

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.decomposition import PCA
from sklearn.metrics import accuracy_score

# Loading the dataset
data = load_iris()
X, y = data.data, data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# WITHOUT PCA
model_no_pca = RandomForestClassifier(random_state=42)
model_no_pca.fit(X_train, y_train)
y_pred_no_pca = model_no_pca.predict(X_test)
accuracy_no_pca = accuracy_score(y_test, y_pred_no_pca)

#APPLYING PCA
pca = PCA(n_components=2) # Example: Reducing to 2 components for visualization
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

#TRAINING WITH PCA
model_with_pca = RandomForestClassifier(random_state=42)
model_with_pca.fit(X_train_pca, y_train)
y_pred_with_pca = model_with_pca.predict(X_test_pca)
accuracy_with_pca = accuracy_score(y_test, y_pred_with_pca)

print(f'Accuracy without PCA: {accuracy_no_pca:.2f}')
print(f'Accuracy with PCA: {accuracy_with_pca:.2f}')

    Accuracy without PCA: 1.00
    Accuracy with PCA: 0.97
```

Using pca we took best 2 cols and we able to represent the data with an accuracy of 97%

