Word Count: 747

CE802 Pilot Study

This task requires me to identify how likely a customer of a travel insurance company will be to make a claim. There are two ways of looking at this problem. The first is as a binary classification problem. Where the task is to identify either if a customer is going to make an insurance claim or if they will not. Being a classification problem, the output will either be true or false (1 or 0) splitting the customers into two categories which in this scenario will split them by who gets the discounted premium on their travel insurance with the ones who are predicated to not make a claim will get the discount. However, this can also be a regression problem, where instead of predicating if they will make the claim or not, we instead predict how likely they are to make a claim giving us a continuous output instead of a categorical one. Although these are both different methods of modelling a problem there is some overlap. The method I suggest using to solve this problem is a classification one. This being that I would use a classification algorithm to predict a continuous value. As in a regression solution. However, the continuous values will be grouped in sets in the form of a label. Making it a classification solution. For example, there could be 10 labels. As the labels ascend so does the probability that the customer will make a claim. 1 being very unlikely and 10 being very likely. Each customer will then be classed by either one of these labels.

Some ideal features that could be used to help put each customer into a class would be things such as have they made a previous claim (those who have claimed before are statistically more likely to do so again). Information of any existing or previous health conditions of a customer would also be a useful feature to look at, if the customer is in poor health and needs treatment abroad then they will more than likely make a claim. The destination and purpose of any trip is also important to look at. Some places and activities will be more high risk than others for example comparing a business trip to an adventure holiday.

There are a few options for which machine learning algorithm to use here. As we are looking at this as an multi class classification problem these include k-nearest neighbour, decision trees, support vector machines and Naïve Bayes. Firstly, I would eliminate the idea of using k-nearest neighbour. This is because we do not know the size of the dataset being provided. This method has a very large time and space complexity when being used on larger datasets. Therefore, if we do use this and are provided a large dataset there will be a large computation cost. Another option is to use Naïve Bayes. This could be a good option for us. It works well with features which are not relevant to the classification as the assumptions are of equal weights. However, there is still an issue with using it. This method expects that the features are mutually independent to each other. This may not be the case in our dataset and therefore may not be the best algorithm to use. Another option is using a decision tree. This is likely a good option for the problem we have defined. One of the main benefits of this is that this method can provide an understandable explanation about the prediction. From the travels company point of view this data will be very important to them. It will allow them not only to understand the decision, but it may also provide more important data about their customers. No one algorithm will ever be perfect, however. Using a decision tree does have some issues. The first is that the tree may grow to be complex if the dataset is complicated. It may also be prone to outliers and overfitting. However, this can be countered by switching to a random forest algorithm. This will produce a collection of trees and take the average to select the predicated output.

To help evaluate the decision tree I would use for this problem I would use cross-validation. This will allow me to find the total error from multiple runs where I subset the data into test and training, and then swap the roles after the first iteration. However, this may be computationally expensive.