

DESCRIÇÃO DO TRABALHO

December 8, 2021

1 Descrição do Trabalho - Parte 3

Aluno: Gabriel Souto Ferrante - 12620303

Arquivos do .ZIP da entrega:

Este PDF com as informações do relatório.

Um Notebook para o sistema Voilà.

Um script Python para o download do DATASET COCO

Um arquivo CSV com os dados das embeddings vindos da parte 2.

Instruções de entrega:

- 1 - Nesta etapa final, você deve entregar um sistema de Visualização Analítica (obtido com Voilà, p.e.)
- 2 - Seguindo os feedbacks recebidos nas etapas anteriores, aplique uma ou mais técnicas de interação nas suas visualizações de dados (Etapa 2)
- 3 - Caso seja necessário (e possível), permita ao usuário final do seu sistema interagir com os algoritmos de processamento, de forma que ele/ela possa decidir qual parâmetro funciona melhor para o seu caso. Por exemplo: número de clusters, alternar entre técnicas de normalização, etc.

2 Sobre os dados e observações

O conjunto de dados utilizado para aplicar os processos de visualização é composto por 50 imagens de cada uma das 10 classes ('bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe') referentes à animais do COCO Dataset, totalizando 500 imagens. Houve alguns insights e modificações na proposta, e por fim, escolhi utilizar a posição no plano bidimensional de cada imagem fazendo o uso de embeddings (vetores de características). Estas embeddings são calculadas utilizando uma rede neural convolucional EfficientNet-B0 pré-treinada com o conjunto ImageNet, implementada na biblioteca image_embeddings (https://github.com/rom1504/image_embeddings/tree/e6233e78b857285c32755587f585fa251cabae17).

Logo após a criação das embeddings, é analisado a dimensionalidade, sendo 499 linhas por 1280 colunas. Para tratar dessa alta dimensionalidade, é utilizado o método de redução de dimensionalidade T-SNE para facilitar a implementação de visualização, gerando no final um conjunto com 499 linhas e 2 colunas. Após a redução, é acrescentado mais duas colunas, uma para identificação de qual imagem aquela embedding pertence e outra coluna para a classe original da imagem.

Seguindo os grupos taxonômicos do site From Data to Viz, o conjunto utilizado já redimensionado integra no grupo dos híbridos (numéricos e categóricos). No conjunto possui 4 colunas, sendo duas numéricas, uma identificadora e uma categórica:

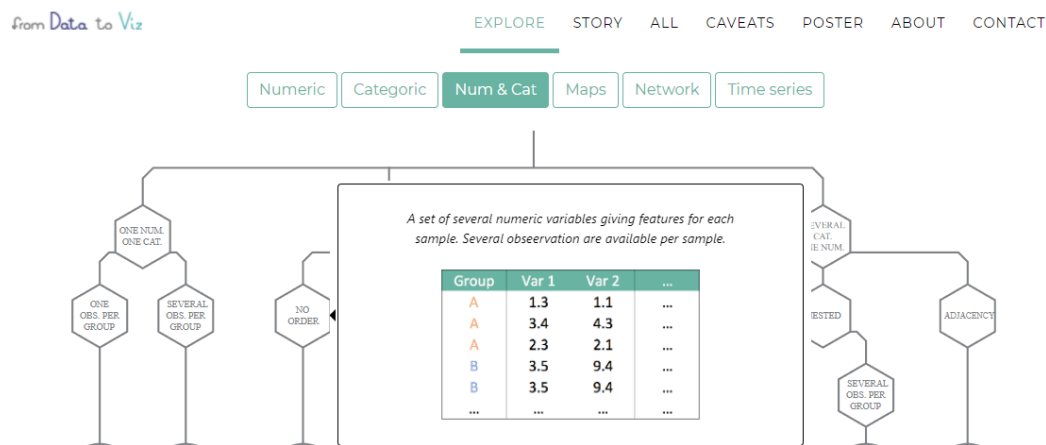
x (numérico)

y (numérico)

ImageName (Identificador)

ClasseOriginal (Categórico)

A forma escolhida para apresentar os dados, é o Scatter Plot. Por ser uma ótima opção em apresentar conjuntos onde duas variáveis numéricas estão correlacionadas.



3 Passos de execução

O sistema necessita que você tenha as imagens baixadas localmente, para a funcionalidade de XAI possa funcionar. Para isso, baixe o annotations do COCO Dataset (!wget http://images.cocodataset.org/annotations/annotations_trainval2017.zip) e execute o script em python SCRIPT-DATASET.py em seguida. Será criado uma pasta chamada valid, que conterá as imagens.

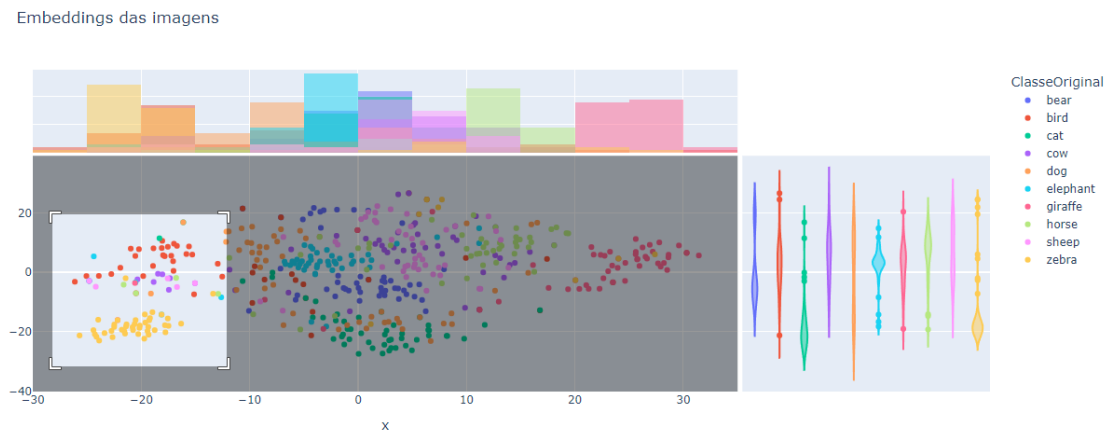
Com as imagens baixadas, execute o Notebook TrabalhoVisualizacao-P3-GabrielSoutoFerrante.ipynb para gerar o Voilà final. O voila final irá gerar uma visualização interativa com um scatter plot das embeddings das imagens e um campo de busca para gerar a explicação da classificação do modelo de CNN.

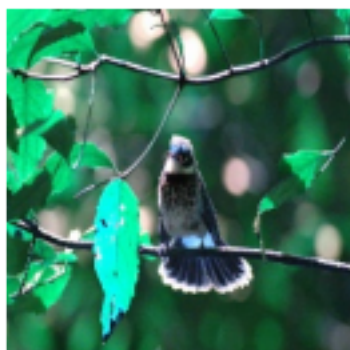
4 Técnicas de interatividade

É utilizado no gráfico de espalhamento através da biblioteca Plotly, as técnicas de Zoom (Geometric zooming) para destacar determinado espaço dentro do gráfico. Além disso, pode ser feito um filtro (Filtering), removendo/mantendo dados da apresentação de acordo com determinado atributo categórico (no caso a Classe Original da imagem).

Para o mecanismo de XAI, foi feito um campo de texto e um botão para buscar imagens de acordo com o index informado (mapeamento das imagens localmente), representando a técnica de Dynamic

Querys. O resultado da busca é uma imagem, que apresenta as principais características que foram relevantes para a classificação de tal imagem e sua localização no gráfico de espalhamento. Essas querys ajudaram a perceber detalhadamente como foi a classificação dos outliers, identificando imagens que possuem os animais em questão, porém em um cenário adverso.

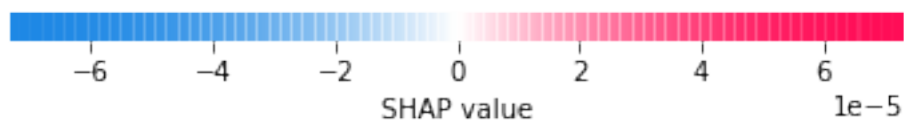
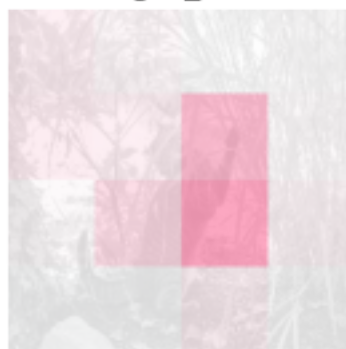




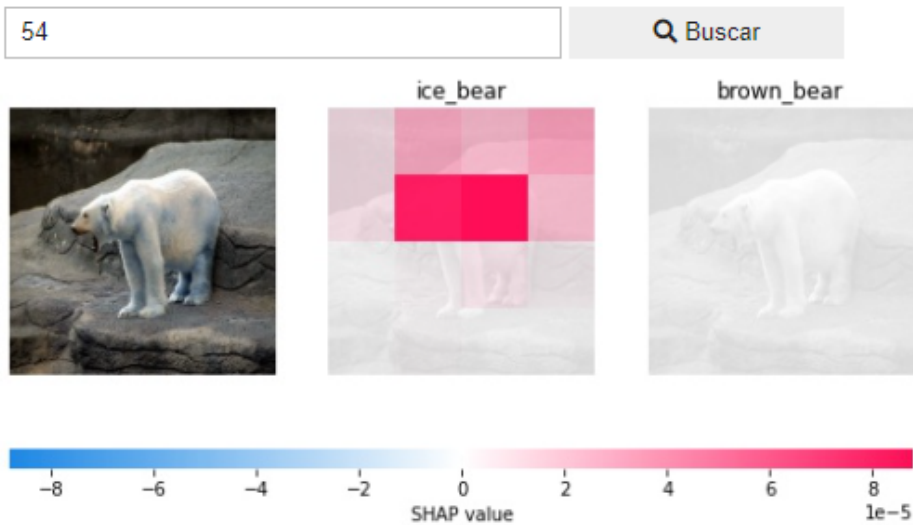
hummingbird



tiger_cat



Entre com um index de uma imagem entre 0 à 498 para visualizar as características de classificação



Os insights que pode ser retirado da visualização do Scatter Plot, é que as apesar dos erros, os grupos referentes a cada classe possuem a maioria das imagens da classe correta. Entretanto, em vários casos, o modelo não acertou, gerando embeddings próximas à outra classe. Outra característica que podemos observar é que imagens de certas classes foram bem classificadas, pois suas embeddings formaram pseudos-clusters bem definidos, como no caso das imagens das classes Zebra, Sheep e Elephant. A classe Bird e Dog foram as que menos possuíram aglomerações de pontos, demonstrando que a classificação não foi tão precisa. Para a visualização da explicação da classificação, é observado que o modelo pode classificar bem em imagens com poucas oclusões, diferentemente para imagens com algum tipo de obstrução. É interessante também que pode ser observado a diferença entre o que o modelo classificou com a classe real da imagem e sua posição no gráfico de espalhamento de sua embedding, nem sempre são iguais ou equivalentes.