





```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
```

```
column_names = ['sepal.length', 'sepal.width', 'petal.length', 'petal.width', 'Species']
df = pd.read_csv('Iris.csv', header=None, names=column_names)
df
```




| | sepal.length | sepal.width | petal.length | petal.width | Species |  |
|------------|--------------|-------------|--------------|-------------|----------------|---|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |  |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |  |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa | |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa | |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa | |
| ... | ... | ... | ... | ... | ... | |
| 145 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica | |
| 146 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica | |
| 147 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica | |
| 148 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica | |
| 149 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica | |

150 rows × 5 columns

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```
df_encoded = pd.get_dummies(df, columns=['Species'])
```

```
print("\nOne-Hot Encoded Dataset:\n", df_encoded.head())
```



One-Hot Encoded Dataset:

| | sepal.length | sepal.width | petal.length | petal.width | Species_Iris-setosa \ |
|---|--------------|-------------|--------------|-------------|-----------------------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | True |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | True |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | True |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | True |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | True |

| | Species_Iris-versicolor | Species_Iris-virginica |
|---|-------------------------|------------------------|
| 0 | False | False |
| 1 | False | False |
| 2 | False | False |
| 3 | False | False |
| 4 | False | False |

✓ 2. Perform the One Hot encoding , by considering Species as target variable.

```
# Initialize OneHotEncoder with correct parameter
encoder = OneHotEncoder(sparse_output=False) # Use 'sparse_output' instead of 'sparse'
encoded_species = encoder.fit_transform(df[['Species']])

# Convert encoded data into a DataFrame
encoded_df = pd.DataFrame(encoded_species, columns=encoder.get_feature_names_out(['Specie

# Concatenate with original dataset (excluding original Species column)
df_final = pd.concat([df.drop(columns=['Species']), encoded_df], axis=1)

print("\nOne-Hot Encoded DataFrame:\n", df_final.head())
```



One-Hot Encoded DataFrame:

| | sepal.length | sepal.width | petal.length | petal.width | Species_Iris-setosa \ |
|---|--------------|-------------|--------------|-------------|-----------------------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | 1.0 |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | 1.0 |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | 1.0 |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | 1.0 |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | 1.0 |

| | Species_Iris-versicolor | Species_Iris-virginica |
|---|-------------------------|------------------------|
| 0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 |

✓ What is One-Hot Encoding?

One-hot encoding is a technique used to convert categorical data into a numerical format for machine learning models. It transforms categorical values into separate binary columns, preventing models from mistakenly assigning numerical meaning to categories.

How It Works

Instead of assigning a single numerical label (as in label encoding), one-hot encoding creates new columns, each representing a unique category. The presence of a category is indicated by 1, while the absence is 0.

Using one-hot encoding, this would be transformed into: | Color | Red | Blue | Green |

| Red | 1 | 0 | 0 |

| Blue | 0 | 1 | 0 |

| Green | 0 | 0 | 1 |

| Red | 1 | 0 | 0 |

| Green | 0 | 0 | 1 |

Why Use One-Hot Encoding?

- Avoids Ordinal Assumptions: Unlike label encoding, it prevents unintended ranking relationships (e.g., Red \neq 0, Blue \neq 1, Green \neq 2).
- Compatible with ML Models: Some models, like linear regression, perform better when categorical values are one-hot encoded.