```
#  Using the Optical Recognition of Handwritten Disease database, suppose the SVM
#  classifier gives better performance. Create a table with the following structure:
# | Nature  | Learning Rate | F1 Score |
# |---------|---------------|----------|
# | Linear  | 0.1           | -        |
# |         | 0.3           | -        |
# |         | 0.5           | -        |
# | Poly    | 0.1           | -        |
# |         | 0.3           | -        |
# |         | 0.5           | -        |
# | Radial  | 0.1           | -        |
# |         | 0.3           | -        |
# |         | 0.5           | -        |
# Justify the best nature of the SVM classifier for this database

# Import necessary libraries
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import f1_score
import pandas as pd

# Load the dataset
digits = datasets.load_digits()
X, y = digits.data, digits.target

# Split the data 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)

# Define kernel types and C (learning_rate) values
kernels = ['linear', 'poly', 'rbf']
```

```
C_values = [0.1, 0.3, 0.5]
results = []

# Train and evaluate the model for each kernel and C value
for kernel in kernels:
    for C in C_values:
        svm = SVC(kernel=kernel, C=C)
        svm.fit(X_train, y_train)

        # Predict and evaluate the model
        y_pred = svm.predict(X_test)
        f1 = f1_score(y_test, y_pred, average='weighted')

        # Append results
        results.append({'Nature': kernel, 'Learning_rate': C, 'F1_score': f1})

# Create a DataFrame to display results in tabular format
results_df = pd.DataFrame(results)
print(results_df)
```