Aprendizado de Máquina (Algoritmos de ML) - Exploração de Dados e Pré-processamento

1. **Coleta de Dados Relevantes para o Negócio Proposto pela Empresa**
   Identificação das Fontes de Dados Relevantes Para a HealthTech Balderi Solutions, as fontes de dados relevantes podem incluir:

Dados de Pacientes: Informações demográficas, histórico médico, resultados de exames.

Dados de Consultas Médicas: Informações sobre consultas realizadas, diagnósticos, prescrições.

Dados de Exames Médicos: Resultados de exames laboratoriais, imagens médicas, relatórios de radiologia.

Dados Operacionais: Dados administrativos, registros de atendimento, tempos de espera.

Extração dos Dados, Garantindo Integridade e Qualidade Exemplo de Extração de Dados:

Exemplo de Extração de Dados:

```
import pandas as pd

# Carregar dados de pacientes
pacientes_df = pd.read_csv('dados_pacientes.csv')

# Carregar dados de consultas médicas
consultas_df = pd.read_csv('dados_consultas.csv')

# Carregar dados de exames médicos
exames_df = pd.read_csv('dados_exames.csv')

# Verificar a integridade e qualidade dos dados
print(pacientes_df.info())
print(consultas_df.info())
print(exames_df.info())
```

2. **Limpeza e Pré-processamento dos Dados**
   Tratamento de Valores Ausentes, Outliers e Dados Inconsistentes.
   Exemplo de Tratamento de Dados:

```
# Tratamento de valores ausentes
pacientes_df.fillna(method='ffill', inplace=True)
consultas_df.fillna(method='bfill', inplace=True)
exames_df.fillna(exames_df.mean(), inplace=True)

# Identificação e tratamento de outliers
```

```
import numpy as np

# Remover outliers utilizando o método do IQR
Q1 = pacientes_df['idade'].quantile(0.25)
Q3 = pacientes_df['idade'].quantile(0.75)
IQR = Q3 - Q1

outliers = pacientes_df[(pacientes_df['idade'] < (Q1 - 1.5 * IQR)) |
(pacientes_df['idade'] > (Q3 + 1.5 * IQR))]
pacientes_df = pacientes_df[~pacientes_df.index.isin(outliers.index)]

# Dados inconsistentes
# Normalizar os nomes dos pacientes para caixa baixa
pacientes_df['nome'] = pacientes_df['nome'].str.lower()
```

Padronização de Formatos e Unidades Exemplo de Padronização de Dados:

```
# Padronizar os formatos de data
consultas_df['data_consulta'] =
pd.to_datetime(consultas_df['data_consulta'], format='%d/%m/%Y')

# Padronizar unidades de medida (exemplo: glicose em mg/dL)
exames_df['glicose'] = exames_df['glicose'].apply(lambda x: x if x > 1
else x * 100)

# Exibir uma amostra dos dados padronizados
print(pacientes_df.head())
print(consultas_df.head())
print(exames_df.head())
```

3. **Verificação da Matriz de Confusão**
   Utilização da Matriz de Confusão para Avaliar o Desempenho de Classificadores.
   Exemplo de Matriz de Confusão:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Exemplo de dados
X = exames_df[['glicose', 'pressao_sanguinea', 'IMC']]
y = exames_df['resultado_exame']

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Treinar o modelo de classificação
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
```

```
# Fazer previsões
y_pred = clf.predict(X_test)

# Gerar a matriz de confusão
cm = confusion_matrix(y_test, y_pred)

# Visualizar a matriz de confusão
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Matriz de Confusão')
plt.show()

# Relatório de classificação
print(classification_report(y_test, y_pred))
```