

```
Y = pd.DataFrame(dataset.target)  
Y
```

	0
0	0
1	0
2	0
3	0
4	0
...	...
145	2
146	2
147	2
148	2
149	2

150 rows × 1 columns

Next steps: [Generate code with Y](#)[View recommended plots](#)

```
X = pd.DataFrame(dataset.data, columns=dataset.feature_names)
X
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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](#)

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
print(X_train.shape)
print(X_test.shape)
```

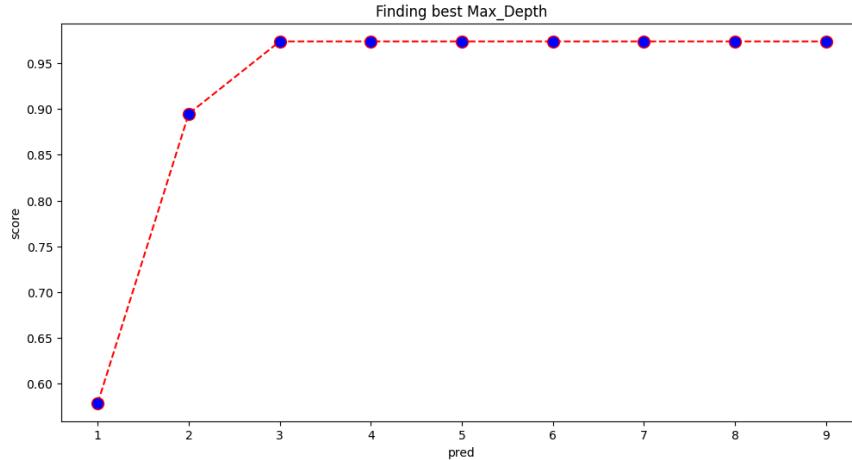
→ (112, 4)
 (38, 4)

```
accuracy = []
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

for i in range(1, 10):
    model = DecisionTreeClassifier(max_depth = i, random_state = 0)
    model.fit(X_train, y_train)
    pred = model.predict(X_test)
    score = accuracy_score(y_test, pred)
    accuracy.append(score)

plt.figure(figsize=(12, 6))
plt.plot(range(1, 10), accuracy, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Finding best Max_Depth')
plt.xlabel('pred')
plt.ylabel('score')
```

Text(0, 0.5, 'score')



```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion = 'entropy', max_depth = 3, random_state = 0)
model.fit(X_train,y_train)
```

DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)

```
y_pred = model.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

AttributeError Traceback (most recent call last)
`<ipython-input-9-f7e4d1143014>` in <cell line: 2>()
 1 y_pred = model.predict(X_test)
----> 2 print(np.concatenate((y_pred.reshape(len(y_pred),1),
 y_test.reshape(len(y_test),1)),1))

```
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in __getattr__(self, name)
      5987      ):
  5988          return self[name]
-> 5989          return object.__getattribute__(self, name)
  5990
  5991      def __setattr__(self, name: str, value) -> None:
```

AttributeError: 'DataFrame' object has no attribute 'reshape'

Next steps: [Explain error](#)

```
from sklearn.metrics import accuracy_score
print("Accuracy of the Model: {}%".format(accuracy_score(y_test, y_pred)*100))
```

Accuracy of the Model: 97.36842105263158%

```
# prompt:
```

```
# Making Predictions
new_pred = model.predict([[5, 2.9, 1, 0.2]])
print(new_pred)
```

[0]
`/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClass`
`warnings.warn(`

