Building a model to predict whether a customer having a health insurance, will also take Vehicle Insurance or not.

Project by:

Darkunde Pandurang

Kedar Gaurav

Gugale Anand

Project Guide: Prof. Sarika Khirid

Modern College of Arts, Science and Commerce, Pune – 05

Progressive Education Society

Modern College Arts, Science and Commerce.

Shivajinagar, Pune - 411005

CERTIFICATE

This is to certify that Miss.-____________________, Roll No., "Bachelor of Science in Statistics" (Third Year) has successfully completed his project entitled "Modelling of Magic Gamma Telescope Data by various methods". As prescribed by the Savitribai Phule University, for the academic year 2020-2021.

Date:

Project Guide **Head of Department**

Prof. Khirid Sarika M. **Dr.P.G.Dixit**

Internal Examiner **External Examiner**

Acknowledgement

We would like to offer our sincere gratitude to Prof. Sarika Khirid, our project guide, who guided us throughout the process. We would also like to thank Dr. P.G.Dixit, Head of Department of Statistics and all teaching and non teaching staff of Department of Statistics, Modern College of Arts Science and Commerce, Shivajinagar, Pune, for their invaluable assistance. Lastly, we would like to thank our parents and friends for their support and encouragement

Index

Contents

Objectives

- To analyse business problem.
- To understand different tools of visualization.
- To apply and understand different machine learning models according to problems.
- To find best model according to need of a client.
- To give the brief solution to above business problem so that company ear

Explanatory Data Analysis

Health Insurance

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.

For example, you may pay a premium of Rs. 5000 each year for a health insurance cover of Rs. 200,000/- so that if, God forbid, you fall ill and need to be hospitalised in that year, the insurance provider company will bear the cost of hospitalisation etc. for upto Rs. 200,000. Now if you are wondering how can company bear such high hospitalisation cost when it charges a premium of only Rs. 5000/-, that is where the concept of probabilities comes in picture. For example, like you, there may be 100 customers who would be paying a premium of Rs. 5000 every year, but only a few of them (say 2-3) would get hospitalised that year and not everyone. This way everyone shares the risk of everyone else.

Car Insurance

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.

Problem Statement

Our client is an Insurance company that has provided Health Insurance to its customers now they need our 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.

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.

Now, in order to predict, whether the customer would be interested in Vehicle insurance, we have information about demographics (gender, age, region code type), Vehicles (Vehicle Age, Damage), Policy (Premium, sourcing channel) etc.

Description Of Data

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

Relation Between response and predictor variable

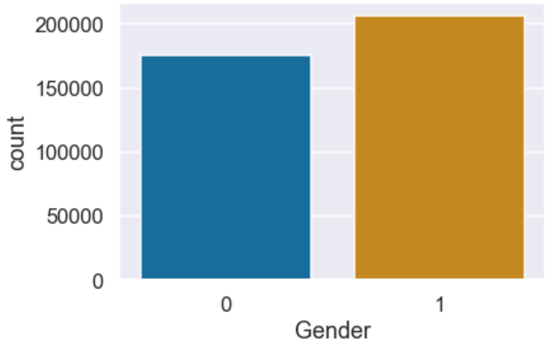
Before fitting the appropriate model, it is very important to check whether are the response and predictor variable exhibit any relation or not.

If we see carefully then our response variable is categorical so we cannot check relation by using count plot (bar plot).

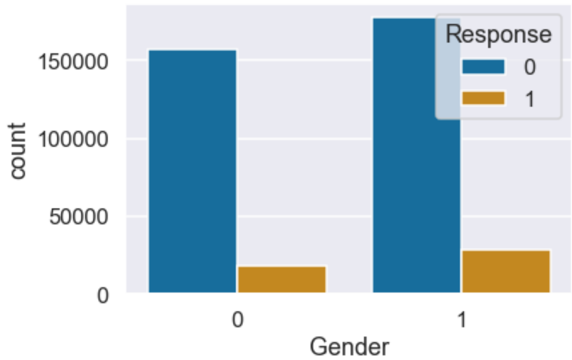
1) Id

Id variable is unique id of a person it is distinct for distinct person so we will remove it from our database.

2)Gender



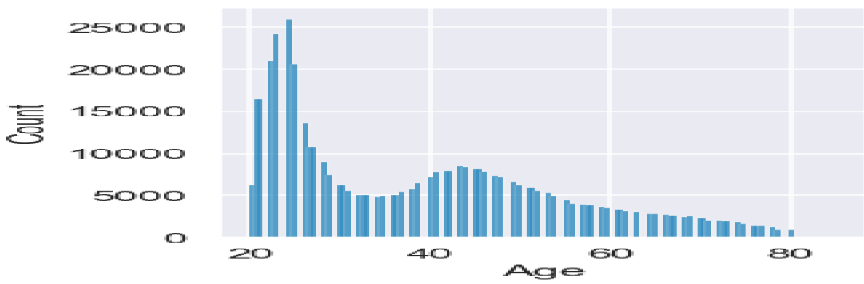
No of health insurance taken by male is greater than female.



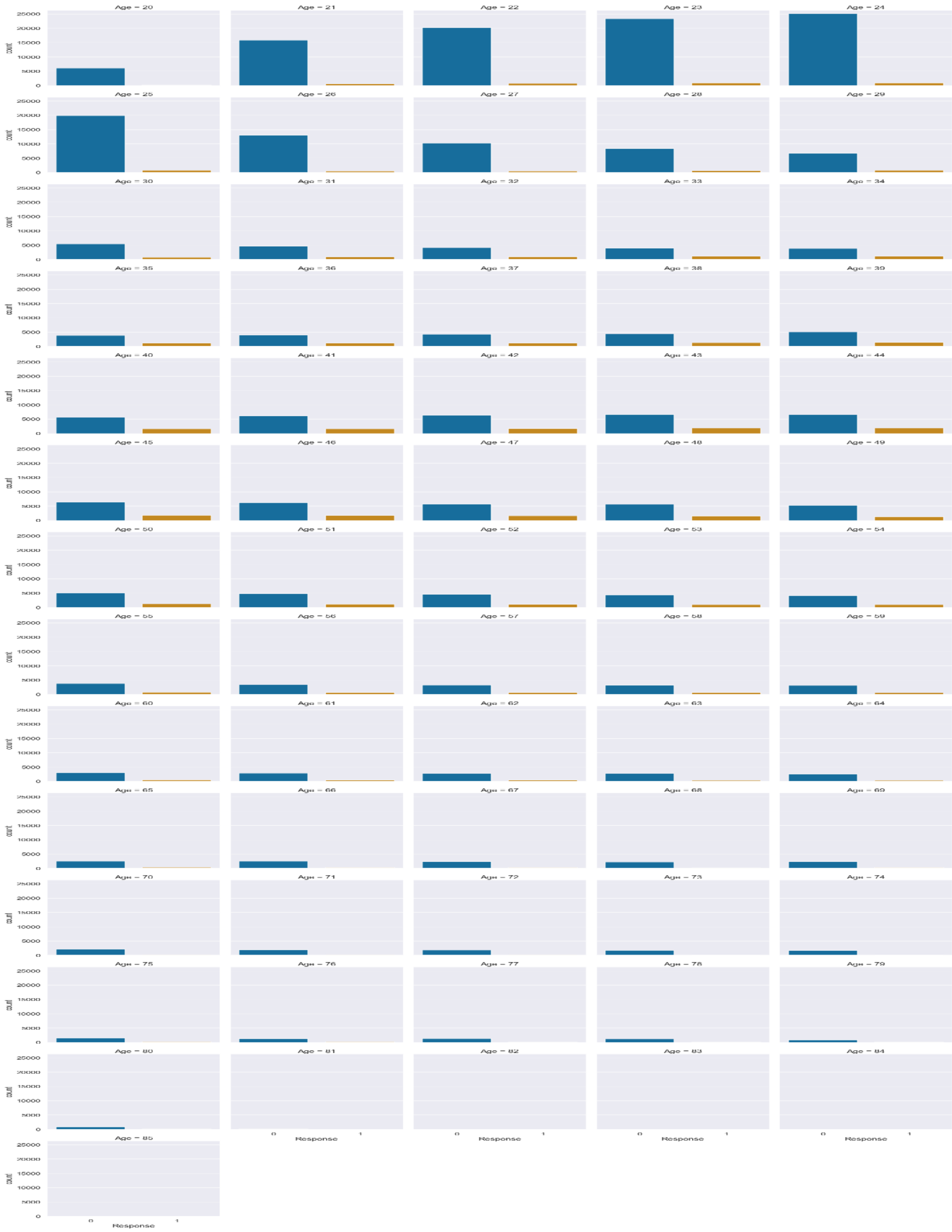
If we see carefully then no. of car insurance taken by male is grether than female. So we can say that responses is depends on gender of a person.

3) age

Distribution of age



If we see above plot carefully then we can recognize that client belong to age group 20-40 have taken the life insurance more than the other age group. And after age 50 we see decreasing pattern.

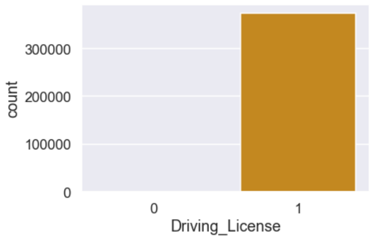


From above graph it is clear that client belonging to age group 30- 60 have taken the car insurance more frequently.

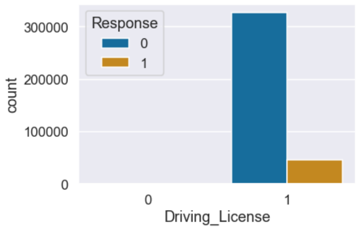
Also, no. of client belonging to age group 81 to 85 have taken life insurance and car insurance is very less. So, this age group is outlier for our dataset so we will exclude data related to those clients from our dataset.

No of Clients belongs to age less than 30 have is very less. So we can say that our response variable responses is depends on a age of a client.

4)Driving licence



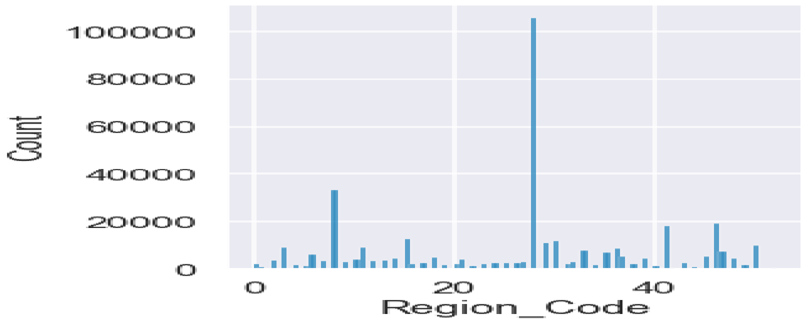
We can see that all the clients have driving licence.



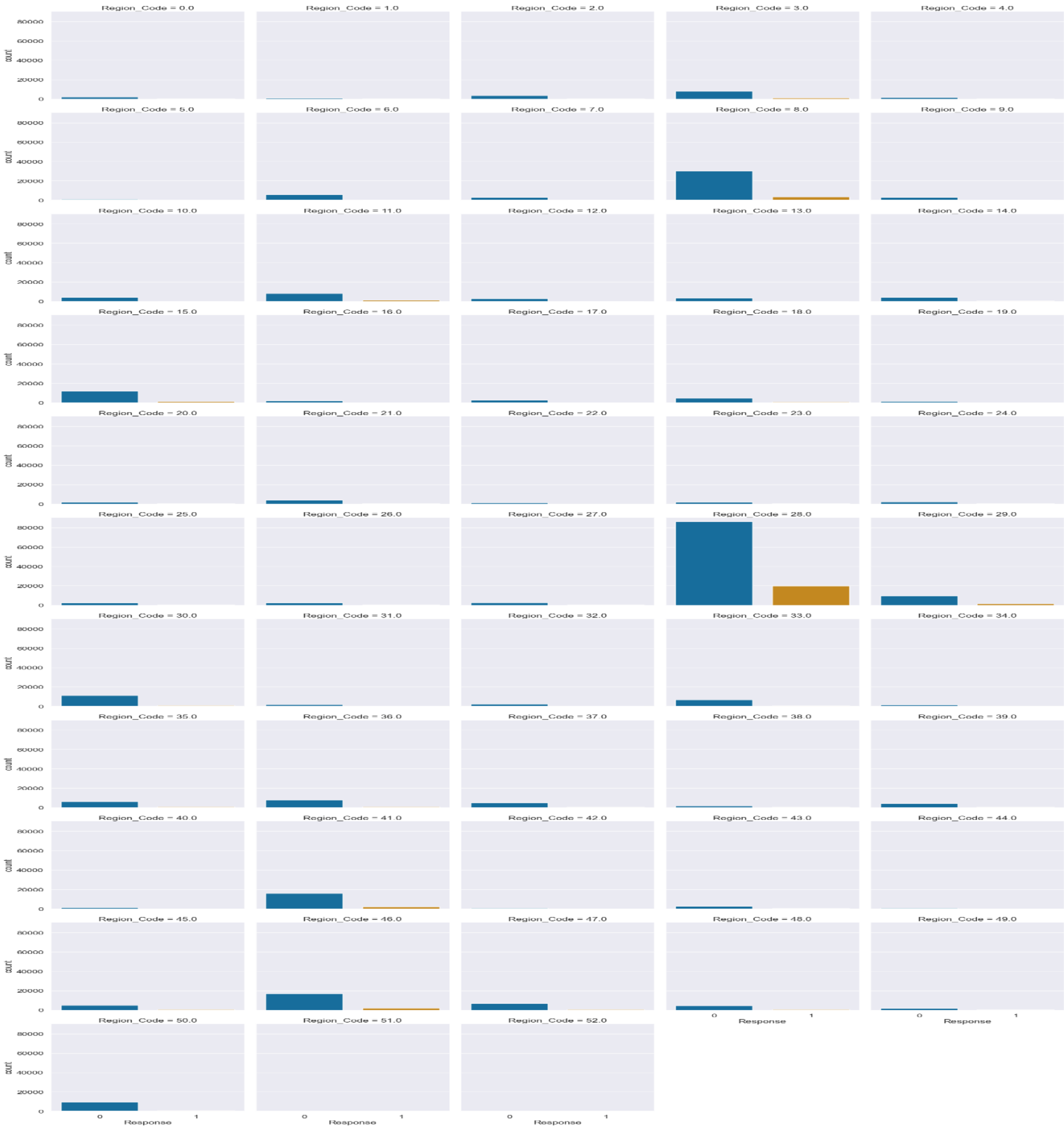
So, it is clear from above plot that response variable responses does not depend on driving licence of a client.

We will drop whole column of variable driving licence from our dataset.

5) Region_Code

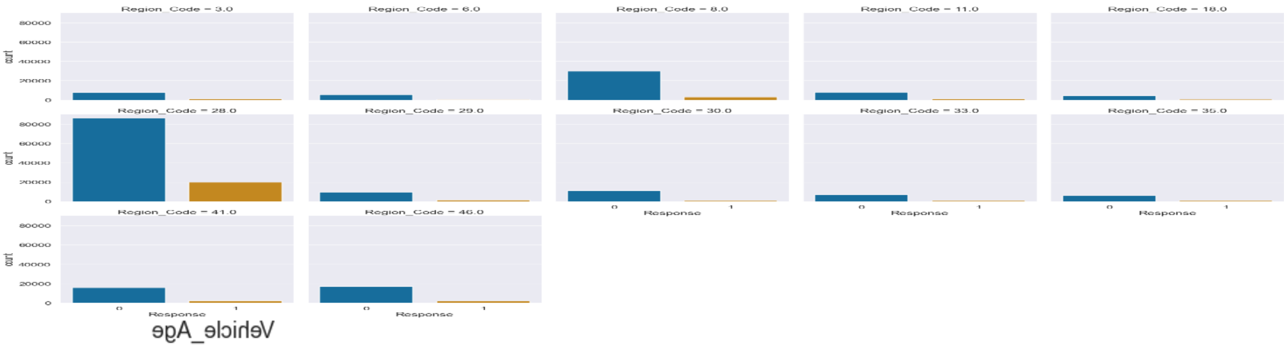


It is clear that client's approach towards the health insurance is depends on Region from which they belonging.



We can see that there are some region's from which no. of clients taking health and car insurance is too much less our negligible as compare to other regions.

That's why we will focus on those regions from where we are getting more clients. And we will keep record of those regions only.



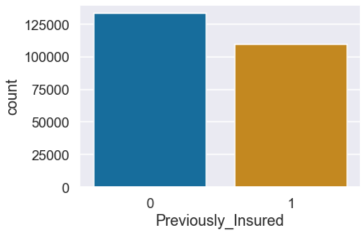
Our most of the clients belonging to Region no. 3,6,8,11,18,28,29,30,33,35,41 and 46.

So we will keep records of these regions only.

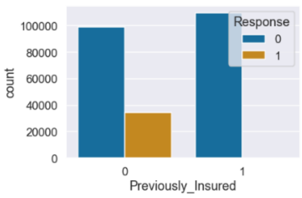
According to total no of clients from the particular region we will rank the classes and encode them as

Region_code 18 as = 0,Region_code 35, as = 1,Region_code 6, as = 2,Region_code 33, as = 3,Region_code 11, as = 4,Region_code 3, as = 7,Region_code 29, as = 5,Region_code 30, as = 6,Region_code 41, as = 8,Region_code 46, as = 9,Region_code 8, as = 10,Region_code 28, as = 11

6) Previously_Insured



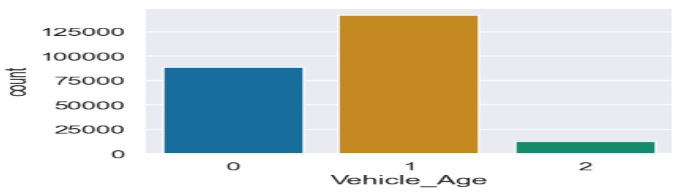
It is obvious that's if the client already has car insurance, then he will not take car insurance. let see what the result given data show to us.



As expected, client who already have the car insurance that client is not taking car insurance.

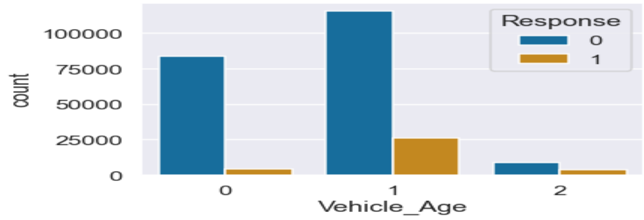
So response variable responses is depends on the Previously_Insured variable.

7) Vehicle age



0 means vehicle_age is less than 1 years,1 means vehicle_age is between 1-2 years

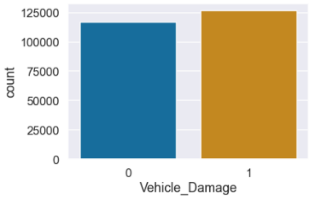
2 means vehicle_age is greater than 2 years.



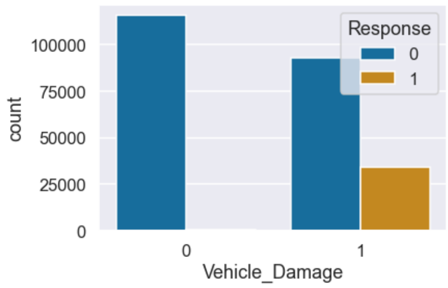
There are very less client whose vehicle age is less than 1 year and grether than 2 year which takes the car insurance.

So from above we can conclude that response variable response is depends on a age of a vehicle.

8) vehicle damage



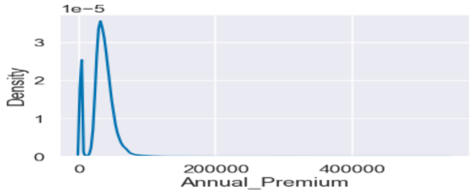
Now we are interested to check that client whose car is damage that client take the car insurance or a client whose car is not damage that client take the car insurance.



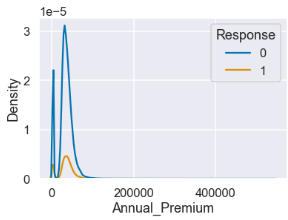
So here we got the one interesting result that the client whose car is well that client is not taking car insurance.

So the our response variable responses is depends on vehicle_damage variable.

9)Annual_premium



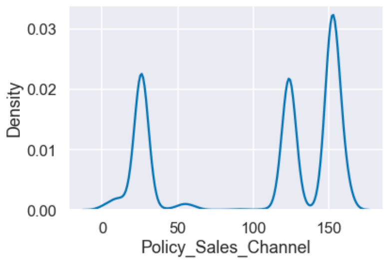
Now we will check whether response variable responses is depends on a Annual_premium paid by a client or not.



Now it is clear from above density plot that response variable responses does not change with respect to premium so Annual_premium does not give any valid information about responses. So they are uncorrelated.

10) Policy_Sales_Channel

Distribution of Policy Sales channel

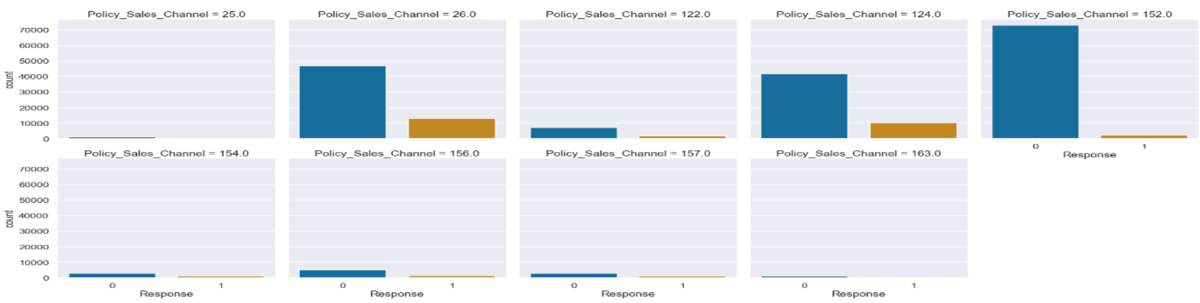


So it is very important to note that channel wise no of clients changes. No. of clients approach by different channel is different.

Now we will keep the record of those channel which gives the insurance company more clients.

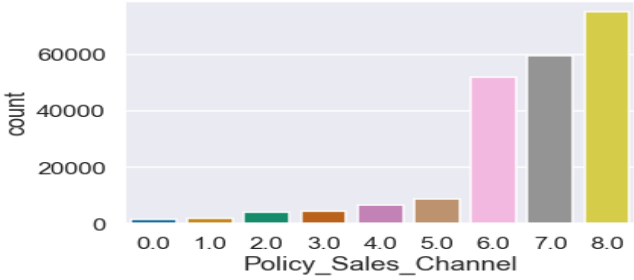
Now from below plot we can see that there are only few channels which giving more clients to the insurance company.

25,26,122,124,152,154,156,157 and 163 these are the only channels which approaches to more clients so we will keep the record of these channels only.



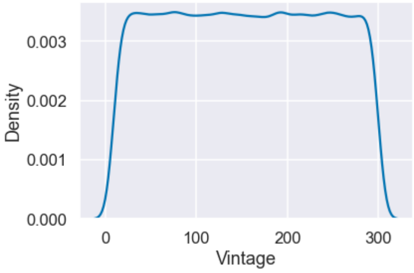
Now we will arrange these channels according to no of clients they have approached and encode them accordingly.

Policy_Sales_Channel no 25 as 0, Policy_Sales_Channel no 163 as 1, Policy_Sales_Channel no 154 as 2, Policy_Sales_Channel no 157 as 3, Policy_Sales_Channel no 156 as 4, Policy_Sales_Channel no 122 as 5, Policy_Sales_Channel no 124 as 6, Policy_Sales_Channel no 26 as 7, Policy_Sales_Channel no 152 as 8

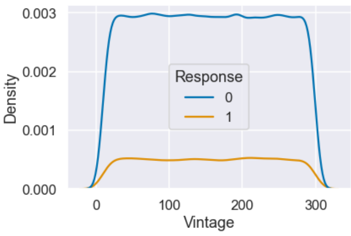


11) Vintage

Distribution of vintage.



Now we will check whether response variable responses is depends on a vintage or not.



If we carefully then density of both the responses is same with change in vintage. So we will drop vintage column from dataset.

Now we are ready for analysis.

Further we will standardise the non-categorical variables. And fit the classification models

Supervise machine learning Models for classification

1) Random forest model.

Reference Link:- https://towardsdatascience.com/understanding-random-forest-58381e0602d2

2) Logistic Regression model.

Reference Link:-Logistic Regression in Machine Learning - A Basic Guide For 2021 (jigsawacademy.com)

3)K-nearest-neighbour model

Reference link:- https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=Summary-,The%20k%2Dnearest%20neighbors%20(KNN)%20algorithm%20is%20a%20simple,both%20classification%20and%20regression%20problems.

4)Decision Tree model:-

Reference Link:- https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

5)Naïve Bayes model:-

Reference Link:- https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/#:~:text=It%20is%20a%20classification%20technique,presence%20of%20any%20other%20feature.

6)Kernel SVM Model:-

Reference model:- https://towardsdatascience.com/svm-classifier-and-rbf-kernel-how-to-make-better-models-in-python-73bb4914af5b

Fitting of a different classification models on a training set.

##1)fiting a Random forest model

```
from sklearn.ensemble import RandomForestClassifier
classifier1 = RandomForestClassifier(n_estimators = 100, criterion = 'entropy', random_state = 0)
classifier1.fit(X_train, y_train)
```

```
RandomForestClassifier(criterion='entropy', random_state=0)
```

###Confusion matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report,precision_recall_fscore_support
y_pred1 = classifier1.predict(X_test)
report1=precision_recall_fscore_support(y_test,y_pred1)
cm = confusion_matrix(y_test, y_pred1)
print(cm)
print(accuracy_score(y_test, y_pred1))
a1=accuracy_score(y_test, y_pred1)
print(classification_report(y_test, y_pred1))
```

```
[[17728   511]
 [ 2810   312]]
0.8445297504798465
              precision    recall  f1-score   support

           0       0.86      0.97      0.91     18239
           1       0.38      0.10      0.16      3122

    accuracy                           0.84     21361
   macro avg       0.62      0.54      0.54     21361
weighted avg       0.79      0.84      0.80     21361
```

##2) fiting logistic Regression model

```
import statsmodels.api as sm
logit_model=sm.Logit(y_train,X_train)
classifier2=logit_model.fit()
print(classifier2.summary2())
```

###Confusinon Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report,precision_recall_fscore_support
y_pred2 = (classifier2.predict(X_test)>0.5)
report2=precision_recall_fscore_support(y_test,y_pred2)
cm = confusion_matrix(y_test, y_pred2)
print(cm)
print(accuracy_score(y_test, y_pred2))
a2=accuracy_score(y_test, y_pred2)
print(classification_report(y_test, y_pred2))
```

```
[[18006   233]
 [ 3004   118]]
0.8484621506483779
              precision    recall  f1-score   support

           0       0.86      0.99      0.92     18239
           1       0.34      0.04      0.07      3122

    accuracy                           0.85     21361
   macro avg       0.60      0.51      0.49     21361
weighted avg       0.78      0.85      0.79     21361
```

3)Fiting KNN model

```
from sklearn.neighbors import KNeighborsClassifier
classifier3 = KNeighborsClassifier(n_neighbors = 3, metric = 'minkowski', p = 2)
classifier3.fit(X_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

###Confusion matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report,precision_recall_fscore_support
y_pred3 = classifier3.predict(X_test)
report3=precision_recall_fscore_support(y_test,y_pred3)
cm = confusion_matrix(y_test, y_pred3)
print(cm)
print(accuracy_score(y_test, y_pred3))
a3=accuracy_score(y_test, y_pred3)
print(classification_report(y_test, y_pred3))
```

```
[[16793  1446]
 [ 2358   764]]
0.8219184495107907
              precision    recall  f1-score   support

           0       0.88      0.92      0.90     18239
           1       0.35      0.24      0.29      3122

    accuracy                           0.82     21361
   macro avg       0.61      0.58      0.59     21361
weighted avg       0.80      0.82      0.81     21361
```

##4) training decision tree model

```
from sklearn.tree import DecisionTreeClassifier
classifier4 = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier4.fit(X_train, y_train)
```

```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

###Confusion matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report,precision_recall_fscore_support
y_pred4 = classifier4.predict(X_test)
report4=precision_recall_fscore_support(y_test,y_pred4)
cm = confusion_matrix(y_test, y_pred4)
print(cm)
print(accuracy_score(y_test, y_pred4))
a4=accuracy_score(y_test, y_pred4)
print(classification_report(y_test, y_pred4))
```

```
[[17809   430]
 [ 2863   259]]
0.8458405505360236
              precision    recall  f1-score   support

           0       0.86      0.98      0.92     18239
           1       0.38      0.08      0.14      3122

    accuracy                           0.85     21361
   macro avg       0.62      0.53      0.53     21361
weighted avg       0.79      0.85      0.80     21361
```

5) Fitting of a naive Bayes Classification model on a training dataset

In [88]:
```
from sklearn.naive_bayes import GaussianNB
classifier5 = GaussianNB()
classifier5.fit(X_train, y_train)
```

Out[88]:
```
GaussianNB()
```

###Confusion Matrix

In [89]:
```
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report,precision_recall_fscore_support
y_pred5 = classifier5.predict(X_test)
report5=precision_recall_fscore_support(y_test,y_pred5)
cm = confusion_matrix(y_test, y_pred5)
print(cm)
print(accuracy_score(y_test, y_pred5))
a5=accuracy_score(y_test, y_pred5)
print(classification_report(y_test, y_pred5))
```

```
[[10850  7389]
 [   91  3031]]
0.6498291278498197
              precision    recall  f1-score   support

           0       0.99      0.59      0.74     18239
           1       0.29      0.97      0.45      3122

    accuracy                           0.65     21361
   macro avg       0.64      0.78      0.60     21361
weighted avg       0.89      0.65      0.70     21361
```

##6) Training Kernal SVM

```
from sklearn.svm import SVC
classifier6 = SVC(kernel = 'rbf', random_state = 0)
classifier6.fit(X_train, y_train)
```

```
SVC(random_state=0)
```

Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report,precision_recall_fscore_support
y_pred6 = classifier6.predict(X_test)
report6=precision_recall_fscore_support(y_test,y_pred6)
cm = confusion_matrix(y_test, y_pred6)
print(cm)
print(accuracy_score(y_test, y_pred6))
a6=accuracy_score(y_test, y_pred6)
print(classification_report(y_test, y_pred6))
```

```
[[18239     0]
 [ 3122     0]]
0.8538457937362482
              precision    recall  f1-score   support

           0       0.85      1.00      0.92     18239
           1       0.00      0.00      0.00      3122

    accuracy                           0.85     21361
   macro avg       0.43      0.50      0.46     21361
weighted avg       0.73      0.85      0.79     21361
```

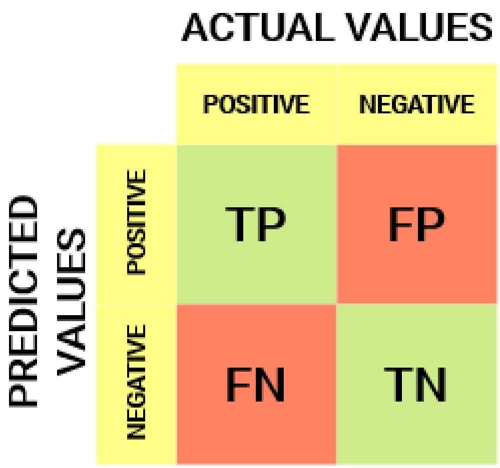
Project_python_file_link:-
https://drive.google.com/file/d/14hg3IucH8VQoYERJC6dtHT2YIcERw19r/view?usp=drivesdk

Model evaluation metric

Confusion matrix: -

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:



- The target variable has two values: **Positive** or **Negative**
- The **columns** represent the **actual values** of the target variable
- The **rows** represent the **predicted values** of the target variable

True Positive (TP)

- The predicted value matches the actual value
- The actual value was positive and the model predicted a positive value

True Negative (TN)

- The predicted value matches the actual value
- The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

- The predicted value was falsely predicted
- The actual value was negative but the model predicted a positive value
- Also known as the **Type 1 error**

False Negative (FN) – Type 2 error

- The predicted value was falsely predicted

- The actual value was positive but the model predicted a negative value
- Also known as the **Type 2 error**

1) Accuracy: -

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2)Precision:- *Precision tells us how many of the correctly predicted cases actually turned out to be positive.*

$$Precision = \frac{TP}{TP + FP}$$

This would determine whether our model is reliable or not.

3)Recall: - *Recall tells us how many of the actual positive cases we were able to predict correctly with our model.*

$$Recall = \frac{TP}{TP + FN}$$

4)F1Score

In practice, when we try to increase the precision of our model, the recall goes down, and vice-versa. The F1-score captures both the trends in a single value:

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Evaluation of Model

Our main aim is to find the model which will correctly classify the client in yes class. Our main focus is yes class. That's why first we will check the recall value for yes class. Their after we will check for precision and accuracy to find the best model.

- Recall value for yes class =

$$\frac{\text{Number of clients correctly classified in yes class}}{\text{Number of clients correctly classified in yes class+Number.of clients wrongly classified in No class}}$$

- Precision value for yes class=

$$\frac{\text{Number of clients correctly classified in yes class}}{\text{Number of clients correctly classified in yes class+Number.of clients wrongly classified in yes class}}$$

- Accuracy=

$$\frac{TP+TN}{TP+TN+FP+FN}$$

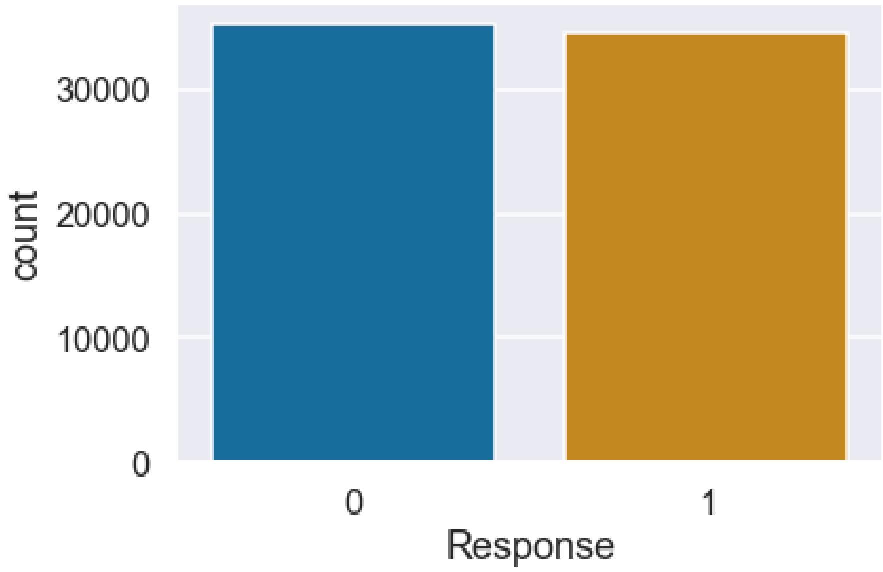
Result and Model evaluation

RESULT

No.	Name of fitted model	Recall value	Precision Value	Accuracy
1)	Random forest	0.10	0.38	0.84
2)	Logistic Regression	0.04	0.34	0.85
3)	KNN	0.24	0.35	0.82
4)	Decision Tree	0.08	0.38	0.85
5)	Naïve Bayes	0.97	0.29	0.65
6)	Kernel SVM	0.00	0.00	0.85

- According to our need we wont a model which will more and correctly classify the client into yes class.
- Best result is shown by Naïve Bayes classifier.
- Recall value= 0.97
  It means that out of 100 clients which are belonging to yes class our model classifies 97 of them in yes class correctly, there is error of 3% only.
- Precision value= 0.29
  It means that out of 100 clients which our model classifies in yes class, 29 of them are belonging to yes class.
- Accuracy=0.65
  It means that out of 100 clients 65 clients are correctly classifies.
  Though accuracy is less as compare to other models but still Naïve bayes classifier is best for the given business problem.

Solution to the given business problem using Naïve bayes classifier.



- We successfully classify all the clients.
- Out of 69649 clients Naïve bayes classifier classifies 35177 in No class and 34472 into Yes class.
- From 34472 clients which are classified in yes class, 29% means 9997 will definitely take the car insurance. (Precision=0.29)

FINAL_RESULT

Gender	Age	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Policy_Sales_Channel
M	1.538439	11	0	1	1	5
F	0.550071	7	0	1	1	7
F	-1.1631	11	0	1	1	6
M	0.286506	11	0	2	1	7
M	0.48418	11	0	1	1	7
M	-0.04295	6	0	1	1	2
F	-0.70186	11	0	0	1	8
F	1.340765	5	0	1	1	7
M	-1.22899	11	0	0	1	6
M	-1.1631	11	0	0	1	8
F	0.879527	11	0	2	1	7
M	0.088833	11	0	1	1	6
F	0.154724	11	0	1	1	6
F	-1.29488	9	0	0	1	8

Final_result_link:-
https://drive.google.com/file/d/13TQHSmgDImJtRc-Z4QHN-qZRi8URE-VS/view?usp=drivesdk

Conclusion

- **Finally, the best result is shown by Naïve Bayes Classifier.**
- **Out of 69649 clients who had a health insurance our model classify 34472 in yes class(Clients which will take the car insurance) and out of 34472 clients, 29% clients will take the car insurance.**

REFERENCES

- Data was provided by the insurance company, of clients who are insured with a health insurance policy.
- Data reference link:-
    1. Training dataset:- https://drive.google.com/file/d/1WPV-A7S_9fhU02IRYQ5jArqlkTzlBcnm/view?usp=drivesdk
    2. Test dataset:- https://drive.google.com/file/d/1xsnqj_Jlvqj2_rkJcI-B1SFDI-HHkRME/view?usp=drivesdk