

## ✅ Project Title:

### Predicting Internship Success Using Student Profiles and Participation Metrics

CODE:

# Step 1: Import libraries

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
```

# Step 2: Load data (sample structure)

# Assume you have a CSV file: 'internship\_data.csv'

# Columns: ['cgpa', 'skills\_score', 'attendance\_rate', 'projects\_completed', 'mentor\_score', 'success']

```
df = pd.read_csv('internship_data.csv')
```

# Step 3: Features and Target

```
X = df.drop('success', axis=1) # Features
```

```
y = df['success']           # Target (1 = Success, 0 = Fail)
```

# Step 4: Preprocessing

```
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)
```

# Step 5: Split Data

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

# Step 6: Train Model

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

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

# Step 7: Predict

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

# Step 8: Evaluate

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

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

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

OUTPUT:

	cgpa	skills_score	attendance_rate	projects_completed	mentor_score	success
--	------	--------------	-----------------	--------------------	--------------	---------

8.2	80	0.90	3	4.5	1
-----	----	------	---	-----	---

6.5	40	0.60	1	2.0	0
-----	----	------	---	-----	---

7.8	75	0.85	2	4.0	1
-----	----	------	---	-----	---

...	...	...	...	...	...
-----	-----	-----	-----	-----	-----

FLOWCHART:

Start

|



Collect Historical Data

(Student Profiles, Participation Metrics, Outcomes)

|



Clean & Preprocess Data

(Handle missing values, encode categories, normalize)

|



Explore Data (EDA)

(Visualize patterns, correlations, distributions)

|



Feature Engineering

(Create useful features from raw data)



Split Dataset

(Train/Test Split)



Train Machine Learning Models

(Logistic Regression, Random Forest, etc.)



Evaluate Models

(Accuracy, Precision, Recall, Confusion Matrix)



Select Best Model

(Based on performance metrics)



Predict Internship Success for New Students



(Optional) Deploy Model as Web App

(Flask / Streamlit Interface)



End

## Project Overview

- The goal is to use machine learning to **predict student success** in internship programs based on **profile features** (e.g., CGPA, skills) and **participation metrics** (e.g., attendance, mentor feedback).
  - This helps training & placement cells, educators, and companies identify suitable interns and guide underperforming students.
- 

## Objectives

- Analyze student data to find patterns linked to internship success.
  - Build a predictive model that classifies students as **likely to succeed** or **likely to struggle**.
  - Provide **data-driven recommendations** to improve readiness for internships.
- 

## Data Requirements

- **Profile Features:** CGPA, technical skills, communication skills, department, resume analysis.
  - **Participation Metrics:** Attendance in sessions, project completion, mentor ratings, assignment submissions.
  - **Target Variable:** Internship outcome (e.g., success/fail, score, rating).
- 

## Methods & Techniques

- **Data Cleaning & Preprocessing:** Handle missing values, encode categorical variables, scale numerical features.
  - **Exploratory Data Analysis (EDA):** Understand correlations and data distributions.
  - **Machine Learning Models:** Logistic Regression, Random Forest, XGBoost, SVM.
  - **Model Evaluation:** Accuracy, F1-score, Confusion Matrix, ROC-AUC.
- 

## Outcomes

- Predictive system for early identification of students needing support.
  - Customized training plans based on predicted weaknesses.
  - Insights into key success factors (e.g., skill level, engagement).
- 

## Tools & Libraries

- Python, pandas, NumPy

- scikit-learn, XGBoost
  - matplotlib, seaborn
  - (Optional) Streamlit or Flask for deployment
- 

#### Use Cases

- Placement departments can **recommend students** to companies based on predictions.
  - Educators can **offer remedial actions** for students predicted to struggle.
  - Students get **personalized feedback** on areas to improve before internships.
-