Health Insurance Cross Sell Prediction

I. INTRODUCTION

Machine learning algorithms have a long history with the classification problem which has been used extensively for a variety of applications. In our project, we attempt to use the tools of machine learning to identify whether the customers who bought health insurance from our company last year will also be interested in Vehicle Insurance provided by the company. The aim of this project is to analyze the performance of various classification algorithms and quantify their suitability for this problem.

II. METHODOLOGY

1. Business understanding

1.1 The problem:

Our client is an Insurance company that has provided Health Insurance to its customers. Now they need your help in building a model to predict whether the policyholders (customers) from past year will also be interested in Vehicle Insurance provided by the company.

An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee.

Just like medical insurance, there is vehicle insurance where every year a customer needs to pay a premium of a certain amount to the insurance provider company so that in case of an unfortunate accident by the vehicle, the insurance provider company will provide a compensation (called 'sum assured') to the customer.

1.2 Project Goal:

Building a model to predict whether the customers who bought health insurance from our company last year will also be interested in Vehicle Insurance provided by the company.

1.3 Criteria for successful project:

Building a model to predict with acceptable accuracy for the classes (0 and 1). The metric for this problem is Accuracy, F1, recall and precision score to see how well our models perform on both classes.

A good model should achieve an accuracy score (preferably 90% or higher), and should have high F1 score, precision, and recall score as high as possible based on our dataset (if we encounter an imbalanced dataset)

Regarding recall and precision score in our problem. The model should have a higher precision score than recall if possible, high precision score means in our positive predictions, there should be a high percentage of true class 1. While recall could lead to a low percentage of true class 1 in our predictions. This could lead to the company wasting money and resources on a high number of false positive clients.

1.3 Analytics Approach: Classification model

2. Data Understanding

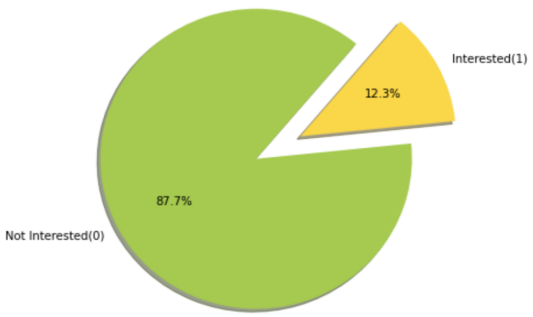
2.1 Data Description

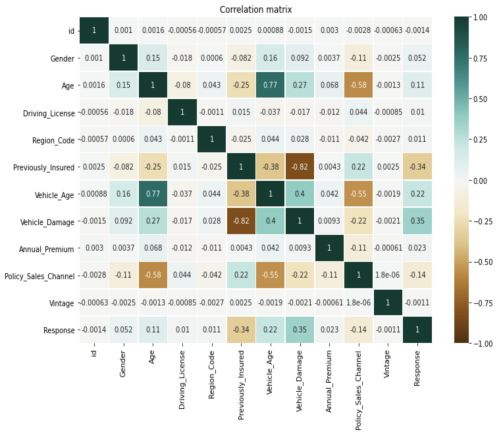
This project relates to the Vehicle Insurance data set, which can be found in the file train.csv in the data folder. It contains a number of variables for 381109 different ID customers in a company. The variables are:

- **ID:** Unique ID for the customer
- **Gender:** Gender of the customer: 1: Male 0: female
- **Age:** Age of the customer
- **Driving_License:** Customer does not have DL(0), Customer already has DL(1)
- **Region_Code:** Unique code for the region of the customer
- **Previously_Insured:** Customer already has Vehicle Insurance (1) , Customer doesn't have Vehicle Insurance (0)
- **Vehicle_Age:** Age of the Vehicle
- **Vehicle_Damage:** Customer got his/her vehicle damaged in the past (1), Customer didn't get his/her vehicle damaged in the past (0)
- **Annual_Premium:** The amount customer needs to pay as premium in the year
- **Policy_Sales_Channel:** Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
- **Vintage:** Number of Days customer has been associated with the company
- **Response:** Customer is interested (1), Customer is not interested (0)

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
0	1	Male	44	1	28.0	0	> 2 Years	Yes	40454.0	26.0	217	1
1	2	Male	76	1	3.0	0	1-2 Year	No	33536.0	26.0	183	0
2	3	Male	47	1	28.0	0	> 2 Years	Yes	38294.0	26.0	27	1
3	4	Male	21	1	11.0	1	< 1 Year	No	28619.0	152.0	203	0
4	5	Female	29	1	41.0	1	< 1 Year	No	27496.0	152.0	39	0

2.2 Exploratory Data Analysis:





Category Ratio: The dataset has a large difference in the category ratio: 87.7% Not interested, 12.3% Interested

Outliers: A quick look of our statistics, we can see that Age and Annual_Premium has outliers with max value exceeds 1.5*IQR (the highest age is 85 and the highest annual_premium is 540k rupees). Other fields don't have outliers.

Correlation: From the heatmap we can see how these features affect each other and affect the response variable. The score ranges from -1 to 1 and indicates if there is a strong linear

relationship — either in a positive or negative direction. When the score is close to 0, saying: “Nothing interesting here”. The correlation matrix is symmetric which means that the correlation is the same whether you calculate the correlation of A and B or the correlation of B and A.

Based on the correlation matrix, we performed deep mining for variables that are correlated with the response variable. There are the interesting things we discovered:

- **Vehicle damage:** People who have vehicle damage are more interested in our vehicle insurance (23.7%).

- **Previous_Insured:** People who did not have Vehicle insurance more interested in our Vehicle insurance (22.5%)

- **Vehicle Age:**
    + Most of the people interested in our insurance have used vehicles for 1-2 years or over 2 years. Their interest in Vehicle insurance at 17.3% and 29.3% respectively.
    + Only 4.3% of the customers who have used the vehicle for less than 1 year are interested in Vehicle insurance out of 43.2% of the total people who have used the Vehicle for less than 1 year.

- **Policy Sales Channel:** Channels from 145 - 160 reach the most customers, but only 1/5 (20%) customers who reached are interested in our insurance, other channels about 50% customers interested in Vehicle insurance

- **Customer's Age:**
    + The customer group in the age group of 20 - 30 accounts the most in the dataset, but the level of interest in Vehicle insurance is only 20%.
    + About 67% of customers in the age group of 30 - 60 and 50% of customers in the age group of 60 - 80 are interested in Vehicle insurance.

- **Customer's Gender:** There is no big difference in the distribution of customers' gender in the dataset. Male tend to be slightly more interested in Vehicle insurance than Female (Male: 13.8%, female: 10.3%)

- **Annual_Premium:** People don't pay much for Insurance. The annual premium is distributed mostly in the range of 2630 - 100000 VND

3. Data Preparation:

“Good features make good models”, this is actually the most important and time-consuming step. According to some insights from the dataset, we apply several techniques and algorithms to preprocess dataset, include:

- **Drop Outliers:** Remove outliers from Annual Premium and Age column.
- **Resample:** Handle imbalanced dataset with Up-Sampling technique
- **Feature Selection:** Remove 'id', 'Driving_License', 'Region_Code', 'Vintage' column due to these features almost don't have any correlation with our target field.
- **Normalize features:**
- **Split Dataset:** Split dataset into 3 set: train set, test set, validation set

4. Modeling & Evaluation

4.1 K-nearest neighbors

KNN (K — Nearest Neighbors) is one of many (supervised learning) algorithms used in data mining and machine learning, it's a simple classifier algorithm where the learning is based on "how similar" is a data (a vector) from others .

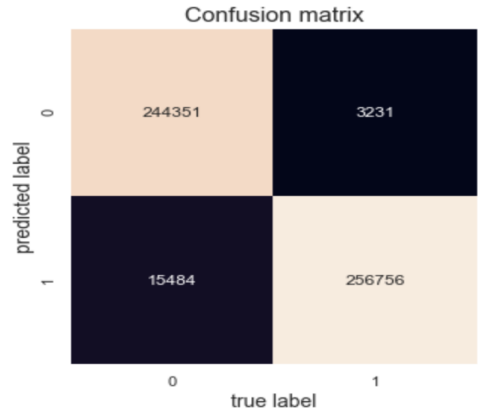
KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification).

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

In the case of classification and regression, we saw that choosing the right K for our data is done by trying several Ks and picking the one that works best. We have done this by using Gridsearch which is a very useful algorithm for estimating parameters when training.

4.2 Random Forest




```
Accuracy_score: 96.4
Precision_score: 98.8
Recall_score:  94.3
F1_score: 96.5
```

Random forest is an ensemble method for classification that operates by constructing a set of decision trees and outputting the class that is the mode of the classes output by individual trees. Unlike a single decision tree, which is likely to suffer from high Variance or high Bias, Random Forests can find a natural balance between the two extremes. For each tree in the forest, we construct our training set by sampling with replacement within the training instances (bootstrap sampling)

The following figure shows the train accuracy with 200 decision trees in the forest. We can see the accuracy reach the limit about 96%.
Bagging and other resampling techniques are usually used to reduce the variance in ensemble methods. To make a prediction, all of the models in the ensemble are polled and their results are averaged. Random Forests works by training numerous decision trees each based on a

different resampling of the original training data, in which numerous replicates of the original data sets are created.

4.3 Catboost

CatBoost is an algorithm for **gradient boosting on decision trees**. Developed by Yandex researchers and engineers. During training, a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to the previous trees.

CatBoost trains our model on GPU gives a better speedup when compared to training the model on CPU. This feature enables CatBoost to learn faster and make predictions 13-16 times faster than other algorithms.

CatBoost can improve the performance of the model while reducing overfitting and the time spent on tuning. The default settings of the parameters in CatBoost would do a good job. it produces good results without extensive hyperparameter tuning. In this project, we tuned depth, iteration and learning_rate in CatBoost to get a better result. We also use Gridsearch to estimate the best parameters as Random Forest and K-neighbors model.

4.4 Soft Voting

The idea behind the VotingClassifier is to combine conceptually different machine learning classifiers and use a majority vote or the average predicted probabilities to predict the class labels. Such a classifier can be useful for a set of equally well performing models in order to balance out their individual weaknesses.

In contrast to majority voting (hard voting), soft voting returns the class label as argmax of the sum of predicted probabilities. Every individual classifier in soft voting provides a probability value that a specific data point belongs to a particular target class. The predictions are weighted by the classifier's importance and summed up. Then the target label with the greatest sum of weighted probabilities wins the vote.

For this project we train our classification problem with 2 different algorithms — *K-nearest neighbors, Random Forest ,* then using the soft voting to balance these individual weaknesses

4.5 Result

	Model	Accuracy_training	Accuracy_testing	F1_scores_training	F1_scores_testing
0	KNN	0.95	0.89	0.95	0.89
1	RandomForest	0.96	0.92	0.96	0.92
2	catboost	0.81	0.81	0.83	0.83
3	SorfVoting	0.96	0.90	0.96	0.90

We present different models built by multi-class supported K-nearest neighbors, Random Forest, CatBoost and Voting . With each model, we used Gridsearch to get the best parameters and tune it when the results are not as good as expected. The performance of the various models in predicting the interest of our customers on our vehicle insurance is tested and presented above.

We can see that the random forest model outperforms other algorithms with a train accuracy of 96% and test accuracy of 92%. Its F1 score is as high as accuracy score on both training set and test set.
The Precision score of all models larger than recall score , it ranges from 93% to 98%. This will help the company avoid wasting money and resources on a high number of false positive clients.

III. CONCLUSION

Based on the performance on the training set and test set using different methods, we can see the best performing model is Random Forest. It has 96% accuracy and 96% weighted-average F1 score on training set. Therefore we decided to use Random Forest model.

Random Forest is a little bit overfitted because it has a higher difference between training set and testing set (0.96 vs 0.92), but with 96% True Positive precision, it obviously will boost conversion ratio after this model is implemented in the company. It will boost sales/marketing team performance because now they know which customer to be targeted (The Predicted Yes Response).

IV. REFERENCES

1. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms*; Taylor & Francis: Abingdon, UK, 2012. [Google Scholar]
2. Schwenker, F.; Trentin, E. Pattern classification and clustering: A review of partially supervised learning approaches. *Pattern Recognit. Lett.* 2014, *37*, 4–14. [Google Scholar] [CrossRef]