Model Decision Tree:

#1

```
DecisionTreeClassifier(max_depth=10, min_samples_leaf=10,
min_samples_split=3,random_state=42, splitter='best')

Decision Tree Accuracy: 0.9473684210526315
Decision Tree Confusion Matrix:
[[39  4]
 [ 2 69]]
Decision Tree Precision: 0.9452054794520548
Decision Tree Recall: 0.971830985915493
Decision Tree F1 Score: 0.9583333333333334
```

#2

```
DecisionTreeClassifier(max_depth=1000, min_samples_split=2,
min_samples_leaf=10, random_state=42, splitter='best')

Decision Tree Accuracy: 0.9473684210526315
Decision Tree Confusion Matrix:
[[39  4]
 [ 2 69]]
Decision Tree Precision: 0.9452054794520548
Decision Tree Recall: 0.971830985915493
Decision Tree F1 Score: 0.9583333333333334
```

#3

```
DecisionTreeClassifier(max_depth=1000, min_samples_leaf=10,
splitter='best',min_samples_split=15, random_state=42)
Decision Tree Accuracy: 0.9473684210526315
Decision Tree Confusion Matrix:
[[39  4]
 [ 2 69]]
Decision Tree Precision: 0.9452054794520548
Decision Tree Recall: 0.971830985915493
Decision Tree F1 Score: 0.9583333333333334
```

#4

```
DecisionTreeClassifier(max_depth=1000, min_samples_leaf=2,
min_samples_split=30,random_state=42, splitter='best')

Decision Tree Accuracy: 0.9385964912280702
Decision Tree Confusion Matrix:
[[39  4]
 [ 3 68]]
Decision Tree Precision: 0.9444444444444444
Decision Tree Recall: 0.9577464788732394
Decision Tree F1 Score: 0.951048951048951
```

#5

```
DecisionTreeClassifier(max_depth=1000, min_samples_leaf=2,
min_samples_split=30,random_state=42, splitter='random')

Decision Tree Accuracy: 0.9649122807017544
Decision Tree Confusion Matrix:
[[40  3]
 [ 1 70]]
Decision Tree Precision: 0.958904109589041
Decision Tree Recall: 0.9859154929577465
Decision Tree F1 Score: 0.9722222222222222
```

Random Forest Model:

#1

```
RandomForestClassifier(max_depth=10, min_samples_leaf=10,
min_samples_split=10,n_estimators=1000, random_state=42)

Random Forest Accuracy: 0.9649122807017544
Random Forest Confusion Matrix:
[[40  3]
 [ 1 70]]
Random Forest Precision: 0.958904109589041
Random Forest Recall: 0.9859154929577465
Random Forest F1 Score: 0.9722222222222222
```

#2

```
RandomForestClassifier(max_depth=10, min_samples_leaf=10,
min_samples_split=10,n_estimators=10, random_state=42)

Random Forest Accuracy: 0.9649122807017544
Random Forest Confusion Matrix:
[[40  3]
 [ 1 70]]
Random Forest Precision: 0.958904109589041
Random Forest Recall: 0.9859154929577465
Random Forest F1 Score: 0.9722222222222222
```

#3

```
RandomForestClassifier(max_depth=1000, min_samples_leaf=10,
                       min_samples_split=10, n_estimators=1000,
                       random_state=42)
Random Forest Accuracy: 0.9649122807017544
Random Forest Confusion Matrix:
[[40  3]
 [ 1 70]]
Random Forest Precision: 0.958904109589041
Random Forest Recall: 0.9859154929577465
Random Forest F1 Score: 0.9722222222222222
```

#4

```
RandomForestClassifier(max_depth=1000, min_samples_leaf=100,
                       min_samples_split=10, n_estimators=1000,
                       random_state=42)
Random Forest Accuracy: 0.9473684210526315
Random Forest Confusion Matrix:
[[38  5]
 [ 1 70]]
Random Forest Precision: 0.9333333333333333
Random Forest Recall: 0.9859154929577465
Random Forest F1 Score: 0.958904109589041
```

#5

```
RandomForestClassifier(max_depth=1000, min_samples_leaf=100,
                       min_samples_split=100, n_estimators=1000,
                       random_state=42)
Random Forest Accuracy: 0.9473684210526315
Random Forest Confusion Matrix:
[[38  5]
 [ 1 70]]
Random Forest Precision: 0.9333333333333333
Random Forest Recall: 0.9859154929577465
Random Forest F1 Score: 0.958904109589041
```

#6

```
RandomForestClassifier() #default values
Random Forest Accuracy: 0.9649122807017544
Random Forest Confusion Matrix:
[[40  3]
 [ 1 70]]
Random Forest Precision: 0.958904109589041
Random Forest Recall: 0.9859154929577465
Random Forest F1 Score: 0.9722222222222222
```

AdaBoost

#1

```
AdaBoostClassifier() #default values

AdaBoost Accuracy: 0.9736842105263158
AdaBoost Confusion Matrix:
[[41  2]
 [ 1 70]]
AdaBoost Precision: 0.9722222222222222
AdaBoost Recall: 0.9859154929577465
AdaBoost F1 Score: 0.9790209790209791
```

#2

```
AdaBoostClassifier(n_estimators=1000)

AdaBoost Accuracy: 0.9736842105263158
AdaBoost Confusion Matrix:
[[41  2]
 [ 1 70]]
AdaBoost Precision: 0.9722222222222222
AdaBoost Recall: 0.9859154929577465
AdaBoost F1 Score: 0.9790209790209791
```

#3

```
AdaBoostClassifier(learning_rate=10, n_estimators=1000)

AdaBoost Accuracy: 0.868421052631579
AdaBoost Confusion Matrix:
[[30 13]
 [ 2 69]]
AdaBoost Precision: 0.8414634146341463
AdaBoost Recall: 0.971830985915493
AdaBoost F1 Score: 0.9019607843137255
```

#4

```
AdaBoostClassifier(learning_rate=10)

AdaBoost Accuracy: 0.8771929824561403
AdaBoost Confusion Matrix:
[[31 12]
 [ 2 69]]
AdaBoost Precision: 0.8518518518518519
AdaBoost Recall: 0.971830985915493
AdaBoost F1 Score: 0.9078947368421053
```

#5

```
AdaBoostClassifier(learning_rate=10, random_state=42)
AdaBoost Accuracy: 0.8771929824561403
AdaBoost Confusion Matrix:
[[31 12]
 [ 2 69]]
AdaBoost Precision: 0.8518518518518519
AdaBoost Recall: 0.971830985915493
AdaBoost F1 Score: 0.9078947368421053
```

#6

```
AdaBoostClassifier(algorithm='SAMME',learning_rate=10,
random_state=42)
AdaBoost Accuracy: 0.8947368421052632
AdaBoost Confusion Matrix:
[[39  4]
 [ 8 63]]
AdaBoost Precision: 0.9402985074626866
AdaBoost Recall: 0.8873239436619719
AdaBoost F1 Score: 0.9130434782608695
```

```
from sklearn.manifold import TSNE
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

X = my_data.data
y = my_data.target
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
tsne = TSNE(n_components=2, random_state=42)
X_embedded = tsne.fit_transform(X_scaled)
plt.figure(figsize=(8, 6))
for label, color, name in zip([0, 1], ['red', 'green'],
my_data.target_names):
    plt.scatter(
```

```
        X_embedded[y == label, 0], X_embedded[y == label, 1],
        c=color, label=name, s=10)
plt.legend(title="Cancer Type")
plt.title("TSNE Visualization of Breast Cancer Dataset")
plt.xlabel("TSNE Component 1")
plt.ylabel("TSNE Component 2")
plt.show()
```