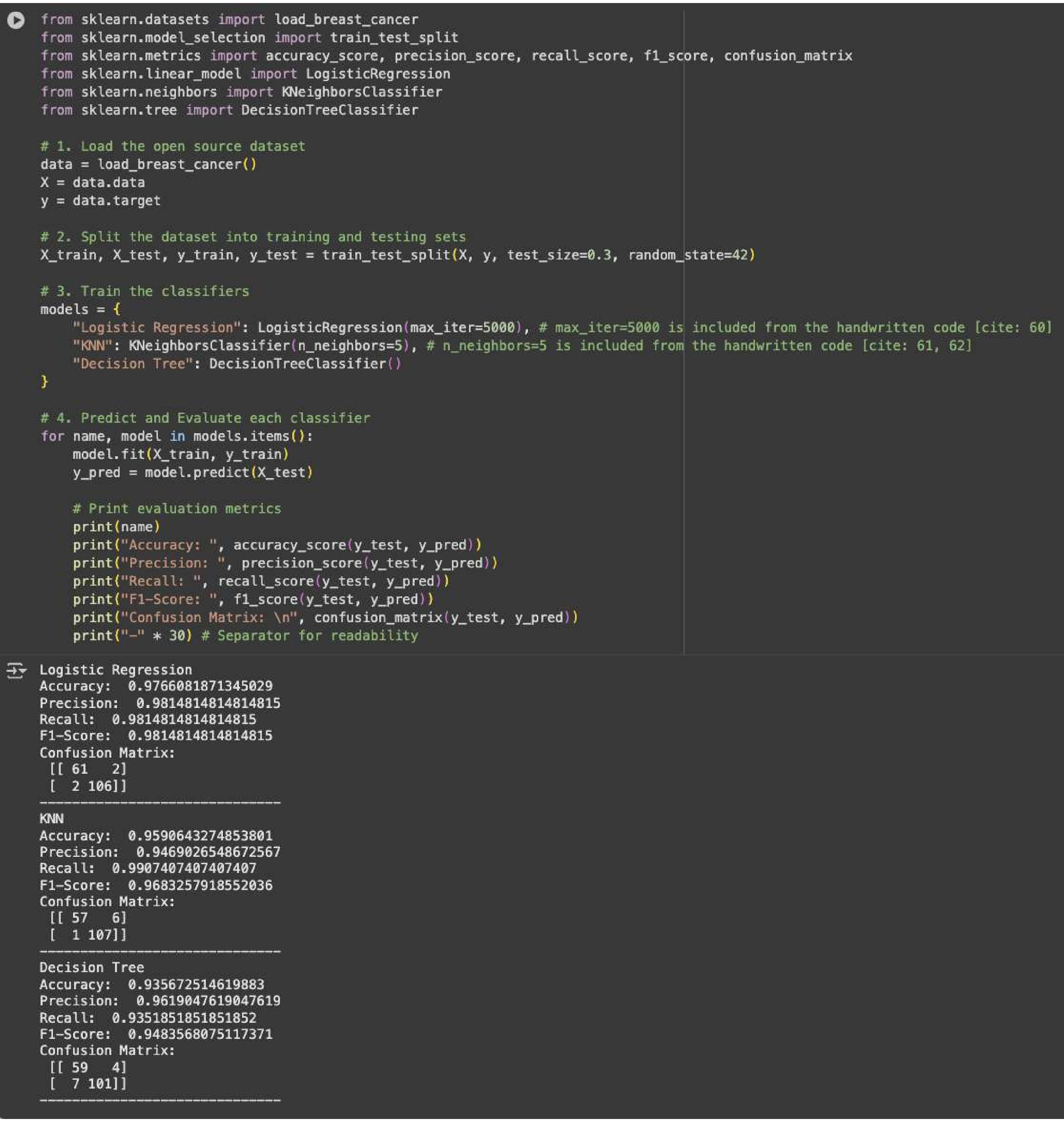

```python
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier

# 1. Load the open source dataset
data = load_breast_cancer()
X = data.data
y = data.target

# 2. Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 3. Train the classifiers
models = {
    "Logistic Regression": LogisticRegression(max_iter=5000), # max_iter=5000 is included from the handwritten code [cite: 60]
    "KNN": KNeighborsClassifier(n_neighbors=5), # n_neighbors=5 is included from the handwritten code [cite: 61, 62]
    "Decision Tree": DecisionTreeClassifier()
}

# 4. Predict and Evaluate each classifier
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Print evaluation metrics
    print(name)
    print("Accuracy: ", accuracy_score(y_test, y_pred))
    print("Precision: ", precision_score(y_test, y_pred))
    print("Recall: ", recall_score(y_test, y_pred))
    print("F1-Score: ", f1_score(y_test, y_pred))
    print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
    print("-" * 30) # Separator for readability
```

```
Logistic Regression
Accuracy:  0.9766081871345029
Precision:  0.9814814814814815
Recall:  0.9814814814814815
F1-Score:  0.9814814814814815
Confusion Matrix: 
 [[ 61   2]
 [  2 106]]
------------------------------
KNN
Accuracy:  0.9590643274853801
Precision:  0.9469026548672567
Recall:  0.9907407407407407
F1-Score:  0.9683257918552036
Confusion Matrix: 
 [[ 57   6]
 [  1 107]]
------------------------------
Decision Tree
Accuracy:  0.935672514619883
Precision:  0.9619047619047619
Recall:  0.9351851851851852
F1-Score:  0.9483568075117371
Confusion Matrix: 
 [[ 59   4]
 [  7 101]]
------------------------------
```

1-8-25

Exp-3 Study of the classifieas with respect to statistical parameters.

Aim :-

To study and evaluate the performance of three different classifier using statistical matrics such as accuracy, precision, etc on an open source dataset.

Description :-

Accuracy :-

It measures the proportion of correctly predicted instances among all predictions

formula ⇒ $$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Precision :-

It measures the correctness among positive predictions.

$$\text{formula} = \frac{TP}{(TP + FP)}$$

Recall :-

Measures how well actual positive are identified formula $$\text{Recall} = \frac{TP}{(TP + FN)}$$

F Source :-

Harmonic mean of precision and recall.

formula ⇒ $$F_1 \text{ Score} = \frac{2 \times (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

Confusion matrix :

This is a table showing the correct and incorrect predictions across classes. It helps visualize model performance with True Positive (TP), False Positive (FP), False Negatives (FN), and True Negatives (TN).

Procedure :

1.) Load the open source dataset

2.) Split the dataset into training and testing sets.

3.) Train the classifier : Logistic Regression, KNN and Decision Tree.

4.) Predict labels on the test data.

5.) Evaluate each classifier using Accuracy, Precision, etc.

6.) Visualize the confusion matrix.

Program :

```
from sklearn.datasets import load- breast-cancer
from sklearn.model-selection import train-test-spli
from sklearn.metrics import acuracy-score, precision-score
       recall-score, f1-score, confusion-matrix
from sklear.linear-model import Logistic Regressio
from sklearn.neighbours import KNeighbours Classifie
from sklear.tree import DecisionTree Classifier
```

```
data= load-breast_datac'
 x = data.data
 y= data.target
 x_train , x-test, y-train, y-test = train_test-split
 (x, y, test-size =0.3, random-state =42)

models = {"logistic Regression": Logistic Regression
                    (max-Ptr =5000),
   "KNN": KNeighbours Classifier (n-neighbors =5),
   "Decision Tree" : Decision Tree classifier ()}

for name, model in models.items()
   model.fit(x-train, y-train)
    y-pred = model.predict (x-test)
   print(name)
  print("Accuracy: ", accuracy-score(y_test y-pred))
print("Precision ", precision-score (y-test, y-pred))
print ("Recall: ", recall-score(y-test, y-pred))
print ("f1score", f1score(y-test, y-pred))
print ("Confusion Matrix:\n", confusion_matrix
                               score (y-test,
                                  y-pred))
```

Result:

The classification of Logistic Regressior, KNN and Decision Tree were successfully evaluated using the Breast cancer dataset. All the models performed well, with Logistic Regression achieiing the highest overall perfomance across all metrics.

Classifier	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.98	0.98	0.98	0.98
KNN	0.96	0.95	0.99	0.97
Decision Tree	0.94	0.97	0.94	0.95

Logistic Regression:

61 (TP)	2 (FN)
2 (FP)	106 (TN)

where malignment (cancerous)
↓
+ve

Benign (non-canc
↓
-ve

KNN

57 (TP)	6 (FN)
1 (FP)	107 (TN)

Decision Tree

60 (TP)	3 (FN)
7 (FP)	101 (TN)