Big Data Analytics Project Document

Health Insurance Cross Sell Prediction

Contents

Problem description

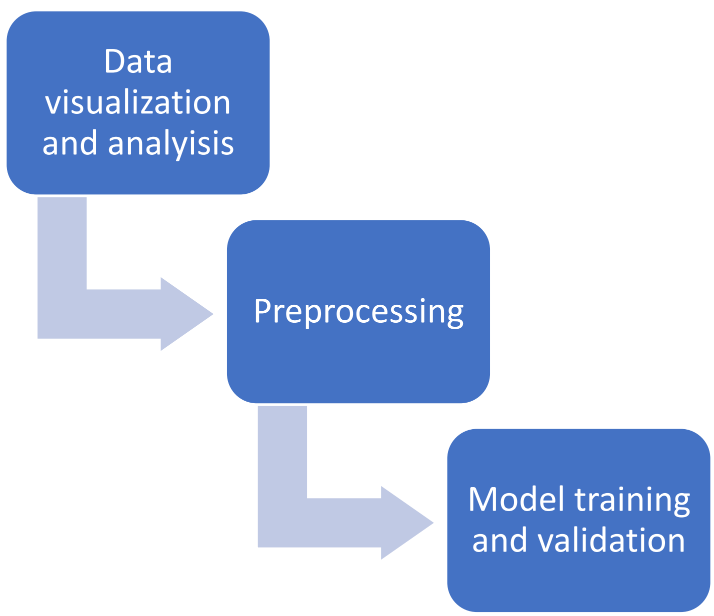
The client is an insurance company that has provided Health Insurance to its customers. Now, they need to build a model to predict whether the customers from past year will also be interested in Vehicle Insurance provided by the company.

vehicle insurance means that every year customer needs to pay a certain amount of money 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.

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.

This is a running competition on Kaggle that we are willing to compete on along with other competitors.

Project pipeline



The data set

We were provided by a medium size data set that consists of 11 features per customer which are:

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
PolicySalesChannel	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

the dataset has the info of 381109 customer, Some snippets from the set below:

id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
1	Male	44	1	28	0	> 2 Years	Yes	40454	26	217	1
2	Male	76	1	3	0	1-2 Year	No	33536	26	183	0
3	Male	47	1	28	0	> 2 Years	Yes	38294	26	27	1
4	Male	21	1	11	1	< 1 Year	No	28619	152	203	0
5	Female	29	1	41	1	< 1 Year	No	27496	152	39	0
6	Female	24	1	33	0	< 1 Year	Yes	2630	160	176	0
7	Male	23	1	11	0	< 1 Year	Yes	23367	152	249	0
8	Female	56	1	28	0	1-2 Year	Yes	32031	26	72	1
9	Female	24	1	3	1	< 1 Year	No	27619	152	28	0
10	Female	32	1	6	1	< 1 Year	No	28771	152	80	0
11	Female	47	1	35	0	1-2 Year	Yes	47576	124	46	1

Analysis and visualization

Statistics

Index	Age	Driving_License	Region_Code	Previously_Insurec	Annual_Premium	olicy_Sales_Chann	Vintage	Response
count	381109	381109	381109	381109	381109	381109	381109	381109
mean	38.8226	0.997869	26.3888	0.45821	30564.4	112.034	154.347	0.122563
std	15.5116	0.0461095	13.2299	0.498251	17213.2	54.204	83.6713	0.327936
min	20	0	0	0	2630	1	10	0
25%	25	1	15	0	24405	29	82	0
50%	36	1	28	0	31669	133	154	0
75%	49	1	35	1	39400	152	227	0
max	85	1	52	1	540165	163	299	1

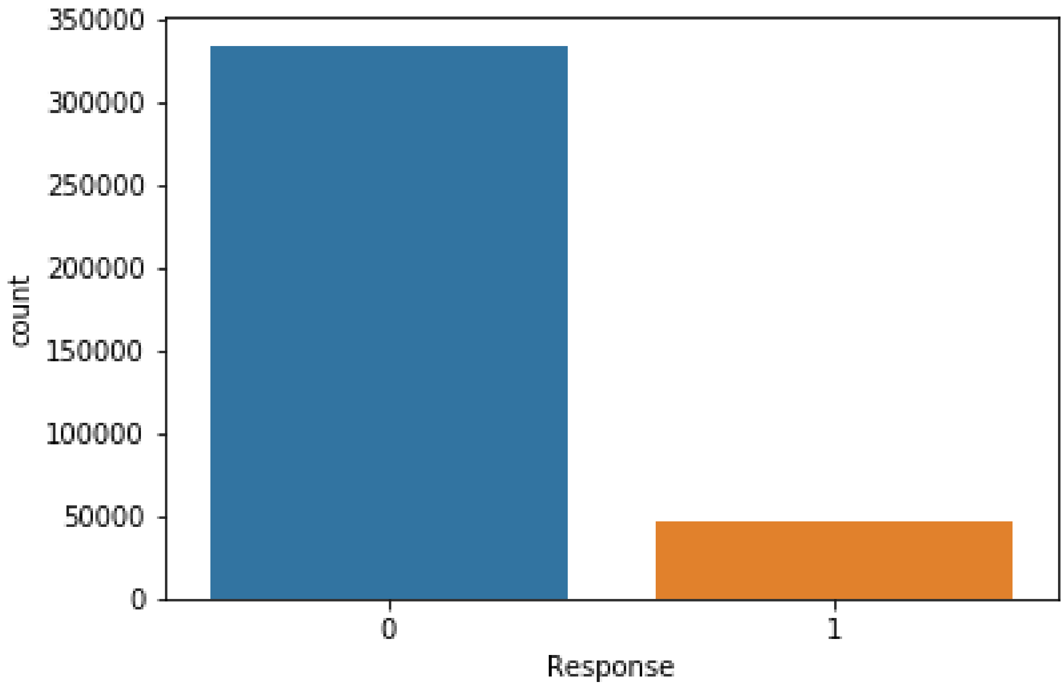
Correlation between variables

Index	Age	Driving_License	Region_Code	Previously_Insured	Annual_Premium	Policy_Sales_Channel	Vintage	Response	Gender_Male	Vehicle_Age_< 1 Year	Vehicle_Age_> 2 Years	Vehicle_Damage_Yes
Age	1	-0.079782	0.042574	-0.254682	0.067507	-0.577826	-0.00126408	0.111147	0.145545	-0.787775	0.220694	0.267534
Driving_License	-0.079782	1	-0.00108088	0.0149694	-0.0119065	0.0437305	-0.000848049	0.0101552	-0.0183738	0.0402149	-0.00621147	-0.016622
Region_Code	0.042574	-0.00108088	1	-0.0246588	-0.0105875	-0.0424202	-0.00274963	0.0105699	0.000604179	-0.0442504	0.0145546	0.028235
Previously_Insured	-0.254682	0.0149694	-0.0246588	1	0.00426876	0.219381	0.00253679	-0.34117	-0.0819322	0.358773	-0.191352	-0.824143
Annual_Premium	0.067507	-0.0119065	-0.0105875	0.00426876	1	-0.113247	-0.000608417	0.0225747	0.00367274	-0.0225554	0.0619177	0.00934929
Policy_Sales_Chann…	-0.577826	0.0437305	-0.0424202	0.219381	-0.113247	1	1.84994e-06	-0.139042	-0.111159	0.571516	-0.146238	-0.224377
Vintage	-0.00126408	-0.000848049	-0.00274963	0.00253679	-0.000608417	1.84994e-06	1	-0.00105037	-0.0025169	0.00241032	0.000600056	-0.00206437
Response	0.111147	0.0101552	0.0105699	-0.34117	0.0225747	-0.139042	-0.00105037	1	0.0524399	-0.209878	0.1093	0.3544
Gender_Male	0.145545	-0.0183738	0.000604179	-0.0819322	0.00367274	-0.111159	-0.0025169	0.0524399	1	-0.16628	0.0431546	0.0916059
Vehicle_Age_< 1 Year	-0.787775	0.0402149	-0.0442504	0.358773	-0.0225554	0.571516	0.00241032	-0.209878	-0.16628	1	-0.18275	-0.370778
Vehicle_Age_> 2 Years	0.220694	-0.00621147	0.0145546	-0.191352	0.0619177	-0.146238	0.000600056	0.1093	0.0431546	-0.18275	1	0.206961
Vehicle_Damage_Yes	0.267534	-0.016622	0.028235	-0.824143	0.00934929	-0.224377	-0.00206437	0.3544	0.0916059	-0.370778	0.206961	1

This shows some:

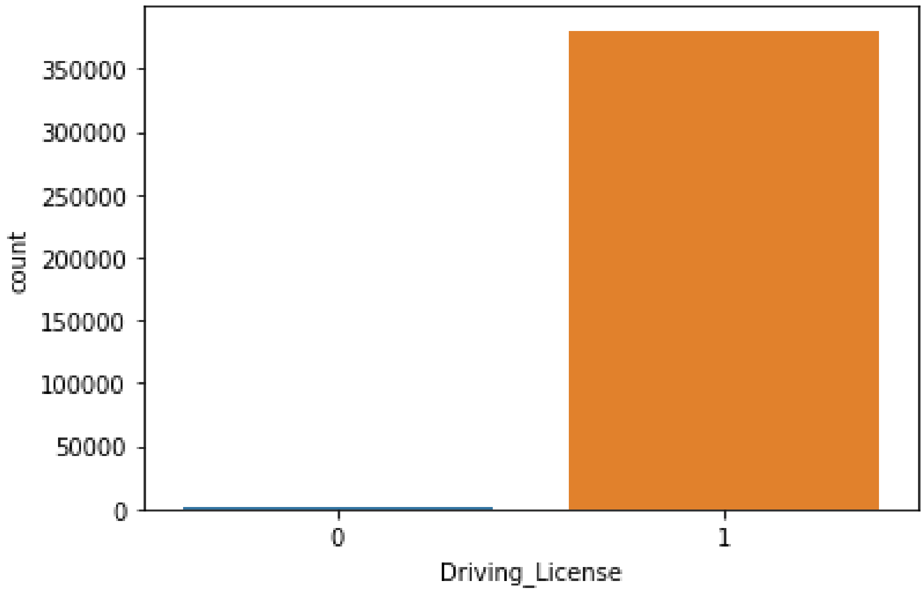
- Positive correlation between:
  - Age of the customer and Vehicle Age > 2
  - Vehicle Age < 1 and previously insured
  - Vehicle Age < 1 and policy sales channel
  - Response and vehicle damage
- Negative correlation between:
  - Vehicle damage and previously insured
  - Vehicle Age < 1 and vehicle damage
  - Response and previously insured

Visualization: We started by showing the response



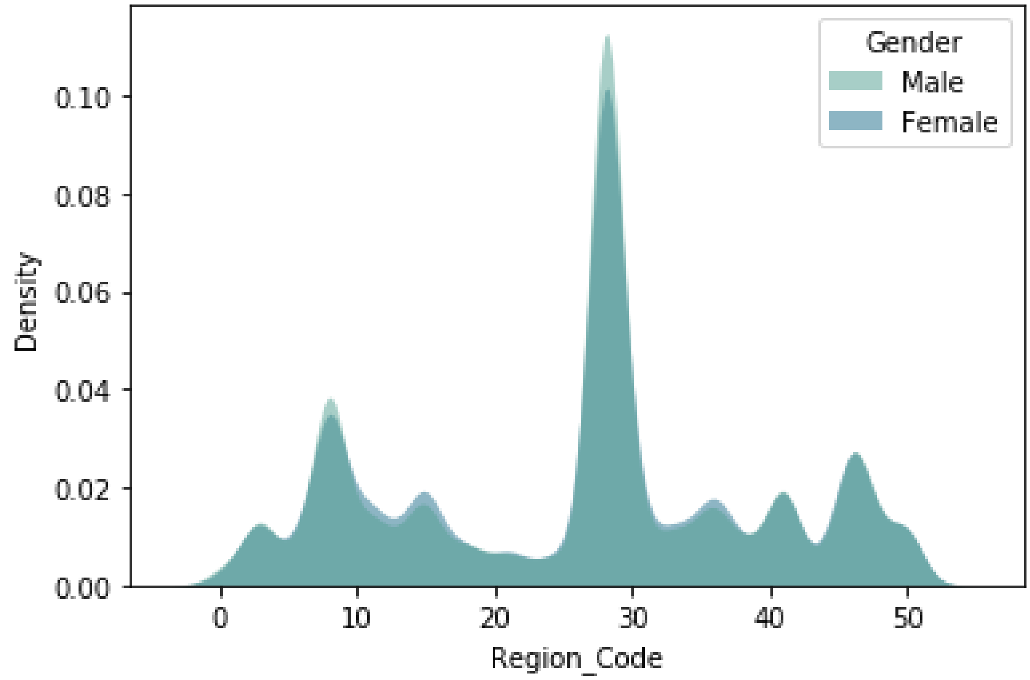
Clearly, the data is imbalanced.

percentage of customers having a **driving license**



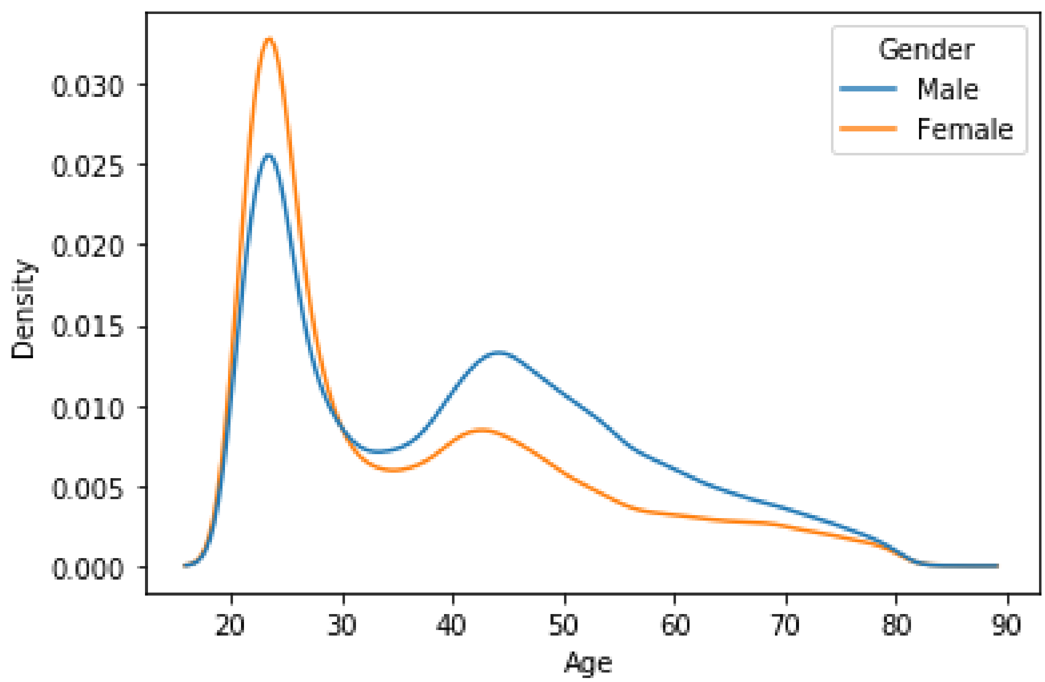
We see that 100% of our customers have a driving license.

Then, **Gender vs region** code

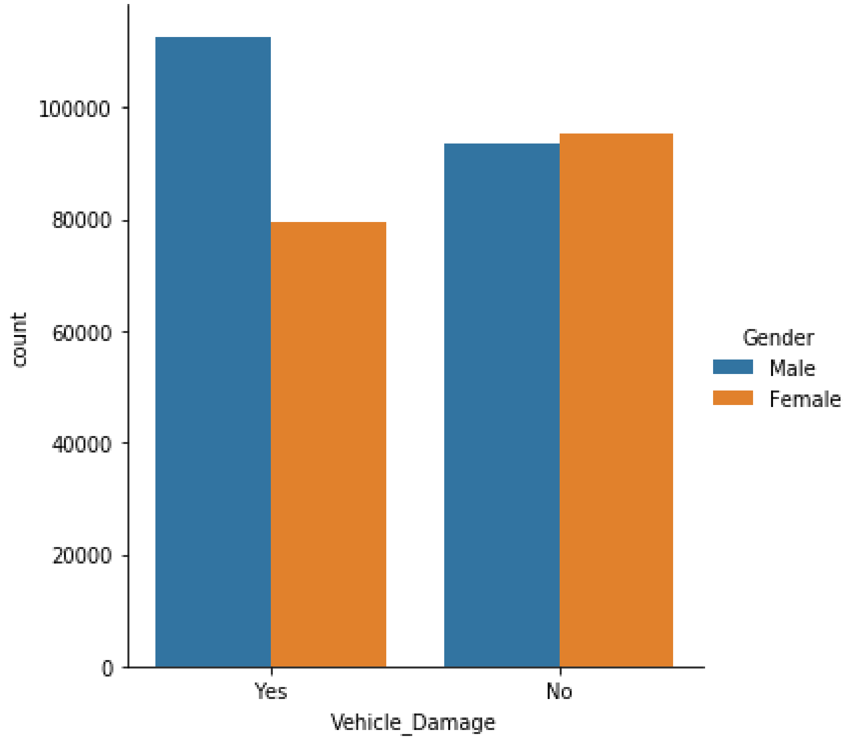


This plot shows a very high peak at regions 25-33 for both male and female customers

Then, **Gender vs age**

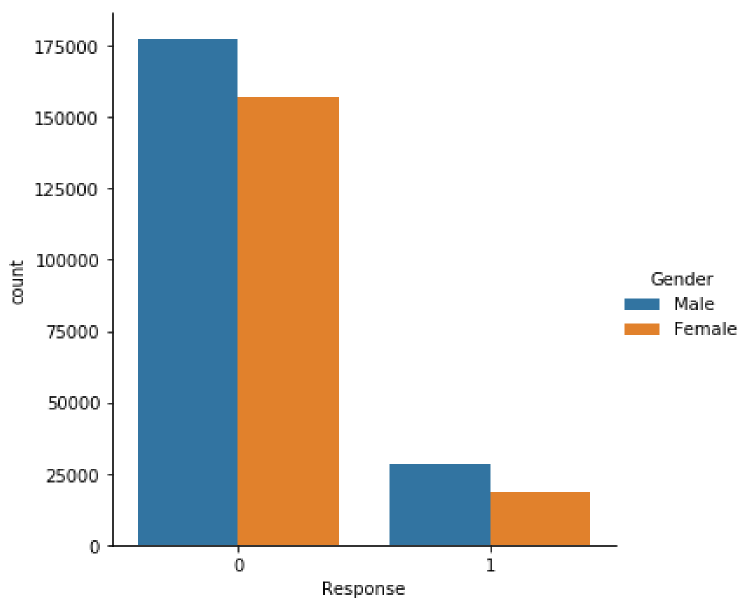


Then, the relation between the **gender and vehicle damage**:



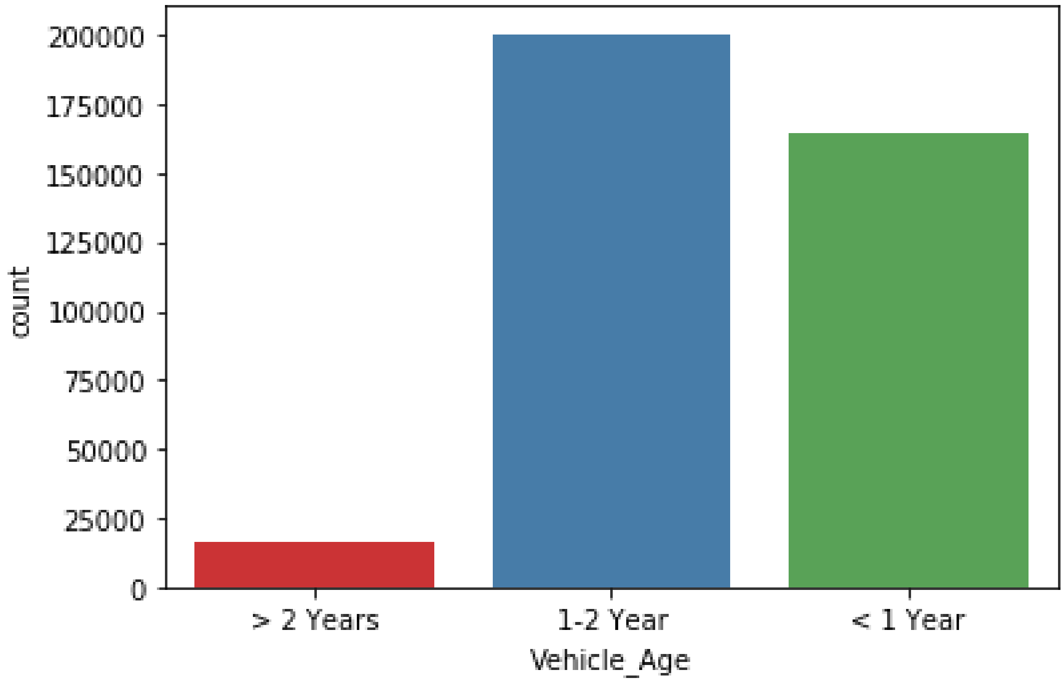
We notice that males have more chance to damage their car than the females (!)

Then we plotted the **gender VS response** (1 means interested, 0 means not interested)

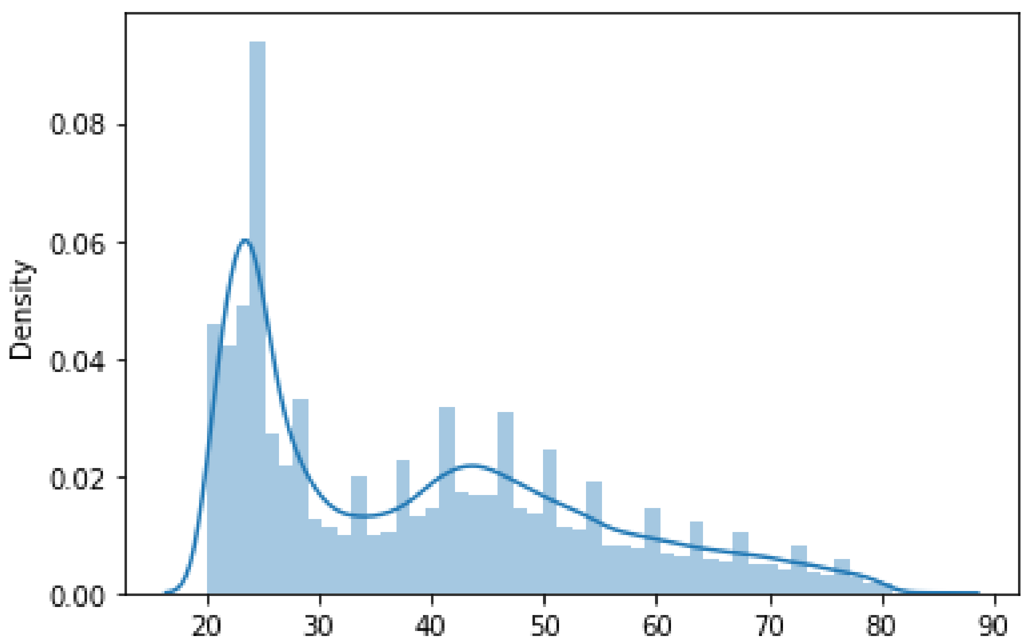


Among the customers who were interested, males were more likely to be interested.

Next, we explored the **age of vehicles**.

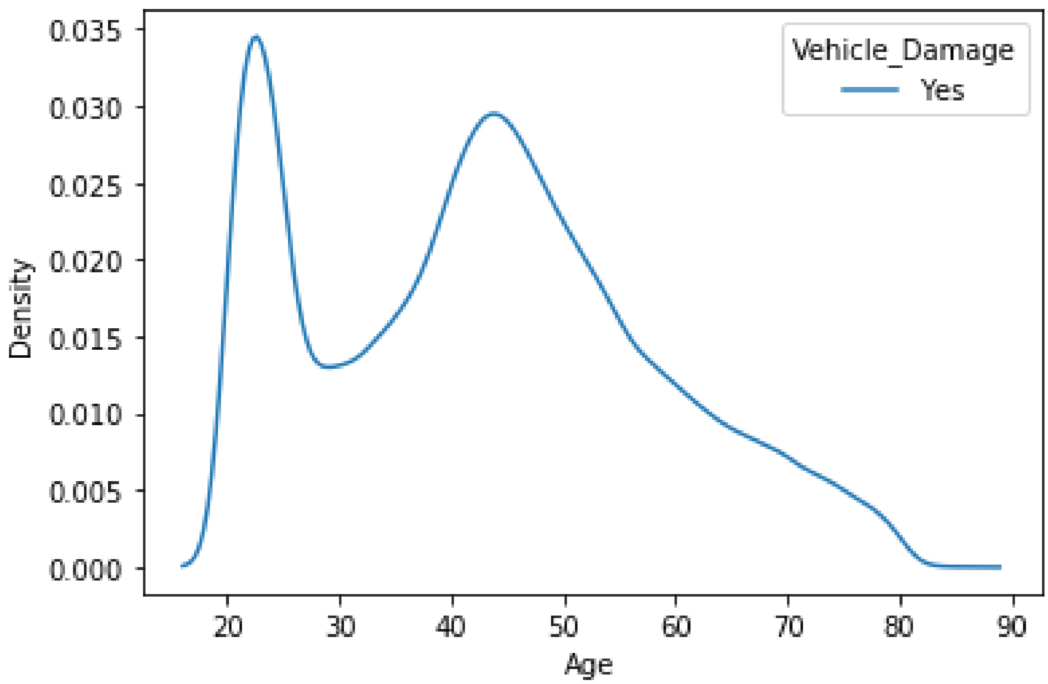


The distribution of customer **ages**

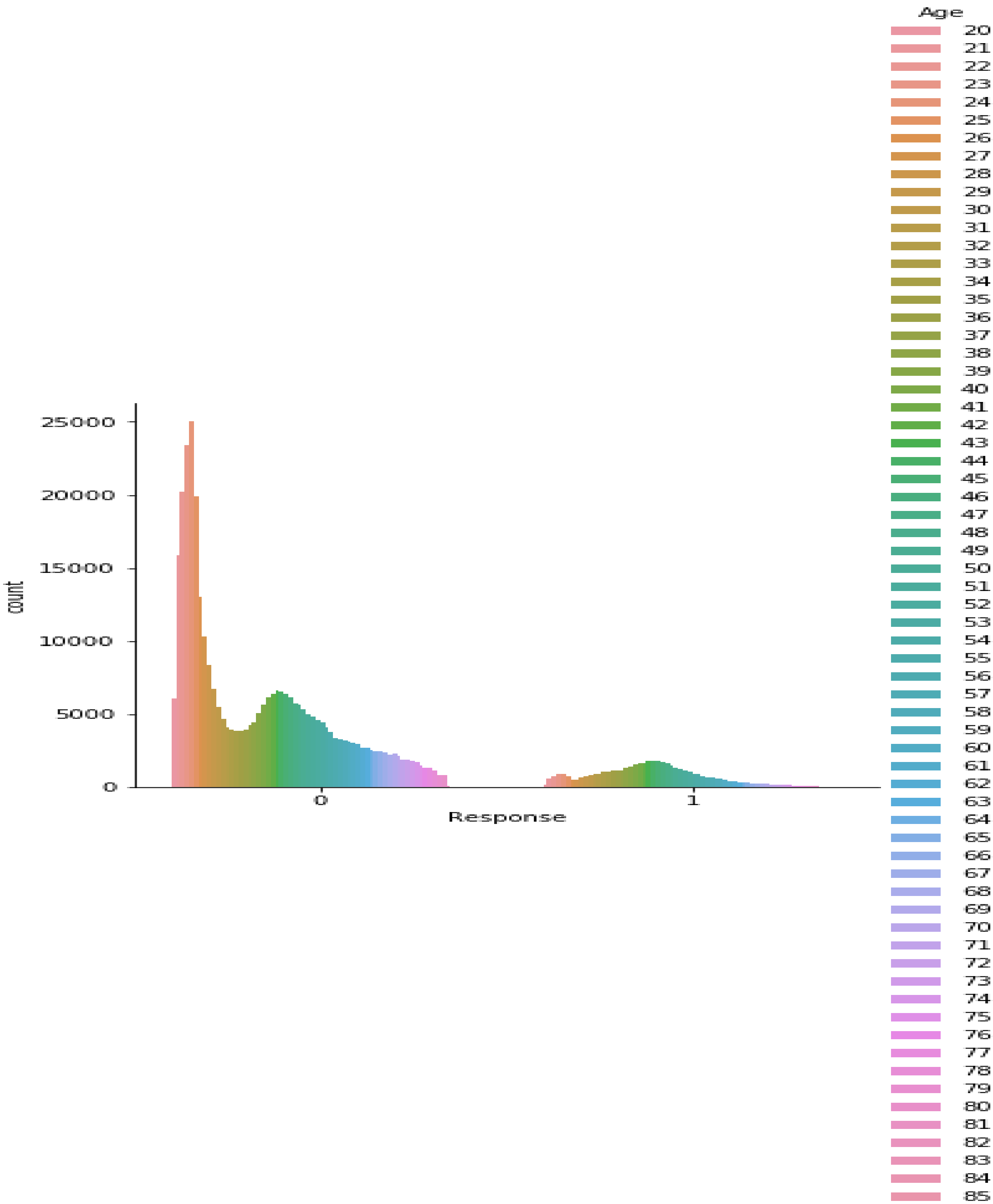


obviously, the distribution showed a high peak at younger ages (20 - 33) and a smaller peak at grown persons (40 - 54).

Next, we explored customer **age VS vehicle damage**.

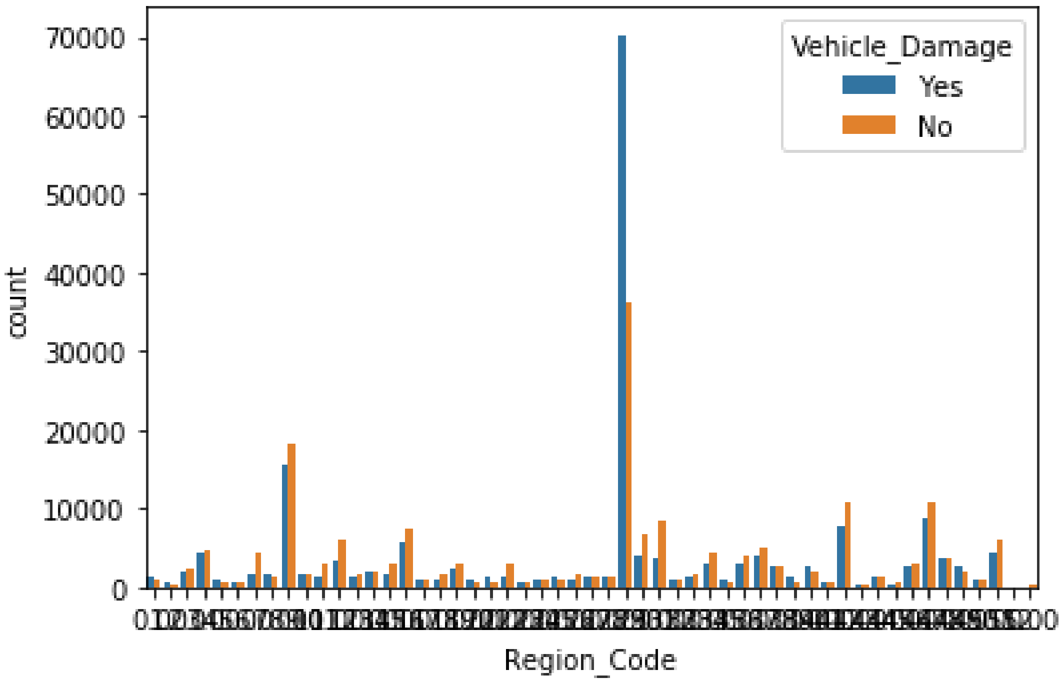


Then, **age vs response** (remember 1 means interested, 0 means not interested)



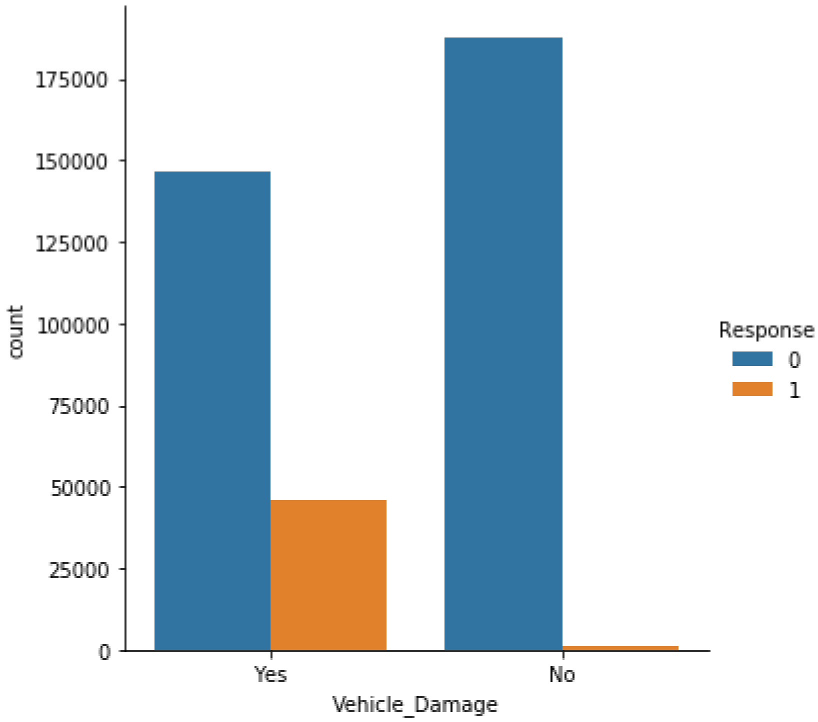
This plot shows some peak for positive response at ages [29-60]

Then, we showed **Region_code VS Vehicle_Damage**

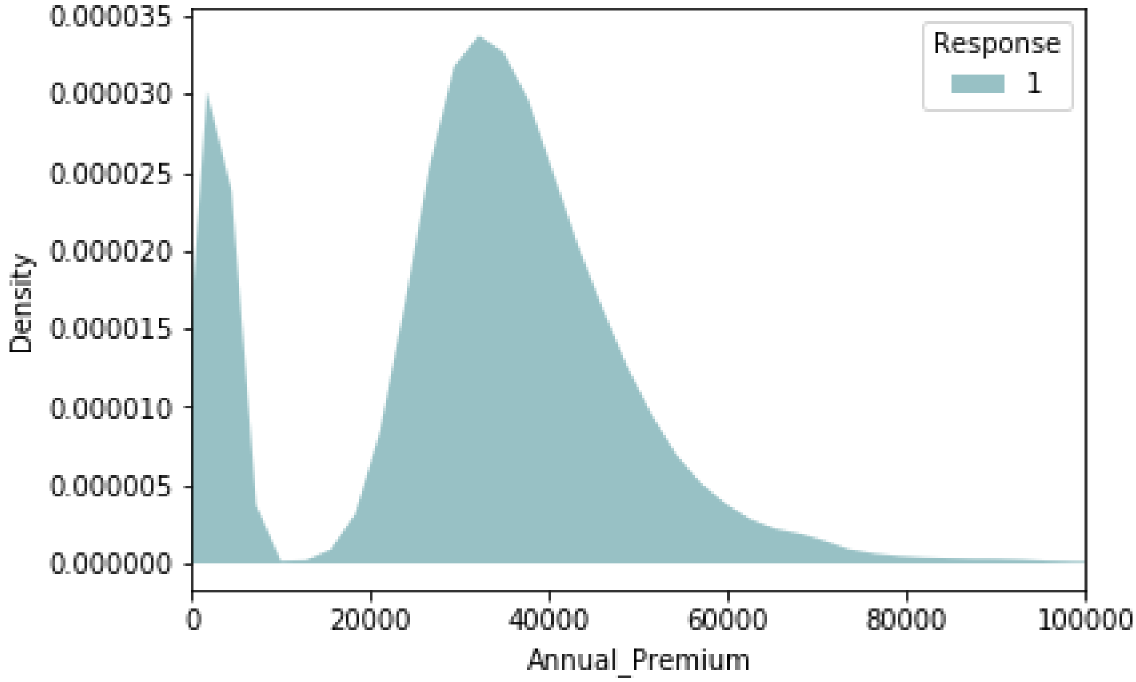


high peak of damaged vehicles at regions 25-33

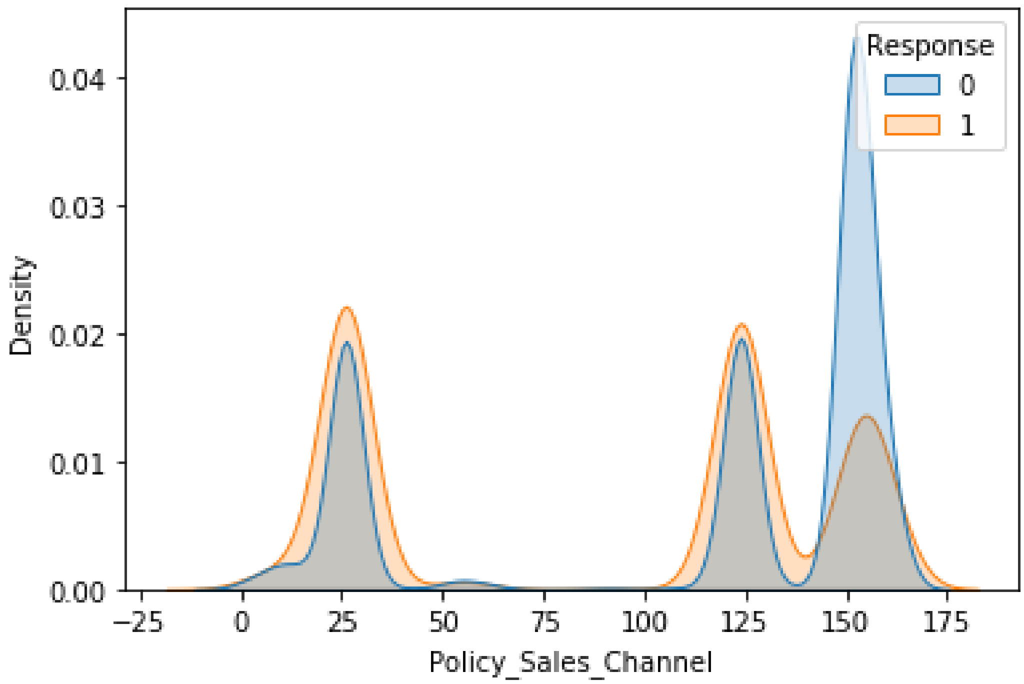
then vehicle damage vs response



this shows a relation between damage and being interested in vehicle insurance annual premium vs positive response

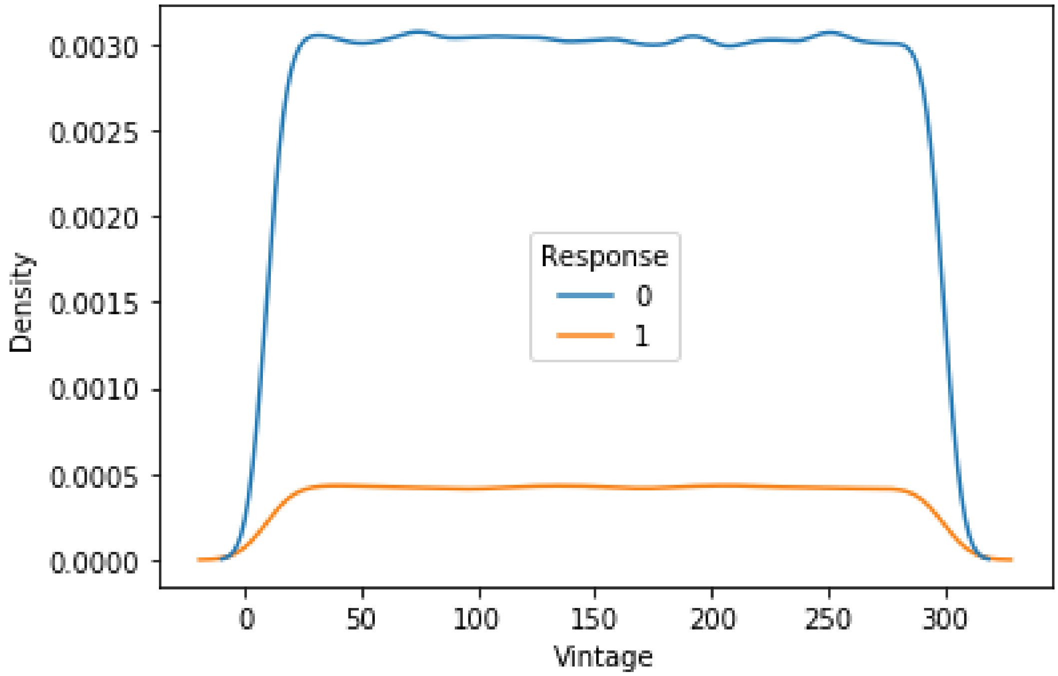


Policy channel vs response

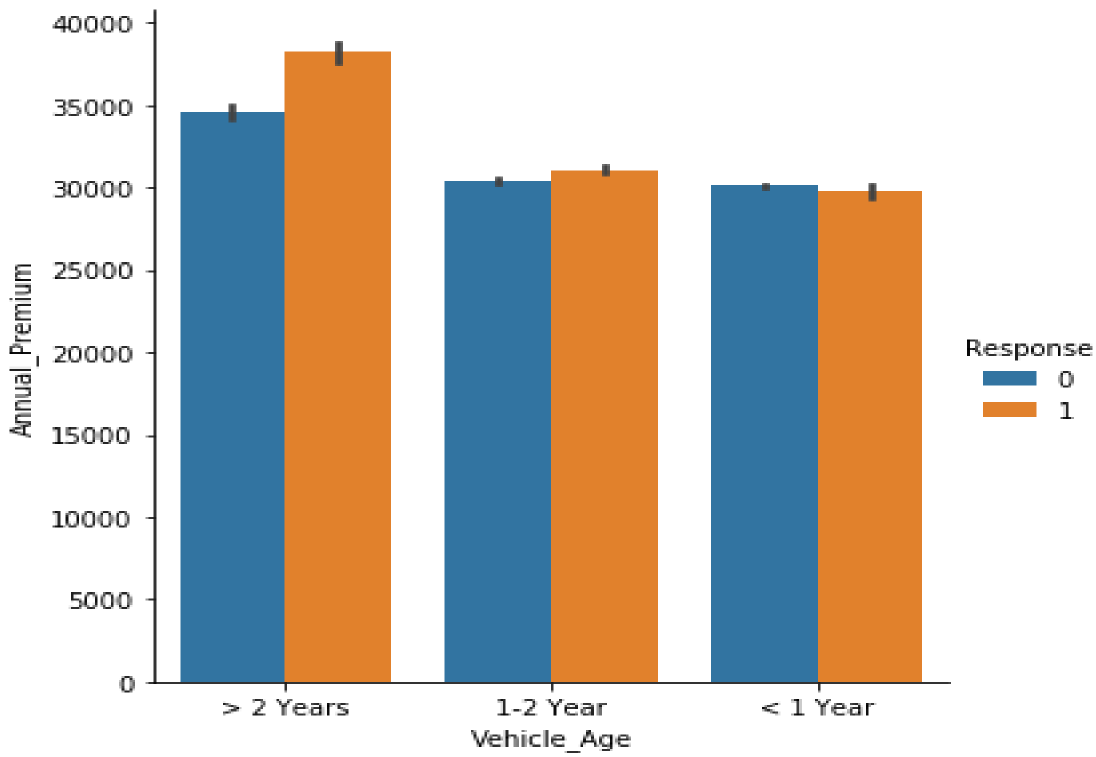


This shows peaks at channels 25 and 125 for positive response.

Vintage vs response

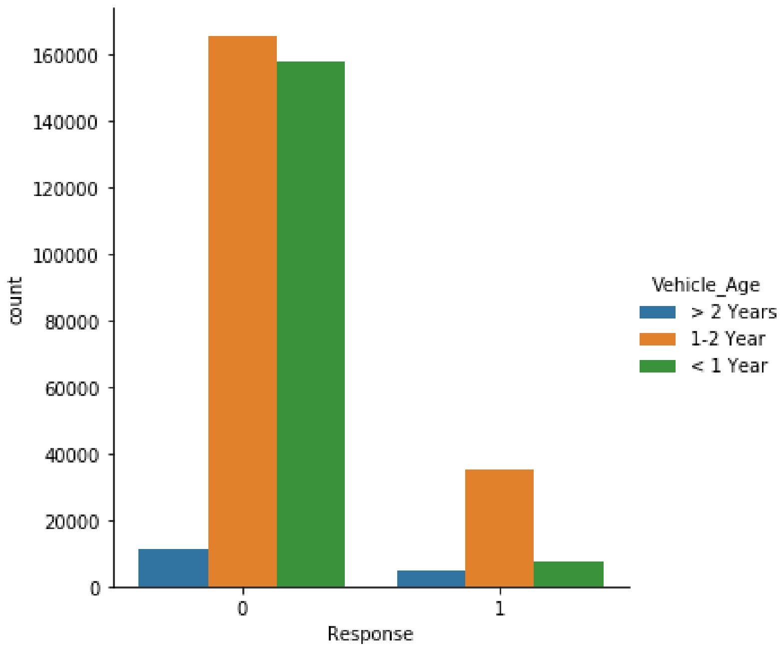


Vehicle age vs annual premium vs response

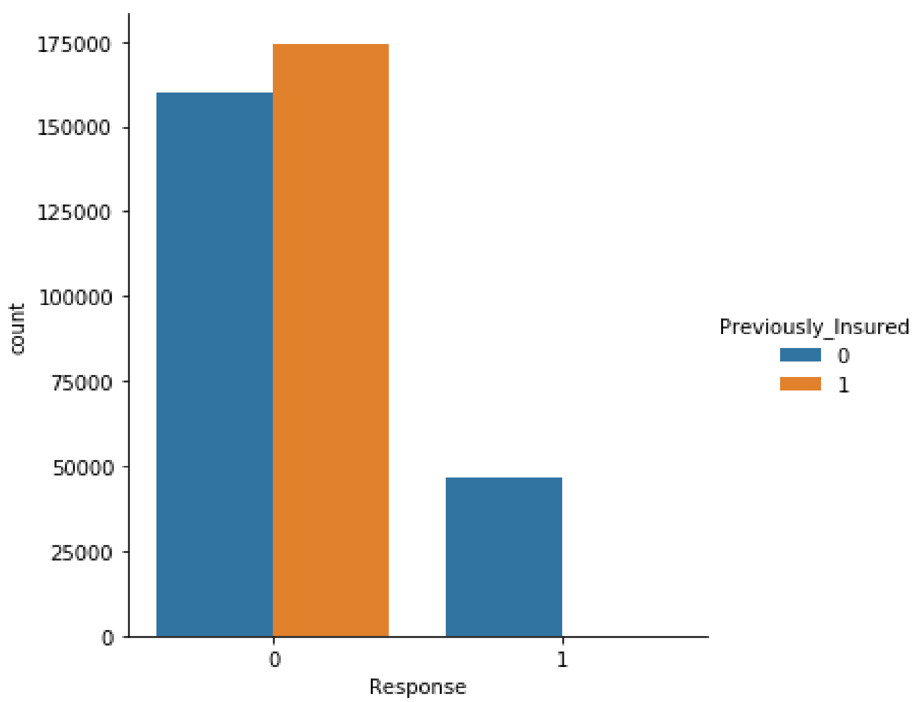


This shows that customers having cars with 2+ yeas have higher chance of being interested

Vehicle age VS response



Previously insured VS response



this shows that all interested customers hasn't insured for their cars before "interesting"

Analysis results:

We have imbalanced data.

We can see that the interested customer might have these features:

- Doesn’t have insurance
- Have a damaged vehicle
- His vehicle age is 1-2 years
- His age is between 30 and 55 years old
- his annual premium between 30K and 40K
- He is more likely to use channels such as channel 25 or channel 125 than channel 155

Preprocessing:

First thing to notice is there is some categorial variables, these must be factorized.
After factorization, we found another problem that some variables have large values that might affect our model, so we scaled all the variables to lie in the range [0,1].
The last problem that might affect the model is the imbalanced data, as we saw above
So, we balanced it using the **smote** technique.
Shape of the dataset before SMOTE: **(381109, 11)**
Shape of the dataset after SMOTE: **(668798, 11)**
Balance of positive and negative response after smote (%):
0 -> 50.0
1 -> 50.0

Models’ training

We tried several models to find the best fit for the data, for each model we split the data with the ratio 75% training and 25% testing.
below are some results:

Models without preprocessing:

Model 1: Logistic regression

The results were as follows:

On training set:

	precision	recall	accuracy
0	0.88	1	0.88
1	0.0	0.0	

On test set:

	precision	recall	accuracy
0	0.88	1	0.88
1	0.0	0.0	

Model 2: Random Forest

On training set:

	precision	recall	accuracy
0	1	1	100%
1	1	1	

On test set:

	precision	recall	accuracy
0	0.89	0.97	0.87
1	0.37	0.12	

Model 3: KNN

On training set:

	precision	recall	accuracy
0	0.89	0.99	0.88
1	0.61	0.17	

On test set:

	precision	recall	accuracy
0	0.88	0.97	0.88
1	0.22	0.06	

Model 4: naïve bays

On training set:

	precision	recall	accuracy
0	0.91	0.89	0.82
1	0.31	0.35	

On test set:

	precision	recall	accuracy
0	0.91	0.89	0.82
1	0.31	0.35	

We notice quite good accuracy, but if we looked closely, we find the recall for response = 1 is 0 or a very small value, that's because the data is skewed towards the negative response, the model learnt that every example is a 0 response, so this model cannot be used.

Now we trained the models on the preprocessed, balanced data:

Models with preprocessing

Model 1: Logistic regression:

On training set:

	precision	recall	accuracy
0	0.96	0.59	0.78
1	0.71	0.97	

On test set:

	precision	recall	accuracy
0	0.93	0.88	0. 78
1	0.88	0.94	

Model 2: Random Forest:

On training set:

	precision	recall	accuracy
0	1	1	100%
1	1	1	

On test set:

	precision	recall	accuracy
0	0.93	0.88	0.91
1	0.88	0.94	

Model 3: KNN

On training set:

	precision	recall	accuracy
0	0.98	0.81	0.90
1	0.84	0.99	

On test set:

	precision	recall	accuracy
0	0.96	0.75	0.86
1	0.80	0.97	

Model 4: naïve bays:

On training set:

	precision	recall	accuracy
0	0.96	0.59	0.78
1	0.71	0.98	

On test set:

	precision	recall	accuracy
0	0.96	0.59	0.78
1	0.71	0.98	

The performance improved for classifying positive response.

So eventually we can say that the best model is a random forest trained on a preprocessed, balanced dataset.