TAG

Tampere University

MACHINE LEARNING PROJECT

Logistic Regression vs Artificial Neural Network

Jan Rebek

November 2020
Tampere University of Applied Sciences

Table of Contents

1. INTRODUCTION

1.1 Topic

The topic of the project is implementing machine learning models that will be used to get suitable predictions for a classification case. The contexts of the project is that an imaginary insurance company (client) that had provided health insurance to its customers now need help in building models to predict whether the policyholders (customers) from past year will also be interested in vehicle insurance provided by the company.

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

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 optimize its business model and revenue.

1.2 Dataset

In order to predict, whether the customer would be interested in vehicle insurance, there is information about demographics (gender, age, region code type), vehicles (vehicle age, damage), policy (premium, sourcing channel) etc. The currency is Indian Rupee.

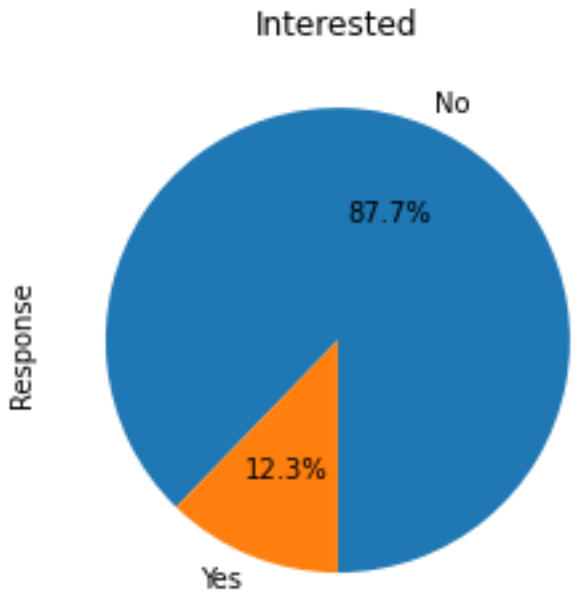
Variable	Definition
id	Unique ID for the customer
Gender	Gender of the customer
Age	Age of the customer
Driving_License	0 : Customer does not have DL, 1 : Customer already has DL
Region_Code	Unique code for the region of the customer
Previously_Insured	1 : Customer already has Vehicle Insurance, 0 : Customer doesn't have Vehicle Insurance
Vehicle_Age	Age of the Vehicle
Vehicle_Damage	1 : Customer got his/her vehicle damaged in the past. 0 : Customer didn't get his/her vehicle damaged in the past.
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	1 : Customer is interested, 0 : Customer is not interested

2. DATA ANALYSIS

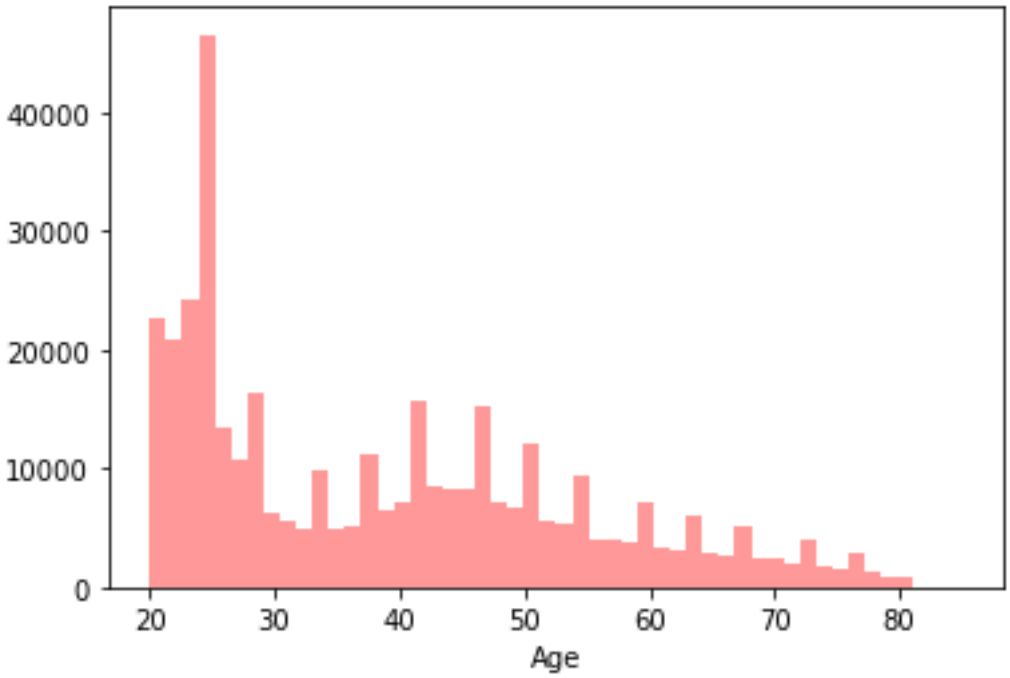
After analyzing the dataset, the columns id, Region_Code and Policy_Sales_Channel were discarded from the data frame because of either uselessness or the risk that they could prevent the model from learning properly.

3. DATA VISUALIZATION



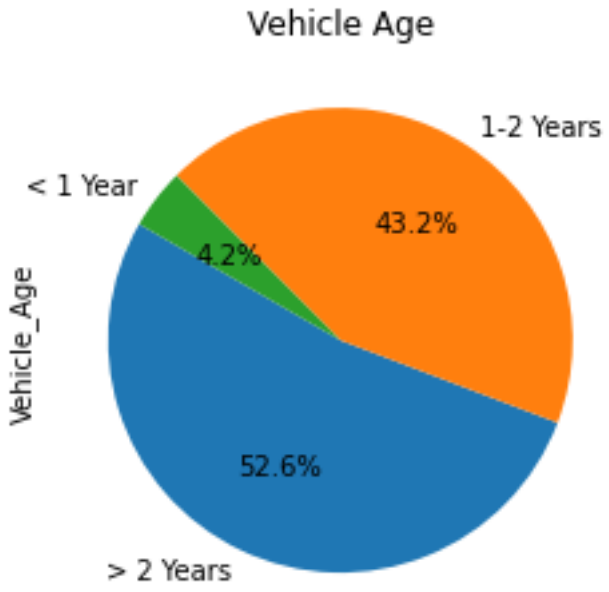
Percentage of people interested in vehicle insurance

A visualization shows that majority of the policyholders (87.7%) from previous years are not interested in vehicle insurance purchase. Only 12.3% of the customers is interested in purchasing vehicle insurance.



Age of customers

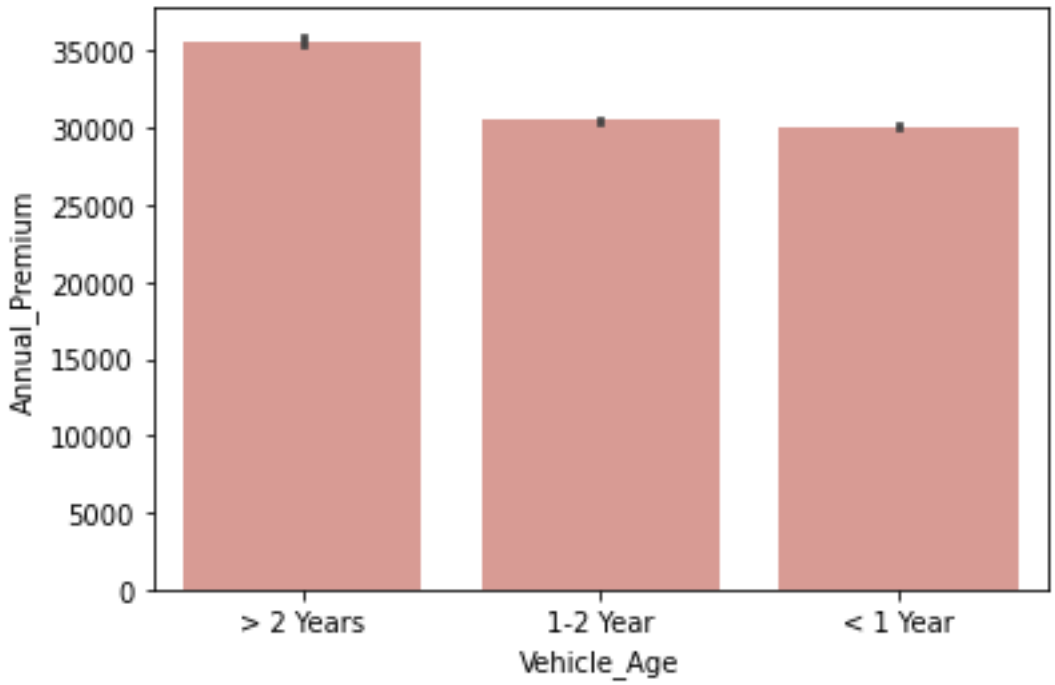
The largest age group of customers is between 20 and 30 years old. The second largest group is customers until approximately 50 years old. In the older age groups, the higher the age the smaller number of customers.



Vehicle age of cars

Most customers have cars that are at least one year old. 52.6% of the customers have a car that is more than 2 years old whilst 43.2% of the customers have a car that is 1-2

years old. The smallest group is the customers that own a car that is less than a year old. This group is only 4.2%.



Annual Premium compared with Vehicle Age

Visualized data shows that owners of cars older than 2 years pay the highest annual premium from the three analyzed group. The difference in annual premium between the owners of cars that are between 1 and 2 years or and owners whose cars are less than 1 year old is insignificant.

4. APPLYING MACHINE LEARNING MODELS

4.1 Logistic Regression

Initially, appropriate libraries were imported, then an appropriate variable was initiated using the CSV file and a visualization of data was made.

Then, the data was trimmed by excluding unnecessary columns and keeping only those that are meaningful for training the model. Then the data was split into training and testing sets. The columns containing gender, vehicle damage and vehicle age were transformed to get proper categories for the model to train.

```
# make X and y

X_columns = ['Gender', 'Age', 'Driving_License', 'Previously_Insured', 'Vehicle_Age', 'Vehicle_Damage', 'Annual_Premium', 'Vintage']
X = df.loc[:, X_columns]
X_original = X
y = df.iloc[:, 11:12]
y = y.values


# Splitting the dataset into the training and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# get categories using ColumnTransformer and OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(drop='first'), ['Gender', 'Vehicle_Damage', 'Vehicle_Age'])], remainder='passthrough')
X_train = np.array(ct.fit_transform(X_train)) # First column is Gender, second is Vehicle_Damage, third and fourth are about Vehicle_Age
X_test = np.array(ct.fit_transform(X_test)) # First column is Gender, second is Vehicle_Damage, third and fourth are about Vehicle_Age

```

Then it is time to train the model using logistic regression. When the model finished the training, it can be then used to predict the results using the test set.

```
#Training the Logistic Regression model on the training set
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(random_state = 0)
model.fit(X_train, y_train)

# predicting the Test set results
y_pred = model.predict(X_test)

```

4.2 Results of using Logistic Regression

For assessing on how well the model did a confusion matrix and accuracy scored were used. The confusion matrix result was:

[[66831 15]
[9368 8]]

There is a large quantity of true positives, however a significant number of false negative is worrying and suggest the problem with the dataset.

The accuracy score was 0.8768990580147464

```
# Making the Confusion Matrix and accuracy score
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)

print ('cm:')
print(cm)
print(f'accuracy_score: {accuracy_score(y_test, y_pred)}')
```

4.3 Artificial Neural Network

To build artificial neural network model the same for splitting dataset to training and testing parts. A feature scaling was applied to columns in X_train and X_test.

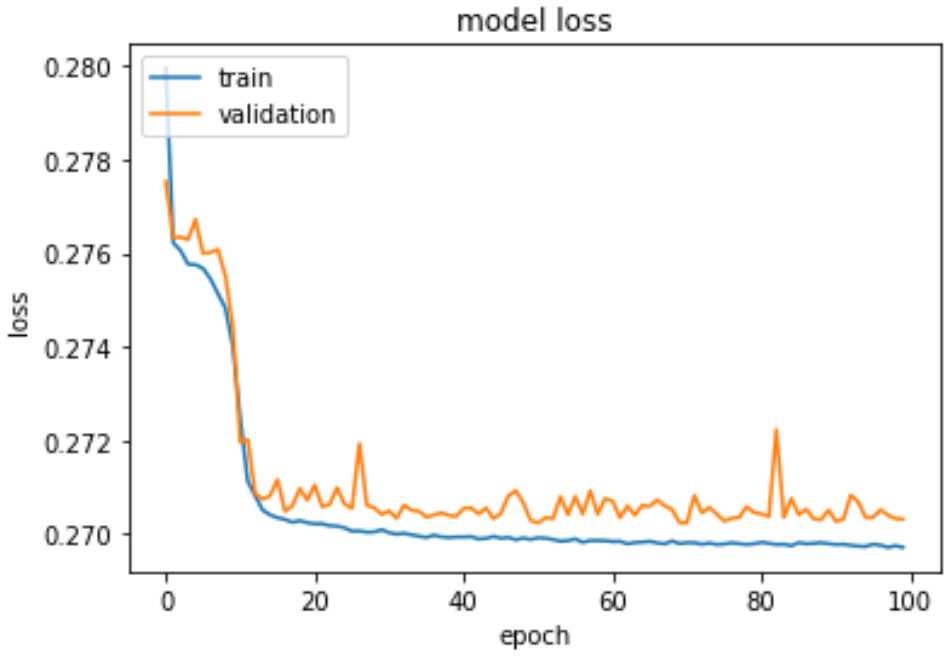
```
35  # feature scaling, min-max scaler (0-1 range)
36  scaler_x = MinMaxScaler()
37  X_train = scaler_x.fit_transform(X_train)
38  X_test = scaler_x.transform(X_test)
39
```

To build a model a ReLu and Sigmoid activation functions were used as well as binary cross entropy to compile. Batch size was set to 32 and epochs to 100.

```
45  # build model, use relu & sigmoid as the output activation function
46  model = Sequential()
47  model.add(Dense(12, input_dim=9, kernel_initializer='normal', activation='relu'))
48  model.add(Dense(8, activation='relu'))
49  model.add(Dense(1, activation='sigmoid'))
50  model.summary()
51  model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
52
53  history = model.fit(X_train, y_train, epochs=100, batch_size=32,  verbose=1, validation_data=(X_test,y_test))
54
55  # visualize training
```

4.4 Results of using Artificial Neural Network

The plot visualizes the learning progress of the ANN model.



Model learning visualization

For assessing how well did the model do a confusion matrix and accuracy score were used. The matrix result was:

[[66846 0]
 [9376 0]]

There is a large quantity of true positives, but no true negatives and a significant number of false negatives.

The accuracy score was: 0.8769908950171866

4.5 Comparison of used models

Comparing the accuracy score both models did almost the same (87%). The problem can be easily seen in the confusion matrix where both models have given unsatisfying results.

5. CONCLUSION

To conclude, the comparison of logistic regression and artificial neural network models has not given a clear winner in this classification case. A suspected problem behind insufficient confusion matrix results is believed to be the dataset that didn't allow the model to give better predictions. Just looking at pretty high accuracy score could give a feeling that the model predicts the results very well, however it is important to evaluate the model with more than one metric in order to determine whether the model is really giving accurate results. In this case, both models predicted a lot of false negative results in the confusion matrix. This underlines the importance of choosing the dataset properly. In case of a low quality dataset the results can be confusing, unclear or totally wrong.