Clusterização nos dados de Valorant

Luiz Felipe De Oliveira Barbosa Nunes (RA 255403)

1 Introdução

Valorant é um jogo de tiro em primeira pessoa organizado em partidas onde a equipe atacante tenta plantar e defender uma bomba, chamada Spike, até sua detonação para ganhar pontos. Enquanto isso, a equipe defensora tenta desarmar a Spike ou eliminar todos os adversários para também ganhar pontos. As partidas são jogadas no formato "melhor de 24 rodadas", onde a equipe que primeiro alcançar 13 vitórias é declarada vencedora.

No Valorant, os jogadores são categorizados em quatro classes: Duelistas, Controladores, Sentinelas e Iniciadores. Cada classe possui habilidades distintas que definem suas funções estratégicas no jogo, tornando essencial a escolha e utilização adequada das classes para o sucesso da equipe.

Este trabalho aplica a análise de clusterização por Modelos de Mistura Gaussiana (GMM) aos dados de desempenho dos jogadores de Valorant, buscando identificar padrões e agrupamentos entre os jogadores com base em suas estatísticas de jogo. O foco é analisar quais classes têm maior impacto no jogo. Além disso, é realizada uma comparação com a clusterização pelo método k-means para avaliar a eficácia e a qualidade dos agrupamentos gerados.

Os scripts e códigos utilizados neste trabalho estão disponíveis em: https://github.com/LuizNunes2020/Atividade2.ME921. Para a leitura dos dados do site https://www.vlr.gg/stats, utilizei uma API (Application Programming Interface) que também está disponível no GitHub. Esta API permite a extração e manipulação dos dados.

2 Matérias e Métodos

O banco de dados utilizado possui um total de 14 variáveis e 167 observações, representando os jogadores. Embora haja mais jogadores disponíveis, selecionei apenas os 167 primeiros para este estudo. No entanto, para o experimento, utilizei apenas as 5 variáveis com as maiores cargas nas componentes principais, conforme identificado pela Análise de Componentes Principais (PCA). Sendo elas:

- **kills_per_round**: número de kills por rodada.
- **kill_deaths**: razão de mortes por mortes.
- **headshot_percentage**: porcentagem geral de headshots.
- **clutch_success_percentage**: porcentagem de sucesso em situações de clutch.
- **rating**: pontuação geral do jogador.

Com os dados em mãos, não foi necessário realizar nenhuma limpeza ou filtragem adicional. Experimentei várias transformações nos dados, incluindo transformações logarítmicas, de Box-Cox e quadráticas, com o objetivo de melhorar a qualidade dos agrupamentos. No entanto, essas transformações não resultaram em melhorias significativas para o modelo. Portanto, os dados foram analisados sem transformações adicionais.

Para criar o modelo de Mistura Gaussiana (GMM), foi elaborado um scatterplot 2 a 2 das variáveis das observações, bem como o gráfico BIC. O primeiro gráfico nos auxilia na verificação da suposição do modelo,

enquanto o segundo auxilia na escolha de seus parâmetros. Por fim, analisei o resultado da clusterização utilizando o plot da incerteza associada a cada uma das observações.

Além do clustering por GMM, para fins de comparação, também realizei uma clusterização por k-means. A escolha desse algoritmo visa comparar um método de clusterização baseado em modelo estatístico com um que não é.

2.1 Análise de Agrupamento com Modelos de Mistura Gaussiana

Modelos de Mistura Gaussiana (GMMs) são eficazes na análise de dados contínuos, tratando cada observação x_i como parte de uma composição de G componentes gaussianos. Cada componente é definido por um vetor médio μ_k, uma matriz de covariância Σ_k e um peso π_k. A função densidade de probabilidade é:

$$f(x_i; \Psi) = \sum_{k=1}^{G} \pi_k \varphi(x_i; \mu_k, \Sigma_k),$$

onde φ é a densidade gaussiana multivariada. Os clusters são elipsoidais, centrados nos vetores médios μ_k, com características geométricas determinadas pelas matrizes de covariância Σ_k. A decomposição das matrizes de covariância é: $\Sigma_k = \lambda_k D_k A_k D_k^\top$, onde λ_k modula o volume, A_k ajusta a forma e D_k configura a orientação do elipsoide.

A estimação dos parâmetros é realizada pelo algoritmo de Expectativa-Maximização (EM), que alterna entre calcular expectativas condicionais e maximizar a função de log-verossimilhança, promovendo uma convergência eficiente para as estimativas ótimas dos parâmetros. Como o número de parâmetros a serem estimados cresce rapidamente com o aumento do número de componentes da mistura e de variáveis associadas a cada observação, é comum utilizar restrições no modelo para controlar esse crescimento. Essas restrições vêm na forma de limitações das matrizes de covariância, que se traduzem em limitações na geometria de cada uma das componentes da mistura; as possíveis restrições são no volume, no formato e na orientação das componentes.

2.2 Algoritmo EM para Misturas Gaussianas

O Algoritmo EM ajusta Modelos de Mistura Gaussiana (GMMs) em dois passos principais. No passo E (Expectation), as probabilidades condicionais são calculadas como:

$$z_{ik} = \frac{\pi_k \varphi(x_i; \mu_k, \Sigma_k)}{\sum_{g=1}^{G} \pi_g \varphi(x_i; \mu_g, \Sigma_g)},$$

onde φ é a densidade gaussiana multivariada. No passo M (Maximization), os parâmetros da mistura são atualizados para maximizar a log-verossimilhança:

$$\pi_k = \frac{n_k}{n}, \quad \mu_k = \frac{\sum_{i=1}^{n} z_{ik} x_i}{n_k},$$

com $n_k = \sum_{i=1}^{n} z_{ik}$.

No pacote mclust do R, o algoritmo EM é inicializado usando partições obtidas a partir do agrupamento hierárquico aglomerativo baseado em modelos (MBAHC). Este processo inicia com k clusters, mesclando-os recursivamente para maximizar a verossimilhança, utilizando a verossimilhança de classificação Gaussiana como critério, conforme proposto por Banfield e Raftery (1993).

2.3 Clusterização no mclust

A função `Mclust()` no pacote mclust do R ajusta Modelos de Mistura Gaussiana (GMMs) para análise de agrupamento. Ela requer uma matriz numérica ou *data frame* com n observações e d variáveis, e permite especificar até $G = 9$ componentes de mistura. Como critério para escolher o número de componentes da mistura e as possíveis restrições do modelo, é possível testar várias combinações e determinar qual delas possui a maior

verossimilhança integrada. Uma aproximação dessa verossimilhança, que não é simples de calcular diretamente, é dada pelo Critério de Informação Bayesiana (BIC), que é calculado como:

$$BIC = 2 \log p(D \mid \hat{\theta}_{M_k}, M_k) - \nu_{M_k} \log(n),$$

onde ν_{M_k} é o número de parâmetros a serem estimados no modelo M_k, e $\log p(D \mid \hat{\theta}_{M_k}, M_k)$ é o logaritmo da verossimilhança do modelo ajustado aos dados D dados os parâmetros estimados $\hat{\theta}_{M_k}$. Este termo mede o quão bem o modelo ajusta os dados: valores maiores indicam um melhor ajuste.

Para uma melhor identificação de grupos, utiliza-se também o critério ICL (verossimilhança completa integrada de dados): $\text{ICL} = \text{BIC} + 2\sum_{i=1}^{n}\sum_{k=1}^{G} c_{ik} \log(z_{ik})$, onde z_{ik} é a probabilidade condicional de que x_i derive do componente k-ésimo, e c_{ik} é 1 se x_i estiver atribuído ao cluster k.

A incerteza associada à atribuição da i-ésima observação a um cluster é medida por: $\text{Uncer}_i = 1 - \max_{g=1,\ldots,G} \hat{z}_{i,g}$,onde Uncer_i será maior para pontos de dados com probabilidades $\hat{z}_{i,g}$ semelhantes e menor quando uma das probabilidades estiver próxima de 1. Se todos os clusters fossem igualmente plausíveis, $\text{Uncer}_i = 1/G$. Dessa forma, o gráfico de incerteza nada mais é que uma representação gráfica da incerteza de cada observação pertencer ao cluster em que foi alocada.

3 Resultados e Discussão

Como falado inicialmente, apliquei a Análise de Componentes Principais (PCA) para reduzir a dimensionalidade do conjunto de dados. A PCA transforma as variáveis originais em um conjunto de componentes principais, que são combinações lineares dessas variáveis. Os loadings representam os pesos dessas combinações lineares. Identifiquei as variáveis com as maiores cargas para os primeiros cinco componentes principais, conforme mostrado na Figura 1.

```
                PC1                 PC2                 PC3
"kills_per_round"   "kill_deaths"   "headshot_percentage"
                PC4                 PC5
"clutch_success_percentage"   "rating"
```

Figura 1: Variáveis com as maiores cargas para os primeiros cinco componentes principais.

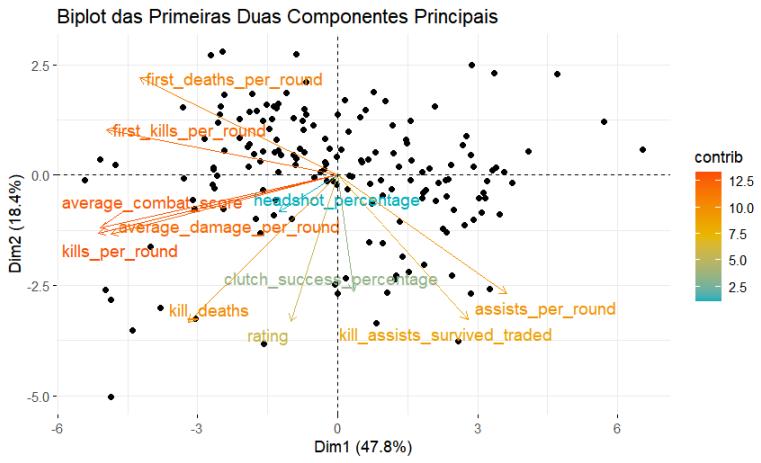
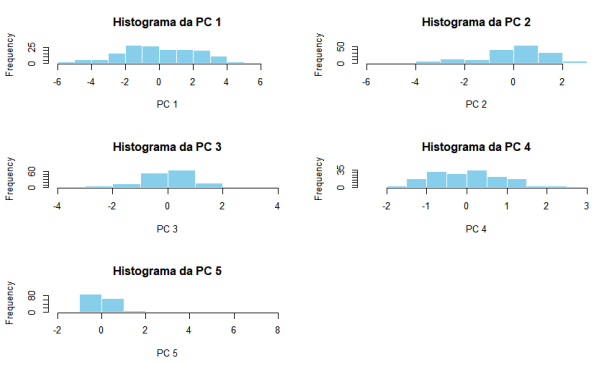


Figura 2: Biplot das duas primeiras componentes principais.

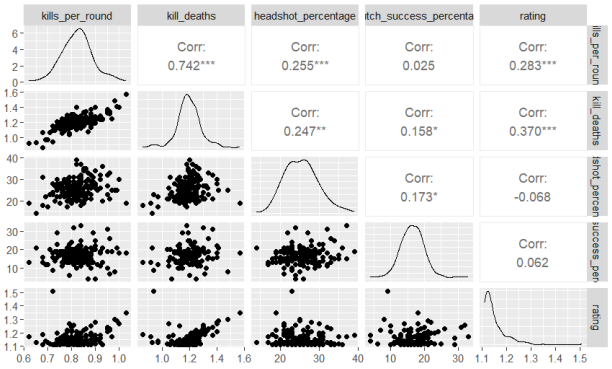
Na figura 2, podemos observar que as cinco variáveis escolhidas têm uma contribuição significativa. As duas primeiras componentes principais explicam 47,8% e 18,4% da variância total dos dados, respectivamente.

Uma análise descritiva foi realizada pela figura 3 (a) e 3 (b). Na figura 3 (a), podemos observar que o histograma da PC1,PC3,PC4 estão centrados no zero, o que faz sentido pois são variáveis que variam bastante.

E as outras variaveis tem um deslocamento. No mais, analisando a Figura 3 (b) especificamente, nota-se que há a formação, em todos os scatterplots, de pelo menos um cluster bem definido. Além disso, todos aparentam ser representados por uma distribuição normal, satisfazendo o requisito do GMM.



(a) Histograma das cinco primeiras componentes



(b) ggpairs() das variáveis com as maiores cargas

Figura 3: Análises das Componentes Principais

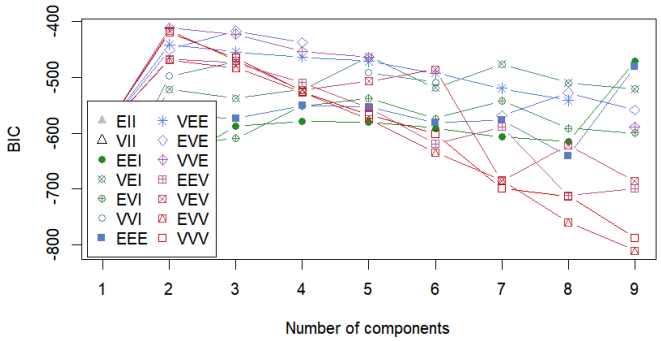


Figura 4: Plot do BIC com zoom.

```
Top 3 models based on the BIC criterion:
    VVE,2     VEV,2     EVE,3
-411.5780 -416.6202 -417.0060
```

Figura 5: O top 3 valores de BIC

```
Top 3 models based on the ICL criterion:
    EVE,3     VVE,2     VEV,2
-423.3228 -429.4328 -432.4821
```

Figura 6: O top 3 valores de ICL

Através das Figuras 4 e 5, podemos observar que o melhor modelo baseado no BIC é aquele com 2 clusters e estrutura VVE (distribuição elipsoidal, volume variável, forma variável e orientação do eixo igual). Por outro lado, pela figura 7, o melhor modelo baseado no ICL é aquele com 3 clusters e estrutura EVE (distribuição elipsoidal, volume igual, forma variável e orientação do eixo igual).

Para verificar a adequação desses modelos, realizei a plotagem dos gráficos de classificação:

```
modelo1 <- Mclust(dados_selecionados, 2, modelNames = "VVE")
modelo2 <- Mclust(dados_selecionados, 3, modelNames = "EVE")
plot(modelo1, what = "classification")
plot(modelo2, what = "classification")
```

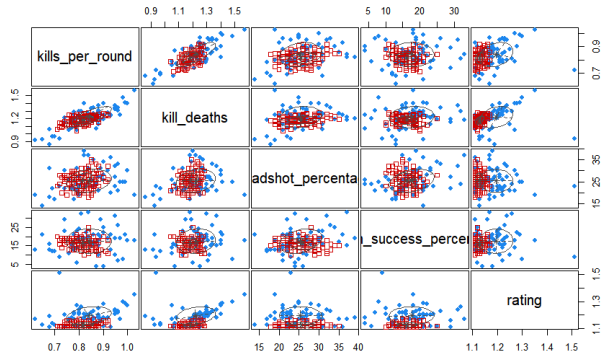
Após a análise dos gráficos, concluí que o modelo com 2 clusters e estrutura VVE (modelo1) apresenta um melhor ajuste aos dados. Portanto, considerando $C = 2$ clusters e $p = 5$ variáveis no espaço dos dados, isso implica que o número total de parâmetros a ser estimado é 20.

Na figura 7 (a) fornece uma representação gráfica do resultado da clusterização por GMM. Observa-se que alguns clusters estão sobrepostos, o que prejudica a interpretação clara dos grupos formados.

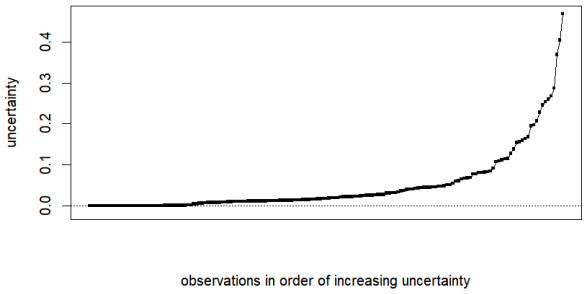
Analisando o resultado da clusterização GMM pelo gráfico da incerteza, Figura 7 (b), verifica-se que, para mais da metade das observações, há uma baixa incerteza em sua classificação (menor que 0.1). No entanto, para o restante das observações, a incerteza associada aumenta linearmente de 0.1 a 0.5.

O Gráfico 7 (c) apresenta a soma de quadrados interna (WSS) por número de clusters, utilizada para determinar o número ideal de grupos para o método k-medoids. Empregando o método Elbow, concluímos que 4 é o número ideal de clusters para o k-medoids. Como observado, o número de clusters identificado pelo k-medoids difere do número de clusters obtido pelo GMM.

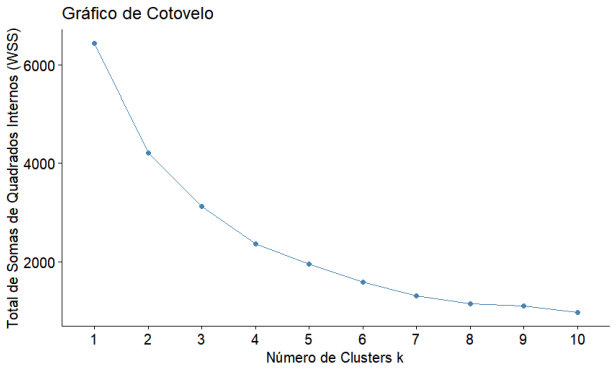
Adicionalmente, o resultado do método k-medoids pode ser verificado pelo gráfico de silhueta na Figura 7 (d). Com uma silhueta média de 0.32, podemos afirmar que a clusterização foi satisfatória.



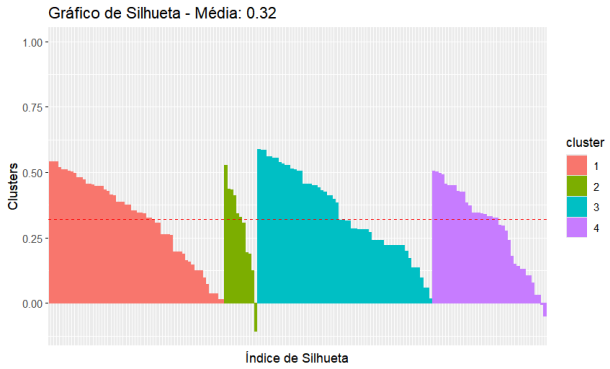
(a) Modelo Mclust EVE com 2 clusters



(b) Gráfico de Incerteza do modelo



(c) Gráfico de Cotovelo do K-means



(d) Gráfico de Silhueta K-means

Figura 7: Análises dos Modelos de Clusterização

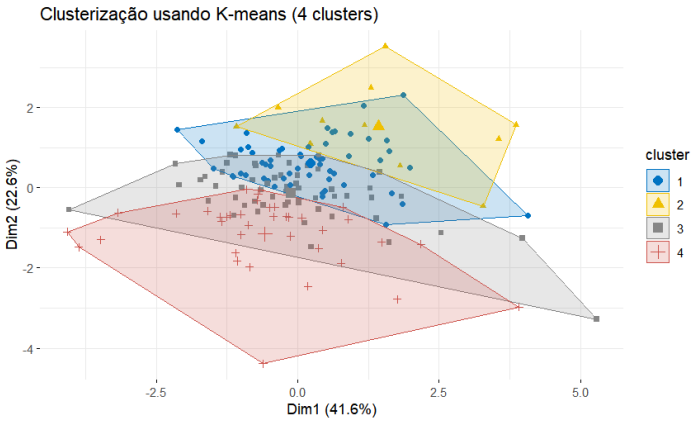


Figura 8: Clusterização usando K-means

4 Conclusão

Apresentados os resultados, a primeira conclusão é que os dois métodos de clusterização produziram resultados diferentes, mas ambos apresentaram o problema de sobreposição dos clusters. Isso indica que, possivelmente, não há uma distinção clara de qual classe no jogo Valorant é mais impactante do que as outras.

Esperava-se uma alta performance da classe Duelista, pois esses personagens, especializados em combate direto, são fundamentais para iniciar confrontos, penetrar defesas adversárias e realizar eliminações críticas. Teoricamente, eles deveriam ter o melhor desempenho dentro do jogo, mas não foi possível chegar a essa conclusão. Porém, pude concluir que todas as classes são fundamentais para o jogo e que o desempenho depende muito mais da habilidade do jogador do que da classe escolhida.

Além disso, considerando a quantidade razoável de observações com alta incerteza e o valor da silhueta média dos clusters, é importante ressaltar que, embora os resultados dos dois métodos tenham sido satisfatórios, eles não foram excelentes. Ademais, trabalhei apenas com uma pequena amostra. Portanto, não podemos descartar a possibilidade de que a divisão esperada possa se revelar com o uso de outras técnicas ou um tamanho maior de observações.

Por fim, dada a preferência por uma abordagem mais simples e rápida, o k-means mostrou-se suficiente para mim.

Referências

[1] Universidade Estadual de Campinas. *Notas de Aula de ME921, G. Ludwig*. Disponível em: https://moodle.ggte.unicamp.br/pluginfile.php/3926125/mod_resource/content/1/aula15.pdf.

[2] C. Bouveyron, G. Celeux, T. Brendan Murphy e A. E. Raftery. *Model-Based Clustering and Classification for Data Science*. Cambridge University Press, 2019.

[3] B. S. Everitt, S. Landau, M. Leese e D. Stahl. *Cluster Analysis*, 5ª edição. John Wiley e Sons, 2011.

[4] *Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models*. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5096736/#FN2.