Introduction to Machine Learning
Practice Questions

Terminology. For each term on the left, write the letter that best matches the description from the right

☐ Learning

A. Frequency at which a machine learning model makes mistakes (with respect to some gold standard).

☐ Training

B. Using a machine learning model to obtain predictions for a set of features.

☐ Testing

C. Machine learning model that separates elements in a dataset into two groups.

☐ Evaluation

D. Using previous experience to improve a performance metric on a given task.

☐ Error rate

E. Analyzing the strengths and weaknesses of a machine learning model based on a set of metrics, e.g., the error rate of the model.

☐ Binary classifier

F. Learning algorithm that takes the dot product of the features and a vector of weights. This result is then used to make predictions.

☐ Perceptron algorithm

G. Process of using a learning algorithm and past experience to construct a machine learning model such as a classifier.

The perceptron algorithm. Consider the following API for a Vector data type.

```java
public class Vector {
    // Creates a Vector object from the given array.
    public Vector(double[] values)
    // Returns the sum of this vector and other.
    public Vector plus(Vector other)
    // Subtracts other from this and returns the difference.
    public Vector subtract(Vector other)
    // Returns the inner product of this Vector and other
    public double dot(Vector other)
    // Returns a String representation of the vector.
    // Example: (2.0, 1.0, -1.0)
    public String toString()
}
```

The following client simulates one weight update of the perceptron algorithm. Please complete the client with the most appropriate code snippet from the options presented. Each option can be used once, several times, or not at all.

Code snippets

A. `weights.plus(point)`	H. `2.0`
B. `weights.subtract(point)`	I. `4.0`
C. `point.subtract(weights)`	J. `-4.0`
D. `weights.dot(point)`	K. `-6.0`
E. `weights`	L. `(-2.0, 0.0, 1.0)`
F. `1`	M. `(-2.0, 2.0, -1.0)`
G. `-1`	N. `(-2.0, -2.0, 0.0)`

Client

```
double[] w = { -2.0, 0.0, 1.0 };
Vector weights = new Vector(w);
double[] x = { 0.0, -2.0, 2.0 };
Vector point = new Vector(x);
int binaryLabel = -1;

// compute dot product between the weights and the data point
double dotProduct = [    ];

StdOut.println("The dot product should be [    ]");;
StdOut.println("dot product = " + dotProduct);

StdOut.println("The perceptron would predict a [    ]");;

// update the weights as the perceptron would
// knowing we have a negative binaryLabel

weights = [    ];

StdOut.println("The new weights should be [    ]");;
StdOut.println("updated weights = " + weights);

// the new weights should produce a smaller dot product
// with the data point

double newProduct = [    ];

StdOut.println("The new dot product should be [    ]");;
StdOut.println("new dot product = " + newProduct);

StdOut.println("The perceptron would now predict a [    ]");;
```

Model evaluation. We have trained a binary classifier to predict whether a new comment in an online forum contains hate speech. For our evaluation, we constructed a confusion matrix using a test dataset of 1000 comments:
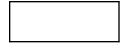
	prediction	
binary label	hate speech	no hate speech
hate speech	45	5
no hate speech	55	895

Answer the following questions based on the information provided in this matrix.

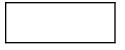A. What is the test error rate for this binary classifier?

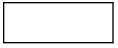B. How many comments were predicted to contain hate speech?

C. How many comments had a binary label corresponding to hate speech?

D. Out of the comments that were predicted to contain hate speech, what percentage had a binary label corresponding to hate speech? This quantity is known as the **precision** of the classifier, and refers to how accurate our predictions are.

E. Out of the comments with a binary label corresponding to hate speech, what percentage was predicted to contain hate speech? This quantity is known as the **recall** of the classifier, and refers to how well the model can find the class of interest, i.e., comments with actual hate speech.

F. We wish to moderate our online forum with the help of this binary classifier. However, we have determined that we will only use it if the precision and recall values we found during our evaluation are greater than 80%. Select the option, or options, that best fit our situation.

- ◯ Our recall is lower than desired. Thus, we may fail to identify a large number of comments with hate speech, degrading the quality of the online discourse in our forum.

- ◯ Our precision is lower than desired. Thus, we may over-moderate the forum, flagging legitimate comments that are not hate speech.

- ◯ We are satisfied with our evaluation and ready to use our binary classifier for moderation.

Note: We are using 80% for pedagogical purposes 🙂