

✓
16s [1]

```
# Step 1: Install dependencies
!pip install scikit-learn pandas

# Step 2: Import libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import joblib
from google.colab import files
```



```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.15.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```



Step 3: Load dataset and train model

```
url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv'
columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
           'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']
df = pd.read_csv(url, names=columns)

X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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



RandomForestClassifier ⓘ ?

RandomForestClassifier()

[4]

```
# Step 4: Evaluate model
accuracy = accuracy_score(y_test, model.predict(X_test))
print(f"Model Accuracy: {accuracy * 100:.2f}%")
```



Model Accuracy: 72.73%

```
[8] # Step 5: Save model
joblib.dump(model, 'diabetes_model.pkl')

# Step 6: Upload user CSV file
print("\nPlease upload a CSV file with the following columns:\n")
print(columns[:-1]) # Show required columns (excluding 'Outcome')

uploaded = files.upload()
```



Please upload a CSV file with the following columns:

['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

diabetes_model.pkl.csv

- **diabetes_model.pkl.csv**(text/csv) - 23873 bytes, last modified: 5/8/2025 - 100% done

Saving diabetes_model.pkl.csv to diabetes_model.pkl.csv

```
[9] # Step 7: Load uploaded file
for filename in uploaded.keys():
    user_df = pd.read_csv(filename)
    print(f"\nUploaded Data Preview:\n{user_df.head()}")
```



Uploaded Data Preview:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

[12]

```
#Step 9: Show results
user_df['DiabetesPrediction'] = ['Likely Diabetic' if p == 1 else 'Unlikely Diabetic' for p in predictions]
print("\nPrediction Results:\n")
print(user_df[['DiabetesPrediction']])
```



Prediction Results:

	DiabetesPrediction
0	Likely Diabetic
1	Unlikely Diabetic
2	Likely Diabetic
3	Unlikely Diabetic
4	Likely Diabetic
..	...
763	Unlikely Diabetic
764	Unlikely Diabetic
765	Unlikely Diabetic
766	Likely Diabetic
767	Unlikely Diabetic

[768 rows x 1 columns]

```
[13] # Optional: Download results
user_df.to_csv("prediction_results.csv", index=False)
files.download("prediction_results.csv")
```

```
[+]
```