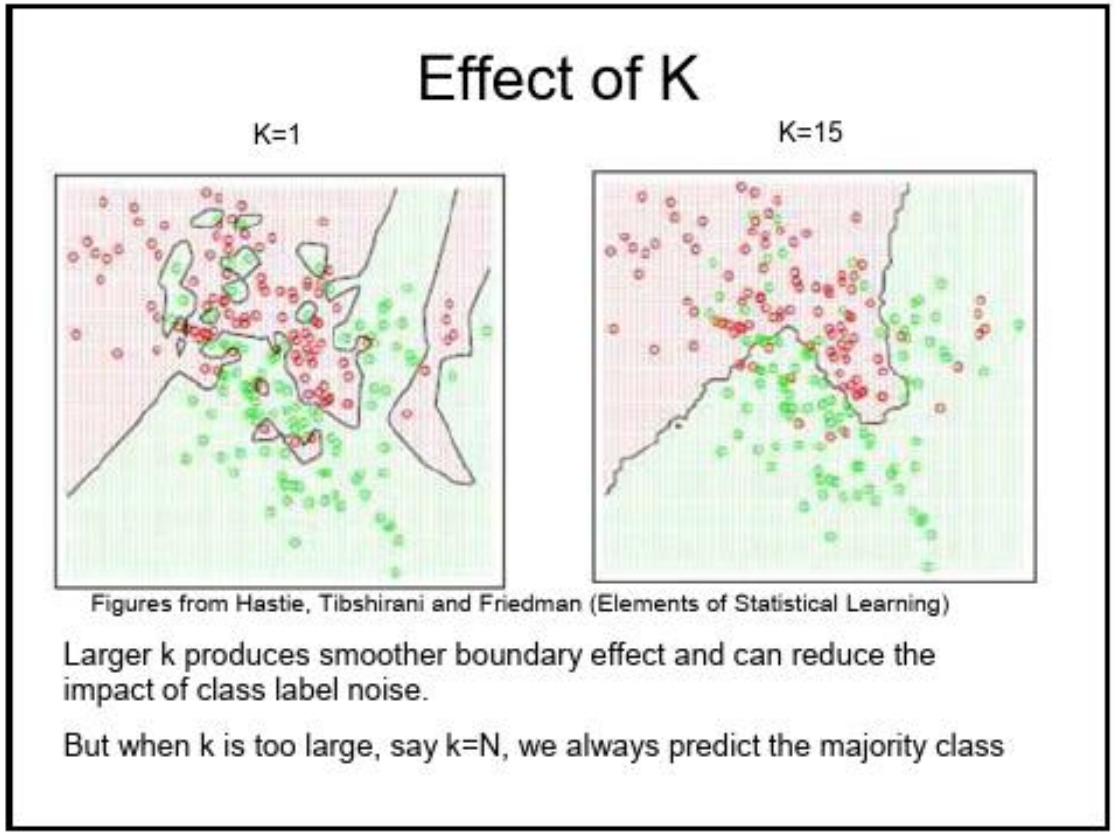
Feature Engineering and Selection…
Chapter 3: A Review of the Predictive Modeling Process
pt. 3

Agenda

- **3.5 Tuning Parameters and Overfitting**
- **3.6 Model Optimization and Tuning**
- **3.7 Comparing Models Using the Training Set**
- **3.8 Feature Engineering Without Overfitting**

3.5 Tuning parameters and overfitting

- Tuning parameters (AKA hyperparameters): *"Many models include parameters that, while important, cannot be directly estimated from the data."*
- E.g. K in KNN (K Nearest Neighbors)



https://cse3521.artifice.cc/images/effect-of-k.png

3.6 Model Optimization and Tuning

- Tune *K = 1, 3, … 201*
- Using 10-fold cross-validation (colored: individual folds; black : average)

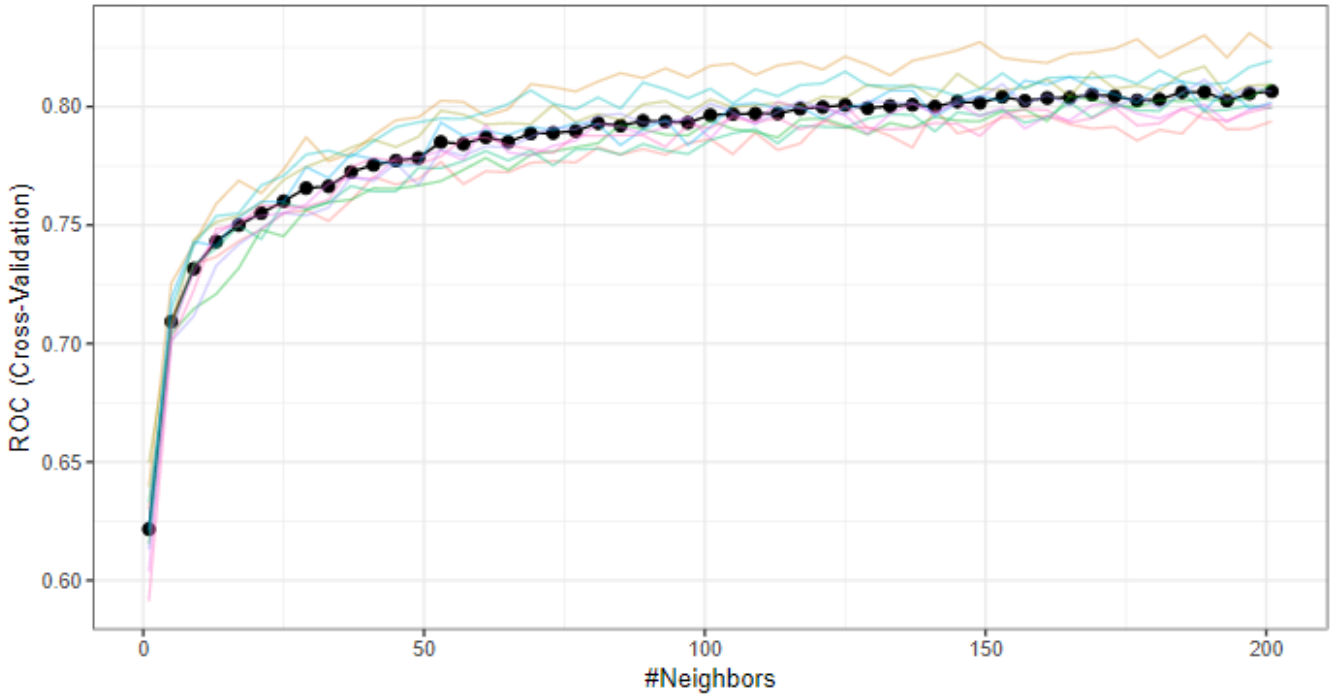


Figure 3.11: The resampling profile generated by a simple grid search on the number of neighbors in a K-NN classification model for the OkC data. The black line represents the averaged performance across 10 resamples while the other lines represent the performance across each individual resample.

3.6 Model Optimization and Tuning

- Tune *K = 1, 3, … 201*
- Using 10-fold cross-validation (colored: individual folds; black : average)
- Can have lots of hyper parameters…

What problems might this cause?

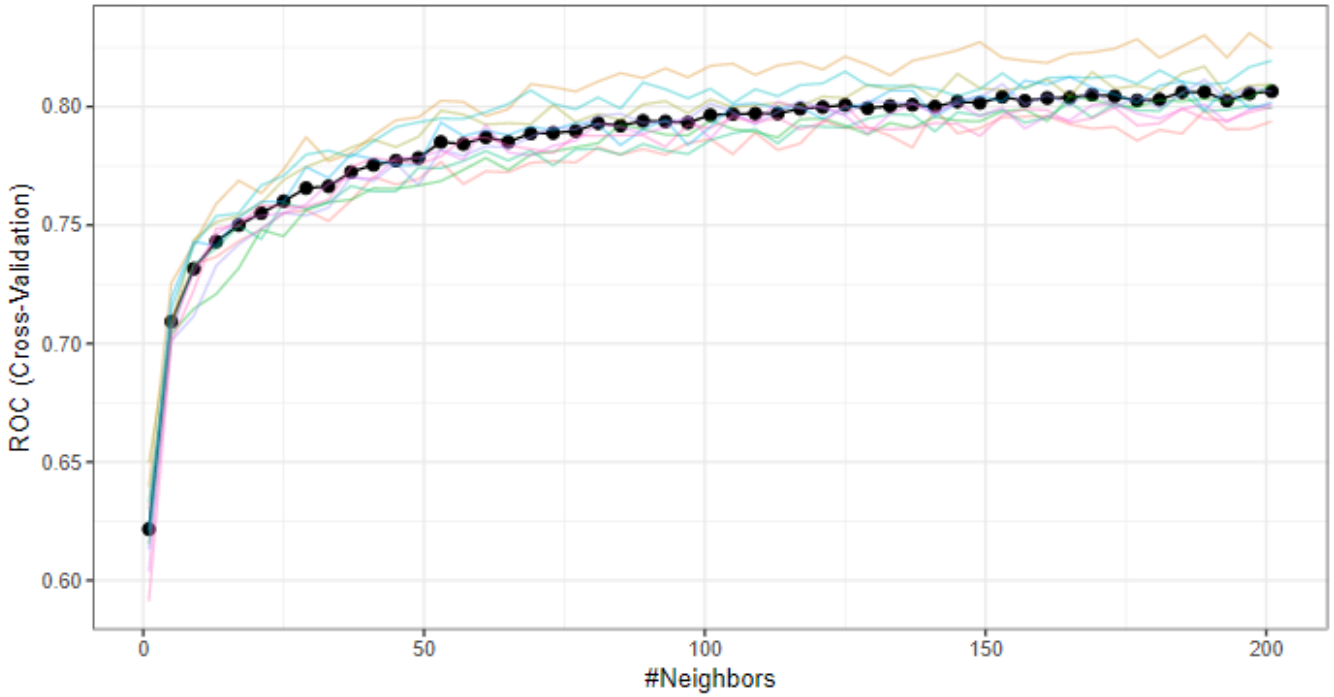


Figure 3.11: The resampling profile generated by a simple grid search on the number of neighbors in a K-NN classification model for the OkC data. The black line represents the averaged performance across 10 resamples while the other lines represent the performance across each individual resample.

Table 3.3: The settings and results for a random search of the neural network parameter space.

Units	Dropout	Batch Size	Learn Rate	Grad. Scaling	Decay	Act. Fun.

3.6 Model Optimization and Tuning

- Tune *K = 1, 3, … 201*
- Using 10-fold cross-validation (colored: individual folds; black : average)
- Can have lots of hyper parameters…
- 2 main search strategies
  1. Predefine

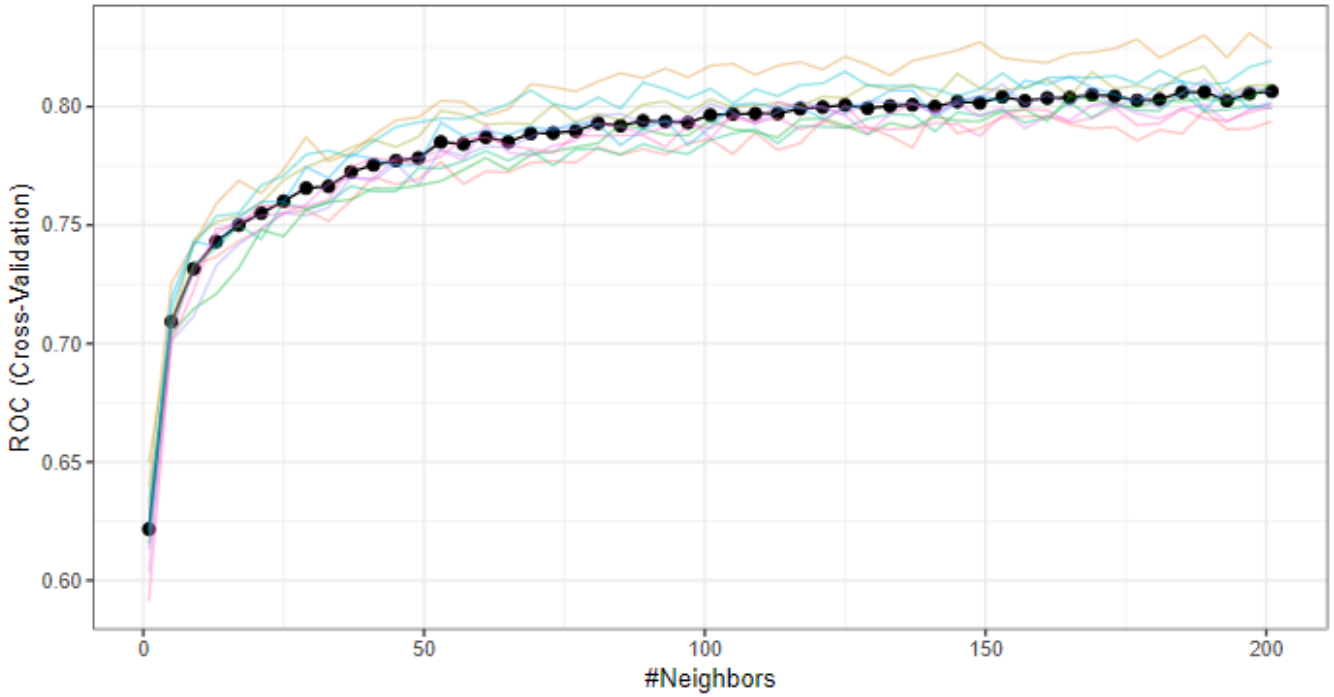


Figure 3.11: The resampling profile generated by a simple grid search on the number of neighbors in a K-NN classification model for the OkC data. The black line represents the averaged performance across 10 resamples while the other lines represent the performance across each individual resample.

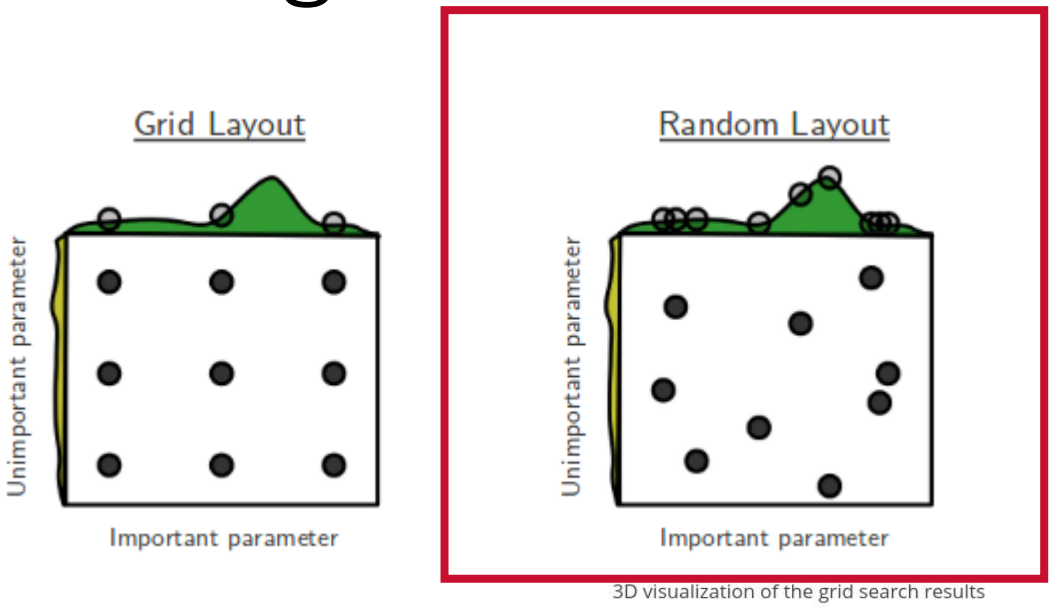
Table 3.3: The settings and results for a random search of the neural network parameter space.

Units	Dropout	Batch Size	Learn Rate	Grad. Scaling	Decay	Act. Fun.

3.6 Model Optimization and Tuning

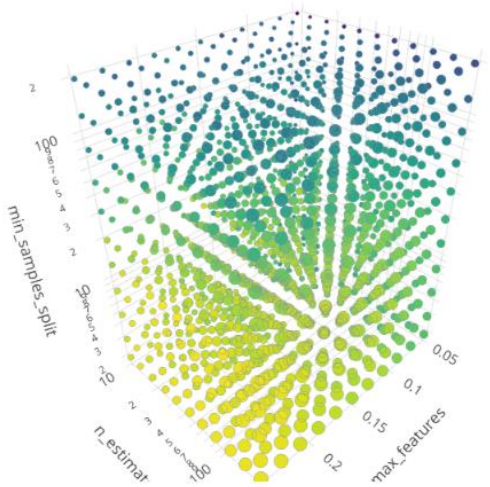
Grid Search & Random Search (predefine)

- Grid search: “a multidimensional grid search can be conducted where candidate parameter combinations and the grid of combinations are evaluated… **this can be very inefficient.**”
- Random search: “Define a range of possible values for each parameter and to randomly sample the multidimensional space enough times to cover a reasonable amount…” of the parameter space



3D visualization of the grid search results

As # of parameters grows, search space grows exponentially.



* https://medium.com/@senapati.dipak97/grid-search-vs-random-search-d34c92946318
* https://towardsdatascience.com/using-3d-visualizations-to-tune-hyperparameters-of-ml-models-with-python-ba2885eab2e9

3.6 Model Optimization and Tuning

Example random search in simple feed forward neural network

Table 3.3: The settings and results for a random search of the neural network parameter space.

Units	Dropout	Batch Size	Learn Rate	Grad. Scaling	Decay	Act. Fun.	ROC
9	*0.592*	*26231*	*0.00398*	*0.45653*	*4.46e-02*	*relu*	*0.837*
16	0.572	25342	0.63142	0.29769	5.10e-01	relu	0.836
13	0.600	10196	0.33286	0.00675	1.76e-01	relu	0.835
10	0.286	19893	0.94679	0.49468	4.61e-01	relu	0.832
8	0.534	36267	0.38651	0.50524	1.49e-01	relu	0.832
14	0.617	21154	0.25216	0.29679	6.54e-03	sigmoid	0.829
17	0.250	11638	0.22683	0.72089	1.38e-02	tanh	0.825
20	0.392	17235	0.01681	0.36270	2.90e-04	relu	0.821
16	0.194	8477	0.20389	0.15507	1.82e-02	tanh	0.820
10	0.282	18989	0.87800	0.25988	6.39e-02	tanh	0.820

Units	Dropout	Batch Size	Learn Rate	Grad. Scaling	Decay	Act. Fun.	ROC
2	0.586	19404	0.72107	0.15870	1.85e-01	sigmoid	0.818
18	0.305	5563	0.83560	0.45655	1.05e-02	sigmoid	0.805
10	0.513	17877	0.49598	0.76990	3.61e-05	tanh	0.796
4	0.536	32689	0.71944	0.48349	1.11e-04	sigmoid	0.755
12	0.150	4015	0.55811	0.21277	1.37e-05	relu	0.732
14	0.368	7582	0.93841	0.12600	1.04e-05	relu	0.718
3	0.688	12905	0.63325	0.69805	1.36e-05	relu	0.598
3	0.645	5327	0.95232	0.71696	1.47e-05	sigmoid	0.560
2	0.688	14660	0.68575	0.44476	3.26e-05	relu	0.546
15	0.697	4800	0.80618	0.13876	3.23e-05	sigmoid	0.538

3.6 Model Optimization and Tuning

Incremental search strategies

- Bayesian optimization: “initial pool of samples are evaluated using grid or random search. The optimization procedure creates a separate model to predict performance as a function of the tuning parameters and can then make a recommendation as to the next candidate set to evaluate.”
- Nelder-Mead simplex search procedure
- Simulated annealing
- Genetic algorithms
- **Chapter 12 will go into in more depth…**

3.7 Comparing Models Using the Training Set

- "When multiple models are in contention, there is often the need to have formal evaluations between them to understand if any differences in performance are above and beyond what one would expect at random."
- Logistic Regression vs. Neural Network
- Performance on each of 10-fold cross validation

Table 3.4: Matched resampling results for two models for predicting the OkCupid data. The ROC metric was used to tune each model. Because each model uses the same resampling sets, we can formally compare the performance between the models.

	ROC Estimates		
	Logistic Regression	Neural Network	Difference
Fold 1	0.830	0.827	-0.003
Fold 2	0.854	0.853	-0.002
Fold 3	0.844	0.843	-0.001
Fold 4	0.836	0.834	-0.001
Fold 5	0.838	0.834	-0.004
Fold 6	0.840	0.833	-0.007
Fold 7	0.839	0.838	-0.001
Fold 8	0.837	0.837	0.000
Fold 9	0.835	0.832	-0.003
Fold 10	0.838	0.835	-0.003

What are scenarios you might want a formal performance test?

3.7 Comparing Models Using the Training Set

- <u>Simple approach:</u>
- one-sample or paired t-test between ROC estimates
  1. It prevents the test set from being used during the model development process and
  2. Many evaluations (via assessment sets) are used to gauge the differences.
- *"estimated difference in the ROC values is -0.003 with 95% confidence interval (-0.004, -0.001)"*
- (See refs in book to Bayesian hierarchical model or repeated measures model if comparing > 2 models)

Table 3.4: Matched resampling results for two models for predicting the OkCupid data. The ROC metric was used to tune each model. Because each model uses the same resampling sets, we can formally compare the performance between the models.

	ROC Estimates		
	Logistic Regression	Neural Network	Difference
Fold 1	0.830	0.827	-0.003
Fold 2	0.854	0.853	-0.002
Fold 3	0.844	0.843	-0.001
Fold 4	0.836	0.834	-0.001
Fold 5	0.838	0.834	-0.004
Fold 6	0.840	0.833	-0.007
Fold 7	0.839	0.838	-0.001
Fold 8	0.837	0.837	0.000
Fold 9	0.835	0.832	-0.003
Fold 10	0.838	0.835	-0.003

3.8 Feature Engineering Without Overfitting

- When tuning models: *“The crux of this approach was to evaluate a parameter value on data that were not used to build the model.”*
- *“ the same idea should be applied to any other feature-related activity, such as engineering new features/encodings or when deciding on whether to include a new term into the model. There should always be a chance to have an independent piece of data evaluate the appropriateness of the design choice.”*