SARVAGNA INNOVATION PRIVATE LIMITED

📚 ML Internship Week 1: Internship Fit Predictor (Student Guide)

🌼 Objective

This task involves:

- Loading and preprocessing a dataset
- Exploring data through visualizations
- Training a Random Forest classifier
- Predicting whether a new student is a good fit

Build a machine learning model that predicts whether a student is a good fit for an internship based on their GPA, technical skills, project experience, certifications, and GitHub activity. This task introduces the full machine learning pipeline using Python on Google Colab.

📄 Tools Used

✅ Google Colab

Google Colab is an online platform that lets you write and run Python code from your browser. It provides free access to computing power (including GPUs), and you don't need to install anything.

1. Visit Google Colab
2. Sign in with your Google account
3. Create a new notebook and start coding!

Python Libraries

Library	Use
pandas	Used to load and manipulate structured data (CSV). Think of it like Excel for Python.
numpy	Supports mathematical operations and arrays. Often used behind the scenes.
matplotlib.pyplot	Generates basic graphs and plots such as line, bar, and scatter plots.
seaborn	A library built on matplotlib that creates beautiful statistical plots easily.
sklearn (scikit-learn)	Main machine learning library with tools for data preprocessing, modeling, evaluation, and prediction.

Dataset

The dataset you will use is `internship_fit_predictor_250_dataset.csv`. Each row is a student record, and we want to predict if that student is a good fit for an internship.

Columns (Features):

- **GPA**: Grade Point Average (e.g., 6.5, 8.2)
- **Skills**: Programming or technical skills (e.g., "Python, ML")
- **Preferred Domain**: Area the student is interested in (e.g., "Data Science")
- **Certifications**: Online courses completed (e.g., "Coursera ML")
- **Internships Completed**: Number of internships already done
- **Hackathon Participation**: Whether student joined hackathons (Yes/No)
- **GitHub Profile Score**: Activity score on GitHub (out of 10)

- **Selected**: Final outcome (Yes/No) — our prediction goal

♻️ Step-by-Step ML Workflow

① Upload the Dataset to Colab

```
import pandas as pd
df = pd.read_csv("internship_fit_predictor_250_dataset.csv")
df.head() # Preview first 5 rows
```

② Data Cleaning

- Drop unnecessary columns like Name, Projects
- Convert 'Yes'/'No' to 1/0 using .map()
- Convert strings to numbers using LabelEncoder
- Remove any missing data using df.dropna()

③ Feature Scaling

Scaling ensures all numerical values are in a similar range.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df[['GPA', 'GitHub_Profile_Score']] = scaler.fit_transform(df[['GPA', 'GitHub_Profile_Score']])
```

🖌️ Exploratory Data Analysis (EDA)

What is EDA?

Exploratory Data Analysis helps us understand the dataset better before training our model.

Common Plots Used:

- **Countplot**: Compares selected vs not selected students

```
sns.countplot(x='Selected', data=df)
```

- **Histplot**: Shows GPA distribution

```
sns.histplot(df['GPA'], kde=True)
```

- **Boxplot**: Spread of GitHub scores by selection

```
sns.boxplot(x='Selected', y='GitHub_Profile_Score', data=df)
```

- **Barplot**: Average selection rates across skills

```
sns.barplot(x='Skills', y='Selected', data=df)
```

- **Heatmap**: Correlation between features

```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

🤖 Understanding scikit-learn (sklearn)

`scikit-learn` provides all essential ML tools.

Modules You'll Use:

- `LabelEncoder`: Converts text to numbers
- `StandardScaler`: Scales numeric columns
- `train_test_split`: Divides data into training and testing sets
- `RandomForestClassifier`: Builds prediction model
- `accuracy_score`, `classification_report`, `confusion_matrix`: Evaluate model performance

📊 Model Training and Testing

What Is Training vs Testing?

- **Training data**: Used to teach the model (80%)
- **Testing data**: Used to see how well the model learned (20%)

Train-Test Split

```
from sklearn.model_selection import train_test_split
X = df.drop('Selected', axis=1)
y = df['Selected']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Model Training

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

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

Evaluation

```
from sklearn.metrics import accuracy_score, classification_report
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

🔮 Prediction Task

Steps:

1. Create a dictionary with student values
2. Encode string values
3. Normalize the data
4. Use the model to predict

```
new_student = {
 'GPA': 8.2,
 'Skills': 'Python, ML',
 'Preferred Domain': 'Data Science',
 'Certifications': 'Coursera ML',
 'Internships Completed': 1,
 'Hackathon Participation': 'Yes',
 'GitHub_Profile_Score': 7
}
```

Final Prediction

```
Use model.predict() or model.predict_proba()
```

- If probability > 0.75, student is a good fit

✉️ Submission Instructions

- Save your Colab notebook as .ipynb

- Export as PDF: File > Download > PDF
- Upload both files to the Google Form which will be shared with you on Friday

---

📈 Learning Outcomes

By the end of this internship task, you will:

- Understand the ML lifecycle from loading data to predicting outcomes
- Learn to clean and prepare datasets
- Visualize relationships and trends in the data
- Train and test an ML model using Random Forest
- Predict the suitability of students for internships
- Build confidence in applying ML logic independently

Note: This guide is designed for complete beginners. You are encouraged to experiment, think logically, and write the code on your own. Your trainer will support you if you get stuck.

Let's build your first ML-powered internship recommender system together!