

FINAL: 90 Minutes

Last Name: _____

First Name: _____

RIN: _____

Section: 4100 / 6100 (circle one)

Answer **ALL** questions.

NO COLLABORATION or electronic devices. Any violations result in an **F**.

NO questions allowed during the test. Interpret and do the best you can.

ALWAYS show your work and justify each answer.

GOOD LUCK!

1	2	3	4	5	6	7	8	9	10	11	TOTAL
/1	/1	/1	/1	/1	/1	/1	/1	/2	/4	/1	/15

- You do not have time to waffle.
- Keep your answers precise and concise.
- Questions (or parts) are graded 0%, 50%, 100%.

1. What is the definition of *learning*? What is the 2-step approach to learning and why do it that way?

2. Every card has a letter on one side and a number on the other.
Hypothesis: If a card has a P on it, then the other side is a 5.



Above are some cards which you may turn (data). Can the hypothesis be falsified by the data. If not, why? If so, which cards are the fewest you need to turn over see if the hypothesis is false?

3. The professor of a class released the previous final just one day before the final (hence only giving students a limited time to study the previous final). Why? [*Hints: The professor wanted to prevent the students from doing what? The professor wanted the students to use the previous final as a what?*]

10. Give formulas for the form of the final *classification* hypothesis in each case. Specify which parameters (and their dimension) are to be learned from data. The data are (\mathbf{x}_n, y_n) , where $\mathbf{x}_n \in \mathbb{R}^d$ and $y_n = \pm 1$.

(a) Linear model trained with the pocket algorithm.

(b) Linear model with feature transform $\mathbf{z} = \mathbf{x}^T \mathbf{x}$, trained with the pocket algorithm.

(c) Nearest neighbor.

(d) Nonparametric RBF.

(e) k -RBF network.

(f) Two hidden-layer neural network with function $\theta(\cdot)$ in each hidden unit.

(g) SVM with feature transform $\mathbf{z} = \phi(\mathbf{x})$.

(h) SVM with kernel $K(\mathbf{x}, \mathbf{y})$.

11. How do your formulas above change if you are doing logistic regression instead of classification?

SCRATCH