Laboratory Exercise: Support Vector Machine (SVM) Implementation and Kernel Performance Analysis

- Implement SVM classifiers with various kernel functions
- Visualize and interpret SVM decision boundaries
- Evaluate models using different metrics
- Tune hyperparameters (C, gamma, degree)
- Practice essential data preprocessing

Prerequisites

- Python programming (NumPy, Pandas, Matplotlib)
- Basic machine learning (classification, training/testing split)
- Familiarity with classification metrics

Required Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, accuracy_score, precision_recall_fscore_support
import warnings
warnings.filterwarnings('ignore')
```

Part 1: Data Preparation

1.1 Synthetic Datasets

Write functions to generate:

- **Linearly separable data** (make_classification)
- **Non-linear moon data** (make_moons)
- **Non-linear circle data** (make_circles)

1.2 Real Dataset

Write a function to load the **best dataset as per your choice** from sklearn but use only the *first two features* (and create a binary target: class 0 vs others).

Part 2: Preprocessing

- Always scale your features using StandardScaler (fit only on train, transform on both train and test).

Part 3: SVM Implementation

- Train SVM classifiers with *different kernels*: 'linear', 'poly', 'rbf', 'sigmoid'
    - For polynomial use degree=3.
- Write a function to return a dictionary of trained models per kernel.

- Use SVC from sklearn.

Part 4: Hyperparameter Tuning

- Use GridSearchCV to optimize at least two parameters for each kernel (e.g., C, gamma for RBF; C, degree for Poly).

Part 5: Evaluation & Visualization

- Evaluate each model on held-out test data (accuracy, precision, recall, F1). Use precision_recall_fscore_support for metrics.
- Plot confusion matrices (sns.heatmap).
- Plot decision boundaries using contour plots (for 2D data). Highlight support vectors.

Part 6: Experiments

- Run complete workflow on **Linear**, **Moons**, **Circles**, and **Wine** datasets.
- Create a results table comparing all models and kernels.

Part 7: Parameter Analysis

- Plot accuracy heatmaps for RBF kernel using various values of C and gamma.

Part 8: Lab Report

Analyze:

- Which kernel works best for which dataset and why?
- How do C and gamma affect decision boundaries and metrics?
- What happens with/without scaling?
- Any computational/training-time differences?

Submit:

- Code (Jupyter notebook)
- Plots and tables generated
- Written response (analysis, findings, recommendations)

Advanced

- Extend to multi-class (full wine dataset, compare one-vs-rest vs one-vs-one)
- Implement your own custom kernel function

Helpful Links

- [Scikit-learn SVM Documentation]
- [SVM Example - Decision Boundaries]