CS449/549 Computational Learning Practice Problems 2 (Fall 2023)

True/False Circle one. Provide short explanations to your answers.

1. (T/F) The dual Perceptron algorithm makes the same number of mistakes as the Perceptron algorithm.
   True: it is the same algorithm expressed differently (primal versus dual form).

2. (T/F) For Weighted Majority, the weight update can be revised to $w_i \leftarrow \beta w_i$ for any $\beta \in (0, 1)$.
   True: but the mistake bound will change accordingly.

3. (T/F) The Drifting Weighted Majority algorithm knows when the best expert has changed.
   False: it is oblivious to the changes in the best expert sequence (this is a feature, not a bug).

4. (T/F) The AdaBoost algorithm must decide on the number of boosting rounds ahead of time.
   False: it can compute an upper bound the training error as it goes (it need not know the advantage of each weak hypothesis over random guessing).

5. (T/F) The quadratic kernel $k(\vec{x}, \vec{z}) = (\langle \vec{x}, \vec{z} \rangle + c)^2$ over $\vec{x}, \vec{z} \in \mathbb{R}^2$ allows a dual Perceptron looking for a separating hyperplane in $\mathbb{R}^6$.
   True: the expanded vector contains $(x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2c}x_1, \sqrt{2c}x_2, c)$.

Perceptron Consider the dual Perceptron algorithm.

```
/* x[i] = i-th feature vector, y[i] = (-1/+1) label of x[i] */
m = zero_vector(len(x[0]))
Oops = True /* mistake alert */
while Oops:
  Oops = False
  for i = 1 to num_points:
    if y[i]*sum(m[j]*y[j]*dot_product(x[i],x[j]) for j in range(num_points)) < 0:
      m[i] = m[i] + 1
      Oops = True
return (m,x,y)
```

(T/F) The above is a correct implementation of the dual Perceptron algorithm.
False: the dimension of m *is incorrect.*