SVM Implementation for Pulsar Classification

"This project focuses on identifying pulsars, which are special neutron stars that emit detectable radio signals on Earth. These celestial objects are valuable for studying space-time, interstellar matter, and various states of matter. The challenge lies in detecting genuine pulsar signals among numerous false positives caused by radio interference and noise.

The detection process works by:

1. Collecting potential pulsar signals ('candidates')
2. Averaging these signals over multiple pulsar rotations
3. Using machine learning to automatically identify real pulsar signals

The problem is structured as a binary classification task where:

- Positive class: Real pulsar signals (minority)
- Negative class: False signals/interference (majority)

The implementation requirements are:

1. **Data Split**:
    - 80% for training
    - 20% for testing
2. **Data Normalization**:
    - Calculate mean and variance using only training data
    - Normalize both training and test sets using these values
    - Goal: Zero mean and unit variance for each feature
3. **SVM Implementation**:
    - Use cvxopt.solvers for the dual optimization problem
    - Implement linear kernel
    - Test with different C values: [0.1, 1, 10, 100, 1000]
    - Measure classification accuracy for each C value

The hyperparameter C controls how strictly the model enforces classification boundaries, balancing between decision boundary smoothness and classification accuracy."

Would you like me to explain any specific aspect of this objective in more detail?

Instructions:

1. Do not import any more libraries or modify any functions given in the skeleton code.
2. Input for evaluating the test cases; do not change the hyperparameter C value.
3. The output will be in decimal points.
4. You must use random_state=42 during the train test split.

Dataset: The dataset contains samples of pulsar candidates collected during the High Time Resolution Universe Survey (South). It has around 17898 instances with 8 continuous attributes.

The target attribute is “Class” which can be legitimate (1) or spurious (0). Please note that the dataset may contain missing values. To handle these missing values, you should use appropriate techniques.

Data Filename: pulsar_star_dataset.csv

Dataset description: The first four attributes are simple statistics obtained from the integrated pulse profile (folded profile). This is an array of continuous variables that describe a longitude-resolved version of the signal that has been averaged in both time and frequency. The remaining four variables are similarly obtained from the DM-SNR curve. These are summarized below:

1. Mean of the integrated profile.
2. Standard deviation of the integrated profile.
3. Excess kurtosis of the integrated profile.
4. Skewness of the integrated profile.
5. Mean of the DM-SNR curve.
6. Standard deviation of the DM-SNR curve.
7. Excess kurtosis of the DM-SNR curve.
8. Skewness of the DM-SNR curve.
9. Class.

Here, DM-SNR stands for two things: Dispersion Measure (DM) and Signal-to-Noise Ratio (SNR). DM, as the name suggests, measures the dispersion or spread of pulsar's signals during their journey from pulsar to earth. SNR, on the other hand, measures the strength of a pulsar's signal relative to background noise. DM is calculated from the time delay of each signal when it arrives on earth, while SNR is calculated at the peak intensity of each signal.

Sample Test Cases:

"input": "0.9\n",

"output": "0.97\n"

"input": "9\n",

"output": "0.975\n"

"input": "90\n",

"output": "0.975\n"

"input": "900\n",

"output": "0.98\n"

"input": "9000\n",

"output": "0.98\n”