**Natural Language Processing (NLP)**

**Session 2**

1. **Text Classification**:
   1. *Supervised*:
      1. *Input*: a document, a fixed set of classes and a training set of labelled data
      2. *Output*: a learned classifier / model
         1. Generative: build a model by aggregating the features of a class e.g., Naïve Bayes.
         2. Discriminative: build a model by separating the features of a class e.g., logistic regression.
         3. Discriminative vs. Generative:
            1. SVMs and other discriminative classifiers can outperform NB and generative approaches with enough data.
            2. However, SVMs and other more complex approaches are less interpretable than Bayesian approaches.
            3. Logistic Regression can give a bit of the best of both worlds, as its co-efficient (feature weights) can be interpreted, and it can reach high-performance with optimization.
2. **Naïve Bayes**: Based on Bayes Rule (ML – Notes, Week 2, Session 2, Page 2). If we make very naive assumptions about the generative model for each label, we can find a rough approximation of the generative model for each class, and then proceed with the Bayesian classification.
   1. To prevent underflow of numerical probabilities, summing up of logarithmic probabilities can be implemented.
   2. Advantages:
      1. Works quickly and can save a lot of time.
      2. Suitable for solving multi-class prediction problems.
      3. If its assumption of the independence of features holds true, it can perform better than other models and requires much fewer training data.
   3. Disadvantages:
      1. Naive Bayes assumes that all predictors (or features) are independent, rarely happening in real life. This limits the applicability of this algorithm in real-world use cases.
      2. This algorithm faces the ‘zero-frequency problem’ where it assigns zero probability to a categorical variable whose category in the test data set wasn’t available in the training dataset. It would be best if you used a smoothing technique to overcome this issue.
      3. Its estimations can be wrong in some cases, so you shouldn’t take its probability outputs very seriously.
3. ***Maximum a posteriori* (MAP) hypothesis for classification**:
   1. Density estimation is the problem of estimating the probability distribution for a sample of observations from a problem domain.
   2. MAP estimation is a probabilistic framework for solving the problem of density estimation.
   3. MAP involves calculating a conditional probability of observing the data given a model weighted by a prior probability or belief about the model.
   4. MAP provides an alternate probability framework to maximum likelihood estimation for machine learning.
4. **Smoothing**: Zero probabilities (noise) cannot be conditioned away, however for training purposes, the noise can be problematic i.e., infinity values, etc. Data smoothing uses an algorithm to remove noise from a data set, allowing important patterns to stand out.
5. **Precision**: Precision is the fraction of true positive examples among the examples that the model classified as positive. In other words, the number of true positives divided by the number of false positives plus true positives.
6. **Recall**: Recall, also known as sensitivity, is the fraction of examples classified as positive, among the total number of positive examples. In other words, the number of true positives divided by the number of true positives plus false negatives.
7. **F-score**: The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into ‘positive’ or ‘negative’.  
     
   The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model’s precision and recall.  
     
   The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing.
8. **Feature Classification**:
   1. *Tokenisation*
   2. *Stemming*: Porter’s algorithm
   3. *Feature selection*:
      1. Mutual information: In training set, choose k words which best discriminate the categories.
      2. Information gain: Metric much used in ML to decide on the importance of features. Measures how much a given feature (word) reduces the entropy.
9. **Logistic Regression**: Discriminative nonlinear classifier. Summation of the element-wise product of the weight matrix and the feature vector.
10. **Hyperplanes**: To enable linear classification in dimensions higher than 2, need hyperplanes.
11. **Support Vector Machine (SVM)**: For optimum solution to the different hyperplanes, supporting vectors which are those datapoints that the margin pushes up against. SVMs maximize the margin around the separating hyperplane.