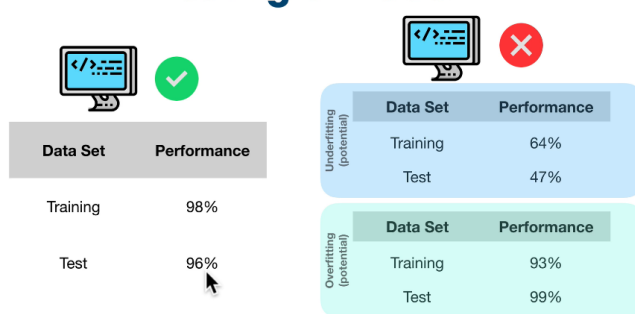


Modelling - Comparison

- How will our model perform in real world

▼ Testing a Model

- A good model will yield similar results on the training, validation and test sets, it's not uncommon to see a slight decline in performance from the model on the training and validation set to the test



▼ Overfitting and Underfitting

Overfitting and Underfitting Definitions

Before we get into the experimentation side of things, it's worth having a little reminder of overfitting and underfitting are.

All experiments should be conducted on different portions of your data.

- **Training data set** — Use this set for model training, 70–80% of your data is the standard.
- **Validation/development data set** — Use this set for model hyperparameter tuning and experimentation evaluation, 10–15% of your data is the standard.
- **Test data set** — Use this set for model testing and comparison, 10–15% of your data is the standard.

These amounts can fluctuate slightly, depending on your problem and the data you have.

Poor performance on training data means the model hasn't learned properly and is **underfitting**. Try a different model, improve the existing one through hyperparameter or collect more data.

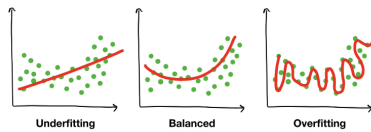
Great performance on the training data but poor performance on test data means your model doesn't generalize well. Your model may be **overfitting** the training data. Try using a simpler model or making sure your the test data is of the same style your model is training on.

Another form of **overfitting** can come in the form of better performance on test data than training data. This may mean your testing data is leaking into your training data (incorrect data splits) or you've spent too much time optimizing your model for the test set data. Ensure your training and test datasets are kept separate at all times and avoid optimizing a models performance on the test set (use the training and validation sets for model improvement).

Poor performance once deployed (in the real world) means there's a difference in what you trained and tested your model on and what is actually happening. Ensure the data you're using during experimentation matches up with the data you're using in production.

- ▼ Overfitting and underfitting of both examples of a model not being able to generalize well, which is what we don't want.

Overfitting and underfitting



- Now there are several reasons why underfitting and overfitting can happen, but the main ones are data leakage and data mismatch.
- Data leakage happens when some of your test data leaks into your training data, and this often results in overfitting or a model doing better on the test set than on the training data set.



- Data mismatch happens when the data you're testing on is different to the data you're training on, such as having different features in the training data to the test data.



Fixes for Overfitting and underfitting

▼ Underfitting

- Try a more advanced model
- Increase model hyper parameters
- Reduce amount of features
- train longer

▼ Overfitting

- Collect more data
- Try a less advanced model

▼ Comparison

- Finally, when comparing two different models to each other, it's important to ensure you're comparing apples with apples and oranges with oranges
- Example

Experiment

	Inputs	Model	Outputs	Accuracy	Training time	Prediction time
1	→	Model 1	→	87.5%	3 min	0.5 sec
2	→	Model 2	→	91.3%	92 min	1 sec
3	→	Model 3	→	94.7%	176 min	4 sec

▼ Things to remember

- Avoid overfitting and underfitting
- Keep the test set separate at all costs
- Compare apples to apples
- One best performance metric does not equal best model