WELCOME TO HOGWARTS

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

You are a famous wizard in Hogwarts. The residents of Hogwarts have a common characteristic: they are afraid of crows because they believe that they are spies sent by Voldemort. To save them, you have **to build an algorithm that will detect any crow flying over Hogwarts** and alert the population.

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

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

Your goal is to build an algorithm able to classify new images taken by security cameras from Hogwarts. There are a lot of decisions to make:

- What is the evaluation metric (i.e. how do you decide your model is good)?
- How do you structure your data into train/dev/test sets?

Problem 1: Metric of success

The City Council tells you that they want an algorithm that

- Has high accuracy.
- Runs quickly and takes only a short time to classify a new image.
- 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

Problem 2

After further discussions, the city narrows down its criteria to:

- "We need an algorithm that can let us know a crow is flying over Hogwarts 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?

A)

Test Accuracy	Runtime	Memory size
97%	1 sec	3MB

B)

Test Accuracy	Runtime	Memory size
99%	13 sec	9MB

C)

Test Accuracy	Runtime	Memory size
97%	3 sec	2MB

D)

Test Accuracy	Runtime	Memory size
98%	9 sec	9MB

Problem 3: Structuring your data

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?

A)

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

B)

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

C)

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

D)

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

Problem 4

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 Hogwarts are so scared of crows 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.

You should not add the citizens’ data to the **training set**, because this will cause the training and dev/test set distributions to become different, thus hurting dev and test set performance. **T/F**

- True
- False

Problem 5

One member of the City Council knows a little about machine learning, and thinks you should add the 1,000,000 citizens' data images to the **test set**. You object because:

- This would cause the dev and test set distributions to become different. This is a bad idea because you're not aiming where you want to hit.
- The test set no longer reflects the distribution of data (security cameras) you most care about.
- A bigger test set will slow down the speed of iterating because of the computational expense of evaluating models on the test set.
- The 1,000,000 citizens' data images do not have a consistent x ⇒ y mapping as the rest of the data.

Problem 6

You train a system, and its errors are as follows (Error = 100% - Accuracy):

Training set error	4.0%
Dev set error	4.5%

This suggests that one good avenue for improving performance is to train a bigger network so as to drive down the 4.0% training error. Do you agree?

- Yes, because having 4.0% training error shows you have high bias.
- Yes, because this shows your bias is higher than your variance.
- No, because this shows your variance is higher than your bias.
- No, because there is not enough information to tell.

Problem 7

Note: **Bayes error** is the lowest possible error rate for any classifier of a random outcome (into, for example, one of two categories) and is analogous to the irreducible error (you can not reduce it anymore). **Human-level performance** is the best outcome for any human being.

You ask a few people to label the dataset so as to find out what is human-level performance (i.e. what accuracy would a human being get if they were to manually spot the crow). You find the following levels of accuracy:

Crow watching expert #1	0.3% error

Crow watching expert #2	0.5% error
Normal person #1 (not a crow watching expert)	1.0% error
Normal person #2 (not a crow 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)
- 0.3% (accuracy of expert #1)
- 0.4% (average of 0.3 and 0.5)
- 0.75% (average of all four numbers above)

Problem 8

Which of the following statements do you agree with?

- A learning algorithm’s performance can be better than human-level performance but it can never be better than Bayes error.
- A learning algorithm’s performance can never be better than human-level performance but it can be better than Bayes error.
- A learning algorithm’s performance can never be better than human-level performance nor better than Bayes error.
- A learning algorithm’s performance can be better than human-level performance and better than Bayes error.

Problem 9

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:

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.)

- Train a more complex model to try to do better on the training set.
- Get a bigger training set to reduce variance.

- Try increasing regularization.
- Try decreasing regularization.

Problem 10

You also evaluate your model on the test set, and find the following:

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 try to get a bigger dev set.
- You have underfit to the dev set.
- You should get a bigger test set.

Problem 11

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%

What can you conclude? (Check all that apply.)

- This is a statistical anomaly (or must be the result of statistical noise) since it should not be possible to surpass human-level performance.
- It is now harder to measure avoidable bias, thus progress will be slower going forward.
- With only 0.05% further progress to make, you should quickly be able to close the remaining gap to 0%
- If the test set is big enough for the 0.05% error estimate to be accurate, this implies Bayes error is ≤0.05

Problem 12

It turns out Hogwarts has hired one of your competitors to build a system as well. Your system and your competitor both deliver systems with about the same running time and memory size. However, your system has higher accuracy! However, when Hogwarts tries out your and your competitor's systems, they conclude they like your competitor's system better, because even though you have higher overall accuracy, you have more false negatives (failing to raise an alarm when a crow is in the air). What should you do?

- Look at all the models you've developed during the development process and find the one with the lowest false negative error rate.
- Ask your team to take into account both accuracy and false negative rate during development.
- Rethink the appropriate metric for this task, and ask your team to tune to the new metric.
- Pick false negative rate as the new metric, and use this new metric to drive all further development.

Problem 13

You've handily beaten your competitor, and your system is now deployed in Hogwarts and is protecting the citizens from the spies! But over the last few months, a new species of a crows has been slowly migrating into the area, so the performance of your system slowly degrades because your data is being tested on a new type of data.
[crow image]
You have only 1,000 images of the new species of crow. The city expects a better system from you within the next 3 months. Which of these should you do first?

- Use the data you have to define a new evaluation metric (using a new dev/test set) taking into account the new species, and use that to drive further progress for your team.
- Put the 1,000 images into the training set so as to try to do better on these crows.
- Try data augmentation/data synthesis to get more images of the new type of crow.
- Add the 1,000 images into your dataset and reshuffle into a new train/dev/test split.

Problem 14

The City Council thinks that having more Cats in the city would help scare off crows. They are so happy with your work on the crow detector that they also hire you to build a Cat detector. 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.)

- Needing two weeks to train will limit the speed at which you can iterate.
- Buying faster computers could speed up your teams' iteration speed and thus your team's productivity.
- Having built a good crow 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.
- If 100,000,000 examples is enough to build a good enough Cat detector, you might be better of training with just 10,000,000 examples to gain a ≈ 10x improvement in how quickly you can run experiments, even if each model performs a bit worse because it's trained on less data.