GENERAL ASSEMBLY

RECAP MODEL EVALUATION

Juan Ginzo

RECAP MODEL EVALUATION

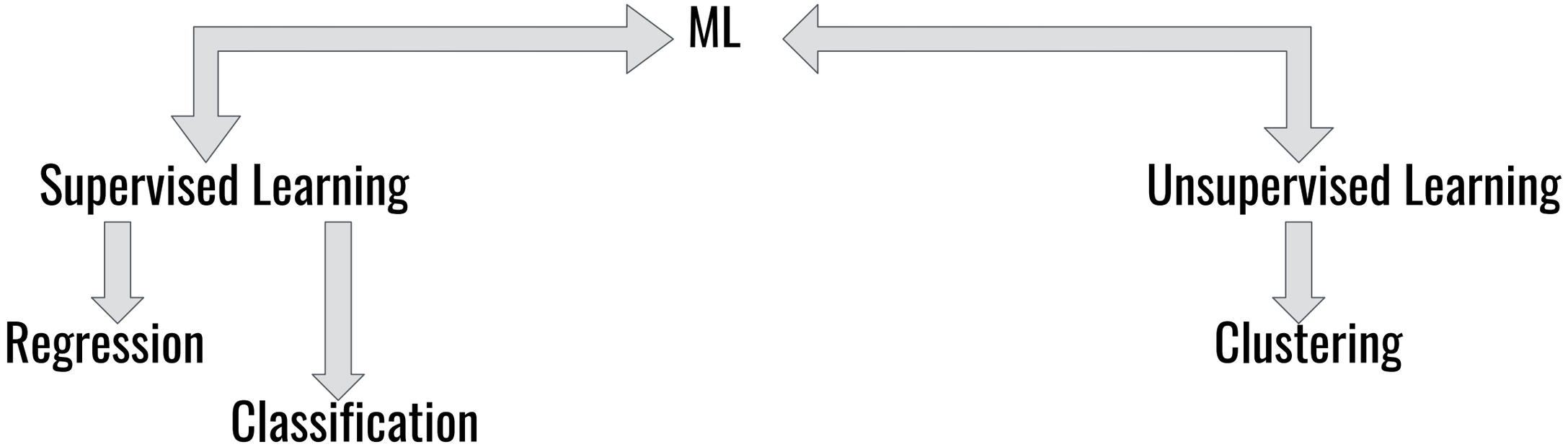
WHAT'S THE AIM HERE?

RECAP MODEL EVALUATION

- What’s the aim of Data Science?
    - It’s building predictive models with data.*

RECAP MODEL EVALUATION

Machine Learning Tree



RECAP MODEL EVALUATION

DS Workflow, the short version:

Get Data -> Initial Dataframe - > EDA -> Feature Matrix -> Modeling -> Assessment

RECAP MODEL EVALUATION

Modeling workflow

Even though we cover many different models, the steps that we need to take from getting from raw data to results, its usually very similar.

Basic Workflow:

1. Do Training/Test split on the dataframe.
2. Create an Instance of the model.
3. Fit the model with the training data.
4. Asses the model we just fitted with the test data.
5. (Optional) Use predict to generate new data.

* For a code example of this workflow, please check the file **Basic_modeling_workflow.ipynb** in the Workflow folder.

RECAP MODEL EVALUATION

Sources of Error

Even the best of models is only an approximation of reality, and at the same time, we only have a subset of the population to train the model on. So we can always expect to have some errors in our predictions.

We can decompose the error from our models into bias and variance:

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

RECAP MODEL EVALUATION

Bias Variance Tradeoff and model complexity.

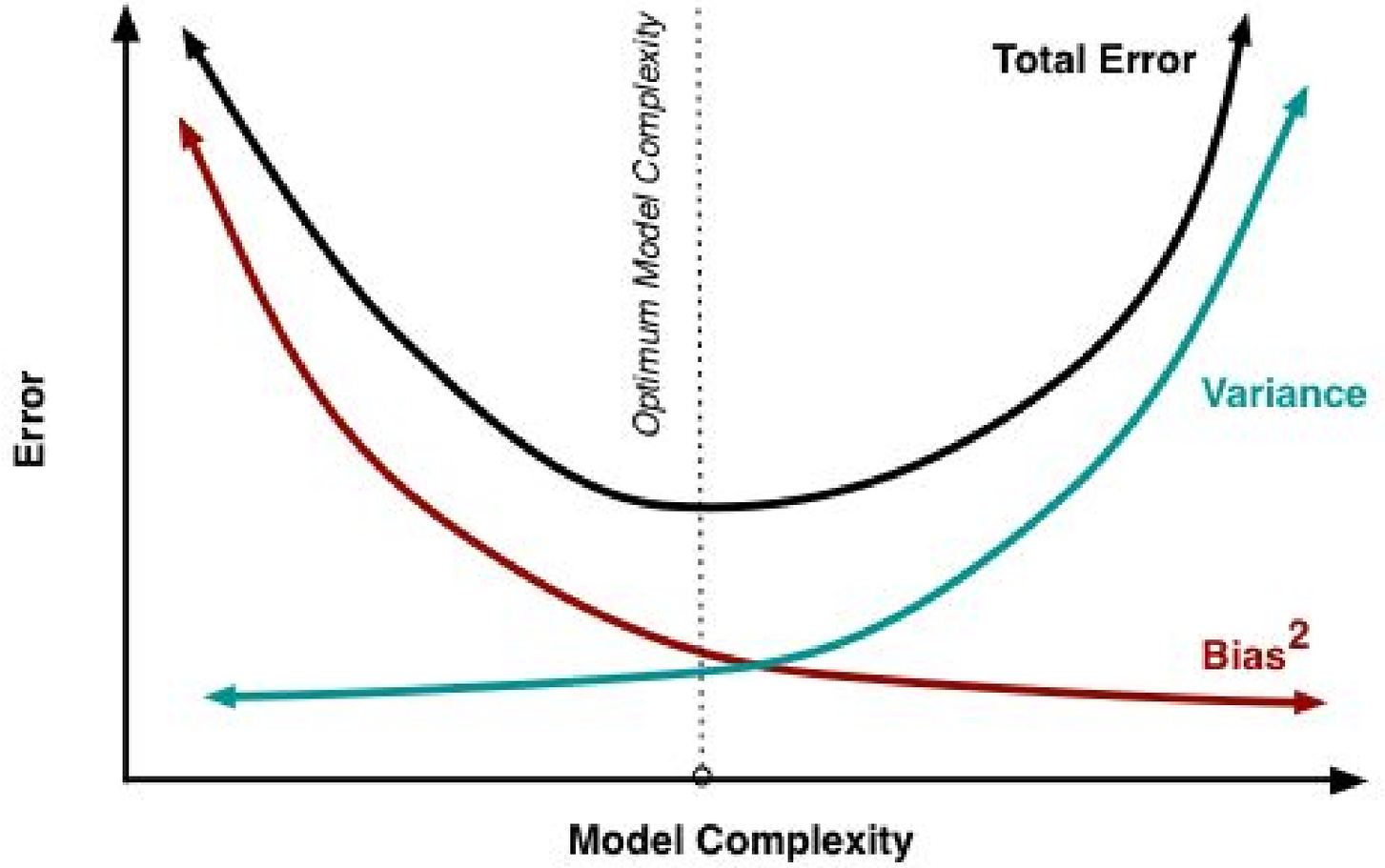
Bias: The error that's introduced by approximating a real-life process, with a simplifying model.

Variance: The error that comes from using a particular sample of data.

Bias–Variance tradeoff is the problem of simultaneously minimizing bias and variance that prevent machine learning models from generalizing beyond their training set (predict unseen data).

RECAP MODEL EVALUATION

Bias Variance Tradeoff and model complexity.



RECAP MODEL EVALUATION

Regularization

So it would be great to be able to decrease the variance of more complex models, even if we had to increase the bias a little bit.

One method to do this is **Regularization**, in which we penalise the size of the coefficients, introducing bias into our model, with the hope of decreasing variance even more (always check though!).

Two main flavours of Regularization are:

- Ridge
- Lasso

RECAP MODEL EVALUATION

Cross validation

If our aim is to predict unseen data with our models, we need a way to check for this.

One approach is to split our data into two sets, a training and a testing set, train on the first one and then predict on the second one, comparing to the actual values of the target.

This has two problems though:

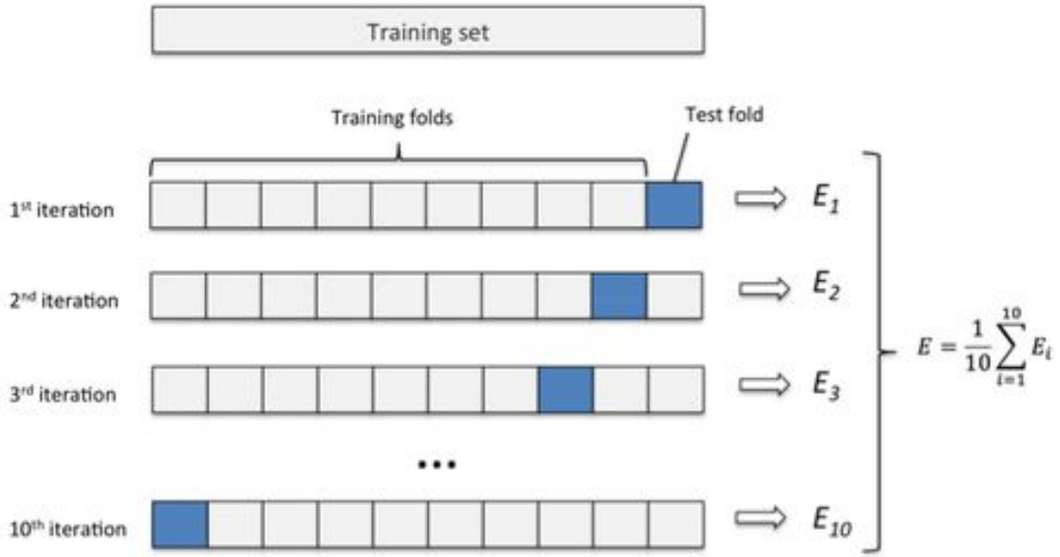
- We are not using all our data to train, so we can be underestimating model performance.
- Doing one split give us a performance metric that has a lot of variability (think what would happen if the split was different).

RECAP MODEL EVALUATION

Cross validation

Given the two problems above, we can use **cross-validation** to assess our model performance.

We will split our data into k different sets, train on k-1 and test on the other. We repeat this K times (so we use all k sets to test). We then average out the results, to getter a better estimate of our model performance.



GENERAL ASSEMBLY

INTRO TO CLASSIFICATION

Juan Ginzo

LEARNING OBJECTIVES

- Define class label and classification
- Build a K-Nearest Neighbors using the sci-kit-learn library
- Evaluate and tune model by using metrics such as classification accuracy/error

OPENING

INTRO TO CLASSIFICATION

INTRO TO CLASSIFICATION

- So far, we've worked primarily with regression problems. We've focused on predicting a continuous set of values.

- That means we've been able to use distance to measure how accurate our prediction is.

- However, for other problems, we need to predict binary responses.
  E.g.: A loan will default or it won't. An email is spam or isn't spam.

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. What if we want to build a model to predict a set of values, like a photo color or the gender of a baby?
2. Can we use regression for binary values?
3. Do the same principles apply?

DELIVERABLE

Answers to the above questions

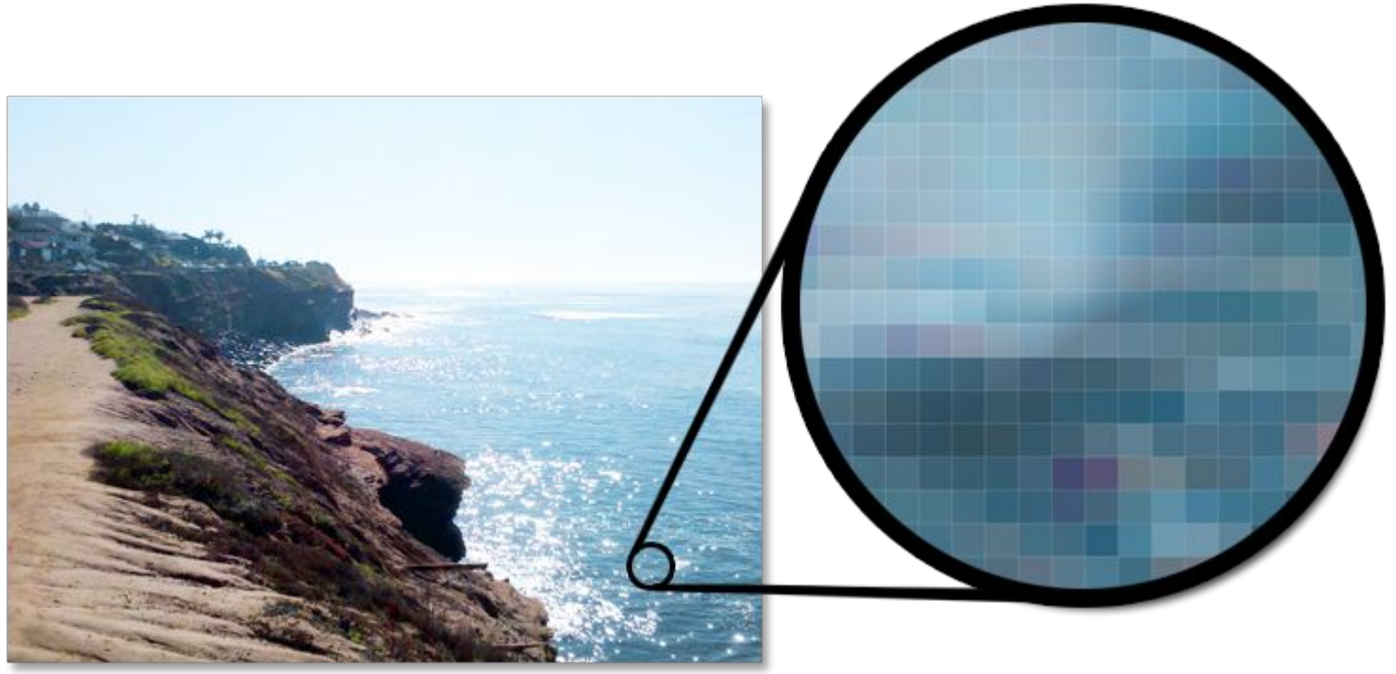
WHAT IS CLASSIFICATION?

WHAT IS CLASSIFICATION?

- **Classification** is the problem of identifying to which set of categories a new observation belongs to.
- **Classification models** in machine learning help us identify those labels, given some features in our data.
- In many classification problems are trying to predict *binary* values or yes/no answers.
- For example, we may be using patient data (medical history) to predict whether the patient is a smoker or not.

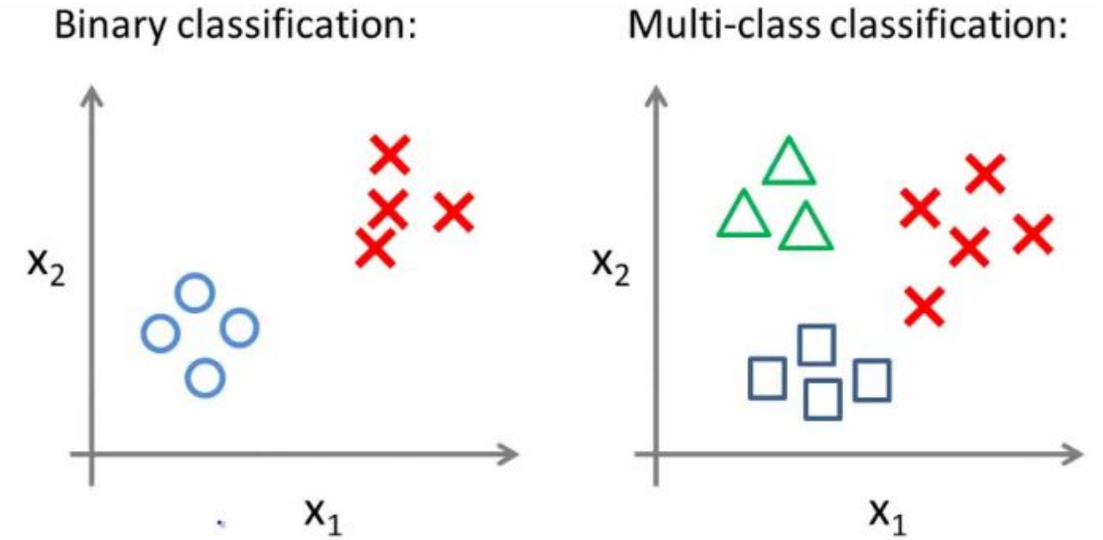
WHAT IS CLASSIFICATION?

- Some problems don't appear to be binary at first glance. However, you can boil down the response to a *boolean* (true/false) value.
- What if you are predicting whether an image pixel will be `red` or `blue`?
- We don't need to predict that a pixel is `blue`, just that it is not `red`.
- This is similar to the concept of dummy variables.



WHAT IS CLASSIFICATION?

- Binary classification is the simplest form of classification.
- However, classification problems can have multiple *class labels*.
- Instead of predicting whether the pixel is red or blue, you could predict whether the pixel is red, blue, or green.



WHAT IS A CLASS LABEL?

- A **class label** is a representation of what we are trying to predict: our *target*.

- Examples of class labels from before are:

Data Problem	Class Labels
Patient data problem	is smoker, is not smoker
pixel color	red, blue, green

DETERMINING REGRESSION OR CLASSIFICATION

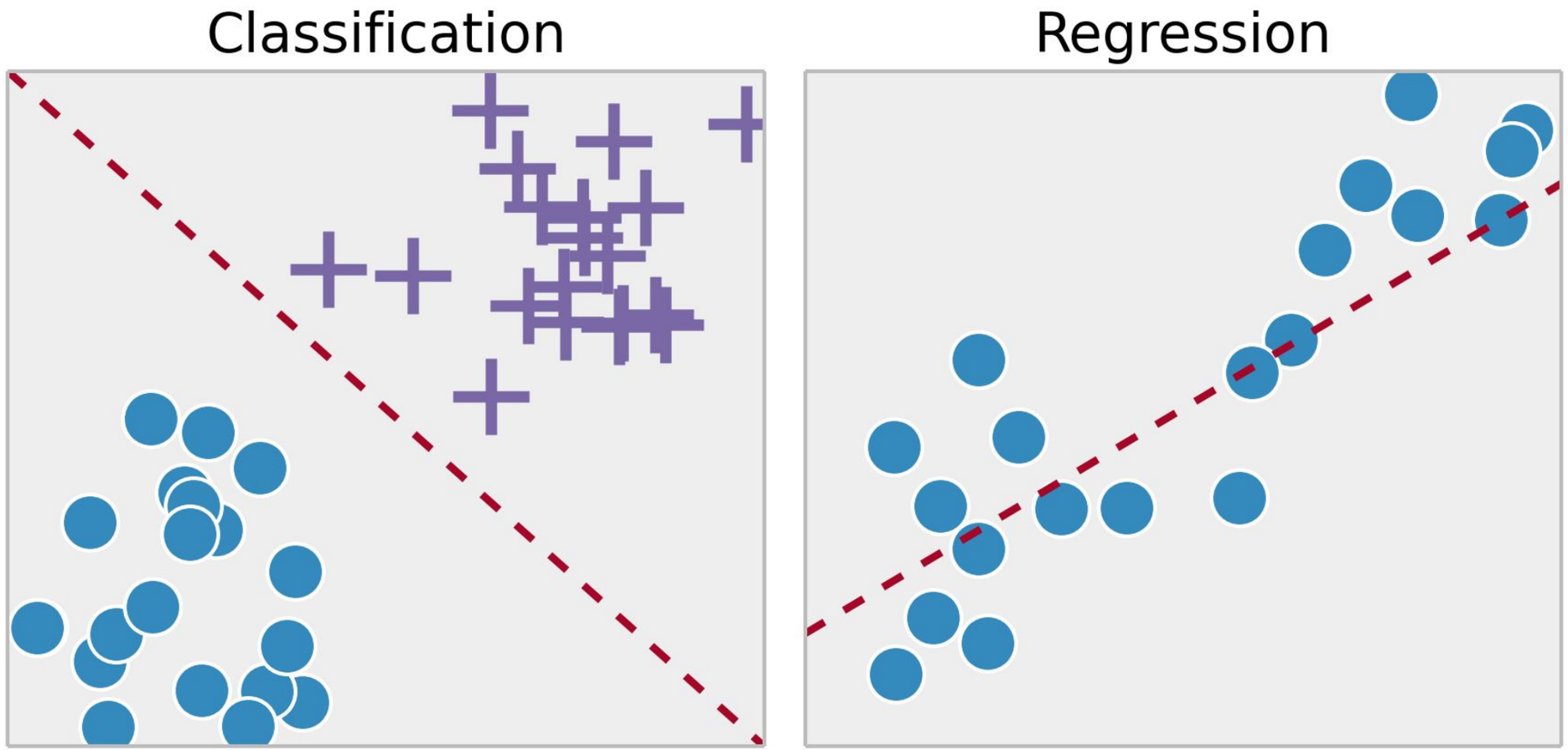
- One of the easiest ways to determine if a problem is regression or classification is to determine if our *target* variable can be ordered mathematically.

- For example, if predicting company revenue, `$100MM` is greater than `$90MM`. This is a *regression* problem because the target can be ordered.

- However, if predicting pixel color, `red` is not inherently greater than `blue`. Therefore, this is a *classification* problem.

DETERMINING REGRESSION OR CLASSIFICATION

- Classification and regression differ in what you are trying to predict.



GUIDED PRACTICE

REGRESSION OR CLASSIFICATION?

ACTIVITY: REGRESSION OR CLASSIFICATION?

EXERCISE

DIRECTIONS (20 minutes)

Review the following situations and decide if each one is a regression problem, classification problem, or neither:

1. Using the total number of explosions in a movie, predict if the movie is by JJ Abrams or Michael Bay.
2. Determine how many tickets will be sold to a concert given who is performing, where, and the date and time.
3. Given the temperature over the last year by day, predict tomorrow's temperature outside.
4. Using data from four cell phone microphones, reduce the noisy sounds so the voice is crystal clear to the receiving phone.
5. With customer data, determine if a user will return or not in the next 7 days to an e-commerce website.

DELIVERABLE

Answers to the above questions and provide a simple 3 line example of what the dataset would look like.

WHAT IS K NEAREST NEIGHBORS?

WHAT IS K NEAREST NEIGHBORS?

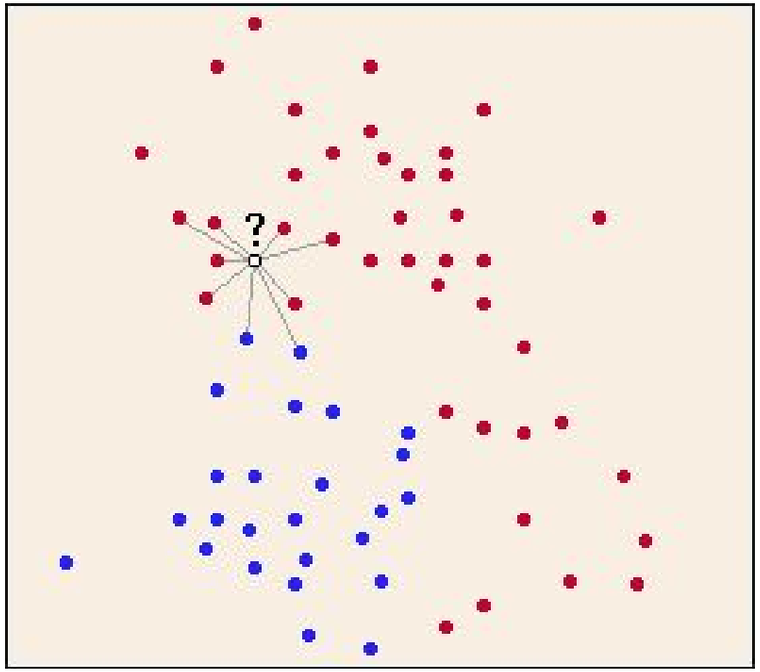
- **K Nearest Neighbors (KNN)** is a classification algorithm that makes a prediction based upon the closest data points.

- The KNN algorithm:
  - For a given point, calculate the distance to all other points.
  - Given those distances, pick the *k* closest points.
  - Calculate the probability of each class label given those points.
  - The original point is classified as the class label with the largest probability (“votes”).

WHAT IS K NEAREST NEIGHBORS?

- KNN uses distance to predict a class label. This application of distance is used as a measure of similarity between classifications.

- We’re using shared traits to identify the most likely class label.



WHAT IS K NEAREST NEIGHBORS?

- Suppose we want to determine your favorite type of music. How might we determine this without directly asking you?

- Generally, friends share similar traits and interests (e.g. music, sports teams, hobbies, etc). We could ask your five closest friends what their favorite type of music is and take the majority vote.

- This is the idea behind KNN: we look for things similar to (or close to) our new observation and identify shared traits. We can use this information to make an educated guess about a trait of our new observation.

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. In what other tasks do we use a heuristic similar to K Nearest Neighbors?

DELIVERABLE

Answers to the above questions

DEMO

KNN IN ACTION

KNN IN ACTION

- The following code demonstrates using KNN via sklearn.

```
from sklearn import datasets, neighbors, metrics
import pandas as pd

iris = datasets.load_iris()
# n_neighbors is our option in KNN. We'll tune this value to attempt
to improve our prediction.
knn = neighbors.KNeighborsClassifier(n_neighbors=5, weights='uniform')
knn.fit(iris.data[:,2:], iris.target)
print knn.predict(iris.data[:,2:])
print iris.target
print knn.score(iris.data[:,2:], iris.target)
```

WHAT HAPPENS IN TIES?

- What happens if two classes get the same number of votes?
- This could happen in binary classification if we use an even number for *k*. This could also happen if there are multiple class labels.
- In sklearn, it will choose the class that it first saw in the *training set*.

WHAT HAPPENS IN TIES?

- We could implement a *weight*, taking into account the distance between the point and its neighbors.
- This can be done in sklearn by changing the `weights` parameter to `"distance"`.
- Try changing the `weights` parameter. How does this affect accuracy?

WHAT HAPPENS IN HIGH DIMENSIONALITY?

- Since KNN works with distance, higher dimensionality of data (i.e. more features) requires *significantly* more samples in order to have the same predictive power.

- Consider this: with more dimensions, all points slowly start averaging out to be equally distant. This causes significant issues for KNN.

- Keep the feature space limited and KNN will do well. Exclude extraneous features when using KNN.

WHAT HAPPENS IN HIGH DIMENSIONALITY?

- Consider two different examples: classifying users of a newspaper and users of a particular toothpaste.
- The features of the newspapers are very broad and there are many: sections, topics, types of stories, writers, online vs print, etc.
- However, the features of a toothpaste are more narrow: has fluoride, controls tartar, etc.
- For which problem would KNN work better?

WHAT HAPPENS IN HIGH DIMENSIONALITY?

- KNN would work better on classifying users of a particular toothpaste since the feature set is more narrow and distinct.

CLASSIFICATION METRICS

INTRODUCTION TO CLASSIFICATION METRICS

- Metrics for regression do **not** apply to classification.
- We *could* measure the distance between the probability of a given class and an item being in that class. Guessing 0.6 for a 1 is a 0.5 error.
- But this overcomplicates our goal: understanding binary classification, whether something is black or white, right or wrong.
- To do this, we'll measure "correctness" or "incorrectness".

INTRODUCTION TO CLASSIFICATION METRICS

- We'll use two primary metrics: *accuracy* and *misclassification rate*.
- **Accuracy** is the number of *correct* predictions out of all predictions in the sample. This is a value we want to *maximize*.
- **Misclassification rate** is the number of *incorrect* predictions out of all predictions in the sample. This is a value we want to *minimize*.
- These two metrics are directly opposite of each other.
- 1 - misclassification rate = accuracy

INTRODUCTION TO CLASSIFICATION METRICS

- **WARNING**: You cannot use regression evaluation metrics for a classification problem, or vice versa. This is a common mistake.

- sklearn will not intuitively understand if you are doing regression or classification, so make sure to manually review your metrics.

INDEPENDENT PRACTICE

SOLVING FOR K

ACTIVITY: SOLVING FOR K

DIRECTIONS (35 minutes)

One of the primary challenges of KNN is solving for k - how many neighbors do we use?

The **smallest** k we can use is 1. However, using only one neighbor will probably perform poorly.

The largest k we can use is n-1 (every other point in the data set). However, this would result in always choosing the largest class in the sample. This would also perform poorly.

Use the lesson 8 starter code and the iris data set to answer the following questions:

1. What is the accuracy for k=1?
2. What is the accuracy for k=n-1?
3. Using cross validation, what value of k optimizes model accuracy. Create a plot with *k* as the x-axis and *accuracy* as the y-axis (called a “fit chart”) to help find the answer.

DELIVERABLE

Answers to the above questions

ACTIVITY: SOLVING FOR K

EXERCISE

STARTER CODE

```
from sklearn import grid_search

params = {'n_neighbors': }

gs = grid_search.GridSearchCV(
    estimator=,
    param_grid=,
    cv=,
)
gs.fit(iris.data, iris.target)
gs.grid_scores_
```

ACTIVITY: SOLVING FOR K

EXERCISE

DIRECTIONS

Bonus Questions:

1. By default, the KNN classifier in sklearn uses the *Minkowski metric* for distance.
   a. What *type* of data does this metric work best for?
   b. What *type* of data does this distance metric not work for?
   c. You can read about distance metrics in the sklearn documentation.

2. It is possible to use KNN as a regression estimator. Determine the following:
   a. Steps that KNN Regression would follow
   b. How it predicts a regression value

DELIVERABLE

Answers to the above questions

TOPIC REVIEW

REVIEW

- What are class labels? What does it mean to classify?
- How is a classification problem different from a regression problem? How are they similar?
- How does the KNN algorithm work?
- What primary parameters are available for tuning a KNN estimator?
- How do you define: accuracy, misclassification?

BEFORE NEXT CLASS

BEFORE NEXT CLASS

DUE DATE

- Project: Final Project, Deliverable 1

LESSON

Q & A