Compilado de decisões e resultados para a criação dos modelos de ML

1. Definição dos marcadores (labels)

**“Esta é uma das grandes dificuldades do aprendizado de máquina especialmente no que diz respeito à análise de sentimentos por dois motivos principais: a alta subjetividade envolvida na tarefa e a demanda de tempo necessária para que especialistas definam a polaridade de muitas sentenças.”**

Nessa etapa foram escolhidos 2 dicionários para avaliar qual se ajustaria melhor aos dados existentes.

* **Vader:** Foi criado com base em outros 3 dicionários diferentes (LIWC, ANEW e o GI)
* **Opinion-lexic:** Diferente do anterior por estar mais atualizado a lidar com expressões em redes sociais e reviews de produtos.

O opinion-lexic foi escolhido tanto por ter apresentado uma sensibilidade maior em sua classificação, quanto pelo fato de sua origem ter vinda de reviews de produtos, colaborando para o objetivo desse projeto que envolve classificar reviews de filmes.

A group of blue and orange bars

Description automatically generated

Nas imagens acima os valores **Standard** representam classificações feitas apenas com a remoção básica das palavras mais comuns na língua inglesa (stop words), já o **Custom** contém algumas palavras a mais que aparecem com muita frequência nos textos e não representam uma relevância significativa na classificação das polaridades.

A close-up of words

Description automatically generatedA close up of words

Description automatically generated

1. Modelos de aprendizado de máquina

O intuito do projeto é realizar uma comparação entre 2 modelos de aprendizado de máquina e verificar se existe alguma melhora. Os modelos escolhidos foram o **Na**ï**ve Bayes (NB)** e o **Bidirectional Encoder Representations from Transformers (BERT)**, sendo um mais voltado ao ML clássico e o outro para o Deep Learning (aprendizado profundo). A seguir apresento em uma estrutura de tópicos as cronologias de decisão tomadas para a criação de cada modelo, assim como os resultados obtidos.

* 1. Balanceamento de classes

Devido a um desbalanceamento de classes, um método de resampling foi aplicado para evitar que os modelos aprendam muito mais de uma do que outra classe, já que quando são apresentados menos exemplos de uma determinada classe x, o modelo tender a não capturar tão bem os padrões desejados.

* 1. Divisão dos dados (Split)

A regra de Pareto foi pensada para poder dividir os dados em treino (80%) e teste (20%)

* 1. Estado aleatório (Random state)

Um parâmetro de aleatoriedade foi fixado, para que seja garantida a reprodutibilidade dos treinos e testes.

* 1. Particularidades do NB

1. Vetorização: Um processo onde se converte palavras em números para o modelo consiga atuar na predição.
   1. Particularidades do BERT
2. Foi utilizado um modelo já construído pela empresa [Hugging Face](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjfqJfF-8ODAxUZppUCHYSTAs0QFnoECAUQAQ&url=https%3A%2F%2Fhuggingface.co%2F&usg=AOvVaw2MwohX3MqsEeuiTlqUr-jn&opi=89978449) 🤗🤗🤗.
3. Necessário ter sido feito o retreino dele, devido a sua construção padrão estar voltada para uma classificação binária (o que não é o caso desse trabalho, pois temos 3 classes).
4. Foram adicionado um otimizador ADAM (mais amplamente utilizado) e um critério de medição de função de perda (Cross Entropy Loss -> Também adaptado para multi-classes).
5. 3 épocas foram escolhidas por conta da demora no treinamento, mas devido a métrica de média de perda, é possível verificar que poderíamos ter mais treinamentos pois o modelo ainda estava mostrando um potencial aprendizado, levando em consideração que ela estava sendo reduzida por 50% a cada época.
6. Resultados

Acurácias: NB = 0.81 ------- BERT = 0.89

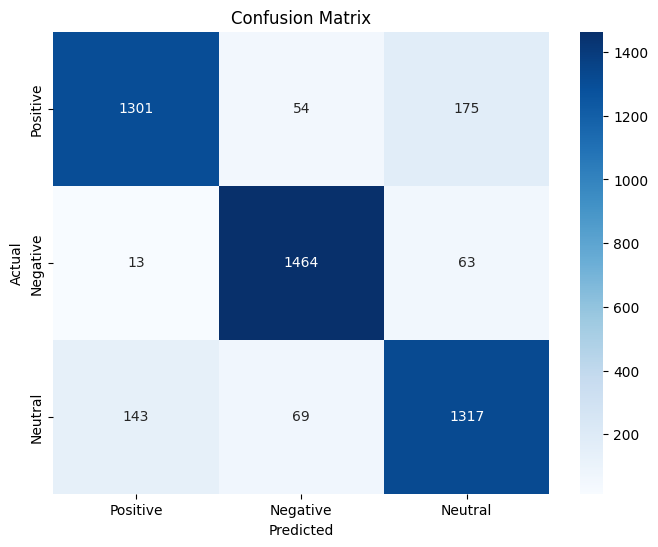


Figura 2 - BERT

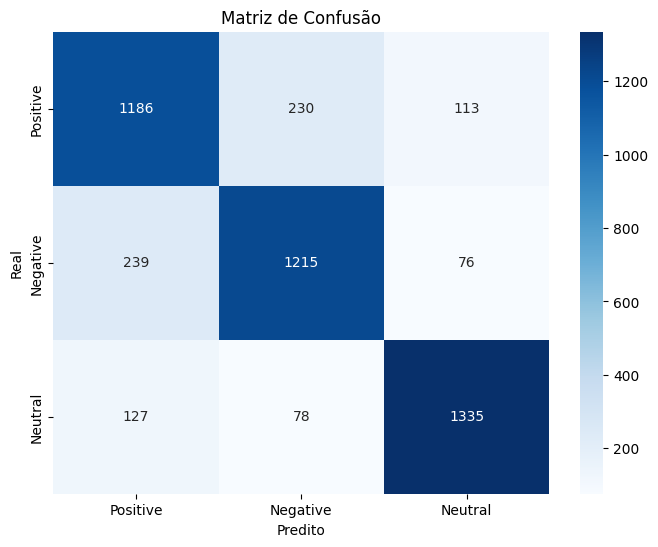


Figura 1 - NB

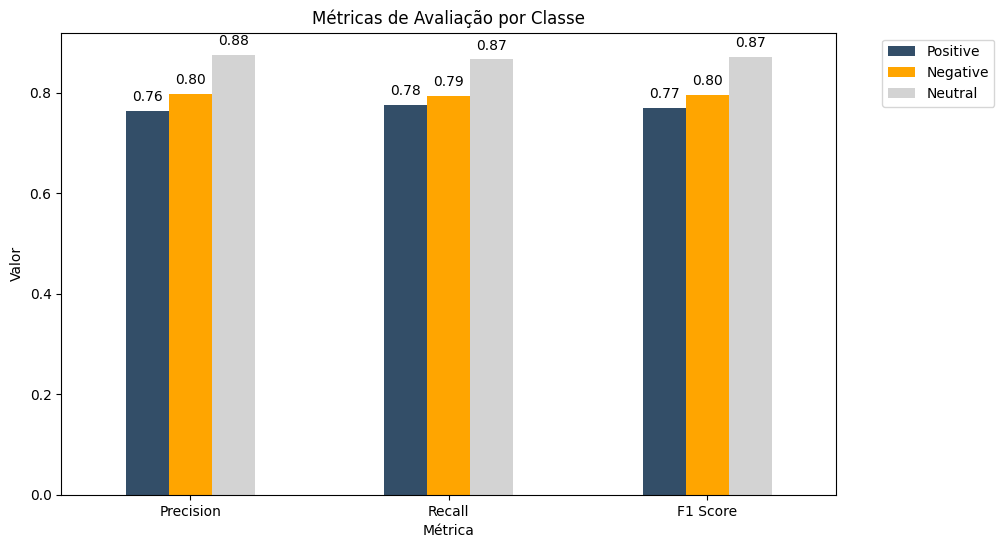


Figura 3 - NB

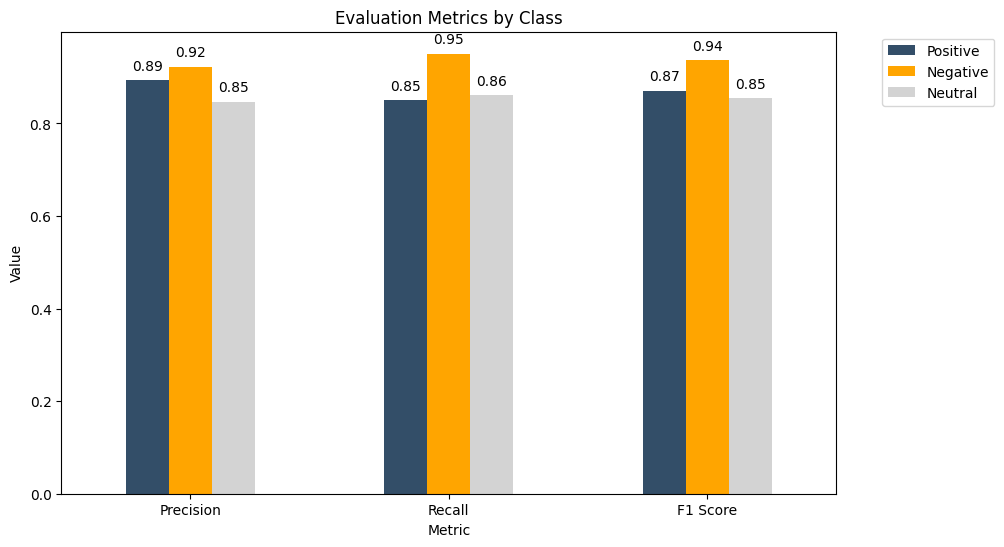


Figura 4 - BERT