Kaggle Problem: Prudential Insurance

Matt Scislowski, Ben Stoehr, Jaskirat Vig

Introduction:

For this problem, we are working to develop a multi-step classification algorithm in an effort to correctly map an new individual client to the proper insurance policy. Buying individual life insurance is unpopular because there is no “one-size-fits-all” plan; 40% of adults in the US do not have life insurance. By leveraging a semi-large training dataset (10,000), we hope to develop a method to accurately select the recommended plan for the clients given in the testing dataset. We have used Linear Regression, from the sklearn package, to build our first iteration of the algorithm.

Related Work:

Support Vector Machines for Multi-class Classification

<https://www.cmpe.boun.edu.tr/~ethem/files/papers/iwann99.pdf>

This paper gives a nice outline of the use of an SVM in a multi-class classification problem. They first explain the use of an “one-against-all” SVM for a 3-class system. It describes how the decision regions change as you change the normalization of the output of the support vector. The first method of normalization they explain is based on optimizing a linear combination of the (phi) of the support vectors from each SVM so that the resulting combination has a 2-norm of 1.

The second method they explain is the use of an SVM for each class which can give information beyond a yes/no response. By leveraging the outputs of each SVM, and a mixture matrix holding weights for each SVM, given that the mixture matrix is “designed to minimize the mean square error between the output of the svm and {-1,+1, … , -1}”

Implementation:

Our implementation revolves around a simple linear regression currently. While not optimal we used this as a basis to explore and understand the dataset. Our initial approach consisted of splitting up certain elements such as Product Info 2 into separate columns in addition to adding new variables like the total count of a patient's Medical Keywords. By providing further depth to the dataset it is easier for the classification to distinguish between patients and assign the proper scores. Using sklearn, we created a generic linear regression that was trained on the library provided and the used to predict the responses for the testing set. From this we separated the predictions from a public set of cutoffs from the previous competition as a starting place for our analysis. The aim is to later implement our own script in determining these cutoffs.

Result Interpretation:

Our current classifier is clearly not ideal with approximately a 0.4 classification score and shows a pretty evident disparity between our results and the expected results. We hope to explore further classifiers to better fit this dataset along with fitting the data for an analysis more properly.

Group Work:

Matt S - Implementation, Paper on Implementation

Ben S - Paper: Intro and Related Work

Jaskirat Vig: Implementation