In [14]:

```
# import random search, random forest, iris data, and distributions
from sklearn.model_selection import cross_validate
from sklearn import datasets
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import seaborn as sns
sns.set()
```

In [15]:

```
import pandas as pd
data = pd.read_csv('HaitiPixels_good.csv')
data.head()
```

Out[15]:

	Type	Red	Green	Blue
0	nonblue	104	89	63
1	nonblue	101	80	60
2	nonblue	103	87	69
3	nonblue	107	93	72
4	nonblue	109	99	68

In [16]:

```
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 [17]:

```
# 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%
```

In [18]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
```

In [19]:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5, metric='euclidean')
knn.fit(X_train, y_train)
```

Out[19]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='euclidean',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

In [21]:

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

In [23]:

```
confusion_matrix(y_test, y_pred)
```

Out[23]:

```
array([[  2225,      9],
       [    27, 259883]], dtype=int64)
```

https://rstudio-pubs-static.s3.amazonaws.com/71575_4068e2e6dc3d46a785ad7886426c37db.html (https://rstudio-pubs-static.s3.amazonaws.com/71575_4068e2e6dc3d46a785ad7886426c37db.html)

https://medium.com/@hjhuney/implementing-a-random-forest-classification-model-in-python-583891c99652 (https://medium.com/@hjhuney/implementing-a-random-forest-classification-model-in-python-583891c99652)

In [22]:

```
sns.scatterplot(
    x='mean area',
    y='mean compactness',
    hue='benign',
    data=X_test.join(y_test, how='outer')
)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-22-004a12312523> in <module>
      3     y='mean compactness',
      4     hue='benign',
----> 5     data=X_test.join(y_test, how='outer')
      6 )

~\Anaconda3\lib\site-packages\seaborn\relational.py in scatterplot(x, y, hue, s
tyle, size, data, palette, hue_order, hue_norm, sizes, size_order, size_norm, m
arkers, style_order, x_bins, y_bins, units, estimator, ci, n_boot, alpha, x_jit
ter, y_jitter, legend, ax, **kwargs)
   1333         x_bins=x_bins, y_bins=y_bins,
   1334         estimator=estimator, ci=ci, n_boot=n_boot,
-> 1335         alpha=alpha, x_jitter=x_jitter, y_jitter=y_jitter, legend=legen
d,
   1336     )
   1337 

~\Anaconda3\lib\site-packages\seaborn\relational.py in __init__(self, x, y, hu
e, size, style, data, palette, hue_order, hue_norm, sizes, size_order, size_nor
m, dashes, markers, style_order, x_bins, y_bins, units, estimator, ci, n_boot,
 alpha, x_jitter, y_jitter, legend)
    850 
    851         plot_data = self.establish_variables(
--> 852             x, y, hue, size, style, units, data
    853         )
    854 

~\Anaconda3\lib\site-packages\seaborn\relational.py in establish_variables(sel
f, x, y, hue, size, style, units, data)
    140                 if isinstance(var, string_types):
    141                     err = "Could not interpret input '{}'".format(var)
--> 142                     raise ValueError(err)
    143 
    144             # Extract variable names

ValueError: Could not interpret input 'mean area'
```

In [ ]:

```
plt.scatter(
    X_test['mean area'],
    X_test['mean compactness'],
    c=y_pred,
    cmap='coolwarm',
    alpha=0.7
)
```

In [ ]:

```
confusion_matrix(y_test, y_pred)
```

https://towardsdatascience.com/k-nearest-neighbor-python-2fccc47d2a55 (https://towardsdatascience.com/k-nearest-neighbor-python-2fccc47d2a55)

In [ ]: