

```
In [6]: import pandas as pd
import numpy as np

df = pd.read_csv('/Users/juanrquilesjr/Downloads/UCI_MachineLearningDataSets/Connect-4_data/connect-4.data', nam
               'b5', 'b6', 'c1', 'c2', 'c3', 'c4', 'c5', 'c6', 'd1', 'd2',
               'd3', 'd4', 'd5', 'd6', 'e1', 'e2', 'e3', 'e4', 'e5',
               'e6', 'f1', 'f2', 'f3', 'f4', 'f5', 'fg', 'g1', 'g2',
               'g3', 'g4', 'g5', 'g6', 'Target'])
pd.set_option('display.max_columns', 45)

df.index = df.index + 1
df.head(5)
```

```
Out[6]:
```

	a1	a2	a3	a4	a5	a6	b1	b2	b3	b4	b5	b6	c1	c2	c3	c4	c5	c6	d1	d2	d3	d4	d5	d6	e1	e2	e3	e4	e5	e6	f1	f2	f3	f4	f5	f
1	b	b	b	b	b	b	b	b	b	b	b	b	x	o	b	b	b	b	x	o	x	o	x	o	b	b	b	b	b	b	b	b	b	b	b	b
2	b	b	b	b	b	b	b	b	b	b	b	b	x	b	b	b	b	b	x	o	x	o	x	o	o	b	b	b	b	b	b	b	b	b	b	b
3	b	b	b	b	b	b	o	b	b	b	b	b	x	b	b	b	b	b	x	o	x	o	x	o	b	b	b	b	b	b	b	b	b	b	b	b
4	b	b	b	b	b	b	b	b	b	b	b	b	x	b	b	b	b	b	x	o	x	o	x	o	b	b	b	b	b	b	o	b	b	b	b	b
5	o	b	b	b	b	b	b	b	b	b	b	b	x	b	b	b	b	b	x	o	x	o	x	o	b	b	b	b	b	b	b	b	b	b	b	b

```
In [8]: #initializing label encoder and encoding df string values to numeric
from sklearn import preprocessing #Import label encoder

le = preprocessing.LabelEncoder()
df_encoded = df.apply(le.fit_transform)

#splitting data into feature and target sets
features = df_encoded.drop(['Target'], axis = 1).values # drop target columns to create the features data set
target = df_encoded['Target'].values
```

```
In [11]: #splitting data into training and test sets
from sklearn.model_selection import train_test_split # Import train_test_split function

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.3, random_state=1)
```

```
In [22]: #Import Random Forest Model
        from sklearn.ensemble import RandomForestClassifier

        #Create a Gaussian Classifier
        clf=RandomForestClassifier(n_estimators=100)

        #Train the model using the training sets y_pred=clf.predict(X_test)
        clf.fit(X_train,y_train)

        y_pred=clf.predict(X_test)
```

```
In [23]: #Import scikit-learn metrics module for accuracy calculation
        from sklearn import metrics

        # Model Accuracy, how often is the classifier correct?
        print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.8111308466548254

```
In [ ]:
```