.

1. Descreva como você aplicaria as ferramentas vistas para conduzir um projeto de análise de dados e modelagem em um contexto de dados clínicos (dados da área da saúde). Lembre-se que o contexto da saúde envolve alguns requisitos fundamentais de sigilo e ética com os dados dos pacientes. Leia um pouco sobre a LGPD (Lei Geral de Proteção de Dados) no Brasil.

O processo de ciência de dados envolvendo dados clínicos envolve algumas questões sensíveis a serem tratadas. Pois os dados a serem manipulados são tratados pela LEI Nº 13.709, DE 14 DE AGOSTO DE 2018 como dado pessoal sensível, CAPÍTULO I, artigo 5°, *"dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural"* **.**

Um aspecto fundamental antes de iniciar a aquisição dos dados para as futuras atividades de ciência de dados, é verificar o status destes dados; caso sejam de *acesso público* como definidos no CAPÍTULO II, artigo 7°, inciso 3°, ou sejam de *acesso privado*, neste caso como contém dados clínicos deve-se observar se existe o consentimento dos titulares dos dados conforme CAPÍTULO I, artigo 5°, XII. Outro ponto crucial a observar é aplicar a "anonimização **"** dos dados para que os mesmos não sejam associados diretamente a um indivíduo conforme a CAPÍTULO I, artigo 5°, XI da Lei Nº 13.709. O artigo 12 do capítulo CAPÍTULO II Seção II da mesma lei considera que os dados anonimizados não são considerados como dados pessoais

Uma vez que detenhamos dados em conformidade com a lei, podemos começar a aquisição dos dados para as atividades de ciência de dados. O processo de ciência de dados começa com a definição de qual problema será resolvido. Será a definição de um problema de negócio de uma clínica ou hospital ? Será um problema sobre como agilizar o tratamento de pacientes? Existem inúmeras possibilidades neste ínterim. Antes de adquirir os dados devemos sempre buscar o entendimento do problema a ser resolvido, definindo quais os objetivos deste projeto de ciência de dados, definir um planejamento deste projeto..

Por segundo, após a definição dos objetivos começa a aquisição e leitura dos dados, normalmente executamos essa tarefa utilizando a linguagem de programação python 3 com a biblioteca pandas, na maioria dos casos utilizamos a função read_csv,

Com a leitura começa a análise exploratória dos dados , com a plotagem de gráficos e exibição de métricas para identificação de variáveis que possam integrar parâmetros para a modelagem dos dados. Tarefa normalmente executada com as bibliotecas pandas (e a funções describe() e ProfileReport()), matplotlib e seaborn plotam os gráficos necessários.

Após a exploração dos dados começa a preparação dos mesmos para a modelagem, retirando dados ausentes ou nulos, quando necessário, identificando dados duplicados e excluindo do target do projeto, identificando outliers e recalculando parâmetros segundo métricas . Para estas etapas são utilizadas as bibliotecas do python 3 como pandas, numpy, missingno entre outras.

Depois de limpos os dados necessitam ser separados em dados de treino e teste, conforme os modelos a serem aplicados, como visto em aula o python 3 realiza a tarefa com poucas linhas de código.Como por exemplo no código abaixo:

```
# separando dados de treinamento e dados de teste

data = df2.sample(frac=0.9, random_state=786)

data_teste = df2.drop(data.index)



data.reset_index(drop=True, inplace=True)

data_teste.reset_index(drop=True, inplace=True)



print('Dados para Treinamento: ' + str(data.shape))

print('Dados para testes: ' + str(data_teste.shape))
```

Chegou a hora de realizar a etapa de treinamento de modelos , isto depende lógico da característica do problema a ser resolvido, seja classificação ou regressão, seja para aprendizado por regras, detecção de anomalias, existem hoje mais de 60 algoritmos de machine learning disponíveis para uso em tarefas de ciência de dados, com bibliotecas gratuitas em python 3. O pacote pyCaret permite o treinamento e avaliação de qual o modelo com melhor performance a aplicar nos dados a serem analisados.

Após realizado o treinamento do modelo desejado deve-se avaliar os resultados dos modelos com a métricas estatísticas competentes a cada modelo específico. O pacote pyCaret permite a avliação do modelo aplicado assim que treinado. Uma vez que o modelo foi apurado como satisfatório válida-se o modelo com a aplicação do mesmo aos dados de teste. Depois salvamos o modelo final e o testamos em dados diferentes dos aplicados anteriormente, verificando se as métricas corroboram com as atingidas anteriormente.

Caso o modelo venha a integrar alguma solução via aplicativo , software, o modelo final está pronto para ser enviado para deploy, ou produção. Ressaltamos que como os dados deste modelo podem não ser públicos, devemos atentar para a anonimização dos resultados em casos de dados sensíveis com o contexto da saúde, a solução de software por lei não pode apresentar relatórios, dashboards ou

planilhas contendo dados de clientes que não tenham dado o devido consentimento a essa aplicação conforme a Lei geral de Proteção de Dados..

2. O *Prophet* é uma ferramenta Python aberta, desenvolvida pelo time de pesquisa do Facebook. O *Prophet* é uma ferramenta bastante robusta para previsão de séries temporais, principalmente para dados com uma temporalidade ao longo dos anos, sensíveis à eventos de datas específicas, como feriados. Comente como você usaria as ferramentas vistas no curso para preparar uma base de dados temporais de um cliente, para avaliar a performance do *Prophet* nos dados do cliente.

Para analisar dados de séries temporais com o pacote Prophet do Facebook, primeiro temos que instalar o pacote no python 3, o Prophet contém uma dependência de outro pacote o pystan, necessitamos instalar os dois. Em um jupyter notebook os comandos seriam:

$ pip install pystan # ! pip install pystan (no google colab)

$ pip install fbprophet # ! pip install fbprophet (no google colab)

Para a análise de uma base de dados oriunda de um arquivo .csv importamos a biblioteca pandas , lemos o arquivo com a função read_csv, e exibimos um overview com o .head():

import pandas as pd

data = pd.read_csv('abacates.csv')

data.head()

Para plotagens podemos instalar também o pacote plotly, muito útil na visualização em séries temporais

$ pip install plotly # ! pip install plotly (no google colab)

A entrada de dados do pacote Prophet nada mais é do que um dataset com duas colunas mínimas 'ds' e 'y', onde 'ds' refere-se a um período de tempo no formato usual YYYY-MM-DD ou YYYY-MM-DD HH:MM:SS. A coluna 'y' trata-se de uma coluna com dados numéricos que desejam realizar uma predição ou previsão.

Para melhores visualizações importamos o matplotlib e o numpy, que auxiliaram em futuris gráficos:

import numpy as np

import matplotlib.pyplot as plt

Agora vamos imprimir as informações sobre o dataset com os tipos de dados das colunas , verificando se contém dados nulos ou não:

```
data.info()
```

Para mostrar as variáveis :

```
data.describe(include = ‘o”)
```

	Date	type	region
count	18249	18249	18249
unique	169	2	54
top	11/8/2015	conventional	BuffaloRochester
freq	108	9126	338

Ao analisar séries temporais necessitamos utilizar o sklearn LabelEncoder na conversão de variável categórica em numérica :

```
Features = LabelEncoder()
data.iloc[:,10] = le.fit_transform(dataset.iloc[:,10])
```

Mais uma overview de como ficaram os dados

```
data.head(2)
```

Criando os dados de entrada (X) e a variável target (y):

```
X= data[['Date',‘Total Volume’, ‘4046’, ‘4225’, ‘4770’,
‘Small Bags’, ‘Large Bags’, ‘XLarge Bags’, ‘type’]]
y= data.iloc[:,1]
```

Criando a base de dados de treinamento no Prophet :

```
train_data= pd.DataFrame()
train_data['ds'] = pd.to_datetime(X["Date"])
train_data['y']=y
train_data.head(2)
```

Aplicando a modelagem no Prophet com os valores default:

```
prophet_basic = Prophet()
```

prophet_basic.fit(train_data)

Prevendo valores futuros utilizando o make_future_dataframe() e visualizando o final da tabela com o .tail():

:dados_futuros= prophet_basic.make_future_dataframe(periods=300)

:dados_futuros.tail(2)

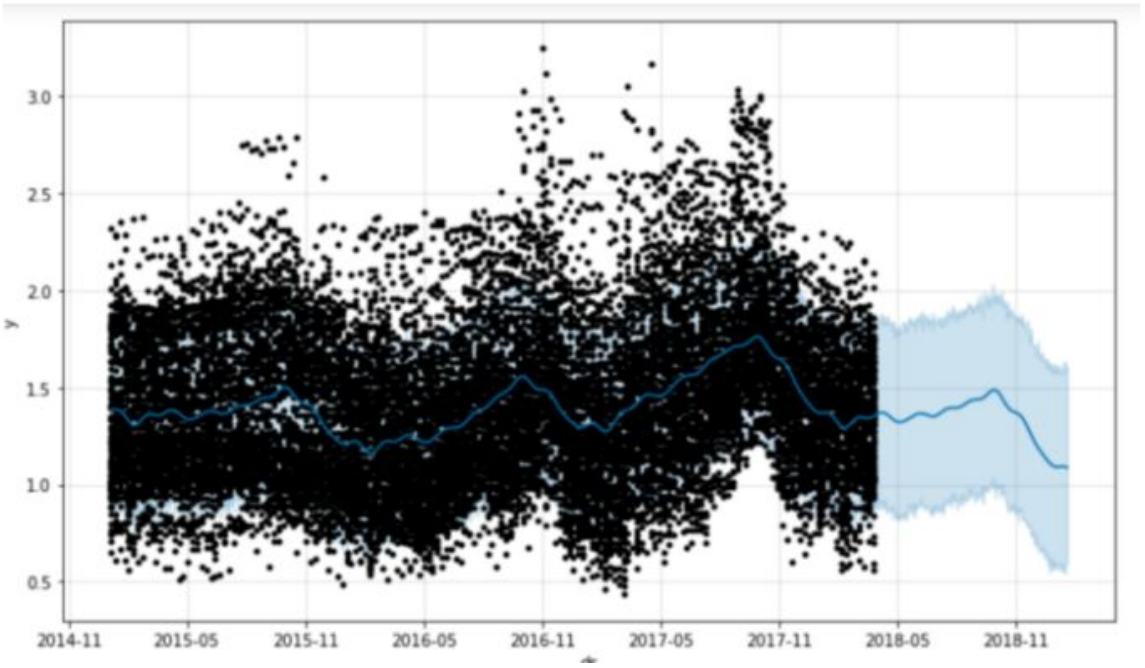
	ds
18547	2019-01-18
18548	2019-01-19

Realizando as previsões com os dados futuros:

previsoes=prophet_basic.predict(dados_futuros)

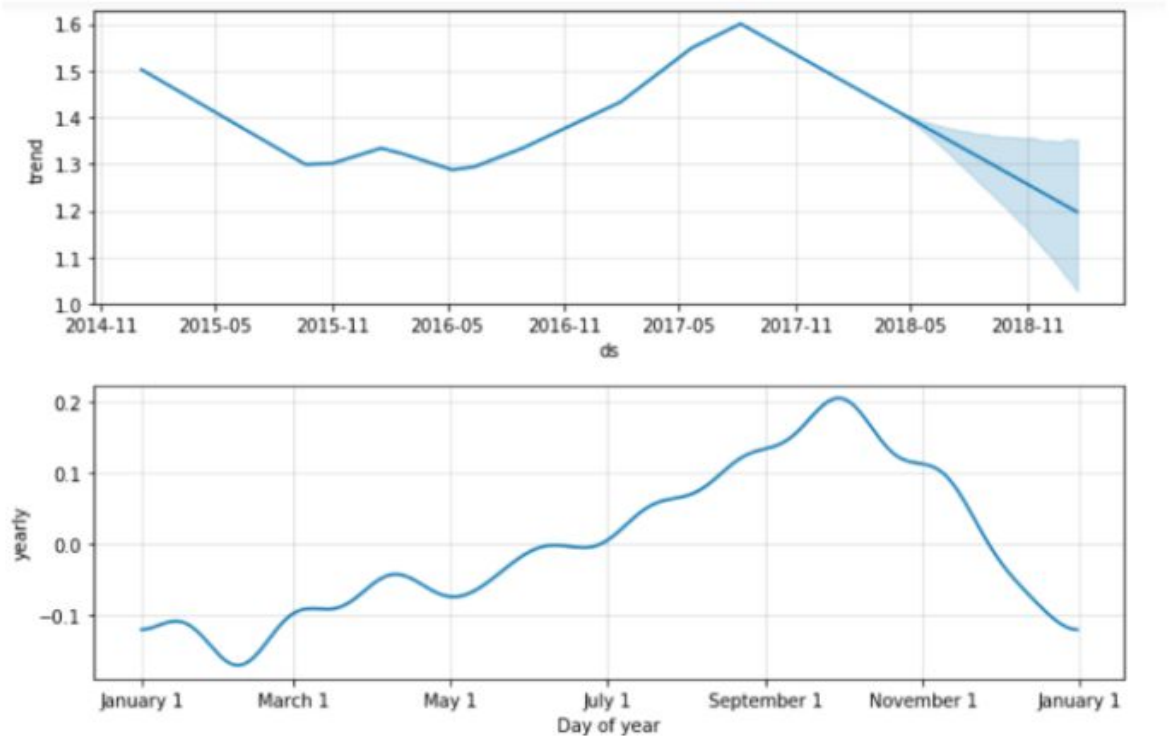
Plotando as previsões:

graph1 =prophet_basic.plot(previsoes)



Plotando os componentes da previsão:

graph1 = prophet_basic.plot_components(previsoes)



Observamos que em função da utilização da plataforma de ensino, não conseguimos anexar as imagens ao estudo.

O prophet permite desenvolver de forma rápida a aplicação de previsões para dados temporais, comparativamente muito mais rápido e com menos linhas de código utilizando somente as bibliotecas numpy e pandas.

3. Referências:

Aprendizado de máquina – Wikipédia, a enciclopédia livre (wikipedia.org)

Time series prediction using Prophet in Python | by Renu Khandelwal | Towards Data Science

Quick Start | Prophet (facebook.github.io)

Installation | Prophet (facebook.github.io)

TimeSeries/Prophet_Avacado.ipynb at master · arshren/TimeSeries · GitHub