# TASK 03: Implement SVM for Cats vs Dogs Classification

# SkillCraft Technology Internship - Machine Learning with Python

# Step 1: Import Libraries

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import make\_classification

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, confusion\_matrix

# Step 2: Generate Synthetic Data (Simulate cat/dog image features)

# 1000 samples, 100 features to mimic image data, 2 classes (cats & dogs)

X, y = make\_classification(

n\_samples=1000,

n\_features=100,

n\_informative=75,

n\_redundant=25,

n\_classes=2,

random\_state=42

)

# Step 3: Split Data into Train and Test

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.4, random\_state=42

)

# Step 4: Train and Evaluate Different SVM Kernels

kernels = ['linear', 'rbf', 'poly']

accuracies = {}

print("\nComparing different SVM kernels:\n-------------------------------")

for kernel in kernels:

model = SVC(kernel=kernel)

model.fit(X\_train, y\_train)

acc = model.score(X\_test, y\_test)

accuracies[kernel] = acc

print(f"{kernel.upper()} kernel accuracy: {acc:.4f} ({acc\*100:.2f}%)")

# Step 5: Pick Best Kernel (Highest Accuracy)

best\_kernel = max(accuracies, key=accuracies.get)

best\_acc = accuracies[best\_kernel]

print(f"\nBest performing kernel: {best\_kernel.upper()} ({best\_acc:.4f})")

# Step 6: Final Model using Best Kernel

final\_model = SVC(kernel=best\_kernel)

final\_model.fit(X\_train, y\_train)

y\_pred = final\_model.predict(X\_test)

# Step 7: Classification Report

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred, target\_names=["Cat", "Dog"]))

# Step 8: Confusion Matrix

cm = confusion\_matrix(y\_test, y\_pred)

print("\nConfusion Matrix:")

print(cm)

# Optional Visualization

plt.figure(figsize=(4,4))

plt.imshow(cm, cmap='Blues')

plt.title('Confusion Matrix')

plt.colorbar()

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

print("\nSVM classification complete!")

print("This program used synthetic data to simulate cat/dog image features.")

print("In real implementation, replace this with actual Kaggle image preprocessing.")