Feedback — XI. Machine Learning System Design

You submitted this quiz on **Tue 7 Apr 2015 10:16 AM CEST**. You got a score of **5.00** out of **5.00**.

Question 1

You are working on a spam classification system using regularized logistic regression. "Spam" is the positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier, and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is:

		Actual Class	
		1	**0**
Predicted Class	**1**	85	890
	0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's F_1 score (as a value from 0 to 1)? Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

You entered:

0.15814

Your Answer		Score	Explanation
0.15814	✔	1.00	Precision is 0.087 and recall is 0.85, so F_1 score is (2 * precision * recall) / (precision + recall) = 0.158.
Total		1.00 / 1.00	

Question 2

Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. Which are the two?

Your Answer		Score	Explanation
☐ When we are willing to include high order polynomial features of x (such as $x_1^2, x_2^2, x_1x_2,$ etc.).	✔	0.25	As we saw with neural networks, polynomial features can still be insufficient to capture the complexity of the data, especially if the features are very high-dimensional. Instead, you should use a complex model with many parameters to fit to the large training set.
☑ A human expert on the application domain can confidently predict y when given only the features x (or more generally, if we have some way to be confident that x contains sufficient information to predict y accurately).	✔	0.25	It is important that the features contain sufficient information, as otherwise no amount of data can solve a learning problem in which the features do not contain enough information to make an accurate prediction.
☑ Our learning algorithm is able to represent fairly complex functions (for example, if we train a neural network or other model with a large number of parameters).	✔	0.25	You should use a complex, "low bias" algorithm, as it will be able to make use of the large dataset provided. If the model is too simple, it will underfit the large training set.
☐ We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).	✔	0.25	If the model has a small number of parameters, then it will underfit the large training set and not make good use of all the data.
Total		1.00 / 1.00	

Question 3

Suppose you have trained a logistic regression classifier which is outputing $h_\theta(x)$. Currently, you predict 1 if $h_\theta(x) \geq \text{threshold}$, and predict 0 if $h_\theta(x) < \text{threshold}$, where currently the threshold is set to 0.5. Suppose you **decrease** the threshold to 0.1. Which of the following are true? Check all that apply.

Your Answer		Score	Explanation
☐ The classifier is likely to have unchanged precision and recall, and thus the same F_1 score.	✔	0.25	By making more y = 1 predictions, we increase true and false positives and decrease true and false negatives. Thus, precision and recall will certainly change.
☐ The classifier is likely to now have higher precision.	✔	0.25	Lowering the threshold means more y = 1 predictions. This will increase both true and false positives, so precision will decrease, not increase.
☑ The classifier is likely to now have lower precision.	✔	0.25	Lowering the threshold means more y = 1 predictions. This will increase both true and false positives, so precision will decrease.
☐ The classifier is likely to have unchanged precision and recall, but higher accuracy.	✔	0.25	By making more y = 1 predictions, we increase true and false positives and decrease true and false negatives. Thus, precision and recall will certainly change. We cannot say whether accuracy will increase or decrease.
Total		1.00 / 1.00	

Question 4

Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

Your Answer	Score	Explanation

☑ If you always predict non-spam (output $y = 0$), your classifier will have a recall of 0%.	✔	0.25	Since every prediction is y = 0, there will be no true positives, so recall is 0%.
☐ If you always predict spam (output $y = 1$), your classifier will have a recall of 0% and precision of 99%.	✔	0.25	Every prediction is y = 1, so recall is 100% and precision is only 1%.
☑ If you always predict spam (output $y = 1$), your classifier will have a recall of 100% and precision of 1%.	✔	0.25	Since every prediction is y = 1, there are no false negatives, so recall is 100%. Furthermore, the precision will be the fraction of examples with are positive, which is 1%.
☑ If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, and it will likely perform similarly on the cross validation set.	✔	0.25	The classifier achieves 99% accuracy on the training set because of how skewed the classes are. We can expect that the cross-validation set will be skewed in the same fashion, so the classifier will have approximately the same accuracy.
Total		1.00 / 1.00	

Question 5

Which of the following statements are true? Check all that apply.

Your Answer		Score	Explanation
☐ It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.	✔	0.20	You cannot know whether a huge dataset will be important until you have built a first version and find that the algorithm has high variance.

☐ After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.	✔	0.20	You can and should adjust the threshold in logistic regression using cross validation data.
☑ The "error analysis" process of manually examining the examples which your algorithm got wrong can help suggest what are good steps to take (e.g., developing new features) to improve your algorithm's performance.	✔	0.20	This process of error analysis is crucial in developing high performance learning systems, as the space of possible improvements to your system is very large, and it gives you direction about what to work on next.
☐ If your model is underfitting the training set, then obtaining more data is likely to help.	✔	0.20	If the model is underfitting the training data, it has not captured the information in the examples you already have. Adding further examples will not help any more.
☑ On skewed datasets (e.g., when there are more positive examples than negative examples), accuracy is not a good measure of performance and you should instead use F_1 score based on the precision and recall.	✔	0.20	You can always achieve high accuracy on skewed datasets by predicting the most the same output (the most common one) for every input. Thus the F_1 score is a better way to measure performance.
Total		1.00 / 1.00	