Health Insurance Cross Sell:

Identifying Vehicle Insurance Prospects from Existing Customers

Samuel P. Sears

Bellevue University

Abstract

In this project, a series of methods are used to identify cross sell opportunities within a data set of a company's existing health insurance customers. Whether or not the customer is interested in vehicle insurance is known. The goal of the analysis is to be able to identify whether or not a customer would be interested in vehicle insurance for a new data set in which this information was not known. The methods described include customer segmentation based on grouping known information about the customer through exploratory data analysis, predictive modeling, and customer segmentation based on probabilities given by the predictive models. For predictive modeling, a wide range of algorithms are used including, but not limited to, Light Gradient Boosting Machine and densely connected neural networks. In identifying customers who are interested in vehicle insurance, the project aims to reduce the number of customers the company would have to solicit the additional product to and provide information needed to create a more targeted communication strategy.

Keywords: neural network, unbalanced classification, light gradient boosting machine, machine learning, binary classification

Health Insurance Cross Sell:

Identifying Vehicle Insurance Prospects from Existing Customers

According to Investopedia, cross-selling is one of the primary methods, and perhaps one of the easiest, ways companies can generate new revenue and grow their business (Hayes, 2021). A cross-sell is when a company sells additional products to an existing customer. In a recent Kaggle competition, Anmol Kumar tasked data science enthusiasts with creating a way to identify cross-sell opportunities though predictive modeling (Kumar, 2020). Provided a dataset of health insurance customers, could a model be created that identifies customers who were interested in vehicle insurance?

Data

The data set was composed of customer information along with a response variable for whether or not the customer was interested in vehicle insurance. Each row represented one customer. Customer information included demographic information like gender and age, vehicle information, premium, and sales channel. See Appendix A for a full list of variables and their explanations. Overall, there were a total of 381,109 records with 10 explanatory variables, one response variable, and one id variable. The response variable indicated whether or not the customer was interested in vehicle insurance.

Business Problem and Hypothesis

According to Goose Digital, many insurance companies rely on ad hoc conversations between customer service representatives or producers for cross selling (Executing, 2020). Further, Goose Digital states that customer service representatives are typically busy with other tasks and may not always try to cross sell (Executing, 2020). In the data for this project, only 12.26% of existing customers were interested in vehicle insurance. The ad hoc nature of this strategy means the company is likely not reaching all the customers that would be receptive to a cross sell. However, reaching out to all customers may not be feasible or welcomed. This project proposes solutions to this problem through

customer segmentation and predictive modeling to identify groups of existing health insurance customers who are more likely to be interested in vehicle insurance.

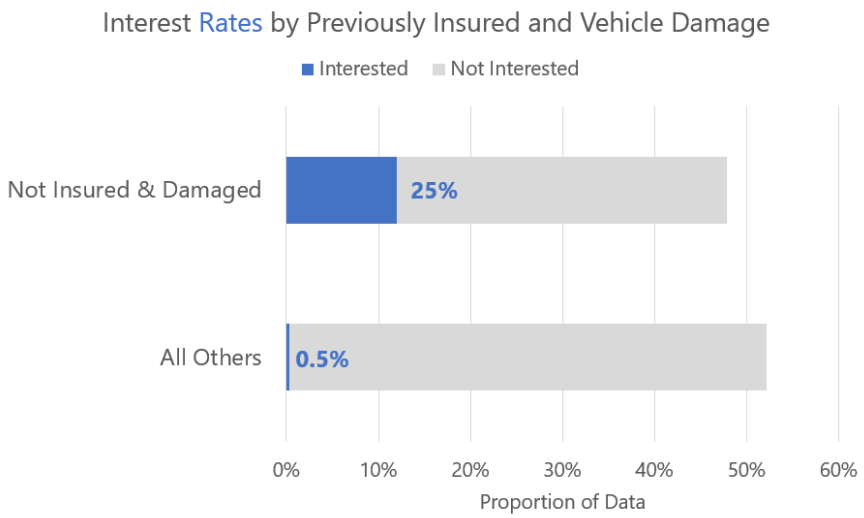
Exploratory Data Analysis

In the first portion of the project, rates of interest were gathered when grouping the data by the different explanatory variables. Several insights were gathered which could reduce the number of customers the company would need to reach out to for cross selling opportunities. One of the variables in the data set was Previously_Insured. This variable indicated whether or not the customer already had vehicle insurance. Approximately 46% of existing health insurance customers already had vehicle insurance and 54% did not. Of the 46% of customers who already had vehicle insurance, less than 0.1% of them were interested in the company's vehicle insurance. In eliminating the focus on these customers, the company could reduce the number of cross sell communications by almost half. This segment of customers had an interest rate of almost double that of the entire dataset at 22.5%.

Another group with a large difference between interested and not interested were those who had vehicle damage and those who did not. Those with vehicle damage made up approximately half of the data while those without vehicle damage made up the second half. Customers who had vehicle damage were interested almost 24% of the time and those who did not have vehicle damage were interested less than 1% of the time. With the vehicle damage and previously insured explanatory variables both considered, a segment of the existing customer base was revealed that had an interest rate of 25%, as seen in Figure 1. This segment was made up of customers who had no vehicle insurance and had vehicle damage. It made up about 48% of all existing health insurance customers. The other 52% of existing health insurance customers had a rate of interest in vehicle insurance of 0.5%. By splitting the data in this way, the company could reduce the number of customers to cross sell to by over half.

Figure 1

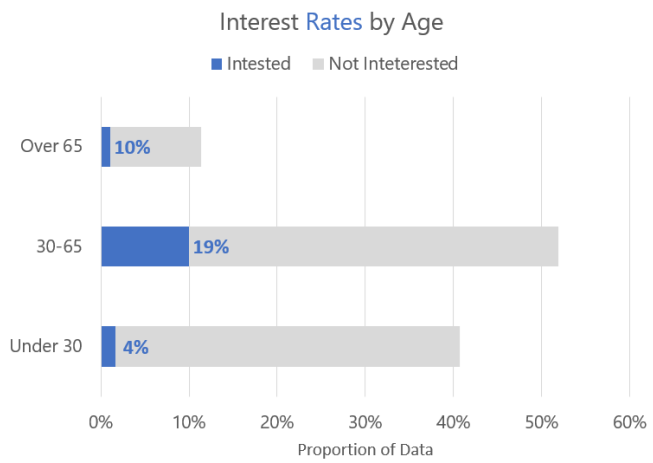
Interest Rate by Previously Insured and Vehicle Damage



Other useful information was also gathered from exploratory data analysis. When grouping customers by age, it can be seen that customers aged 30 through 60 had above average interest in vehicle insurance at 19%. This group represented 52% of all existing customers. Customers under 30, which represented 41% of existing customers, had low rates of interest in vehicle insurance at 4%. Finally, customers over 65 had marginal interest rates at 10%, but also only represented 11% of the data. This can be seen in Figure 2.

Figure 2

Interest Rates by Age



Predictive Modeling

Another method for identifying groups of existing health insurance customers that would be interested in vehicle insurance is predictive modeling. The output of a binary classification model assigns a probability to each record that describes the likelihood that the record is of a certain class according to the model. In this case, each customer is assigned a probability, according to the model, that they are interested in vehicle insurance. That probability is then used to make a prediction on whether or not the customer is interested.

Typically, accuracy is an intuitive measure for how well the model performs. For this data set, only 12.26% of the customers were actually interested in vehicle insurance. This means that if the model predicted that no customers were interested in vehicle insurance, it would have an accuracy of 87.74%. However, that would provide no value to the business since it would promote not cross selling. For the model to be useful, recall would need to be maximized. This means the model would have to predict that the customer was interested for as many customers who were actually interested as possible. The cost of missing out on a potential cross sell would be deemed much higher than soliciting a cross sell to a non-interested customer. A metric used for this type of unbalanced classification problem is AUC, which discourages choosing a model that is representational, but does a poor job of separating the target classes.

Predictive Modeling using PyCaret

There are countless machine learning models available that are applicable to a binary classification problem like this one. Some may have more success than others. A great starting point to figure out which machine learning model to use is PyCaret's compare_models function (Compare, n.d.). This function, coupled with PyCaret's automated preprocessing, quickly trains several machine learning models based on the data provided. Figure 3 shows the output from training 13 different models on the data.

Figure 3

PyCaret Model Comparisons

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ridge	Ridge Classifier	0.8772	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1460
ada	Ada Boost Classifier	0.8772	0.8516	0.0003	0.2936	0.0006	0.0004	0.0062	4.5680
gbc	Gradient Boosting Classifier	0.8772	0.8555	0.0001	0.3333	0.0002	0.0002	0.0047	17.0410
lightgbm	Light Gradient Boosting Machine	0.8769	0.8569	0.0030	0.3636	0.0060	0.0040	0.0233	1.4220
lda	Linear Discriminant Analysis	0.8768	0.8338	0.0027	0.3305	0.0053	0.0033	0.0198	0.4130
lr	Logistic Regression	0.8728	0.7329	0.0352	0.1974	0.0592	0.0389	0.0560	1.8200
rf	Random Forest Classifier	0.8663	0.8342	0.1201	0.3652	0.1807	0.1277	0.1515	17.0310
et	Extra Trees Classifier	0.8621	0.8268	0.1448	0.3514	0.2051	0.1437	0.1608	13.4700
knn	K Neighbors Classifier	0.8586	0.5955	0.0550	0.2109	0.0872	0.0383	0.0488	10.1820
svm	SVM - Linear Kernel	0.8380	0.0000	0.0782	0.1101	0.0485	0.0108	0.0144	6.0860
dt	Decision Tree Classifier	0.8224	0.6009	0.3071	0.2896	0.2981	0.1966	0.1967	0.9780
nb	Naive Bayes	0.8161	0.8139	0.3833	0.3034	0.3387	0.2336	0.2358	0.1960
qda	Quadratic Discriminant Analysis	0.6400	0.7483	0.8257	0.2234	0.3482	0.2150	0.2949	0.2940

From this output, the importance of not relying on accuracy can be illustrated. One of the top models, Ridge Classifier, had an accuracy of 87.72, but had an AUC of 0. That means regardless of where the threshold for the probability of a positive prediction was set, the classification would always be that the customer was not interested. The highest performing model in terms of AUC was the Light Gradient Boosting Machine. However, it can be seen that recall was only 0.3%. It is important to note that this recall is based on a probability threshold of 0.5. This means that if the model determined the probability of a positive, that the customer was interested, was above 0.5, the customer would be classified as interested. If the threshold was moved down, the recall would increase. For example, the same model with a probability threshold of 0.1 yielded a recall of 96% and a precision of 27%. So, the model found 96% of the interested customers, but when it predicted the customer would be interested, it was only correct 27% of the time. These results were promising, but could they be improved?

Modeling using a Neural Network

Another method for modeling cross sell opportunities that was explored was a neural network. Neural networks are a subset of machine learning and are a type of deep learning algorithm (IBM Cloud

Education, 2020). Since they have a reputation for being able to find complex patterns in data for accurate predictions, they were a natural next step.

Preprocessing. Without the use of PyCaret to preprocess the data automatically, these steps had to be completed manually. First, seven of the explanatory variables were exploded so the model would view them as categorical data. These variables included Gender, Driving_License, Region_Code, Previously_Insured, Vehicle_Age, Vehicle_Damage, and Policy_Sales_Channel. Two of these variables had numerical data, Region_Code and Policy_Sales_Channel, but they represented categories. For example, each numbered policy sales channel was a specific channel, as opposed to a continuous measurement. When the variables are exploded, that means that a variable with multiple possible values becomes multiple variables with two possible values; 0 for false, 1 for true. For example, if the new variable "Policy_Sales_Channel_132" had a 1, that would mean the customer was in policy sales channel 132. If it had a 0, the customer was not in that sales channel.

The next step in preprocessing the data was to scale the continuous data. This was an essential step to prevent a variable with a large range of values from becoming more important to the classification than it should be. There were three explanatory variables to scale; Age, Annual_Premium, and Vintage. Initially, the Age variable had a range of values from 20 to 85. After scaling the variable, the range was between 0 and 1. In doing this, each explanatory variable had a range between 0 and 1.

Splitting the Data. After the variables were processed, the resulting data set was randomly split between training, validation, and testing data. The training data represented 80% of all data and the test data represented 20%. The training data was then randomly split again so 20% of it was left over for validation data. When training any model, it is important to have data set aside that the model has not learned from to test on. This is especially important with neural networks, where the model can overfit, or memorize the data. The validation set is used to evaluate the model during each epoch, or round, of training. Then the final model is evaluated on the test set.

Oversampling. With an unbalanced dataset, where only 12.26% of the data represented customers who were interested, it was difficult for the neural network to learn the minority class. In this business case where we were needing the model to identify these customers, steps had to be taken to ensure the model learned the minority class. One such method is to oversample the data of the minority class; the customers who were interested in vehicle insurance. A way this can be accomplished is to use the Synthetic Minority Oversampling Technique, or SMOTE. Instead of simply duplicating examples from the minority class, which doesn't add any new information for the model to learn from, SMOTE synthesizes new examples from existing ones (Brownlee, 2020). This technique was used for one of the neural networks as well as the models produced by PyCaret.

Weighting. Another approach to the unbalanced data set problem is to use class weights. Since there are not as many interested customers for the model to learn from, we want the model to heavily weight the minimal number of examples that do represent interested customers. As a TensorFlow tutorial on unbalanced classification describes, class weights encourage the model to pay more attention to the under-represented class (Classification, 2021). For this data, a weight of 4.08 was given to customers who were interested and a weight of 0.57 was given to customers who were not.

Neural Network with Oversampling. For the neural network that utilized oversampling, a densely connected sequential architecture with four layers was constructed. This can be seen as a moderately shallow network. The evaluation metric used during training was AUC as it had already been illustrated that accuracy would be misleading. The model was trained for 1500 epochs, or rounds of training, during which it approached 0.91 AUC on the validation data. This was promising as it surpassed the AUC of the next best model. However, when evaluated on the test data, the AUC was only 0.73. A confusion matrix, which shows a breakdown of the classifications, along with other metrics for this model can be seen in Appendix B, Figure B1.

Neural Network with Weighting. Following this attempt at constructing a neural network, another network was constructed following the guidance from the TensorFlow tutorial mentioned previously. To prepare the data for this model, similar preprocessing steps were used. The main difference in preprocessing was that class weights were used instead of oversampling. The architecture for this model was also slightly different. Instead of 4 densely connected layers, the network architecture was composed of 3 layers and one dropout layer. A drop out layer is deemed one of the most effect and common ways to regularize a neural network, or prevent overfitting (Chollet, 2018). In other words, it helps to prevent the model from simply memorizing the training data. In the preceding attempt at fitting a neural network, the AUC dropped dramatically when testing on the unseen data. Therefore, it is plausible that the model was overfit to the training data. The last, and potentially largest, difference for this model was instead of oversampling the minority class, the class weights explained previously were fed into the network.

The results of this model can be found in Appendix B, Figure B2. This model performed much better on the test data yielding an AUC of 0.855. This was slightly under that of the Light Gradient Boosting Machine. A major difference between those two results was that with a probability threshold of 0.5, the neural network achieved a recall of 91% where the Light Gradient Boosting Machine had a recall of almost 0%. For the Light Gradient Boosting Machine, the model's probability that a customer would be interested in vehicle insurance was almost always below 0.5. In fact, only 14 of the tested 114,333 observations resulted in a probability greater than 0.5. The probabilities for the neural network had a range close to 0 to 1 where the Light Gradient Boosting machine had a range close to 0 to 0.5. However, both models were able to do a fairly good job of separating the interested customers from the non-interested customers. This range of probabilities for each model can be seen in Figure C1 and Figure C2 in Appendix C.

Potential Implementations

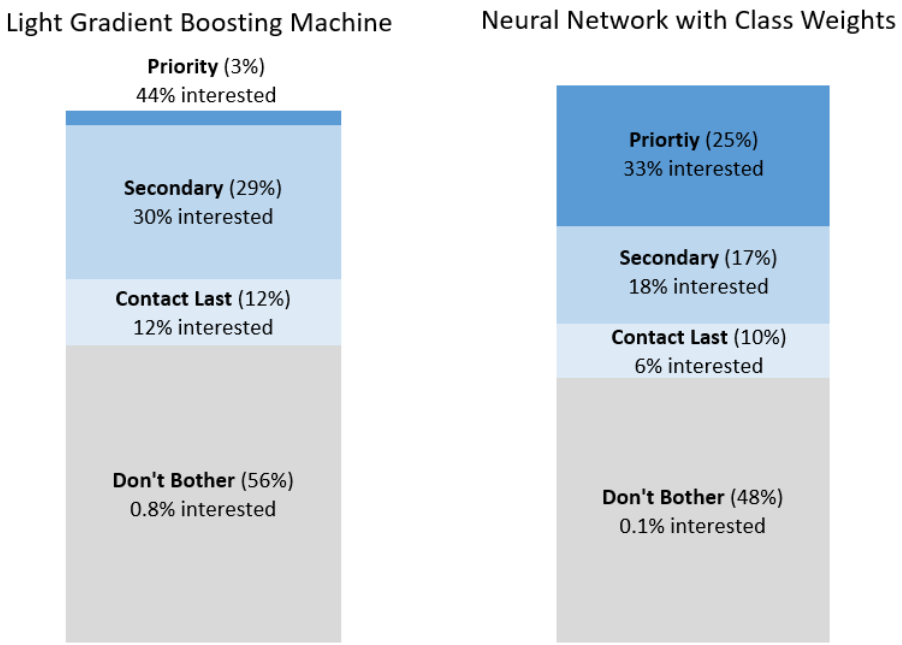
There are several different ways a company may go about using this data to improve their communication strategy for cross selling vehicle insurance. For example, an intuitive way to use a predictive model may be to simply only solicit the vehicle insurance to customers the model predicts as interested in vehicle insurance. However, according to a case study by Xpanse AI, this may not always be met with a high degree of welcome (Xpanse AI Team, n.d.). In the case study on cross selling opportunities that the team shared, it was explained that the company was reluctant to trust a model for identifying cross sell opportunities. The company did not want to miss out on potential revenue through false negatives, where the model may predict the customer was not interested but they in fact were. In the case study, the team devised a strategy of grouping the customers into larger subsets to be used in prioritization. The company was much more receptive to this format.

In order to apply that same methodology to this business case, existing customers could be grouped by the models' probability that the customer was interested. This can be seen for each model in Figures D1 and D2 in Appendix D. For the Light Gradient Boosting Machine this could mean creating three or four groups of customers. The first group could be customers where the model's probability was greater than 0.4. This group of customers contained 1308 interested customers out of 2935 total. Thus, there would be nearly a 45% chance of one of the customers in this group being receptive to a cross sell. The second group could be customers could be formed by model probabilities 0.2 to 0.4, which had a 30% rate of interest. The third group could be 0.1 to 0.2 which had a 14% rate of interest. Finally, the last group would have a 1% rate of being interested in vehicle insurance. Similarly, we can form four groups of customers with the probabilities provided by the neural network with class weights. The first group could be probabilities greater than 0.7 which had an interest rate of 33%. The second group could be for probabilities between 0.4 and 0.7 which had an interest rate of 18%. The third group could be for probabilities between 0.1 and 0.4 which had an interest rate of 6%. Finally, the last group could be for probabilities below 0.1 which had a rate of interest of less than 1%.

To communicate these groupings to the business, a stacked bar chat can be used as shown in Figure 4. The stacked bar chart has the advantage of illustrating both the size of the customer segment as well as the prioritization. The top group for each chart would represent customer segments that should be the business's top priority. These are customers who are the most likely to be interested in vehicle insurance. The second segment represents customers who should be second priority. There are interested customers in this segment, but not at the rate of the priority segment. The third segment represents customers with relatively low likelihood of being interested in vehicle insurance. It could be communicated to the business that these customers should be contacted last, if at all. Finally, the last customer segment is communicated as a group of customers the company should not waste their time with. The rate of interest in vehicle insurance is extremely low.

Figure 4

Customer Groupings by Model Probability



The final question to be answered is which model would be best? The answer to that question depends on how many customers the business is willing to solicit vehicle insurance to. For example, if the business is willing to solicit vehicle insurance to the top three groups, they would identify 99% of the

interested customers for the neural network and 96% for the Light Gradient Boosting Machine. However, for the neural network, they would have to reach out to 52% of all existing customers to achieve this recall. For the Light Gradient Boosting Machine, they would only need to reach out to 44% of all existing customers to achieve the 96% recall. In the end, which model the business chooses to use and how the customers are grouped would depend on how many customers the business planned on soliciting vehicle insurance to. Both models are relatively similar in their ability to separate the interested and non-interested customers, with the Light Gradient Boosting Machine having a slight edge in terms of AUC.

Further Analysis

Although this is a great start to solving the problem of identifying existing health insurance customers who are interested in vehicle insurance, there are still some other methods that could be explored. One in particular would be a decision tree for customer segmentation. In the exploratory data analysis section, it was shown how groupings of categories within one or two variables could reveal customer segments with extremely high or extremely low rates of interest. These segments were easily interpretable. For example, it was shown that customers who had vehicle damage and did not have insurance currently represented almost half of the existing health insurance customers and had an interest rate of 25%. To contrast, the probability groupings from the models revealed customer segments that were much more complicated to describe since all variables were considered. A decision tree algorithm would have a clear path for separating the data like the exploratory data analysis, but it would allow for more depth in the decision making. Groups would be formed using more than just two variables.

Lastly, more business information would be required to fully evaluate the solution to the problem. Information surrounding the costs associated with the cross-sell communication strategy could be leveraged to refine the model. For example, if the cost of a false positive, when the model predicts

the customer is interested when they are not, and the cost of a false negative, when the model predicts the customer is not interested when they are, was known, the model could be penalized proportionally for different types of errors. This would allow a model to be trained that would optimize cost savings for the company. For this project, it was only assumed through research that the cost of a false positive was much less than the cost of a false negative. In other words, missing out on a potential customer would cost the company more money than soliciting vehicle insurance unsuccessfully. However, the exact cost was unknown and would vary depending on the business and their communication strategy.

References

Brownlee, J. (2020, January 17). *SMOTE for Imbalanced Classification with Python*. Machine Learning Mastery. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co.

Compare Models. (n.d.). PyCaret. Retrieved June 16, 2021, from https://pycaret.org/compare=models/

Classification on imbalanced data. (2021, June 17). TensorFlow. Retrieved June 18, 2021, from https://www.tensorflow.org/tutorials/structured_data/imbalanced_data

Executing a Successful Insurance Cross Sell Strategy. (2020, July 14). Goose Digital. Retrieved June 26, 2021, from https://goosedigital.com/media-types/articles/executing-a-successful-cross-sell-strategy/

Hayes, A. (2021, May 2). *Cross-Sell*. Investopedia. https://www.investopedia.com/terms/c/cross-sell.asp

IBM Cloud Education. (2020, August 17) *Neural Networks*. IBM. https://www.ibm.com/cloud/learn/neural-networks

Kumar, A. (2020, September 11). *Health Insurance Cross Sell Prediction*. Kaggle. https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction/

T, D. (2019, July 25). *Confusion Matrix Visualization*. Medium. https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea

Xpanse AI Team. (n.d.). *Predictive Analytics in Marketing – Case Study 2: Cross-Sell for Insurance*. Xpanse AI. https://xpanse.ai/case-studies/predictive-analytics-in-marketing-case-study-2-cross-sell-for-insurance/

Appendix A

Variable	Type	Description
id	Integer	unique identifier
Gender	Categorical (Male, Female)	Gender of customer
Age	Integer	Age of customer
Driving_License	Boolean (0, 1)	0 – Customer does not have a driver's license 1 – Customer has a driver's license
Region_Code	Integer	Unique code for region of customer
Previously_Insured	Boolean (0, 1)	0 – Customer doesn't have vehicle insurance 1 – Customer does have vehicle insurance
Vehicle_Age	Integer	Age of vehicle
Vehicle_Damage	Boolean (0, 1)	0 – Customer didn't get vehicle damaged in past 1 – Customer did have vehicle damaged in past
Annual_Premium	Integer	Annual premium customer pays for health insurance
Policy_Sales_Channel	Integer	Anonymized code for channel of outreach to customer
Vintage	Integer	Number of days customer has been associated with insurance company
Response	Boolean (0, 1)	0 – Customer is interested 1 – Customer is not interested

Appendix B

Modeling Results

```
AUC:  0.7277058766481167
Accuracy:  0.7827136522263913
Precision:  0.3145559210526316
Recall:  0.6548255938369356
```

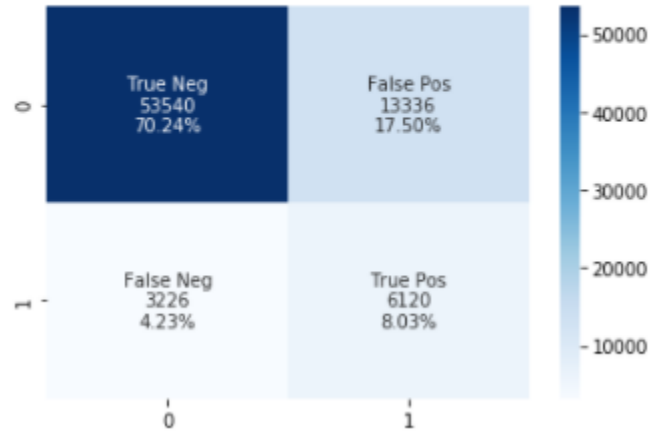


Figure B1. Results from Neural Network fit after using SMOTE. This confusion matrix plot was created referencing an article found on Medium by Dennis T (T, 2019).

```
accuracy :  0.70122796
precision :  0.2815575
recall :  0.92269087
auc :  0.85537124
prc :  0.36233035

Non-Interested Customers Correctly Classified (True Negatives):  44808
Non-Interested Customers Incorrectly Classified (False Positives):  22049
Interested Customers Incorrextly Classified (False Negatives):  724
Interested Customers Correctly Classified (True Positives):  8641
Total Interested Customers:  9365
```

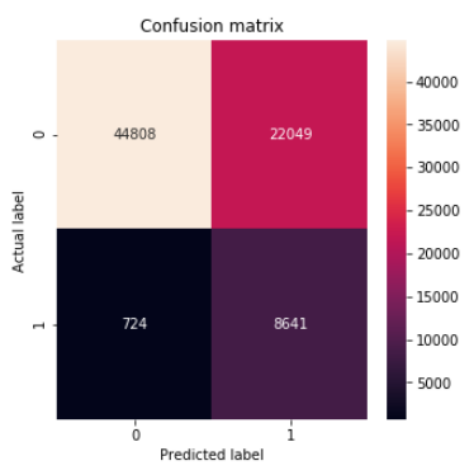


Figure B2. Results from Neural Network fit after using class weighting.

Appendix C

Interested and Non-Interested Customers by Model Probability Range

Probability	Interval Total	Interested	Not Interested
0.0-0.1	64094	544	63550
0.1-0.2	14202	2059	12143
0.2-0.3	15384	3910	11474
0.3-0.4	17718	6123	11595
0.4-0.5	2921	1302	1619
0.5-0.6	14	6	8
0.6-0.7	0	0	0
0.7-0.8	0	0	0
0.8-0.9	0	0	0
0.9-1.0	0	0	0

Figure C1. Interested and Non-Interested customers binned by the Light Gradient Boosting Machine model's probability of that the customer was interested.

Probability	Interval Total	Interested	Not Interested
0.0-0.1	36212	58	36154
0.1-0.2	2184	62	2122
0.2-0.3	2583	138	2445
0.3-0.4	2752	237	2515
0.4-0.5	2983	325	2658
0.5-0.6	3734	608	3126
0.6-0.7	6511	1428	5083
0.7-0.8	15702	4998	10704
0.8-0.9	3561	1460	2101
0.9-1.0	0	0	0

Figure C2. Interested and Non-Interested customers binned by the neural network using class weights model's probability of that the customer was interested.

Appendix D

Customer Segmentation Using Model Prediction Probabilities

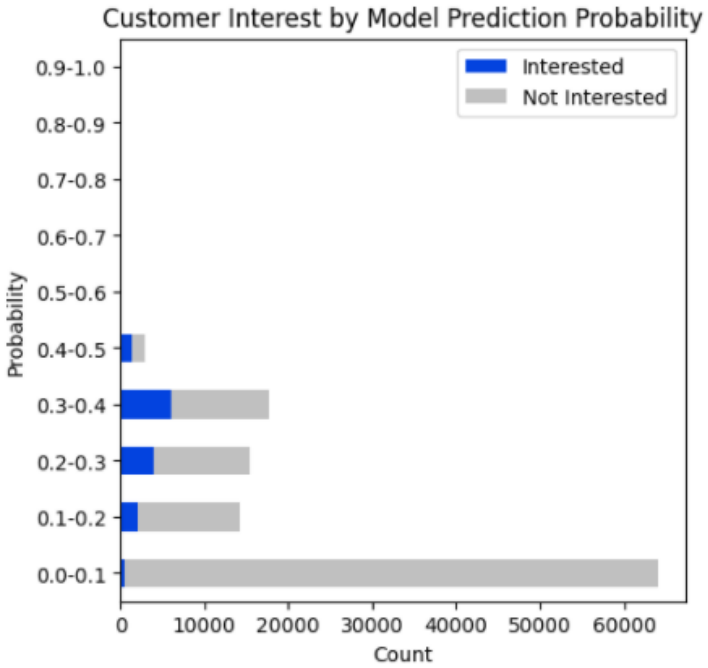


Figure D1. Customer segmentation using the probability from the Light Gradient Boosting Machine.

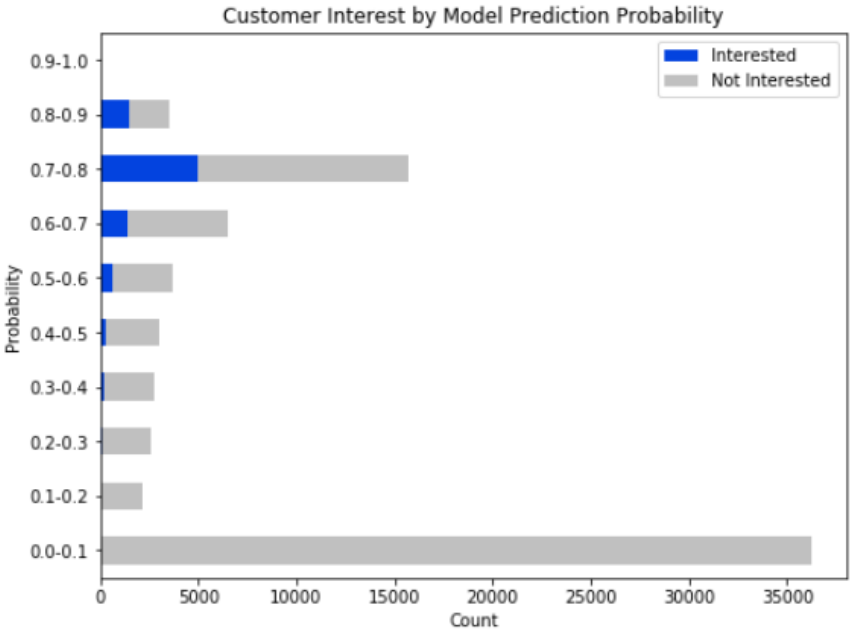


Figure D2. Customer segmentation using the probability from the neural network using class weights.

Expected Questions

Does the probability given by the model equal the likelihood the customer is interested in vehicle insurance?

Not necessarily. The probability can be seen as how confident the model is that the customer is interested in vehicle insurance. A particular probability can mean different things for different models on the same set of data. Refer to Appendix D for an illustration of this.

How can customer segments be used by the business?

The business can decide not to solicit vehicle insurance to customers that fall within segments that have low rates of interest. In doing so, the company could reduce costs associated with soliciting vehicle insurance.

How would the business determine the customer's interest in vehicle insurance for training the model?

The business would have to randomly sample the existing health insurance customer base and retrieve information on whether the customers were interested in vehicle insurance or not.

How large would the random sample need to be for the training data?

If the business is already reaching out to all existing customers randomly for cross selling vehicle insurance, then the results of those interactions could be used for training. Otherwise, the larger the sample, the more representational it would be and the better the model would perform.

Could the customers be segmented differently by model probability?

Yes. The groupings do not need to be in 10% increments like the tables in Appendix C either. Depending on how many customers the business is wanting to reach out to, optimal segmentation can be achieved.

Could customers simply be ranked according to the model's probability?

Yes. This is another way you could utilize the output from the model.

Would the model need to be retrained regularly?

Potentially. If the observed interest rates for the segments change in the future, that could indicate that the distribution has changed and the model would need to be retrained on more recent data. For example, if a customer segment was expected to have an interest rate of 44% and after a year the interest rate dropped to 20%, there would be evidence for a need to retrain.

How long does training the model take?

Training both models on the data from this project took less than an hour. The majority of the time was spent testing different models and different pre-processing methods. These tasks would be streamlined for future training iterations.

Do other companies use this method?

Yes. Refer to the case study posted by the Xpanse AI Team for an example of a successful implementation of this customer segmentation method (Xpanse AI Team, n.d.).

Could the models be improved upon?

Absolutely. There are plenty of models that were not tested for this project. Beyond that, there are countless ways to tune the models to perform better.