Projeto PCS5029 - Amazon pet product reviews classification

1st Eduardo de Andrade Nogueira
Escola Politécnica da Universidade de São Paulo (Poli/USP)
São Paulo - SP, Brasil
eduardonogueira@usp.br

***Abstract*—Este resumo apresenta a metodologia utilizada na resolução do problema de classificação de texto de avaliação de produtos de PETs. A base de dados e as descrições do desafio Kaggle onde foi proposto este problema pode ser visto no LINK**

***Index Terms*—NLP, WordTokenizer, MultiLayer Perceptron, Support Vector machine, K-Nearest Neighbors, Decision Tree**

I. Introdução

Este trabalho tem como objetivo a resolução do desafio do Kaggle de identificar por meio do review dos clientes de um produto de pet obtido do site da Amazon, para qual categoria de animal ele está destinado. Abaixo observa-se alguns exemplos de reviews para cada uma das classes presente no Dataset:

- **Birds** - My little Cockatiel Remus (Remy) loves it! He adores the flower, which I called Alien because it looks like a UFO sorta...just found out it was supposed to be a flower. Its his favorite one!
- **Bunny Rabbit Central** - Our piggy had a plastic bowl and she would always knock it over. This one is much deeper, holds more feed & she doesnt knock it over like she did the plastic.
- **Cats** - My cats do not like this toy. I've had it for a few weeks and never see them playing with it. I would not recommend it.
- **Dogs** - Sam has an everlast treat each nite before bed, like a good tooth brushing. The only downside is finding a place that keeps them in stock as well as multiple flavors.
- **Fish Aquatic Pets** - This is my second one and I love it. Keeps the tank clean.The main problem is once you take the cover off to replace the filter I could never get it to fit right again. but still a good buy.
- **Small Animals** - Our rescue rat, George, really likes this food. (He prefers this to lab pellets. And seeded diets are not recommended.) Good quality and healthy. Well worth the money. Oxbow is a good brand.

Nas próximas seções é apresentado a metodologia usada para a solução do problema, bem como os resultados obtidos com a metodologia proposta.

II. Resolução do problema

A. Limpeza dos dados

- **LowerCase:** Um exemplo da aplicação da transformação LowerCase pode ser observado abaixo:
  - **String Original:** ESTOU TODO MAIUSCULO
  - **String Modificada:** estou todo maiusculo
- **Remoção de espaços:** Um exemplo da remoção de espaços na string pode ser observado abaixo:
  - **String Original:** EU TENHO MUITO ESPAÇO
  - **String Modificada:** EU TENHO MUITO ESPAÇO
- **Remoção de número junto com texto:** Um exemplo da remoção de número junto ao texto na string pode ser observado abaixo:
  - **String Original:** EssaStringTemNUMEROS289
  - **String Modificada:** EssaStringTemNUMEROS
- **Remoção de pontuação:** Um exemplo da remoção de pontuação na string pode ser observado abaixo:
  - **String Original:** EU.TENHO.MUITA.PONTUAÇÃO
  - **String Modificada:** EU TENHO MUITA PONTUAÇÃO
- **Remoção de StopWords:** Um exemplo da remoção de StopWords na string pode ser observado abaixo:
  - **String Original:** all i want to do is remove this stopwords
  - **String Modificada:** want remove stopwords
- **Remoção de Underline "_":** Um exemplo da remoção de underline na string pode ser observado abaixo::
  - **String Original:** ESTOU_REMOVENDO_ESSES_UNDERLINES
  - **String Modificada:** ESTOU REMOVENDO ESSES UNDERLINES
- **Remoção de URLs:** Um exemplo da remoção de URL na string pode ser observado abaixo:
  - **String Original:** Vou remover urls http://www.vouserremovido.com
  - **String Modificada:** Vou remover urls

B. Word Tokenizer

Com o objetivo de converter as strings de atributos em vetores onde é possível aplica-los nos classificadores foi utilizado a biblioteca CountVectorizer do SKLearn que transforma as strings em matrizes matrizes.

C. Pipeline de teste e métricas de avaliação

O CountVectorizer, possibilita a seleção da quantidade de elementos no vetor, neste trabalho foi testado doze situações diferentes: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 e aplicado os classificadores, Perceptron Multicamadas (MLP), Máquina de vetor de suporte (SVM), K-Próximos Vizinhos (KNN) e Árvore de Decisão (DT).

As métricas de avaliação usadas neste trabalho foram: Acurácia, Recall, Precisão e F1.

D. Avaliação da performance dos classificadores

Após realizado o teste proposto na seção II-C de variação do tamanho do vetor de saída do CountVectorizer, os resultados foram dispostos de forma gráfica nessa seção, onde é possível observar as quatro métricas usadas (Acurácia e o Score F1), respectivamente nas imagens 1 e 2.

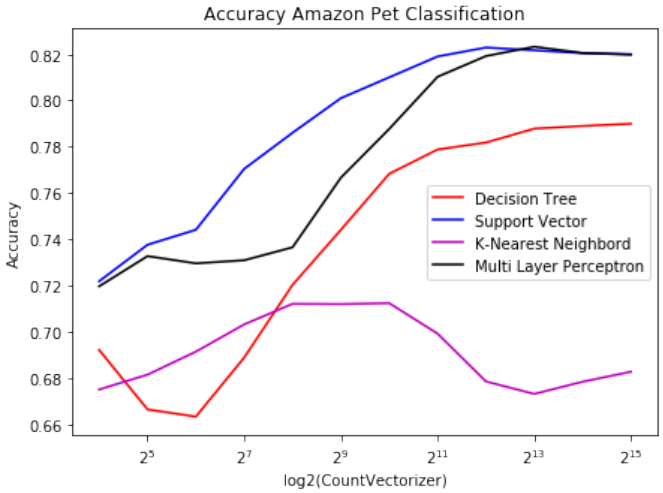

Fig. 1. Acurácia no teste de classificação de produtos de PET

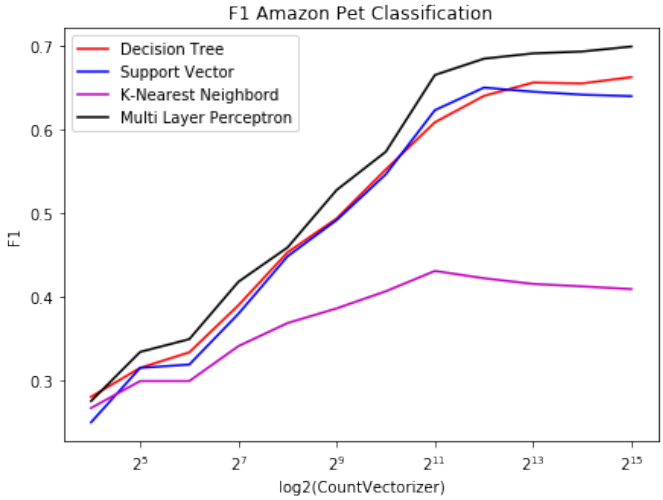

Fig. 2. F1 no teste de classificação de produtos de PET

Analisando a acurácia obtida em conjunto com o Score F1, pode-se afirmar que a melhor configuração de classificador é o Perceptron Multi Camadas (MLP) com o CountVectorizer com 8192 features.

III. Resultado desafio Kaggle

Após aplicado a metodologia descrita neste resumo, o resultado foi submetido ao Kaggle para avaliação obtendo um score de 0.80947 o qual é obtido pela micro-averaged da pontuação F1.

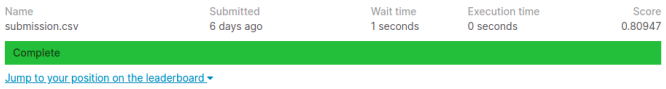

Fig. 3. Resultado Kaggle

IV. Conclusão

Após a limpeza dos dados apresentado na seção II-A, foi aplicado o pipeline de testes a fim de se obter a melhor configuração de classificador e tamanho do CountVectorizer. A seleção da melhor configuração foi realizada por meio da acurácia obtida e do score F1.

Após selecionado o melhor classificador como o Perceptron Multi Camadas e o tamanho do vetor do CountVectorizer como tendo 8192 features, o resultado foi submetido ao Kaggle obtendo um score de 0,80947 atingindo o objetivo deste trabalho de classificar para qual pet estava destinado o review.

V. Acesso ao código e a base de dados

Os links para acessar o código e a base de dados usadas no desenvolvimento deste trabalho estão abaixo:

- **Código Fonte -** https://github.com/eduardoanog/Projeto-PCS5029
- **Base de dados -** https://www.kaggle.com/c/amazon-pet-product-reviews-classification/overview

References

[1] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010.

[2] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: A library for support vector machines." ACM transactions on intelligent systems and technology (TIST) 2.3 (2011): 1-27.

[3] Zhang, Zhongheng. "Introduction to machine learning: k-nearest neighbors." Annals of translational medicine 4.11 (2016).

[4] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.