Method:

Throughout the experiment, we have tried four different Machine learning algorithms:

1. Naïve Bayes: At first, we run the Multinomial Naïve Bayes model due to the speed and ease of training of the model. It is also consider as common baseline for text classification. However, the model assumes a very strong assumption which assumes that the probability of each word occurrence in a comment is conditionally independent of the occurrence of other words in a comment.

Given *𝑥(i) = (𝑥1 (i), 𝑥2 (i), … 𝑥j (i))* being the training example (i), we first maximize the joint likelihood of the data:

*L (φy, φj|y=1, …, φj|y=k) = ∏i  p(𝑥(i), y(i) ) = ∏i  ∏j p (𝑥j (i)| y(i)) \* p (y(i))*

Maximizing the joint likelihood of the training data gives:

*p (𝑥j (i)| y(i) = k) = φj|y=k = (∑I 1 {𝑥j (i) = 1∧ y(i)=k}) / (∑I 1 {y(i)=k})*

*p (y(i)=k) = φy=k = (∑I 1 {y(i)=k}) / m,* where m = number of training examples.

Then, class label will be based on the highest posterior probability:

*y = argmaxk (∏j φj|y (𝑥 j (i)) \* φy=k ) / (∑k ∏j φj|y (𝑥 j (i)) \* φy=k)*

1. Multinomial Logistic Regression: As second model, we have implemented a multinomial regression using the softmax function since we are dealing with more than 2 outcomes.

As objective, we want to find the parameter *θ* that maximize the log-likelihood:

*ℓ(θ) = ∑i log ∏j (exp(θjT 𝑥(i)) / ∑k exp(θkT 𝑥(i))1 {y( i)= l})*

Then based on the *θ*, the training example will be assigned to the class with the highest conditional probability defined as follows:

*p (y = k |𝑥;𝜃) = exp(θkT 𝑥(i)) / ∑j exp(θjT 𝑥(i))*

1. Ridge Classifier: Thirdly, we decide to experiment with a Tikhonov Regularization which is also known as Ridge Classifier. The Ridge Classifier is often used prevents to over fit the data.

We determine the ridge coefficients (*W*) that minimize the following loss function:

*∑i || 𝑥(i) W – y(i) ||2 2 + α || W ||2 2 , α ≥ 0*

The second term is used to penalize the matrix *W* being too large. In other words, if the matrix *W* takes on large values, regularized loss function will be penalized. This will encourage the fitted model to be a simple model rather than a complex model and usually this will prevent overfitting.

1. Linear Support Vector Classification: Finally, we use the Linear Support Vector Classification to perform the multi-class classification. SVM has an objective to maximize the margin between the classes. This can be achieve by determining the parameters of the hyperplane (*w* and *b*) subject to *wT x + b = 0*.

Given *𝑥(i) = (𝑥1 (i), 𝑥2 (i), … 𝑥j (i))* being the training example (i) and the output *y ∈ {−1, 1}*, the goal is to solve the following problematic. For the SVM k, we wish to:

*min (1/2 WkT Wk + c ∑i ζk(i))* , where Wk ζk bk are the parameters for kth SVM

*wk,bk,ζk*

subject to *y(i) (WkT 𝑥(i) + bk) ≥ 1 - ζk(i) and ζk(i) ≥ 0*

In our experiment, since we are dealing with multi-class classification, we will employ one-vs.-all fashion so k SVMs is build, where k is the number of class label.