Teste de performance: Local(PC) vs Remoto(AWS) utilizando artifícios e modelo de Machine Learning

Tags	Teste
Ultima edição por	
Autor(es)	Leticia Campos Valente
Created	@January 17, 2022 7:40 AM
Última edição	@January 18, 2022 2:48 PM
APLICAÇÃO	Gerais ANBU
Sprint	N/A
Modelo	N/A

Resumo

O presente documento apresenta especificações de duas máquinas em comparação, uma física e uma virtual, bem como os resultados de performance em termos de tempo de processamento de um notebook com um modelo de machine learning genérico criado para fins de teste, com o intuito de utilizar diversas funções consideradas “exigentes” na rotina de um cientista de dados.

1 - Introdução

Certos modelos de aprendizado de máquina considerados superiores no mercado, por seus altos scores em problemas mais complexos, refinados e com maior volume de dados, exigem capacidades de máquina acima do padrão para problemáticas menos exigentes ou mais simples.

Assim, a fim de facilitar o processo, os servidores em nuvem foram criados com o intuito de diminuir a necessidade de aparelhagem física por meio de um ‘aluguel’ de suas máquinas feitas remotamente, para que o usuário usasse e pagasse apenas pelo tempo utilizado para o modelo, tirando a necessidade de despender de muita verba para atualizar o próprio computador. Amazon, Netflix, iFood, CVC, Futbol Club Barcelona, Nubank, BTG Pactual e até o Departamento de Estados do Eua são serviços que utilizam no geral os produtos da Amazon AWS, o servidor em nuvem no qual esse estudo se apoia. (Fonte: 10 sites que usam AWS (Amazon Web Services) e você não sabia - Flexa)

O Amazon SageMaker, serviço o qual será utilizado neste estudo, fornece esses serviços de forma segura e de maneira a deixar o usuário no controle total do tráfego de arquivos ou rede, sendo assim a escolha de serviços conhecidos e renomados, como Intuit, ADP e Intercom em ramos similares ao da Mundiale. (Fonte: Clientes do Amazon SageMaker – Amazon Web Services (AWS))

2 - Objetivos

2.1 - Objetivos gerais

Rodar um notebook criado pela autora local e remotamente, utilizando-se de funções para registro de tempo nos passos de um modelo de Machine Learning que mais despendam deste

2.2 - Objetivos específicos

- Criar um notebook ‘exigente’, em termos de requerimento computacional, para um problema de Machine Learning
- Obter o tempo decorrido para importar um arquivo de 1,35GB (dataset), nas máquinas local e remota
- Obter o tempo decorrido para a parte de ETL, nas máquinas local e remota
- Obter o tempo decorrido na hiperparametrização (GridSearch), nas máquinas local e remota
- Obter o tempo decorrido para rodar o modelo na validação por k-cross, nas máquinas local e remota

3 - Especificações das máquinas

Não se sabe ao certo a comparação das duas características em conversão direta, apenas que são análogas. Por isso o estudo a fim de confirmar a hipótese de AWS Sagemaker ser o serviço mais apropriado para futuros modelos mais complexos de Machine Learning

3.1 Local

Processador	i5 11ª geração
RAM instalada	8GB

3.2 Remoto

Instância: ml.t3.medium (Gratuita por 2 meses, 0,081 USD por hora após finalização do período gratuito)

VPC	2
Memória	4GiB

4. Modelo testado, parâmetros e artifícios de Machine Learning

Acredita-se que apenas um modelo é suficiente para este teste e, no caso, o escolhido foi o K-Nearest Neighbors por ser conhecido, com base em testes da autora e opiniões públicas, por ser um modelo que exige da máquina um tempo maior para ser processado. O parâmetro para hiperparametrização será o ‘n_neighbors’ e é o suficiente para aplicação do método GridSearch que já é o mais exigente processo do notebook controle e despende de muito tempo na máquina local (informação comprovada por outros trabalhos da autora)

5. Notebook e dataset utilizados

Notebook

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/30f5da9b-9cf7-42fe-a0bc-8c3223a11051/nbpronto.ipynb

Dataset

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/62f51fbe-4f92-4692-b536-9323bbf72ff1/vehicles.csv

6. Resultados

A instância gratuita de avaliação do serviço, ml.t3.medium, não suportou rodar o notebook, causando interrupção do kernel na linha de leitura do csv. Suspeita-se que o motivo seja baixa memória ram ou insuficiência do processador alocado para essas instâncias. As Figuras 1 e 2 mostram as únicas informações obtidas na falha.

Figura 1: Prompt de interrupção do Kernel

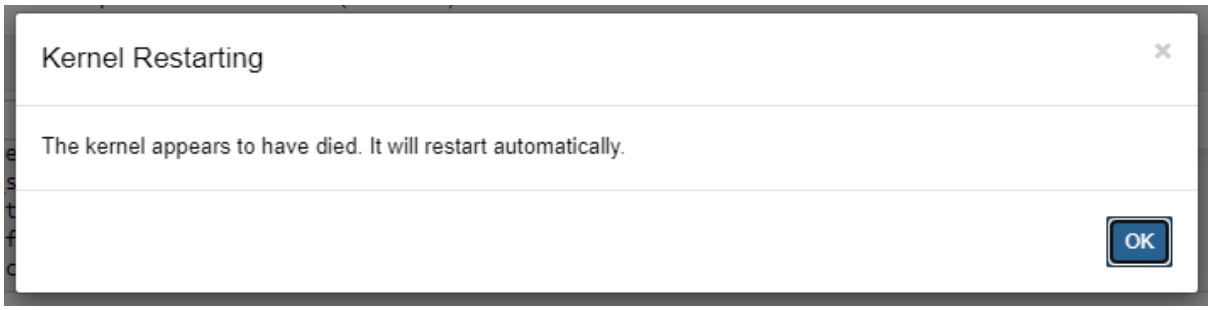
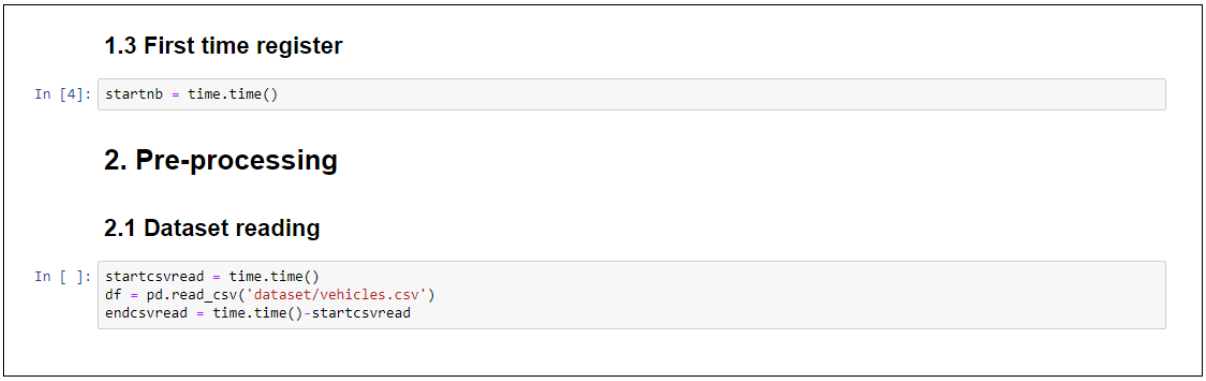


Figura 2: Última linha executada antes da interrupção do Kernel



1.3 First time register

```
In [4]: startnb = time.time()
```

2. Pre-processing

2.1 Dataset reading

```
In [ ]: startcsvread = time.time()
        df = pd.read_csv('dataset/vehicles.csv')
        endcsvread = time.time()-startcsvread
```

Sendo assim, os únicos tempos documentados foram da execução local, explicitados na Tabela 1 abaixo:

Tabela 1: Resultados coletados de tempo de execução

	Tempo de execução local (s)	Tempo de execução remoto (AWS Sagemaker)(s)
Leitura do dataset em csv	32 segundos	—
ETL	3 segundos	—
Hiperparametrização por cross-validation (GridSearch + K-cross)	1104 segundos	—
Treino do modelo e scoring completo	4 segundos	—

Nota: Cada vez que se roda o notebook pode ser que o tempo dê diferentes resultados, sendo estes demonstrados na tabela acima o tempo rodando localmente com aplicativos ‘de costume’ abertos e número médio de abas de navegador que a autora costuma deixar aberto durante toda a jornada de trabalho.