As Avaliações do Spotify Ao Longo do Tempo

Alisson Rosa

Resumo

A maioria das análises [1] empregando o banco de dados utilizado nesse trabalho utiliza de técnicas de processamento natural de linguagem para predizer o review do usuário, entretanto o que foi desenvolvido no presente trabalho é em uma perspectiva de série temporal, onde são utilizados modelos adequados para tal abordagem.

Sumário

1 Introdução

O Spotify é um serviço digital que dá acesso instantâneo a milhões de músicas, podcasts, vídeos e outros conteúdos de criadores no mundo todo. Pode ser acessado pelo *browser*, sendo possível também baixar o aplicativo, estando disponível para diversas plataformas digitais. Nesse ensaio, vamos analisar os *reviews* feitos na Play Store sobre a versão para *Android*, dessa maneira vamos examinar o comportamento da média dos reviews (estrelas) diária em uma perspectiva de série temporal, uma outra abordagem utilizando técnicas de NLP pode ser consultada clicando-se aqui.

2 Análise Descritiva

Devemos iniciar avaliando o comportamento da série ao longo do tempo, para averiguar a existência de evidências de não estacionariedade ou sazonalidade, tal fato pode ser visto pela Figura 1.

[1] Podem ser consultadas aqui

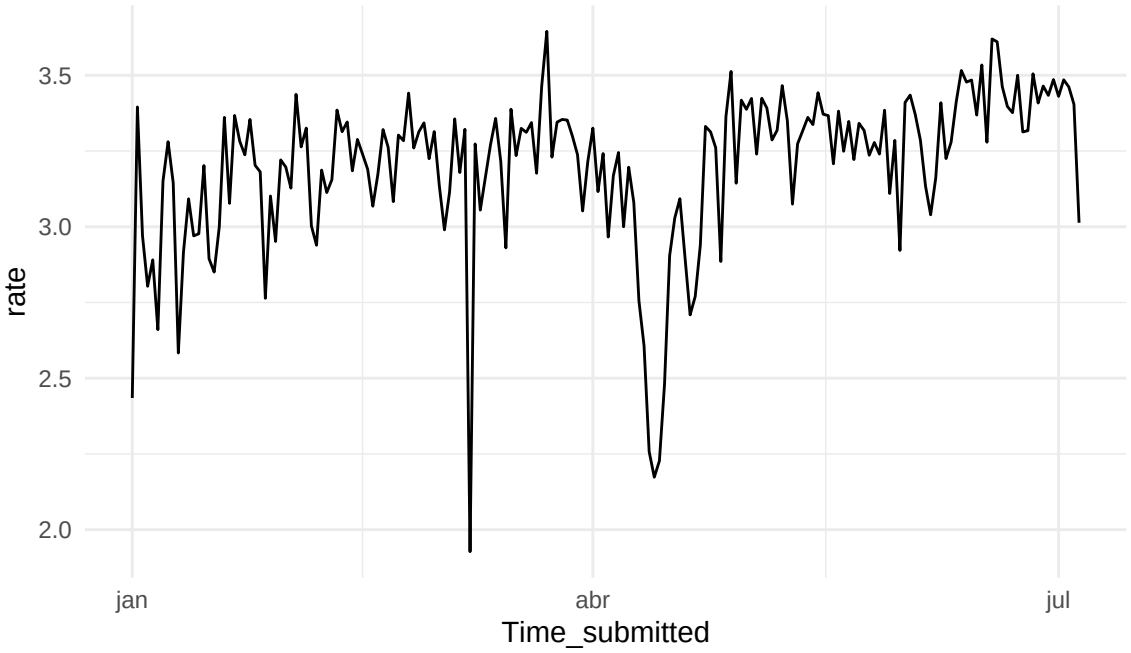
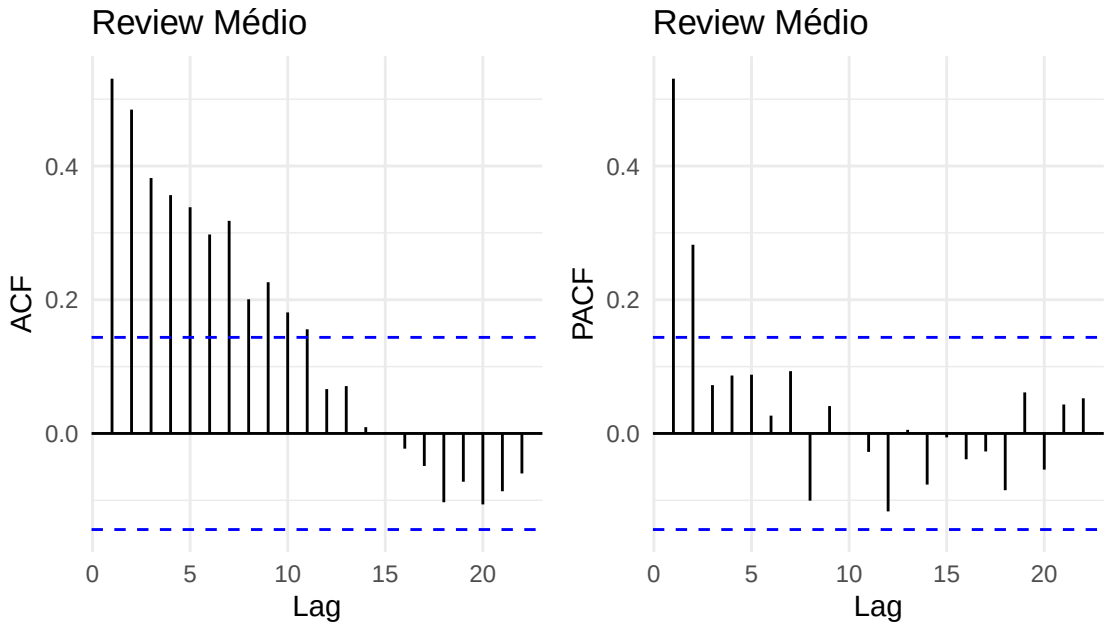


Figura 1: Comportament Médio dos Reviews ao Longo do Tempo

Nota-se que a série tem alguns picos de avaliações médias baixas, porém tende a variar em torno de uma média, pode ser visto também a não existência de sazonalidade. Além da análise visual faz-se necessário a utilização de teste de hipóteses para averiguar os fatos constatados anteriormente, dessa forma, antes da escolha do modelo realizou-se o teste de raiz unitária de Dickey-Fuller Aumentado (ADF). Sendo obtido um de p-valor= 0.05, ao nível de significância de 5%, a hipótese de não estacionariedade é rejeitada. Assumindo que a série estacionária procedeu-se a utilização dos modelos descritos na seguinte seção.

A Figura **??** fornece a função de autocorrelação e autocorrelação parcial, Pode-se notar por inspeção visual da figura, que não existem evidências da série não ser estacionária, assim como visto pelo teste ADF e pela Figura 1.



Pela Tabela 1 nota-se que a média e mediana do review médio diário estão bem proximas, o máximo e mínimo das avalaições não tendem a atingir limiares extremos, como uma avaliação invidídual pode alcançar.

Tabela 1: Resumo da Avaliação Média

variable	mean	median	sd	min	max	na_count
y	0.64	0.653	0.053	0.385	0.729	0

3 Metodologia

Nessa seção são apresentados os modelos aqui utilizados, a saber: βARMA, KARMA e ARIMA[2]

3.1 Modelo βARMA

A distribuição beta é bastante conhecida pois consegue modelar variáveis aleatórias definidas em intervalos limitados, dessa maneira um caso particular importante é quando o intervalo é unitário iniciando em zero. Dessa maneira, foi desenvolvido em [1] o modelo com abordagem temporal para variáveis que podem ser modeladas pela distribuição beta.

Portanto, assumindo que a variável resposta está definida no intervalo $(0, 1)$ o modelo assume que a cada variável Y_t pode ser escrita da seguinte maneira:

$$g(\mu_t) = \alpha + x_t^T\beta + \sum_{i=1}^{p} \varphi_i[g(y_{t-i}) - x_{t-1}^T\beta] + \sum_{j=1}^{q} \theta_j r_{t-j} \tag{1}$$

[2]ARIMA: O modelo ARIMA já é bastante conhecido na literatura, portanto aqui é dispensado a sua introdução

3.2 Modelo KARMA

O modelo KARMA foi introduzido no contexto de séries temporais com o intuito de acomodar a presença de correlação serial na modelagem da mediana condicional da distribuição Kumaraswamy. O modelo KARMA[3] proposto em [2] assume que a mediana de cada Y_t pode ser escrita da seguinte maneira:

$$g(\mu_t) = \alpha + x_t^T\beta + \sum_{i=1}^{p}\varphi_i[g(y_{t-i}) - x_{t-1}^T\beta] + \sum_{j=1}^{q}\theta_j r_{t-j} \tag{2}$$

4 Ajuste dos Modelos

Na Tabela 2 são apresentadas as estimativas, o erro padrão e o p-valor associado ao teste de significância dos parâmetros do modelo βARMA(4,3), obtido pelo menor AIC. Verifica-se que os parâmetro são significativos ao nível de significância de 5%

```
## [1] BARMA model
```

Tabela 2: Resumo do Ajuste BARMA

	Estimate	Std. Error	z value	Pr(>|z|)
alpha	0.0626	0.0572	1.0949	0.2736
phi1	1.1044	0.3749	2.9459	0.0032
phi3	-0.4319	0.3951	1.0931	0.2743
phi4	0.1333	0.1343	0.9919	0.3213
phi5	0.0857	0.1026	0.8356	0.4034
theta1	-0.7302	0.3830	1.9063	0.0566
theta2	-0.0599	0.1746	0.3428	0.7318
theta3	0.2400	0.2972	0.8076	0.4193
precision	139.4347	14.6081	9.5450	<0.001

A Figura 2 apresenta o gráfico da série com os valores reais juntamente com os valores estimados pelo modelo. O gráfico indica uma boa qualidade do ajuste, já que valores reais e previstos são muito próximos.

[3]Para mais detalhes consultar o artigo

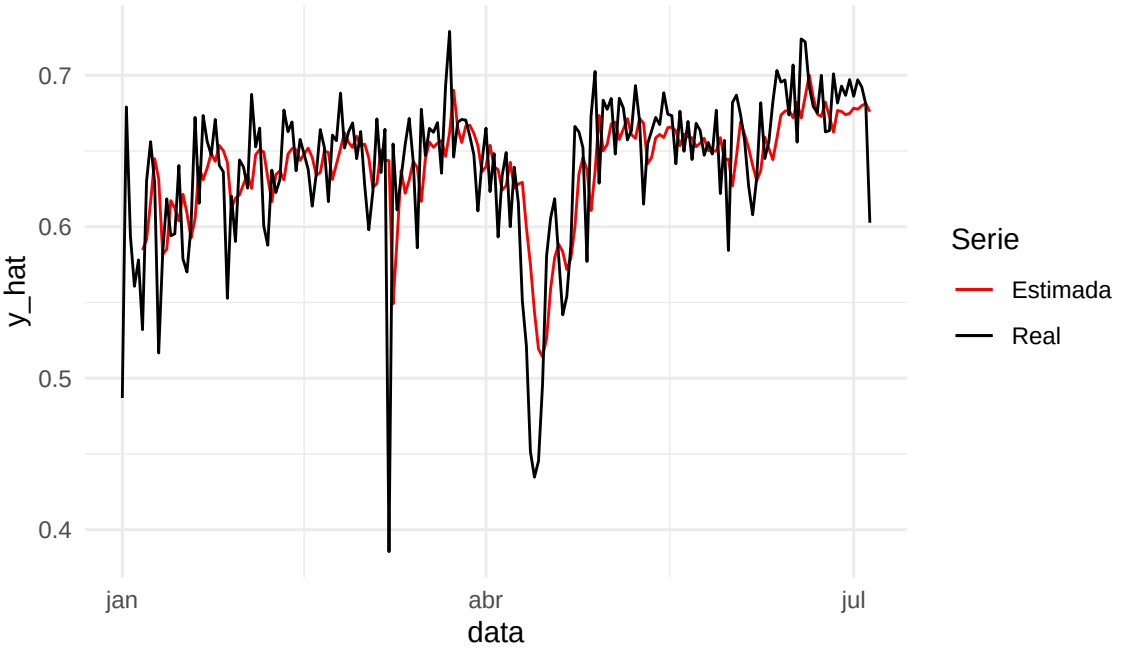


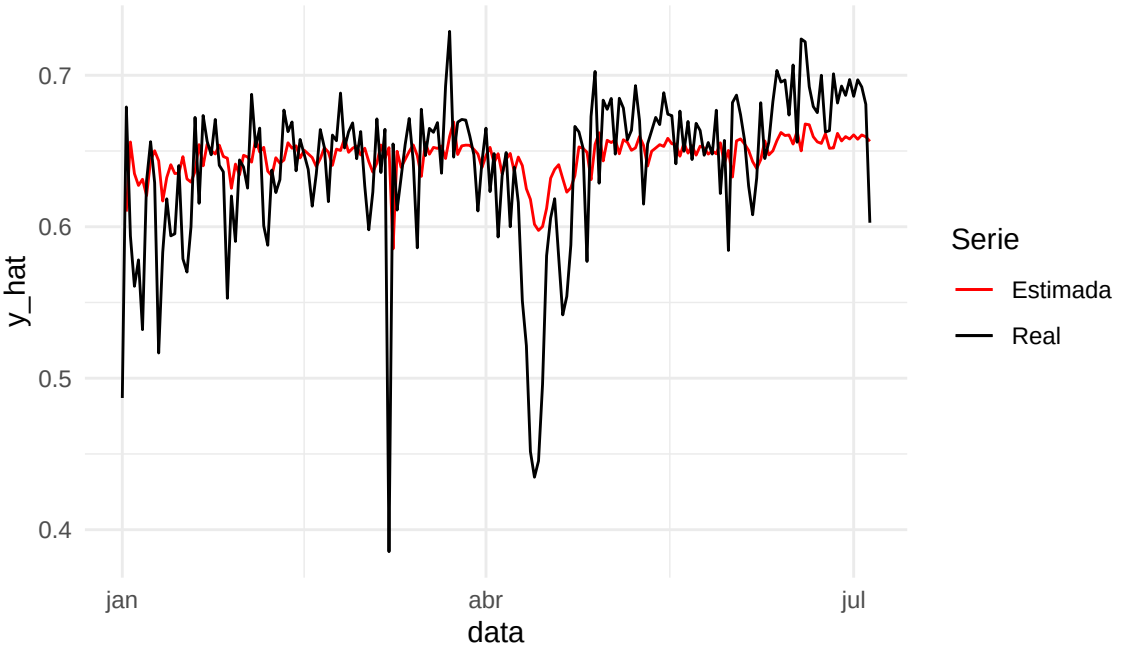
Figura 2: Valores Reais versus Ajustados pelo Modelo Barma

A Tabela 3 fornece as estimativas que foram apresentadas para o modelo βARMA, só que agora para o modelo KARMA(1,0).

Tabela 3: Resumo do Ajuste KARMA

	Estimate	Std. Error	z value	Pr(>|z|)
alpha	0.4602	0.0287	16.037	<0.001
phi1	0.2470	0.0458	5.394	<0.001
precision	19.1667	1.1315	16.939	<0.001

Podemos pela Figura **??** ver que o modelo KARMA visualmente estima bem a série.



Porém fica inviável comparar qual tem melhor desempenho somente por gráficos, dessa maneira assim na seção rr são fornecidas medidas para comparação de modelos. Dessa maneira para confirmar se o modelo está bem ajustado foi realizada a análise de diagnóstico na seguinte seção.

4.1 Análise de Diagnóstico

Considerando as Figura 3 que fornece os gráficos para diagnóstico do modelo βARMA, pode-se observar que em apenas uma defasagem o valor é superior ao intervalo de confiança, dessa forma, aparentemente a suposição de resíduos não correlacionados é satisfeita, ainda baseado no teste de Ljung-Box, testou-se correlação nula até a defasagem 20, tem-se p-valor=0.3568, logo os resíduos são não correlacionados. A Figura também apresenta os resíduos padronizados ao longo dos indíces, observa-se que apresentam um comportamento aleatório em torno de zero, com os valores dentro dos limites estipulados. Para o gráfico Q-Q plot, verifica-se que a maioria dos pontos se encontra sobre a linha diagonal, indicando a proximidade dos resíduos da distribuição Normal.

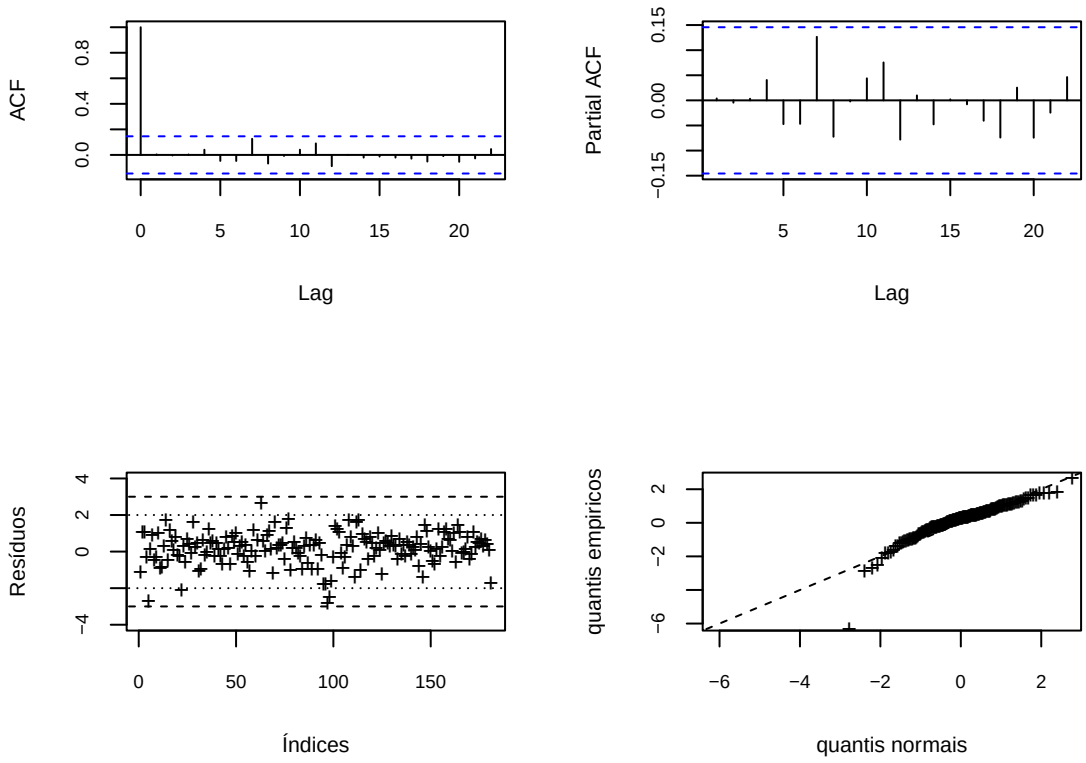


Figura 3: Comportamento dos Resíduos para o Modelo Barma

Pela Figura 4, temos os mesmos gráficos porém para o modelo KARMA, nesse caso nenhum dos pressupostos é satisfeitos, dado que temos inúmeras autocorrelações superiores aos intervalos de confiança, indicando autocorrelação nos resíduos. Os resíduos versus índices indica um padrão e também os resíduos estão bastantes "distantes" da distribuição normal.

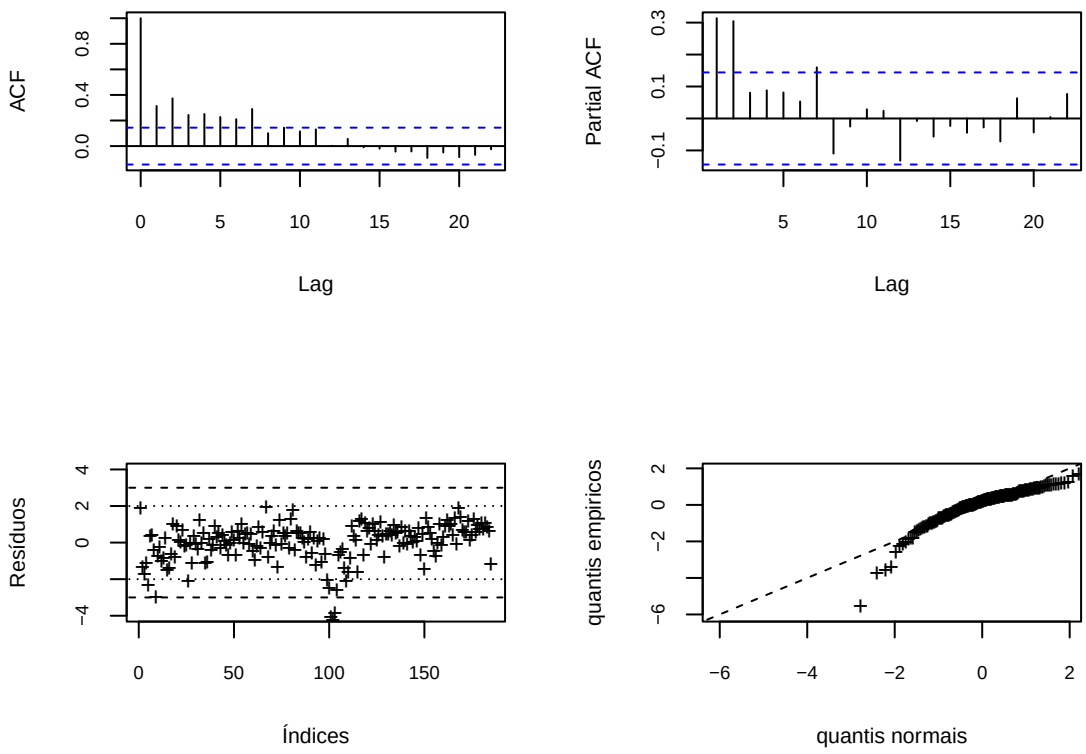


Figura 4: Comportamento dos Resíduos para o Modelo KARMA

4.2 Comparação dos Modelos

Para comparar os modelos vamos utilizar de duas medidas que avaliam a diferença entre os valores reais y e os valores preditos pelo modelo $\hat{\mu}$, sendo elas o erro quadrático médio (EQM) e o erro percentual absoluto médio (MAPE), sendo definidas:

$$\text{EQM} = \frac{1}{h}\sum_{i=1}^{h}(y_i - \hat{\mu}_i)^2 \quad ;\text{MAPE} = \frac{1}{h}\sum_{i=1}^{h}\frac{|y_i - \hat{\mu}_i|}{|y_i|} \tag{3}$$

Pela tabela 4 notamos o que era esperado pela análise de diagnóstico, o modelo KARMA tende a ter um desempenho inferior aos demais, e por uma leve diferença o modelo βARMA tende a ser o melhor nos dados de treino/ajuste.

Tabela 4: Medidas nos Período de Ajuste

	EQM	MAPE
BARMA	0.0016	0.0456
KARMA	0.0021	0.0537
ARIMA	0.0017	0.0466

Entretanto os resultados são diferente para o caso dos dados de teste/previsão. pois o que se nota pela Tabela 5 é que o modelo βARMA tende a ter um desempenho inferior em ambas as métricas. Uma situação que evidência que alinhamento de pressupostos nem sempre acarreta em melhores previsões.

Tabela 5: Medidas nos Período de Previsão

	EQM	MAPE
BARMA	0.0019	0.0613
KARMA	0.0018	0.0590

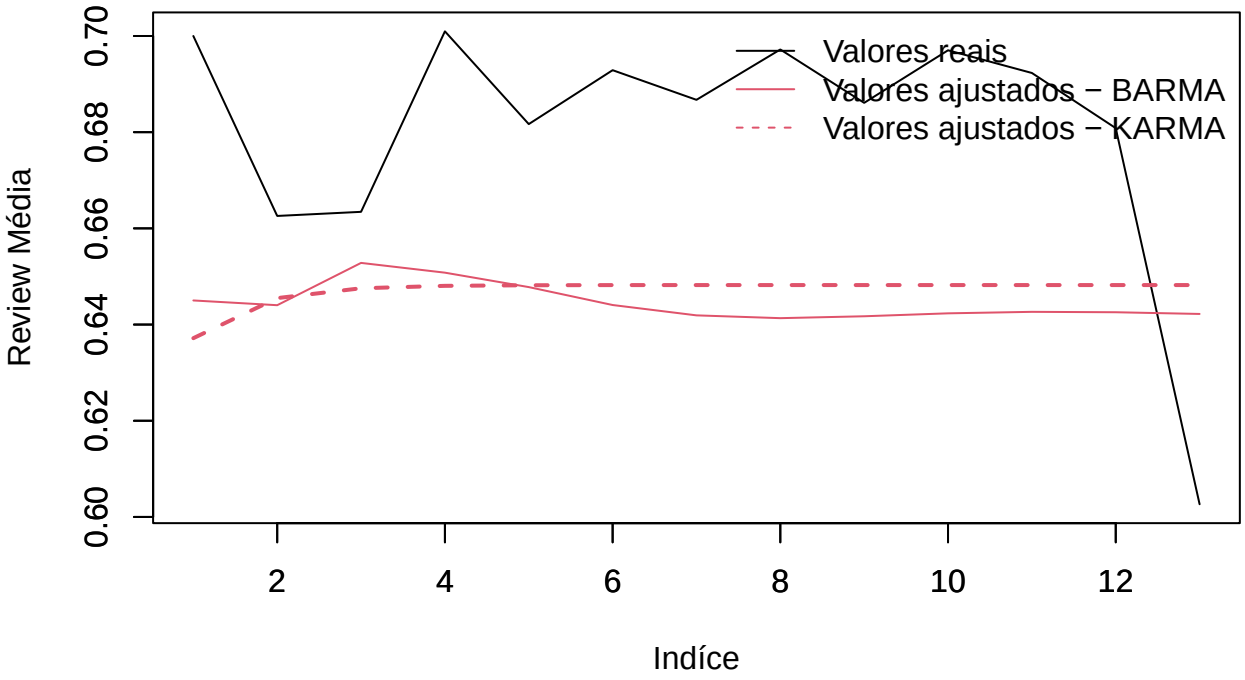


Figura 5: Dados Reais e Ajustados pelo Modelo para os dados de teste

5 Conclusão

Contemplamos o comportamento dos reviews médio diário do Spotify na *Play Store*, nota-se que não existem evidências de não estacionaridade e nem para a existência de sazonalidade. Para os modelos ajustados, o modelo KARMA não teve um desempenho agradável nem atende aos pressupostos teóricos, dado o comportamento dos resíduos, o modelo βARMA foi o que obteve destaque, tanto em termos de pressupostos quanto em perfomance nos dados de treino, entretanto para os dados de teste sofreu a perda em termos da métrica para o KARMA.

6 Código

```r
# R code
library(tidyverse)
library(forecast)
library(lubridate)
library(patchwork)




eqm <- function(model, serie = y, ar = 2) {
  ar <- ar + 1
  resid <- serie[ar:length(serie)] - model$fitted[ar:length(serie)]
  return(sum(resid^2) / (length(resid) - 1))
}

mape <- function(model, serie = y, ar = 2) {
  ar <- ar + 1
  resid <- serie[ar:length(serie)] - model$fitted[ar:length(serie)]
  return((1 / length(serie)) * sum(abs(resid) / abs(y[ar:length(y)])))
}

source("analise_barmax/barma.r")
source("KARMA_BARMA/karma.r")

df <- read_csv("../data/rates.csv")

df <- df |>
  mutate(Time_submitted = as.Date(Time_submitted, format = "%Y-%m-%d"))


df <- df |>
  group_by(
    Time_submitted
  ) |>
  summarise(rate = mean(Rating))

df <- df[1:(nrow(df) - 4), ]

df <- df |> bind_cols(index = 1:nrow(df))
y <- df$rate / 5
y <- ts(y, start = decimal_date(as.Date("2022-01-01")), frequency = 365)
plot_serie <- df |>
  ggplot(aes(x = Time_submitted, y = rate)) +
  geom_line()

plot_serie
pvalor <- tseries::adf.test(y)$p.value
ggAcf(df$rate) + labs(title = "Review Médio") + ggPacf(df$rate) + labs(title = "Review Médio")
y |>
```

```r
  tibble() |>
  fastrep::describe() |>
  fastrep::tbl("Resumo da Avaliação Média")
fit_barma <- barma(y, ar = c(1, 3, 4, 5), ma = c(1, 2, 3), h = 13, diag = 0)

fit_barma$model |>
  as.data.frame() |>
  mutate(`Pr(>|z|)` = format.pval(`Pr(>|z|)`, eps = 0.001)) |>
  fastrep::tbl("Resumo do Ajuste BARMA")

real_hat <- function(fitted) {
  fitted |>
    tibble() |>
    rename(y_hat = fitted) |>
    bind_cols(y = df$rate / 5, data = df$Time_submitted) |>
    ggplot() +
    geom_line(aes(data, y_hat, color = "Estimada")) +
    geom_line(aes(data, y, color = "Real")) +
    scale_color_manual(name = "Serie", values = c("Estimada" = "red", "Real" = "black"))
}



real_hat(fit_barma$fitted)
fit_karma <- karma(y, ar = c(1), h = 13, diag = 0)

fit_karma$model |>
  as.data.frame() |>
  mutate(`Pr(>|z|)` = format.pval(`Pr(>|z|)`, eps = 0.001)) |>
  fastrep::tbl("Resumo do Ajuste KARMA")
real_hat(fit_karma$fitted)
par(mfrow = c(2, 2))

res <- fit_barma$resid1
resi_padrao <- as.vector((res) / (sd(res)))
acf(res, cex.lab = 0.8, cex.main = 0.3, cex.axis = 0.7, main = "")
pacf(res, cex.lab = 0.8, cex.main = 0.3, cex.axis = 0.7, main = "")

n <- length(res)
t <- seq(-5, n + 6, by = 1)

plot(res, main = " ", xlab = "Índices", ylab = "Resíduos", pch = "+", ylim = c(-4, 4), cex.lab = 0.8, cex.ma
lines(t, rep(-3, n + 12), lty = 2, col = 1)
lines(t, rep(3, n + 12), lty = 2, col = 1)
lines(t, rep(-2, n + 12), lty = 3, col = 1)
lines(t, rep(2, n + 12), lty = 3, col = 1)



max_r <- max(res, na.rm = T)
```

```r
min_r <- min(res, na.rm = T)
qqnorm(resi_padrao,
  pch = "+",
  xlim = c(0.95 * min_r, max_r * 1.05),
  ylim = c(0.95 * min_r, max_r * 1.05),
  main = "", xlab = "quantis normais", ylab = "quantis empiricos", cex.lab = 0.8, cex.main = 0.3, cex.axis =
)
lines(c(-10, 10), c(-10, 10), lty = 2)
par(mfrow = c(2, 2))

res <- fit_karma$resid1
resi_padrao <- as.vector((res) / (sd(res)))
acf(res, cex.lab = 0.8, cex.main = 0.3, cex.axis = 0.7, main = "")
pacf(res, cex.lab = 0.8, cex.main = 0.3, cex.axis = 0.7, main = "")

n <- length(res)
t <- seq(-5, n + 6, by = 1)

plot(res, main = " ", xlab = "Índices", ylab = "Resíduos", pch = "+", ylim = c(-4, 4), cex.lab = 0.8, cex.ma
lines(t, rep(-3, n + 12), lty = 2, col = 1)
lines(t, rep(3, n + 12), lty = 2, col = 1)
lines(t, rep(-2, n + 12), lty = 3, col = 1)
lines(t, rep(2, n + 12), lty = 3, col = 1)
```

```r
max_r <- max(res, na.rm = T)
min_r <- min(res, na.rm = T)
qqnorm(resi_padrao,
  pch = "+",
  xlim = c(0.95 * min_r, max_r * 1.05),
  ylim = c(0.95 * min_r, max_r * 1.05),
  main = "", xlab = "quantis normais", ylab = "quantis empiricos", cex.lab = 0.8, cex.main = 0.3, cex.axis =
)
lines(c(-10, 10), c(-10, 10), lty = 2)
modelo <- auto.arima(y,
  max.p = 5, max.q = 5, max.P = 5, max.Q = 5, max.order = 5, max.d = 2, max.D = 1,
  start.p = 1, start.q = 1, start.P = 1, start.Q = 1, stationary = F
)
metrics <- data.frame(
  EQM = c(eqm(fit_barma, ar = 5), eqm(fit_karma), eqm(modelo)),
  MAPE = c(mape(fit_barma, ar = 5), mape(fit_karma), mape(modelo))
)
row.names(metrics) <- c("BARMA", "KARMA", "ARIMA")

metrics |> fastrep::tbl("Medidas nos Período de Ajuste")

karma_pred <- fit_karma$forecast

barma_pred <- fit_barma$forecast
```

```r
test <- y[(length(y) - 12):length(y)]



###################### EQM e MAPE período previsao ######################

############### EQM #################
residuos_beta_prev <- (test - barma_pred)
# residuos_sarima_prev = (test-as.vector(modelo$))
residuos_arma_prev <- (test - as.vector(karma_pred))

eqm_beta_prev <- (sum(residuos_beta_prev^2)) / length(residuos_beta_prev)
# (eqm_sarima_prev = (sum(residuos_sarima_prev^2))/length(residuos_sarima_prev))
eqm_arma_prev <- (sum(residuos_arma_prev^2)) / length(residuos_arma_prev)

############### MAPE #################

maple_beta_prev <- sum(abs(residuos_beta_prev) / abs(test)) / length(residuos_beta_prev)
# (maple_sarima_prev = sum( abs(residuos_sarima_prev)/abs(za) )/ length(residuos_sarima_prev))
maple_ar_prev <- sum(abs(residuos_arma_prev) / abs(test)) / length(residuos_arma_prev)


metrics <- data.frame(
  EQM = c(eqm_beta_prev, eqm_arma_prev),
  MAPE = c(maple_beta_prev, maple_ar_prev)
)
row.names(metrics) <- c("BARMA", "KARMA")

metrics |> fastrep::tbl("Medidas nos Período de Previsão")


plot(test, col = 1, type = "l", axes = T, main = "", xlab = "Indíce", ylab = "Review Média")
lines((barma_pred), lty = 1, lwd = 1, col = 2)
lines((karma_pred), lty = 2, lwd = 2, col = 2)
legend("topright", c("Valores reais", "Valores ajustados - BARMA", "Valores ajustados - KARMA"), # pch=vpch,
  pt.bg = "white", lty = c(1, 1, 2), col = c(1, 2, 2), bty = "n"
)
axis(1)
axis(2)
```

Bibliografia

[1] Rocha AV, Cribari-Neto F. Beta autoregressive moving average models. Test. 2009;18:529.

[2] Bayer FM, Bayer DM, Pumi G. Kumaraswamy autoregressive moving average models for double bounded environmental data. Journal of Hydrology. 2017;555:385–396.

[3] R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2021. Available from: https://www.R-project.org/.

[4] Rosa A. Fastrep: fastrep. 2022.