

Relatório <7 - Prática> - <Mergulho nas IDEs Online para Aprendizado de Máquina (I)>

<Ronny Gabryel Colatino de Souza>

Descrição da atividade

Esse módulo mostrou o ecossistema Jupyter e as alternativas pra desenvolver em Python. Confesso que não sabia que tinha tanta opção diferente, cada uma resolvendo problemas específicos que eu nem percebia que existiam até testar na prática



Python Shell e IPython

Primeiro foi mostrado brevemente o python pelo terminal e até funciona né, um print() roda normal, mas aí você tenta fazer qualquer coisa um pouco mais complexa e já vira uma confusão. O shell básico serve mais pra testar comando isolado mesmo, tipo uma conta rápida ou ver se uma função tá funcionando.

A solução mais óbvia seria criar arquivo .py e rodar com python tipo "arquivo.py" mas isso acaba com toda a interatividade. Cada vez que você quer mudar alguma coisa tem que salvar, sair, rodar de novo até funciona mas é bem chato de trabalhar assim.

O IPython foi uma surpresa boa arcaica mas boa até descobri que ele é tipo o pai do Jupyter tanto o Notebook quanto o Lab usam ele como base. A diferença parece pequena mas muda bastante: ele coloca cor no código igual nas IDEs normais, numera as execuções pra você acompanhar o que já rodou e mantém aquela parte boa do shell.

Jupyter Notebook

A instalação eu pulei porque já tinha feito antes. O importante mesmo foi entender como funciona essa ideia de células - ele reforça bastante coisa que eu já sabia.

Basicamente tem dois tipos de célula principais as células "Code" rodam Python normal, e as células "Markdown" deixam você escrever documentação junto com o código. Isso muda muito a forma de trabalhar porque você não tá só escrevendo código, tá documentando o raciocínio junto.

O Markdown é bem simples de usar só coloca # pra título, ## pra subtítulo, 1. 2. 3. e vai formando lista automático. Parece bobeira mas poder estruturar o pensamento direto no ambiente onde você tá codando, em vez de ter que fazer isso num documento separado, facilita demais inclusive lembra muito os README do GitHub que agora entendo que seguem a mesma lógica

A grande vantagem do Jupyter Notebook sobre arquivo Python normal não é nem técnica, é mais sobre como você trabalha você consegue ver código resultado e explicação tudo junto na mesma tela quando você volta no projeto depois de uns dias não precisa ficar tentando lembrar o que tava pensando. Tá tudo documentado ali junto

Jupyter Lab

Essa foi a primeira vez usando o Jupyter Lab parece uma IDE de verdade, não só um lugar pra abrir notebook. O legal que tipo o Notebook clássico funciona bem quando é um projeto simples de um arquivo só mas pra gerenciar vários arquivos ele é bem limitado achei legal irei usar mais.

O que reforça o negócio do Lab Tem aquele painel de arquivos na esquerda igual VSCode, sistema de abas pra alternar entre vários notebooks abertos, tudo no navegador ainda. Tem uma aba de extensões pra adicionar funcionalidades, e você consegue ver todos os kernels que estão rodando isso já me salvou várias vezes porque eu esquecia notebook rodando e comia toda a RAM algumas vezes crachava meu not por conta disso.

Outra coisa legal o Markdown no Lab tem uma aba "Contents" que transforma os títulos em tipo um índice clicável. Ou seja, Markdown não é só pra deixar bonito, serve pra organizar mesmo. Em notebook grande isso faz toda diferença, você acha as coisas em segundos em vez de ficar rolando pra achar.

Google Colab

O Colab é basicamente Jupyter Lab mas na nuvem parece mudança pequena mas muda muita coisa na prática. O GPU de graça pra treinar modelo isso é absurdo de diferente. Meu note levaria horas pra fazer o que o Colab faz em minutos. E outra vantagem é que você não enche seu computador de biblioteca e dependência que às vezes até conflitam entre si tudo roda isolado nos servidores do Google

Isso é especialmente bom pra quem não tem computador potente um Chromebook básico consegue rodar análise pesada porque quem tá processando é o servidor remoto. A barreira pra começar em data science fica bem menor.

Kaggle

O grande diferencial é a integração com datasets e modelos. No canto superior direito tem uma aba "Input" que deixa você baixar dataset direto pro ambiente de uma biblioteca gigante. Precisa de dataset de imagem você acha que precisa alguns cliques e tá disponível no projeto. Isso elimina aquela parte chata de fazer upload configurar caminho de arquivo verificar se baixou certo.

Outra coisa interessante quando você salva o notebook, o Kaggle roda tudo do zero automático. No começo achei meio irritante porque demora mais pra salvar, mas percebi que é parecido com uma validação. Se alguma coisa quebra nessa execução completa, você descobre na hora que criou dependência escondida entre as células

Conclusões

O importante mesmo foi entender que cada uma resolve um problema específico. Antes eu achava que era só escolher uma e usar pra tudo, mas na prática você acaba usando várias dependendo da situação é tipo, muito bom e interessante esse modulo.

Referencias

Card 7 – video: Jupyter Notebook Complete Beginner Guide -
https://www.youtube.com/watch?v=5pf0_bpNbkw