```
from google.colab import files
uploaded = files.upload()
```

Choose Files train.csv

- **train.csv**(text/csv) - 2029405 bytes, last modified: 12/11/2019 - 100% done

```
Saving train.csv to train.csv
```

```
import pandas as pd

df = pd.read_csv("train.csv")
df.head()
```

	Id	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadways	Hill
0	1	2596	51	3	258	0	510	
1	2	2590	56	2	212	-6	390	
2	3	2804	139	9	268	65	3180	
3	4	2785	155	18	242	118	3090	
4	5	2595	45	2	153	-1	391	

5 rows × 56 columns

```
# Basic info
print("Dataset shape:", df.shape)
print("\nColumn types:\n", df.dtypes)

# Check for missing values
print("\nMissing values:\n", df.isnull().sum())

# Check target column name and unique values
print("\nColumn names:\n", df.columns.tolist())
```

```
Soil_Type28                           0
Soil_Type29                           0
Soil_Type30                           0
Soil_Type31                           0
Soil_Type32                           0
Soil_Type33                           0
Soil_Type34                           0
Soil_Type35                           0
Soil_Type36                           0
Soil_Type37                           0
Soil_Type38                           0
Soil_Type39                           0
Soil_Type40                           0
Cover_Type                            0
dtype: int64

Column names:
 ['Id', 'Elevation', 'Aspect', 'Slope', 'Horizontal_Distance_To_Hydrology', 'Vertical_Distance_To_Hydrology', 'Horizontal_Distance_To_
```

```
from sklearn.model_selection import train_test_split

# Drop the ID column
df = df.drop(columns=['Id'])

# Split features and target
X = df.drop(columns=['Cover_Type'])
y = df['Cover_Type']

# Train-test split (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Train shape:", X_train.shape)
print("Test shape:", X_test.shape)
```

```
Train shape: (12096, 54)
Test shape: (3024, 54)
```

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

# Train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict on test set
y_pred = model.predict(X_test)

# Evaluation
print("✅ Accuracy:", accuracy_score(y_test, y_pred))
print("\n📊 Classification Report:\n", classification_report(y_test, y_pred))
print("📉 Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
✅ Accuracy: 0.8723544973544973

📊 Classification Report:
               precision    recall  f1-score   support

           1       0.79      0.78      0.79       421
           2       0.83      0.72      0.77       438
           3       0.83      0.83      0.83       428
           4       0.93      0.98      0.96       449
           5       0.89      0.96      0.92       416
           6       0.86      0.86      0.86       432
           7       0.95      0.97      0.96       440

    accuracy                           0.87      3024
   macro avg       0.87      0.87      0.87      3024
weighted avg       0.87      0.87      0.87      3024

📉 Confusion Matrix:
 [[327  58   0   0  15   0  21]
 [ 74 314  15   0  30   4   1]
 [  0   0 357  20   2  49   0]
 [  0   0   6 441   0   2   0]
 [  1   3   9   0 400   3   0]
 [  0   3  42  13   3 371   0]
 [ 10   2   0   0   0   0 428]]
```

```
df.to_csv("forest_cover_dataset.csv", index=False)
```