

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

Felipe Fonseca

Descrição da atividade

Durante a atividade foi apresentada algumas formas de utilizar o Jupyter Notebook e Jupyter Lab, primeiramente de forma local, depois algumas alternativas via nuvem, e por final, o apresentador do vídeo também mostrou alguns atalhos de comando e dicas para aumentar a produtividade enquanto trabalha com essas tecnologias.



Python Shell e IPython

Primeiramente é apresentado que o python pode ser rodado via terminal, independente do sistema operacional (Mac, windows ou linux). Como demonstração, é apresentado um código print() no python, e ele realmente funciona. Porém o problema do python rodando no terminal é que ele não é útil para fazer blocos de códigos ou um algoritmo completo, pela forma como ele funciona, só é útil para realizar pequenos comandos, como um print, ou uma função de soma. Para executar o python pelo terminal com uma quantidade maior de código, é utilizado um [arquivo.py](#), isso é, você cria um arquivo .py e executa ele utilizando o comando "python <nome do arquivo .py>", porém ainda não é o ideal para programação.

Após isso foi apresentado o Ipython, que foi de certa forma o "precursor" do jupyter, porque o jupyter notebook e o jupyter lab são feitos em cima do IPython. O funcionamento do IPython é bem similar ao Python, mas adicionando um pouco mais de interatividade. Eles adicionam cores nas funções e nos códigos, como se fosse uma IDE, além de irem mostrando as execuções com a contagem do lado igual é no jupyter notebook.

Jupyter Notebook

Ele começa ensinando a instalar o jupyter notebook, porém isso eu nem prestei atenção pois já tenho instalado né. Depois ele explica um pouco como utilizar o notebook, ou seja, criar novas células, cortar, e explica também explica os tipos de célula "Code", que é a célula onde a gente de fato programa em python, e a célula "markdown", que serve para escrever textos.

Ele explica um pouco como utilizar o markdown, coisa que eu não sabia até então e achei interessante: Você pode utilizar uma # para criar um título, ## para criar um subtítulo, e pode usar números seguidos de um ponto como 1. para criar uma lista, é realmente um texto bem interativo e acredito que seja semelhante ao README do github por exemplo.

Basicamente a vantagem do jupyter notebook para o código normal é a facilidade que tem de visualizar não só o código mas também de entender o processo de pensamento de quem estava fazendo.

Jupyter Lab

O jupyter lab a primeira vista parece mais como uma IDEA do que o jupyter notebook, do meu ponto de vista, o jupyter notebook parece ser usado quando tem que programar com apenas um arquivo, pois a navegação de arquivos dele por exemplo é bem básica. Nesse quesito o jupyter lab é bem superior, ele possui uma aba com gerenciamento de arquivos no lado esquerdo, e assim como em uma IDEA como o VSCODE por exemplo, você pode abrir vários arquivos como se fossem abas e alternar rapidamente entre elas, isso sem sair da página do jupyter lab no navegador. Outro aspecto do jupyter lab é a existência de uma aba de extensões, além da possibilidade de ver todos os arquivos e kernels que estão ativos.

O apresentador destaca um ponto importante do jupyter lab/notebook, que é o fato de que você não precisa rodar os códigos em ordem, que é algo que realmente já é bem perceptível desde o começo do bootcamp, porém ele trás também o aviso de que é importante manter a ordem de forma que ele possa ser executado de uma vez, pois evita bagunçar a cabeça do programador e permite rodar tudo de uma vez sem problemas, o que indica que a possibilidade de rodar as células de código em ordem distintas da que estão é mais para teste enquanto programa do que para uso prático mesmo.

Voltando ao markdown, outra curiosidade sobre ele apresentado é que utilizando a aba de contents, você pode navegar no notebook através dos títulos e subtítulos colocados no markdown, mostrando que o markdown é muito mais do que apenas um comentário, mas também serve para achar facilmente tópicos no código.

Google Colab

Google colab é basicamente um jupyter lab, porém que roda nas nuvens, o que nos dá muitas vantagens, como o fato de podermos utilizar de GPUs para rodar os códigos, que é algo que será muito útil no treinamento de modelos, além de não ter que instalar os pacotes e bibliotecas no nosso dispositivo. Por conta desses fatores, o google colab também é uma alternativa muito boa para quem não possui um computador ou um notebook muito bom.

Kaggle

Kaggle também fornece um serviço na nuvem e é gratuito para uso, de forma geral ele é bem similar ao google colab. Algo interessante sobre o kaggle é a facilidade de importar datasets ou modelos de treinamentos, notebooks etc. Isso porque no canto superior direito há uma aba de input, onde podemos baixar diretamente na pasta do kaggle através de um simples botão de download. Ele permite a busca, ou seja, podemos navegar em uma vasta biblioteca de datasets por exemplo e facilmente instalar no nosso projeto.

Outra coisa interessante do kaggle é que quando você vai salvar o arquivo ele roda todo o código desde o começo até o final, o que serve para 'avisar' caso tenha algo fora de ordem.

Além disso, os notebooks podem ser deixados públicos no kaggle. De forma geral, ele é basicamente uma ferramenta feita para aprendizagem de data science. Segundo a Fundação Oswaldo Cruz, 'O Kaggle é uma **plataforma de aprendizagem e competição para cientistas de dados**'.

Atalhos:

No final do vídeo foi apresentado alguns comandos que podem ser utilizados para aumentar a produtividade enquanto trabalho com notebooks, então eu coloquei esses comandos de forma simples em uma tabela para facilitar o entendimento:

Atalhos	
Rodar as células e ir para a próxima:	Tecla shift + enter.
Rodar a célula e manter nela:	Tecla control + enter.
Sair da célula:	Tecla esc.
Entrar na célula:	Tecla Enter.
Mover entre as células:	Setas para cima e para baixo.
Adicionar uma célula onde está:	Tecla A.
Adicionar uma célula abaixo de onde está:	Tecla B.
Deletar a célula:	Tecla D 2x.
Transformar a célula em Markdown:	Tecla M.
Transformar a célula em Código:	Tecla Y

Uma curiosidade é que o apresentador do vídeo apresentou o comando para transformar uma célula em Markdown, mas não apresentou como transformar de volta em código, então eu adicionei esse a mais.

Prática:

Na prática eu fiz um código utilizando o kaggle, isso porque ele permite uma fácil adição de datasets. Como o foco do card não é código em si, 'é mais sobre a plataforma, achei que era a plataforma interessante de usar pois é o recurso mais único entre elas.

Conclusões

Como conclusão eu tenho que o jupyter notebook é uma excelente ferramenta para trabalhar com códigos pela versatilidade que temos graças a forma como as células funcionam, além disso o markdown permite uma organização de código que comentários normais não são capazes de fazer.

Quanto às ferramentas para trabalhar com o jupyter lab eu achei todas interessantes, mais particularmente os via Nuvem, ou seja, o google colab e o kaggle. O google colab dentro os apresentados é o meu favorito, porém considerando que é via Nuvem, algumas vezes o código

demora a ser executado por conta de internet, por isso, por enquanto, ainda seguirei utilizando o jupyter notebook através das extensões disponíveis no VSCode.

O ponto mais útil da aula na minha opinião foi os atalhos de teclado, pois eu não sabia de nenhum deles, e nos poucos segundos de prática que eu fiz eles já me ajudaram bastante.

Como uma crítica sem fundamento nenhum, o único problema que eu tive com todos esses interpretadores foi a falta de sugestões enquanto escrevia o código, fazendo com que eu tenha que escrever tudo na mão. Não cheguei a olhar exatamente se dá para baixar extensões para resolver isso, mas foi o que me incomodou enquanto eu fazia o Card.

Referências

Videoaula:

https://www.youtube.com/watch?v=5pf0_bpNbkw

Citação usada no relatório sobre o Kaggle:

<https://pcdas.icict.fiocruz.br/rede-de-parcerias/kaggle/>