Publicação de Aplicação para Predição de Diagnóstico de Câncer

Bel Cogo e Bruno Hoffmann 2024/2 | Ciência de Dados & Big Data

Relembrando do Grau A...

- Modelo capaz de predizer o diagnóstico de câncer de pacientes;
- Dados: 1500 pacientes;
- Algoritmo usado: Random Forest;
- Métricas:
 - Acurácia: 89.3%;
 - Precisão: 92.6%;
 - Recall: 78.9%;
 - F1-Score: 85.2%.



Construção dos gráficos

Importando dados

```
# Carrega o dataset e manipula dados quantitativos em qualitativos para melhor visualização dos gráficos
dataset = pd.read_csv('./src/resources/The_Cancer_data_1500_V2.csv')
dataset_for_plots = dataset.copy()
dataset_for_plots['GeneticRisk'] = dataset_for_plots['GeneticRisk'].map({0: 'Baixo', 1: 'Médio', 2: 'Alto'})
dataset_for_plots['Gender'] = dataset_for_plots['Gender'].map({0: 'Masculino', 1: 'Feminino'})
dataset_for_plots['Smoking'] = dataset_for_plots['Smoking'].map({0: 'Não', 1: 'Sim'})
dataset_for_plots['CancerHistory'] = dataset_for_plots['CancerHistory'].map({0: 'Sem histórico', 1: 'Com histórico'})
```

Renderizando gráfico

```
st.pyplot(plot.get_figure())
st.pyplot(fig)
st.plotly_chart(graph)
```

Criando plot

```
plt.clf()
plot = sns.countplot(x=dataset_for_plots['Gender'])

plt.title('Distribuição por Gênero')
plt.ylabel('Contagem')
plt.xlabel('Gênero')
```

Como consumir o modelo?

Exportando modelo e *encoders*

```
import pickle

# Salvando modelo treinado em arquivo pkl

model_file_name = './src/resources/trained_random_forest.pkl'
pickle.dump(classifier, open(model_file_name, 'wb'))

# Salva encoders para usar na UI
scaler_encoder_file_name = './src/resources/scaler_encoder.pkl'
one_hot_encoder_file_name = './src/resources/one_hot_encoder.pkl'

pickle.dump(one_hot_encoder, open(one_hot_encoder_file_name, 'wb'))
pickle.dump(scaler, open(scaler_encoder_file_name, 'wb'))
```

Importando modelo e *encoders*

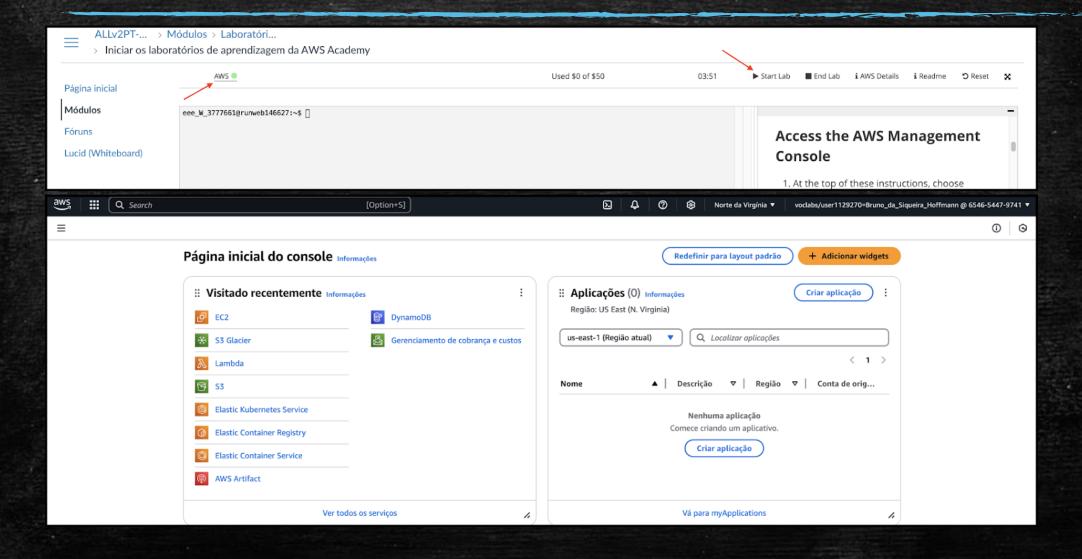
```
# Importação do modelo e dos encoders usados no treinamento
model = pickle.load(open('./src/resources/trained_random_forest.pkl', 'rb'))
one_hot_encoder = pickle.load(open('./src/resources/one_hot_encoder.pkl', 'rb'))
scaler_encoder = pickle.load(open('./src/resources/scaler_encoder.pkl', 'rb'))
```

Chamando modelo para predizer

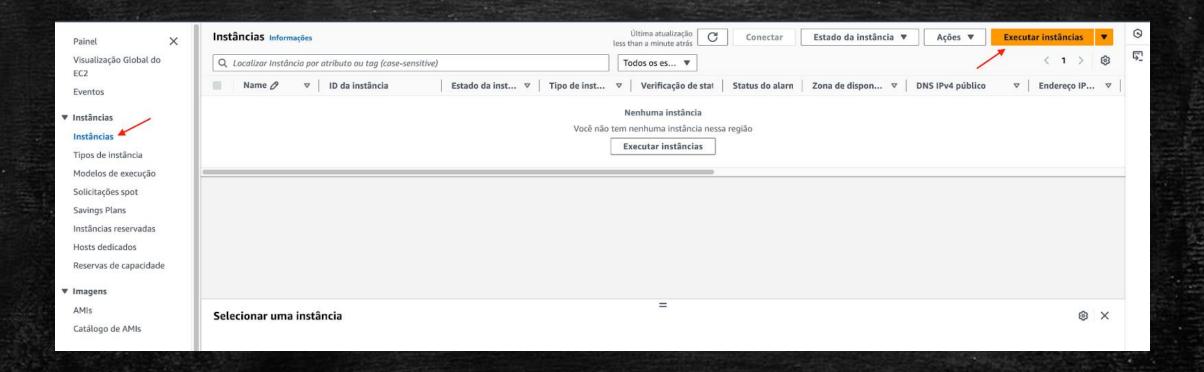
```
# Monta registro do paciente
patient = {
    'Age': age,
    'Gender': int(gender == "Feminino"),
    'BMI': bmi,
    'Smoking': int(smoking),
    'GeneticRisk': get_formated_genetic_risk(genetic_risk),
    'PhysicalActivity': physical_activity,
    'AlcoholIntake': alcohol_intake,
    'CancerHistory': int(cancer_history)
# Coloca o registro em dataframe
df = pd.DataFrame([patient])
values = df.values
# Aplica enconders em cima do registro
hot_encoded_values = one_hot_encoder.transform(values)
scaler_encoded_values = scaler_encoder.transform(hot_encoded_values)
# Realiza a predição em cima dos dados.
results = model.predict(scaler_encoded_values)
# Retorna para usuário o resultado.
result_to_show = ":red[Diagnosticado]" if results[0] else ":green[Não Diagnosticado]"
st.subheader("Resultado da Predição: " + result_to_show)
```

```
# Método para obter o valor numérico de GeneticRisk
def get_formated_genetic_risk(risk):
    map = {
        'Baixo': 0,
        'Médio': 1,
        'Alto': 2
    }
    return map[risk]
```

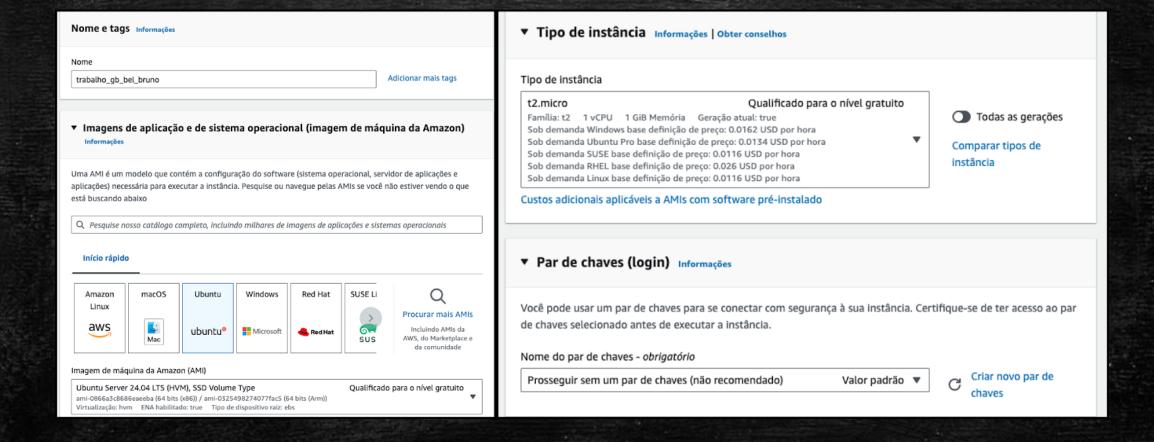
Configuração AWS



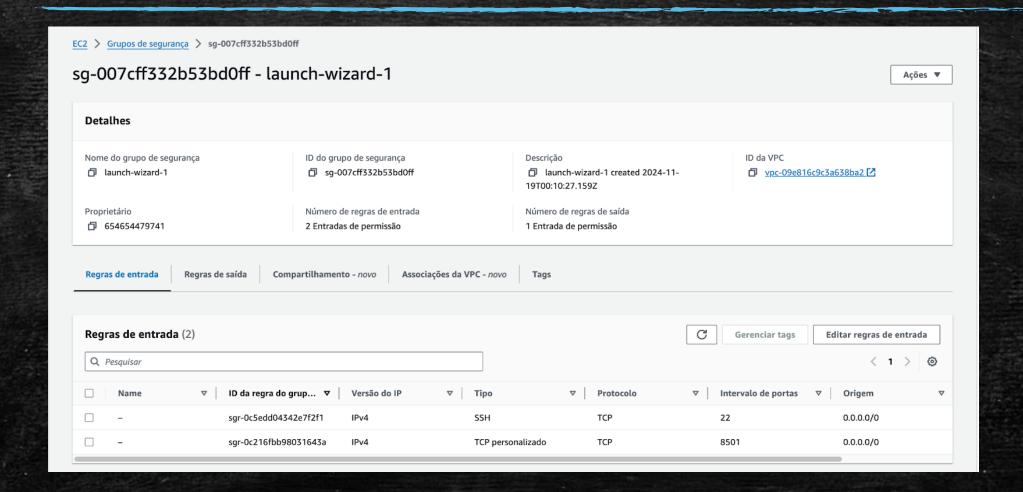
Criação da Instância EC2



Configuração da Instância EC2



Configuração de Grupos de Segurança



Conectar-se à instância Informações

Conecte-se à sua instância i-0bdb2bcd859ae6331 (trabalho_gb_bel_bruno) usando qualquer uma destas opções

Conexão de instância do EC2 Console de série do EC2 Gerenciador de sessões Cliente SSH ID da instância i-0bdb2bcd859ae6331 (trabalho_gb_bel_bruno) Tipo de conexão Conectar-se usando o EC2 Instance Connect O Conectar-se usando o endpoint do EC2 Instance Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address. Conecte-se usando o cliente baseado em navegador do EC2 Instance Connect, com um endereço IPv4 privado e um endpoint da VPC. Endereço IPv4 público 54.81.66.6 Endereço IPv6 Nome de usuário Insira o nome de usuário definido na AMI usada para iniciar a instância. Se você não definiu um nome de usuário personalizado, use o nome de usuário padrão, ubuntu. Q ubuntu (i) Observação: na maioria dos casos, o nome de usuário padrão, ubuntu, está correto. No entanto, leia as instruções de uso da AMI para verificar se o proprietário da AMI alterou o nome de usuário da AMI padrão.

Conexão ambiente EC2

Configuração ambiente EC2

Configuração de pacotes do python

- sudo apt update
- sudo apt-get update
- sudo apt upgrade -y
- sudo apt install python3-pip
- sudo apt install python3.12-venv

Download do repositório git

•git clone
https://github.com/BrunoHoffmann15/trab-1big-data-gb.git

Acessar pastas

•cd trab-1-big-data-gb

Configurar venv

- python3 -m venv venv
- source venv/bin/activate

Download das dependências da aplicação

• pip install -r src/requirements.txt

Executar a aplicação

• streamlit run src/app.py

Obrigada!

Dúvidas?