



Ensemble Methods and Regressions

Data Boot Camp
Lesson 19.1



Class Objectives

By the end of this lesson, you will be able to:



Calculate and apply bagging and boosting methods to create and use ensemble algorithms.



Apply regularization parameters for regressions and select the appropriate parameters for a specified problem.



Use Random Forests and LASSO regressions to aid the feature selection process.

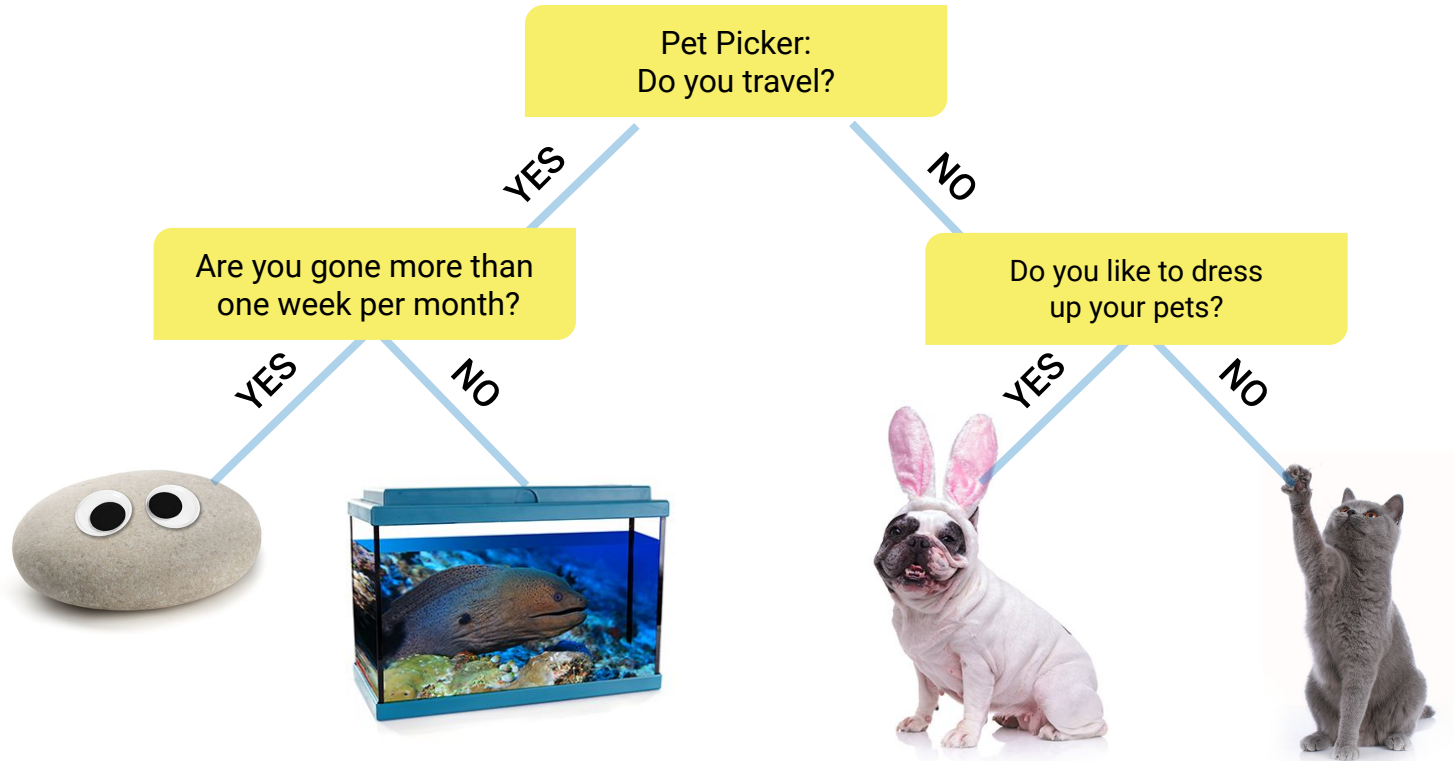


Instructor Demonstration

Decision Trees

Decision trees encode a series of true/false questions

Instructor Do: Decision Trees



Instructor Do: Decision Trees

These true/false questions can be represented with a series of if/else statements



Do you travel?

Yes Travel:



Are you gone for more than one week per month?

Yes: Pet Rock

No: Pet Fish

No Travel:

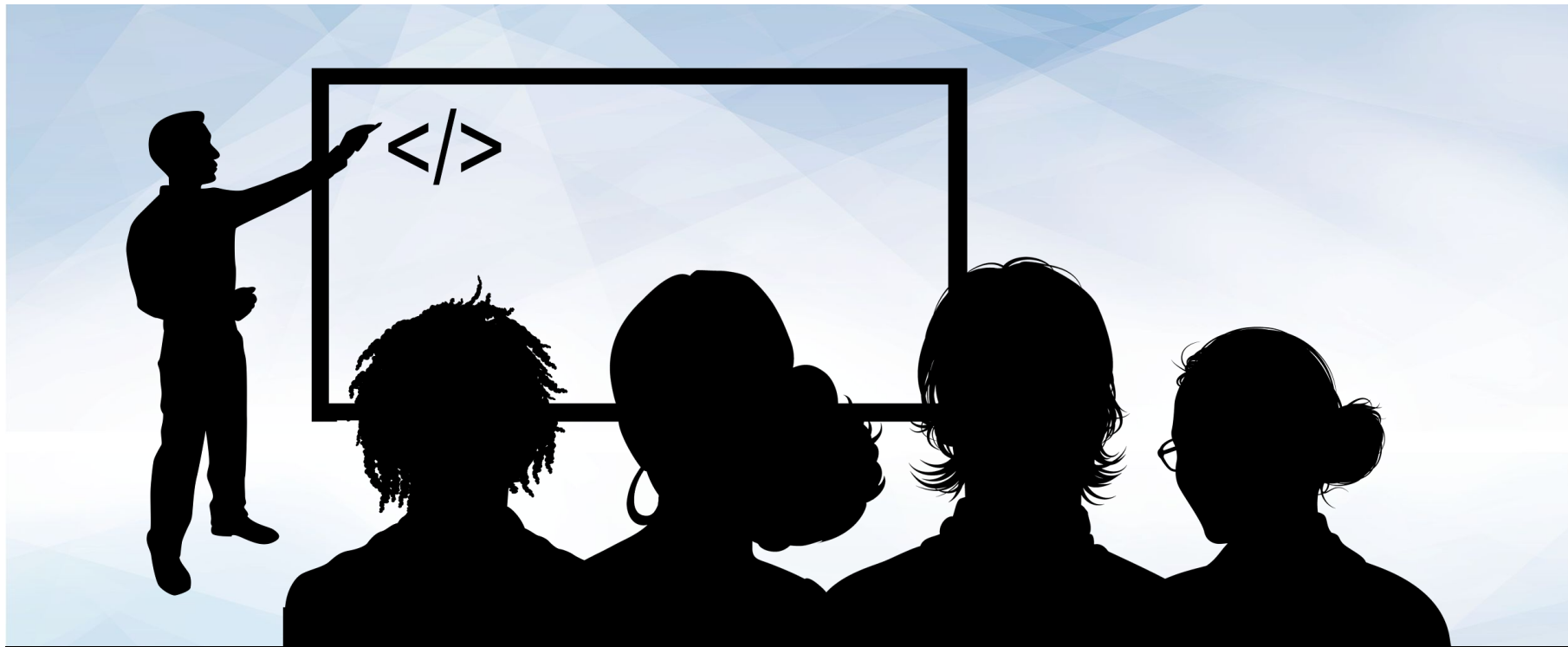


Do you like to dress up your pet?

Yes Dress Up: Pet Dog

No Dress Up: Pet Cat

```
if (travel):  
    if (time > week):  
        print("Rock")  
    else:  
        print("Fish")  
else:  
    if (dress_up):  
        print("Dog")  
    else:  
        print("Cat")
```



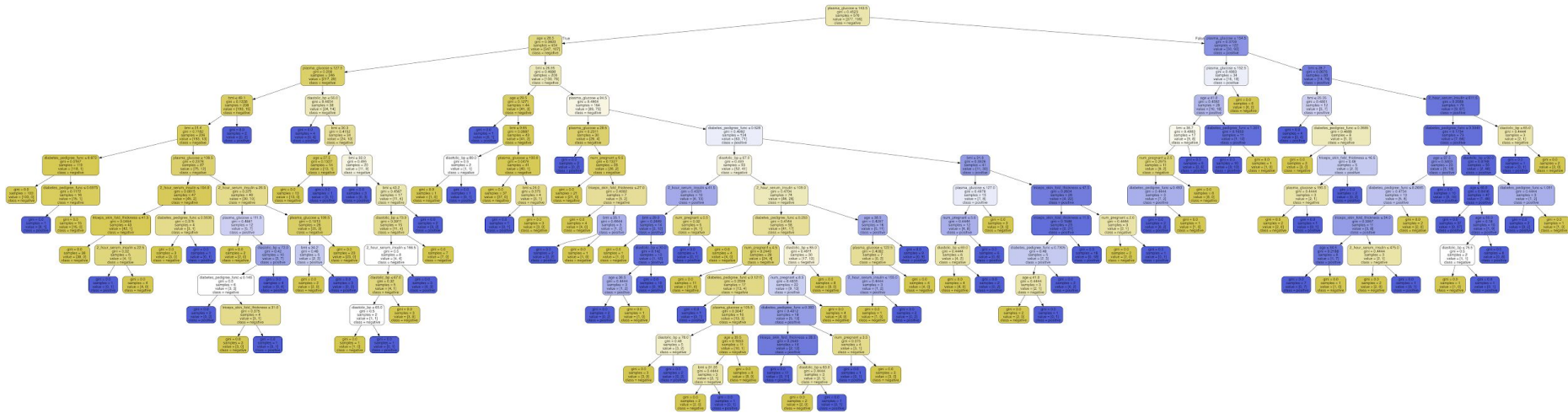
Instructor Demonstration

Ensemble Methods

Decision trees Complexity

Instructor Do: Ensemble Methods

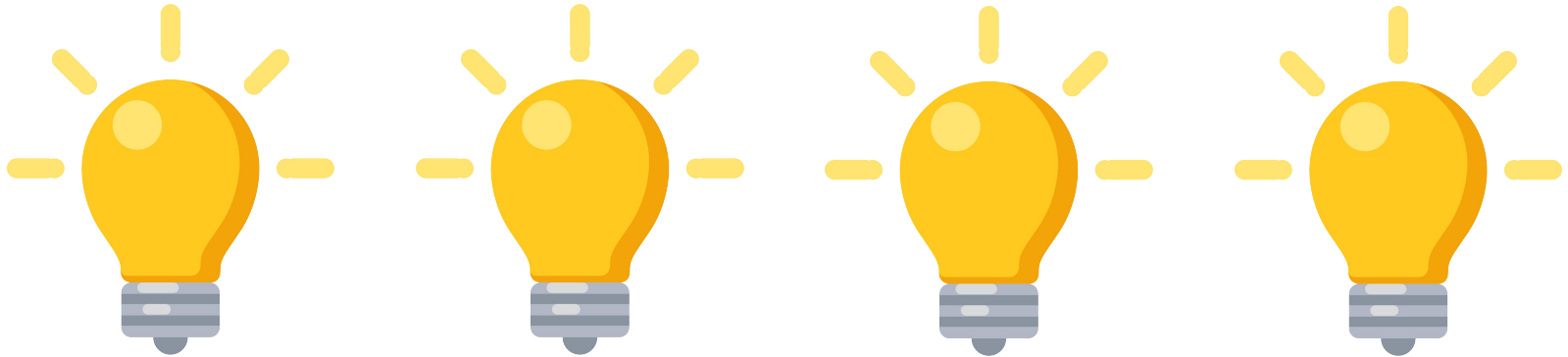
- Decision trees can become very complex and may not generalize well.



Aggregation

Instructor Do: Ensemble Methods

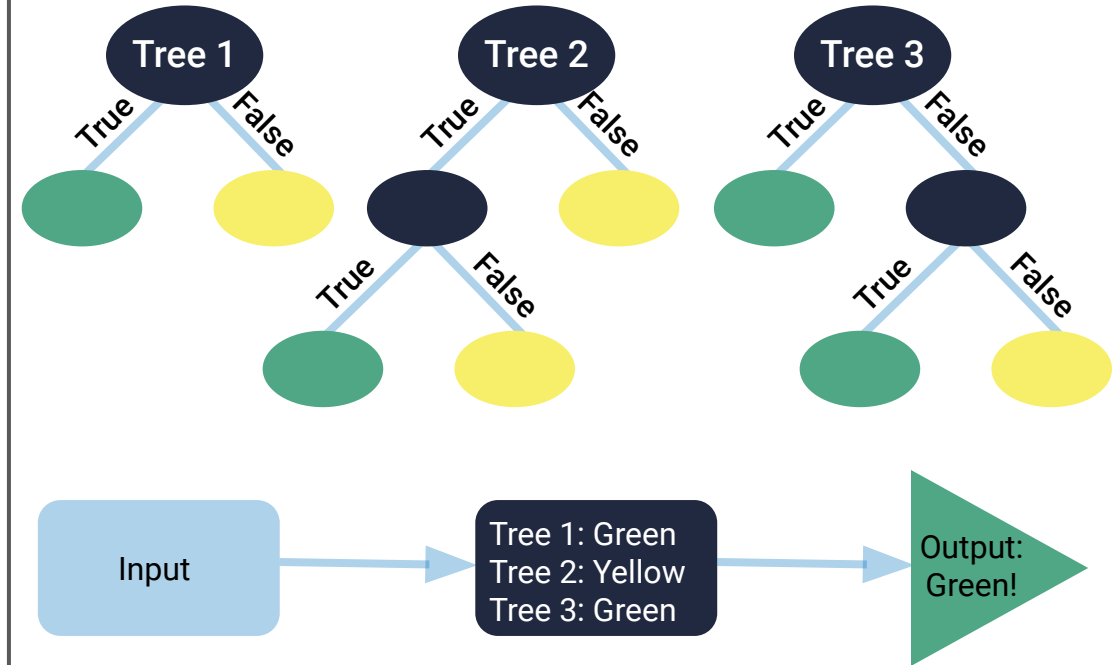
- Instead of one complicated algorithm, make a lot of random simple algorithms, and get their consensus.



Random Forests

Instructor Do: Ensemble Methods

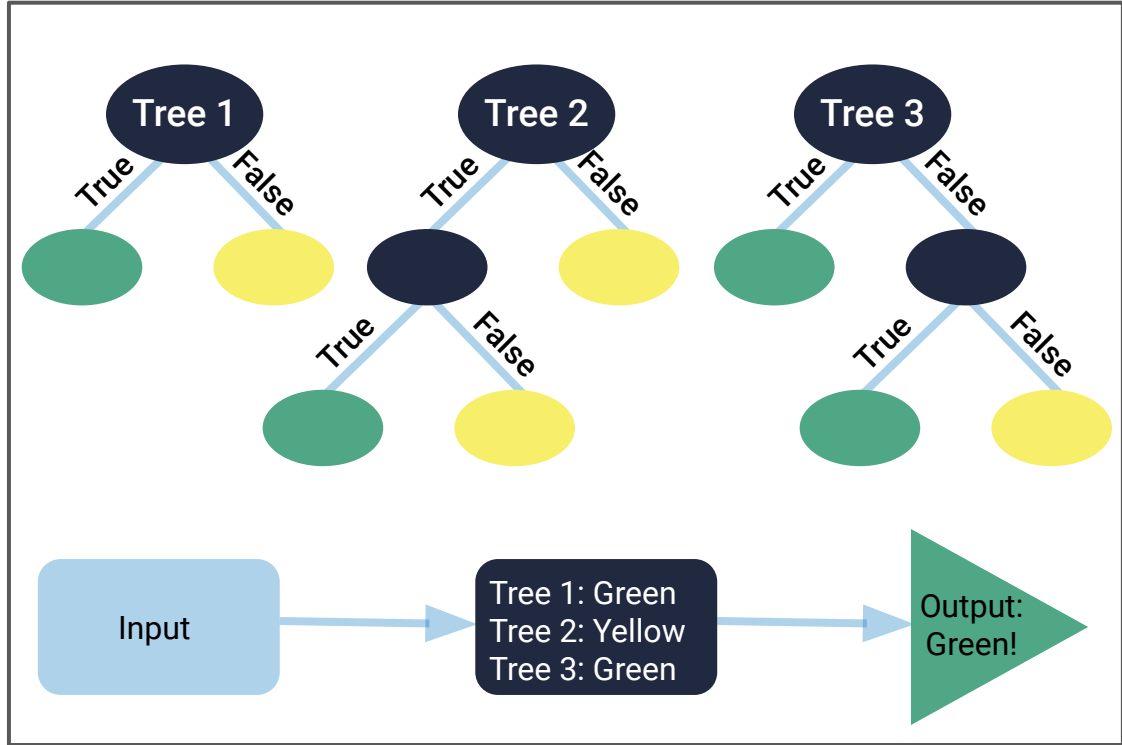
- Instead of a single, complex tree, a random forest algorithm will sample the data and build several smaller, simpler decisions trees (i.e., a forest of trees).
- Each tree is much simpler because it is built from a subset of the data.
- Each tree is considered a “weak classifier” but when you combine them, they form a “strong classifier.”



Extremely Random Trees

Instructor Do: Ensemble Methods

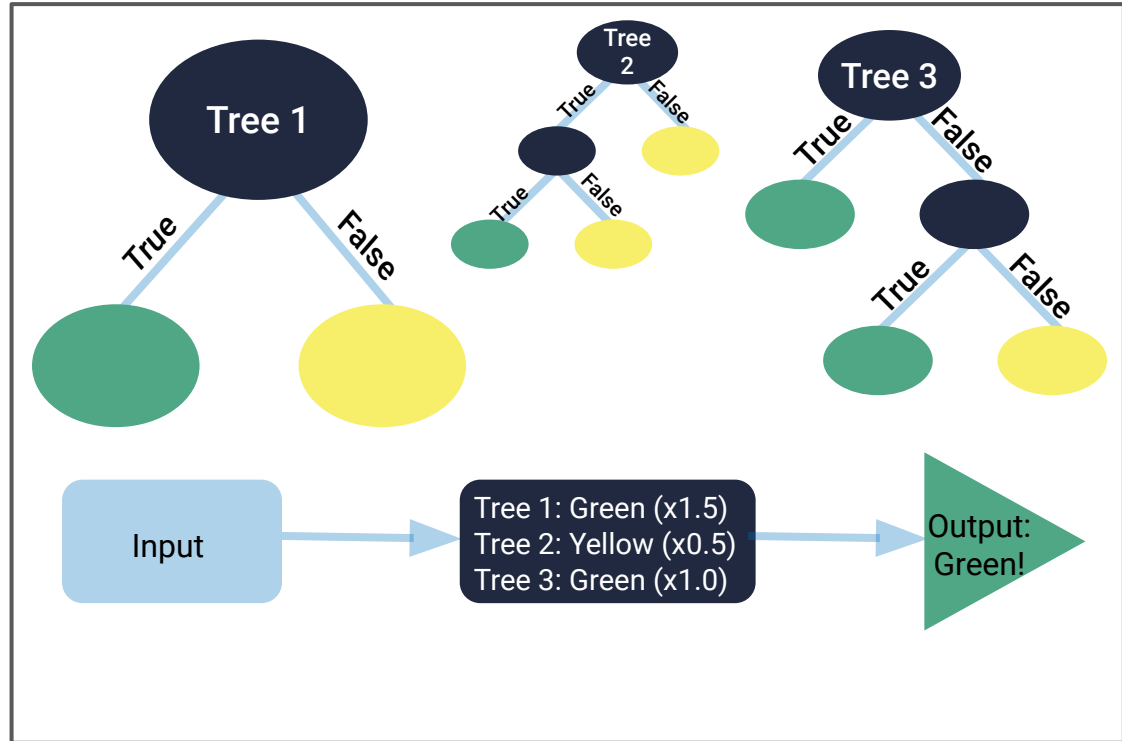
- Extremely Random Trees (or ExtraTrees) is a similar method to Random Forests, but where Random Forests resample the data for each new decision tree, Extremely Random Trees reuse the entire sample each time.
- The decision trees for Random Forests also choose the optimal split point for each decision, while Extremely Random Trees pick at random. This makes Extremely Random Trees a much faster algorithm, but Extremely Random Trees increase bias (and decrease variance).



Boosting

Instructor Do: Ensemble Methods

- Boosting is an aggregation method where individual weak learners get “boosted”, or amplified if they are contributing to correct solutions, and diminished if they are not. This is a very effective technique in terms of accuracy, however, it takes much longer to fit. The increase in accuracy is often worth the extra time.



Questions?





Activity: Bag and Boost

In this activity, you will apply three aggregate classifiers to predict diabetes from the Pima Diabetes dataset.

Suggested Time:
15 Minutes



Instructions:

Activity: Bag and Boost

- Import a Random Forests classifier, and then fit the model to the data.
- Import an Extremely Random Trees classifier, and then fit the model to the data.

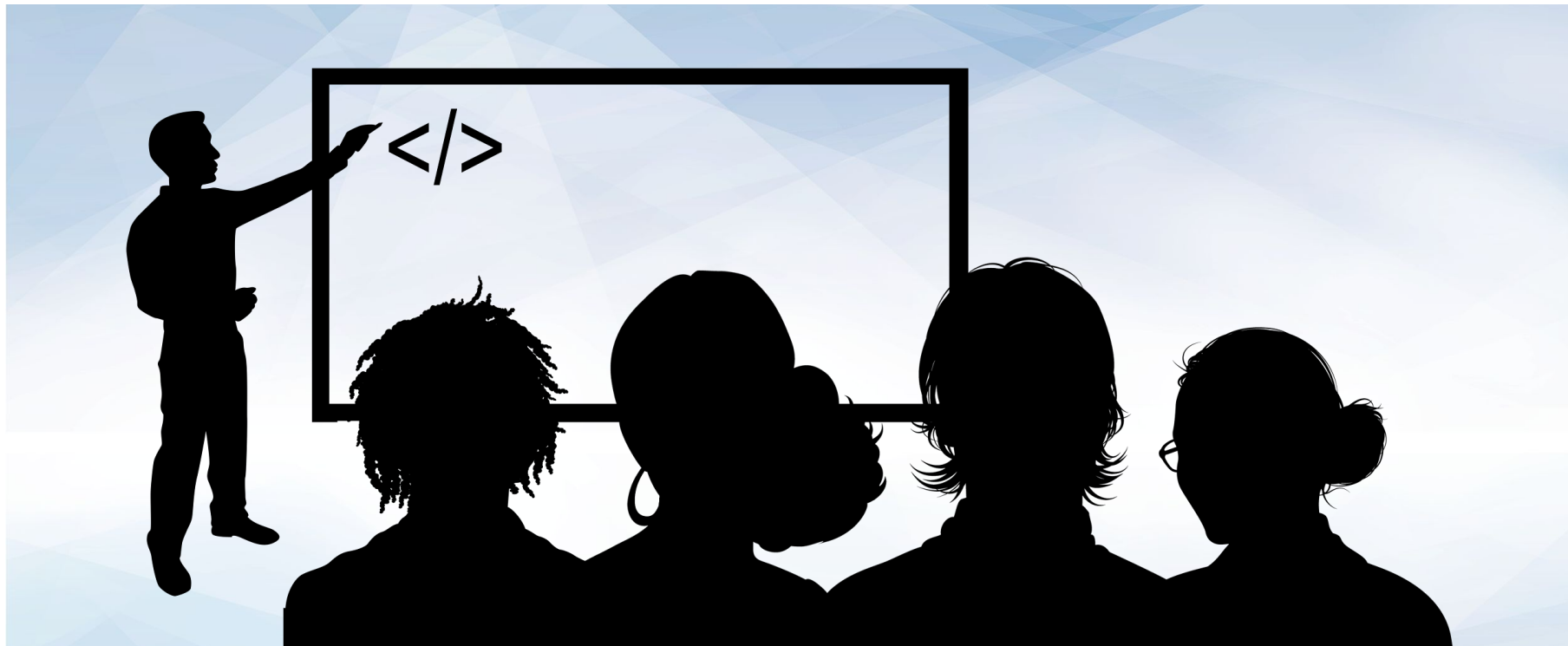
- **Bonus:**

- Refactor to reduce repetitive code. Create a function that takes in a model and a dataset and prints a classification report to compare different models.
- Choose one of the models, and then read the scikit-learn documentation. Use your newly created function to try different parameters. Can you improve the model?

- Import an Adaptive Boosting classifier, and then fit the model to the data.
- Calculate the classification report for each model. Also, calculate the score for both the training and the testing set. Compare the performance of the three models.



Let's Review



Instructor Demonstration

Feature Selection with Random Forests

Feature Selection

Instructor Do: Feature Selection with Random Forests

- Machine Learning models can be “confused” by an overabundance of features, fitting to the noise of irrelevant features.
- Feature Selection is a process of selecting a subset of relevant features, reducing the width of our dataset.
- There are many reasons to perform feature selection.
 - Simplified models are less likely to overfit,
 - Simplified models reduce training time,
 - Simplified models are easier to interpret, etc.
- There are many ways to perform feature selection. One technique uses the information from a Random Forest model.

Feature Selection with Random Forests

Instructor Do: Feature Selection with Random Forests

- Remember, Random Forests use decision trees that try to select the best feature at every split.
- Therefore, how often a feature gets selected over the whole Random Forest model gives us an indication of how important that feature is.
- Feature importances are accessible after fitting a RandomForestClassifier in scikit-learn with the `feature_importance`



Activity: Finding the Features From the Trees

In this activity, you will apply three aggregate classifiers to predict diabetes from the Pima Diabetes dataset.

Suggested Time:
15 Minutes



Instructions:

Activity: Finding the Features from the Trees

1. Import the arrhythmia data, and then fit a Random Forests model to the scaled and split data.
2. Import SelectModel to extract the best features from the Random Forests model.
3. Fit a logistic regression to the original dataset, and then print its score.
4. Fit a logistic regression to the selected dataset, and then print its score.
5. Compare the scores of the two logistic regression models.



Let's Review



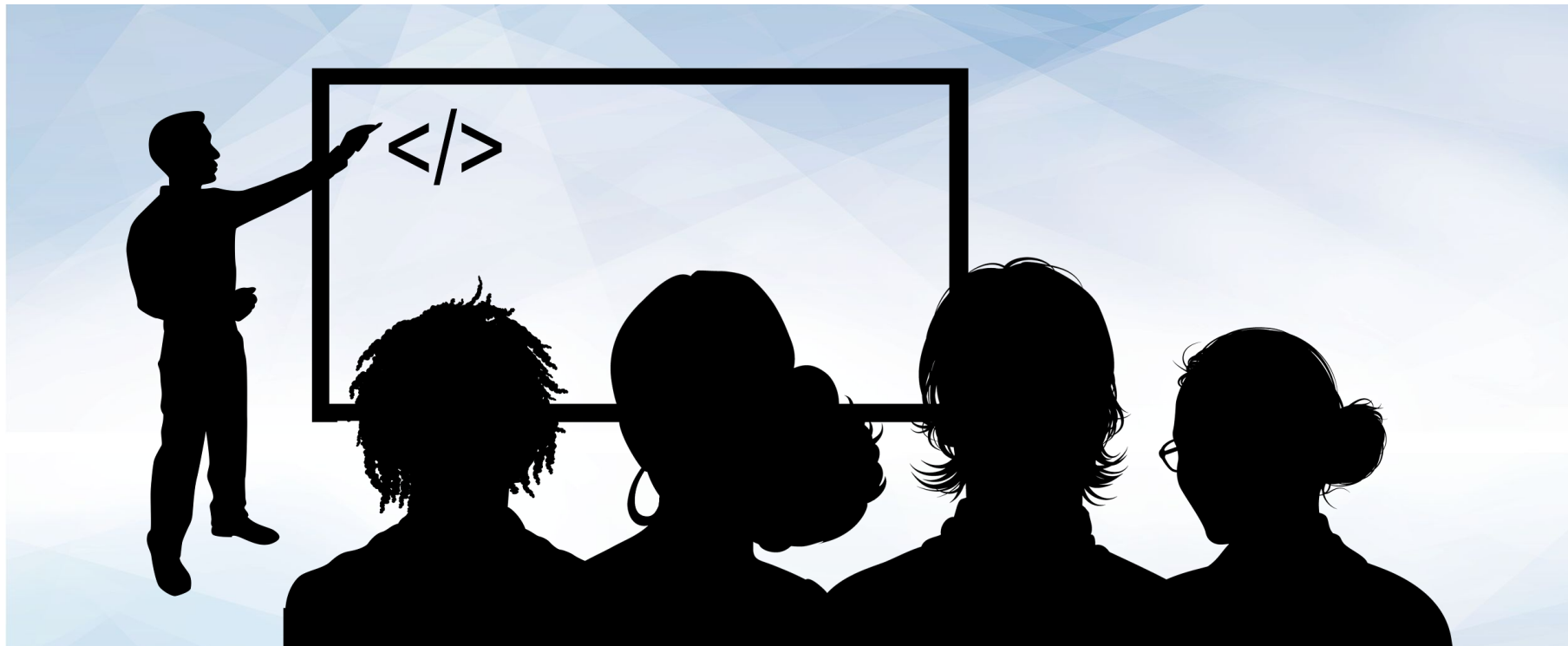
Countdown timer

15:00

(with alarm)



Break



Instructor Demonstration

Regression

Regressions

Instructor Do: Regressions

The Regressions we know...

- Regressions are models that output continuous values.

- We've already seen Linear Regression before, and it is still the basis of the first regression models we are going to look at.

Regularization

- Ridge regression
- Lasso regression
- ElasticNet regression

- Ridge, Lasso, and ElasticNet all add “regularization” terms to a regular linear regression model.
 - The regularization terms at the size of the coefficients of the model.
 - Adding the regularization terms to the model helps keep the coefficients of the model more consistent.



Activity: Regularized Regression

In this activity, you will use regularized models to predict the housing prices of buildings in Tehran.

Suggested Time:
15 Minutes



Instructions:

Activity: Finding the Features from the Trees

→ For each of the following four regression models, import the model from scikit-learn, fit the model to the data, and then print the model's score:

- Linear Regression
- LASSO
- Ridge
- ElasticNet



Let's Review



Everyone Do: Feature Selection with LASSO

In this activity, we will select features from the LASSO model to improve the linear regression from a weak model.

Suggested Time:
10 Minutes



<Time to Code>





Activity: Feature Selection with LASSO

In this activity, you will use regularized models to predict the housing prices of buildings in Tehran.

Suggested Time:
15 Minutes



Instructions:

Activity: Feature Selection with LASSO

1. Fit a linear regression to the imported and split dataset. Print the score of the model.
2. Fit a LASSO regression model to the imported and split dataset.
3. Import SelectModel to extract the best features from the LASSO model.
4. Fit a new linear regression to the data with only the selected features (by using SelectModel). Print the score of the model.
5. Compare the scores of the two linear regression models.



Let's Review



Instructor Demonstration

Familiar Regressors

<Time to Code>

