Disciplina: Projeto e Análise de Experimentos – CSI 710

Aluno: Israel Matias do Amaral
Tema: Análise de discurso de ódio na comunidade gamer do YouTube com o uso de modelos de linguagem

---

Projeto integrado com o Trabalho de Conclusão de Curso I.

De complexidade alta, pois vai envolver:

- Modelos de aprendizado de máquina e LLMs (ex.: BERT e ChatGPT)
- Experimentação comparativa com projeto fatorial
- Domínio de ferramentas computacionais e APIs

---

1. Contextualização e definição clara do problema

A popularização das transmissões ao vivo no YouTube, especialmente entre streamers da comunidade gamer, tem sido acompanhada pelo aumento de comentários ofensivos nos chats dessas lives. Muitos desses comentários são expressos por meio de ironias e expressões codificadas, o que dificulta sua detecção automática.

O projeto foca em transmissões ao vivo realizadas por uma **subcomunidade gamer do YouTube**, composta por streamers como *luangameplay* e *renanplay*, conhecidos por adotar um estilo de humor mais ácido e politicamente incorreto. Essa comunidade apresenta alto engajamento — alguns canais chegam a mais de 1,4 milhões de inscritos e mantêm cerca de 2 mil espectadores por transmissão.

A escolha desse grupo se justifica por combinar 2 elementos críticos para o estudo: (1) um **volume elevado de interações** (fator engajamento) e (2) uma **alta incidência de linguagem ambígua ou ofensiva**, o que o torna um ambiente ideal para avaliar a eficácia de modelos de detecção de discurso de ódio.

---

2. Objetivos do estudo

- Aplicar e comparar modelos de linguagem para detectar discurso de ódio em português, em comentários extraídos de chats ao vivo.
- Avaliar o impacto do pré-processamento textual no desempenho dos classificadores.
- Analisar a ocorrência de linguagem velada, ambígua ou irônica em comentários ofensivos.

- Relacionar padrões de toxicidade ao engajamento e perfil dos canais.

---

3. Fonte dos dados e descrição das variáveis

Fonte dos dados:
Comentários extraídos de chats ao vivo de transmissões no YouTube, utilizando exclusivamente a API oficial do YouTube, em conformidade com os termos da plataforma. As coletas incluem metadados da transmissão e mensagens dos espectadores.

As variáveis foram extraídas de dois arquivos distintos:

- **chat.csv:** contém os comentários dos espectadores com autor, timestamp e conteúdo textual.
- **metadados.csv:** inclui informações do vídeo como título, canal, datas (publicação e início da live), e métricas agregadas como número de visualizações, curtidas e comentários.

Classificação geral das variáveis:

- **Categóricas nominais:** *autor, canal, id_video*
- **Quantitativas contínuas:** *timestamp*
- **Quantitativas discretas:** *espectadores_atuais, likes, visualizacoes, comentarios*
- **Variáveis não estruturadas (texto livre):** *mensagem, título, descrição*

---

4. Cenários e ambiente de experimentação (ferramentas, plataformas e recursos computacionais)

Plataforma:
O projeto será conduzido com código aberto e documentado no repositório público: https://github.com/imdoamaral/TCC-1

Recursos computacionais:

- Sistema operacional: Zorin OS 17.3, CPU: AMD Ryzen 5 5600, Memória: 16GB RAM, GPU: NVIDIA GTX 1060 6GB.
- **Execuções preferencialmente locais.** O uso de APIs (ex: OpenAI) será considerado apenas se necessário, devido ao custo.

- **O Google Colab está fora do escopo inicial**, podendo ser utilizado somente em caso de limitação de hardware.

Ferramentas e modelos utilizados:

- **Python** com bibliotecas como Pandas e Matplotlib: para manipulação, análise e visualização dos dados.
- **YouTube Data API v3**: para coleta legal dos dados de chat e metadados das transmissões.
- **BERTimbau**: modelo pré-treinado para a língua portuguesa, utilizado como classificador de toxicidade.
- **Classificadores baseados em LLM**, como o ChatGPT (GPT-3.5 Turbo): avaliado por Oliveira et al. (2023) como competitivo mesmo sem fine-tuning

---

5. Técnicas que se pretende utilizar (exploratória + técnica principal)

Técnica exploratória: Será conduzida com o objetivo de compreender padrões de comportamento nos chats ao vivo, antes da aplicação de modelos de linguagem. As análises exploratórias ajudarão a formular hipóteses, identificar possíveis vieses e estruturar os dados para testes posteriores.

Exemplos de análises previstas:

- **Distribuição da frequência de mensagens por streamer e por transmissão:** Identificar canais com maior volume de interações e potenciais outliers usando histogramas e boxplots.
- **Análise de volume de mensagens por minuto:** Observar picos de atividade no chat, que podem indicar momentos polêmicos ou mais engajados da live.
- **Tamanho médio das mensagens ao longo da live:** Hipótese de que mensagens mais curtas podem indicar maior reatividade ou agressividade.
- **Análise do vocabulário (palavras mais frequentes):** Explorar termos, emojis e gírias comuns por canal ou transmissão. Pode incluir visualizações como nuvem de palavras.
- **Distribuição de mensagens por usuário:** Analisar concentração de mensagens em poucos autores, o que pode evidenciar "superusuários" ou desequilíbrio na participação.
- **Comparação entre canais grandes e pequenos:** Verificar diferenças em métricas como número médio de mensagens, engajamento e dispersão temporal.

- **Comparação entre streamers homens e mulheres:** Observar se há diferenças no volume de mensagens ou padrões de linguagem, mesmo sem classificar como tóxicas.
- **Distribuição estatística das variáveis numéricas:** Aplicar medidas como média, mediana, desvio padrão, skewness e curtose para variáveis como timestamp, espectadores simultâneos e tamanho das mensagens.
- **Verificação de distribuição teórica dos dados:** Usar histogramas, KDEs ou testes formais para verificar se variáveis seguem distribuições conhecidas (normal, exponencial, etc.).

Essas análises são fundamentais para descrever o comportamento dos dados, levantar hipóteses e preparar o terreno para a aplicação do projeto fatorial e dos classificadores.

Técnica principal:
Projeto Fatorial 2^2 com *r* replicações, testando dois fatores manipuláveis:

Fator	**Níveis**
modelo_classificador	BERT vs. LLM
tipo_preprocessamento	Sem vs. com tratamento

- **Com tratamento:** remoção de emojis, stopwords, usuários, URLs, padronização de caixa e lematização.
- **Sem tratamento:** uso direto do texto original da mensagem.

Variável resposta (o que vai ser medido):

- **Opção contínua:** probabilidade de toxicidade (entre 0 e 1), retornada pelos classificadores.
- **Opção binária:** rótulo direto (tóxico ou não tóxico).

A escolha será decidida posteriormente com base nos dados e objetivos da modelagem.

---

6. Referências das bases de dados, métodos, entre outros

Referências de métodos e modelos:

- Souza, F. et al. (2020). *BERTimbau: Pretrained BERT Models for Brazilian Portuguese*.
- Oliveira, A. S. et al. (2023). *How Good Is ChatGPT for Detecting Hate Speech in Portuguese?*
- Assis, R. A. et al. (2024). *Exploring Portuguese Hate Speech Detection in Low-Resource Settings*.

Reflexões sobre o projeto:

- **O que se pretende aprender?**
  Visualizar como diferentes modelos de linguagem avaliam a toxicidade em comentários e como escolhas técnicas afetam a performance.
- **Por que esse problema é importante?**
  Porque a moderação automatizada ainda apresenta limitações.
- **Como a análise de experimentos pode ajudar?**
  Tomar decisões baseadas em dados sobre qual modelo e tratamento textual são mais eficazes.