```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd

iris = load_iris()

df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\
n    {\n      \"column\": \"sepal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8280661279778629,\n        \"min\": 4.3,\n        \"max\": 7.9,\n
\"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n
4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.435866284936698,\n        \"min\":
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n
\"samples\": [\n          2.3,\n          4.0,\n          3.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7652982332594667,\n        \"min\": 1.0,\n        \"max\": 6.9,\n
\"num_unique_values\": 43,\n        \"samples\": [\n          6.7,\n
3.8,\n          3.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.7622376689603465,\n        \"min\":
0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n
\"samples\": [\n          0.2,\n          1.2,\n          1.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
df['target'] = iris.target

df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\
n    {\n      \"column\": \"sepal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8280661279778629,\n        \"min\": 4.3,\n        \"max\": 7.9,\n
\"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n
4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.435866284936698,\n        \"min\":
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n
```

\"samples\": [\n   2.3,\n   4.0,\n   3.5\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"petal length (cm)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.7652982332594667,\n   \"min\": 1.0,\n   \"max\": 6.9,\n
\"num_unique_values\": 43,\n   \"samples\": [\n   6.7,\n
3.8,\n   3.7\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"petal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.7622376689603465,\n   \"min\":
0.1,\n   \"max\": 2.5,\n   \"num_unique_values\": 22,\n
\"samples\": [\n   0.2,\n   1.2,\n   1.3\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"target\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 0,\n
\"min\": 0,\n   \"max\": 2,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   0,\n   1,\n   2\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.tail()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 5,\n \"fields\": [\n
{\n   \"column\": \"sepal length (cm)\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.30331501776206193,\n
\"min\": 5.9,\n   \"max\": 6.7,\n   \"num_unique_values\":
5,\n   \"samples\": [\n   6.3,\n   5.9,\n
6.5\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sepal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.31937438845342625,\n   \"min\":
2.5,\n   \"max\": 3.4,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   3.0,\n   2.5,\n   3.4\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"petal length (cm)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.14832396974191348,\n   \"min\": 5.0,\n   \"max\": 5.4,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   5.0,\n
5.1,\n   5.2\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"petal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.23021728866442667,\n   \"min\":
1.8,\n   \"max\": 2.3,\n   \"num_unique_values\": 4,\n
\"samples\": [\n   1.9,\n   1.8,\n   2.3\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"target\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 0,\n
\"min\": 2,\n   \"max\": 2,\n   \"num_unique_values\": 1,\n
\"samples\": [\n   2\n   ],\n   \"semantic_type\":

```
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```

```
df['species'] = df['target'].map({0: 'setosa', 1: 'versicolor', 2:
'virginica'})
```

```
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\
n    {\n      \"column\": \"sepal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8280661279778629,\n        \"min\": 4.3,\n        \"max\": 7.9,\n
\"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n
4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.435866284936698,\n        \"min\":
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n
\"samples\": [\n          2.3,\n          4.0,\n          3.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7652982332594667,\n        \"min\": 1.0,\n        \"max\": 6.9,\n
\"num_unique_values\": 43,\n        \"samples\": [\n          6.7,\n
3.8,\n          3.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.7622376689603465,\n        \"min\":
0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n
\"samples\": [\n          0.2,\n          1.2,\n          1.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"target\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 2,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          0,\n          1,\n          2\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"species\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"setosa\",\n
\"versicolor\",\n          \"virginica\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
df.tail()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"sepal length (cm)\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.30331501776206193,\n
\"min\": 5.9,\n        \"max\": 6.7,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          6.3,\n          5.9,\n
```

```
6.5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.31937438845342625,\n        \"min\":
2.5,\n        \"max\": 3.4,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          3.0,\n          2.5,\n          3.4\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.14832396974191348,\n        \"min\": 5.0,\n        \"max\": 5.4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          5.0,\n
5.1,\n          5.2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.23021728866442667,\n        \"min\":
1.8,\n        \"max\": 2.3,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          1.9,\n          1.8,\n          2.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"target\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 2,\n        \"max\": 2,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          2\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"species\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          \"virginica\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```


```
X = df.drop(['target', 'species'], axis=1)
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

clf = DecisionTreeClassifier()

clf.fit(X_train, y_train)
```

```
DecisionTreeClassifier()
```

```
predictions = clf.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

```
print("\nEnter flower details to predict species:")
sepal_length = float(input("Sepal length (cm): "))
sepal_width = float(input("Sepal width (cm): "))
petal_length = float(input("Petal length (cm): "))
```

```
petal_width = float(input("Petal width (cm): "))
user_input = [[sepal_length, sepal_width, petal_length, petal_width]]
prediction = clf.predict(user_input)[0]
predicted_species = iris.target_names[prediction]
print("\n The predicted species is:", predicted_species)


Enter flower details to predict species:
Sepal length (cm): 3
Sepal width (cm): 2
Petal length (cm): 1
Petal width (cm): 4

---------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
/tmp/ipython-input-4-433288988.py in <cell line: 0>()
      5 petal_width = float(input("Petal width (cm): "))
      6 user_input = [[sepal_length, sepal_width, petal_length,
petal_width]]
----> 7 prediction = clf.predict(user_input)[0]
      8 predicted_species = iris.target_names[prediction]
      9 print("\n The predicted species is:", predicted_species)

NameError: name 'clf' is not defined

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
import pandas as pd

# Load data and create DataFrame
iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['target'] = iris.target
df['species'] = df['target'].map({0: 'setosa', 1: 'versicolor', 2:
'virginica'})


print("\nEnter flower details to predict species:")
sepal_length=float(input("sepal length(cm):"))
sepal_width=float(input("sepal width(cm):"))
petal_length=float(input("petal length(cm):"))
petal_width=float(input("petal width(cm):"))
user_input = [[sepal_length, sepal_width, petal_length, petal_width]]

# Define clf here
clf = DecisionTreeClassifier()
# Fit the classifier
X = df.drop(['target', 'species'], axis=1)
```

```
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
clf.fit(X_train, y_train)

prediction = clf.predict(user_input)[0]
predicted_species = iris.target_names[prediction]
print("\n The predicted species is:", predicted_species)


Enter flower details to predict species:
sepal length(cm):3
sepal width(cm):2
petal length(cm):1
petal width(cm):4

 The predicted species is: setosa

/usr/local/lib/python3.11/dist-packages/sklearn/utils/
validation.py:2739: UserWarning: X does not have valid feature names,
but DecisionTreeClassifier was fitted with feature names
  warnings.warn(

print(iris.target_names)

['setosa' 'versicolor' 'virginica']
```