Introduction to Data Mining Project Report

HEALTH INSURANCE VEHICLE CROSS SELLING PREDICTION

Abdur Rahim Khan 19760
Ali Asghar Zeeshan 18593
Abdul Moiz Zahid 19751
Abdullah Amin Chottani 18658

MEMBER ROLES

Abdul Moiz Zahid, Abdullah Amin Chottani and Abdur Rahim Khan – Understanding the problem, the dataset and determining the data type of each feature of the dataset. Use of statistical methods to find out relationships between set of features inside the data. Building of machine learning models and recording of performance scores.

Ali Asghar – Data visualization and extracting meaningful insights from the data.

Problem description

Problem Description: This includes the background of the company, the motivation of the study (challenges faced by the organization), how they are dealing it currently.

Problem: The client is an insurance company that has previously provided Health Insurance to its customers and now requires your assistance in developing a model to predict whether previous year's policyholders (customers) will be interested in the company's Vehicle Insurance.

Data Description

Data Description: Size of the dataset and description of the columns, Quality of the data (missing values proportion, constant columns, etc.)

The training dataset contained 380k rows of which one-third (127k) were selected for training. Each row contains 11 features and 1 boolean prediction feature 'Response'.

1. **Id** : Unique id for each customer
2. **Gender :** nominal variable – male(1) female(0)
3. **Age:** numeric age of the customer
4. **Driving_license:** 0/1 , Customer has no DL(0) , Customer has DL(1)
5. **Region_code:** numeric region code
6. **Previously_insured:** 0/1 , Customer has Vehicle Insurance(1) , Customer doesn't have Vehicle Insurance(0)
7. **Vehicle_Age:** categorical variable , < 1 year , 1-2 years , > 2 years
8. **Vehicle_damage:** 0/1 , If customer got vehicle damaged previously(1), If customer didn't get vehicle damaged previously(0)
9. **Annual_premium:** numeric amount that customer needs to pay as premium in the year
10. **Policy_sales_channel:** categorical variable. Anonymized code for the customer outreach channel, such as different agents, mail, phone, in person, etc.
11. **Vintage:** numeric. Number of days the customer has been associated with the company

Data Pre-Processing

Data Pre-Processing: What steps did you take to prepare the data?

We took the following steps to prepare our data:

1. Applied One-Hot Encoding to certain categorical features.
2. Applied normalizer, used min-max normalization to normalize certain numeric features.
3. Applied low variance filter to remove columns with low variance.
4. Applied correlation filter to remove redundant(correlated) columns.
5. Duplicate row filter to remove redundant rows from the data.
6. Applied PCA to project original features space onto a reduced dimension.

One-Hot Encoding

The following features were transformed using One-Hot encoding:

Gender (Male = 1, Female = 0)

Vehicle Damage (Yes = 1, No = 0)

Vehicle Age was converted to an ordinal variable since each value (< 1 year , 1-2 years , > 2 years) is meaningful

Label Encoding

Label encoding was applied to the features Region_code and Policy_sales_channel since they had too many values and applying One_Hot encoding to them would introduce quite a few unnecessary columns in our data.

Data Pre-Processing

Normalization

The sample that we used for this project comprised of certain features that were not normalized, such as: Age, Annual_premium and Vintage. Since most of the features were binary and ordinal (ranging from 0-2) having features with really large values would affect the quality of the model therefore we normalized Age, Annual_premium and Vintage using Min-Max normalization.

Low Variance Filter

Low variance is a feature selection approach that we used to filter out features that have a variance lower than a certain defined threshold. In our sample, the feature Driving_license had zero variance thus it was removed from our dataset.

Correlation Filter

The correlation filter is another feature selection approach that was used for our sample. The column with the most correlated columns is chosen to survive, while all other columns are removed. This is an iterative process until no more columns can be removed. In our case the column Vehicle_age was filtered out at a correlation threshold of 0.7.

Duplicate Row Filter

The duplicate row filter is a data preprocessing technique that removes all redundant rows from the data. In the sample that we used about a thousand duplicate rows were removed.

Data Pre-Processing

PCA

We used principle component analysis as another dimensionality reduction technique to transform our original features onto a reduced dimension. However, through trial and error of reducing our features to different dimensions (1, 2 and so on) we concluded that our model worked better off without it and therefore we excluded it from our final model.

Model Building + Evaluation: Which models did you try, which turned out to be the best (based on which metric), etc. How long did it take for a single evaluation? The description of the machine(s) on which you ran your experiments.

The following are the various models that were used in our project:

1. The Naïve Bayes Classifier
2. The Decision Tree
3. The Random Forest Classifier (Using Information Gain Ratio)
4. The Gradient Boosted Trees Classifier

Our original data comprised of 380k rows and therefore running it on our machines proved to be a struggle, therefore, we sampled one-third of the dataset for our use in the project. We used the Holdout method to split our dataset into two-third for training and one-third for testing.

Initially we started off with the Naïve Bayes Classifier to predict the class column Response.

Using the model Naïve Bayes was a bit tricky as there were 4 attributes: Default probability, minimum standard deviation, threshold standard deviation and maximum number of unique nominal values per attribute. In our first iteration, we went with the

Model Building

default configuration which yielded reasonable results with PCA. From there on, we kept changing the values of the variables one by one to see how our accuracy is impacted. After going back and forth with changing the values, it was found out that changing Minimum standard deviation, threshold standard deviation and maximum number of unique nominal values per attribute made no difference to the accuracy and the ROC curve.
Therefore, we increased the default probability with 0.004 points which gave us a higher accuracy. By increasing it more, the accuracy went down. We soon realized that the highest accuracy would be yielded by keeping the default probability between 0.005 and 0.006. The best ROC curve was made when the Default probability was 0.0055.
Without PCA, our accuracy was lower from the beginning as shown in the table.

We then moved on towards The Decision Tree Classifier as our second model. Since PCA gave an improvement for Naïve Bayes, we decided to use it for Decision Trees as well.

We used different number of MINIMUM RECORDS PER NODE to create a model to improve our accuracy. Here also we used the accuracy of ROC to analyze the model. The table of accuracy for different values of MINIMUM RECORDS PER NODE in DECISION TREE is shown below.

Minimum number of records per node	**ROC Score**
1	0.5961
5	0.7476
10	0.8116
50	0.8379
100	0.8382
200	0.8336

Model Building

We then applied PCA to reduce dimensions and check whether it increases the accuracy.

Min number of records per node	Roc Score
1	0.5928
5	0.7259
10	0.7985
50	0.85
100	0.8285
200	0.7576

We then moved on towards The Random Forest Classifier as our third model. We gradually increased the number of models to test the accuracy of the model. The measure we used was the accuracy of the ROC curve. The table of accuracy for different number of models in random forest is shown below.

Number Of Models	Roc Score
1	0.574
5	0.705
10	0.764
50	0.821
100	0.828
200	0.831

After that, we applied PCA to reduce dimensions and check whether it increases the accuracy.

Model Building

Number Of Models	Roc Score
1	0.564
5	0.687
10	0.749
50	0.816
100	0.826
200	0.83

We then moved on towards our last model which was the Gradient Boosted Trees Classifier. With all the preprocessing techniques (mentioned above) applied, we began applying the model using the holdout dataset.

Initially the learning rate was set to 0.1 (10%) and the depth to 10, however, after multiple trial and error runs, changing the depth each time, we found that the model was giving its best result at a depth of 3. Here are the results with learning rate set to 0.1 and the depth set to 3.

Number Of Models	ROC Score
1	77.67
5	78.26
10	80.12
50	81.09
100	81.07
200	81.17

Model Building

We then accidentally, in one of our team member's workflow forgot to change Policy_sales_channel and Region_code to a categorical variable, and with the rest of the settings remaining the same, got the following improvement.

Number Of Models	ROC Score
1	81.80
5	83.55
10	83.87
50	85.28
100	85.61
200	85.74

We then tweaked the GBT model a bit, changing the learning rate to 0.2 (20%) and obtained the following readings.

Number Of Models	ROC Score
1	81.78
5	83.64
10	84.53
50	85.59
100	85.73
200	85.80

Model Building

We then started tweaking our model a bit more by testing out if removing any filter would have any affect on the quality of the model. Therefore, we initially removed the Low Variance Filter, with the rest of the preprocessing techniques intact. The readings were measured at a learning rate of 0.1 and a depth of 3.

Number Of Models	ROC Score
1	82.80
5	83.80
10	84.12
50	85.51
100	85.80
200	85.93

We then tweaked this model by just increasing the learning rate to 0.2 with everything else being the same, and these were the readings.

Number Of Models	ROC Score
1	82.80
5	83.91
10	84.80
50	85.81
100	85.92
200	85.96

Model Building

Finally, we tested our model by removing only the Correlation Filter with the rest of the preprocessing techniques intact. (Reinstated the Low Variance Filter back)

These were the readings with a learning rate of 0.1 and a depth of 3.

Number Of Models	**ROC Score**
1	82.80
5	83.80
10	84.12
50	85.60
100	86.02
200	86.17

We then tweaked the model by increasing the learning rate to 0.2 with everything else being the same and this produced our best result.

Number Of Models	**ROC Score**
1	82.80
5	83.90
10	84.79
50	86.03
100	86.20
200	86.23

Model Building

We tried the GBT Classifier on a number of different settings, changing the learning rate, the depth, and the number of models, However, the most optimal result came at a depth of 3 and the number of models being within the 100-200 range. Increasing the depth beyond 3 would lead to a reduced reading and using models above 200 would also reduce the reading. The learning rate was also optimal between 0.1-0.2, and anything beyond that produced a score lower than our best one.

PCA didn't improve any of the models accuracy except for Naïve Bayes. It performed poorly for decision trees, random forest and gradient boosted trees therefore we excluded it from our final model.

Findings: This must include the insight you got from the data set, the limitations of your study, what advice you would like to give to the organization in terms of future data collection and processes, etc. If the organization plans to implement your solution, what are the key points related to deployment? When would the model expire, etc.?

Insights

We mainly used Data Visualization to find out to what extent do features play a role in determining the Class variable Response, to see if there is a connection between two features or not, to visualize any noticeable patterns, any difference in the data distribution, and used bar charts, distribution plots, count plots and boxplots to visualize any outliers.

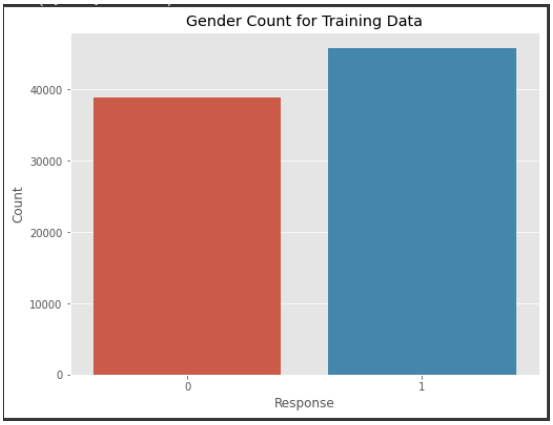
Findings

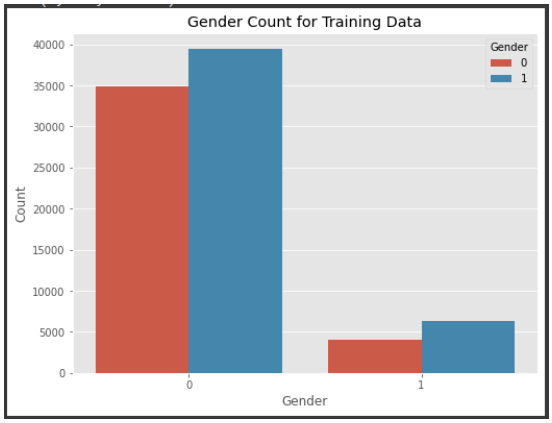
Response Feature

From the above bar chart that shows the distribution of the total responses of whether customers accepted/declined to get the vehicle insurance, it can be observed that the problem is an imbalance binary classification problem. 75k customers opted out of the vehicle insurance while 10k opted for it. Since it was an imbalance classification problem, we tried utilizing oversampling using SMOTE (Synthetic Minority Oversampling Technique) to balance the training set, however, it was taking hours to oversample it therefore we dropped the idea of using it.

Findings

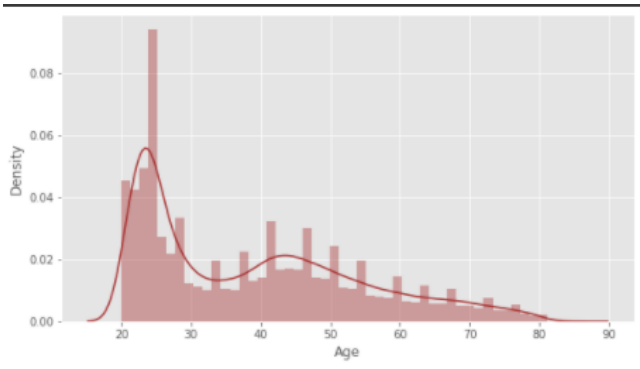
Gender Feature

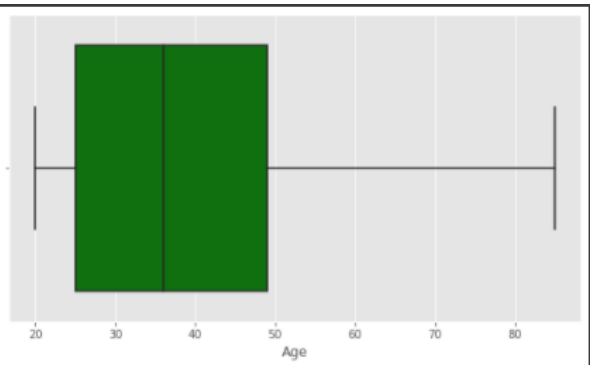




The above bar chart (Left) shows that the gender feature has close to an equal class distribution. However, the bar chart on the right shows that of all the people that opted for vehicle insurance, Males will have a greater chance of getting it. Therefore, the company can advertise their campaign more to males than females to bring about a greater yield in positive responses for them.

Age Feature

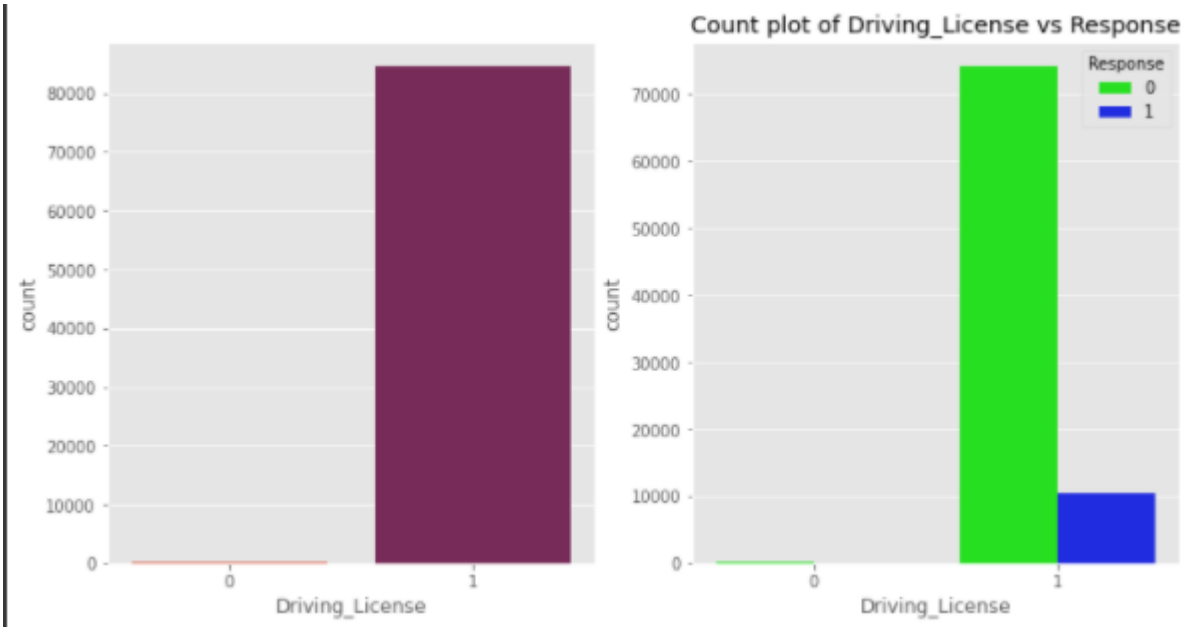




The above distribution plot shows that the count of individuals is greater within the 21-26 age bracket. Through the graph we can also see that the age feature is rightly skewed, and the boxplot helps us to see that there are as such no outliers in this feature.

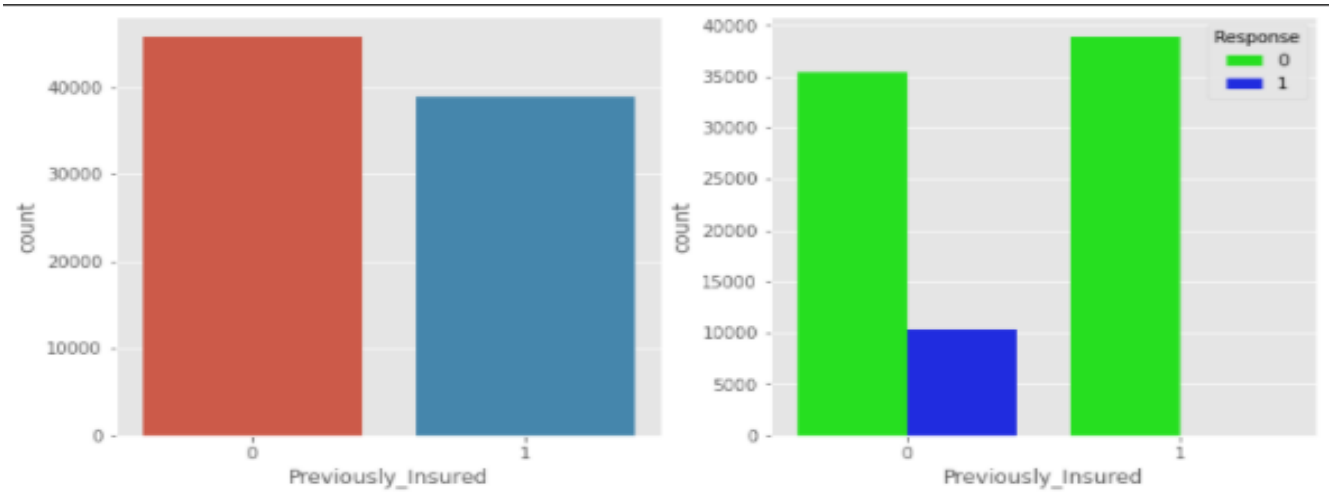
Findings

Driving License Feature.



The above bar chart shows that all the customers have a driving license. There are none who don't have one. And all the customers who have opted or are interested in the vehicle insurance have a driving license. Since this feature has absolutely no variance, therefore it was filtered out by the low variance filter during preprocessing.

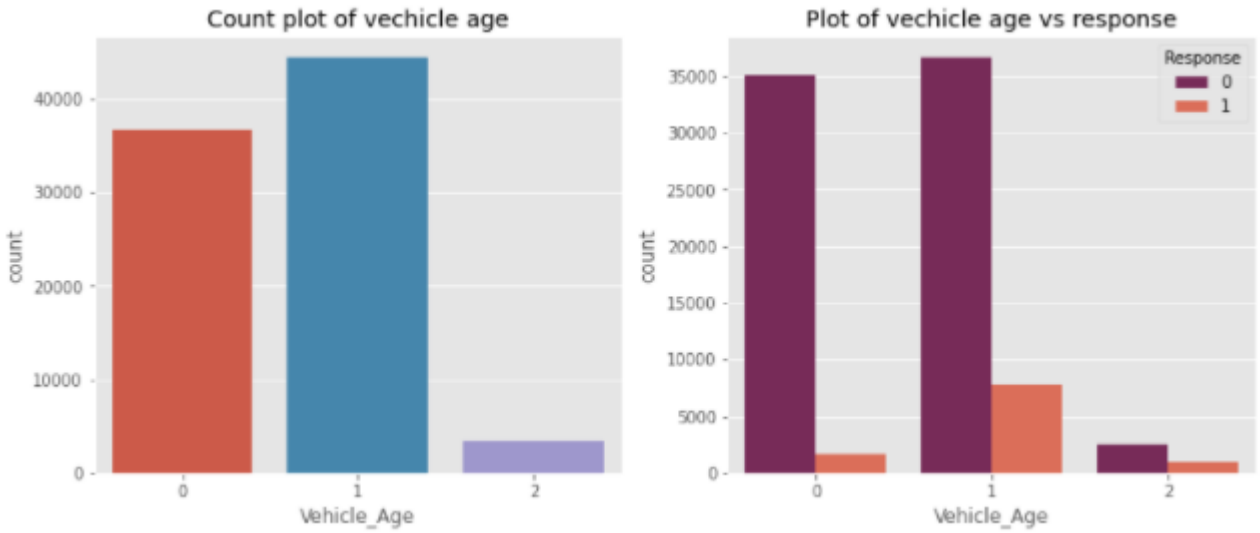
Previously Insured Feature



Findings

From the above bar chart (Left) we can see that the customers who aren't previously insured are greater than those who are previously insured. From the chart on the right, it can be inferred that people who aren't previously insured are most likely to opt for the vehicle insurance. The company can therefore target those people who are not previously insured.

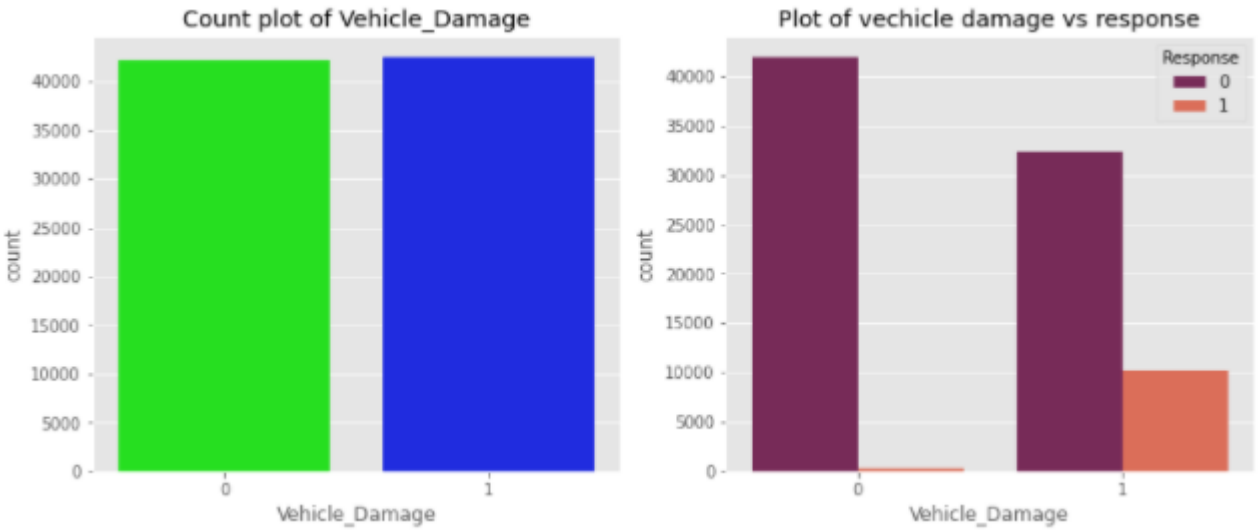
Vehicle Age Feature



The above bar chart shows that majority of the people have new cars, cars that are less than a year old. And it can also be inferred that majority of the people who are likely to be interested in opting for the vehicle insurance are those whose vehicles are between 1-2 years and less than a year old. Therefore, the company can target those customers of theirs who have recently bought a new car.

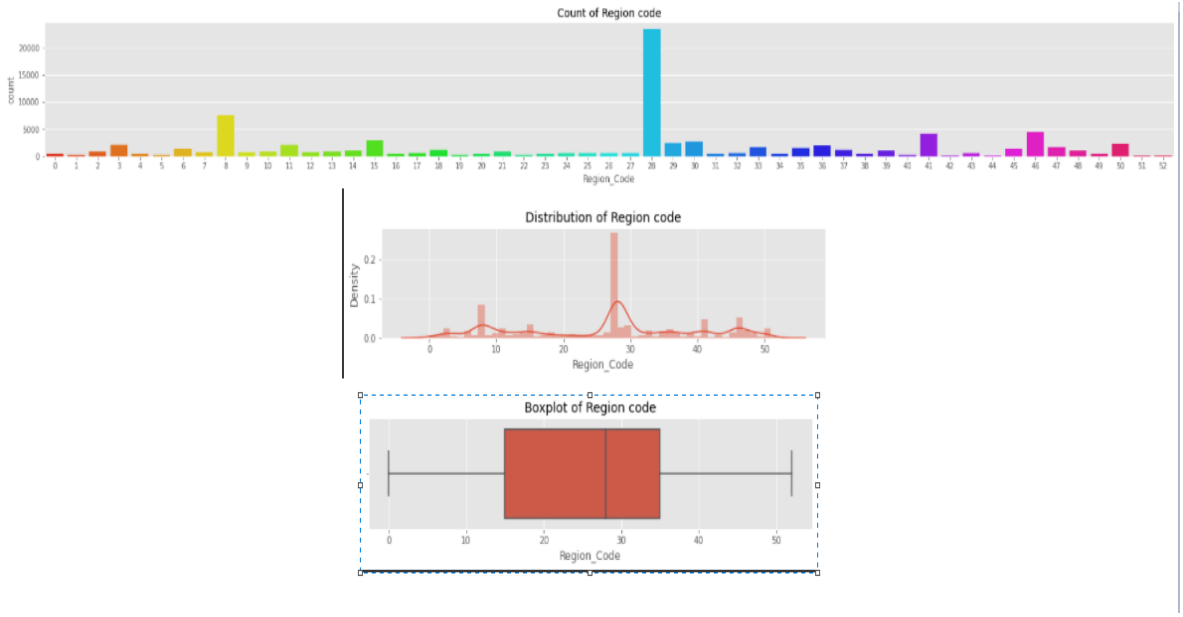
Findings

Vehicle Damage Feature



From the above bar charts, it can be seen that the company's customers are equally distributed when it comes to their vehicles being damaged or not. It can also be inferred those customers who have damaged vehicles are more inclined towards opting for the vehicle insurance.

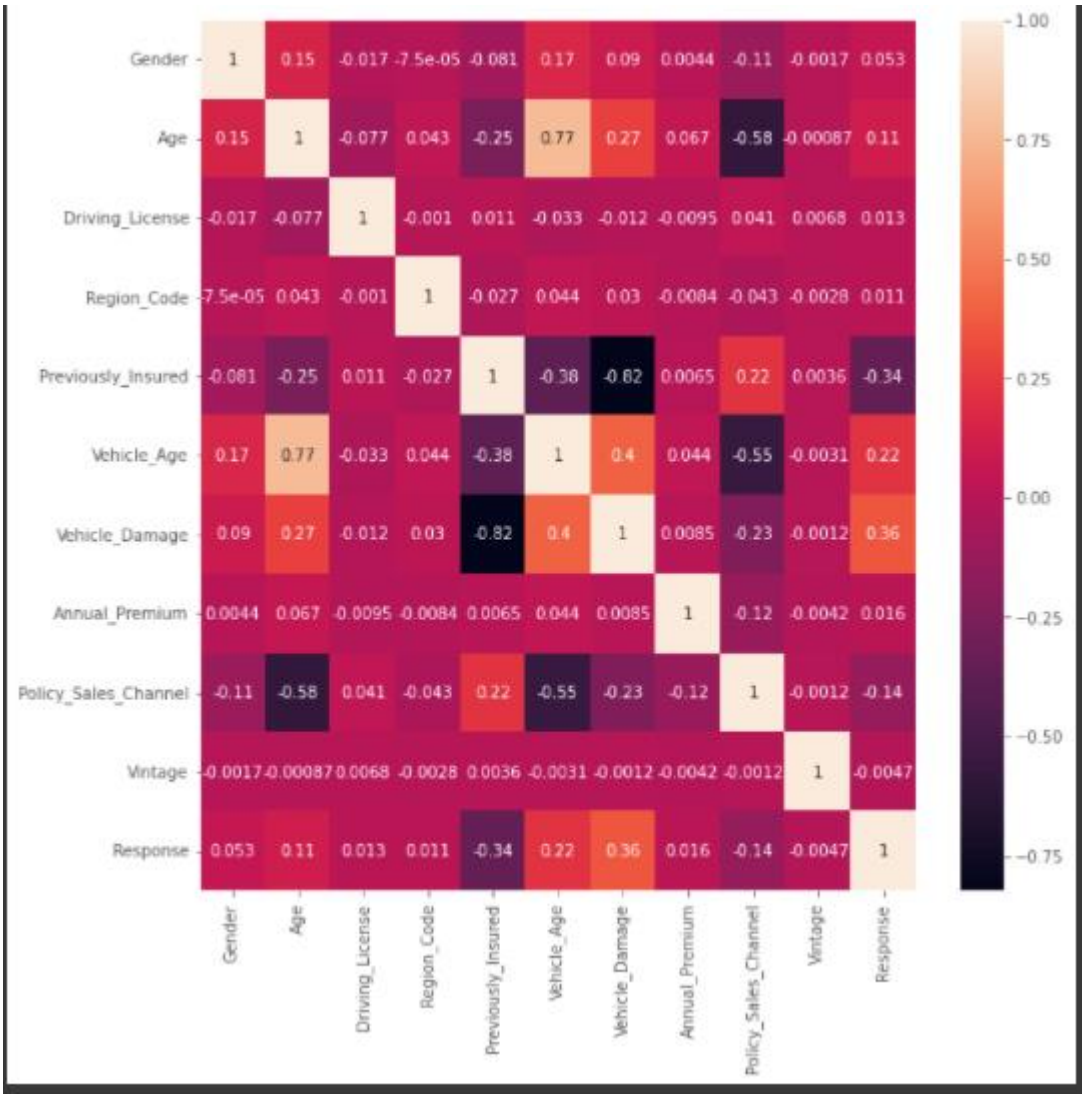
Region Code Feature



Findings

From the above plots, we can see that region code 28 has the highest count of customers. It can also be seen from the box plot that there are no outliers in this feature. Since region 28 has the highest count of customers for the company, the company should focus more on that region in ensuring that their responses towards vehicle insurance increase.

Feature Importance



To determine which features would play and played an important role in determining the positive class variable of the feature Response, we made use of the correlation

matrix of the python seaborn library. In the above correlation matrix, we can see that the columns Previously_insured, Vehicle_age and Vehicle_damage are the most correlated with Response, and therefore have a greater importance in determining it.

We can also see that the features Vehicle_age and Vehicle_damage are highly correlated with most of the columns, therefore this further implies their importance, since the column with the most correlated columns is chosen to survive, while all other columns are removed according to the correlation filter.

To test out just how important they are in determining response. We did a few runs of our best model, removing each of these variables one at a time and testing just how important they are.

Our best model without Vehicle Damage

Number Of Models	**Roc Score**
200	85.03

Our best model without Vehicle Age

Number Of Models	**Roc Score**
200	84.95

Our best model without Previously Insured

Number Of Models	**Roc Score**
200	84.92

Findings

Challenges and Limitations

Challenges involved mainly included preprocessing the data. Identifying which filters to use was extremely key to reach a proper conclusion.

Partitioning the whole data set in to two parts was important as well to be efficient in solving the problem. Deciding at which percentage to partition the data was highly important to solve our problem.

Converting Vehicle age to an ordinal variable was also a bit difficult but was highly important for the algorithms to work properly on the data.

Figuring out which numeric features to normalize was also a challenge. It required a lot of time and effort.

A lot of time was spent in testing various filters. Figuring out the best combination that works correctly for our data was a challenge.

The quality of the model was judged throughout after tweaking a few things and applying various techniques

Using the correct threshold for correlation filter was important in order to make sure that important columns would not be removed from our dataset.

Using principle component analysis and identifying models that worked better with PCA was also a challenge.

One limitation was that we were unable to identify the effect of policy sales channel, vintage, and annual premium on the final response.

When would the model expire?

The model would require retraining if the company decides to change their policy or the government decides the change the minimum age for a person to be eligible for a driver's license.

The model should last a good 5 years unless there are significant changes in the trends of their clients. For example, a lot of people begin to opt out for an insurance.

If the age and gender of the clients change drastically as well, the model will have to be retrained.