SVM and Kernel Methods

Assignment 1: Basic SVM Classification

Context

The One Hour AI Solution Platform wants to analyze its user session data to predict whether users will book an AI engineering consultation based on their interaction with the platform.

Dataset: platform_user_interactions.csv

Task

1. Create a Python script that:
   - Loads the dataset
   - Splits the data into training and testing sets (70% training, 30% testing)
   - Scales/normalizes the features (as emphasized in the reading material)
   - Implements a linear SVM classifier
   - Evaluates the model's performance using accuracy and precision
   - Identifies which user behaviors are most predictive of booking
2. Write a brief report (2-3 sentences) explaining:
   - How SVM's margin maximization helps make reliable predictions
   - Why feature scaling was necessary (as discussed in the reading material)

Deliverables

- Python script (.py file)
- Brief report (.txt file)

Assignment 2: Kernel Selection (Intermediate Level)

Context

The One Hour AI Solution Platform wants to identify patterns in engineer expertise to better match engineers with client problems. You'll use SVM with different kernels to classify engineers based on their skills and experience.

Dataset: engineer_skills.csv

Task

1. Create a Python script that:
    - Loads and prepares the engineer dataset
    - Implements three different SVM kernels (linear, polynomial, RBF) as covered in the reading material
    - Compares the performance of each kernel
    - Creates a visualization showing decision boundaries for two features of your choice
    - Identifies which kernel performs best for this classification task
2. Write a brief explanation (3-4 sentences) about:
    - Why different kernels produce different results (referencing the "kernel trick" from the reading)
    - Which kernel is most appropriate for this data and why

Deliverables

- Python script (.py file)
- Brief explanation (.txt file)

Assignment 3: Support Vector Analysis (Advanced Level)

Context

The One Hour AI Solution Platform wants to optimize their service pricing based on session complexity and engineer expertise. You'll implement an SVM with parameter tuning to classify sessions into different pricing tiers.

Dataset: session_complexity.csv

Task

1. Create a Python script that:
    - Loads and prepares the session pricing dataset
    - Implements a multi-class SVM approach
    - Experiments with different C parameter values as described in the reading material
    - Identifies support vectors and explains their significance
    - Evaluates the model with appropriate metrics
    - Creates a visualization that shows support vectors and decision boundaries
2. Write a brief report explaining:
    - How the C parameter affects the margin width and classification errors (referencing the reading material)
    - Why support vectors are important in SVM classification
    - How the model's performance varies with different C values

Deliverables

- Python script (.py file)
- Brief report (.txt file)