SVM 1vsAll algorithm

Input:
- 3/2 training samples (**Tx**) with labels (**Ty**).
- 1/3 training samples (**Vx**) with labels (**Vy**). This will be used to validate the SVM.
- Parameters: ***g*** (Gaussian kernel) and ***C***.

Output:
- svm_out = predicted labels (Py) of Vx.

Compare **Py** to **Vy** to compute the rate of SVM.
%
For each class assign a label **1** and a label -**1** for all other classes.
|
| * Call the function **Classifiy_svm**.
| - The input parameters are: the **Tx**, labels (**1** and **-1**), **Vx**, **Vy**, **C**, **g**
| - **Classify_svm** will return a real value R -positive or negative- for each element in Vx.
|------ end For

The output is a matrix **M**

	0	1	2	.....	9
Vx(1)	Value ***R*** of Vx(1) when class 0 is versus All	Value ***R*** of Vx(1) when class 1 is versus All	.....	.....	Value ***R*** of Vx(1) when class 9 is versus All
Vx(2)	Value ***R*** of Vx(2) when class 0 is versus All	.....	...	.....	
. . . .	.....	.....		.....	
Vx(1000)					

Now to find the predicted label of the ith element in **Vx** (i=[1..1000]), we look where is the maximal ***R*** in the ith row of the matrix **M**.

For example, if the first row of **M** is (predicted labels of the first element) :

Vx	0	1	2	3	4	5	6	7	8	9
Vx(1)	-2.0	3.53	6.2	-8.9	0	1	**10**	3	5	-20

The maximal value is **10** so the predicted label to be assigned to the element **1** in **Vx** is 6.

Using **M,** we can predict the labels of the elements of **Vx**.

Now, to compute the error rate, we compare the predicted labels to the real labels (**Vy**) .

This should be repeated for each value of ***g*** in [0.1 0.01 0.001 0.0001] and ***C*** in [1 10 100 1000 10000].

The optimal ***g (i.e go)*** and ***C (i.e. Co)*** are those corresponding to the best error rate.

Finally, use ***go*** and ***Co*** to predict the labels of **Test** dataset.