SELEÇÃO – ANALISTA DE DATA SCIENCE JR.

CASE PARA SOLUÇÃO

Candidato: Lucas Martins Belmino

FORTALEZA-CE

09/08/2020

Questão 1:

Para solução desta questão foi utilizado a ferramenta Python para carregar os dados contidos na planilha **"questao1_coronavirus.csv"** e também para manipular os dados afim de obter os indicadores solicitados na questão.

Indicadores:

- Número de casos acumulados;
- Número de mortes acumulado;
- % de mortes sobre infectados;
- % da população já infectada.

Manipulação dos dados com Jupyter Notebook e Python:

No arquivo, **"questao1.ipynb"** é o arquivo no formato jupyter notebook que, se executado, realizará a carga e manipulação dos dados.

A sua execução é dividida nos passos descritos abaixo.

1 – Carregando os dados:

```
dfcov = pd.read_csv('../bases/questao1_coronavirus.csv',
                    encoding='utf-8',
                    sep=';')
dfcov.sort_values(by= 'date',ascending= True,inplace= True)
dfcov.reset_index(inplace= True)
dfcov.drop(columns= 'index',inplace= True)
```

Aqui os dados contidos na planilha **"questao1_coronavirus.csv"** são carregados no DataFrame **dfcov**. Além disso, é alterada a ordenação dos dados de forma a tomar as datas de forma crescente.

2 – Cálculo dos valores acumulados:

Os dois primeiros indicadores solicitados são do cálculo dos valores acumulados de número de casos e número de mortes. O passo mostrado na figura abaixo realiza o cálculo dos acumulados e, por fim, apresenta os dados referentes a cidade de Fortaleza afim de avaliar se os dados estão coerentes.

```
dfcov['new_confirmed_accumulated'] = dfcov.groupby(['city','state'])['new_confirmed'].cumsum()
dfcov['new_deaths_accumulated'] = dfcov.groupby(['city','state'])['new_deaths'].cumsum()
dfcov[dfcov.city == 'Fortaleza'].tail(1)
```

state	city	new_confirmed	new_deaths	estimated_population_2019	new_confirmed_accumulated	new_deaths_accumulated
CE	Fortaleza	60	1	2669342.0	32854	3110

```
dfcov[dfcov.city == 'Fortaleza'][['new_confirmed','new_deaths']].sum()
```

```
new_confirmed    32854
new_deaths        3110
dtype: int64
```

Como desejado, observa-se que, ao somar a coluna **“new_confirmed”** e **“new_death”**, obtemos valor igual ao último elemento da coluna respectiva.

3 – Cálculo dos valores percentuais:

Para o cálculo dos valores percentuais são utilizadas as funções mostradas na figura abaixo.

```
dfcov['deaths_by_infected'] = dfcov['new_deaths_accumulated']*100/dfcov['new_confirmed_accumulated']
dfcov['infected_by_population'] = dfcov['new_confirmed_accumulated']*100/dfcov['estimated_population_2019']
```

Por se tratar da divisão de duas colunas, foram observados diversos ocorridos. Inicialmente, cidades apresentaram falsos-positivos e, assim, foi necessário retificar casos. Esse caso ocorreu em Umuarama-PR. Isso resultou em dados iguais a “NaN” ou infinito.

Desta forma, as funções abaixo corrigiram estes valores substituindo por zero os valores “NaN” (a observação dos dados indica que este valor ocorria em cidades com falsos-positivos que foram retificados). As funções substituíram por 1 os valores infinitos, pois este valor ocorria em cidade com zero infectados e um número superior a zero de mortos.

```
dfcov.fillna(0,inplace= True)
```

```
dfcov.replace(np.inf, 1,inplace= True)
```

4 – Criação de novo arquivo com os dados manipulados:

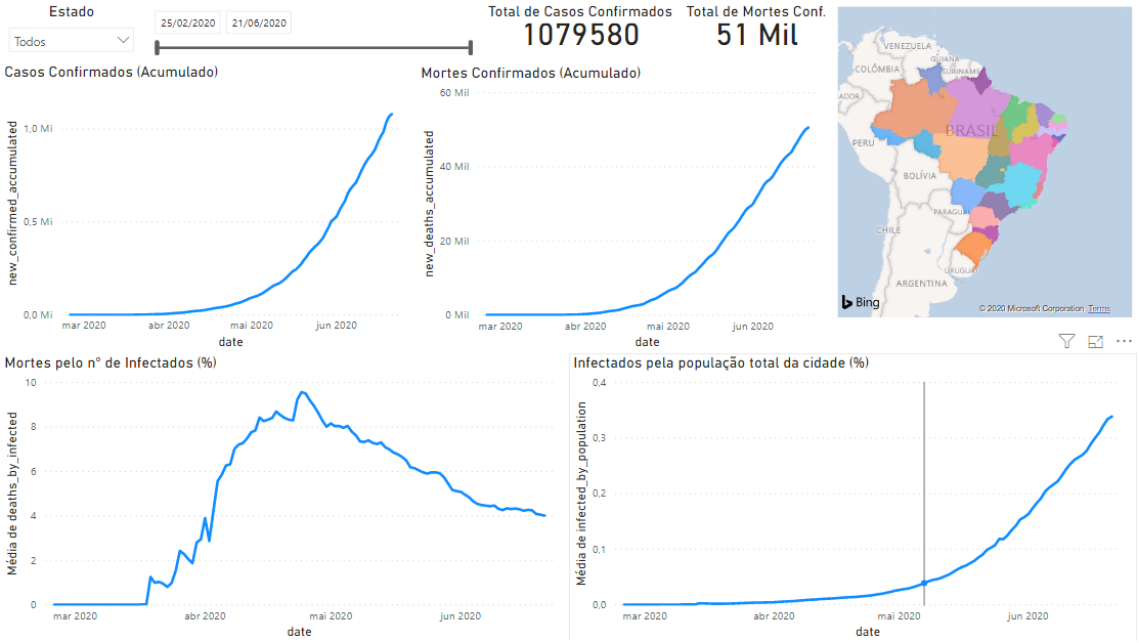
Afim de não alterar os dados disponibilizados, o código abaixo cria outro arquivo com os dados manipulados. A planilha criada, no formado “.csv”, denominada **“questao1_coronavirus_python”** será armazenada na pasta **questao1.**

```
dfcov.to_csv('questao1_coronavirus_python.csv',sep= ';',index= False,decimal = ',')
```

O código completo encontra-se na mesma pasta no arquivo **“questao1.ipynb”**.

Apresentação dos dados:

Para apresentar os dados foi escolhida a ferramenta PowerBI. O arquivo, nomeado **“dashboard_covid”**, é apresentado na figura abaixo.



Como solicitado, são apresentados os gráficos dos indicadores e além disso, no topo, foram adicionados os filtros para avaliar os dados de cada estado e a data desejada. Para filtrar por estado, pode-se escolher na lista no canto superior esquerdo ou clicar no estado referente no mapa. Para filtrar por data, pode-se escolher a data desejada ou arrastar o cursor na parte superior do dashboard.

Questão 2:

A ferramenta utilizada para a solução desta questão foi o “MySQL Workbench”. Esta ferramenta permite realizar consultas em um banco MySQL, além de oferecer editor de texto e ambiente de execução para consultas.

Afim de evitar erros nas consultas realizas, foi elaborado um banco similar ao proposto pela questão e foram adicionados dados simples para permitir que as “querys” fossem executadas e os dados avaliados. O arquivo **“1 - criando_db.sql”** cria o banco e o arquivo **“2 - carga_db.sql”** insere os elementos. Os dois arquivos são disponibilizados na pasta **questao2**.

Query 1 - Todas as compras realizadas no mês de janeiro de 2020 em lojas do estado do Ceará (CE).

Arquivo: **query1.sql**

```sql
select  p.ID_PESSOA as 'ID da pessoa', p.nm_pessoa as 'Nome da pessoa',
        t.dt_ref as 'Data Referência da Venda',
        v.VL_VENDA as 'Valor de Venda'
from f_vendas v
left join d_pessoa p on p.ID_PESSOA = v.ID_PESSOA
left join d_tempo t on t.ID_TEMPO = v.ID_TEMPO
left join d_loja l on l.ID_LOJA = v.ID_LOJA
where t.NU_MES=1 and t.NU_ANO=2020 and l.DS_UF='CE';
```

Foi realizado o Join das diversas tabelas e realizado o filtro dos dados.

Query 2 - Quantidade de compras por cliente no mês de março de 2020.

Arquivo: **query2.sql**

```sql
select  p.ID_PESSOA as 'ID da pessoa',
        count(v.ID_VENDA) as 'Quantidade de compras'
from f_vendas v
left join d_pessoa p on p.ID_PESSOA = v.ID_PESSOA
left join d_tempo t on t.ID_TEMPO = v.ID_TEMPO
where t.NU_MES=3 and t.NU_ANO=2020
group by p.ID_PESSOA;
```

Foi realizado o Join das tabelas, o filtro dos dados e, por fim, o agrupamento para contagem de compras.

Query 3 - Todos os clientes que não fizeram compras no mês de março de 2020

Arquivo: **query3.sql**

```sql
drop view if exists nao_comprou_marco;
create view nao_comprou_marco as
select p.ID_PESSOA, p.NM_PESSOA from f_vendas v
left join d_pessoa p on p.ID_PESSOA = v.ID_PESSOA
left join d_tempo t on t.ID_TEMPO = v.ID_TEMPO
where not (t.NU_MES=3 and t.NU_ANO=2020)
GROUP BY p.ID_PESSOA;

drop view if exists comprou_marco;
create view comprou_marco as
select p.ID_PESSOA, p.NM_PESSOA from f_vendas v
left join d_pessoa p on p.ID_PESSOA = v.ID_PESSOA
left join d_tempo t on t.ID_TEMPO = v.ID_TEMPO
where (t.NU_MES=3 and t.NU_ANO=2020)
GROUP BY p.ID_PESSOA;

select  nm.ID_PESSOA as 'ID da pessoa',
        nm.NM_PESSOA as 'Nome da pessoa'
from nao_comprou_marco nm
left join comprou_marco cm
on nm.id_pessoa = cm.id_pessoa
where cm.id_pessoa is null;

drop view nao_comprou_marco;
drop view comprou_marco;
```

Foi realizada a criação de duas View. Na primeira, foram filtrados os clientes que não fizeram compras em março, no entanto, esta View contém todos os clientes que realizaram compras em outros meses que não março.

Desta forma, foi realizada uma nova View com os clientes que compraram em março e estes foram excluídos da lista anterior, resultando somente em clientes que não fizeram compras em março.

Query 4 - Data da última compra por cliente.

Arquivo: **query4.sql**

```sql
select p.ID_PESSOA, max(dt_ref) from f_vendas v
left join d_pessoa p on p.ID_PESSOA = v.ID_PESSOA
left join d_tempo t on t.ID_TEMPO = v.ID_TEMPO
group by p.id_pessoa;
```

Foi realizado o Join com outros tabelas e o agrupamento por clientes afim de encontrar a data máxima, que representa sua última compra.

Questão 3:

3.1 - Que tipo de problema estamos enfrentando e qual técnica você utilizaria para resolver esse problema?

Ao observar o arquivo **questao3_creditcard,** nota-se que se trata de um conjunto de dados rotulados. Esse conjunto é composto por uma amostra (que representa um cliente) e suas características (Idade, sexo, estado civil e outras). Cada amostra contém seu rótulo, que indica se este cliente vai ou não pagar a próxima fatura. Tendo em vista que se busca identificar, ou classificar, clientes com maior probabilidade de não pagar a próxima fatura, o problema proposto é o de classificação binária entre os bons e os maus pagadores.

Para este tipo de problema, e tendo em vista os dados disponíveis, deve-se utilizar um algoritmo de classificação supervisionado. Estes algoritmos vão utilizar dos rótulos já conhecidos para treinar seus métodos de classificação e desta forma aprimorar sua capacidade de separar cada um dos grupos aprendendo a relação entre características da amostra e seu rótulo.

3.2 - Se você só pudesse enviar comunicação para 10% dos clientes devido ao alto custo, para quais clientes abaixo você enviaria? Explique sua resposta. Utilizar os clientes da base "questao33_creditcard_clientes.csv"

Para visualizar os clientes que devem receber comunicação, acessar o arquivo "**clientes_cobrar"** disponível na pasta **questao3**. A estratégia utilizada para determinar estes clientes está detalhada no arquivo **questao3.ipynb** na mesma pasta.

A estratégia se baseia no uso de modelos de aprendizado de máquina capazes de realizar classificação de amostras. Foram utilizados três modelos: "LogisticRegression", "DecisionTreeClassifier" e "RandomForestClassifier". A escolha destes foi determinada pelo seu amplo uso e reconhecida efetividade para a tarefa de classificação.

Os modelos tiveram seus parâmetros escolhidos através de validação cruzada. A validação cruzada realiza diversas vezes o ciclo de treino e teste do modelo com parâmetros que se deseja avaliar, além de variar os dados que são utilizados para treino/teste. Assim, a validação cruzada determina, entre os parâmetros utilizados, aqueles que apresenta melhor resultado.

Além disso, foi avaliada a estratégia de Padronização dos dados e de Redução de características. A Padronização consiste em dividir cada amostra pela média e dividir pelo desvio padrão dos dados. Isso faz com que os dados fiquem todos com a mesma escala de magnitude, pois todas as características estarão centralizadas no zero com desvio padrão unitário.

A Redução de características, aqui realizada através da Análise de Componentes Principais ou Principal Component Analysis (PCA), é um método de redução de dimensionalidade que costuma ser usado para reduzir a dimensionalidade de grandes conjuntos de dados, transformando um grande conjunto de variáveis em um menor que ainda contém a maior parte das informações do grande conjunto. Embora estes métodos possam reduzir a acurácia do modelo, estes podem simplificar o modelo utilizado.

Escolha do Modelo

Ao fim dos testes com diversos parâmetros os modelos são avaliados através de três parâmetros: Acurácia, Recall e AUC. Estes são escolhidos, pois é preciso reduzir o número de Falsos-Negativos, que são valores negativos classificados como positivos. A escolha de um modelo com Acurácia, Recall e AUC superiores é uma estratégia utilizada para que o modelo apresente menos Falsos-Negativos.

Escolha dos 10% para enviar comunicado

Os 10% para envio do comunicado serão escolhidos através da resposta do modelo escolhido. Através do método **"predict_proba"** o modelo da como resposta aos dados inseridos a probabilidade de a amostra pertencer a primeira ou a segunda classe.

As respostas serão ordenadas da maior para a de menor probabilidade e serão escolhidos os primeiros 272 indivíduos (10% dos dados da planilha **"questao33_creditcard_clientes.csv"**).