**Sample code :**

**import pandas as pd**

**from sklearn.preprocessing import MultiLabelBinarizer, LabelEncoder**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import accuracy\_score**

**# Load the dataset**

**df = pd.read\_csv('career\_data.csv')**

**# Split skills into a list**

**df['Skills'] = df['Skills'].apply(lambda x: [skill.strip() for skill in x.split(',')])**

**# One-hot encode skills**

**mlb = MultiLabelBinarizer()**

**skills\_encoded = pd.DataFrame(mlb.fit\_transform(df['Skills']), columns=mlb.classes\_)**

**# Label encode interest area**

**le\_interest = LabelEncoder()**

**df['Interest\_Encoded'] = le\_interest.fit\_transform(df['Interest Area'])**

**# Combine all features**

**X = pd.concat([df[['10th %', '12th %', 'UG %', 'Interest\_Encoded']], skills\_encoded], axis=1)**

**# Encode the target (Career Path)**

**le\_target = LabelEncoder()**

**y = le\_target.fit\_transform(df['Career Path'])**

**# Split into train/test**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Train the model**

**model = RandomForestClassifier()**

**model.fit(X\_train, y\_train)**

**# Predict and evaluate**

**y\_pred = model.predict(X\_test)**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print("Accuracy:", accuracy)**

**# To predict new input**

**def predict\_career(new\_input):**

**# new\_input: dictionary with keys ['10th %', '12th %', 'UG %', 'Interest Area', 'Skills']**

**interest\_encoded = le\_interest.transform([new\_input['Interest Area']])[0]**

**skill\_vector = [1 if skill in new\_input['Skills'] else 0 for skill in mlb.classes\_]**

**input\_vector = [new\_input['10th %'], new\_input['12th %'], new\_input['UG %'], interest\_encoded] + skill\_vector**

**prediction = model.predict([input\_vector])**

**return le\_target.inverse\_transform(prediction)[0]**

**# Example use:**

**new\_user = {**

**'10th %': 82,**

**'12th %': 80,**

**'UG %': 78,**

**'Interest Area': 'Tech',**

**'Skills': ['Python', 'Java']**

**}**

**print("Predicted Career:", predict\_career(new\_user))**

----------------------------------------------------------------------------------------------------------------

**Python Code for Training the Career Path Predictor:**

**import pandas as pd**

**from sklearn.preprocessing import MultiLabelBinarizer, LabelEncoder**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import accuracy\_score**

**# Load your dataset**

**df = pd.read\_csv('career\_data\_large.csv')**

**# Split and clean skills column**

**df['Skills'] = df['Skills'].apply(lambda x: [skill.strip() for skill in x.split(',')])**

**# One-hot encode the skills**

**mlb = MultiLabelBinarizer()**

**skills\_encoded = pd.DataFrame(mlb.fit\_transform(df['Skills']), columns=mlb.classes\_)**

**# Label encode the Interest Area**

**le\_interest = LabelEncoder()**

**df['Interest\_Encoded'] = le\_interest.fit\_transform(df['Interest Area'])**

**# Combine all features**

**X = pd.concat([df[['10th %', '12th %', 'UG %', 'Interest\_Encoded']], skills\_encoded], axis=1)**

**# Encode the target (Career Path)**

**le\_target = LabelEncoder()**

**y = le\_target.fit\_transform(df['Career Path'])**

**# Split into train and test sets**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Train a Random Forest model**

**model = RandomForestClassifier()**

**model.fit(X\_train, y\_train)**

**# Predict and evaluate**

**y\_pred = model.predict(X\_test)**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print("✅ Model trained successfully!")**

**print("🎯 Accuracy:", round(accuracy \* 100, 2), "%")**

----------------------------------------------------------------------------------------------------------------

**Open your terminal and Run the Python File:**

**python career\_predictor.py**

**Output:**

✅ Model trained successfully!

🎯 Accuracy: 89.45 %

----------------------------------------------------------------------------------------------------------------

**Prediction Function (add at the bottom of your file):**

**# Prediction function**

**def predict\_career(new\_input):**

**interest\_encoded = le\_interest.transform([new\_input['Interest Area']])[0]**

**skill\_vector = [1 if skill in new\_input['Skills'] else 0 for skill in mlb.classes\_]**

**input\_vector = [new\_input['10th %'], new\_input['12th %'], new\_input['UG %'], interest\_encoded] + skill\_vector**

**prediction = model.predict([input\_vector])**

**return le\_target.inverse\_transform(prediction)[0]**

**# 🔍 Example usage:**

**new\_user = {**

**'10th %': 88,**

**'12th %': 86,**

**'UG %': 79,**

**'Interest Area': 'Tech',**

**'Skills': ['Python', 'Java']**

**}**

**print("🎓 Predicted Career:", predict\_career(new\_user))**

----------------------------------------------------------------------------------------------------------------

**To pen open the needed folder in the terminal:**

**cd C:\Users\junai\Desktop\your\_project\_folder**

**To know the elements in the file:**

**dir**

----------------------------------------------------------------------------------------------------------------

**StreamLit:**

**Install Streamlit:**

**pip install streamlit**

**Create a new file:**

**career\_app.py**

**Streamlit Python code:**

**"import streamlit as st**

**import pandas as pd**

**from sklearn.preprocessing import MultiLabelBinarizer, LabelEncoder**

**from sklearn.ensemble import RandomForestClassifier**

**import joblib**

**# Load dataset**

**df = pd.read\_csv('clean\_career\_data.csv')**

**# Preprocess**

**df['Skills'] = df['Skills'].apply(lambda x: [skill.strip() for skill in x.split(',')])**

**mlb = MultiLabelBinarizer()**

**skills\_encoded = pd.DataFrame(mlb.fit\_transform(df['Skills']), columns=mlb.classes\_)**

**le\_interest = LabelEncoder()**

**df['Interest\_Encoded'] = le\_interest.fit\_transform(df['Interest Area'])**

**X = pd.concat([df[['10th %', '12th %', 'UG %', 'Interest\_Encoded']], skills\_encoded], axis=1)**

**le\_target = LabelEncoder()**

**y = le\_target.fit\_transform(df['Career Path'])**

**# Train the model**

**model = RandomForestClassifier()**

**model.fit(X, y)**

**# 🎨 Streamlit App**

**st.title("🎓 Career Path Predictor (AI Project)")**

**st.write("Enter your academic info, interest, and skills to get a suggested career path.")**

**# User inputs**

**tenth = st.slider("10th Percentage", 50, 100, 80)**

**twelfth = st.slider("12th Percentage", 50, 100, 80)**

**ug = st.slider("UG Percentage", 50, 100, 80)**

**interest = st.selectbox("Interest Area", le\_interest.classes\_)**

**selected\_skills = st.multiselect("Select your skills", mlb.classes\_)**

**# Predict button**

**if st.button("Predict Career Path"):**

**interest\_encoded = le\_interest.transform([interest])[0]**

**skill\_vector = [1 if skill in selected\_skills else 0 for skill in mlb.classes\_]**

**input\_vector = [tenth, twelfth, ug, interest\_encoded] + skill\_vector**

**prediction = model.predict([input\_vector])**

**predicted\_career = le\_target.inverse\_transform(prediction)[0]**

**st.success(f"✅ Based on your input, the suggested career path is: \*\*{predicted\_career}\*\*")**

**"**

**Run Streamlit in terminal:**

**streamlit run career\_app.py**

----------------------------------------------------------------------------------------------------------------