Answer each question in Matlab or Python

Python users: you may use numpy (import as np) and pandas (import as pd); if you use other modules, let me know what they are and how you import them.

1) Write a function `magnitude` that takes in a vector/list/numpy-array and outputs the magnitude of the vector.

```
function out = magnitude(vec)
  total=0;
  for i=1:length(vec)
    total = total+vec(i)^2;
  end
  out= total.^(0.5);
```

```
def magnitude (inVec):
  total=0
  for x in inVec:
    total += inVec**2
  return total**(0.5)
```

Assume we have a matrix/list/numpy-array called `Data` which contains 1000 data points, each with 6 features and 1 classifier label.
For matlab/numpy, `Data` has 1000 rows and 7 columns (last column is label). For python list, `Data` is a 1000 element list, with each element being a 7-element list.

2) Write code to separate `Data` into two variable `DataFeats` and `DataLabels`, a matrix of 1000 x 6 features and a vector of 1000 labels respectively.

```
DataFeats=Data(:,1:6);
DataLabels=Data(:,7);
```

Python numpy

```
DataFeats=Data[:,0:5]
DataLabels=Data[:,6];
```

3) Presume there is a classifier function called `classify` that takes in the features for a single data point and outputs the corresponding class.
Write code to compute the accuracy of `classify` function's output across the 1000 data points.
(Classify syntax: `classify(featureVec)`).

```
correct=0;
for i=1:1000
  if classify(Data(i,1:6)) == Data(i,7)
    correct = correct+1;
  end;
end;
acc=correct/1000
```

Python

```
correct=0
for dataPt in Data:
  if classify(dataPt[0:5]) == dataPt[6]:
    correct += 1
acc=correct/1000.0
```