**Descrição do Projeto**

A Film Junky Union, uma nova comunidade para entusiastas de filmes clássicos, está desenvolvendo um sistema para filtrar e categorizar resenhas de filmes. O objetivo é treinar um modelo para detectar automaticamente resenhas negativas. Você usará um conjunto de dados de resenhas de filmes do IMDB com rotulagem de polaridade para criar um modelo para classificar resenhas como positivas e negativas. Ele precisará atingir um valor F1 de pelo menos 0,85.

**Instruções do Projeto**

1. Carregue os dados.
2. Pré-processe os dados, se necessário.
3. Conduza uma AED e tire sua conclusão sobre o desequilíbrio de classe.
4. Pré-processe os dados para modelagem.
5. Treine pelo menos três modelos diferentes para o conjunto de dados de treinamento fornecido.
6. Teste os modelos para o conjunto de dados de teste fornecido.
7. Escreva algumas resenhas você mesmo e classifique-as usando todos os modelos.
8. Verifique as diferenças entre os resultados dos testes dos modelos nos dois pontos acima. Tente explicá-las.
9. Apresente suas descobertas.

Importante! O projeto exemplo já contém alguns trechos de código para sua conveniência, para que você possa usá-los se desejar. No entanto, se você quiser começar do zero, basta remover esses trechos de código.. Aqui está a lista de trechos de código:

* Um pouco de AED com alguns gráficos
* evaluate\_model()
  + uma rotina para avaliar um modelo de classificação que está em conformidade com a interface de predição do scikit-learn
* BERT\_text\_to\_embeddings()
  + uma rotina para converter uma lista de textos em incorporação com BERT

O trabalho principal para construir e avaliar modelos é seu.

Como você pode ver no projeto exemplo, sugerimos tentar modelos de classificação baseados em regressão logística e gradient boosting, mas sinta-se à vontade para tentar outros métodos. Você pode mexer na estrutura do projeto exemplo desde que as instruções do projeto sejam concluídas.

Você não precisa usar BERT para o projeto porque é muito exigente em poder computacional e será muito lento na CPU para o conjunto de dados completo. Por causa disso, o BERT geralmente precisa ser executado na GPU para um desempenho adequado. No entanto, sinta-se à vontade para tentar incluir o BERT no projeto para uma parte do conjunto de dados. Se você quiser fazer isso, sugerimos fazê-lo localmente e pegar apenas algumas centenas de objetos para cada parte do conjunto de dados (treinamento/teste) para evitar esperar muito. Certifique-se de indicar o uso do BERT na primeira célula (o cabeçalho do seu projeto).

**Descrição dos Dados**

Os dados são armazenados no arquivo imdb\_reviews.tsv. [Baixe o conjunto de dados](https://practicum-content.s3.us-west-1.amazonaws.com/datasets/imdb_reviews.tsv).

*Os dados foram fornecidos por Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng e Christopher Potts. (2011). Vetores de aprendizado de palavras para análise de sentimentos. A 49ª Reunião Anual da Association for Computational Linguistics (ACL 2011).*

Aqui está a descrição dos campos selecionados:

* review: o texto da resenha
* pos: o objetivo, '0' para negativo e '1' para positivo
* ds\_part: 'train'/'test' para a parte de treinamento/teste do conjunto de dados, respectivamente

Existem outros campos no conjunto de dados. Sinta-se à vontade para explorá-los se quiser.