Started on	Thursday, 31 October 2024, 10:32 AM
State	Finished
Completed on	Friday, 1 November 2024, 11:33 PM
Time taken	1 day 13 hours
Grade	**1.0** out of 1.0 (**100**%)
Feedback	Well done!

Question 1

Complete

Mark 1.0 out of 1.0

Construct a Matlab/Python function for statistical classification using empirical probability density functions based on Gaussian kernel density estimates. The function call should be:

```
C = kdclassify(traindata, trainclass, data, h)
```

and the parameters and the output should be as follows:

- Matrix *traindata* contains training examples so that each column is a single example.
- Row vector *trainclass* contains the classes of the examples, so that element *i* of *trainclass* is the class of the example in column *i* of *traindata*.
- Matrix *data* contains samples to be classified, one in each column.
- *h* is the length parameter which determines the effective width of the Gaussian.
- Row vector *C* contains the classes and it should include one value for each column in data.

Verify experimentally that the implementation works using the provided data: CSV, MAT.

Hints: The algorithm works as follows:

1. For each sample to be classified, determine the probability of the sample to belong to each class as follows (the sum is over samples in a class): $\hat{p}(\mathbf{x}) = \frac{1}{h^l}\left(\frac{1}{N}\sum_{i=1}^{N}\phi\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)\right)$ where $\mathbf{x}$ is a sample to be classified, *h* is the length parameter, *l* is the dimensionality, *N* is the number of samples in the class-specific training set, ϕ is the indicator function and $\mathbf{x}_i$ is a training set sample from a specific class. For the indicator function ϕ, standardized normal density (zero mean, unit variance) should be used.
2. Choose the class based on the maximum probability.

Current word count: *0*

ArmanGolbidi.zip

Comment: