Allstate Purchase Prediction

Homayoun Sadri

ABSTRACT

In this paper, I do a multivariate analysis of Allstate insurance transactions to find a classifier that can predict the policy option a customer will end up purchasing base on their quotes. Classification algorithms implemented here are K Nearest Neighbor and Random Forests. Different values of K for K Nearest Neighbor algorithm and different values of number of trees and maximum depth for Random Forest algorithm are examined and the optimum values has have been presented.

INTRODUCTION

When customers shop for an insurance policy, they receive several quotes with different coverage options before they make a purchase. The goal here is to predict the purchase coverage options for each customer. The sooner the issuer can predict the eventual purchase, the less likely it is to lose the customer. I attempt to find the answer to the following question:

Can we predict what policy option a customer will end up purchasing based on their quotes?

DATA SET

The data set for this project is from Allstate purchase prediction challenge on Kaggle.com: https://www.kaggle.com/c/allstate-purchase-prediction-challenge/data

Two sets of data have been provided: The training data and the test data. Since in the test set, we only have a partial history of the quotes and do not have the purchased coverage options, I have dropped the test data and use only the training data.

The training data contains transaction history for customers that ended up purchasing a policy. For each customer, their entire quote history is given, the last row of which contains the coverage options they purchased. To make the problem easier, I have only kept the rows corresponding to the purchased option, resulting in one row for each customer.

Each row of the data set contains independent variables (IVs) and correct labels for classes.

Classes: Any combination of the options is a product (i.e. an insurance policy). Therefore, a product is a vector with length seven whose values are chosen from each of the options listed in table 1. The classification task is to identify, for each customer, the combination of seven coverage options they end up purchasing.

Since for each sample we need to predict all options and each option can take up on multiple values, we deal with a multi-class multi-output classification problem. We have seven super-classes or a total of $3 * 2 * 4 * 3 * 2 * 4 * 4 = 2304$ classes.

IVs: For each sample, there are seventeen IVs. The IVs and their descriptions are listed table 2.

Aside from removing the quote history and keeping only the purchase history, the following data preparation is performed:

- The "customer_ID" is removed: since after removing the quote history, there was only one row per customer, this variable is not needed anymore.
- "state", "car_value" and "location" are converted to integers values: this was necessary to comply with the algorithm requirements.
- A new value is added for missing "duration_previous" variable: these are customers that did not have a previous insurance policy with Allstate.

Furthermore, since the test data has been left out and only the training data is kept, this data set is used for both training and testing. First the data set is randomly split into two halves. One half is used for training and the other half is again randomly split into two halves. One used for cross-validation and one used as out-of-sample data.

METHOD

Logistic regression was my first option for the classification but upon further research, I realized this algorithm cannot handle "multi-class multi-output classification" in python's scikit-learn as of right now. So, I implemented: K-Nearest Neighbor and Random Forest.

First, KNN algorithm is trained and tested with K = 1 to K = 35. The accuracy for each K is recorded and best performance is chosen as the optimal K.

The measure of performance here, called accuracy, is the average of seven Accuracies corresponding to seven option policies. Since the algorithms in scikit-learn are not able to handle the multi-class multi-output classification, there is one classifier trained for each super class or policy option. And the accuracy for each option policy is defined as:

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

Next, the Random Forest algorithm is trained and tested as follows: First keeping maximum depth constant at 5, the algorithm is trained for number of trees 1 to 40 and its performance on test data is recorded. Then keeping the number of trees constant at 10, the algorithms is trained for maximum depth of 1 to 40 and the corresponding accuracies are recorded.

RESULTS

In figure 1, average accuracy changes with different values of K in KNN for testing accuracy in red and cross-validation accuracy in blue for is drawn. The testing accuracy deceases as K increases and the cross-validation accuracy increase with K and then decreases. Figure 2 shows the individual and average accuracy for the optimized K in KNN. For K = 14 the accuracy is 0.552639 which is not great.

The Training and cross-validation accuracies for different number of trees at constant maxium depth is shown in figure 3. Figure 4 shows the training and cross-validation accuracies in red and blue respectively, for constant number of trees and maximum depth of 1 to 40. As we can see at depth of 13 the cross-validation accuracy starts to drop and the training accuracy starts to move towards one, which means that the model is overfitting the data. Therefore, maximum depth of 13 is determined as the best maximum depth. Figure 5 shows the accuracies for the optimized Random Forest with number of trees of 25 and maximum depth of 13. The average accuracy for our optimized classifier is 0.647372.

Conclusion

Predicting what policy option a customer will purchase based on their quotes could be very valuable for an insurance company. I implemented two classification algorithms for this prediction. The random forest algorithm showed better results with best accuracy of 64%. This isn't considered a high accuracy and further analysis is required to improve this.

Future work on this problem could involve Principal Component Analysis and factor Analysis to identify more important or easier to interpret variables. Also, different classification algorithms could be implemented and compared with the algorithms used here.

APENDICES

APENDIX 1. Describing data

Table 1. Product Options

Option name	Possible values
A	0, 1, 2
B	0, 1
C	1, 2, 3, 4
D	1, 2, 3
E	0, 1
F	0, 1, 2, 3
G	1, 2, 3, 4

Table 1. Independent Variables

customer_ID	A unique identifier for the customer
shopping_pt	Unique identifier for the shopping point of a given customer
record_type	Unique identifier for the shopping point of a given customer
day	Day of the week (0-6, 0=Monday)
time	Time of day (HH:MM)
state	State where shopping point occurred
location	Location ID where shopping point occurred
group_size	How many people will be covered under the policy (1, 2, 3 or 4)
homeowner	Whether the customer owns a home or not (0=no, 1=yes)
car_age	Age of the customer's car
car_value	How valuable was the customer's car when new
risk_factor	An ordinal assessment of how risky the customer is (1, 2, 3, 4)
age_oldest	Age of the youngest person in customer's group
age_youngest	Age of the youngest person in customer's group
married_couple	Does the customer group contain a married couple (0=no, 1=yes)
c_previous	What the customer formerly had or currently has for product option C (0=nothing, 1, 2, 3,4)
duration_previous	How long (in years) the customer was covered by their previous issuer

APPENDIX 2. K Nearest Neighbor Results

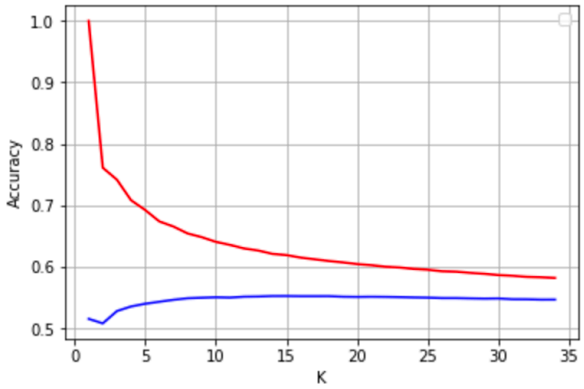

Figure 1. Training & Cross-Validation Accuracy in KNN

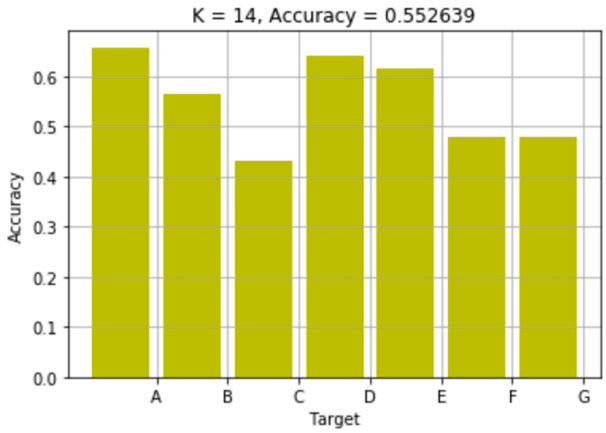

Figure 2. KNN results for best K

APPENDIX 3. Random Forest Results

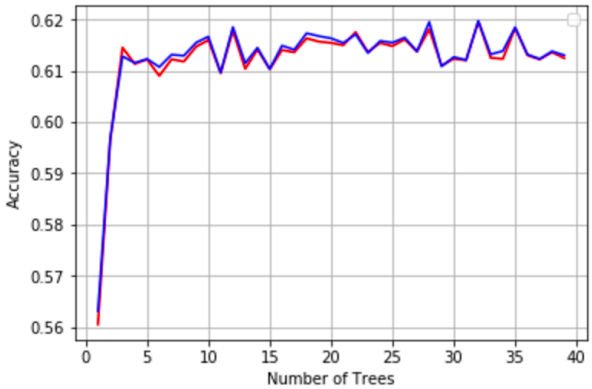

Figure 3. Training & Cross-Validation Accuracy in Random Forest Vs Number of Trees

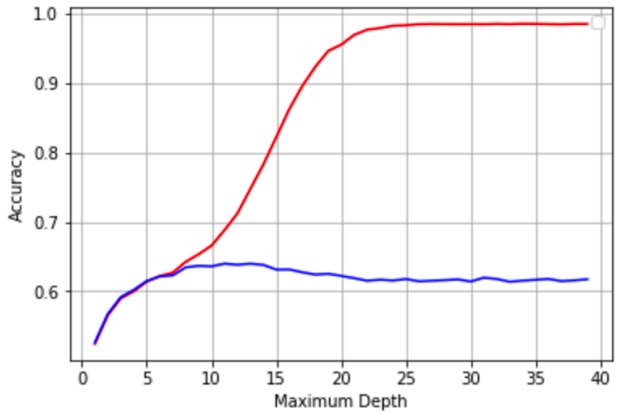

Figure 4. Training & Cross-Validation Accuracy in Random Forest Vs Maximum Depth

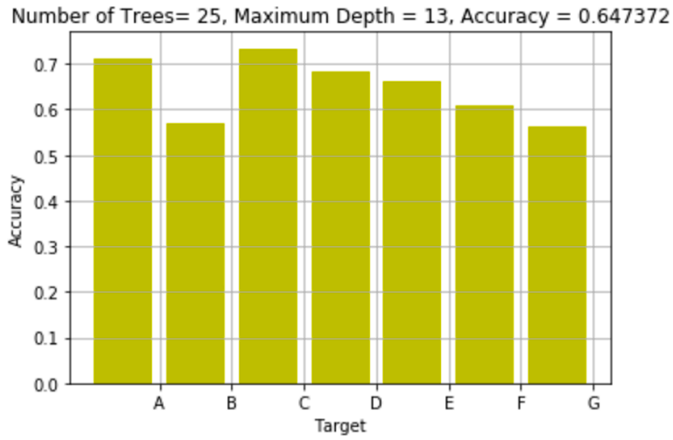

Figure 5. Random Forest Results for Best # of Trees and Maximum Depth