

Task 2: Build a student success prediction model

Code:

```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

import seaborn as sns

import matplotlib.pyplot as plt


# Load success data

success_df = pd.read_csv("student_success_data.csv")


# Features and target

features = ['CGPA', 'Projects', 'Workshops', 'Participated']

X = success_df[features]

y = success_df['Success']


# Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

```
# Feature scaling
```

```
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

```
# Model
```

```
model = RandomForestClassifier(random_state=42)
```

```
model.fit(X_train_scaled, y_train)
```

```
# Predictions
```

```
y_pred = model.predict(X_test_scaled)
```

```
# Evaluation
```

```
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
# Visualize confusion matrix
```

```
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
plt.figure(figsize=(6, 4))
```

```
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",  
xticklabels=['Fail', 'Success'], yticklabels=['Fail', 'Success'])
```

```
plt.xlabel("Predicted")
```

```
plt.ylabel("Actual")
```

```
plt.title("Confusion Matrix for Student Success Prediction")

plt.tight_layout()

plt.show()
```

Output:

Classification Report:

	precision	recall	f1-score	support
0	0.25	1.00	0.40	1
1	1.00	0.40	0.57	5
accuracy			0.50	6
macro avg	0.62	0.70	0.49	6
weighted avg	0.88	0.50	0.54	6

Confusion Matrix:
[[1 0]
[3 2]]

