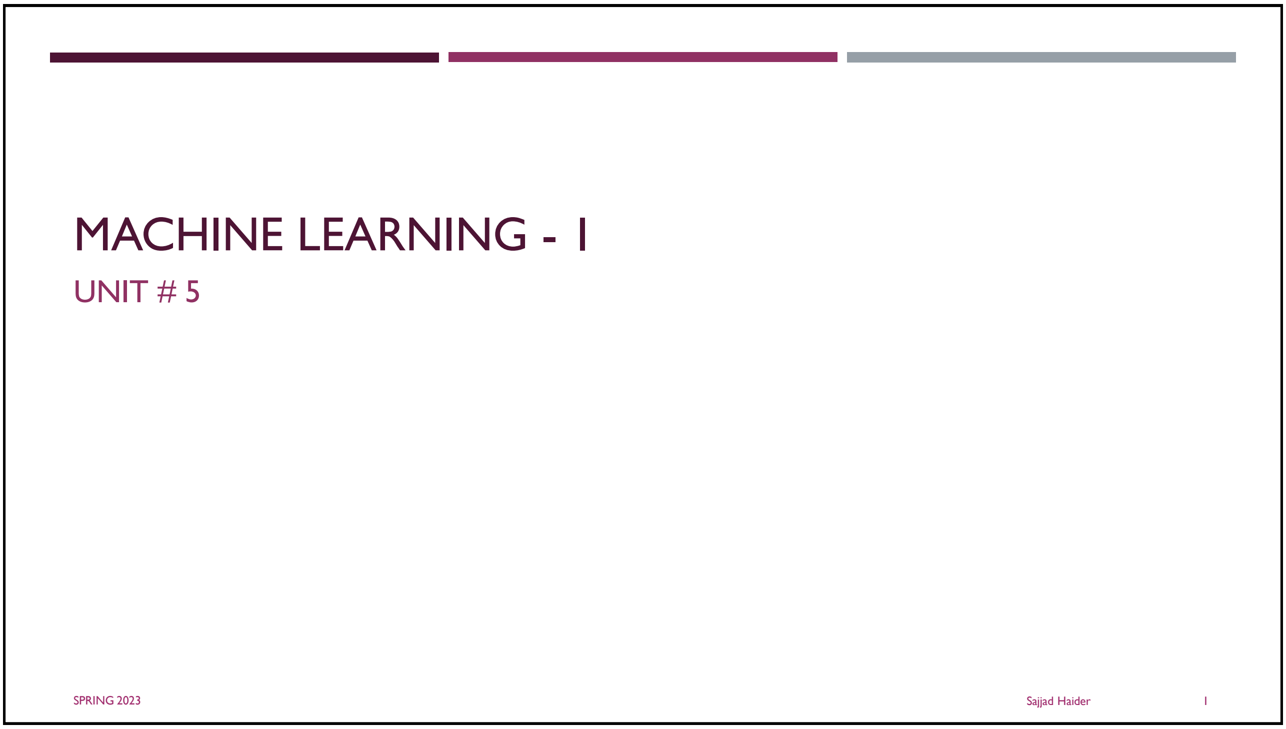
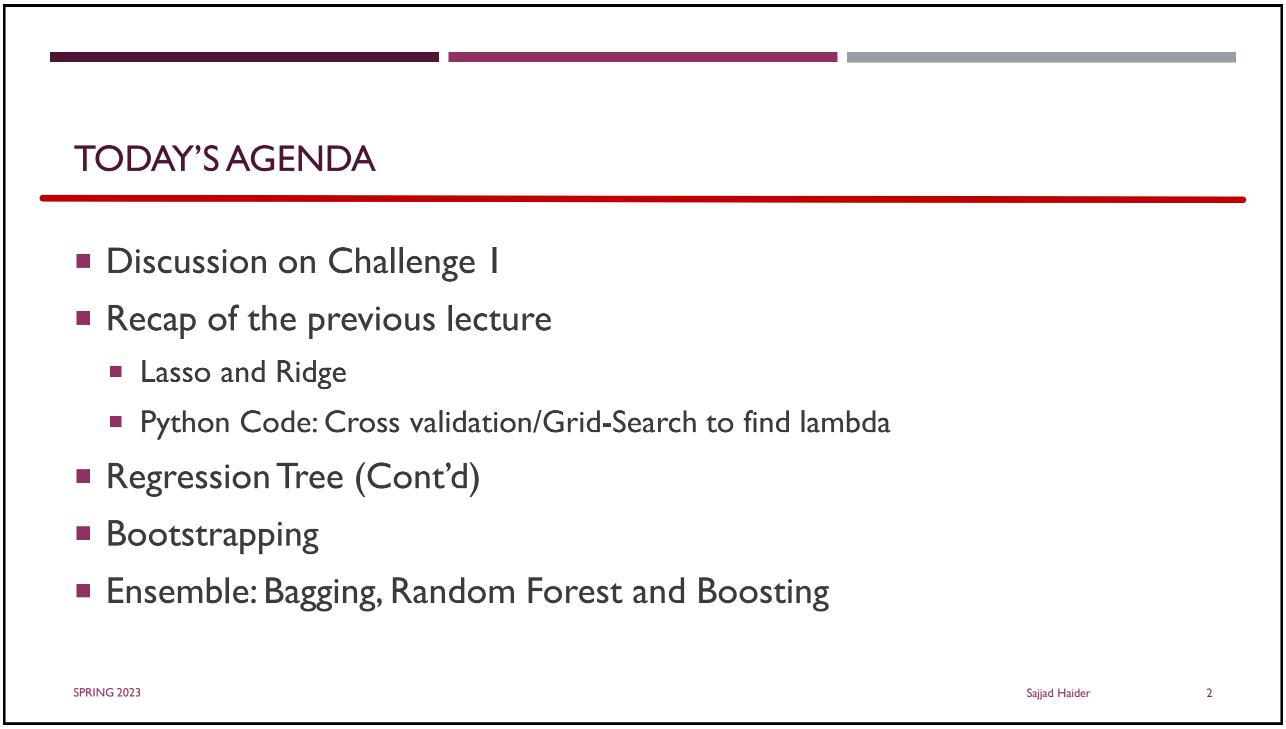


MACHINE LEARNING - I

UNIT # 5

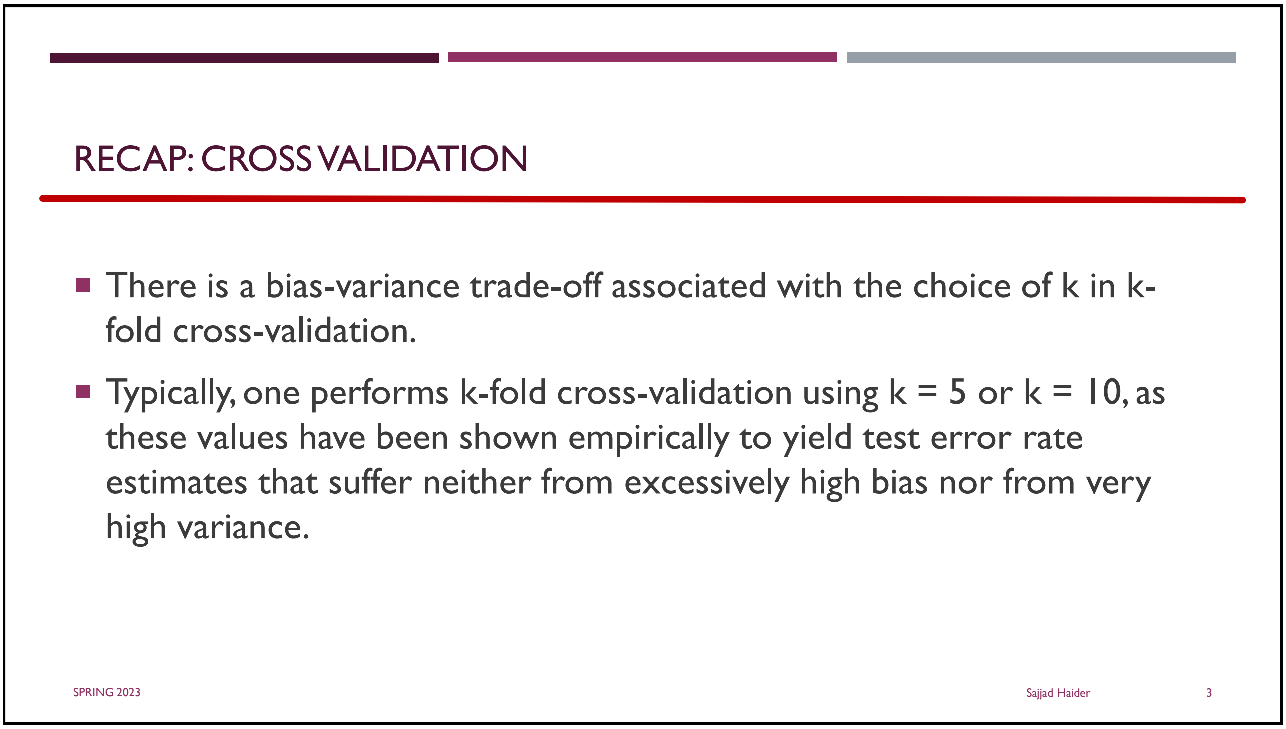
1



TODAY'S AGENDA

- Discussion on Challenge I
- Recap of the previous lecture
  - Lasso and Ridge
  - Python Code: Cross validation/Grid-Search to find lambda
- Regression Tree (Cont'd)
- Bootstrapping
- Ensemble: Bagging, Random Forest and Boosting

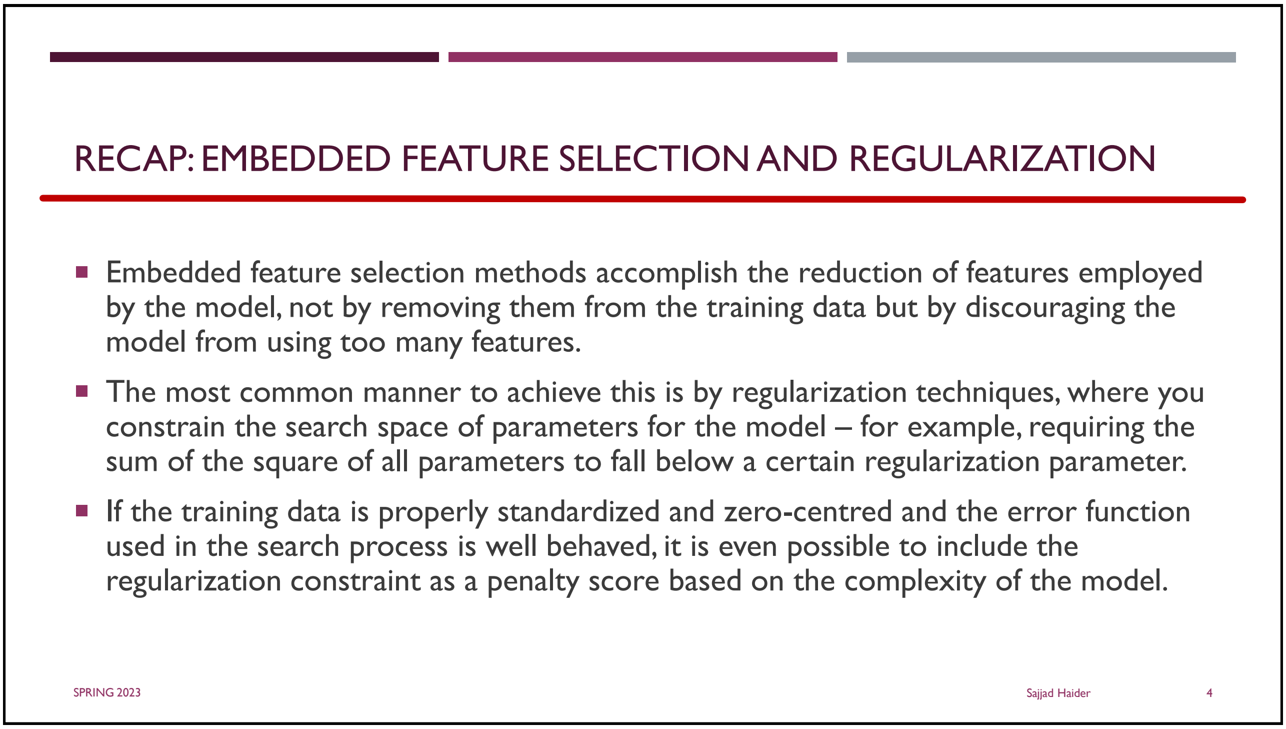
2



RECAP: CROSS VALIDATION

- There is a bias-variance trade-off associated with the choice of k in k-fold cross-validation.
- Typically, one performs k-fold cross-validation using k = 5 or k = 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

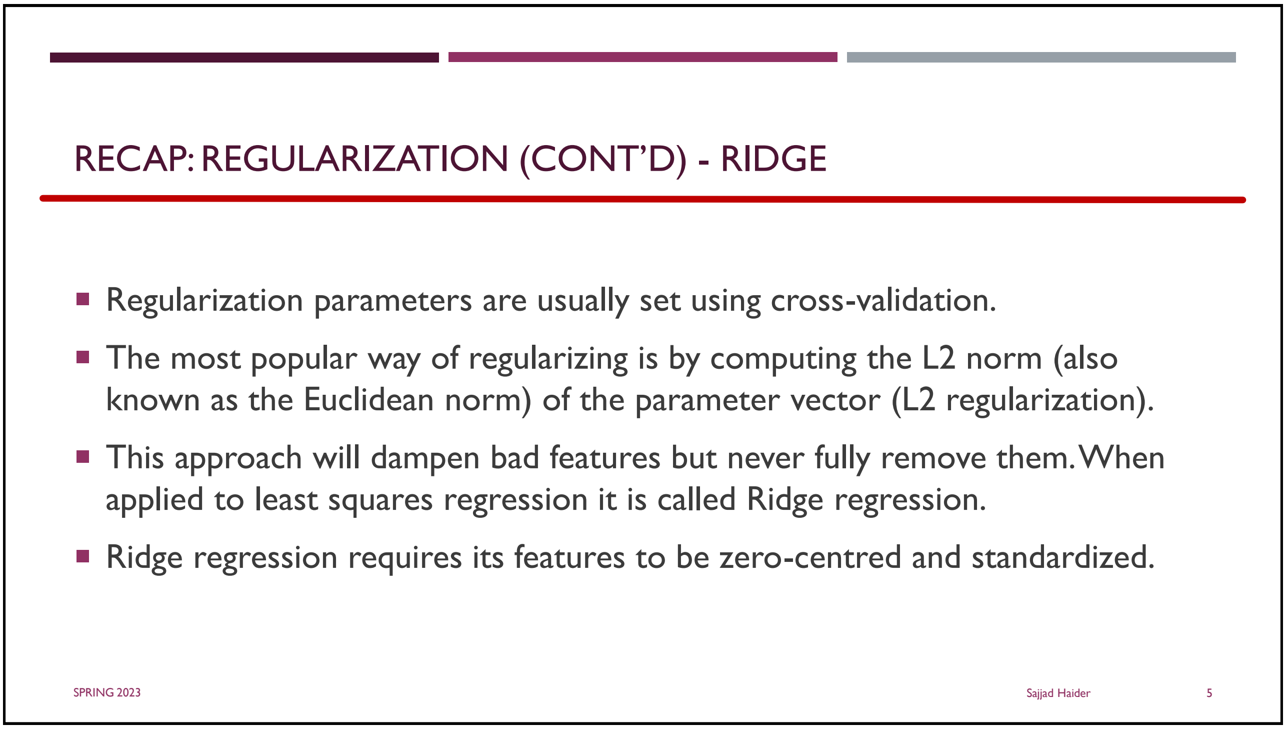
3



RECAP: EMBEDDED FEATURE SELECTION AND REGULARIZATION

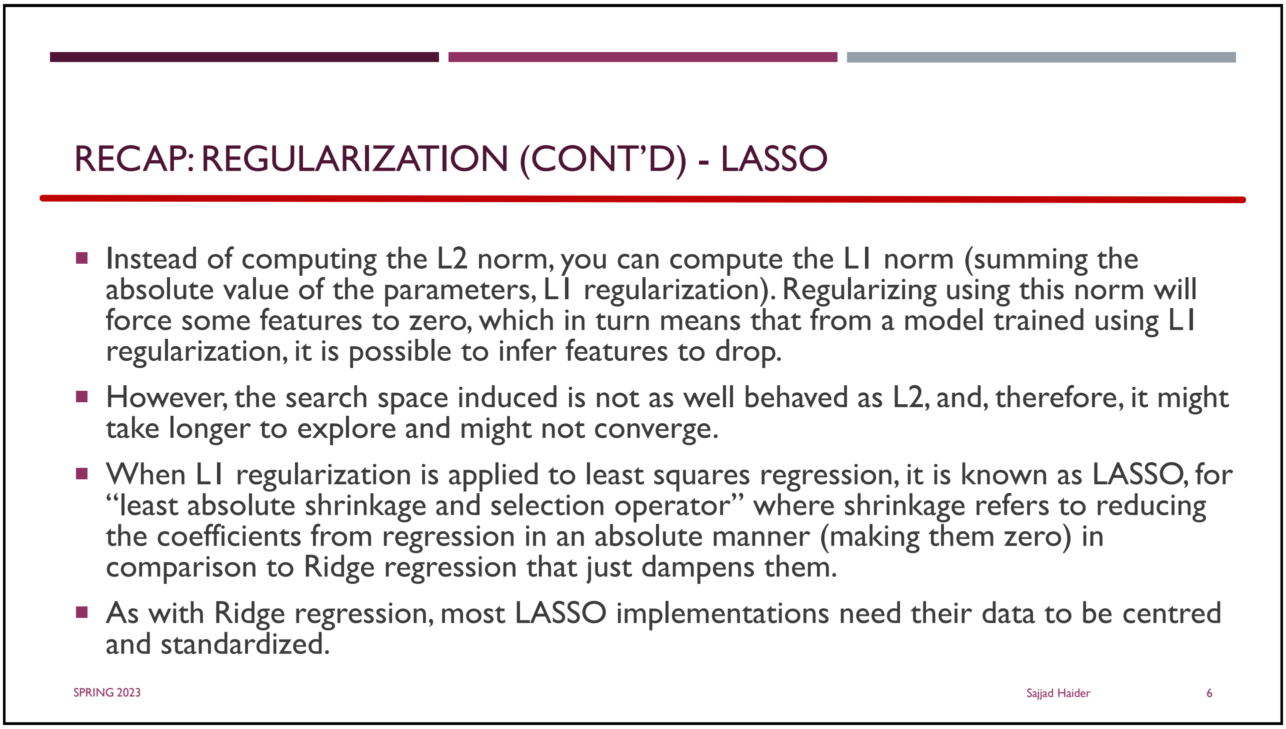
- Embedded feature selection methods accomplish the reduction of features employed by the model, not by removing them from the training data but by discouraging the model from using too many features.
- The most common manner to achieve this is by regularization techniques, where you constrain the search space of parameters for the model – for example, requiring the sum of the square of all parameters to fall below a certain regularization parameter.
- If the training data is properly standardized and zero-centred and the error function used in the search process is well behaved, it is even possible to include the regularization constraint as a penalty score based on the complexity of the model.

4



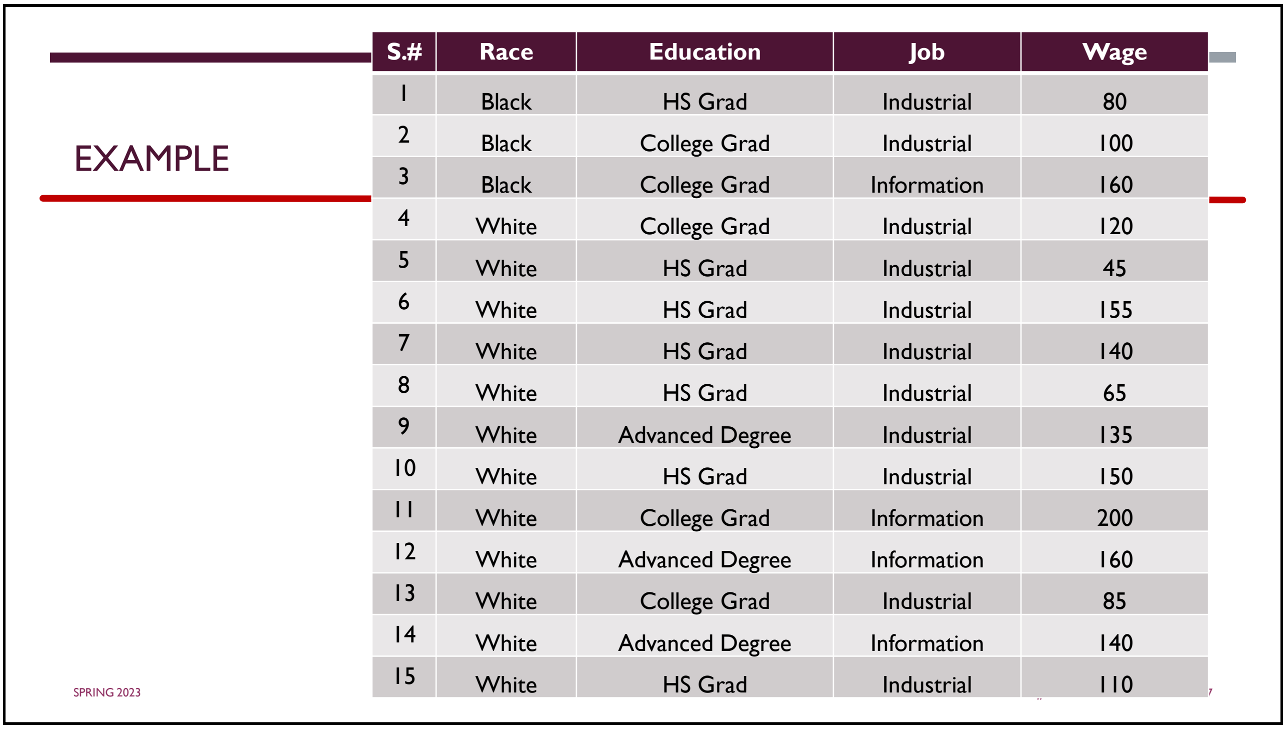
RECAP: REGULARIZATION (CONT'D) - RIDGE

- Regularization parameters are usually set using cross-validation.
- The most popular way of regularizing is by computing the L2 norm (also known as the Euclidean norm) of the parameter vector (L2 regularization).
- This approach will dampen bad features but never fully remove them. When applied to least squares regression it is called Ridge regression.
- Ridge regression requires its features to be zero-centred and standardized.



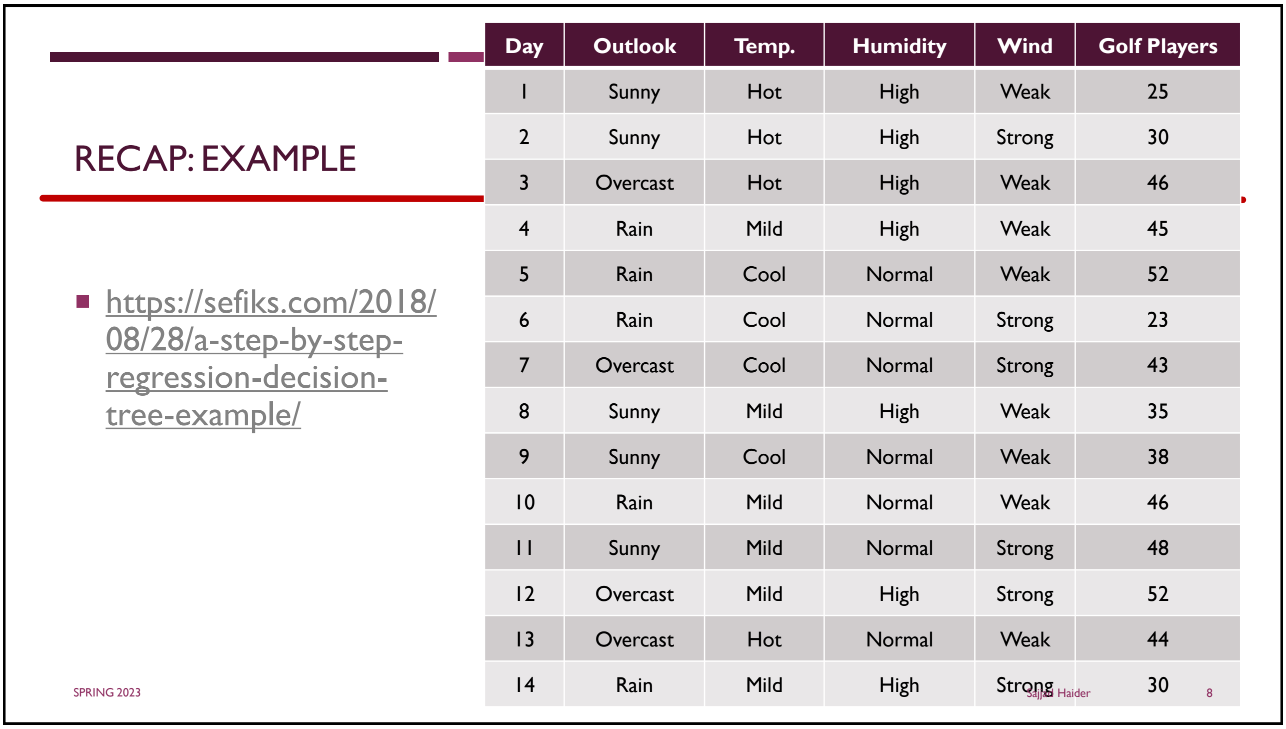
RECAP: REGULARIZATION (CONT'D) - LASSO

- Instead of computing the L2 norm, you can compute the L1 norm (summing the absolute value of the parameters, L1 regularization). Regularizing using this norm will force some features to zero, which in turn means that from a model trained using L1 regularization, it is possible to infer features to drop.
- However, the search space induced is not as well behaved as L2, and, therefore, it might take longer to explore and might not converge.
- When L1 regularization is applied to least squares regression, it is known as LASSO, for "least absolute shrinkage and selection operator" where shrinkage refers to reducing the coefficients from regression in an absolute manner (making them zero) in comparison to Ridge regression that just dampens them.
- As with Ridge regression, most LASSO implementations need their data to be centred and standardized.



EXAMPLE

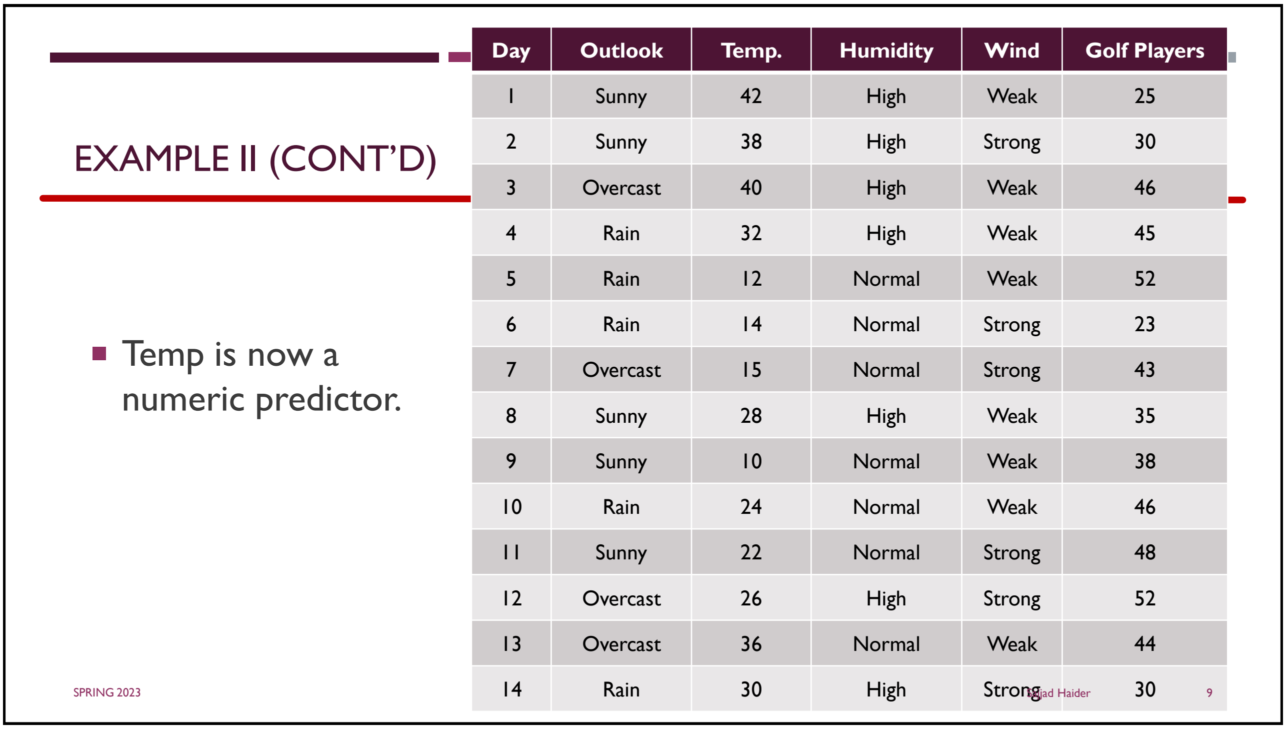
S.#	Race	Education	Job	Wage
1	Black	HS Grad	Industrial	80
2	Black	College Grad	Industrial	100
3	Black	College Grad	Information	160
4	White	College Grad	Industrial	120
5	White	HS Grad	Industrial	45
6	White	HS Grad	Industrial	155
7	White	HS Grad	Industrial	140
8	White	HS Grad	Industrial	65
9	White	Advanced Degree	Industrial	135
10	White	HS Grad	Industrial	150
11	White	College Grad	Information	200
12	White	Advanced Degree	Information	160
13	White	College Grad	Industrial	85
14	White	Advanced Degree	Information	140
15	White	HS Grad	Industrial	110



RECAP: EXAMPLE

- https://sefiks.com/2018/08/28/a-step-by-step-regression-decision-tree-example/

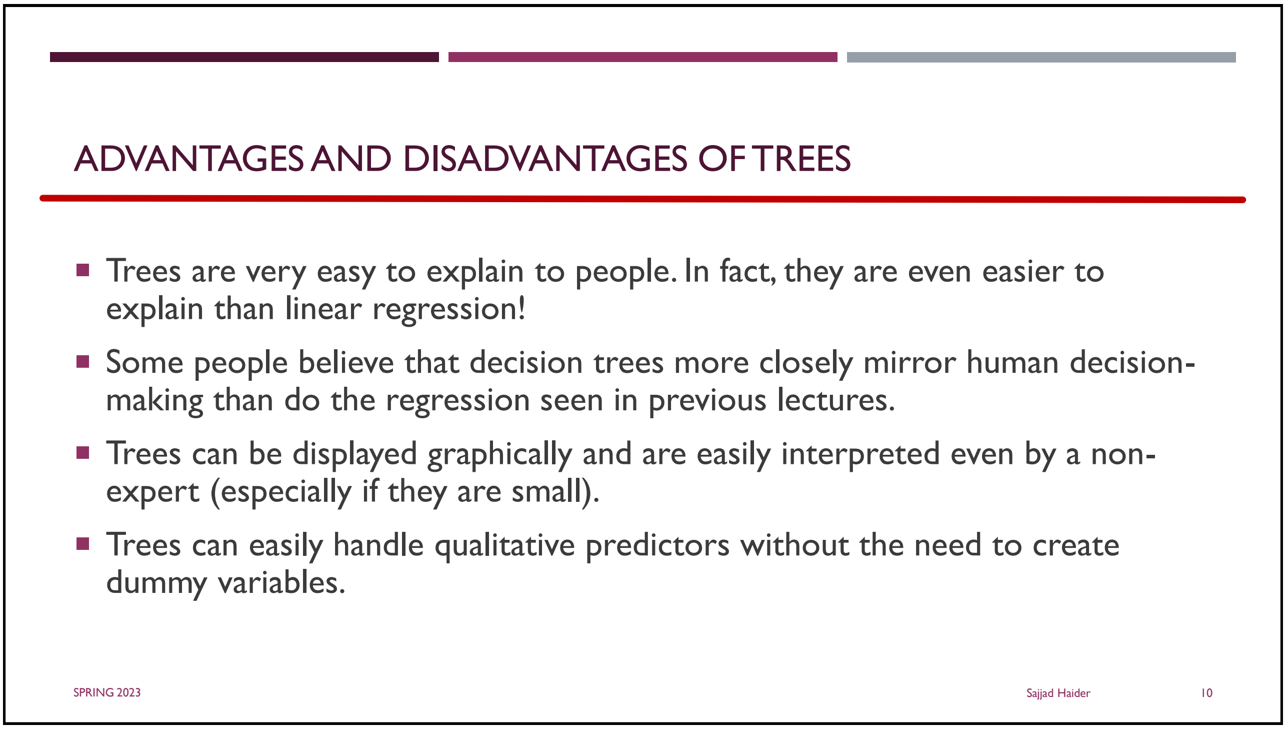
Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30



EXAMPLE II (CONT'D)

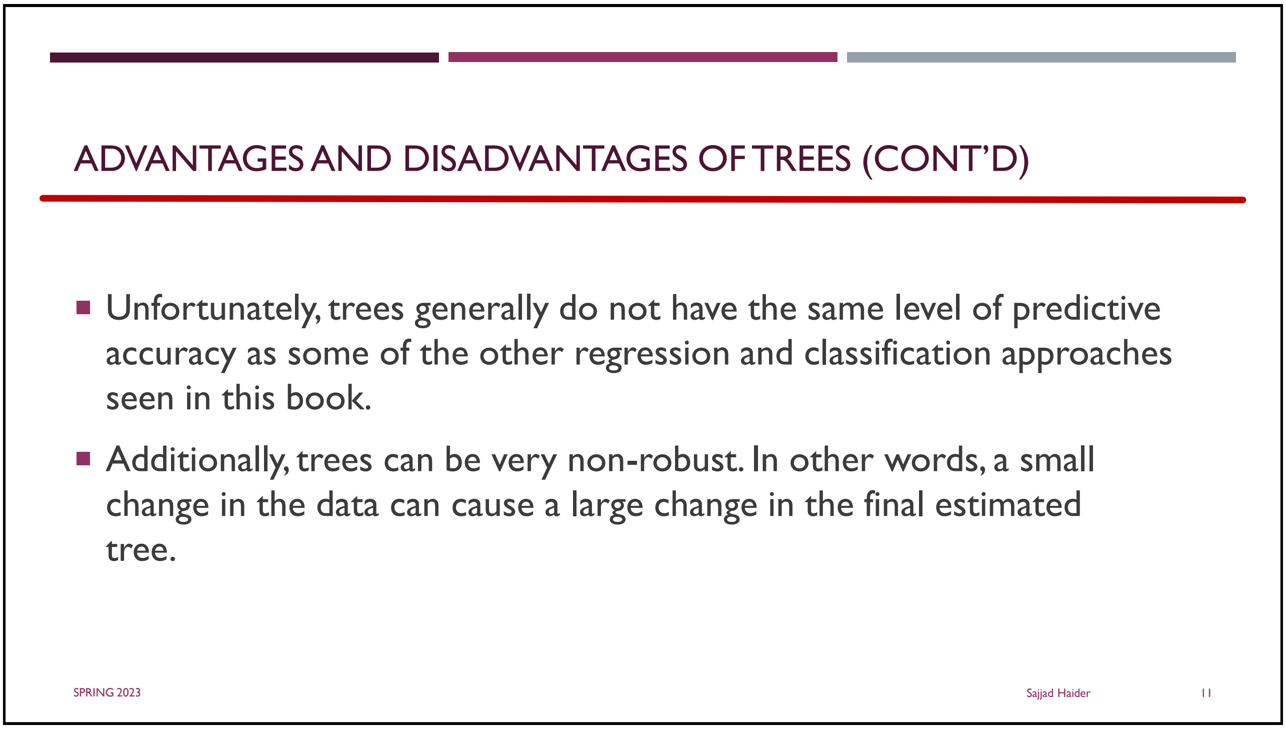
- Temp is now a numeric predictor.

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	42	High	Weak	25
2	Sunny	38	High	Strong	30
3	Overcast	40	High	Weak	46
4	Rain	32	High	Weak	45
5	Rain	12	Normal	Weak	52
6	Rain	14	Normal	Strong	23
7	Overcast	15	Normal	Strong	43
8	Sunny	28	High	Weak	35
9	Sunny	10	Normal	Weak	38
10	Rain	24	Normal	Weak	46
11	Sunny	22	Normal	Strong	48
12	Overcast	26	High	Strong	52
13	Overcast	36	Normal	Weak	44
14	Rain	30	High	Strong	30



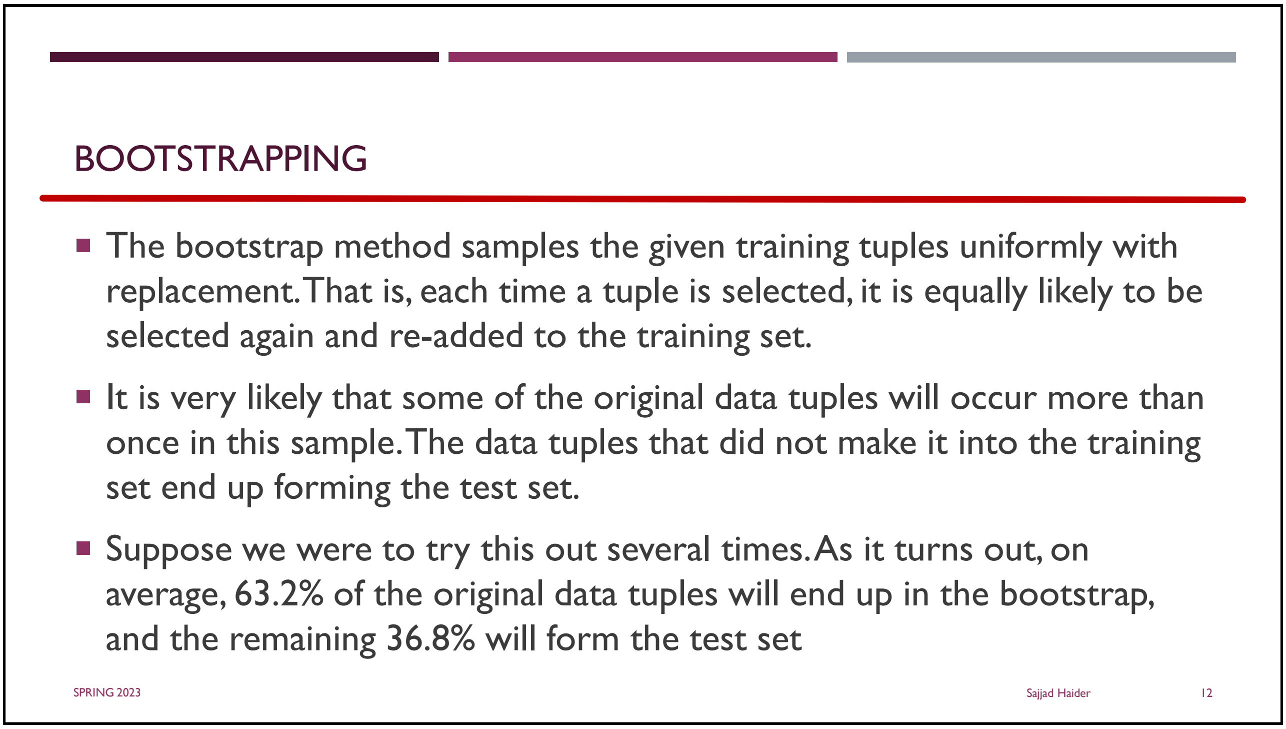
ADVANTAGES AND DISADVANTAGES OF TREES

- Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- Some people believe that decision trees more closely mirror human decision-making than do the regression seen in previous lectures.
- Trees can be displayed graphically and are easily interpreted even by a non-expert (especially if they are small).
- Trees can easily handle qualitative predictors without the need to create dummy variables.



ADVANTAGES AND DISADVANTAGES OF TREES (CONT'D)

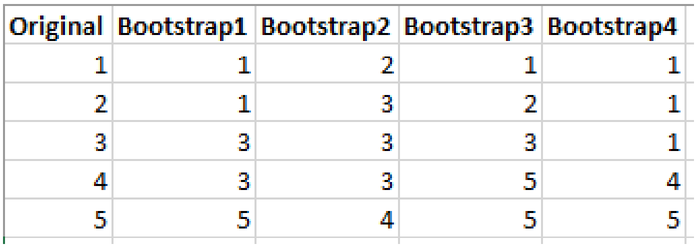
- Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches seen in this book.
- Additionally, trees can be very non-robust. In other words, a small change in the data can cause a large change in the final estimated tree.



BOOTSTRAPPING

- The bootstrap method samples the given training tuples uniformly with replacement. That is, each time a tuple is selected, it is equally likely to be selected again and re-added to the training set.
- It is very likely that some of the original data tuples will occur more than once in this sample. The data tuples that did not make it into the training set end up forming the test set.
- Suppose we were to try this out several times. As it turns out, on average, 63.2% of the original data tuples will end up in the bootstrap, and the remaining 36.8% will form the test set

BOOTSTRAPPING (CONT'D)



Original	Bootstrap1	Bootstrap2	Bootstrap3	Bootstrap4
1	1	2	1	1
2	1	3	2	1
3	3	3	3	1
4	3	3	5	4
5	5	4	5	5

- Source: https://statisticsbyjim.com/hypothesis-testing/bootstrapping/

ENSEMBLE METHOD

- An ensemble method is an approach that combines many simple “building block" models in order to obtain a single and potentially very powerful model.
- These simple building block models are sometimes known as weak learners.
- Popular approaches include bagging, boosting, random forest and stacking.

BAGGING

- The decision trees suffer from high variance.
- This means that if we split the training data into two parts at random, and fit a decision tree to both halves, the results that we get could be quite different.
- In contrast, a procedure with low variance will yield similar results if applied repeatedly to distinct data sets; linear regression tends to have low variance, if the ratio of n to p is moderately large.
- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a statistical learning method.
- In bagging, we take repeated samples from the (single) training data set and generate B different bootstrapped training data sets.

BAGGING (CONT'D)

- To apply bagging to regression trees, we simply construct B regression trees using B bootstrapped training sets and average the resulting predictions.
- These trees are grown deep and are not pruned. Hence each individual tree has high variance, but low bias. And averaging these B trees reduces the variance.
- Bagging has been demonstrated to give impressive improvements in accuracy by combining together hundreds or even thousands of trees into a single procedure.

PSEUDO CODE

```
Initialize the number of models to build (n_models) and the size of the random subsets of the data (subset_size)
For each model in i to n_models:
    Randomly select subset_size samples from the training data
    Train a regression model on the selected subset of data
    Store the trained model
For each sample in the test data:
    Predict the target value using each of the n_models
    Average the predictions of the n_models to get the final prediction
Return the final predictions
```

BOOSTING

- Boosting works in a similar way as bagging, except that the trees are grown sequentially: each tree is grown using information from previously grown trees.
- Boosting does not involve bootstrap sampling; instead each tree is fit on a modified version of the original data set.
- Unlike bagging and random forests, boosting can overfit if B is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select B.
- The number d of splits in each tree, which controls the complexity of the boosted ensemble.
- More generally d is the interaction depth, and controls the interaction order of the boosted model, since d splits can involve depth at most d variables.