

In [*]: *# import random search, random forest, iris data, and distributions*

```
%matplotlib notebook
from sklearn.model_selection import cross_validate
from sklearn import datasets
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_digits
```

In [*]: `import pandas as pd`
`data = pd.read_csv('HaitiPixels_good_01.csv')`
`data.head()`

In [*]: `from sklearn import datasets`
`X=data[['Red', 'Green', 'Blue']] # Features`
`y=data['Type'] # Labels`
`X.columns = ['Red','Green','Blue']`
`y.columns = ['Target']`

<https://www.kaggle.com/diegosch/classifier-evaluation-using-confusion-matrix>
<https://www.kaggle.com/diegosch/classifier-evaluation-using-confusion-matrix>

In [*]: *# Split dataset into training set and test set*
`from sklearn.model_selection import train_test_split`
`x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3) # 70% train`

In [22]: `from sklearn.svm import SVC`
`from sklearn.metrics import accuracy_score, confusion_matrix, precision_recall_f`

`clf = SVC(kernel = 'linear').fit(x_train,y_train)`
`clf.predict(x_train)`
`y_pred = clf.predict(x_test)`

Creates a confusion matrix
`cm = confusion_matrix(y_test, y_pred)`
`cm`

Out[22]: `array([[2404, 237],`
 `[18, 311914]], dtype=int64)`

In [5]: *# Split dataset into training set and test set*
`from sklearn.model_selection import train_test_split`
`X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) # 70% train`

```
In [*]: from sklearn.metrics import confusion_matrix
svm = SVC(kernel='linear', C=1).fit(X_train, y_train)
svm_predicted = svm.predict(X_test)
confusion = confusion_matrix(y_test, svm_predicted)

print('Support vector machine classifier (linear kernel, C=1)\n', confusion)
```

```
In [ ]:
```