NLP WRITE UP FOR TOPIC MODELLING

Task: Topic Modelling. For a given question, the program needs to be able to output the associated topic.

For the task three Natural Language Processing Ideas would be employed to solve them. These ideas include:

1. Naïve Bayes Classification
2. Logistics Regression
3. Latent Dirichlet Allocation (LDA)

# 

1. NAÏVE BAYES CLASSIFICATION
   1. **What is Naïve Bayes Classification?**

Naive Bayes Classification is a probabilistic technique used to assign category/class labels to a given input by assuming conditional independence between features. The class labels are drawn from some finite set. It is a form of supervised learning in that it takes a set of inputs and their corresponding output labels trains on them and is able to take a new input X and for the set of output classes Y1 to YN determines if the class of X is any of the output classes. There are three types of Naive Bayes implementations:

1. Gaussian Naive Bayes: Useful if your data follows a normal distribution
2. Multinomial Naive Bayes: Useful when data can be represented as a vector of counts. Works close to the Bernoulli but goes further than binary outputs.
3. Bernoulli Naive Bayes: Useful if your output classes are binary
   1. **Why is it Useful to solve the Problem?**

For our current task, we have a list of questions and their corresponding topics. The topics can be thought of as classes or categories within which the questions belong. Naive Bayes is useful for breaking down the questions (documents) into features and determining the probability of a given topic generating a particular feature/question. When given a new question, we would be able to use these probabilities to determine the topic more likely to generate the question thereby allowing us to determine the topic.

* 1. **Steps for Our Implementation**

We ultimately looking for the topic Ť that has the maximum probability given a question. This is denoted by:

We would use Bayes rule to break down the conditional probability above into a more convenient form for calculation

Bayes rule is represented by:

For a given topic t and a question q the above becomes:

We take the maximum probability of the calculation to help us determine the classification hence the above equation becomes

The denominator can be dropped from the above equation because it will be the same for all calculations hence removing it entirely would render the same result but would be conveniently similar at the same time.

P(q|t) is referred to as the likelihood of the question. This is the probability of getting the question given the topic

P(c) is referred to the prior probability of the topic. This is the probability of getting the topic on its own. It is computed independently of the documents.

The question q can be represented as a set of features.

For instance, the question ‘Is this a Natural Language Processing Class’ can be represented by the set of features (‘Is’, ‘this’, ‘a’, ‘Natural’, ‘Language’, ‘Processing’, ‘Class’) with a feature being each word in the question.

This being said, the topic can be generated by

Naive Bayes assumes that the order of the features do not matter, only the frequencies do. It also assumes conditional independence, that is, the probabilities of each feature given a class are independent of other features.

*Training the Classifier*

As mentioned earlier, Naive Bayes is a supervised machine learning algorithm. Hence it can learn from a given set of inputs and finite outputs to determine the output of a new given input.

We need to find P(t) and P(fi|t).

For the question prior P(t) we are looking for what percentage of questions are of the topic t. This can be computed by:

P(fi|t) can be found by:

A feature in this case refers to a word in the question. To get the denominator, the total number of features (word) in topic, we simply place all words from all questions into a bag representing each topic. So all words in questions that pertain to BLOCKCHAIN will be put in a bag labelled BLOCKCHAIN. The total number of word counts in this bag will be our denominator. For a given word chosen, its count in the bag of a given topic will be our numerator.

In the event a given word does not appear in a topic, Laplace Smoothing will be used. Laplace Smoothing adds 1 to the count of a given feature and the total sum of additions to the count of all features in a document (the denominator). This is necessary because if a given word has a zero count for a given topic, the numerator would be zero hence P(fi|t) would be equal to zero. When this is multiplied with other probabilities, it would make the entire computation zero. Adding 1 to every count prevents this. Since 1 is added to every count in the numerator, the total number of additions must be added to the denominator to keep the computation ‘balanced.’ We will ignore all unknown words and remove stop words.

1. LOGISTICS REGRESSION

## **What is Logistic Regression?**

Used to classify an observation based on features it possesses. Unlike Naive Bayes which is a generative classifier, that is, it learns and creates models for classes to generate output and given a test input, would generate the more likely output and compare, Logistic Regression is a Discriminative classifier. This means, it does not learn a model of a class but simply looks out for particular features and based on the presence/absence of certain features, makes a classification.

Components of Logistic Regression:

* A feature representation of the input. This is a vector of features for each input observation.
* A classification function that computes ŷ
* A function for minimizing error in training
* Algorithm for optimizing the objective function
  1. **Why is it useful in solving the problem?**

Logistic Regression is useful for similar reason to Naive Bayes described above. We have a list of questions and their corresponding true labels and would like to classify new questions based on these labels. Logistic Regression has the added benefit of being more accurate as it distributes weights among strongly correlated features. This is unlike Naive Bayes whereby conditional independence sees strongly correlated features appear essentially as double features which leads to overfitting.

*How it is implemented*

Since we have multiple class labels on which to classify our inputs we will perform multinomial logistic regression.

*Classification in Multinomial Logistic Regression --Testing*

Given a sample an input, we transform it into a vector of features and the dot product of this and a vector is taken. The result is summed and a bias term is added. This result is placed into a softmax function to compute the probability p(y = c|x) (the probability that y is a particular class given the input). For a vector i of dimensionality k, the softmax is defined as:

The purpose of the softmax is to squash the result of the dot product between the vector of weights and the vector of input features. Each class will have its own weight vector and bias term. Each weight is associated with one of the features of the input x. The weight represents how important a particular feature is to determining the classification.

*Training*

We train to obtain weights w and b using stochastic gradient descent and cross-entropy loss. Cross entropy loss function is a function describes how close the predicted output is to the true output. It seeks to minimize the negative log likelihood loss. This helps identify the parameters w and b more likely to produce a low difference between the predicted output and the true output.

The goal of Gradient Descent is to iteratively optimize the loss function to obtain weights w and b that reduce the likelihood loss between the predicted output and the true output. It does this by attempting to find the minimum of the function. It identifies the part of the functions slope that increases most steeply and moves in the opposite direction. Intuitively, this means it seeks to approach the global minima of a function as this would present the area of lowest loss. The magnitude of the amount of how much to move is called the learning rate. Stochastic Gradient Descent is an algorithm that performs gradient descent. It computes the gradient of the loss function at each point and moves in a direction opposite to the gradient. After one iteration the weights for w and b would change and the process would begin again.

1. LATENT DIRICHLET ALLOCATION (LDA) AND RULE-BASED APPROACHES
   1. **What is Latent Dirichlet Allocation?**

Latent Dirichlet Allocation (LDA) is an example of topic modeling and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. What happens with LDA, given a set of documents containing words, it makes a prediction about the categories of the documents based on the number of similar words it sees between documents. It refines its prediction by making more observations about the similar words in documents. After enough observations, the document is able to categorize documents accurately using the grouping of similar words within the documents.

* 1. **Why it is useful for solving the problem?**

For our current task, we have a list of documents containing sentences and words that can be placed into categories. We seek to classify the sentences in the document to a particular topic. Each sentence has a topic and the words in the sentences for each topic can be said to have similarities hence LDA should be able to identify and these similarities and perform the necessary categorisation

* 1. **Steps for Implementation**

1. The Latent Dirichlet Allocation library was imported to enable us to build the LDA model. Also, the Count Vectorizer library was imported to help normalized each sentence in the training data file and to keep counts of the occurrence of the various words in each sentence.
2. The vectorized data was then transforming to obtain integer representations of each feature per sentence.
3. The LDA model was then built by calling the Latent Dirichlet Allocation class library.
4. A “most\_similar” function was then defined to determine the most similar topic that corresponds to a question passed as a test sentence.