

Relatório – Projeto Final de Previsão de Resultados em Partidas de League of Legends (LoL)

Projeto apresentado à EBAC e à Tech Lead da Riot Games como entrega final do curso de Ciência de Dados.

INTRODUÇÃO

O ecossistema competitivo de League of Legends movimenta milhões de jogadores e uma enorme infraestrutura de análise tática. Cada decisão tomada nos primeiros minutos de jogo, como a conquista de objetivos ou a vantagem em ouro, pode alterar drasticamente o rumo da partida. Inspirado nesse contexto, este projeto buscou desenvolver um modelo de Machine Learning capaz de prever o vencedor de uma partida com base em métricas pós-início de jogo.

O objetivo central foi entregar uma solução interpretável, robusta e replicável, alinhada à visão estratégica da Riot Games para análises de comportamento competitivo e suporte a ferramentas internas de tomada de decisão.

ENTENDIMENTO DO PROBLEMA

A Tech Lead da Riot Games descreveu a necessidade de um modelo confiável que, logo nos primeiros minutos, consiga inferir qual time possui maior probabilidade de vencer. Isso permitiria melhorar sistemas de balanceamento, matchmaking e até ferramentas de revisão e coaching aplicadas a equipes profissionais.

O dataset disponibilizado contém métricas essenciais como:

- Abates, Assistências e Mortes;
- Objetivos (Torres, Dragões, Arautos, Monstros Épicos);
- Wards posicionadas e destruídas;
- Ouro total;
- Sinalizadores de eventos (ex.: First Blood).

A variável alvo é blueWins (1 = vitória do time azul).

PREPARAÇÃO DOS DADOS

Seguindo boas práticas de engenharia de atributos, foi construído um pipeline determinístico que:

1. Calcula diferenças numéricas entre os times (ex.: kills_diff = blueKills – redKills).
2. Gera o indicador first_blood_flag.
3. Remove todas as variáveis originais redundantes (blueX / redX) após a engenharia.
4. Mantém apenas colunas numéricas e relevantes para o modelo.

Os artefatos de EDA incluem:

- Estatísticas descritivas (data_describe.csv);
- Distribuição do target (target_distribution.csv);
- Relatório de missing values (missing_values.csv).

MODELAGEM

Três algoritmos foram treinados e validados com cross-validation:

- Logistic Regression;
- Gaussian Naive Bayes;
- Decision Tree.

A validação cruzada utilizou StratifiedKFold com 5 folds e a métrica primária foi AUC-ROC.

RESULTADOS

Resumo das métricas:

- Logistic Regression
 - Test AUC: ~0.8047
 - Test Accuracy: ~0.7161

	Pred. Neg.	Pred. Pos.
True Neg.	704	286
True Pos.	275	711

- Gaussian Naive Bayes
 - Test AUC: ~0.7915
 - Test Accuracy: ~0.7085

	Pred. Neg.	Pred. Pos.
True Neg.	711	279
True Pos.	297	689

- Decision Tree
 - Test AUC: ~0.6432
 - Test Accuracy: ~0.6432

	Pred. Neg.	Pred. Pos.
True Neg.	649	341
True Pos.	364	622

A regressão logística demonstrou o melhor desempenho, com equilíbrio entre interpretabilidade, velocidade de inferência e estabilidade estatística. Isso a torna a escolha ideal para cenários de produção.

INTERPRETAÇÃO DOS RESULTADOS

O modelo de regressão logística aponta que as diferenças entre os times em métricas de objetivos (torres, dragões, arautos) e vantagem de ouro são os fatores com maior poder discriminativo. Eventos precoces como First Blood possuem menor impacto isolado, mas reforçam tendências quando combinados com outras métricas.

Essa interpretação reforça achados já conhecidos por analistas competitivos: controle de objetivos maiores e macrojogo consistente têm impacto mais previsível no desfecho de partidas.

ENTREGA FINAL

O pacote entregue inclui:

- Modelos treinados em formato joblib;
- Pipelines de inferência (infer_logistic.py, infer_nb.py, infer_tree.py);
- Arquivos de predição (predictions_*.csv);
- Relatórios de matriz de confusão;
- Artefatos de EDA.

CONCLUSÃO

O projeto atingiu seu objetivo ao entregar um modelo sólido e aplicável ao contexto competitivo de League of Legends. A abordagem adotada buscou balancear precisão, explicabilidade e facilidade de manutenção, garantindo que o produto analítico seja útil tanto para analistas quanto para integração em plataformas internas.

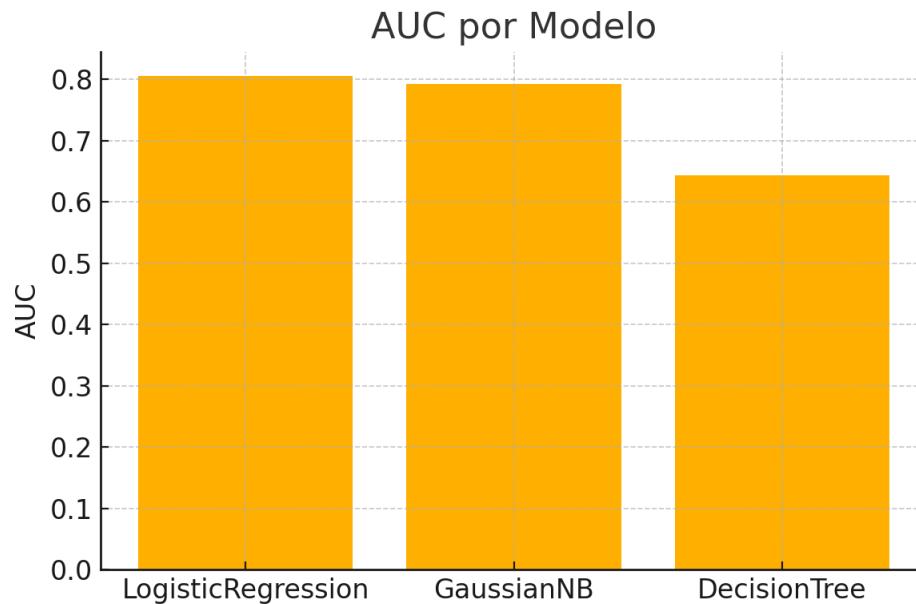
Próximos passos sugeridos:

- Incluir dados temporais por minuto;
- Testar modelos baseados em árvores de gradiente (LightGBM, XGBoost);
- Criar dashboards interativos para equipes competitivas.

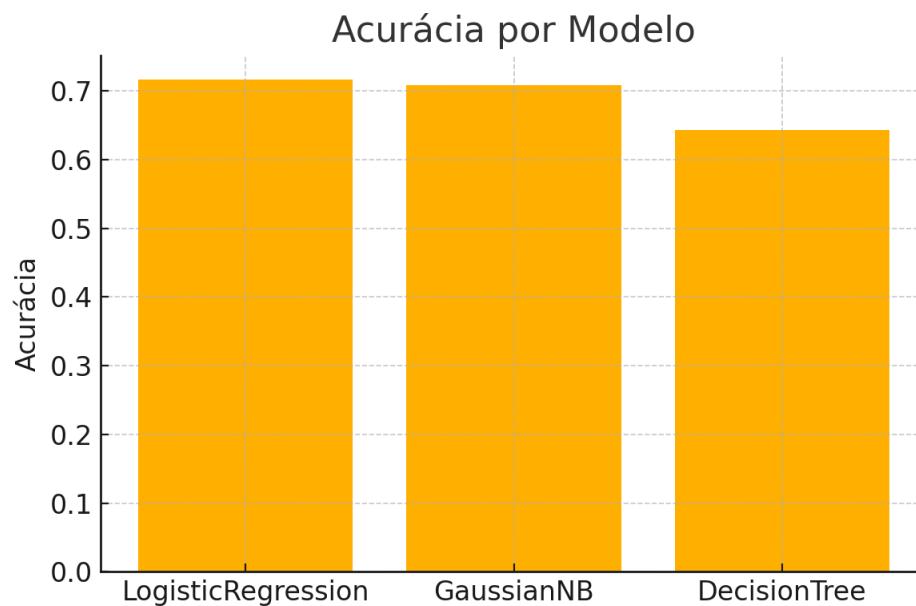
Agradeço à Riot Games e à EBAC pela oportunidade, pela disponibilização da base de dados, e espero que este modelo contribua para análises futuras e potencialize ainda mais o cenário competitivo. Como jogador, foi muito divertido e gratificante conduzir este projeto.

Anexos Complementares

AUC por Modelo:



Acurácia por Modelo:



Curvas ROC por Modelo:

