

# PROJETO FINAL

DADOS

# ESCOLHA DO DATASET

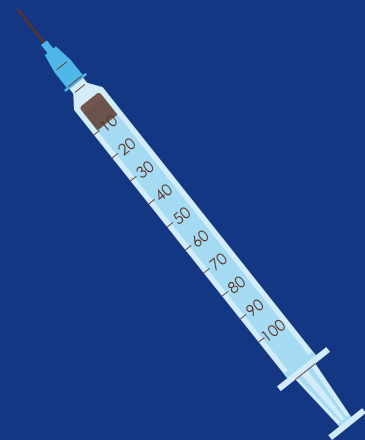
- Como foi feito?

Nosso objetivo foi buscar um Dataset que pudesse ser interessante na questão de informação e ao mesmo tempo possível de ser trabalhado nas tarefas necessárias que tínhamos, dashboard e a predição com machine learning principalmente.

- O que foi escolhido?

O dataset escolhido em questão foi o de diabetes, contendo informações sobre a saúde e práticas (como a de fumar) que podem influenciar na presença ou ausência da doença.

# DIABETES PREDICTION DATASET



# ETAPAS



```
import pandas as pd

#Leitura do dataset
df_diabetes = pd.read_csv("diabetes_prediction_data")
display(df_diabetes)

#Exportação dos dados para excel
df_diabetes.to_excel('Dataset_diabetes.xlsx')

#Transformando valores de texto em números
df_diabetes = df_diabetes.replace('Female', 0)
df_diabetes = df_diabetes.replace('Male', 1)
df_diabetes = df_diabetes.replace('Other', 2)
print("0 - Female \n1 - Male\n")
display(df_diabetes)

#Observando valores de texto na coluna de histórico
#de fumar para se fazer a transformação
display(df_diabetes['smoking_history'].unique())
df_diabetes = df_diabetes.replace('never', 0)
df_diabetes = df_diabetes.replace('No Info', 1)
df_diabetes = df_diabetes.replace('current', 2)
df_diabetes = df_diabetes.replace('former', 3)
df_diabetes = df_diabetes.replace('ever', 4)
df_diabetes = df_diabetes.replace('not current', 5)
display(df_diabetes['smoking_history'])
```

## TRATAMENTO DE DADOS



## DASHBOARD

```
from xgboost import XGBRegressor
xgboost = XGBRegressor(n_estimators = 50, max_depth = 4, learning_rate = 0.1)
xgboost.fit(x_treino, y_treino)

xgboost.score(x_treino, y_treino)

0.7190338974737431

xgboost.score(x_teste, y_teste)

0.6984255168958036

#Importando função para padronização
from sklearn.preprocessing import StandardScaler

#Padronizando o x treino
x_scaler = StandardScaler()
x_treino_scaler = x_scaler.fit_transform(x_treino)

#Padronizando o y treino
y_scaler = StandardScaler()
y_treino_scaler = y_scaler.fit_transform(y_treino.reshape(-1, 1))

#Padronizando o x teste
x_scaler = StandardScaler()
x_teste_scaler = x_scaler.fit_transform(x_teste)

#Padronizando o y teste
y_scaler = StandardScaler()
y_teste_scaler = y_scaler.fit_transform(y_teste.reshape(-1, 1))
```

## MACHINE LEARNING

# Agradecemos!

Gabriel Belchior Vieira Campanile  
João Vitor Evangelista de Souza