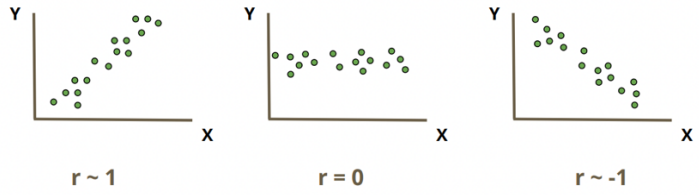
Lecture 10 Classification

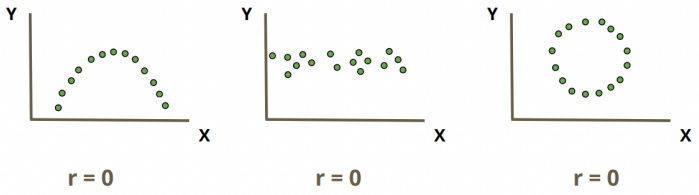
- Predictors , features and attributes are use to determine a class
    - Can create a model that takes attributes as input to a function and outputs a classification based upon attributes
- Sometimes there are multiple or no correct answers to determining what class a specific data point is in
    - This could be due to insufficient or wrong attributes representing the data or because the problem doesn't have an exactly predictable solution
    - For example:
        - If we used age instead of weight to classify rhinos against another animal that would be extremely inefficient
    - Key takeaways
        - There could be many correct answers or no correct answers
        - This is fine because no relationship also gives us information
        - Weather a task is feasible or not depends on the relationship between the predictor varibles and the class
- The feasibility of a classification task completely depends on the relationship between the attributions or predictors of the class
- "All models are wrong but some are useful"
- How do we know if we have good predictors?
    - Correlation

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}$$

    -
    - The numerator represents covariance
    - The left portion of the denominator represents the standard deviation of x
    - The left portion of the denominator represents the standard deviation of y



    - The closer the coefficient is to +1 or -1, the stronger the relationship. A correlation coefficient above 0.75 (or below -0.75) is generally considered a high degree of correlation. Values between 0.9 and 1.0 indicate a very highly correlated relationship.



    -

- Nominal: no order need to look at the means of X and how they differ across each Nominal Value of Y
  - Categories with **no inherent order** or ranking.
  - **Usage**: Just labels or names; you can't say one is “more” or “less” than another
- Ordinal : Instead of using the exact numbers assigned we can compare their rank / position in the data. So it doesn’t matter how far Bad is from Ok it just matters that Ok comes after Bad.
  - Categories that **do have a meaningful order**, but the **differences between categories are not quantified**.

Property	Nominal	Ordinal
Order	No	Yes
Ranking	Not meaningful	Meaningful
Arithmetic	Not allowed	Limited (no subtraction, etc.)
Examples	Colors, Gender, Country	Rating scales, Ranks, Grades

- The Spearman Coefficient
  - The Spearman rank correlation measures how well the relationship between two variables can be described using a monotonic function. It uses ranks instead of raw values and is good for ordinal data or nonlinear relationships

Student	X	Rank(X)	Y	Rank(Y)
A	90	1	88	1
B	70	3	65	4
C	80	2	82	2
D	60	4	70	3
E	50	5	55	5

Computing the difference in the ranks di=Rank(Xi)−Rank(Yi)

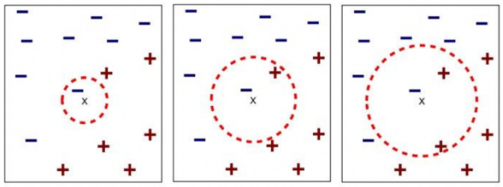
Student	Rank(X)	Rank(Y)	di	d^2i
A	1	1	0	0
B	3	4	-1	1
C	2	2	0	0
D	4	3	1	1
E	5	5	0	0

∑di^2=0+1+0+1+0=2

Plug into the spearman formula $$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

- Causation
  - Testing for causality requires specific testing /experimentation with a control group
- How do we know if we have done well at classification
  - Testing without cheating.
    - Learning not memorizing.
    - Split up our data into a training set and a separate testing set

        - Use the training set to find patterns and create a model
        - Use the testing set to evaluate the model on data it has not seen before
    - Also allows us to check that we have not learned a model TOO SPECIFIC to the dataset
        - Overfitting vs underfitting
        - Goal is to capture general trends
        - Watch out for outliers and noise
    - Training Step
        - Create the model based on the examples / data points in the training set
    - Testing step
        - Use the model to fill in the blanks of the testing set
        - Compare the result of the model to the true values
- Instance Based Classifiers
    - It makes predictions by **exactly matching** an unseen (test) record to one of the stored training records.
    - During training it memorized every training record and its class label
    - During testing if an unseen record extracting matches a stored training it will copy the classification
    - If no exact match is found it refuses to classify or returns an unknown
- Nearest neighbor Classifier
    - Stores all training examples
    - When given a new test instance calculates the distance beterrn the test point and all training points to find the training point that is the closest and assign the label of that point to the test instance
    - Typically using Euclidean distance, Manhattan distance, hamming distance
- K Nearest Neighbor Classifier

  Requires:
    - Training set
    - Distance function
    - Value for k
    - How to classify an unseen record:
        - Compute distance of unseen record to all training records
        - Identify the k nearest neighbors
        - Aggregate the labels of these k neighbors to predict the unseen record class (ex: majority rule)
            - Pick the most common class among those neighbors
    - 

    - Choosing the value of k
        - If k is too small -> sensitive to noise points + doesn't generalize well

        - If k is too big -> neighborhood may include points from other classes
- Pros:
    - Simple to understand why a given unseen record was given a particular class
- Cons:
    - Expensive to classify new points
    - KNN can be problematic in high dimensions (curse of dimensionality)