1. Problem Statement — 1 point

This example is adapted from a real production application, but with details disguised to protect confidentiality.

You are a famous researcher in the City of Peacetopia. The people of Peacetopia have a common characteristic: they are afraid of birds. To save them, you have **to build an algorithm that will detect any bird flying over Peacetopia** and alert the population.

The City Council gives you a dataset of 10,000,000 images of the sky above Peacetopia, taken from the city's security cameras. They are labeled:

- $y = 0$: There is no bird on the image
- $y = 1$: There is a bird on the image

Your goal is to build an algorithm able to classify new images taken by security cameras from Peacetopia.

There are a lot of decisions to make:

- What is the evaluation metric?
- How do you structure your data into train/dev/test sets?

Metric of success

The City Council tells you that they want an algorithm that

1. Has high accuracy.
2. Runs quickly and takes only a short time to classify a new image.
3. Can fit in a small amount of memory, so that it can run in a small processor that the city will attach to many different security cameras.

Note: Having three evaluation metrics makes it harder for you to quickly choose between two different algorithms, and will slow down the speed with which your team can iterate. True/False?

- True
- False

74:57

Expand

2. After further discussions, the city narrows down its criteria to: — 1 point

- "We **need** an algorithm that can let us know a bird is flying over Peacetopia as accurately as possible."
- "We *want* the trained model to take no more than 10 sec to classify a new image."
- "We *want* the model to fit in 10MB of memory."

If you had the three following models, which one would you choose?

74:56

Test Accuracy	Runtime	Memory size
97%	1 sec	3MB

Test Accuracy	Runtime	Memory size
97%	3 sec	2MB

Test Accuracy	Runtime	Memory size
98%	9 sec	9MB

Test Accuracy	Runtime	Memory size
99%	13 sec	9MB

Expand

3. Which of the following best answers why it is important to identify optimizing and satisficing metrics? — 1 point

- Identifying the metric types sets thresholds for satisficing metrics. This provides explicit evaluation criteria.
- Identifying the optimizing metric informs the team which models they should try first.
- Knowing the metrics provides input for efficient project planning.
- It isn't. All metrics must be met for the model to be acceptable.

74:56

Expand

4. Structuring your data — 1 point

Before implementing your algorithm, you need to split your data into train/dev/test sets. Which of these do you think is the best choice?

74:55

Train	Dev	Test
6,000,000	3,000,000	1,000,000

Train	Dev	Test
6,000,000	1,000,000	3,000,000

Train	Dev	Test
9,500,000	250,000	250,000

Train	Dev	Test
3,333,334	3,333,334	3,333,334

Expand

5. After setting up your train/dev/test sets, the City Council comes across another 1,000,000 images, called the "citizens' data". Apparently the citizens of Peacetopia are so scared of birds that they volunteered to take pictures of the sky and label them, thus contributing these additional 1,000,000 images. These images are different from the distribution of images the City Council had originally given you, but you think it could help your algorithm. — 1 point

Notice that adding this additional data to the training set will make the distribution of the training set different from the distributions of the dev and test sets.

Is the following statement true or false?

"You should not add the citizens' data to the training set, because if the training distribution is different from the dev and test sets, then this will not allow the model to perform well on the test set."

- True
- False

74:55

Expand

6. One member of the City Council knows a little about machine learning and thinks you should add the 1,000,000 citizens' data images proportionately to the train/dev/test sets. You object because: — 1 point

- If we add the images to the test set then it won't reflect the distribution of data expected in production.
- The training set will not be as accurate because of the different distributions.
- The additional data would significantly slow down training time.
- The 1,000,000 citizens' data images do not have a consistent x-->y mapping as the rest of the data.

74:55

Expand

7. You train a system, and the train/dev set errors are 3.5% and 4.0% respectively. You decide to try regularization to close the train/dev accuracy gap. Do you agree? — 1 point

- No, because this shows your variance is higher than your bias.
- No, because you do not know what the human performance level is.
- Yes, because having a 4.0% training error shows you have a high bias.
- Yes, because this shows your bias is higher than your variance.

74:54

Expand

8. You ask a few people to label the dataset so as to find out what is human-level performance. You find the following levels of accuracy: — 1 point

Bird watching expert #1	0.3% error
Bird watching expert #2	0.5% error
Normal person #1 (not a bird watching expert)	1.0% error
Normal person #2 (not a bird watching expert)	1.2% error

If your goal is to have "human-level performance" be a proxy (or estimate) for Bayes error, how would you define "human-level performance"?

- 0.0% (because it is impossible to do better than this)
- 0.4% (average of 0.3 and 0.5)
- 0.75% (average of all four numbers above)
- 0.3% (accuracy of expert #1)

74:54

Expand

9. A learning algorithm's performance can be better than human-level performance but it can never be better than Bayes error. True/False? — 1 point

- True.
- False.

74:54

Expand

10. You find that a team of ornithologists debating and discussing an image gets an even better 0.1% performance, so you define that as "human-level performance." After working further on your algorithm, you end up with the following: — 1 point

Human-level performance	0.1%
Training set error	2.0%
Dev set error	2.1%

Based on the evidence you have, which two of the following four options seem the most promising to try? (Check two options.)

- [ ] Try increasing regularization.
- [ ] Try decreasing regularization.
- [ ] Get a bigger training set to reduce variance.
- [ ] Train a bigger model to try to do better on the training set.

74:53

Expand

11. You also evaluate your model on the test set, and find the following: — 1 point

Human-level performance	0.1%
Training set error	2.0%
Dev set error	2.1%
Test set error	7.0%

What does this mean? (Check the two best options.)

- [ ] You have overfit to the dev set.
- [ ] You should get a bigger test set.
- [ ] You should try to get a bigger dev set.
- [ ] You have underfitted to the dev set.

74:53

Expand

12. After working on this project for a year, you finally achieve: Human-level performance, 0.10%, Training set error, 0.05%, Dev set error, 0.05%. Which of the following are likely? (Check all that apply.) — 1 point

- [ ] There is still avoidable bias.
- [ ] The model has recognized emergent features that humans cannot. (Chess and Go for example)
- [ ] Pushing to even higher accuracy will be slow because you will not be able to easily identify sources of bias.
- [ ] This result is not possible since it should not be possible to surpass human-level performance.

74:52

Expand

13. Your system is now very accurate but has a higher false negative rate than the City Council of Peacetopia would like. What is your best next step? — 1 point

- Reset your "target" (metric) for the team and tune to it.
- Expand your model size to account for more corner cases.
- Pick false negative rate as the new metric, and use this new metric to drive all further development.
- Look at all the models you've developed during the development process and find the one with the lowest false negative error rate.

74:52

Expand

14. You've handily beaten your competitor, and your system is now deployed in Peacetopia and is protecting the citizens from birds! But over the last few months, a new species of bird has been slowly migrating into the area, so the performance of your system slowly degrades because your model is being tested on a new type of data. There are only 1,000 images of the new species. The city expects a better system from you within the next 3 months. Which of these should you do first? — 1 point

- Add hidden layers to further refine feature development.
- Augment your data to increase the images of the new bird.
- Put them into the dev set to evaluate the bias and re-tune.
- Add the new images and split them among train/dev/test.

74:52

Expand

15. The City Council thinks that having more Cats in the city would help scare off birds. They are so happy with your work on the Bird detector that they also hire you to build a Cat detector. (Wow Cat detectors are just incredibly useful, aren't they?) Because of years of working on Cat detectors, you have such a huge dataset of 100,000,000 cat images that training on this data takes about two weeks. Which of the statements do you agree with? (Check all that agree.) — 1 point

- [ ] Having built a good Bird detector, you should be able to take the same model and hyperparameters and just apply it to the Cat dataset, so there is no need to iterate.
- [ ] Buying faster computers could speed up your teams' iteration speed and thus your team's productivity.
- [ ] If 100,000,000 examples is enough to build a good enough Cat detector, you might be better off training with just 10,000,000 examples to gain a $\approx$10x improvement in how quickly you can run experiments, even if each model performs a bit worse because it's trained on less data.
- [ ] Needing two weeks to train will limit the speed at which you can iterate.

74:51