ml-diabetes

June 28, 2025

MACHINE LEARNING PARA CLASSIFICAÇÃO

Este notebook apresenta um exercício prático de **Machine Learning para Classificação de Dados** utilizando as bibliotecas: **Pandas**, **Scikit-Learn**, **Matplotlib** e **Seaborn** do Python.

Será utilizado um conjunto de dados formatado como texto delimitado (CSV) contendo dados de pacientes mulheres gravidas. O objetivo é realizar uma previsão que indique a probabilidade da paciente desenvolver diabetes gestacional usando informações clínicas e de saúde da própria pessoa como variáveis preditoras. Serão consideradas as seguintes técnicas de treinamento de modelos de Machine Learning:

- Ingestão de dados.
- Separação dos dados.
- Treinamento do modelo de Machine Learning.
- Avaliação estatística dos resultado.
- Avaliação gráfica dos resultados.

1 - Instalação das Bibliotecas

[ ]:
```
# Instalação do Pandas para manipulação de dados.
!pip install pandas

# Instalação do Scikit-Learn para modelagem de machine learning.
!pip install scikit-learn

# Instalação do Matplotlib para criação de gráficos.
!pip install matplotlib

# Instalação do Seaborn para criação de gráficos mais estéticos.
!pip install seaborn
```

2 - Importação das Bibliotecas

[ ]:
```
# Usada para manipulação de dados.
import pandas

# Usadas para construir os modelos de machine learning.
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
```

```
# Usada para separação dos dados.
from sklearn.model_selection import train_test_split

# Usada para avaliar o desempenho do machine learning.
from sklearn.metrics import accuracy_score, confusion_matrix

# Usadas para construir gráficos.
import seaborn
import matplotlib.pyplot as pyplot
```

3 - Ingestão de Dados

O conjunto de dados contém informações coletadas pelo Instituto Nacional de Diabetes e Doenças Digestivas e Renais de Mulheres dos Estados Unidos (EUA) em um período de gravidez. Ele contém as seguintes colunas:

- **ID**: identificador único para cada paciente.
- **Gravidez**: quantidade de semanas de gravidez.
- **Glicose**: concentração de glicose no sangue.
- **Pressão Arterial**: pressão arterial diastólica.
- **Espessura do Triceps**: espessura da dobra da pele do tríceps.
- **Insulina**: insulina sérica de 2 horas.
- **IMC**: índice de massa corporal.
- **Predisposição Genética**: medida de predisposição genética para diabetes (histórico familiar).
- **Idade**: idade da pessoa em anos.
- **Diabética**: indica se tem diabetes ou não (1 - diabetes; 0 não tem diabetes).

[ ]:
```
# Define o caminho do arquivo de texto delimitado.
arquivo = "Diabetes.csv"

# Carrega o arquivo de texto delimitado.
df = pandas.read_csv(arquivo, sep=";", decimal=".", encoding="utf-8")

# Exibe os dados carregados.
display(df.head(15))
```

4 - Separação dos Dados

[ ]:
```
# Divide os dados para treinamento e validação.
df_treino, df_teste = train_test_split(df, test_size=0.3, random_state=42)

# Separa as variáveis independentes (X) e dependente (y) de treinamento.
X_treino = df_treino[['Gravidez', 'Glicose', 'Pressão Arterial', 'Espessura do␣
 ↪Triceps', 'Insulina', 'IMC', 'Predisposição Genética', 'Idade']]
Y_treino = df_treino['Diabética']

# Separa as variáveis independentes (X) e dependente (y) de validação.
```

```
X_teste = df_teste[['Gravidez', 'Glicose', 'Pressão Arterial', 'Espessura do␣
↪Triceps', 'Insulina', 'IMC', 'Predisposição Genética', 'Idade']]
Y_teste = df_teste['Diabética']
```

5 - Treinamento do Modelo de Machine Learning

[ ]:
```
# Cria o modelo de Regressão Logística.
modelo_RLog = LogisticRegression(random_state=42)

# Ajusta o modelo de Regressão Logística aos dados.
modelo_RLog.fit(X_treino, Y_treino)

# Cria o modelo de SVC (Support Vector Classifier).
modelo_SVC = SVC(random_state=42)

# Ajusta o modelo de SVC aos dados.
modelo_SVC.fit(X_treino, Y_treino)
```

6 - Avaliação Estatística dos Resultados

[ ]:
```
# Faz as previsões usando os modelos de machine learning criados.
RLog_previstos = modelo_RLog.predict(X_teste)
SVC_previstos = modelo_SVC.predict(X_teste)

# Calcula as métricas estatísticas para avaliar o modelo de Regressão Logística.
RLog_acuracia = accuracy_score(Y_teste, RLog_previstos)
RLog_matriz_confusa = confusion_matrix(Y_teste, RLog_previstos)
RLog_erro = 1 - RLog_acuracia

# Exibe os resultados para o modelo de Regressão Logística (RLog).
print("Resultados do Modelo de Regressão Logística:")
print(f"Acurácia: {RLog_acuracia:.2f}")
print(f"Erro: {RLog_erro:.2f}")

# Calcula as métricas estatísticas para avaliar o modelo de SVC (Support Vector␣
↪Classifier).
SVC_acuracia = accuracy_score(Y_teste, SVC_previstos)
SVC_matriz_confusa = confusion_matrix(Y_teste, SVC_previstos)
SVC_erro = 1 - SVC_acuracia

# Exibe os resultados para o modelo de SVC (Support Vector Classifier).
print("Resultados do Modelo SVC:")
print(f"Acurácia: {SVC_acuracia:.2f}")
print(f"Erro: {SVC_erro:.2f}")
```

7 - Avaliação Gráfica dos Resultados

[ ]:
```
# Define os rótulos
rotulos = ['Não Diabético', 'Diabético']

# Construção do gráfico da matriz de confusão da Regressão Logística.
pyplot.figure(figsize=(8, 6))
seaborn.heatmap(RLog_matriz_confusa, annot=True, fmt='d', cmap='Blues',␣
 ↪cbar=False,
                xticklabels=rotulos, yticklabels=rotulos)
pyplot.title('Matriz de Confusão - Regressão Logística')
pyplot.xlabel('Previsões')
pyplot.ylabel('Reais')
pyplot.show()

# Construção do gráfico da matriz de confusão do SVC (Support Vector␣
 ↪Classifier).
pyplot.figure(figsize=(8, 6))
seaborn.heatmap(SVC_matriz_confusa, annot=True, fmt='d', cmap='Greens',␣
 ↪cbar=False,
                xticklabels=rotulos, yticklabels=rotulos)
pyplot.title('Matriz de Confusão - SVC')
pyplot.xlabel('Previsões')
pyplot.ylabel('Reais')
pyplot.show()
```

8 - Consumo dos Modelos de Machine Learning

[ ]:
```
# Inserção manual dos valores para previsão de diagnóstico.
entradas = {
    'Gravidez': 2,
    'Glicose': 120,
    'Pressão Arterial': 70,
    'Espessura do Triceps': 22,
    'Insulina': 80,
    'IMC': 28.5,
    'Predisposição Genética': 0.45,
    'Idade': 33
}

# Cria um Data Frame com os valores inseridos.
df_entradas = pandas.DataFrame([entradas])

# Escolha do modelo de classificação.
escolha = 'RLog'

# Realiza a previsão com o modelo escolhido.
if escolha == 'RLog':
    estimativa = modelo_RLog.predict(df_entradas)
```

```
    print(f"Diagnóstico com Regressão Logística: {'Diabética' if estimativa[0]␣
 ↪== 1 else 'Não Diabética'}")

elif escolha == 'SVC':
    estimativa = modelo_SVC.predict(df_entradas)
    print(f"Diagnóstico com SVC: {'Diabética' if estimativa[0] == 1 else 'Não␣
 ↪Diabética'}")

else:
    print("Modelo escolhido inválido. Por favor, escolha: 'RLog' ou 'SVC'.")
```