Materiais:

Q: Onde encontro os materiais de apoio e os arquivos notebooks para execução dos exercícios?

R: O material de apoio consta no ícone de folha a4 ao lado do título da disciplina, os notebooks do professor André e do professor Juliano estão em uma página do Dropbox, que o estudante poderá fazer o download e executar.

Os notebooks do professor André são os nomeados como (Arquivos YPNB).

Durante a aula 01, o professor André indica que o conjunto de dados Municípios estará nos slides, mas não consta. Mas os notebooks estão disponíveis na folha a4 ao lado do nome da disciplina e ao abrir o notebook ele possui o link e a solução proposta.

Ferramentas:

Q: Gostaria de saber qual é a maneira (linhas de código) de importar um arquivo csv direto do meu computador para o Colab?

R: Diretamente do desktop não é possível ler os arquivos, mas se há acesso ao Google Driveé possível fazer o upload desses arquivos no google drive e então, ler no Google Colab a partir do Google Drive. O Colab oferece uma biblioteca para isso, abaixo um exemplo:

```
from google.colab import drive

drive.mount('/content/drive', force_remount=True)

file_path = "/content/drive/My Drive/new_data_full.csv"

data = pd.read_csv(file_path, sep=';').replace({np.NaN: None})
```

Obs.: Ele irá pedir que você faça o login e dê permissão para acesso ao seu Google Drive.

Obs 2.: Sempre que reiniciar o kernel você terá que fazer isso.

Q: Referente à área de tratamento de textos, como fazer para extrair informações de arquivos PDF, arquivos texto etc. Como transformar esses textos em formatos semiestruturados pelo menos, como CSV?

R: Para soluções de problemas mais simples, é possível usar uma lib como o "tabula".

Segue referências que podem ajudar:

https://tabula-py.readthedocs.io/en/latest/getting_started.html#example

https://betterprogramming.pub/convert-tables-from-pdfs-to-pandas-with-python-d74f8ac31dc2

Para soluções mais robustas pode ser que seja necessária uma plataforma específica para converter e armazenar textos, como por exemplo o elastic search Solr. Depois basta acessar o Solr via Python para fazer os trabalhos de ciência de dados.

Conceitos e Exercícios:

Q: Na resolução da parte 3 do exercício 2 (vídeo parte 2, aprox. 18min) fiquei em dúvida em relação a como foi feita a atribuição da média para as extremidades. Como foi aplicado em duas etapas, a média atribuída para cada extremidade será diferente.

R: Sobre a diferença, ela seria menor que 0.1 numa escala de 100. Na maior parte do caso, não vai fazer diferença no resultado, exploração ou ML. No entanto, o método correto realmente seria substituir ambas as extremidades na mesma instrução, além de ser a maneira mais elegante.

Q: Sobre o exercício a respeito do time que mais marcou gols:

R: A maneira que o professor Juliano agrupou os times considera qual time marcou mais gols como mandante e não ao todo, diferente do que o exercício propôs. O professor Juliano encontrou o Santos como goleador daquela temporada, enquanto na verdade o melhor ataque foi do Corinthians (Campeonato Brasileiro de Futebol de 2015 – Wikipédia, a enciclopédia livre (wikipedia.org)).

Para chegar à resposta correta, devemos somar os gols que cada clube marcou como mandante e como visitante para encontrar quem de fato marcou mais gols.

Q: Sobre o exercício e o conjunto de dados da NBA:

R: Ao pesquisarmos sobre o jogador Wilt Charmberlain no Google é possível descobri que a temporada em que ele mais marcou foi a de 1962, com 4029 pontos listados. Já em 1965, ele marcou na verdade 2534 pontos, metade do que está no banco de dados.

Em 1965 ele trocou de time, e o banco reflete a pontuação dele em cada time (1480 + 1054) e a pontuação total do ano (2534), e é por isso que a soma no código resulta no dobro do valor real de pontuação.

Outro problema no banco de dados é que ele indica que Eddie Johnson possui o maior número de temporadas jogadas, com 33 temporadas. Até a data de fechamento do banco de dados, deveríamos encontrar Kevin Garnett e Kevin Willis, ambos com 21 temporadas.

Como há mudança de clube por parte dos jogadores no meio da temporada, há múltiplas entradas para cada jogador em cada ano, então deveríamos ter filtrado por valores únicos (. nunique()). Ainda assim, teríamos Mike Dunleavy com 26 anos incorretamente em primeiro lugar. Ao pesquisarmos novamente é possível descobrir que é porque Mike Dunleavy pai e filho, ambos de mesmo nome, jogaram na NBA, porém o banco registra apenas o pai.

Q: Levando em consideração as análises de erros em dados acima, o que podemos fazer para evitar que a gente cometa esses erros em análises do mundo real no dia a dia? Como evitar cair nessas possíveis armadilhas dos dados?

R: Na verdade no mundo real nem sempre temos a resposta real tão fácil assim para compararmos as bases de dados, não é? As vezes os dados são agregados, sumarizados e transformados até a base que está sendo utilizada para a construção de alguma transformação.

Nos casos reais, geralmente as bases de dados são alvo de uma boa análise exploratória antes mesmo de construir qualquer tipo de solução ou algo do tipo. Existem analistas

de dados, analistas de negócio também o próprio cientista de dados que faz essa análise utilizando métricas, explorando os max, min, média, mediana entre outros.

Acho que no caso peculiar da base de dados do NBA, também se encaixa técnicas de detecção de outliers, visto que quando verificaste que o Eddie Johnson possuía 33 temporadas é algo meio destoante dos outros, visto que ele necessitaria começar a jogar talvez com 9 anos rsrsrs.

Sobre a questão dos pontos do Wilt, também considero como erro de quem está preenchendo ou até mesmo um quesito que foi considerado na hora de preencher, nesse caso, teríamos que ter um conhecimento prévio da construção do conjunto de dados, que também, nem sempre temos a disposição.

Todas as bases de dados estão sujeitas a ter esse tipo de erro, e é importante a etapa de análise exploratória justamente para identificá-los e verificar como é possível ajustá-los ou até mesmo excluir esses registros.

Q: Ao fazer todo o exercício da aula, percebi que eu estava com muito mais colunas, do que o esperado. Executando o comando info(), percebi que havia várias colunas com nomes iguais mudando apenas o numeral no fim (e.g. desc_clube1 e desc_clube2). O que pode ter acontecido?

R: Se executarmos mais de uma vez cada trecho de código (célula), vai acabar criando outras colunas de dados (*features*). Então, quando estiver com o programa pronto, reinicie o kernel do seu Jupyter ou Colab e execute tudo de uma vez, assim garante que vai executar cada célula apenas uma vez para obter as colunas sem redundância.

Q: Para o valor médio do FTr, acredito que precise ainda de um ajuste pois como se trata de uma taxa, a média não deveria ser maior do que 1.

R: Para o FTr, é possível fazer o ajuste pra ser com base no valor e não em quantile(). Usando como regra o intervalo]0,1[.

**FAQ gerado com base em comentários até o dia 05/12/2021.*