Parte CC

Nós utilizamos a técnica de gradiente boosting para melhorar o modelo anterior que tentava prever se um usuário iria gostar ou não da leitura. Nesse notebook podemos observar dois modelos, um que utilizou grid search para tuning dos parâmetros, e um que utilizou os valores default para comparação.

No nosso modelo com grid search, tivemos uma diferença pequena no best score com parâmetros personalizados e com parâmetros default, mas nosso score em geral foi bom, tendo ficado com 0.775 contra 0.772 dos parâmetros default, para um classificador de interesse de notícias.

Podemos ver pelo resultado das predições do modelo com tuning de parâmetro, que o teste de predição com a linha de teste que foi criada com base no bias do dataset utilizado, teve uma probabilidade muito maior no predict, retornando 0.459 (a 'tendência' gerada pelo gerador de dataset tendencioso foi 40 a 60%, ou seja, teve precisão alta), ao passo que um com outro tema e com uma contagem de palavras muito maior teve predição de 0.039 de chance de leitura.

Quanto aos experimentos e implementação, nós utilizamos o grid search com diferentes parâmetros e métricas, procurando o que retornasse o melhor resultado.

No nosso segundo modelo utilizamos uma técnica de deep learning, utilizamos o modelo de redes neurais convolucionais para analisar imagens, e reconhecer letras, com o objetivo de que nosso modelo seja capaz de ler as notícias, e organizar elas assim que for feito uma nova publicação, o modelo iria classificar a notícia, procurando por tags, pré cadastradas, e depois categorizar o modelo com base nessas informações.

Para essa implementação inicial, usamos uma base de dados com imagens do alfabeto com 26 letras escritas à mão e utilizamos para isso tensores Conv2D, MaxPooling2D, Flatten, Dropout e Dense(Camadas ocultas). A ideia da implementação é que os primeiros tensores que serão os de entrada, vai ter um input no formato (height, width, RGB), e o MaxPooling2D vai pegar os valores máximo da matriz de tamanho (3 * 3) que depois o Flatten vai achatar os vetores para as camadas ocultas, e então as camadas ocultas Dense irão fazer a classificação da imagem dentre as 26 labels.

Ainda no modelo com redes neurais convolucionais, utilizamos para o tuning dos parâmetros a validação cruzada com 5 folds para aumentar a precisão e reduzir bias.

Tivemos juntamente com esse modelo de visão computacional para reconhecer letras, um modelo do mesmo tipo para reconhecer dígitos de 0 a 9. Ele também usa a técnica de CNN, e seu objetivo final seria trabalhar em conjunto com o modelo de caracteres alfabéticos para reconhecer palavras para poder fazer a classificação das publicações, seu modelo tem camadas tais como a do de letras, porém com algumas diferenças na forma e arranjo das camadas.

Dentre os dois modelos, o CNN teve o melhor desempenho, com precisão média de 99.038. Essa diferença se dá por causa que o Gradient boost depende bastante do tuning dos parâmetros, e das informações disponíveis nas colunas, então ainda depende de um insight humano, enquanto o que CNN não precisa, ele vai definir por ele mesmo o que tem mais valor e o que é importante.

Todos os modelos de aprendizado de máquina foram feitas análises da Accuracy, e isso está disponibilizado no git.
Link: https://github.com/JesuisOriginal/SeLiga-na-rede-neural