




```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```
data=pd.read_csv("/content/Iris.csv")
```

data



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica




150 rows × 6 columns

Next steps:


[Generate code with data](#)

 [View recommended plots](#)

data.head()



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa




Next steps:

[Generate code with data](#)


 [View recommended plots](#)



data.info()




```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0    Id              150 non-null   int64
1    SepalLengthCm   150 non-null   float64
2    SepalWidthCm    150 non-null   float64
3    PetalLengthCm   150 non-null   float64
4    PetalWidthCm    150 non-null   float64
5    Species         150 non-null   object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

data.describe()



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	
count	150.000000	150.000000	150.000000	150.000000	150.000000	
mean	75.500000	5.843333	3.054000	3.758667	1.198667	
std	43.445368	0.828066	0.433594	1.764420	0.763161	
min	1.000000	4.300000	2.000000	1.000000	0.100000	
25%	38.250000	5.100000	2.800000	1.600000	0.300000	
50%	75.500000	5.800000	3.000000	4.350000	1.300000	
75%	112.750000	6.400000	3.300000	5.100000	1.800000	
max	150.000000	7.900000	4.400000	6.900000	2.500000	

```
data.shape
```




(150, 6)

```
data.size
```




900



```
data.columns
```




```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',  
      'Species'],  
      dtype='object')
```


```
data.tail()
```



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	
145	146	6.7	3.0	5.2	2.3	Iris-virginica	
146	147	6.3	2.5	5.0	1.9	Iris-virginica	
147	148	6.5	3.0	5.2	2.0	Iris-virginica	
148	149	6.2	3.4	5.4	2.3	Iris-virginica	
149	150	5.9	3.0	5.1	1.8	Iris-virginica	

```
data.isnull()
```




	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	
145	False	False	False	False	False	False	
146	False	False	False	False	False	False	
147	False	False	False	False	False	False	
148	False	False	False	False	False	False	
149	False	False	False	False	False	False	




150 rows × 6 columns

```
X=data.drop (columns=['Id', 'Species'])  
y=data['Species']
```

X



	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8






150 rows × 4 columns

Next steps:

[Generate code with X](#)[View recommended plots](#)

y




```

0      Iris-setosa
1      Iris-setosa
2      Iris-setosa
3      Iris-setosa
4      Iris-setosa
...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object

```

```
X=data.drop(columns=['Id', 'Species'])
```

```
data.columns
```



```


Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')

```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
knn=KNeighborsClassifier(n_neighbors=3)
```

```
knn.fit(X_train, y_train)
```



```

KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)


```

```
y_pred = knn.predict(X_test)
```

```

accuracy =accuracy_score(y_test,y_pred)
print("Accuracy:", accuracy)

```



```
Accuracy: 1.0
```

```

import pickle
with open('iris_model.pkl', 'wb') as file:
    pickle.dump(data, file)

```

