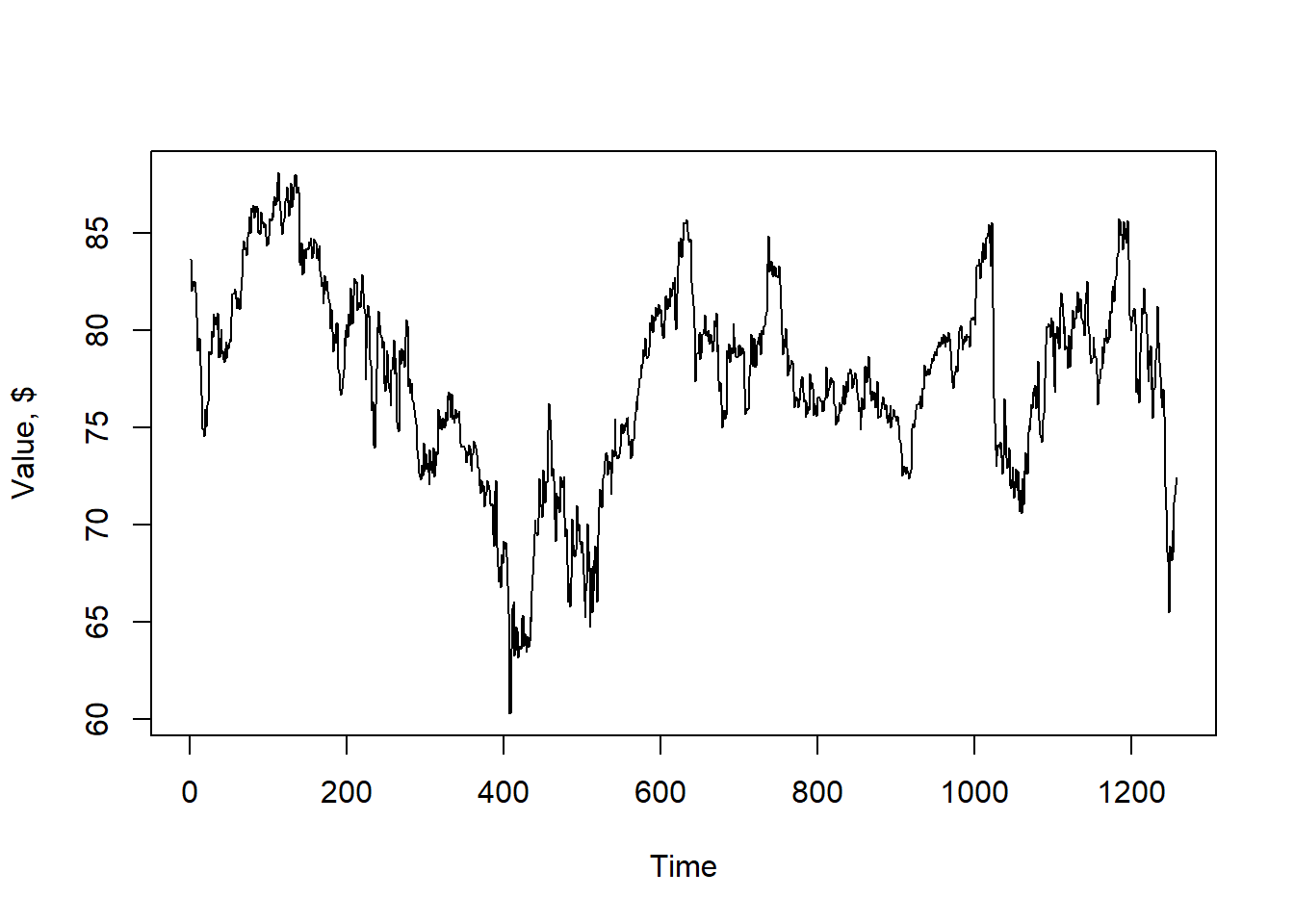


Рисунок 1 – зависимость стоимости от даты

По графику тяжело судить о наличии или отсутствии зависимости стоимости акций от даты.

# Выбор и построение модели

Перед построением модели было принято решение исследовать возможность избавления от выбросов. В случае выбранного набора данных нельзя назвать выбросами слишком большие или слишком малые значения, так как каждое значение зависит от предыдущих. Такие значения могут скорее являться следствием некоторой тенденции. Однако обработка выбросов для данного набора по-прежнему возможна.

В ходе исследования было принято решение считать выбросами такие значения, которые либо больше предыдущего и последующего, либо меньше, причем суммарная относительная разница больше 5%.

**for**(i **in** 2:datalen-1) {

hop <- abs(2 \* data$close[i] - data$close[i - 1] - data$close[i + 1]) /

data$close[i]

data$hop[i] <- hop[1]

}

data <- data[data$hop < 0.05,]

datalen = length(data$close)

В результате избавления от таких выбросов качество предсказания возросло на 2%.

Набор данных был разделен на тренировочную и тестовую выборки. Тренировочная выборка содержит данные с 2014 по 2017 год, а тестовая – данные 2018 года (в выпущенном приложении все данные используются как тренировочная выборка). Выборки имеют частоту временного ряда 12 (помесячно).

Ниже представлен сезонный график для тренировочной выборки:

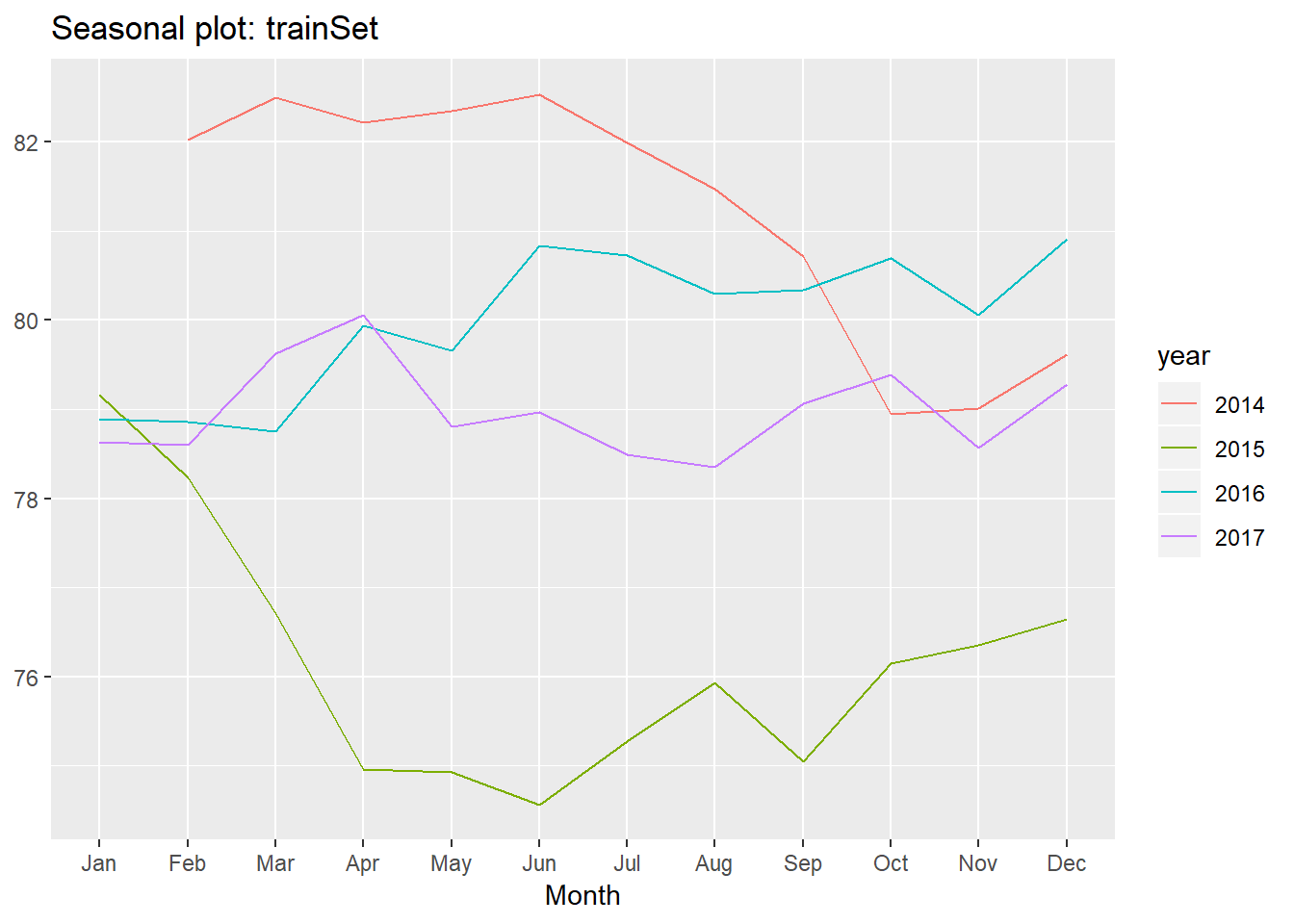


Рисунок 2 – сезонный график для тренировочной выборки

Для выбранного набора данных была выбрана модель анализа временных рядов ARIMA (autoregressive integrated moving average).

monthArima <- auto.arima(trainSet)

monthForecast <- forecast(monthArima, h = 12)

monthForecast

С помощью данной модели было выполнено прогнозирование стоимости акций на ближайшие 12 месяцев.

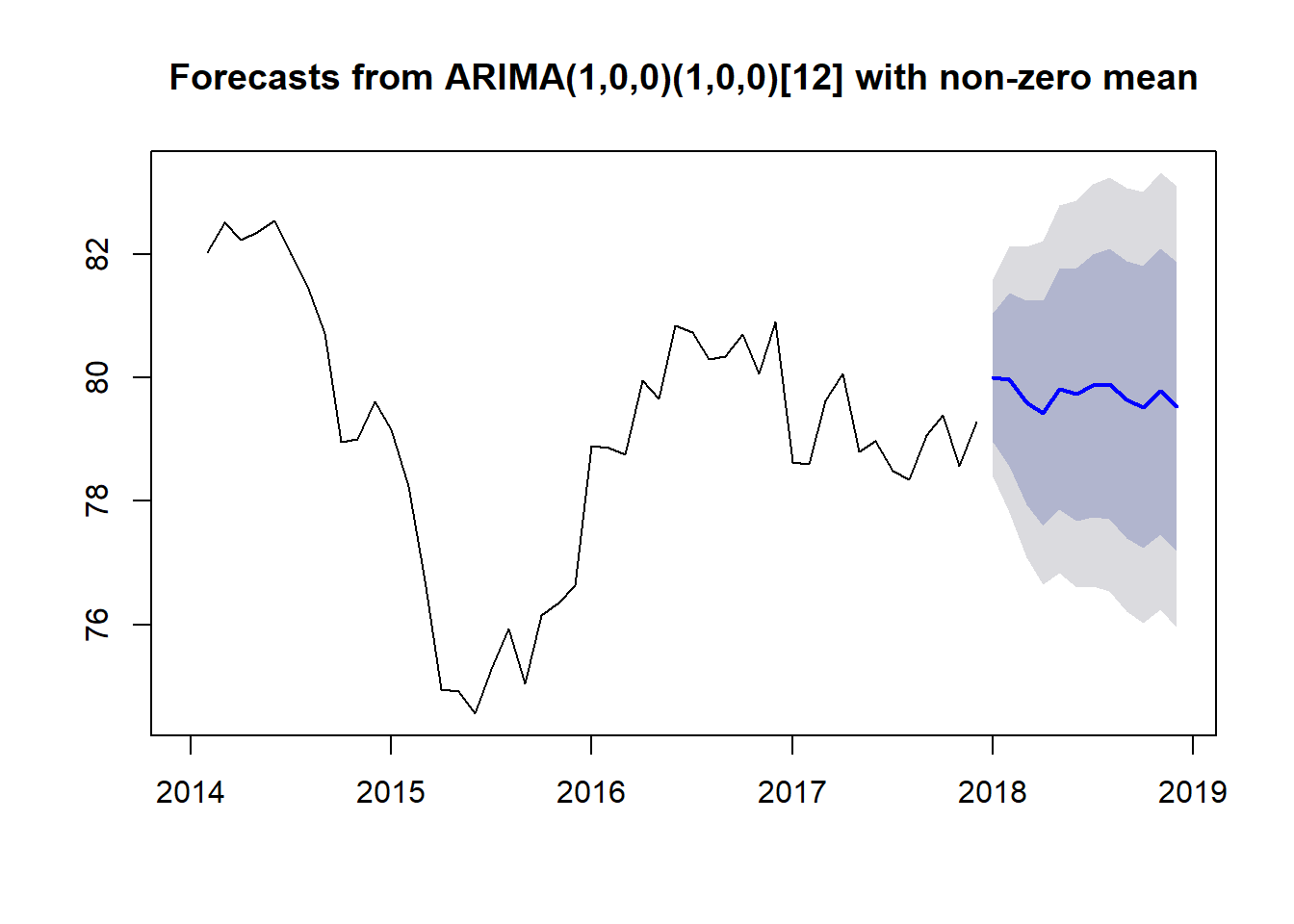


Рисунок 3 – прогноз с помощью ARIMA

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

## Jan 2018 79.99435 78.95099 81.03771 78.39867 81.59003

## Feb 2018 79.96967 78.56268 81.37666 77.81787 82.12147

## Mar 2018 79.59783 77.95195 81.24370 77.08068 82.11498

## Apr 2018 79.42782 77.60963 81.24602 76.64714 82.20851

## May 2018 79.81267 77.86474 81.76059 76.83357 82.79176

## Jun 2018 79.73355 77.68555 81.78155 76.60141 82.86570

## Jul 2018 79.86925 77.74284 81.99566 76.61718 83.12131

## Aug 2018 79.89659 77.70809 82.08509 76.54957 83.24361

## Sep 2018 79.64425 77.40622 81.88229 76.22147 83.06703

## Oct 2018 79.52054 77.24276 81.79832 76.03698 83.00411

## Nov 2018 79.77656 77.46676 82.08636 76.24402 83.30909

## Dec 2018 79.52882 77.19313 81.86450 75.95670 83.10094

В результате прогнозирования были успешно получены предполагаемые значения для каждого месяца 2019 года.

# Проверка модели

Для проверки статистической значимости модели был использован Ljung-Box test:

##

## Ljung-Box test

##

## data: Residuals from ARIMA(1,0,0)(1,0,0)[12] with non-zero mean

## Q\* = 15.2, df = 6.6, p-value = 0.02685

##

## Model df: 3. Total lags used: 9.6

##

## Box-Ljung test

##

## data: monthArima$residuals

## X-squared = 0.28971, df = 3, p-value = 0.962

Неравенство Q\* > X-squared означает, что модель является статистически значимой. Таким образом, данная модель действительно может быть использована в рамках исследования выбранной зависимости.

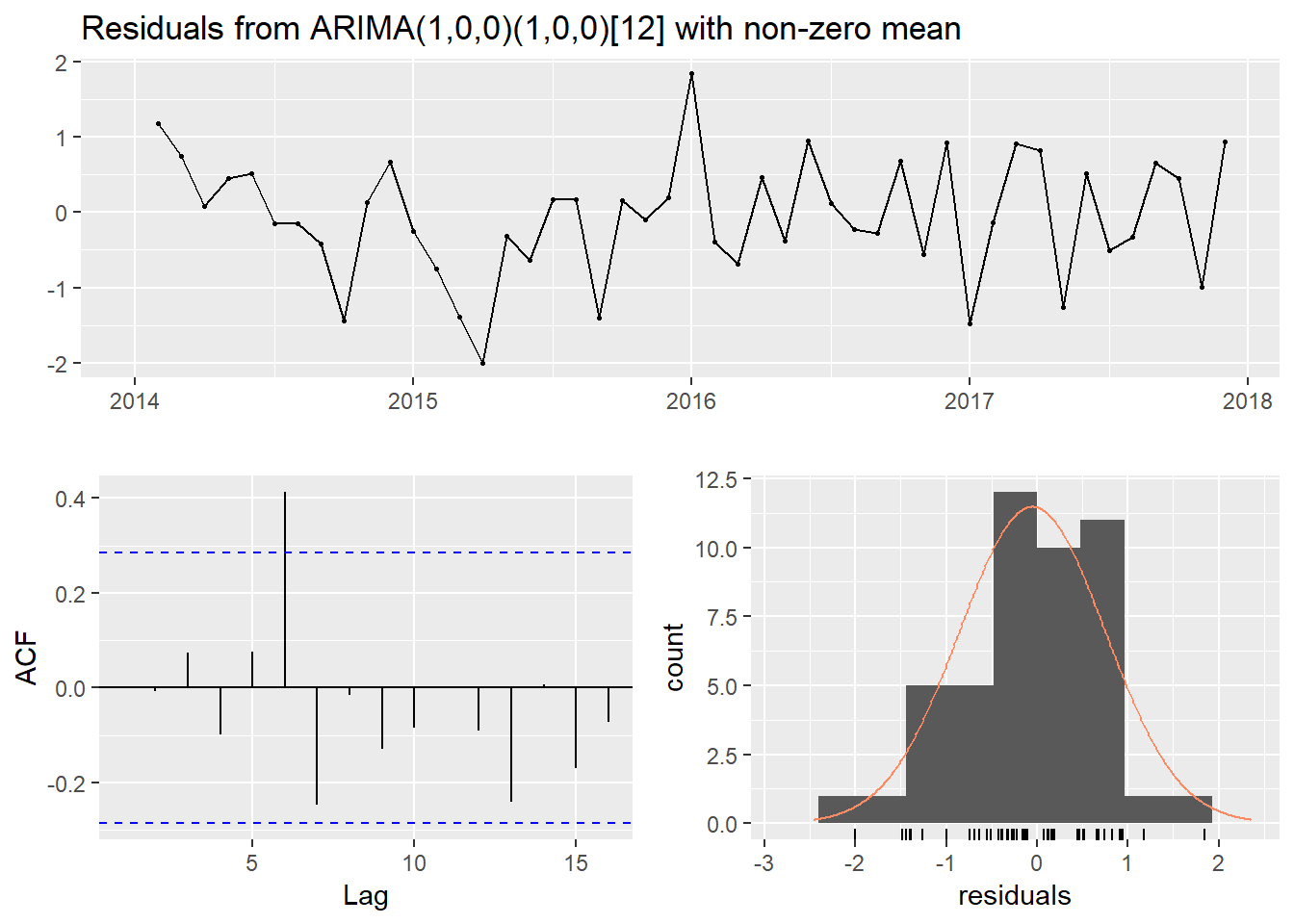


Рисунок 4 – остаточная погрешность

Для оценки качества прогнозирования было проведено сравнение результатов прогнозирования на 2018 год с данными тестовой выборки:

accuracy(monthForecast, testSet)

## ME RMSE MAE MPE MAPE

## Training set -0.05242459 0.7966288 0.6370715 -0.07877675 0.8089357

## Test set 1.51088697 1.9968411 1.7542691 1.83409754 2.1422403

## MASE ACF1 Theil's U

## Training set 0.1868956 0.002760237 NA

## Test set 0.5146442 0.795401922 2.776241

Качество прогнозирования по месяцам достаточно высокое, так среднее отклонение в процентах (MPE, mean percentage error) равно 1.83%

# Разработка приложения

В ходе работы было принято решение о разработке веб-приложения на языке JavaScript. Back-end разработан с помощью среды Node.js, front-end – с помощью фреймворка Vue.js. Общение между клиентом и сервером реализовано с помощью протокола HTTP.

Код, написанный на языке R запускается с помощью RScript, дата для прогноза передается как аргумент запуска. Вывод результата осуществляется с помощью стандартного потока вывода процесса (stdout), затем вывод обрабатывается и отправляется клиенту.

Для возможности прогнозирования по конкретной дате спрогнозированные данные для данного месяца умножаются на коэффициент, численно равный отношению средней стоимости акций для данного числа месяца к средней стоимости акций за весь период.

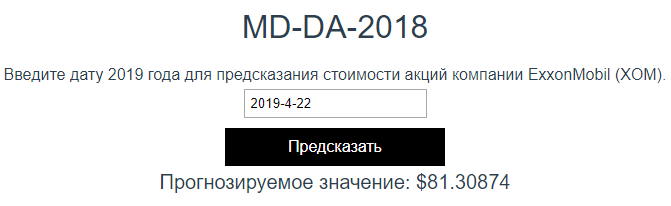


Рисунок 5 – интерфейс веб-приложения

# Заключение

В ходе выполнения данной работы была построена модель, описывающая зависимость стоимости акций компании от даты. Для улучшения качества модели было произведено избавление от выбросов. Данная модель была проверена, она является статистически верной. Данные прогнозирования являются удовлетворительными.

На языке JavaScript с использованием среды Node.js и фреймворка Vue.js было разработано веб-приложение, позволяющее прогнозировать стоимость акций компании для любой даты 2019 года.

В ходе работы были получены практические навыки обработки больших данных, составления моделей данных, прогнозирования и оценки результатов.

Исходный код на языке R представлен в приложении.

# Библиографический список

1. С.А. Измалкова, Т.А. Головина. Использование гловальных технологий “Big Data” в управлении экономическими системами, 2015. – 158 с.
2. Введение в многомерный анализ [электронный ресурс]. Доступ по ссылке – <https://habr.com/post/126810/> . Последняя дата обращения: 17.11.18.
3. G.P. Zhang. Time series forecasting using a hybrid ARIMA and neural network model, 2003. Neurocomputing, Elsevir, v50, p 159-175.

# ПРИЛОЖЕНИЕ А – ИСХОДНЫЙ КОД R

library(forecast)

library(spatstat)

library(lubridate)

spawnArgs <- commandArgs(trailingOnly=TRUE)

if (length(spawnArgs) < 2) {

data <- read.csv("./data/XOM.csv")

fcdate = "2019-02-12"

} else {

data <- read.csv(spawnArgs[1])

fcdate <- spawnArgs[2]

}

fcdate = as.Date(fcdate)

if (year(fcdate) != 2019) {

print("!Ошибка даты")

}

datalen = length(data$close)

data$date <- as.Date(data$date)

data$day <- day(data$date)

data$hop <- numeric(datalen)

closeMean <- mean(data$close)

dayNum <- numeric(31)

dayClose <- numeric(31)

for(i in 2:datalen-1) {

hop <- abs(2 \* data$close[i] - data$close[i - 1] - data$close[i + 1]) / data$close[i]

data$hop[i] <- hop[1]

}

data <- data[data$hop < 0.05,]

datalen = length(data$close)

for (i in 1:datalen) {

dayNum[data$day[i]] <- dayNum[data$day[i]] + 1

dayClose[data$day[i]] <- dayClose[data$day[i]] + data$close[i]

}

dayClose <- dayClose / dayNum

dayCoefficient <- dayClose / closeMean

trainSet <- ts(data$close, start = c(2014, 1, 10), end = c(2018, 12, 31), frequency = 12)

monthArima <- auto.arima(trainSet)

monthForecast <- forecast(monthArima, h = 12)

forecasted <- as.numeric(monthForecast$mean)

res <- forecasted[month(fcdate)]

print(res \* dayCoefficient[day(fcdate)])

# ПРИЛОЖЕНИЕ Б – ИСХОДНЫЙ КОД RMD

---

title: "Course project"

author: "Alexander Ozornin, Natasha Matrosova"

date: "January 16, 2019"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

library(ggplot2)

library(forecast)

library(spatstat)

library(lubridate)

```

## Исследование зависимости котировок XOM от времени

### Курсовая работа по модулю "Методы анализа Big Data"

#### Загрузка данных

```{r}

data <- read.csv("./data/XOM.csv")

```

```{r}

datalen = length(data$close)

```

#### Преобразование даты из формата YYYY-MM-DD в объект даты, поучение номера дня в месяце

```{r}

data$date <- as.Date(data$date)

data$day <- day(data$date)

```

#### Сводка по набору данных

```{r}

summary(data)

```

#### График зависимости котировок от времени

```{r}

plot(data$close, xlab = "Time", ylab = "Value, $", type = "l")

```

#### Среднее значение котировок за все время

```{r}

closeMean <- mean(data$close)

print(closeMean)

```

#### Медиана

```{r}

median(data$close)

```

#### Дисперсия

```{r}

var(data$close)

```

#### Стандартное отклонение

```{r}

sd(data$close)

```

#### Минимальное значение

```{r}

min(data$close)

```

#### Максимальное значение

```{r}

max(data$close)

```

#### Избавление от выбросов

```{r}

for(i in 2:datalen-1) {

hop <- abs(2 \* data$close[i] - data$close[i - 1] - data$close[i + 1]) / data$close[i]

data$hop[i] <- hop[1]

}

data <- data[data$hop < 0.05,]

datalen = length(data$close)

```

#### Получение средних данных по дням месяца и рассчет коэффициентов

```{r}

dayNum <- numeric(31)

dayClose <- numeric(31)

for (i in 1:datalen) {

dayNum[data$day[i]] <- dayNum[data$day[i]] + 1

dayClose[data$day[i]] <- dayClose[data$day[i]] + data$close[i]

}

dayClose <- dayClose / dayNum

dayCoefficient <- dayClose / closeMean

```

#### Создание тренировочной и тестовой выборок. Тренировочная выборка представляет из себя данные до 2018 года, а тестовая - за 2018 год. Выборки имеют частоту "12" - помесячно

```{r}

trainSet <- ts(data$close, start = c(2014, 1, 10), end = c(2017, 12, 31), frequency = 12)

testSet <- ts(data$close, start = c(2018, 1, 1), end = c(2018, 12, 31), frequency = 12)

```

Для будущего предсказания значений все данные, включая 2018 год, будут использоваться как тренировочная выборка.

#### График по сезонам

```{r}

ggseasonplot(trainSet)

```

#### Прогноз значений для ближайших 12 месяцев с помощью ARIMA

```{r}

monthArima <- auto.arima(trainSet)

monthForecast <- forecast(monthArima, h = 12)

monthForecast

```

#### График прогнозов

```{r}

plot(monthForecast)

```

#### Проверка статистической значимости модели

```{r}

checkresiduals(monthArima)

Box.test(monthArima$residuals, lag=3, type='Ljung-Box', fitdf=0)

```

В данном случае модель является статистичеки значимой, так как Q\* больше, чем X-squared.

#### Провека качества модели и прогнозов

```{r}

accuracy(monthForecast, testSet)

```

Качество модели достаточно высокое, в прогнозе средняя ошибка в процентах (MPE) имеет низкое значение.