**Notes of Evaluation:**

* Underfitting: poor performance on the training data.The input features are not expressive enough to describe the target well.
* Overfitting: poor performance on the test data. The model performs well on the training data but does not perform well on the test data. This is because the model is memorizing the data it has seen and is unable to generalize to unseen examples.
* The total number of test examples of any class would? (TP+FN)
* Sum of FN of a class? (sum of the row excluding TP)
* Sum of FP of a class? (sum of column excusing TP)
* Sum of TN of a class? (sum of everything excluding row and column of that class)
* FPR = FP / (FP + TN), FNR = FN / (FN + TP), TPR = TP / (TP + FN), TNR = TN / (TN + FP)
* Accuracy = , Overall Accuracy =
* Cost Matrix = C(TP)a + C(FN)b + C(FP)c +C(TN)d
* Precision refers to the percentage of predictions that are correct.(biased towards TP &FN)
  + Precision =
* Recall refers to the percentage of total relevant results correctly classified by your algorithm. (biased towards TP &FP)
  + Recall =
* F1 score =
* Methods of Estimation
  + Holdout: reserves a certain amount for testing and uses the remainder for training.
  + Cross Validation

1. Data is split into k subsets of equal size (folds)
2. Each subset in turn is used for testing and the remainder for training.
3. **k**-fold: train on **k-1** partitions, test on the remaining one, and repeat this process k times so that each fold is used once for testing.
4. Leave-one-out: **k=n.** Guarantees that each record is used the same number of times for training and testing.
5. The performance metrics are then averaged over the k iterations to give a more accurate estimate of the model's performance. This method is used to reduce the variance of the performance estimate and to ensure that the model is not overfitting to the training data.

**Notes of Clustering:**

* Cosine distance is used to measure similarity of sets as vectors. cos = 1 same orientation, cos = 0 perpendicular.
  + A picture containing font, text, white, diagram

    Description automatically generated
* Jaccard for sets as sets.
  + 
* Euclidean sets as points.
* K-means:
  + Partition objects into K subsets
  + We have the centroids then we compute the distance between each point and the centroid using Euclidean and then pick the min distance.
  + Then we compute new centroids for each cluster. (by calculating the mean)
  + When no changes are made. STOP
  + How to select k? Try different **k**, looking at the change in the average distance to centroid as **k** increases Average falls rapidly until right **k**, then changes little.
  + Evaluating K-means clusters
    - Using SSE,

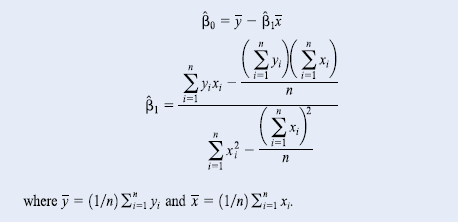
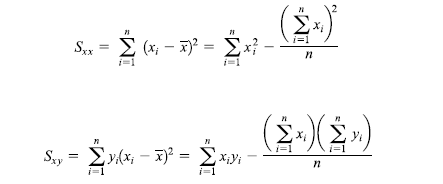
(slide 35-36)

* Hierarchical Clustering
  + Two main types of hierarchical clustering
    - **Agglomerative (bottom-up approach)**
      * Start with the points as individual clusters.
      * At each step, merge the closest pair of clusters until only one cluster (or k clusters) left.
    - **Divisive (top-down approach)**
      * Start with one, all-inclusive cluster.
      * At each step, split a cluster until each cluster contains an individual point (or there are k clusters)
  + Min hierarchical clustering (limit: Sensitive to noise and outliers)
    - Proximity of two clusters is based on the two closest points in the different clusters.
  + Max hierarchical clustering (Strength: Less susceptible to noise and outliers, limit: Tends to break large clusters and biased towards globular clusters)
    - Proximity of two clusters is based on the two most distant points in the different clusters.
  + Group Average hierarchical clustering.
    - We get the average distance.

**Notes of K-NN:**

* The impact of K on overfitting and underfitting: Low k may lead to overfitting (decision boundary tries to separate every single instance), large k may lead to underfitting(decision boundary generalizes in the separation)
* How to pick k
  + Common heuristics:
    - often 3, 5, 7
    - choose an odd number to avoid ties
  + Use development data
    - Experiment with different k values (using cross validation for example) and choose the best k (that has best performance measure)
* Decision trees vs. k-NN
  + Which is faster to train?
    - k-NN doesn’t require any training.
  + Which is faster to classify?
    - For most data sets, decision trees.
  + Do they use the features in the same way to label the examples?
    - k-NN treats all features equally. Decision trees “select” important features.

**Notes of Regression:**

* 
* 

A picture containing font, white, number, calligraphy

Description automatically generated





Hypothesis:



* We would reject H0 beta1 = beta1,0 if
* A picture containing font, diagram, white, line

  Description automatically generatedWe do the same for the intercept.

Beta0 –