eda_modeling

August 3, 2025

1 EDA and Modeling for Chronic Disease Prediction

This notebook demonstrates: - Exploratory data analysis (EDA) - Training a Random Forest Classifier - Saving the model using `joblib` - Explaining the model using SHAP

[3]:
```
import pandas as pd

# Load dataset
df = pd.read_csv('../data/diabetes_sample_data.csv')
df.head()
```

[3]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	0.044451	0.034309	-0.043401	-0.002592	0.019907	-0.017646	1
1	-0.001882	-0.044642	-0.051474	-0.026327	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	1
3	0.006207	0.050680	0.045599	-0.015999	-0.034821	-0.032356	-0.024528	-0.002592	0.022688	-0.009362	1
4	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	-0.002592	-0.031988	-0.046641	0

[4]:
```
from sklearn.model_selection import train_test_split

X = df.drop('target', axis=1)
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[5]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
import joblib

# Train model
```

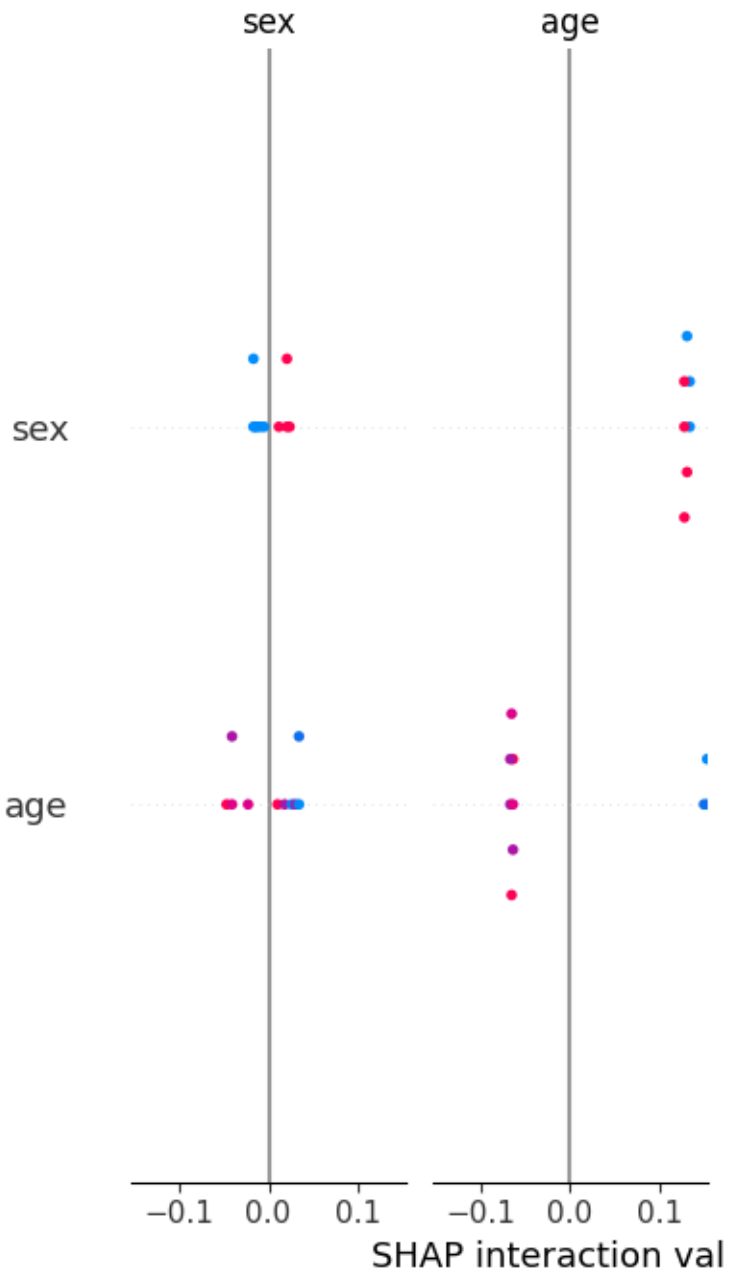
```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Save model
joblib.dump(model, '../models/chronic_model.pkl')
```

[5]:
```
['../models/chronic_model.pkl']
```

[6]:
```
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

[7]:
```
import shap
import matplotlib.pyplot as plt

# Explain model predictions
explainer = shap.Explainer(model, X)
shap_values = explainer(X)

# Visualize SHAP values
shap.summary_plot(shap_values, X)
```


sex
age
sex
age
−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val