

```
In [1]: import pandas as pd
```

```
In [3]: file_path = "Downloads/handwritten_character_dataset.csv"
df = pd.read_csv(file_path)
```

```
In [29]: print(df.info())
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Columns: 786 entries, image_id to label
dtypes: int64(785), object(1)
memory usage: 6.0+ MB
None
```

	image_id	pixel_0	pixel_1	pixel_2	pixel_3	pixel_4	pixel_5	pixel_6	\
0	1	181	154	241	220	82	51	74	
1	2	186	49	90	157	247	58	191	
2	3	86	215	178	16	234	141	168	
3	4	221	27	78	167	216	85	184	
4	5	3	204	159	112	19	27	227	

	pixel_7	pixel_8	...	pixel_775	pixel_776	pixel_777	pixel_778	\
0	244	96	...	237	169	8	103	
1	79	85	...	17	187	5	103	
2	53	164	...	252	177	184	89	
3	223	234	...	97	92	145	251	
4	15	161	...	176	208	106	16	

	pixel_779	pixel_780	pixel_781	pixel_782	pixel_783	label
0	230	166	20	60	41	y
1	75	255	209	224	213	u
2	48	214	205	56	169	v
3	98	3	203	121	214	F
4	254	228	153	233	20	R

[5 rows x 786 columns]

```
In [7]: print(df.isnull().sum())
```

```
image_id      0
pixel_0        0
pixel_1        0
pixel_2        0
pixel_3        0
..
pixel_780      0
pixel_781      0
pixel_782      0
pixel_783      0
label          0
Length: 786, dtype: int64
```

```
In [9]: df = df.dropna().reset_index(drop=True)
```

```
In [11]: import numpy as np

X = df.iloc[:, 1:-1].values.astype(np.uint8)
y = df.iloc[:, -1].values
```

```
print("Shape of X:", X.shape)
print("Shape of y:", y.shape)
```

Shape of X: (1000, 784)
Shape of y: (1000,)

```
In [13]: X = X / 255.0
```

```
In [15]: from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

```
In [17]: print("Unique labels:", label_encoder.classes_)
```

Unique labels: ['0' '1' '2' '3' '4' '5' '6' '7' '8' '9' 'A' 'B' 'C' 'D' 'E' 'F'
'G' 'H'
'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z'
'a' 'b' 'c' 'd' 'e' 'f' 'g' 'h' 'i' 'j' 'k' 'l' 'm' 'n' 'o' 'p' 'q' 'r'
's' 't' 'u' 'v' 'w' 'x' 'y' 'z']

```
In [19]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2,

print("Training data shape:", X_train.shape)
print("Testing data shape:", X_test.shape)
```

Training data shape: (800, 784)
Testing data shape: (200, 784)

```
In [21]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

```
In [23]: clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
```

```
Out[23]: ▼ RandomForestClassifier ⓘ ?
RandomForestClassifier(random_state=42)
```

```
In [24]: y_pred = clf.predict(X_test)
```

```
In [27]: accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")
```

Model Accuracy: 0.01

```
In [ ]:
```