

# Introduction to Data Science

## Data Science Essentials

---



# Goals for today

- Review last session coding tasks
- Machine Learning, Part 2 – Tree-Based Models



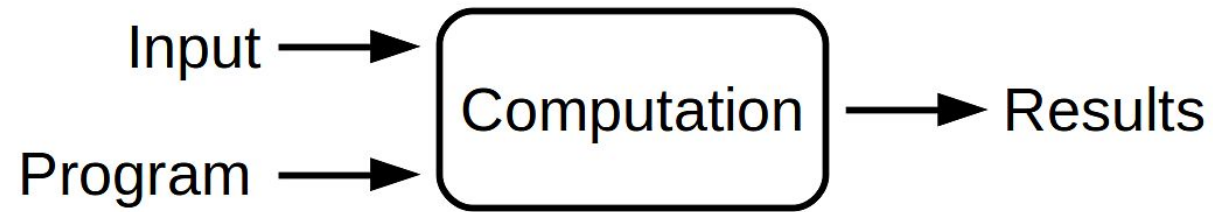
# Review last session coding tasks

**week5\_review** notebook

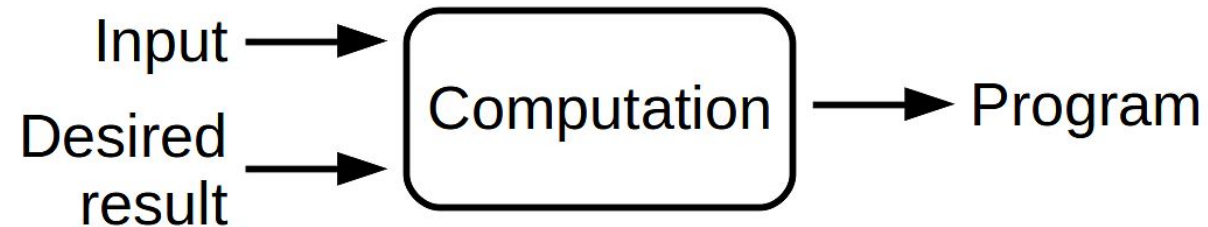


# Supervised Learning

## Traditional programming



## Machine learning



# Recall: Supervised Learning

**Logistic Regression:** Uses a particular functional form to make predictions. (Predict probabilities using a linear combination of the predictor variables.)



## Recall: Supervised Learning

**Logistic Regression:** Uses a particular functional form to make predictions. (Predict probabilities using a linear combination of the predictor variables.)

But, this is not the only type of model we can use to make predictions.



## Recall: Supervised Learning

**Logistic Regression:** Uses a particular functional form to make predictions. (Predict probabilities using a linear combination of the predictor variables.)

But, this is not the only type of model we can use to make predictions.

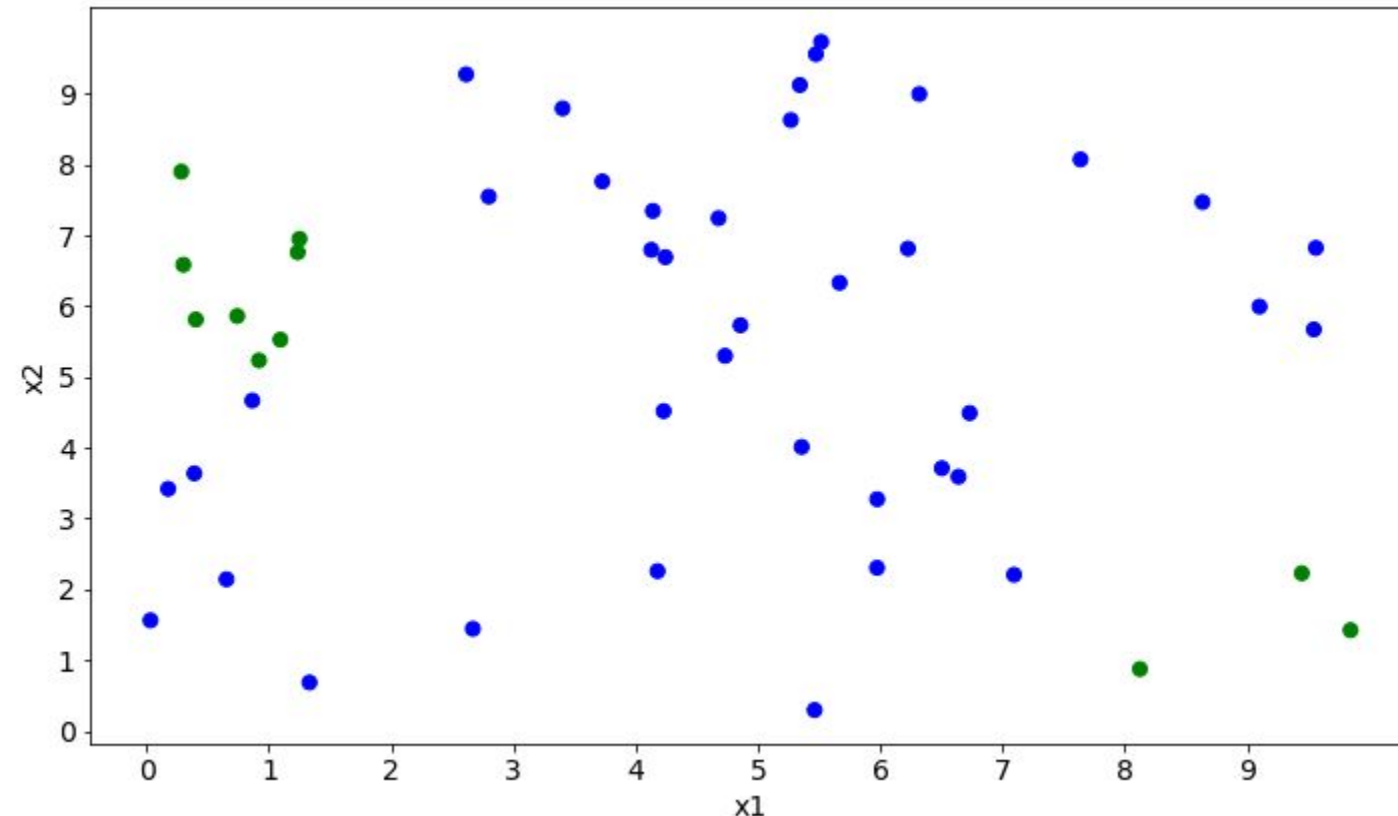
A **decision tree** makes predictions by partitioning the feature space using one feature at a time.



# Decision Trees

Can we build a model that distinguishes blue points from green points?

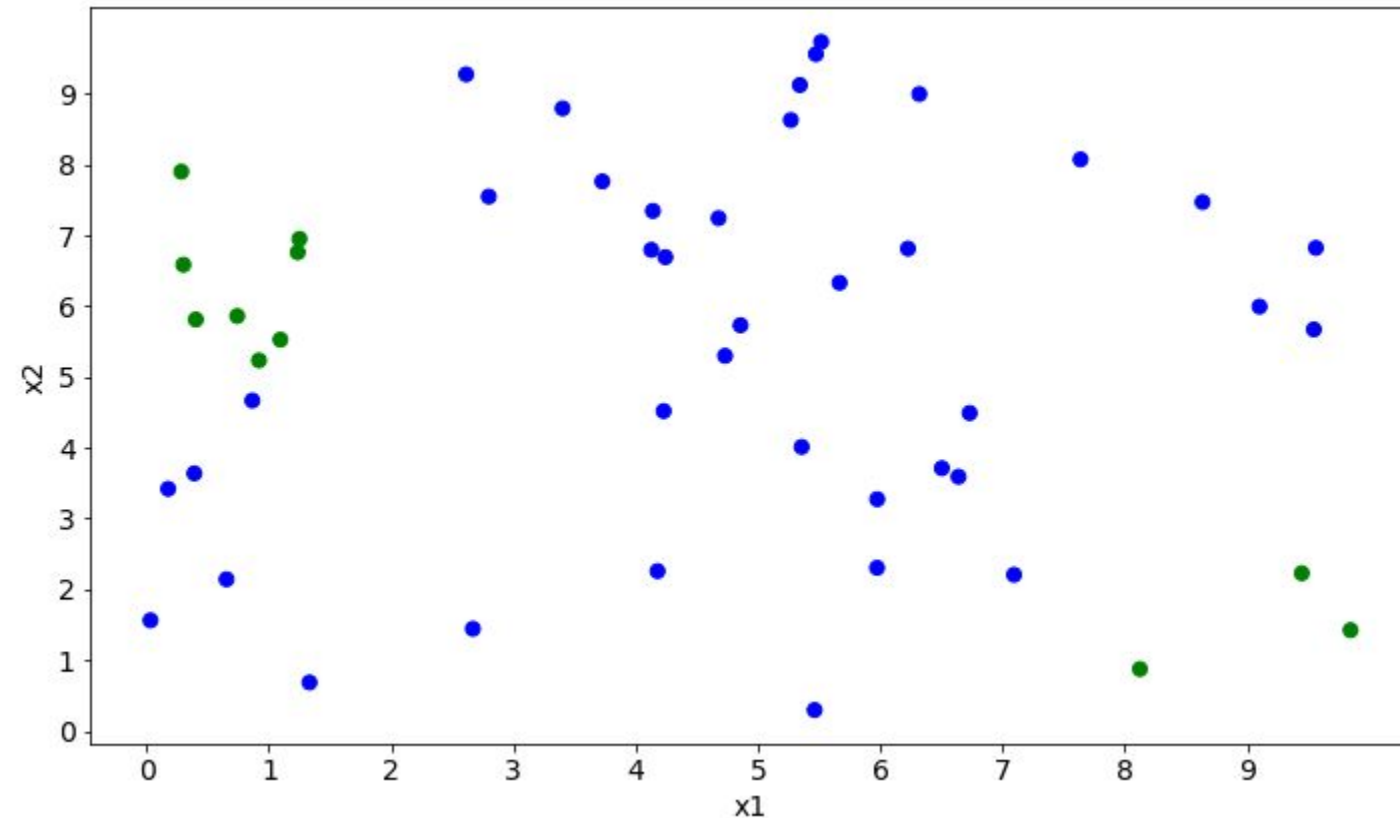
It is impossible to separate these points by a straight line, so a linear method will perform poorly.





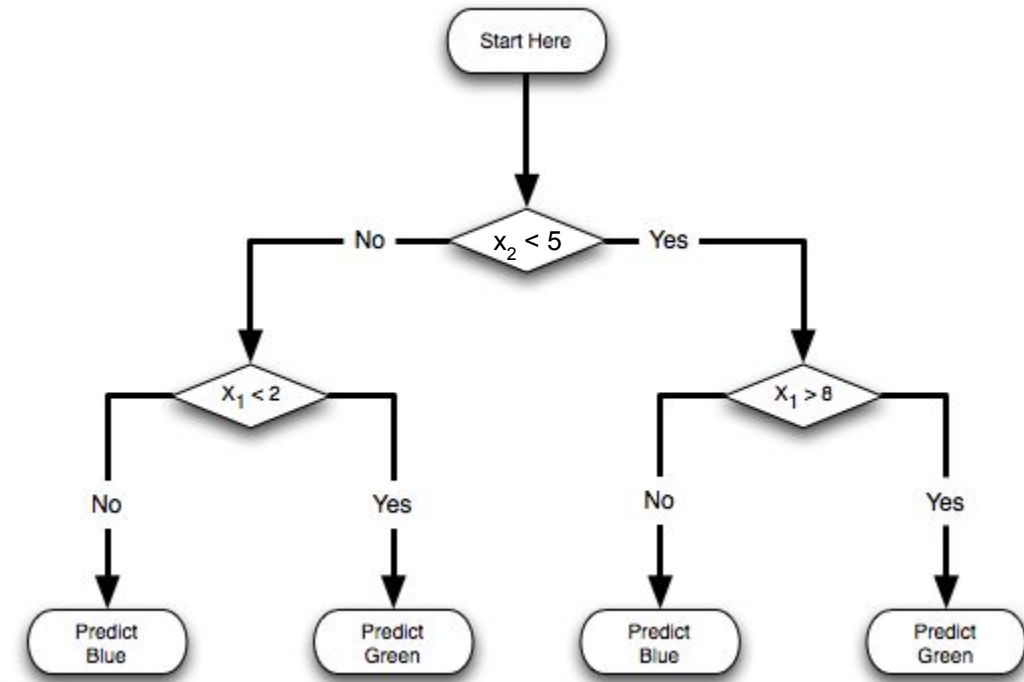
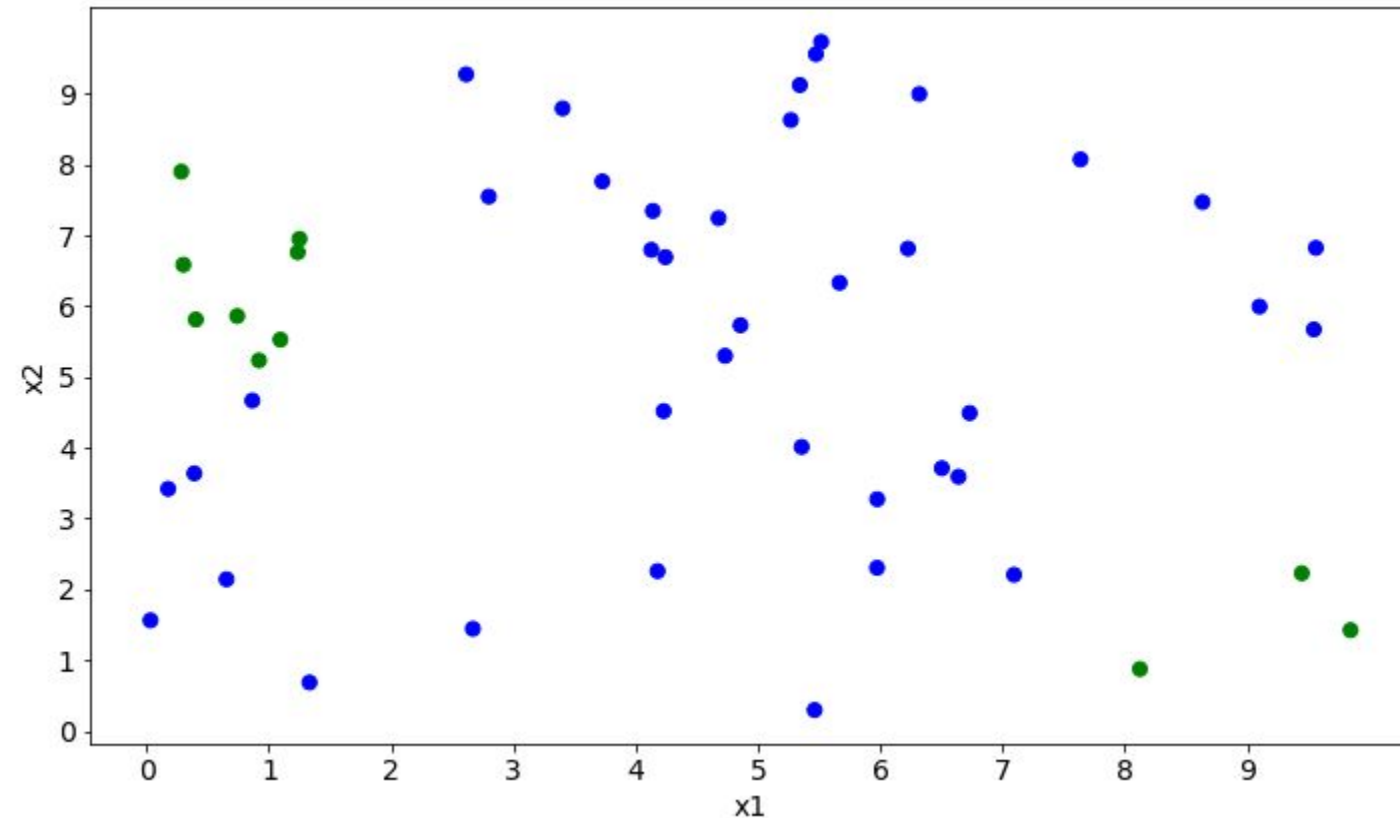
# Decision Trees

Instead, we can use a decision tree approach.



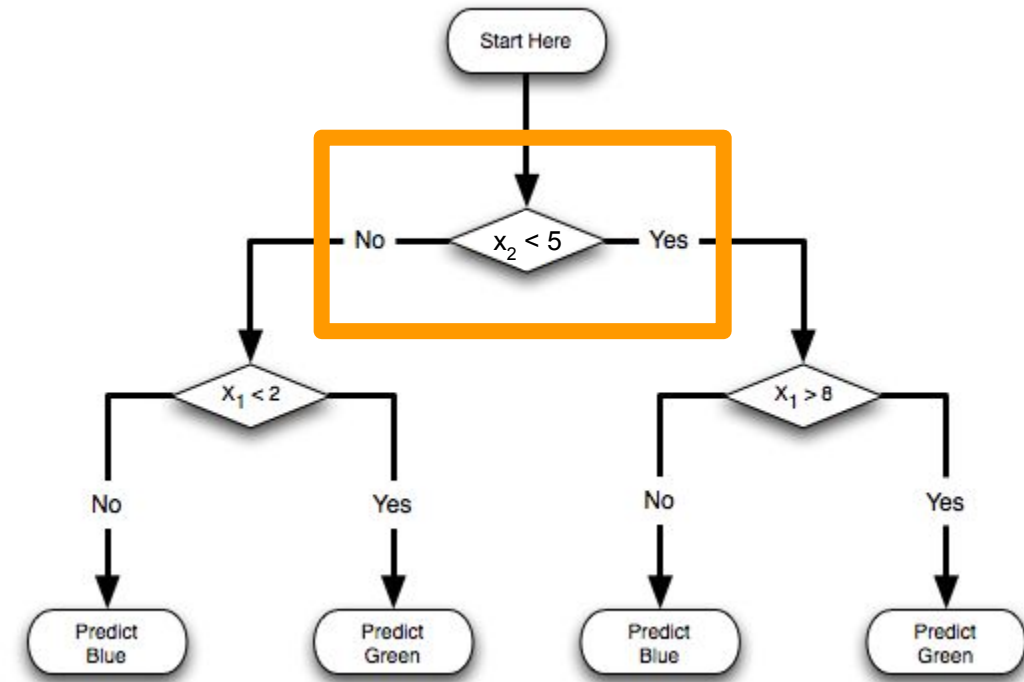
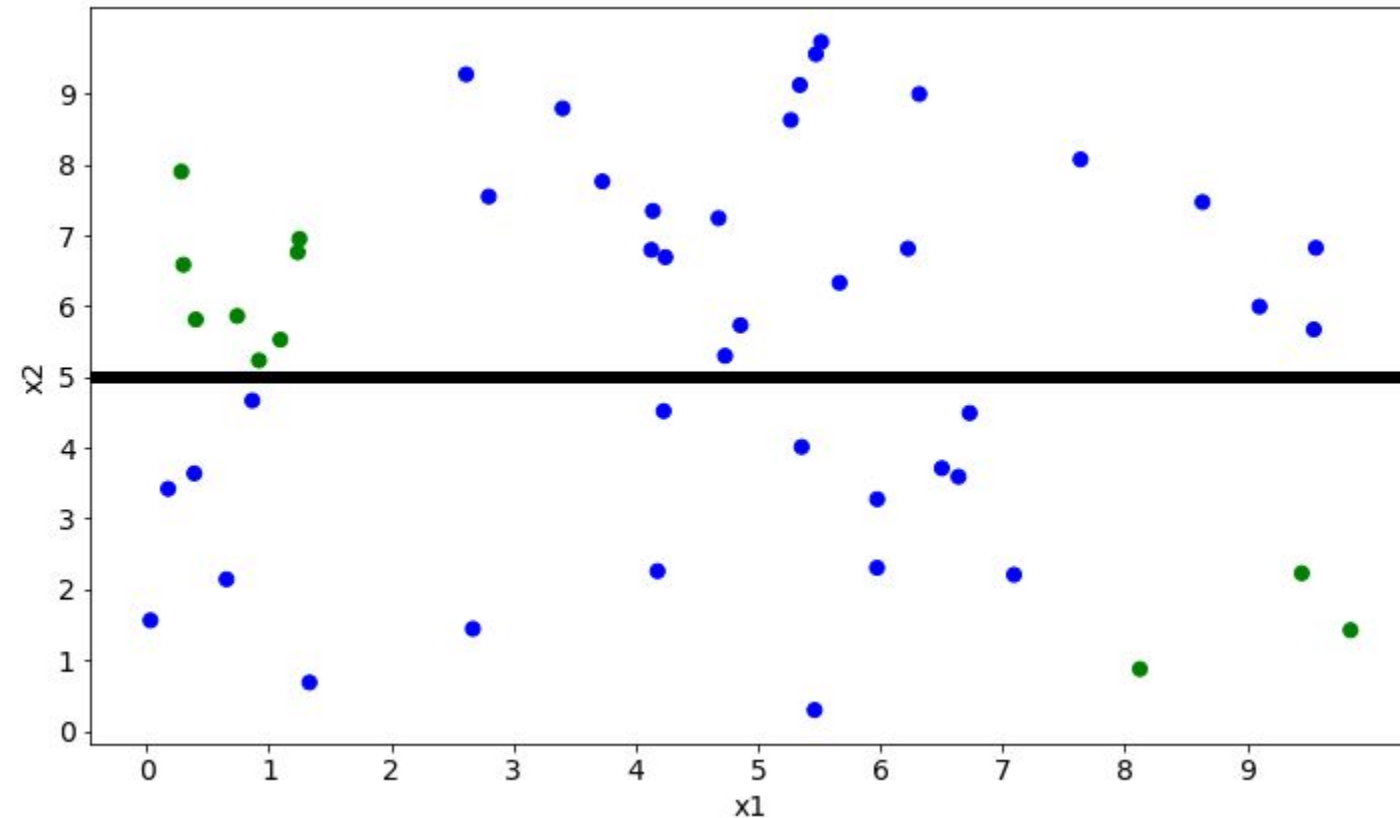
# Decision Trees

Instead, we can use a decision tree approach.



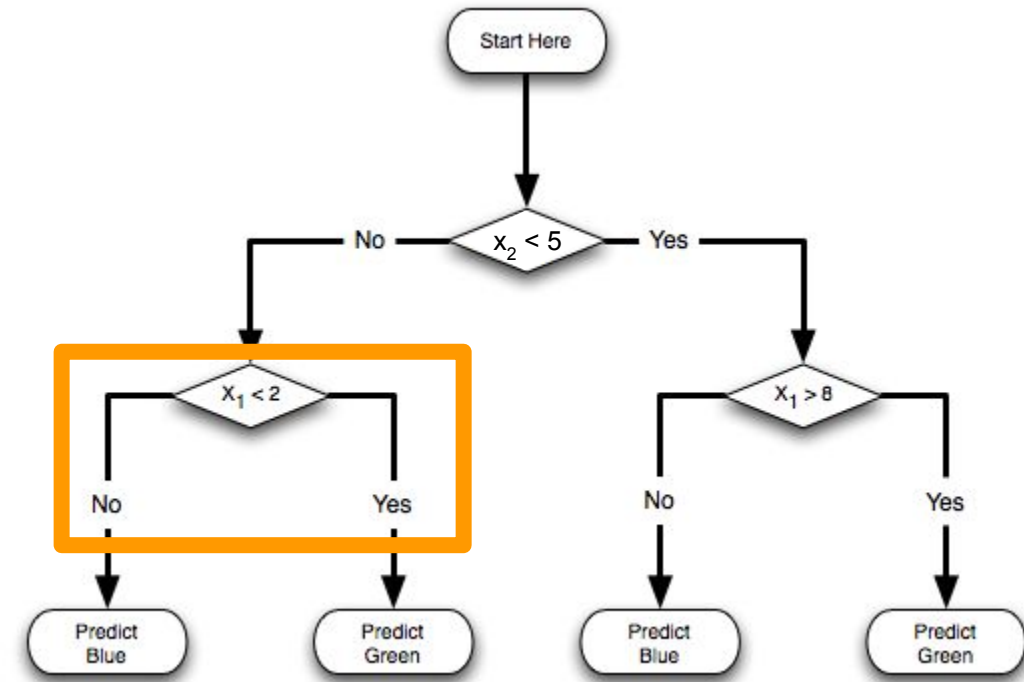
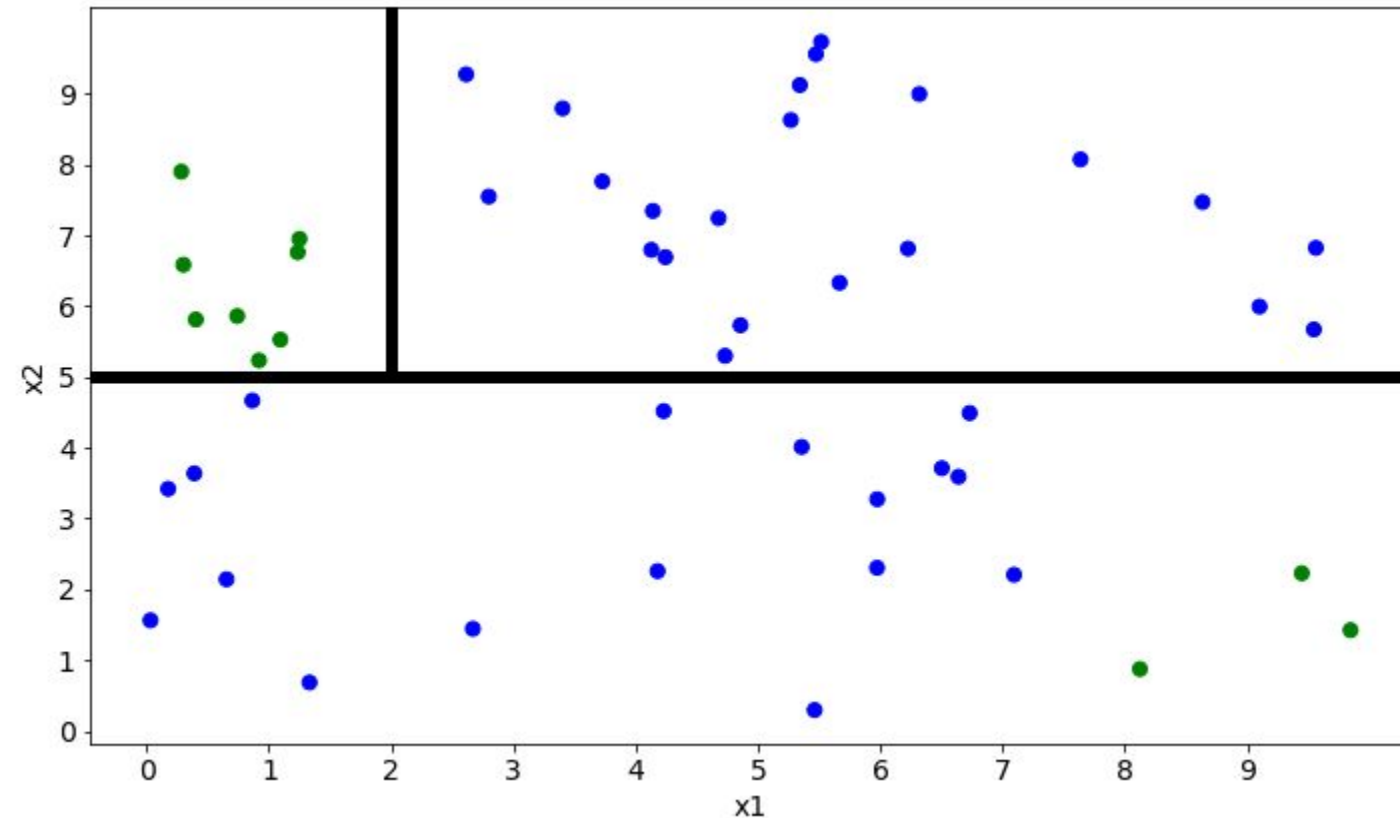
# Decision Trees

Instead, we can use a decision tree approach.



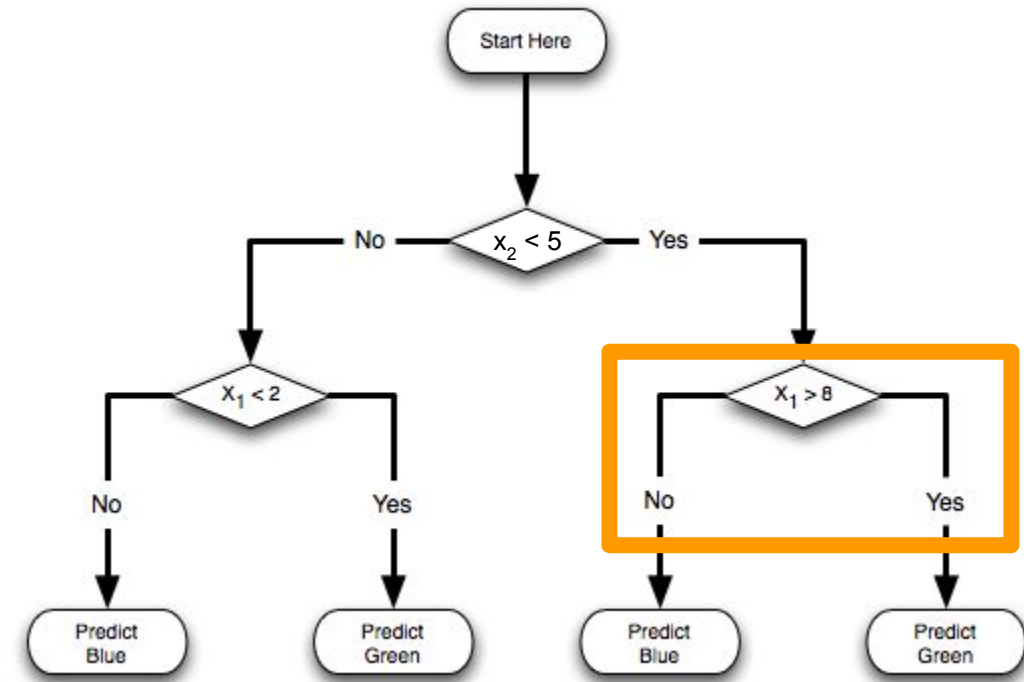
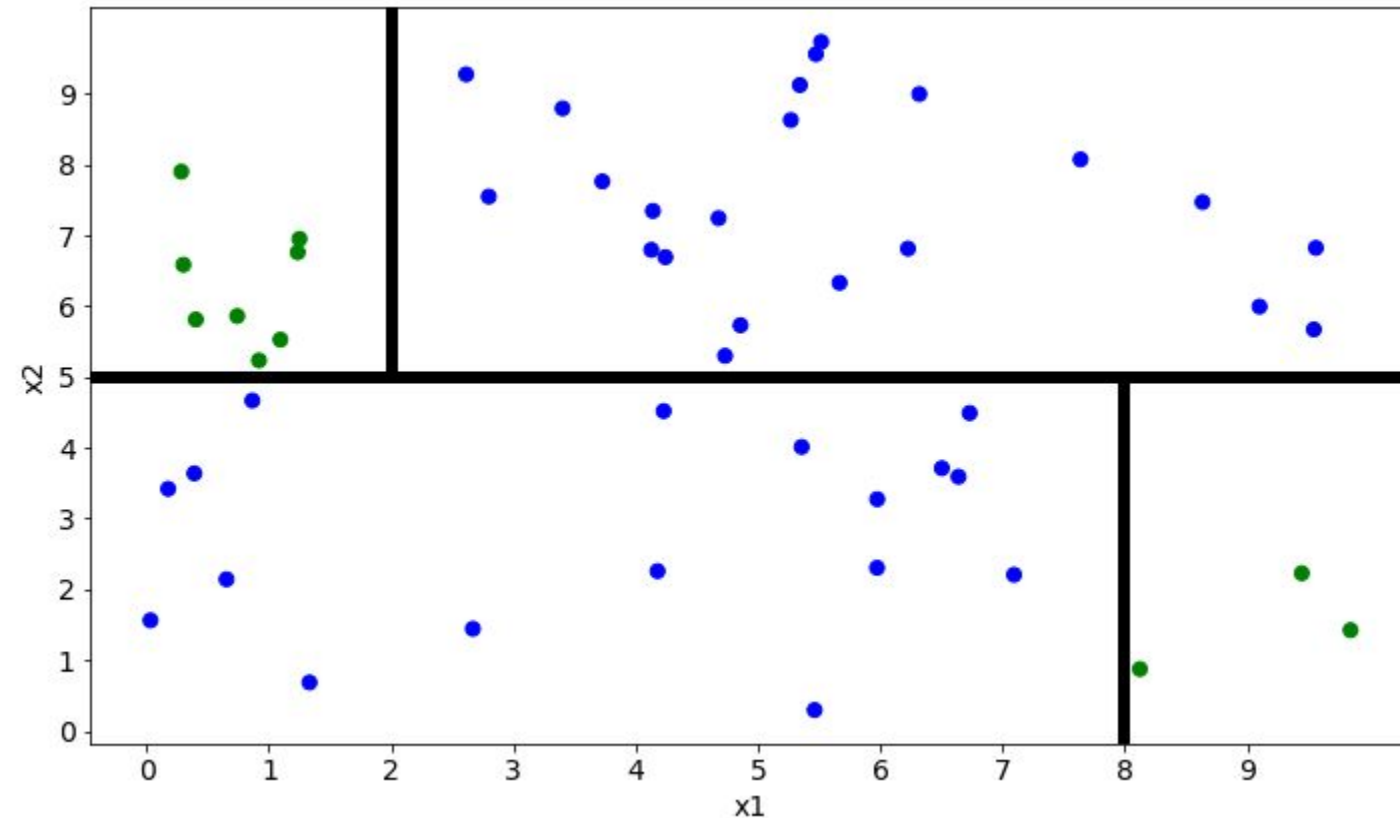
# Decision Trees

Instead, we can use a decision tree approach.



# Decision Trees

Instead, we can use a decision tree approach.

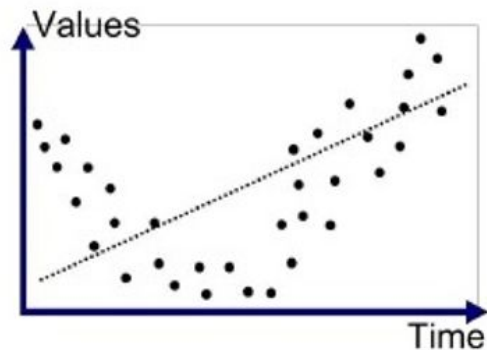


# Underfitting vs. Overfitting

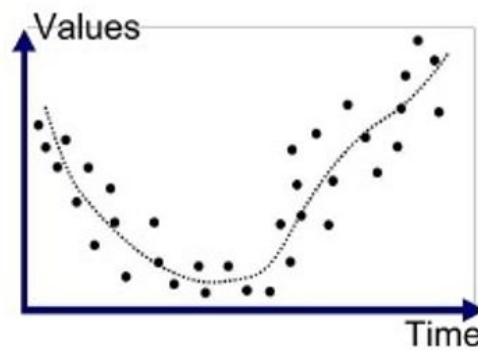
A model is **underfit** when it does not perform well on either the training or test data.

A model is **overfit** when it performs well on training data but does not perform well on unseen data.

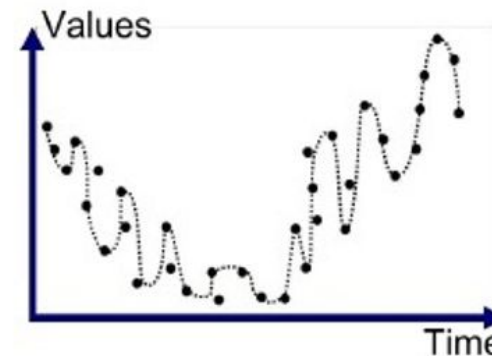
- Can happen when the model “memorizes” the training data or starts fitting to the “noise”
- Often the result of using too flexible of a model.



Underfitted



Good Fit/Robust



Overfitted

# Regularization

One method to mitigate overfitting is **regularization** - forcing your model to be simpler.



# Regularization

One method to mitigate overfitting is **regularization** - forcing your model to be simpler.

For example, **ridge** and **LASSO** regression are variants of linear regression which force the model coefficients to not be too large.

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 \cdot (\text{sepal length}) + \beta_2 \cdot (\text{petal length})$$

Penalize based on  
how large these are





# Random Forests

Single decision trees are extremely flexible and will often not generalize well to new data.

One common way to improve this is by building multiple trees using only subsets of the training data and subsets of predictors at a time and aggregating the predictions (aka creating an **ensemble**).



# Random Forests

Single decision trees are extremely flexible and will often not generalize well to new data.

One common way to improve this is by building multiple trees using only subsets of the training data and subsets of predictors at a time and aggregating the predictions (aka creating an **ensemble**).

**Random Forest** models create a large number of decision trees, each trained on a subset of the training data and a subset of the features in order to decorrelate and reduce the variance of predictions.

To make the final prediction, the predictions from each tree are averaged.



# Questions?

