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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА к проекту по модулю «Методы анализа Big Data» по теме: ЗАВИСИМОСТЬ ПОПУЛЯРНОСТИ МУЗЫКАЛЬНОГО АЛЬБОМА ОТ ДАТЫ ЕГО ВЫПУСКА

Подпись Дата

Преподаватель С.Г. Мирвода

Студенты И.А. Зарудный

Е.И. Манин

Группа РИ-450005

**Содержание**

[Введение 3](#_Toc535427417)

[1 Описание набора данных 4](#_Toc535427418)

[2 Обработка данных 6](#_Toc535427419)

[3 Построение моделей 8](#_Toc535427420)

[3.1 Построение первичной модели с выбросами 9](#_Toc535427421)

[3.2 Проецирование продаж на логарифмическую шкалу 11](#_Toc535427422)

[3.3 Фильтрация набора данных и построение модели 13](#_Toc535427423)

[3.4 Проецирование продаж на гармоническую функцию 15](#_Toc535427424)

[4 Разработка сервиса 18](#_Toc535427425)

[Заключение 20](#_Toc535427426)

[Библиографический список 21](#_Toc535427427)

[ПРИЛОЖЕНИЕ А 22](#_Toc535427428)

# Введение

С развитием современных технологий появились новые возможности для анализа огромного количества данных. Огромные объёмы данных обрабатываются для того, чтобы человек мог получить конкретные и нужные ему результаты для их дальнейшего эффективного применения.

В ходе работы над нашим проектом предметом для исследования была выбрана сфера музыкальной индустрии. Наверняка, многие люди замечали, что в один и тот же день может выйти сразу несколько музыкальных альбомов различных исполнителей, жанры которых могут быть диаметрально противоположными. Данная ситуация заинтересовала нас и было принято решение проверить, существует ли зависимость между популярностью альбома, количеством его продаж и непосредственно датой его выпуска.

Поставленная цель достигается при выполнении следующих задач:

1. Выбор и обработка подходящего набора данных.
2. Анализ полученных данных
3. Фильтрация набора данных.
4. Построение моделей данных.
5. Разработка приложения для работы с созданной моделью.

# Описание набора данных

Музыкальная индустрия – часть экономики и общественной жизни, объединяющая компании и граждан, зарабатывающих деньги посредством создания и продажи музыкальных произведений. Такая индустрия является составной частью более обширного сектора экономики – индустрии развлечений [1].

В музыкальном бизнесе задействовано большое число людей и организаций: музыканты, создающие и исполняющие музыкальные произведения; компании и специалисты, занимающиеся звукозаписью и продажей музыки, в том числе продюсеры, студии звукозаписи, звукорежиссёры, лейблы, музыкальные магазины и коллективные организации по управлению правами; организаторы гастролей; теле- и радиосети, работающие в музыкальном формате; музыкальные журналисты и критики; производители музыкальных инструментов и многие другие.

В качестве своеобразного показателя успешности релиза можно расценивать статус, присвоенный после сертификации альбома. Одной из наиболее авторитетных признана сертификация *RIAA* (*Recording Industry Association of America*).

Сертификация *RIAA* – процедура сертификации объёма продаж звукозаписей на территории США, на соответствие определённому статусу. Осуществляется Американской ассоциацией звукозаписывающих компаний. Появилась в 1958 как попытка стандартизировать методики измерения объёма продаж звукозаписей. Сначала существовал только статус «Золотая награда», присуждаемый за продажу более 500’000 экземпляров записи. В 1976 появился платиновый статус (продажа более 1’000’000 экземпляров). В 1984 появился мультиплатиновый статус (более 2’000’000 экземпляров), в 1999 бриллиантовый (более 10’000’000 экземпляров).

Для нашего исследования набор данных был взят с сайта организации *RIAA*, нас интересовали только произведения, получившие платиновый статус. Поскольку сайт не предоставляет необходимые данные в прямом виде, был написан скрипт на языке *Python*, который, используя *API* сайта данной компании, извлекает нужные данные и сохраняет их в *CSV* файл.

Данный файл содержит в себе следующие колонки:

* ключ записи, число,
* имя артиста,
* название произведения,
* лейбл,
* дата сертификации,
* дата выпуска произведения,
* количество сертифицированных копий,
* жанр,
* выданный статус (gold, platinum, diamond).

# Обработка данных

Сгенерированный *CSV* файл был подгружен в *R* скрипт, после чего были исправлены колонки таблицы, которые были неправильно прочитаны языком *R*. Даты, количество сертифицированных копий были переведены в правильный вид. Для удобства обработанная таблица была отдельно сохранена в формате *Rda*, далее он используется для анализа.

Для отдалённого просмотра всех данных, которые мы имеем, была создана тепловая карта по музыкальным жанрам относительно даты выпуска произведения и количество сертифицированных копий.

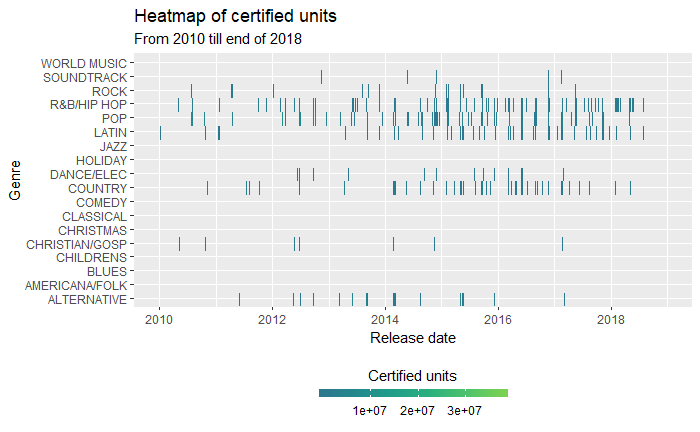


Рисунок 1 – Тепловая карта жанров и сертифицированных копий по дате

Как видно из рисунка 1 у некоторых жанров не хватает или нет записей по сертифицированным копиям, такие жанры не рассматриваются в данной работе. Далее рассмотрим срезку данных по жанру *POP*, с теми же параметрам.

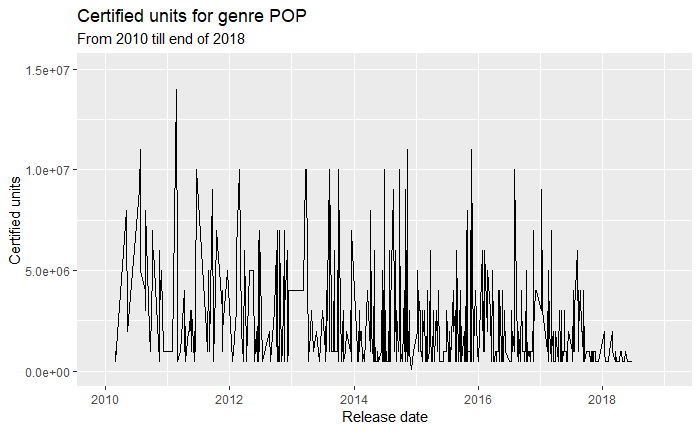


Рисунок 2 – График сертифицированных копий от выпуска для жанра POP

Пока что явной периодичности не видно из данных графиков. Перейдём к созданию модели и проверим есть ли зависимость.

# Построение моделей

Для построения моделей была выбрана модель *ARIMA* *(англ.* *autoregressive integrated moving average, модель Бокса-Дженкинса*) – интегрированная модель авторегрессии — скользящего среднего – модель и методология анализа временных рядов. Является расширением моделей *ARMA* для нестационарных временных рядов, которые можно сделать стационарными взятием разностей некоторого порядка от исходного временного ряда (так называемые интегрированные или разностно-стационарные временные ряды).

Для проверки качества модели используем *Q-тест Льюнг-Бокса* (*Ljung-Box test*), по его результату можно сказать если , то присутствует автокорреляция, что указывает на наличие зависимости.

В последующих моделях используются произведения жанра *POP*, которые были выпущены с 1 января 2000 до 31 декабря 2018. Из-за ограничения алгоритма *ARIMA*, количество сертифицированных копий группируются по месяцам.

При расчёте среднего отклонения модель данные делятся на тренировочные и тестовые. Тренировочные были взяты с промежутком с 1 января 2000 по 31 декабря 2014, в то время тестовые с 1 января 2015 по 31 декабря 2018. Для проверки качества предсказания высчитываются средние отклонения предсказанного от тестовых данных. Для теста *Ljung-Box* модель строится по всему периоду.

## Построение первичной модели с выбросами

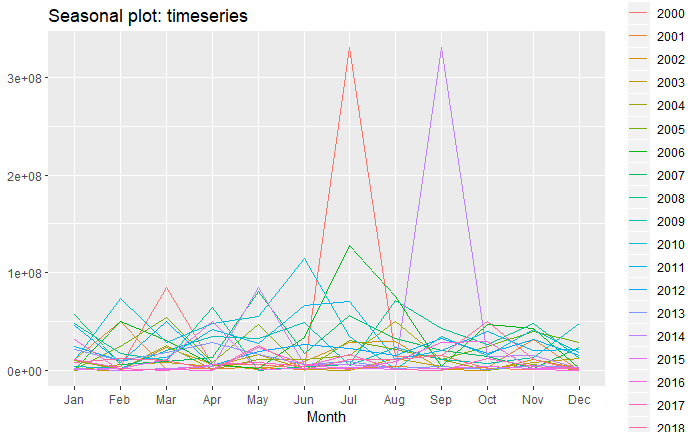


Рисунок 3 – График сертифицированных копий по годам

Из рисунка 3 сразу видно, что есть большие выбросы в двух годах. Некоторая периодичность в данных заметна, но не отчётливо.

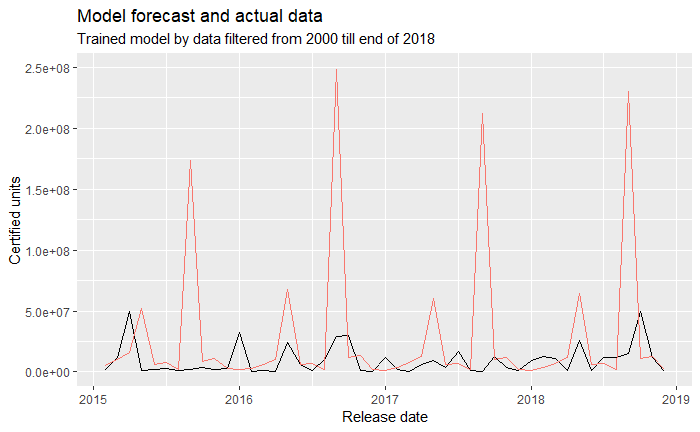


Рисунок 4 – График тестовых данных (чёрная линия) и предсказание модели (красная линия)

Таблица 1  
Средние отклонения модели относительно тестовых данных  


Стоить заметить, что отклонение предсказания по количеству сертифицированных копий произведения не является ошибкой. Модель должна правильно предсказывать пики продаж.

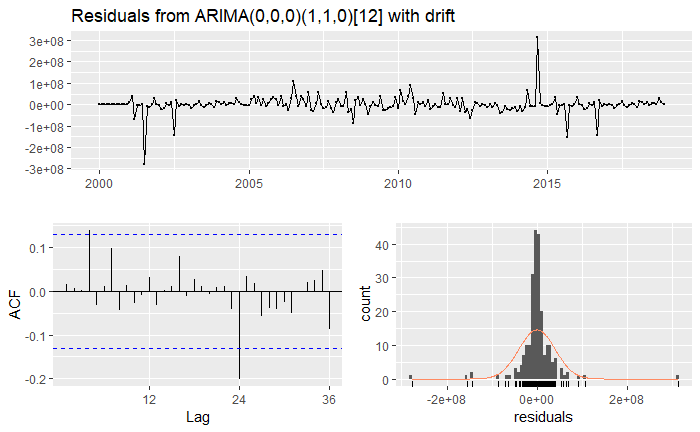


Рисунок 5 – Графики модели, построенной на всём промежутке

*Ljung-Box* тест выдал с и с . Что указывает на то, что нет зависимости сертифицированных копий произведения от даты.

## Проецирование продаж на логарифмическую шкалу

Такое действие было принято, чтобы уменьшить контраст пиков и простоев по продажам.

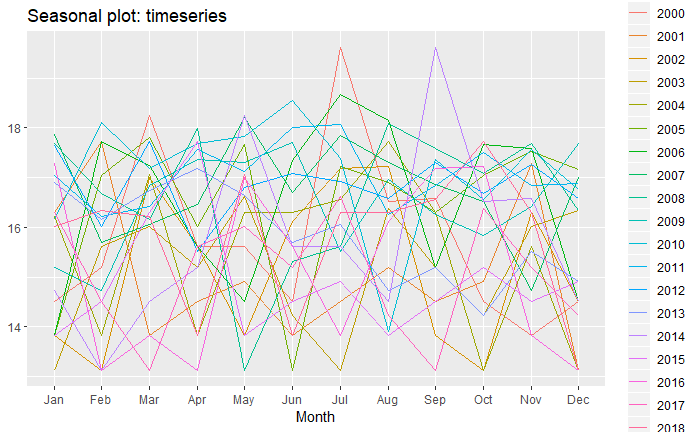


Рисунок 6 – График сертифицированных копий по годам

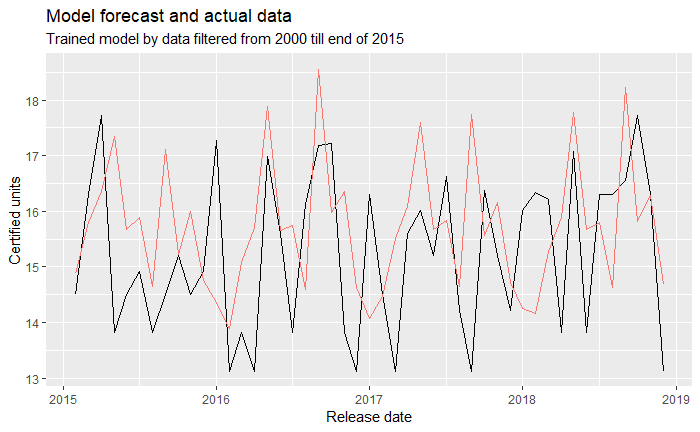


Рисунок 7 – График тестовых данных (чёрная линия) и предсказание модели (красная линия)

Таблица 2  
Средние отклонения модели относительно тестовых данных  


Данная модель уже более отчётливо предсказывает следующие пики, как это можно заметить на рисунке 7.

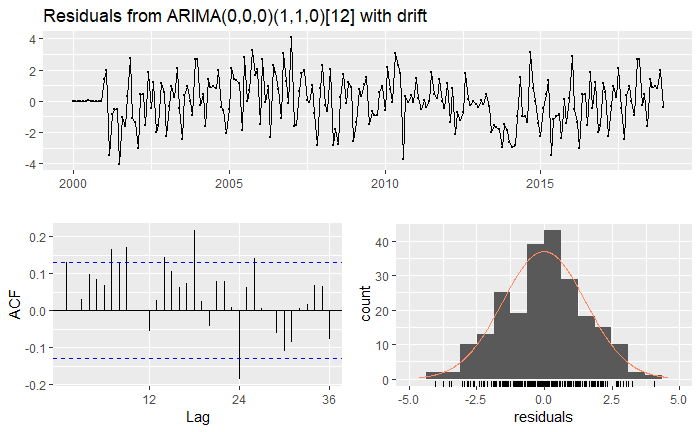


Рисунок 8 – Графики модели, построенной на всём промежутке

*Ljung-Box* тест выдал с и с . Что указывает на то, что присутствует зависимость количества сертифицированных копий от даты выпуска.

## Фильтрация набора данных и построение модели

Модель из пункта 3.2 уже выдала хороший результат, попробуем улучшить её, убрав выбросы. Для этого сделаем фильтрацию данных ориентируясь на Рисунок 6, из чего следует .

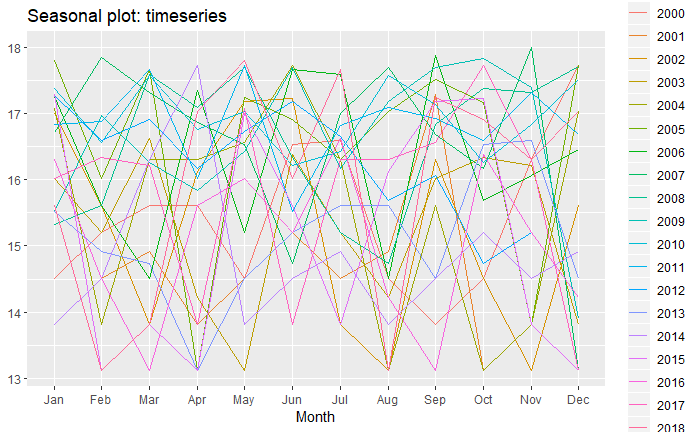


Рисунок 9 – График сертифицированных копий по годам

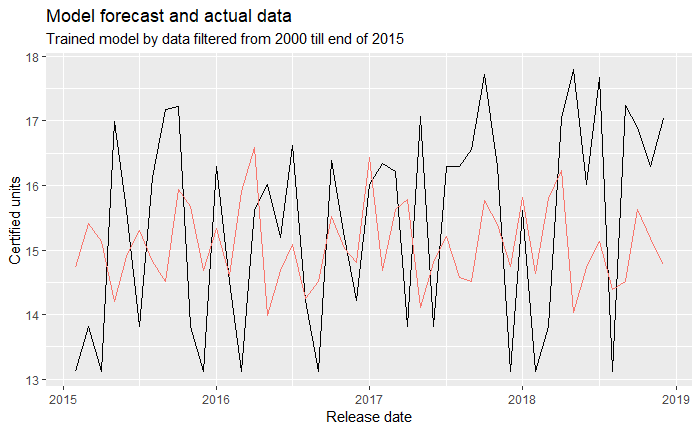


Рисунок 10 – График тестовых данных (чёрная линия) и предсказание модели (красная линия)

Таблица 3  
Средние отклонения модели относительно тестовых данных  

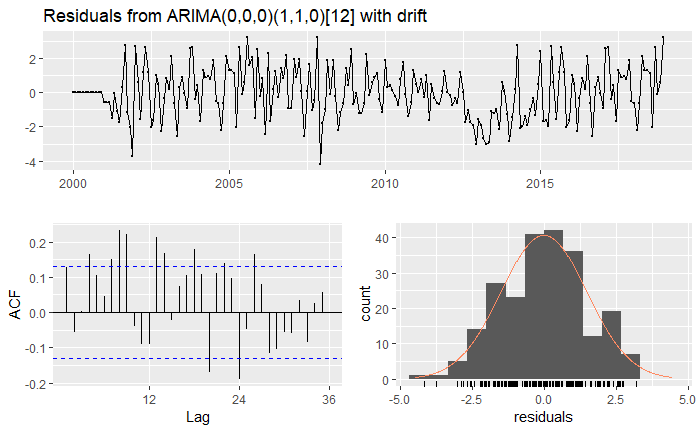



Рисунок 11 – Графики модели, построенной на всём промежутке

*Ljung-Box* тест выдал с и с . Что указывает на то, что присутствует зависимость количества сертифицированных копий от даты выпуска.

После удаление выбросов качество модели улучшилось.

## Проецирование продаж на гармоническую функцию

Если в пунктах 3.2 и 3.4 использовал логарифмическая функция, которая не является периодичной, было принято решение проецировать количество сертифицированных копий на периодичную функцию – тангенс.

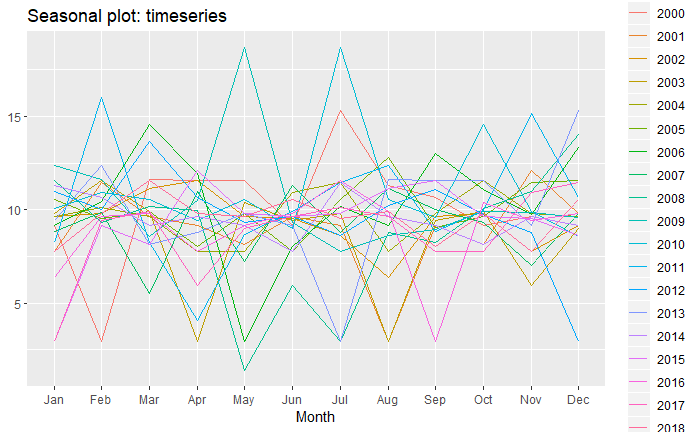


Рисунок 12 – График сертифицированных копий по годам

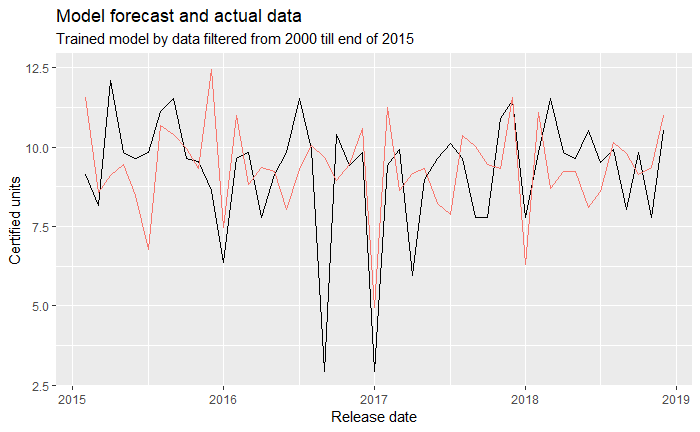


Рисунок 13 – График тестовых данных (чёрная линия) и предсказание модели (красная линия)

Таблица 4  
Средние отклонения модели относительно тестовых данных  

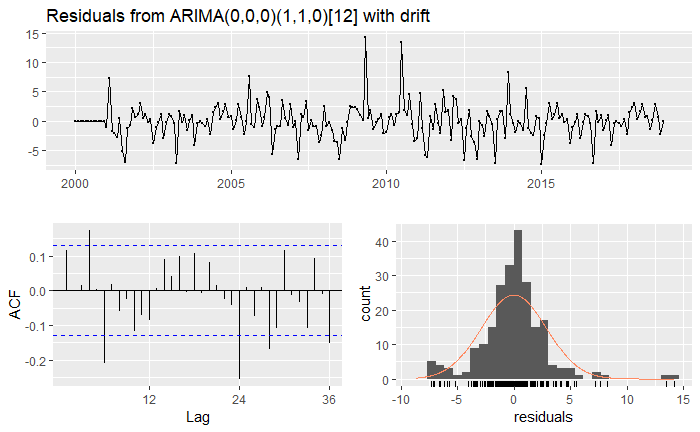



Рисунок 14 – Графики модели, построенной на всём промежутке

*Ljung-Box* тест выдал с и с . Что указывает на то, что присутствует зависимость количества сертифицированных копий от даты выпуска. Но качество модели получилось хуже, чем модель, которая получена в пункте 3.2.

# Разработка сервиса

После анализа моделей и их улучшения была выбрана модель из пункта 3.3. Для реализации сервиса, через который пользователь будет пользоваться аналитической моделью, был выбран язык программирования Python с Django framework.

Сервис позволяет конечному пользователю выбирать параметр жанр (рис. 15), на котором и строится дальше модель. Предлагаются только те жанры, по которым можно построить хорошую модель. Жанры берутся из R скрипта. Полный код R скрипта находится в приложении А.

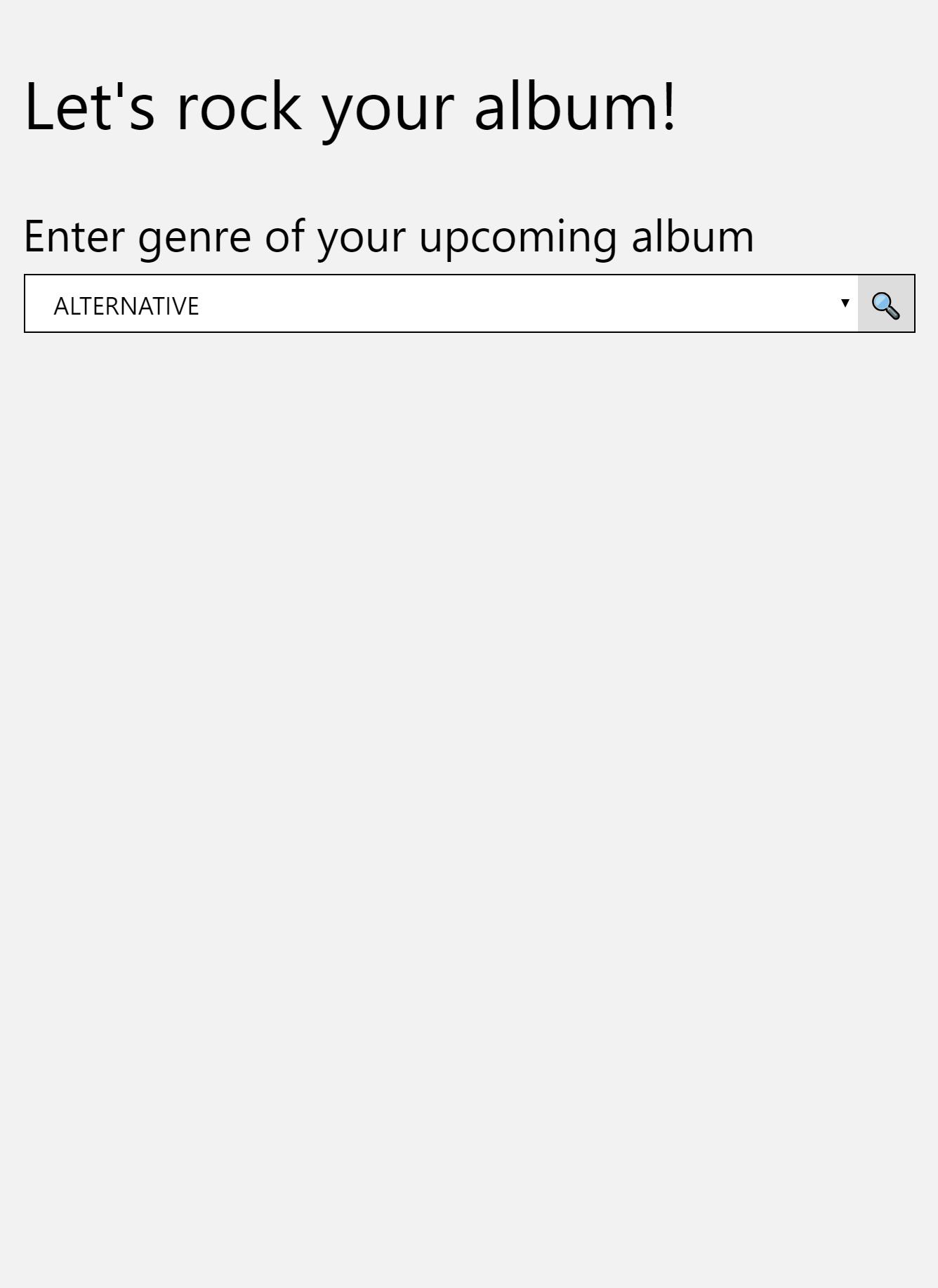


Рисунок 15 – Экран выбора жанра для модели

Когда нужно создать модель и получить её результат, сервис запускает R скрипт через подпроцесс и анализирует то, что ему процесс выдал в *stdout* и *stderr*. Чтобы порядок строк в R скрипт файле не влияли на извлечение данных на сервере, были введены маркеры, которые необходимо ставить перед выводом нужно информации.

Это позволяет убрать прямую зависимость от порядка вывода результата для *R* скрипта. В тоже время расширять данную систему можно при помощи добавления новых маркеров в скрипте и на сервисе.

Сервис предоставляет качество модели в двух представлениях: упрощённый вид и детальный. Упрощённый вид представляет из себя процент, который рассчитывается относительно значений тестов модели. Когда как детальный – перечисление всех параметров качества, их значение и рейтинг.

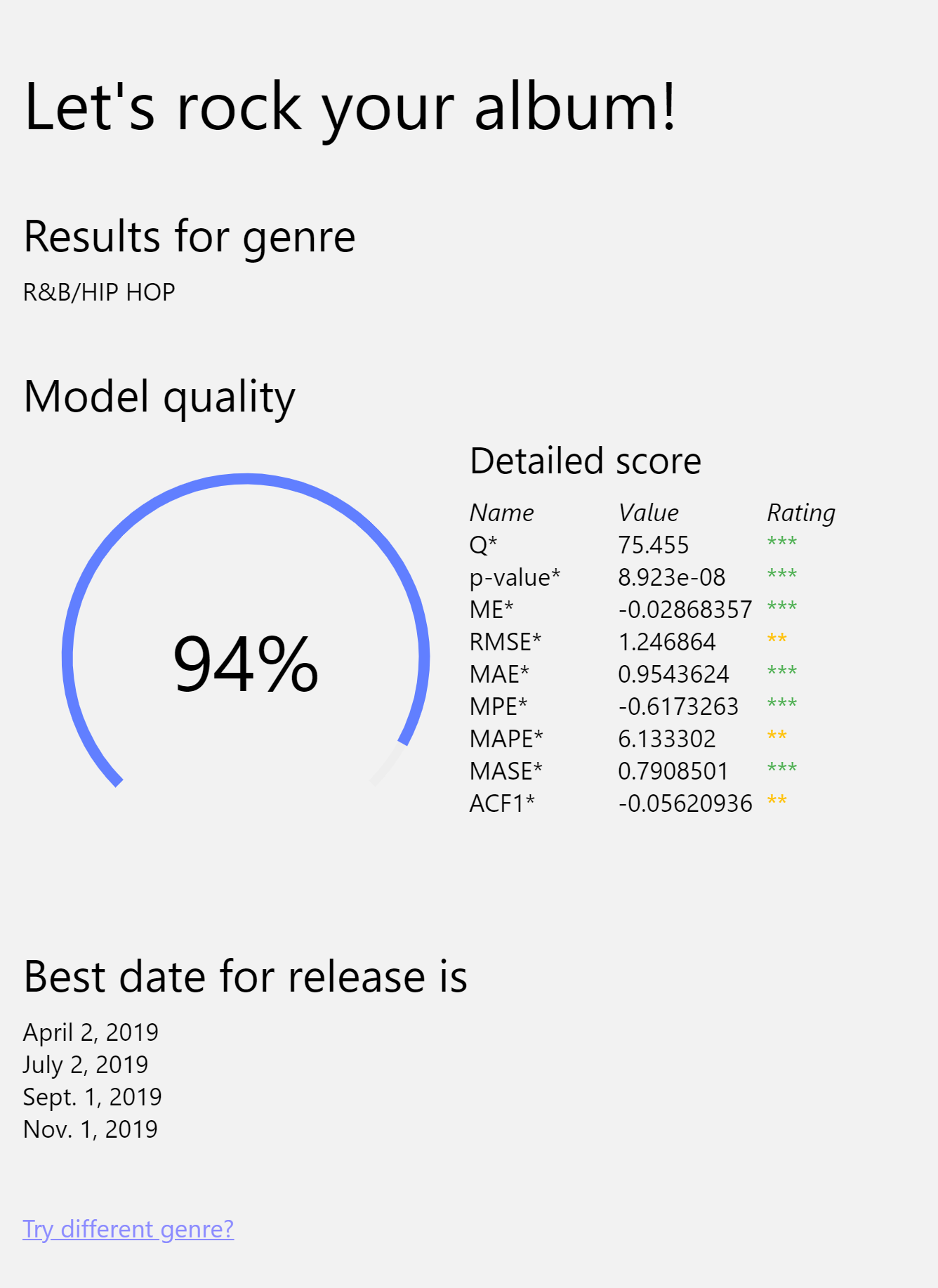


Рисунок 16 – Экран результата выполнения модели

Сервис не зависит от того, на какие данные *R* скрипт основывается, какой промежуток времени выбирает и тому подобное. Аналитик может изменять эту часть без какого-либо изменения со стороны кода сервиса. Данные можно обновлять, увеличивать. Результат выполнения модели хранится в сервисе 30 минут.

# Заключение

Таким образом, в ходе выполнения проекта, был написан скрипт, позволяющий извлечь нужные данные и сохранить их в *CSV* файл, полученный набор данных был обработан и проанализирован. После построения необходимых моделей догадка о существовании зависимости между количеством проданных копий (*Certificated units*) и датой выпуска релиза (*Release date*) подтвердилась.

Для проверки качества модели был использован *Q-тест Льюнг-Бокса,* который показал наличие зависимости. Кроме того, для улучшения качества модели были удалены выбросы и проведено проецирования данных с помощью различных вспомогательных математических функций.

На финальном этапе работы было спроектирован веб-сервис на языке Python с использованием Django framework. Сервис позволяет конечному пользователю выбирать параметр жанр, на котором и строится дальше модель, а также, в удобном формате отображает различные коэффициенты, средние отклонения полученной модели и сам результат.

R скрипт, который используется сервисом для построения модели, находится в приложении А.

# Библиографический список

1. Дубров А.М. Многомерные статистические методы /А.М. Дубров.– 2013. – 252 с.
2. Гришин А.Ф. Статистика: Учеб. пособие./ А.Ф. Гришин - 2003. – 240с.
3. Введение в многомерный анализ [Электронный ресурс]. Доступен по ссылке – https://habr.com/post/126810/ Последняя дата обращения: 25.11.18.
4. Кречетов Н. Продукты для интеллектуального анализа данных / Н. Кречетов, ComputerWeek-Москва. 2013 г. № 14-15. 32-39 с.
5. Вендров А.М. Практикум по проектированию программного обеспечения ЭИС: Учебное пособие / А.М. Вендров, Финансы и статистика, 2014. 215с.

# ПРИЛОЖЕНИЕ А

Исходный код R скрипта

library(optparse)

library(dplyr)

option\_list <- list(make\_option(c('--genre'), action='store', default=NA, type='character', help='Genre'))

opt = parse\_args(OptionParser(option\_list=option\_list))

setwd('..') # for script

load('riaa.Rda')

generate.model <- function (timeseries) {

model <- Arima(timeseries, seasonal=c(1,1,0), include.drift=T)

return(model)

}

test.model <- function (timeseries) {

ts.start <- start(timeseries)

ts.end <- end(timeseries)

ts.middle <- round(((ts.end[1] - ts.start[1]) \* 0.6) + ts.start[1])

model.train <- window(timeseries, start=c(ts.start[1], ts.start[2]),

end=c(ts.middle[1], 12))

model.test <- window(timeseries, start=c(ts.middle[1]+1, 1),

end=c(ts.end[1], ts.end[2]))

model <- generate.model(model.train)

acc <- accuracy(forecast(model, h=length(model.test)), model.test)

print('!&ME') print(acc[,1])

print('!&RMSE') print(acc[,2])

print('!&MAE') print(acc[,3])

print('!&MPE') print(acc[,4])

print('!&MAPE') print(acc[,5])

print('!&MASE') print(acc[,6])

print('!&ACF1') print(acc[,7])

}

if (opt$genre == 'all') {

print(levels(time.range$Genre))

} else {

library(spatstat)

library(forecast)

library(lubridate)

ts.range <- filter(time.range, Genre == opt$genre)

ts.range.start = ts.range$Start[1]

ts.range.end = ts.range$End[1]

ts.lag <- ts.range$Lag[1]

remove('ts.range')

table %>%

filter(Genre == opt$genre) %>%

filter(Release.date > ts.range.start) %>%

filter(Release.date < ts.range.end) %>%

select(c(Certified.Units, Release.date)) %>%

group\_by(Release.date=floor\_date(Release.date, "month")) %>%

summarise(Certified.Units=sum(Certified.Units)) %>%

mutate(Certified.Units=log(Certified.Units)) %>%

filter(Certified.Units > 13 & Certified.Units < 18) %>%

arrange(Release.date) %>%

select(c(Certified.Units)) %>%

ts(.,

start=c(year(ts.range.start), month(ts.range.start)),

end=c(year(ts.range.end), month(ts.range.end)),

frequency=12) -> timeseries

model <- generate.model(timeseries)

ljung.chi <- Box.test(model$residuals, lag=ts.lag, type='Ljung-Box', fitdf=0)

print('!&chisqr') print(ljung.chi$statistic)

print('!&chipvalue') print(ljung.chi$p.value)

print('!&ljung') print(checkresiduals(model, plot=F))

model.forecast <- forecast(model, h=24)

prediction.sales <- model.forecast$upper[,2][1:12]

pks<-which(diff(sign(diff(prediction.sales, na.pad=F)), na.pad=F) < 0) + 2

peaks <- pks[prediction.sales[pks - 1] - prediction.sales[pks] > 0] - 1

dates <- round(peaks / 12 \* 365)

print('!&dates') print(dates)

test.model(timeseries)

}