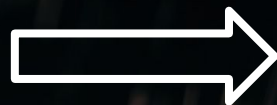
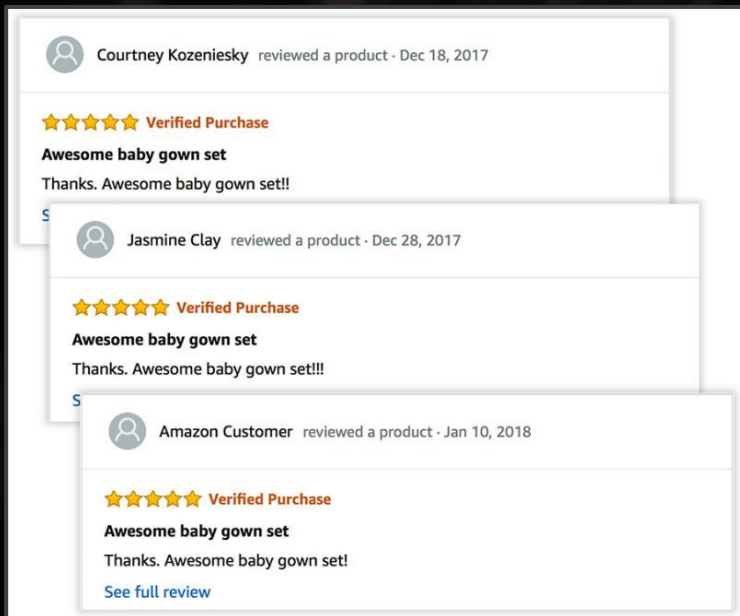


# Análise de Sentimentos com Redes Neurais (PMR3508)

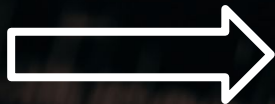
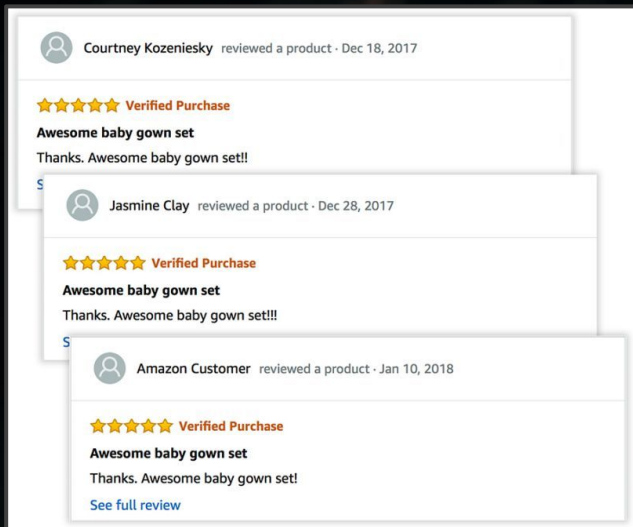
Felipe Maia Polo  
Fabio Cozman

# Análise de Sentimentos



Sentimento  
positivo ou  
negativo?

# Nossa Atividade

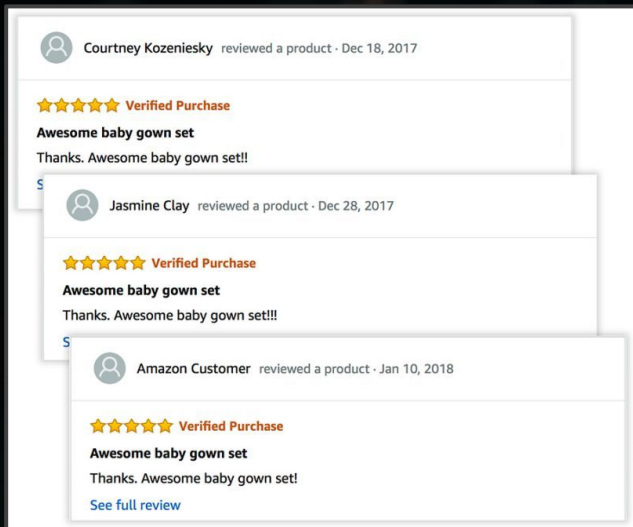


$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$



Sentimento **positivo**  
ou **negativo**?

# Nossa Atividade



$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$



Sentimento **positivo**  
ou **negativo**?

# Introdução

- Atividade realizada no [Kaggle](#);
- Uso de Redes Neurais para a classificação de textos de acordo com o sentimento que ele transmite (*positivo ou negativo*);
- No fim do dia, você treinará modelos MLP (Scikit-Learn, TF/Keras) para a classificação dos textos;
- A ‘vetorização’ dos textos será feita com o auxílio de um modelo *Doc2Vec* [1] pré-treinado;
- O *Doc2Vec* [1] é uma Rede Neural, então é legal ler o artigo original da técnica - se tiver interesse em NLP, ver as primeiras aulas deste [curso](#);

# Dados

- Os dados que utilizaremos formam uma amostra de avaliações de filmes encontradas no site do IMDb;
- Esses dados foram consolidados pelos pesquisadores de [2] e podem ser encontrados em sua versão bruta no endereço <https://ai.stanford.edu/~amaas/data/sentiment/>;
- Utilizaremos 25K textos rotulados para treino e 25K textos rotulados para teste;



# Materiais de Apoio

Além do notebook “Exemplo” no [Kaggle](#), temos:

1. ***analise-de-sentimentos.ipynb*** (uso recomendado): neste notebook mostramos como utilizar o modelo Doc2Vec pré-treinado para a obtenção dos embeddings;
2. ***tensorflow-keras.ipynb*** (opcional): neste notebook mostramos como treinar e usar redes neurais MLP com Tensorflow/Keras;
3. ***introducao-doc2vec.ipynb*** (opcional): neste notebook damos mais detalhes sobre treinamento e aplicações para o Doc2Vec;

Os materiais podem ser encontrados no E-Disciplinas ou [aqui](#).

# Etapas e Avaliação

Aqui temos um resumo do que fazer (**ver Kaggle para + detalhes**):

1. (1 ponto) Pré-processar textos e utilizar modelo de embedding pré-treinado para obter (X\_train, y\_train) e (X\_test, y\_test);
2. (5 pontos) Treinar duas Redes Neurais para a classificação dos textos (fazendo otimização dos hiperparâmetros);
3. (4 pontos) Treinar modelos alternativos e submeter os melhores resultados para a competição.



# Referências

[0] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).

Link: [https://cs.stanford.edu/~quocle/paragraph\\_vector.pdf](https://cs.stanford.edu/~quocle/paragraph_vector.pdf)

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

# Contato

**Felipe Maia Polo**

E-mail: [felipemaiaapolo@gmail.com](mailto:felipemaiaapolo@gmail.com)

Linkedin: [linkedin.com/in/felipemaiaapolo](https://www.linkedin.com/in/felipemaiaapolo)

Site: [felipemaiaapolo.github.io](https://felipemaiaapolo.github.io)

WhatsApp: (16) 99745-6372