Evaluation

Evaluation Metrics

Breakout task: 15 mins

1. Go to: https://www.kaggle.com/competitions
2. Pick a competition from the list
3. Determine problem type: Classification? Regression? Other?!
4. Read the **Evaluation** tab

CAMBRIDGE SPARK

Metrics metrics metrics!

Classification:

- Accuracy (when is this misleading?)
- Mean class accuracy
- Precision:
  - How **precise** are your predictions of class A?
- Recall:
  - Of the data which are truly in class A, how many do you correctly classify?
- F-measures
- Log loss (cross-entropy)
- AUC

Regression:

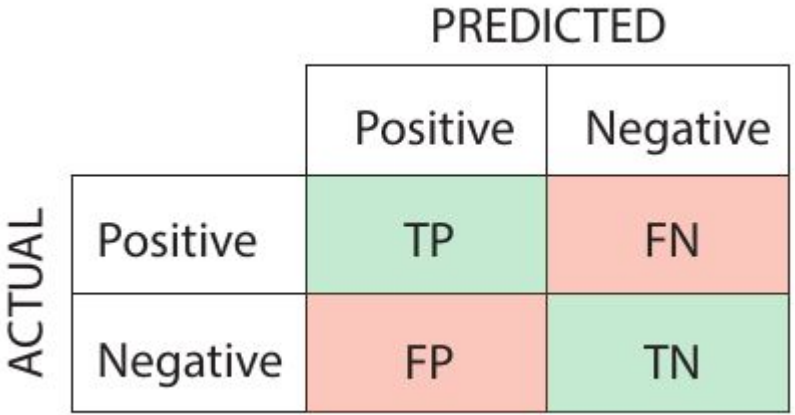
- RMSE - Root Mean Squared Error
- MAE - Mean Absolute Error
  - When would RMSE > MSE and vice versa
- MSLE - Mean Squared Logarithmic error
  - RMSE but log y and y_{pred}
  - Why?!
- R^2

CAMBRIDGE SPARK

True/False Positives/Negatives

- Confusing Terminology!
- True/False - whether your prediction was correct
- Positive/Negative - the class that you predicted
- Example:
  - A true negative: $y_{pred}=0$ and $y=0$
  - A false positive: $y_{pred}=1$ but $y=0$

Relation to metrics



Accuracy:

(TP + TN) / N

Sensitivity/Recall:

TP / (TP + FN)

Precision:

TP / (TP + FP)

CAMBRIDGE SPARK

Hands-on session

evaluation_metrics.ipynb (1 hour)

CAMBRIDGE SPARK

Model validation - Hyperparameter tuning example

For kNN we only need to tune the K. How do we do it?

Hyperparameter tuning

For kNN we only need to tune the K. How do we do it?

1. Train a model on training data
2. Test on the test data
3. Adjust the model
4. Repeat

CAMBRIDGE SPARK

Hyperparameter tuning

For kNN we only need to tune the K. How do we do it?

1. Train a model on training data
2. Test on the test data
3. Adjust the model
4. Repeat

Leads to information about the test set being leaked into the model! (overfitting)

Hyperparameter tuning

For kNN we only need to tune the K. How do we do it?

1. Train a model on training data
2. Test on the test data
3. Adjust the model
4. Repeat

Leads to information about the test set being leaked into the model! (eventually you'll learn a complex model that fits test perfectly - you'll overfit to the test set)

Hyperparameter tuning

For kNN we only need to tune the K. How do we do it?

1. Train a model on training data
2. Test on the test data
3. Adjust the model
4. Repeat

Leads to information about the test set being leaked into the model! (eventually you'll learn a complex model that fits test perfectly - you'll overfit to the test set)

You should only use the test set for a final assessment of the model...

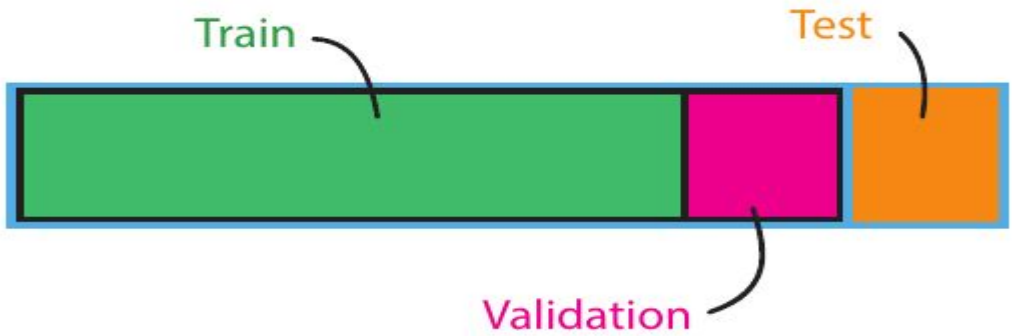
Training-Validation-Testing

1. Take a slice of the training data -> validation set
2. Apply the same procedure as before to adjust the model but using the smaller training set and validation set
3. Report performance on the untouched test set

CAMBRIDGE SPARK

Training-Validation-Testing

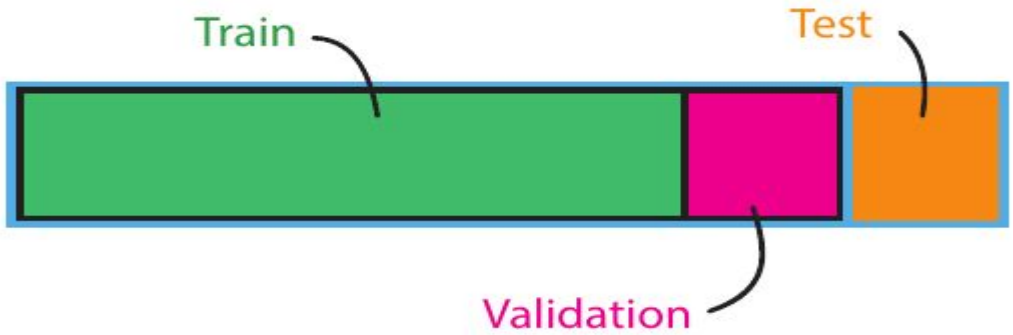
1. Take a slide of the training data -> validation set
2. Apply the same procedure as before to adjust the model but using the smaller training set and validation set
3. Report performance on the untouched test set



CAMBRIDGE SPARK

Training-Validation-Testing

1. Take a slide of the training data -> validation set
2. Apply the same procedure as before to adjust the model but using the smaller training set and validation set
3. Report performance on the untouched test set



Nice but we're not using the whole training data...

CAMBRIDGE SPARK

From validation to cross-validation

We would like to:

- Use something like validation to tune our model without looking at the test set
- Exploit all the data from the initial training set

CAMBRIDGE SPARK

N-fold cross validation

1. Shuffle the training set and slice it in N parts (or folds)

For each part n:

2. Hold part n out as validation data
3. Use the rest of data to train the model
4. Evaluate the model on the held out part n

After evaluated on each of the N folds:

5. Average the evaluations
6. Repeat entire process for each model or hyperparameter setting you want to test

CAMBRIDGE SPARK

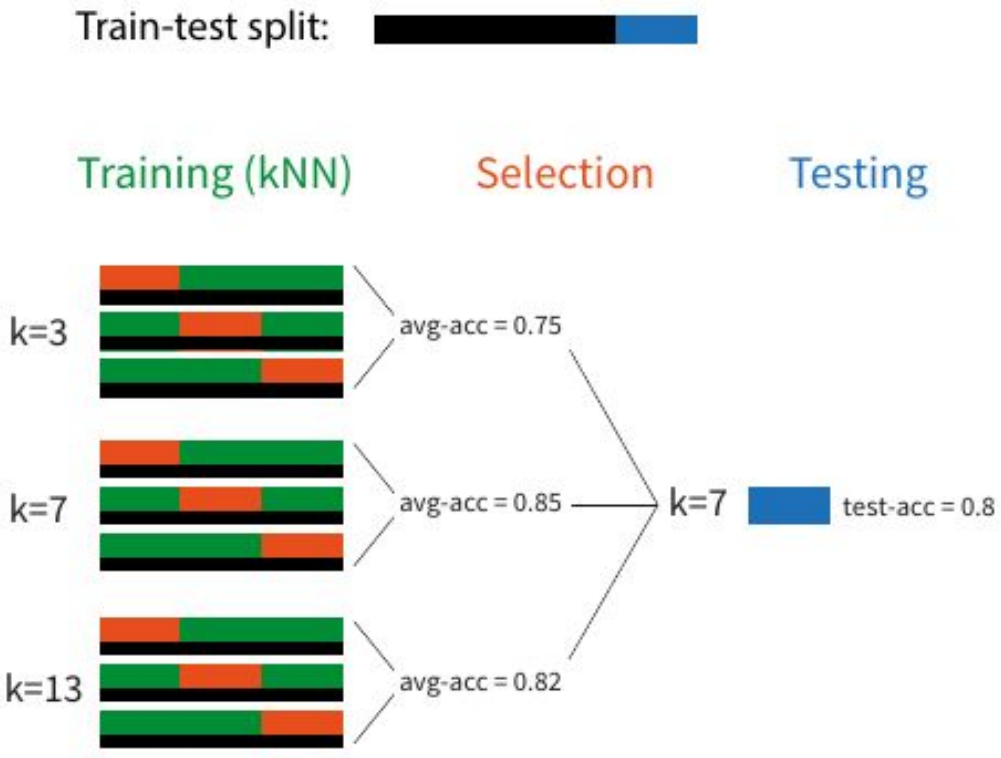
3-fold cross validation example

Here we are doing a hyperparameter comparison of knn with k = {3, 7, 13}.

For each model, we perform 3-fold cross validation.

We then compare the average accuracy and decide to use k=7

Finally, we evaluate on the held out test set.

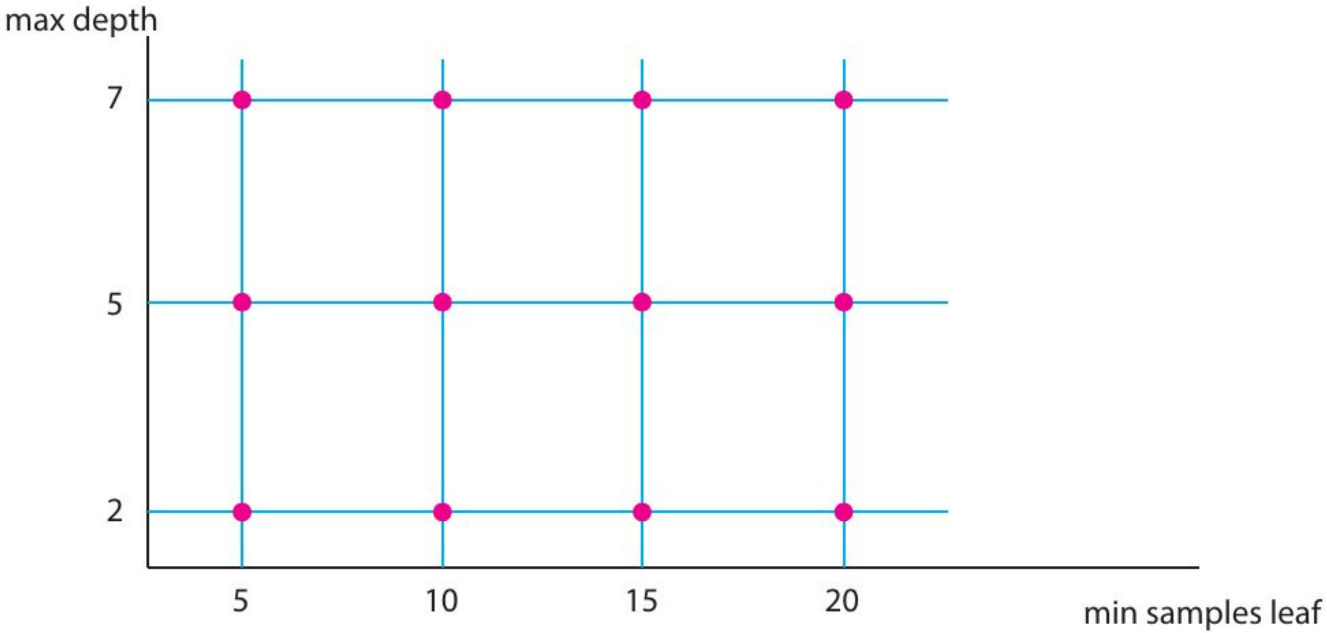


CAMBRIDGE SPARK

N-fold cross validation

Now that we have a way to evaluate models with different parameters, how do we choose the parameters we want to test?

Grid Search



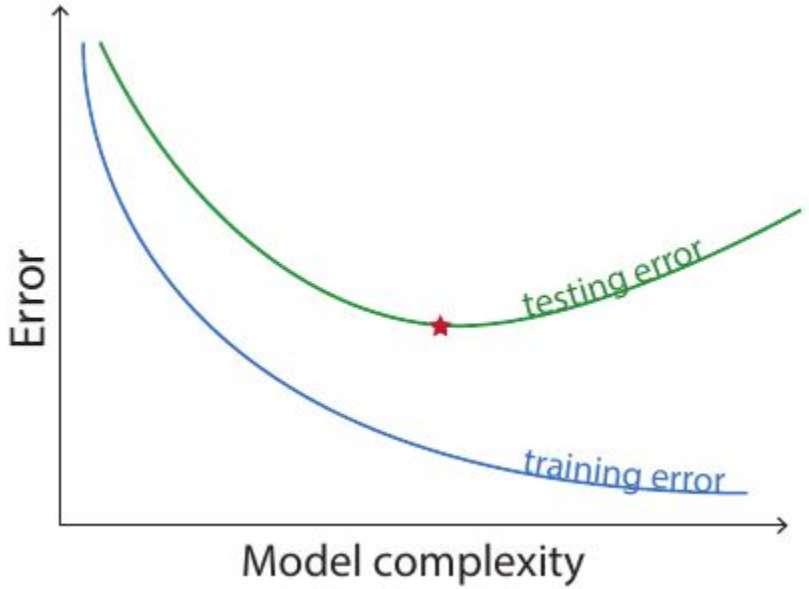
CAMBRIDGE SPARK

Hands-on session

model_validation.ipynb (15 mins)

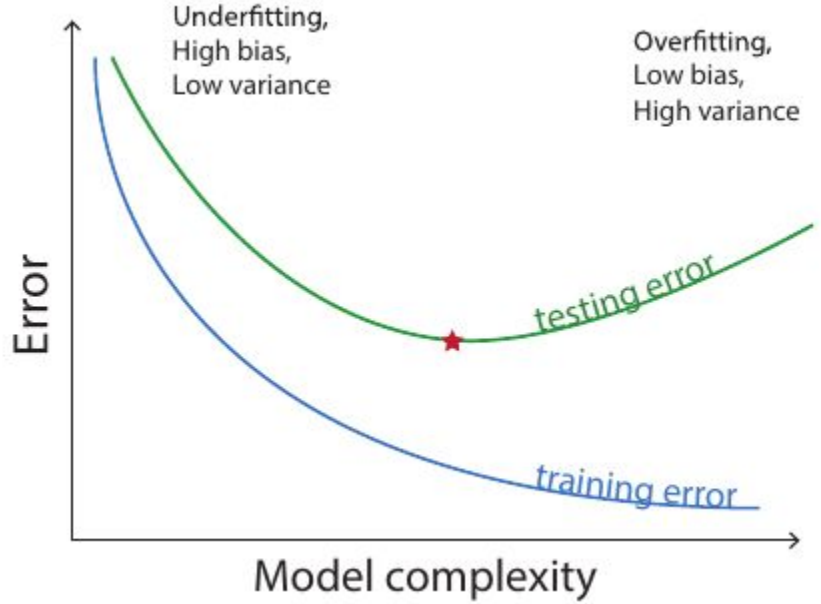
CAMBRIDGE SPARK

A reminder of the goal...



CAMBRIDGE SPARK

Bias-Variance Tradeoff



CAMBRIDGE SPARK

Bias and Variance

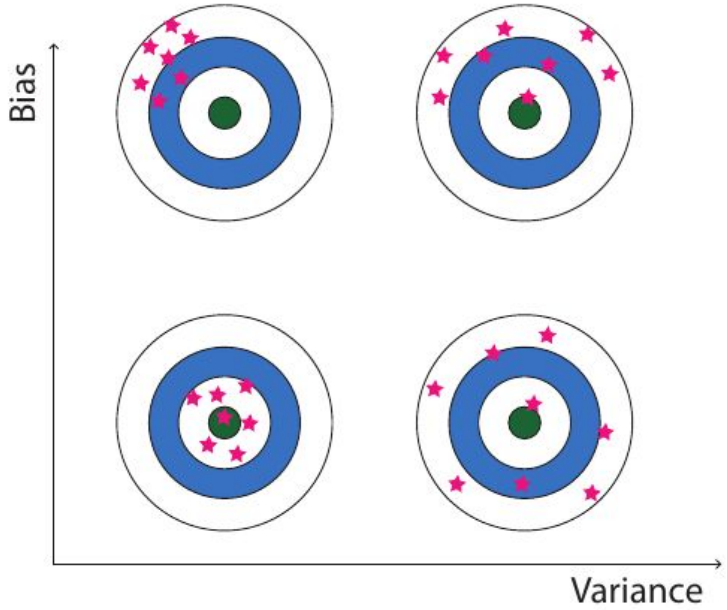
Think of your model as a darts player. One fit of your model is a single dart throw. Each dart board is the result from one player/model.

How would your fit have changed if your data was slightly different?

- Small change -> the model has low variance
- Desirable!
- High variance models could ‘get lucky’ (see bottom right model - one fit is a near bullseye)

How accurate is your model averaged over different training data?

- Low average error -> low bias
- Watch out: variant but unbiased models are still rubbish!



CAMBRIDGE SPARK

Further reading

Look no further than the scikit learn docs:

http://scikit-learn.org/stable/modules/model_evaluation.html