Vehicle Insurance Predictor

Shaili Gandhi
AU1841012
Ahmedabad University
Ahmedabad, India
shaili.g@ahduni.edu.in

Priyank Sangani
AU1841136
Ahmedabad University
Ahmedabad, India
priyank.h@ahduni.edu.in

Yash Arvindkumar Patel
AU1841141
Ahmedabad University
Ahmedabad, India
yash.p4@ahduni.edu.in

Samarth Shah
AU1841145
Ahmedabad University
Ahmedabad, India
samarth.s@ahduni.edu.in

***Abstract*—As Machine Learning is used for data analysis and finding patterns in the data, here we would be using it to predict whether a particular customer will be interested in buying the Vehicle Insurance provided by the Insurance company. We are using Logistic Regression to solve this problem. We started with data pre-processing, data cleaning and data visualisation. Later we tried to fit data in logistic regression model and make the required prediction.**

***Index Terms*—Classification, Insurance, Vehicle, Prediction, Performance**

I. Introduction / Motivation / Background

Building a model to predict whether a customer would be interested in Vehicle Insurance is extremely helpful for the company because it can then accordingly plan its communication strategy to reach out to those customers and optimise its business model and revenue. Here, the data of the customers are provided as follows:

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

II. Mathematical Analysis

We took help of normalization for preprocessing There has been some mathematical equations which have used for data cleaning i.e. for data Normalization. After that we need to design a mathematical intuition for classification algorithms. Till now we understood the mathematical intuition for logistic regression.

A. Maths for Data cleaning and visualization

Normalization is required for the columns that are not in range of $0-1$, as logistic regression is classified in $0-1$, so we need to modify our data accordingly.
There are four columns Region Code, Annual Premium, Policy Sales Channel and Vintage.

- For Regional Code and Vintage, we can see that 2,3 values have most of the values and rest are nearly uniform and less that is why we choose to normalize it.

$$Normal = \frac{Actual - minimum}{maximum - minimum}$$

- For Annual Premium, we can draw inference that data is highly skewed towards left.
- For Policy Sales Channel, the graph is similar to uniformly distributed and thus we can apply normalization to that column.

B. Maths for Logistic Regression

For logistic regression we tried to carry out the sigmoid function for hypothesis and then calculate loss and using gradient descent we tried to pull out the optimal theta parameters.

$$hypothesis = \frac{1}{1+e^{-(\theta^T X)}}$$

$$Loss = -\frac{1}{m}((y)log(h) + (1-y)log(1-h))$$

Here for loss function, if the actual label is 0, then the second term remain and and if the label is 1 then only first term will remain and accordingly we had subtracted the hypothesis from 1.
Then we can go for gradient descent to minimize loss.

$$\theta^{new} = \theta^{old} - \alpha * \frac{\partial loss}{\partial \theta}$$

Here alpha is the learning rate which defines the rate which our model will learn. For the scratch part we had taken 0.001 alpha but it has not yet working and we are implementing it.

III. Experiments and Results

- First, we have converted categorical data into dummy variables using get_dummies() function.

```
v_age = pd.get_dummies(data.Vehicle_Age, prefix_sep='_')
v_damage = pd.get_dummies(data.Vehicle_Damage, prefix_sep='_')
gender = pd.get_dummies(data.Gender, prefix_sep='_')
print(v_age.head())
print(v_damage.head())
print(gender.head())
```

```
   1-2 Year  < 1 Year  > 2 Years
0         0         0          1
1         1         0          0
2         0         0          1
3         0         1          0
4         0         1          0
   No  Yes
0   0    1
1   1    0
2   0    1
3   1    0
4   1    0
   Female  Male
0       0     1
1       0     1
2       0     1
3       0     1
4       1     0
```

- Then, normalized the data and converted them in the range of 0 to 1.
- Below image is before normalization

```
[ ] data.head()
```

	Age	Driving_License	Region_Code	Previously_Insured	Annual_Premium	Policy_Sales_Channel	Vintage
0	44	1	28.0	0	40454.0	26.0	217
1	76	1	3.0	0	33536.0	26.0	183
2	47	1	28.0	0	38294.0	26.0	27
3	21	1	11.0	1	28619.0	152.0	203
4	29	1	41.0	1	27496.0	152.0	39

- Below image is after normalization

```
data.head()
```

_yes	damage_no	gender_F	gender_M	Annual_Premium	Region_Code	Vintage	Policy_Sales_Channel	Age
1	0	0	1	0.070366	0.538462	0.716263	0.154321	0.369231
0	1	0	1	0.057496	0.057692	0.598616	0.154321	0.861538
1	0	0	1	0.066347	0.538462	0.058824	0.154321	0.415385
0	1	0	1	0.048348	0.211538	0.667820	0.932099	0.015385
0	1	1	0	0.046259	0.788462	0.100346	0.932099	0.138462

- Now, we have find the confusion matrix for normalized data as well as normal data using confusion_matrix() function.
- This functions return a 2x2 matrix where
  Matrix[0][0] = true negative
  Matrix[1][0] = false negative
  Matrix[1][1] = true positive
  Matrix[0][1] = false positive

- Below image shows the confusion matrix for normal data.

```
Confusion Matrix:  [[100382      0]
 [ 13951      0]]
```

- Below image shows the confusion matrix for normalized data.

```
Confusion Matrix:  [[100220      4]
 [ 14108      1]]
```

- Now calculating the accuracy, precision and recall of the model. Precision will tell us how how many predicted positives got true and recall will give how many positives are got detected by the model.
- Below image shows the accuracy, precision and recall of the model when data is not normalized.

```
Accuracy Score:  0.8779792360910673
Precision Score:  0.0
Recall Score:  0.0
```

- Below image shows the accuracy, precision and recall of the model when data is normalized.

```
Accuracy Score:  0.876571068720317
Precision Score:  0.2
Recall Score:  7.0876745339854e-05
```

- As we can see that values of False Negative is very large as compared to True Positives we can conclude that model is highly predicting 0's and not 1's and seems like model is highly biased towards Negatives.

IV. Conclusion and Future Works

- We had completed a logistic regression model with the use of libraries with 87 percent accuracy shown above. But we are struggling with the values of confusion matrix, it seems model is predicting more of 0's and looks like a bias model.
- We will work on the bias problem as well as we are working on logistic regression from scratch, as we are struggling with the loss of that model we had not attached the results here.
- After logistic regression we would like to work with Support Vector Machine from scratch and try to compare the results with that of logistic regression.
- We would also like to explore more classification models such as decision tree but only if time permits.

References

[1] Wang, Hui Dong. "Research on the Features of Car Insurance Data Based on Machine Learning." Procedia Computer Science 166 (2020): 582-587.

[2] Grize, Yves-Laurent, Wolfram Fischer, and Christian Lützelschwab. "Machine learning applications in nonlife insurance." Applied Stochastic Models in Business and Industry 36.4 (2020): 523-537.

[3] Mane, Sandeep and Srivastava, Jaideep and Hwang, San-Yin and Vayghan, Jamshid,2004 12,475- 478,Estimation of false negatives in classification, 0-7695-2142-8, 10.1109/ICDM.2004.10048.

[4] https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

[5] https://towardsdatascience.com/understanding-logistic-regression-step-by-step-704a78be7e0a