Travel Insurance Claim Prediction

Pilot Study (1063 words)

CE-802 Machine Learning and Data Mining

We are tasked at predicting if insurance company should offer a discounted premium to customers that are less likely to make a future claim and to work on it, historical data with claim status is provided to you.

We can exploit the capability of machine learning to work on this requirement. We can formulate this task as supervised learning task in which target can have 2 values True and False for the target feature claim_status: claim_status will be true if the claim is filed otherwise it will be false.

Because each record can be classified as either True or False, we can further narrow down this task as a binary classification task.

A classification model can certainly perform very good at this task provided that it has the required info to get trained on. As a Machine Learning Consultant, I would expect below features to be available in data to build a strong model for this task-

- Claim.Status→ This is target we are looking for.
- Purpose of travel→ If you are planning on any adventurous activities like skiing, scuba diving or hiking. you may need to pay extra to get covered as the risk is more.
- Duration of travel→ The longer you are away, the more you're likely to get into some risk.
- Destination of travel→ Local factors, such as the price of medical treatment and repatriation, can make your policy more expensive.
- medical conditions→ If you have a pre-existing medical condition.
- Cost of insurance→ How much did you spent in buying the insurance. More cost, more chances that a person will claim as s/he has spent bigger amount in buying it.
- Gender of insured:→ It might be that men are more likely to claim than women or vice versa.
- Age of insured→ Insurance depends on age. Insurance is cheap in your 20s and 30s, but the older you get the costlier it becomes.

There are many classification algorithms that we can use for this task. Each algorithm has its own advantages and disadvantages. I will perform a comparative study with these algorithms and see which one produces good result. Below are few classification algorithms which I will be testing-

- Decision Trees→ DTs are easy to interpret even by end user (Business). Its rules are in the form of if-else condition that are way easier to interpret in comparison to other algorithm's complex rules such as equations of logistic regression. Also getting important attributes with DTs are trivial because it calculates information gain for all of them: higher the info gain, more important the feature is.

- Random Forest→ Random Forest is the collection of multiple decision trees, combined at the end of the process (using majority votes). A single decision tree is a weak predictor but is relatively fast to build. More trees give you a more robust model and prevent overfitting.

- Gradient Boosting→ Like random forest, Gradient Boosting too is the collection multiple decision trees, but they are built is different way. random forests builds each tree independently while gradient boosting builds one tree at a time in stage-wise manner, introducing a weak learner to improve the shortcomings of existing weak learners

- Logistic Regression→ Logistic regression will work well if there is a single decision boundary with the help of which we can classify the data. We will check if our data is linearly separable.

- Support Vector Machine→ There are many advantages of using SVM. kernel tricks of SVM makes them capable to solving variety of problems. SVM not just only try to find the decision boundary, but also tries to maximize the margin of it for better prediction on future data. It uses only a subset of data (called support vectors) in prediction task.

I will also try to use deep learning model as well (MLP). Sometime with right parameters we can build deep learning models that can outperform machine learning models.

To evaluate the performance of the model we can follow 2 approaches-

First, we can divide the training set in 3 parts randomly (Train, Val and Test). Train and Val part, we can use to train the algorithms and fine tune its parameter for better performance. Once the best algorithms is found we will use the Test part to check the final performance of the model.

Alternatively, we can use k-fold cross validation which I think is a better approach for model selection and parameter tunning. With K-fold validation we don't need to divide data in Train and Val explicitly. K-fold cross validation does it internally its own. We can get a performance score for each data fold which is better than a having a single score of just one run. Because all Train and Val data is being used in calculating the performance score, this score in better representative of modal performance.

No matter which cross validation method are we using, we need a metric to support this whole procedure. A matric takes model predicted values and ground truth values as argument and produces a score. As good as this score is, as good your modal. For classification task we can use below metric. Each matric has its own advantages and disadvantages-

Classification Metrics

- Accuracy.
- Precision
- Recall
- F1-Score
- Confusion Matrix

Here, it might be the case that we do not have balanced dataset because I think most of the rows would fall under no claim. If this is the case accuracy is not a good matric to use.

For our case, I think precession is important, which says among total predicted for a category what percentage were correct predictions. This is due to that fact that if our model predicts some one will not claim and later if that persons claims then it will be a big loss for the company.

Recall also seems a good metric which says of total data available for a certain category, what percentage has returned correctly. This is because if there are many people who will not claim but our model is not able to identify them, it will again be a big loss for the company.

Since we want precision and recall both be high, I will check F1-core as well, which is the harmonic mean of both.

I will also use confusion matrix which can give detailed analysis of how model is performing by drawing a comparison between actual and predicted values in form of a matrix.