Algorithm 1: ksample

Input: inputs - list of ndarray, each input has shape (n,p) where n is samples and p is features, p must be the same for each input

Output: 1) u - ndarray of concatenated inputs of shape (N,p) where N is all samples in all inputs concatenated and p is features 2) v - ndarray of concatenated inputs of shape (N,k) where N is all samples in all inputs concatenated and k is the length(inputs)

function `ksampletransform(`$inputs$`)`**:**

inputsLength$\leftarrow$ $length(inputs)$;
u$\leftarrow$ $stack(inputs)$;
if *inputsLength* $==$ *2* **then**
n1$\leftarrow$ $shape(samples(inputs[0]))$;
n2$\leftarrow$ $shape(samples(inputs[1]))$;
v$\leftarrow$ $stack([zeros((n1, 1), ones(n2, 1)])$;
end
else
for $iinrange(inputsLength$ **do**
n$\leftarrow$ $shape(samples(inputs[i]))$;
encode$\leftarrow$ $zeros(shape(n, inputsLength))$;
encode[:, i] $\leftarrow$ $ones(shape(n))$;
$vs.append(encode)$;
v $\leftarrow$ $concatenate(vs)$;
end
end
return u,v;