

**FAESA CENTRO UNIVERSITÁRIO**  
**CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**GUILHERME NOGUEIRA DESSAUNE DE OLIVEIRA**  
**LUANA AMY NAKASUGA**  
**LUIZ GUILHERME MACHADO ZORTÉA**  
**MATHEUS SILVA HERCULINO**  
**RENATO ARCHANJO RABELLO**

**ANÁLISE DO ALGORITMO DE REGRESSÃO**  
**LOGÍSTICA EM DADOS DO “CHALLENGER” DO ANO**  
**DE 2020 DE LEAGUE OF LEGENDS NO SERVIDOR SUL**  
**COREANO**

**VITÓRIA**

**2023**

**GUILHERME NOGUEIRA DESSAUNE DE OLIVEIRA**

**LUANA AMY NAKASUGA**

**LUIZ GUILHERME MACHADO ZORTÉA**

**MATHEUS SILVA HERCULINO**

**RENATO ARCHANJO RABELLO**

## **GERENCIAMENTO DE ANIMAIS**

Trabalho apresentado a disciplina Projeto de Computação Integrada III, dos cursos de Ciência da Computação, Sistemas de Informação e Tecnologia em Análise e Desenvolvimento de Sistemas, sob orientação do prof. **Howard Cruz Roatti**.

**VÍTORIA**

**2023**

## **SELEÇÃO DO MODELO ESCOLHIDO E JUSTIFICATIVA PARA AS ESCOLHAS**

Foi selecionado o modelo de regressão linear por estarmos lidando com uma variável dependente binária, que é Win ou Loss (Vitória ou Derrota). Assim acontece a modelagem direta da probabilidade de vitória baseando nos diferentes atributos apresentados.

## **BREVE EXPLICAÇÃO SOBRE O FUNCIONAMENTO DO MODELO ESCOLHIDO**

Regressão Logística é um modelo de classificação que também faz previsões ou estimativas de um evento acontecer. Usado quando a variável de resposta é binária/categórica, ou seja, apresenta apenas dois resultados possíveis, como 'sim' ou não', 0 ou 1. Durante o processo de aprendizado, os parâmetros do modelo são adaptados para otimizar a compatibilidade com os dados existentes. Após o treinamento, o modelo pode ser aplicado para prever a classificação de novos dados, com base em um limite de probabilidade estabelecido.

## **DESENVOLVIMENTO DO MODELO**

Nosso código realiza uma análise de dados e modelagem usando regressão logística. Primeiramente importamos as bibliotecas SQLAlchemy, NumPy, Pandas, Matplotlib, e os módulos para pré-processamento e avaliação de modelos do Scikit-learn.

É feita uma conexão a um banco de dados Postgres para obter os dados a serem tratados e analisados.

Os dados dos DataFrames são combinados através da mesclagem de acordo com as colunas 'gameId' e 'teamId', resultando em um novo DataFrame chamado df\_combined. Informações sobre as colunas e tipos de dados dos DataFrames são exibidas utilizando o atributo info().

Uma nova coluna chamada 'win' é adicionada ao DataFrame df\_combined e inicialmente preenchida com zeros. Os valores da coluna 'win' são atualizados para 1 nos índices correspondentes onde as colunas 'teamId' e 'gameId' do DataFrame df\_combined coincidem com as do DataFrame dim\_ganhador.

Algumas colunas categóricas do DataFrame df\_combined são codificadas utilizando LabelEncoder, transformando as variáveis categóricas em valores numéricos. Essas colunas são 'gameId', 'firstBlood', 'firstTower', 'firstInhibitor', 'firstBaron', 'firstDragon' e 'firstRiftHerald'.

As colunas do DataFrame df\_combined são convertidas para o tipo string para evitar possíveis problemas durante a análise.

Os dados são divididos em conjuntos de treinamento e teste, sendo 70% dos dados usados para treinamento e 30% para teste. As variáveis de destino são

separadas em `target_train` e `target_test` correspondentes aos dados de treinamento e teste, respectivamente.

É criado um modelo de regressão logística com base no algoritmo `LogisticRegression` do Scikit-learn. O modelo é treinado utilizando os dados de treinamento. Em seguida, são feitas previsões utilizando os dados de teste.

A matriz de confusão é calculada utilizando a função `confusion_matrix` e exibida na forma de uma matriz. Em seguida, é plotado um gráfico da matriz de confusão utilizando a biblioteca `Matplotlib`.

Diversas métricas de avaliação do modelo são calculadas, incluindo acurácia, precisão, revocação e medida F1. Essas métricas fornecem informações sobre o desempenho do modelo.

Por fim, é plotada a curva ROC (Receiver Operating Characteristic), que representa a taxa de verdadeiros positivos em função da taxa de falsos positivos. A área sob a curva ROC (AUC) também é calculada, fornecendo uma medida da qualidade geral do modelo.

## CONCLUSÃO

O objetivo do nosso modelo foi alcançado, porém, apenas em um nível superficial. Apesar dos resultados apresentados, ele não se aplica adequadamente à realidade de uma partida dentro do jogo.

Em primeiro lugar, é importante ressaltar que o jogo é dinâmico, mas o modelo foi treinado considerando a partida como um fato único. Isso limita sua capacidade de análise e previsão.

Além disso, o modelo se baseia exclusivamente em dados públicos coletados durante o ano de 2020. No entanto, devido a todas as atualizações que ocorreram desde então, esses dados se tornam irrelevantes para as partidas atuais.

Para tornarmos o modelo mais relevante, podemos fazer algumas adições:

- Utilizar os dados para identificar a probabilidade de uma partida ser uma vitória ou derrota na perspectiva de cada jogador.

- Atualizar a base de dados do programa sempre que um elemento relevante for adicionado ou removido do jogo.

- Considerar cada evento em uma partida como um dado individual, permitindo que sejam analisados com base nesses eventos.