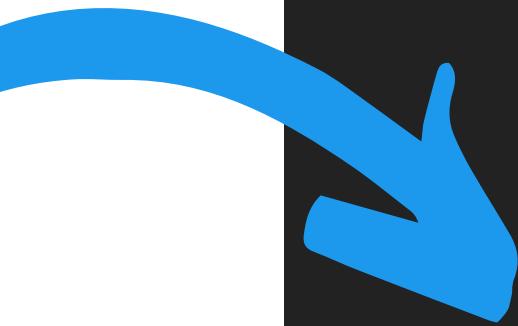


DARNELL KIKOO



Cross Health Insurance Prediction

FINAL PROJECT

Table of Contents

1

Business
Background

2

Data Understanding
and Exploratory
Data Analysis

3

Data Preprocessing

4

Modeling and
Evaluation

5

Business Case
Study

6

Conclusion and
Reccomendation

Business Background



Background

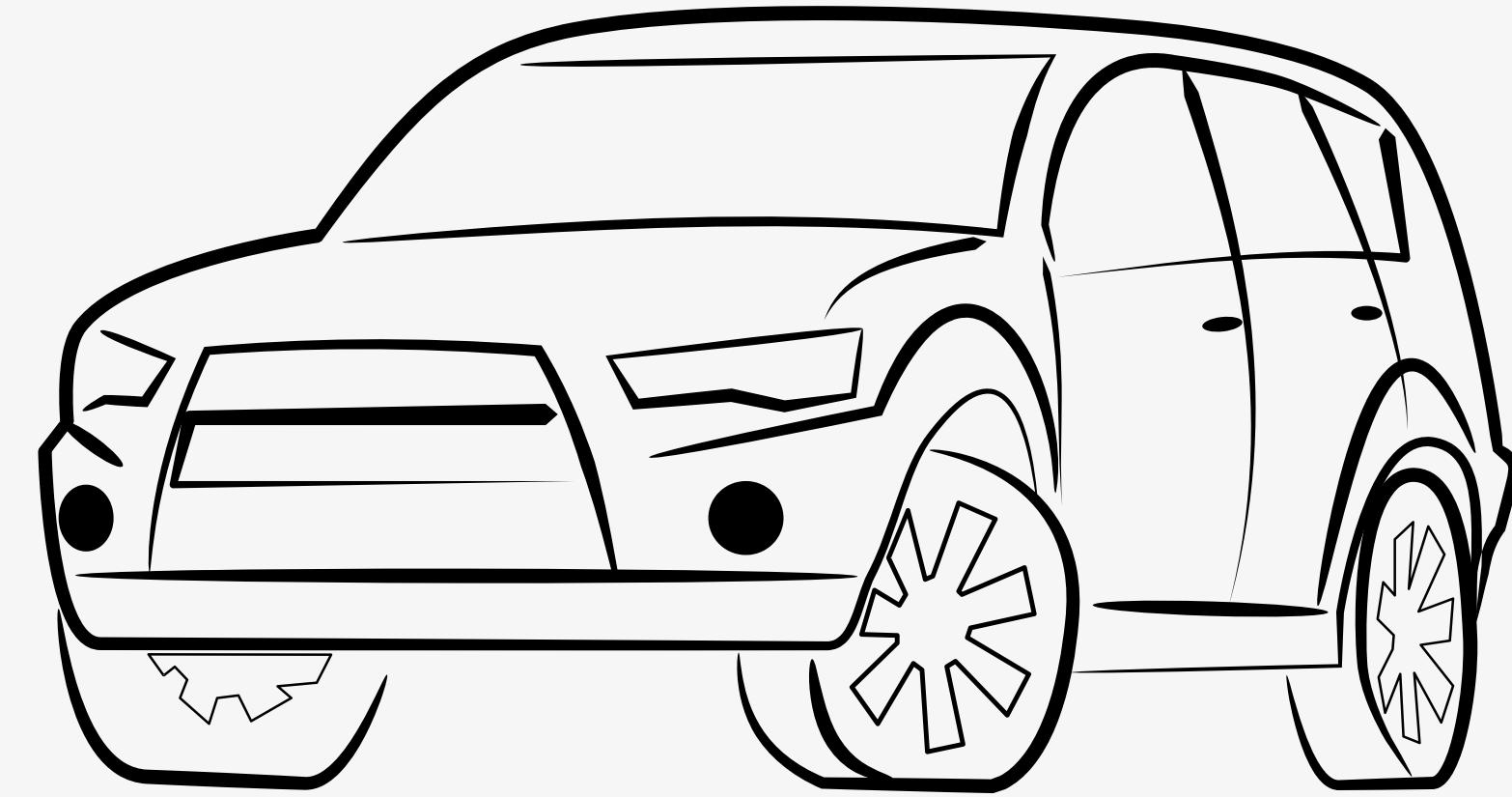


I'm a freelance data scientist, recruited to work at one of the **bigest insurance company** in India

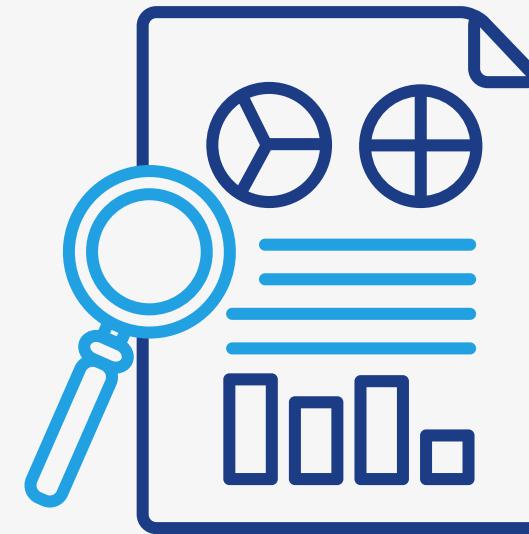
Our company wanted to increase the sales of vehicle insurance. Hence, I am asked to create a model to predict the possible reaction of policy holders if they are asked to buy one

Business Objectives

The objective of this project is to **predict** whether a policy holder would be **interested in insuring their vehicle**. Helping the company to increase their sales and minimizing the range of customers that should be contacted for sales



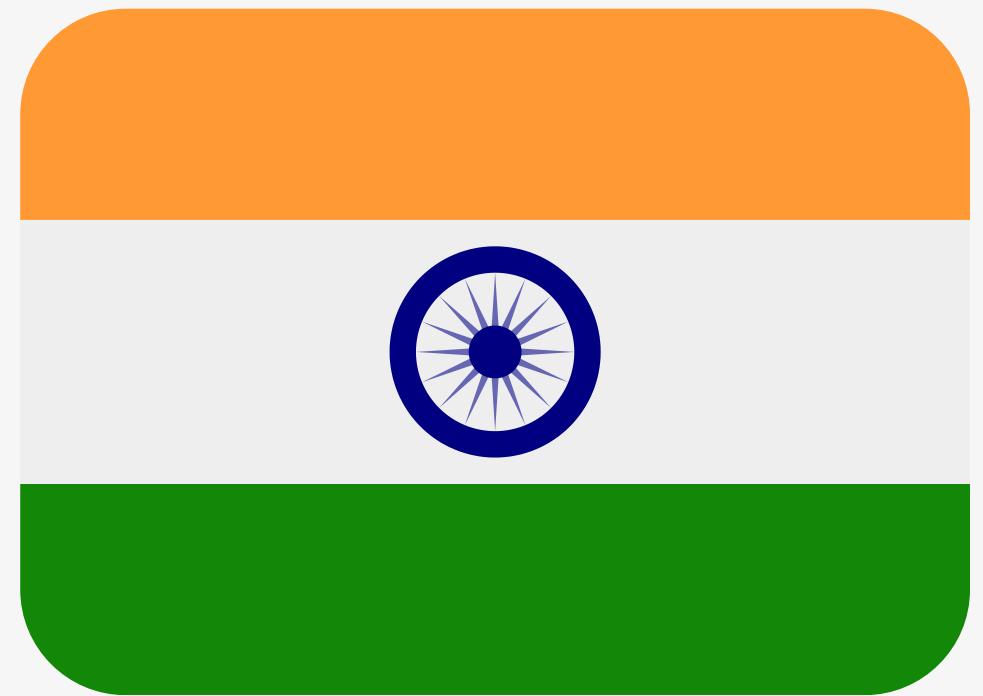
Output



The output of this project
is a **model** to predict the
customer's response
given its feature

The feature includes
Vehicle_Damage,
Previously_Insured,
Gender, etc.

Project Limitation



We have limited knowledge towards some features, such as region_code and anonymized policy_sales_channel

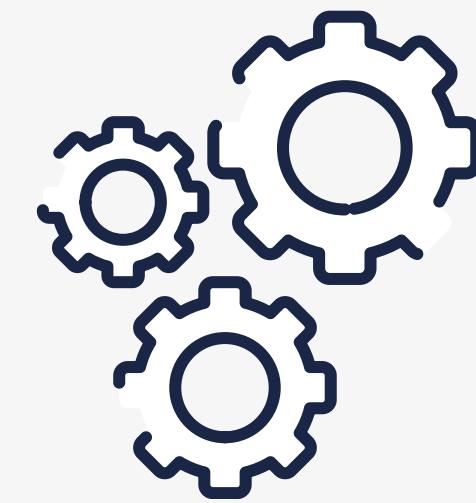
We also decided to remove customer rows who doesn't have Driving_License

Analytic Approach



Machine Learning

**Supervised Learning
(Binary Classification)**
Classify what response
(label) the customer would
give



Performance Measures

ROC_AUC
Precision, Recall, F1 and
Accuracy to support our
measurement

Data Understanding and Exploratory Data Analysis



Data Collection / Gathering



The dataset is provided by Kaggle, with
381109 Rows and 12 Columns

Column Explanation

No	Feature	Data Type	Distinct	Missing (%)	Description
1	id	int64	304887	0.0%	Unique ID for the customer
2	Gender	object	2	0.0%	Gender of the Customer
3	Age	int64	66	0.0%	Age of the customer
4	Driving_License	int64	2	0.0%	0:Customer does not have DL, 1:Customer already has DL
5	Region_Code	float64	53	0.0%	Unique code for the region of the customer
6	Previously_Insured	int64	2	0.0%	1:Customer already has Vehicle Insurance, 0:Customer doesn't have Vehicle Insurance
7	Vehicle_Age	object	3	0.0%	Age of the Vehicle
8	Vehicle_Damage	int64	2	0.0%	1:Customer got his/her vehicle damaged in the past. 0: Customer didn't get his/her vehicle damaged in the past
9	Annual_Premium	float64	46366	0.0%	The amount customer needs to pay as premium in the year
10	Policy_Sales_Channel	float64	152	0.0%	Anonymized Code for the channel of outreach to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
11	Vintage	int64	290	0.0%	Number of Days, Customer has been associated with the company

Divided Dataset

We split the data into the ratio of 80:20, where the 80% is used for development, and the 20% is used for final prediction



80%

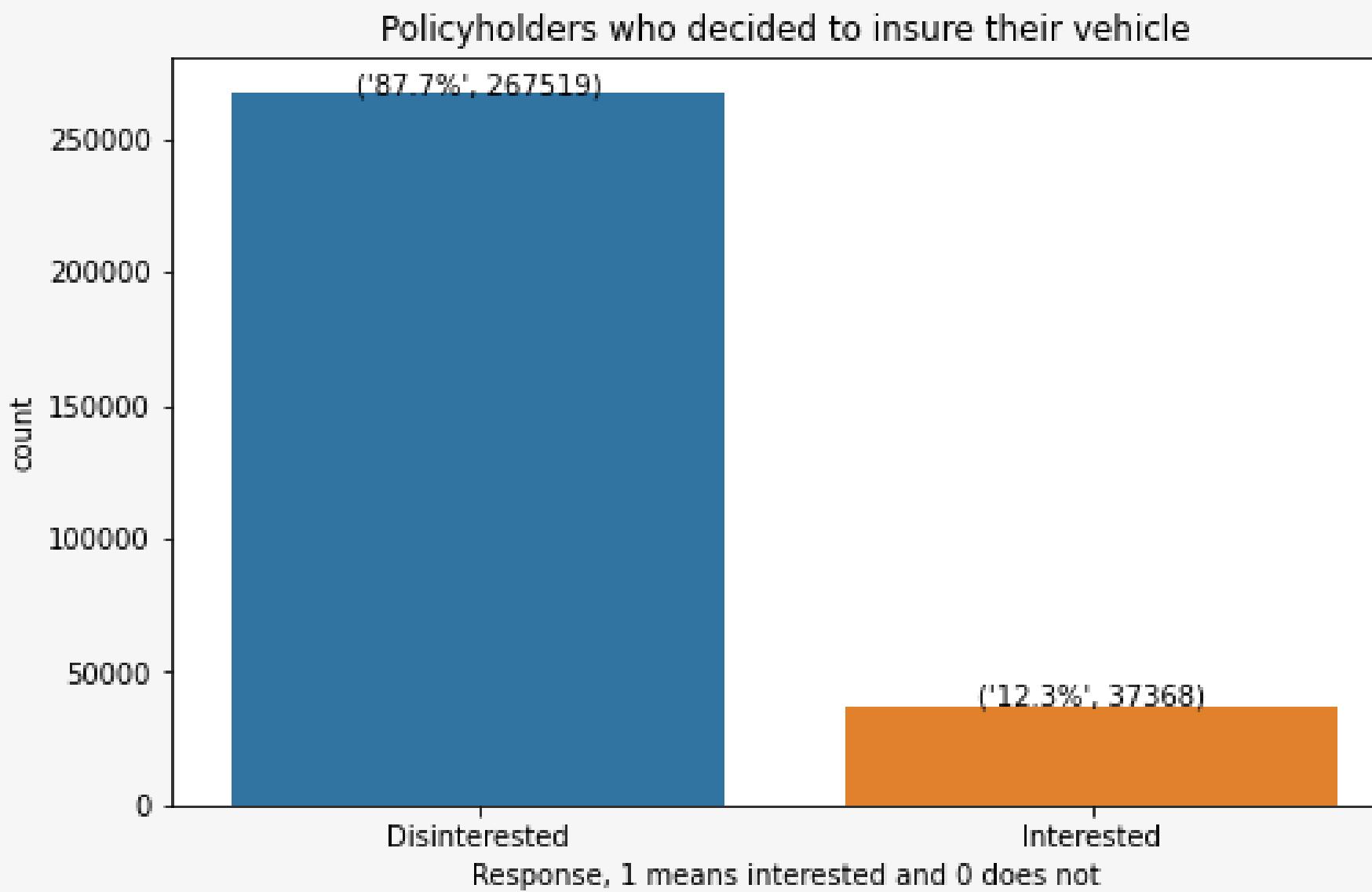


100%



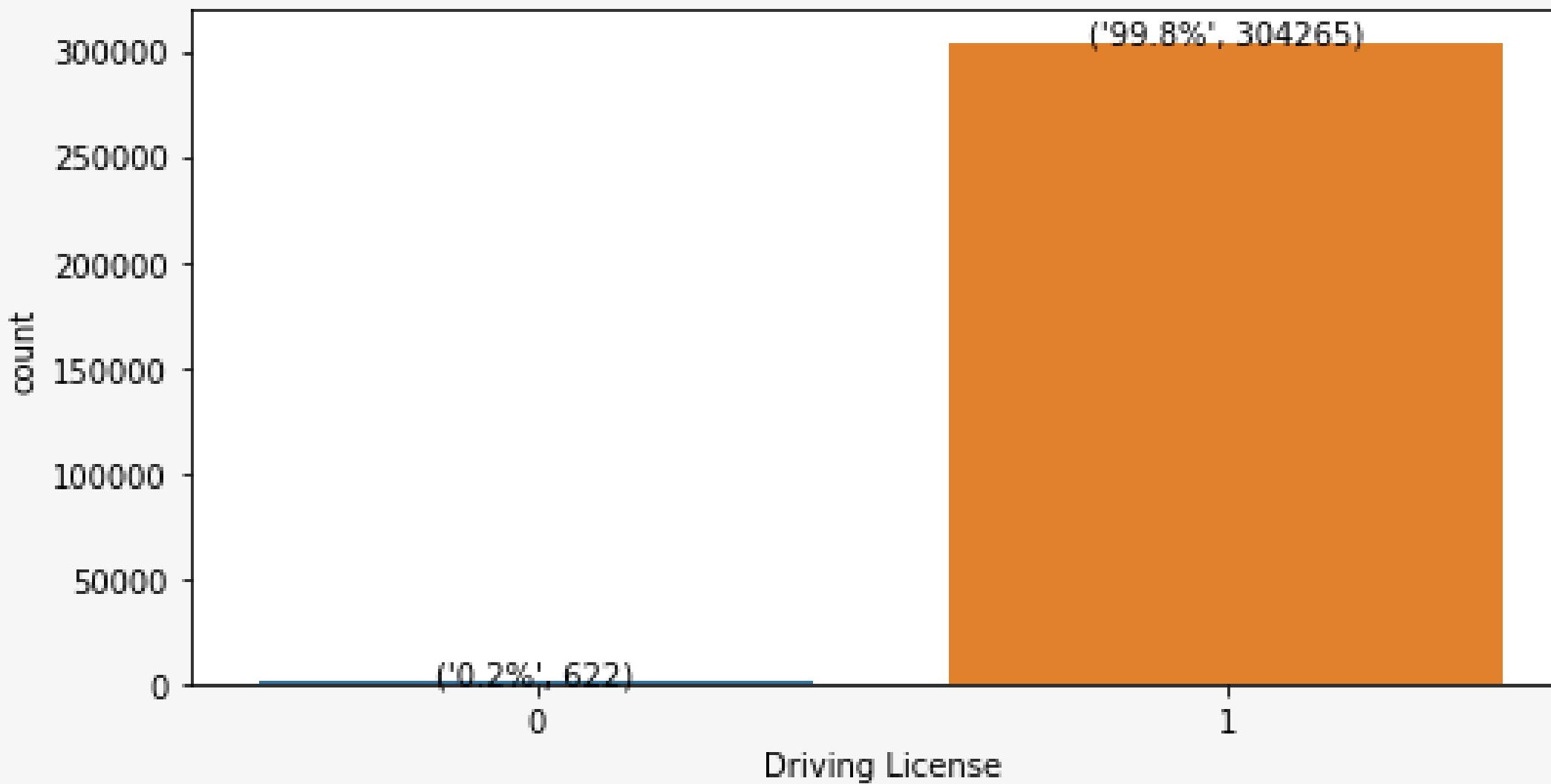
20%

Response Distribution



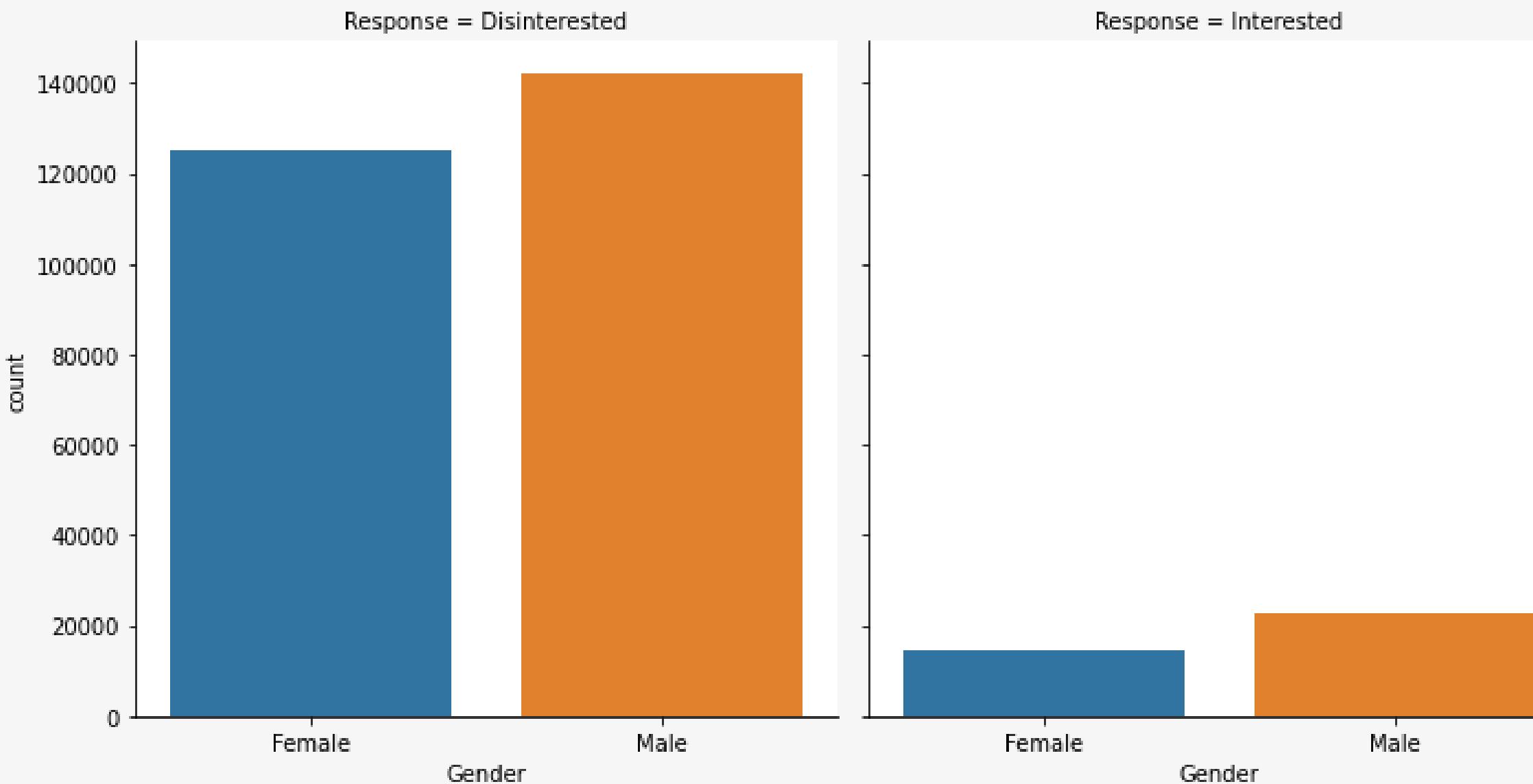
The chart shows the distribution of the response, which majority appeared to be disinterested. This explains that 87%++ policyholders don't seem to be interested in insuring their vehicle

Driving License



The chart shows majority (99.8%) of policyholders do have driving license. In fact, it is rare to see someone who buys vehicle license but doesn't have a driving license. Hence, we assume that this is caused by mistyping and will remove the rows without driving license

Gender on Response

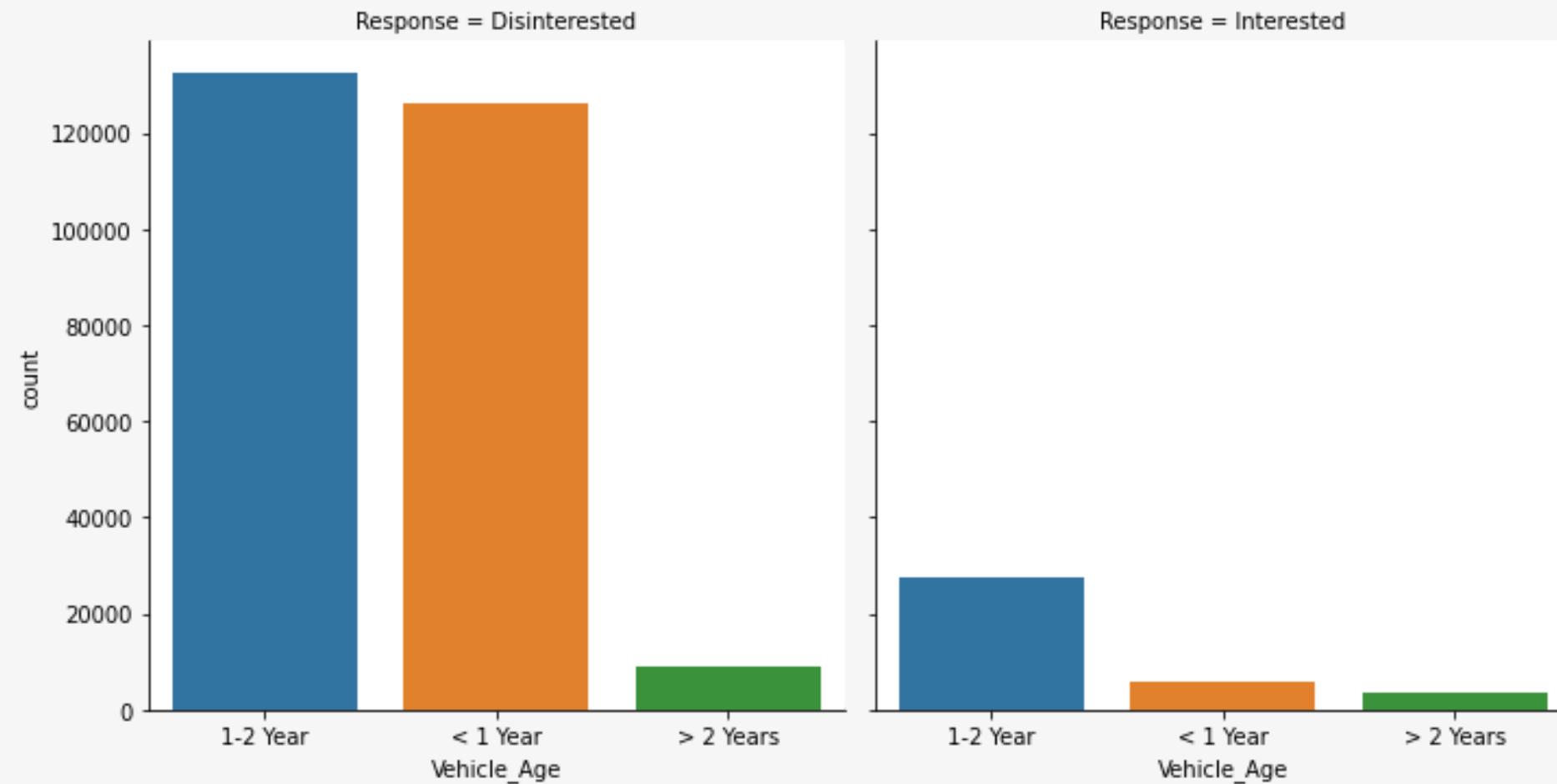


Total male buy Insurance : 13.813196541395326%
Total female buy Insurance : 10.421891187082485%

From the given dataset, males are more likely to buy vehicle insurance regardless of their age, with a bigger margin of 3.4% compared to Female.

We can say that male will give quite a big impact on the model

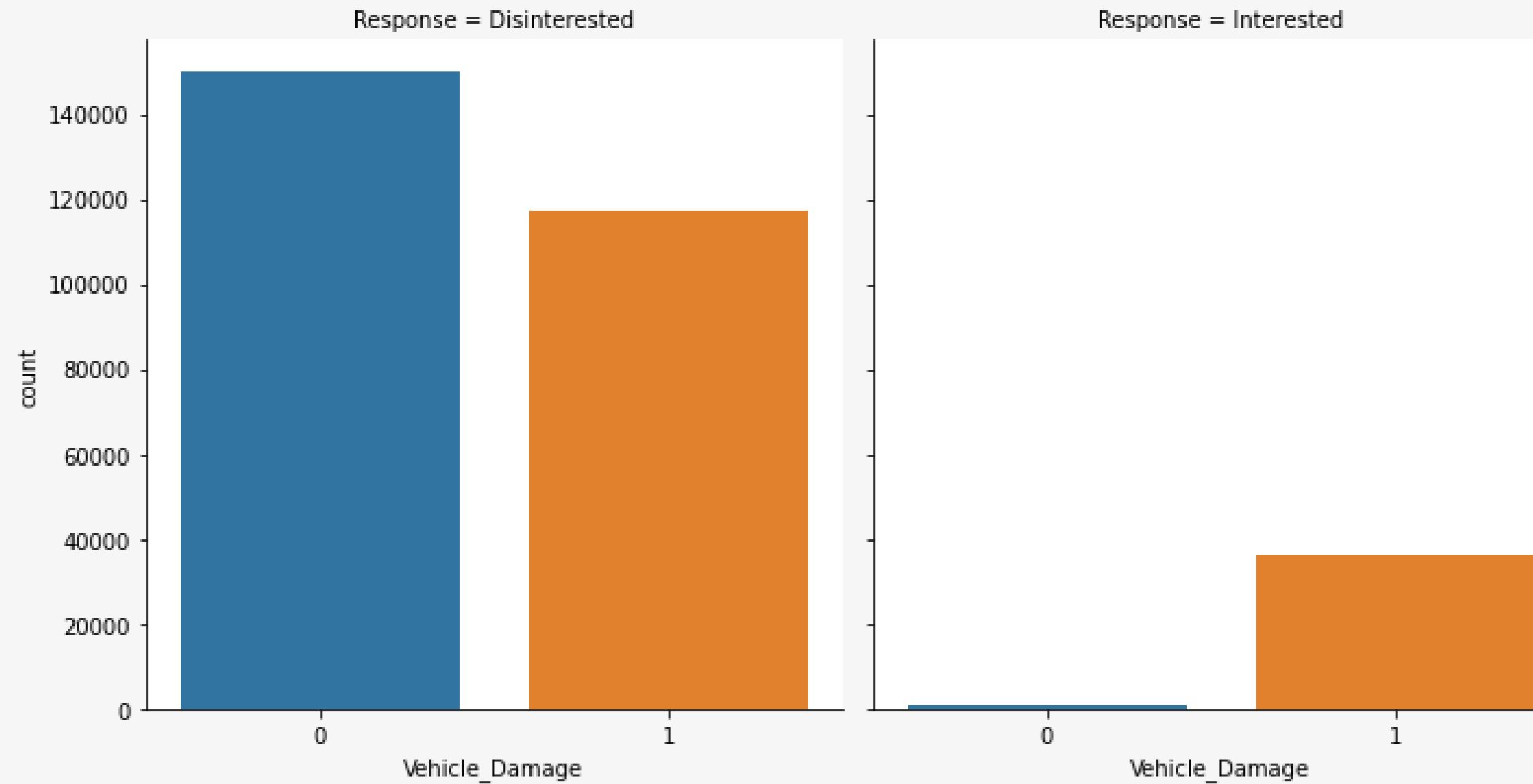
Vehicle Age on Response



Customer's Vehicle Age 1-2 Year and buy Insurance :
17.32 %
Customer's Vehicle Age Less Than 1 Year and buy Insurance :
4.42 %
Customer's Vehicle Age More Than 2 Years and buy Insurance:
29.63 %

Based on the vehicle age, there are more customers who have a vehicle aged between 0-2 years. However, customer whose car age is above 2 years have a bigger number of purchase on vehicle insurance

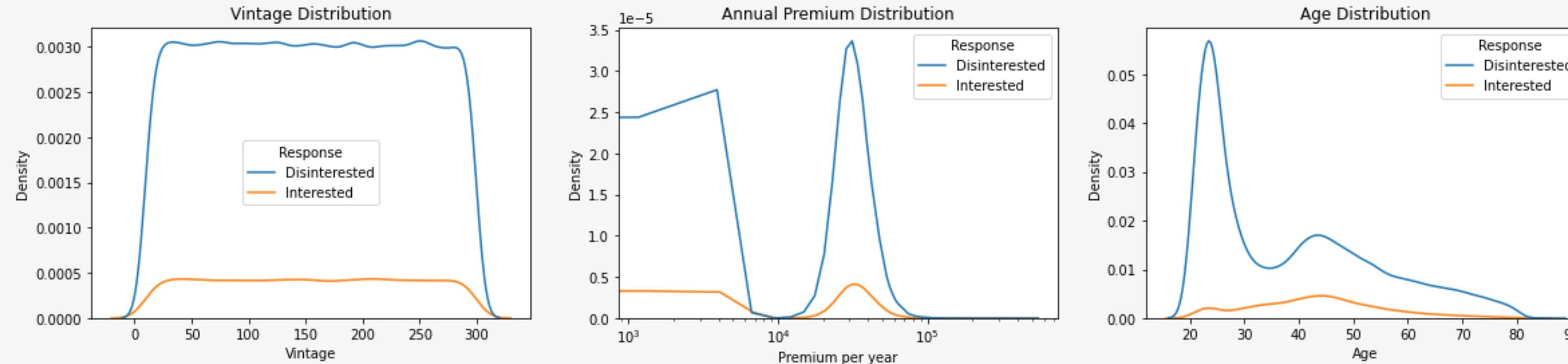
Vehicle Damage on Response



Customers who have ever had their vehicle damaged and buy Insurance: **23.74 %**
Customers who never had their vehicle damaged and buy Insurance : **0.52 %**

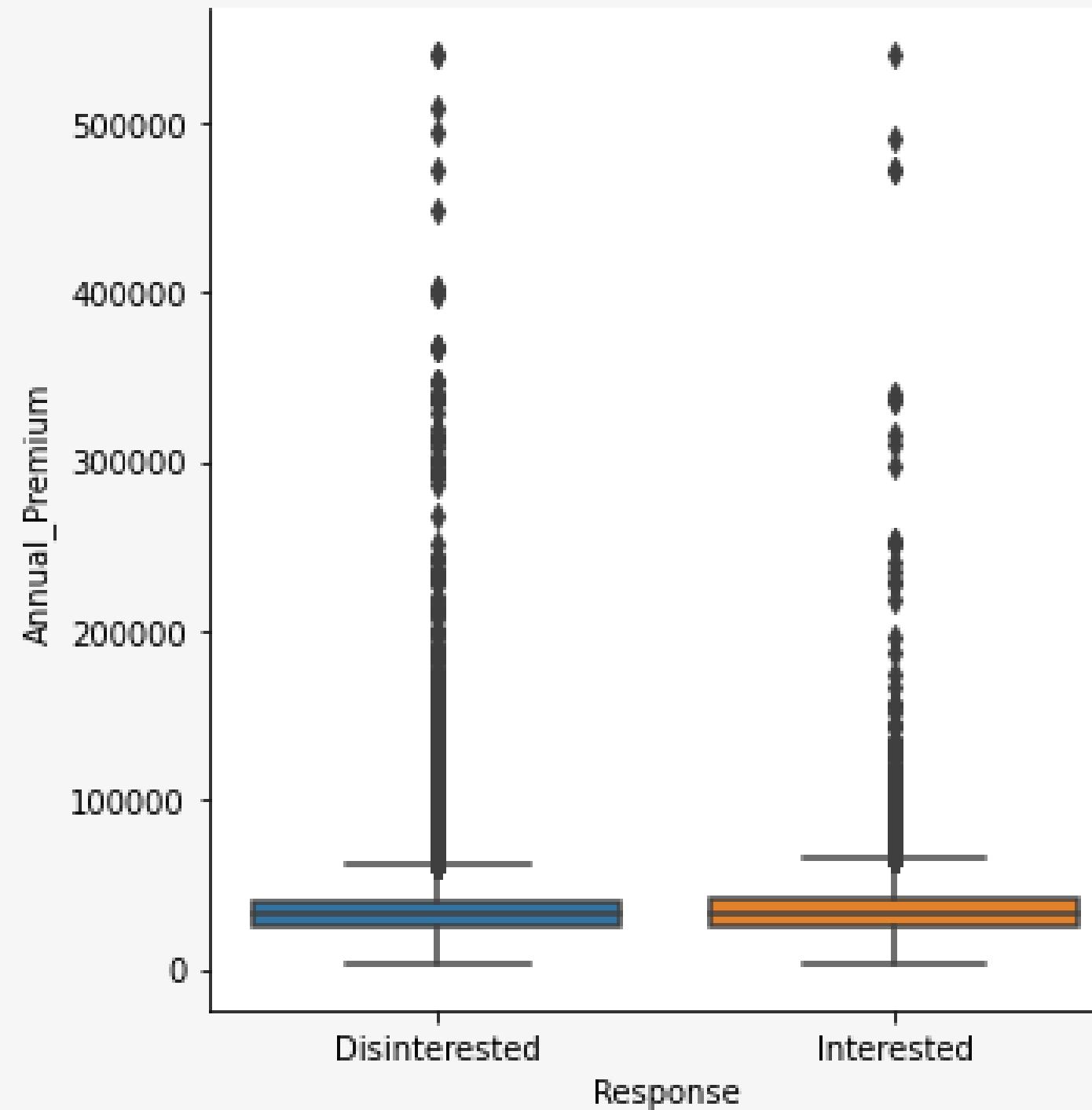
Majority of policyholders who buys vehicle insurance, are those who have ever had their vehicle damaged. This might be affected by the psychological trauma caused by unwanted accidents and damage their vehicle

Numerical Column Distribution



While the first 2 charts doesn't have any strong information, the third chart tells us that majority of customers who buy vehicle insurance were from the 40-50 Age range

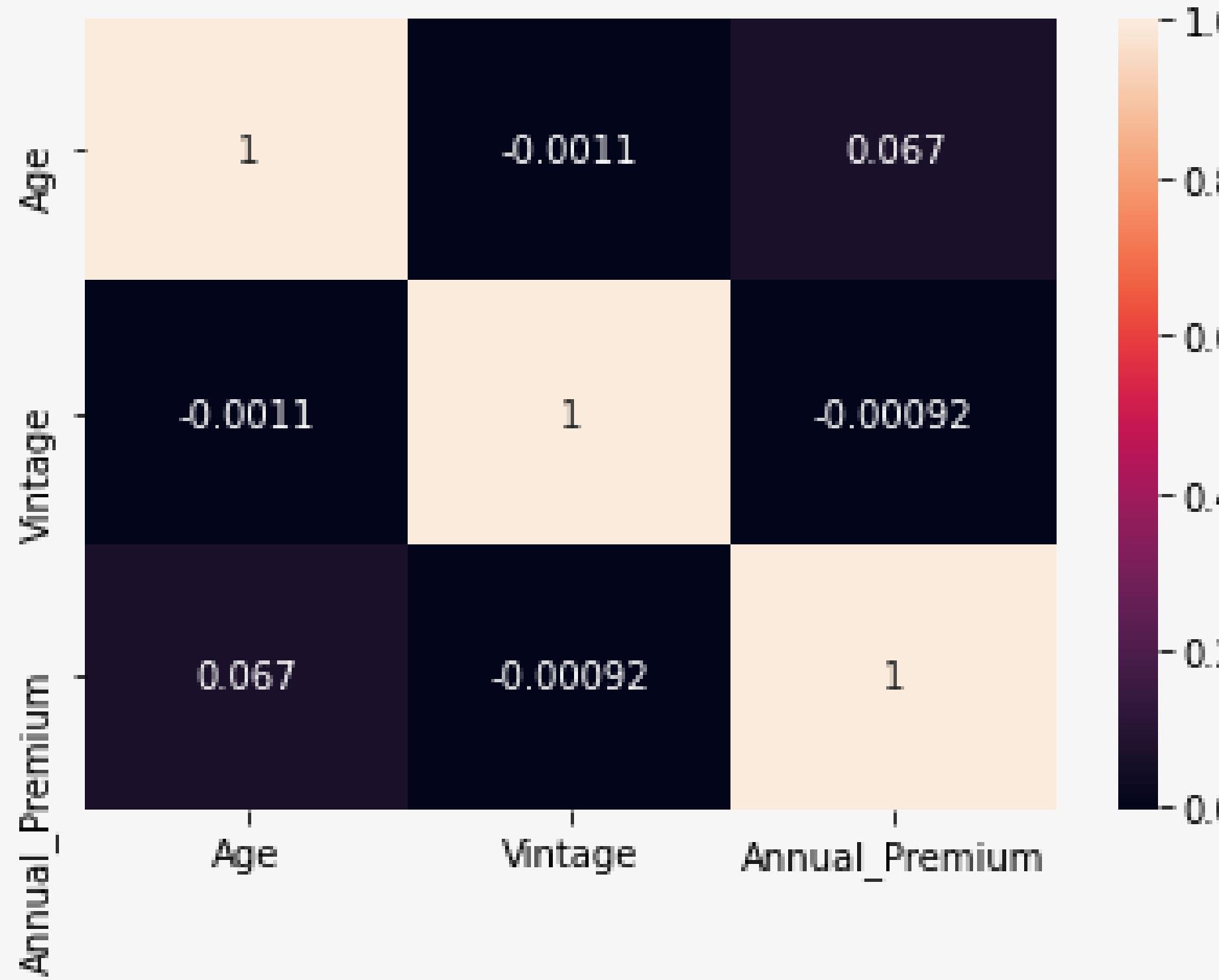
Customer's Annual Premium Distribution



The boxplot represents the outlier from the `Annual_Premium` feature. Most policyholders in India buys insurance below Rs 60,000.

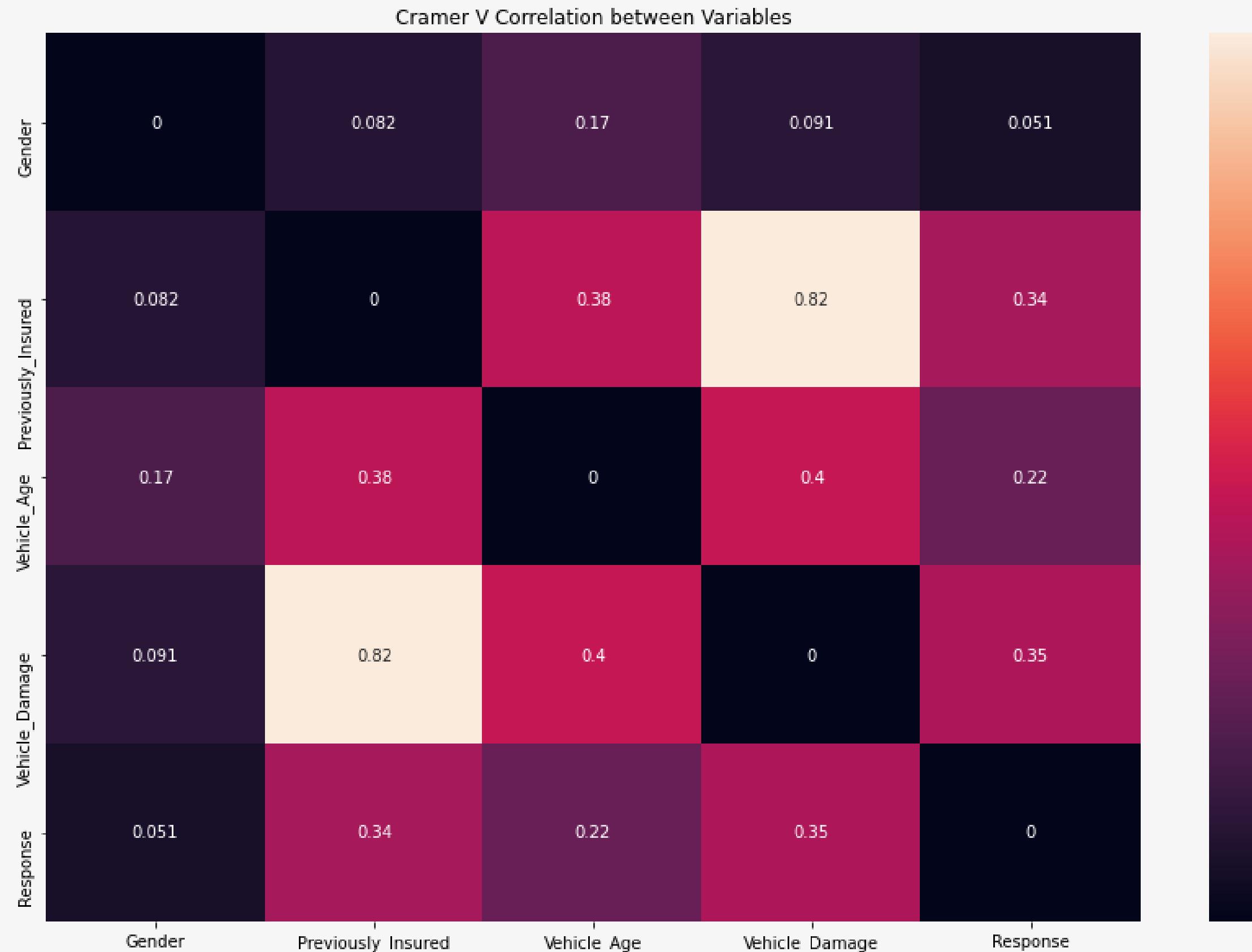
This shows the economic gap distribution between Indian citizens

Numerical Column Correlations



The heatmap doesn't show any strong correlation between features

Categorical Column Correlations With Response Using Cramers V Heatmap



The **Previously_Insured** feature seems to have a strong relation with **Vehicle_Damage**

This is supported by the fact that people who have had their vehicle damaged will buy a vehicle insurance

Categorical Column Correlations With Response Using Cramers V Heatmap

Kruskal-Wallis Test results of Gender

H Stat = 809.6826408737934

P = 4.2354192395808124e-178

Kruskal-Wallis Test results of Previously_Insured

H Stat = 35440.63114163706

P = 0.0

Kruskal-Wallis Test results of Vehicle_Age

H Stat = 14955.41399255504

P = 0.0

Kruskal-Wallis Test results of Vehicle_Damage

H Stat = 38204.02112771305

P = 0.0

Using the statistical Kruskal-Wallis Test, we can safely assume that the categorical columns could help distinguish the Response, as all the P-Values are below 0.05

Data Preprocessing



1

Casting Data Types

2

Remove Unwanted Feature

3

Remove Rows with no Driving License, then remove the column

4

Encode Categorical Columns

5

Remove Outlier on Annual_Premium column

6

Normalized Numerical Columns

Casting Data Types

Casting any categorical columns types to category

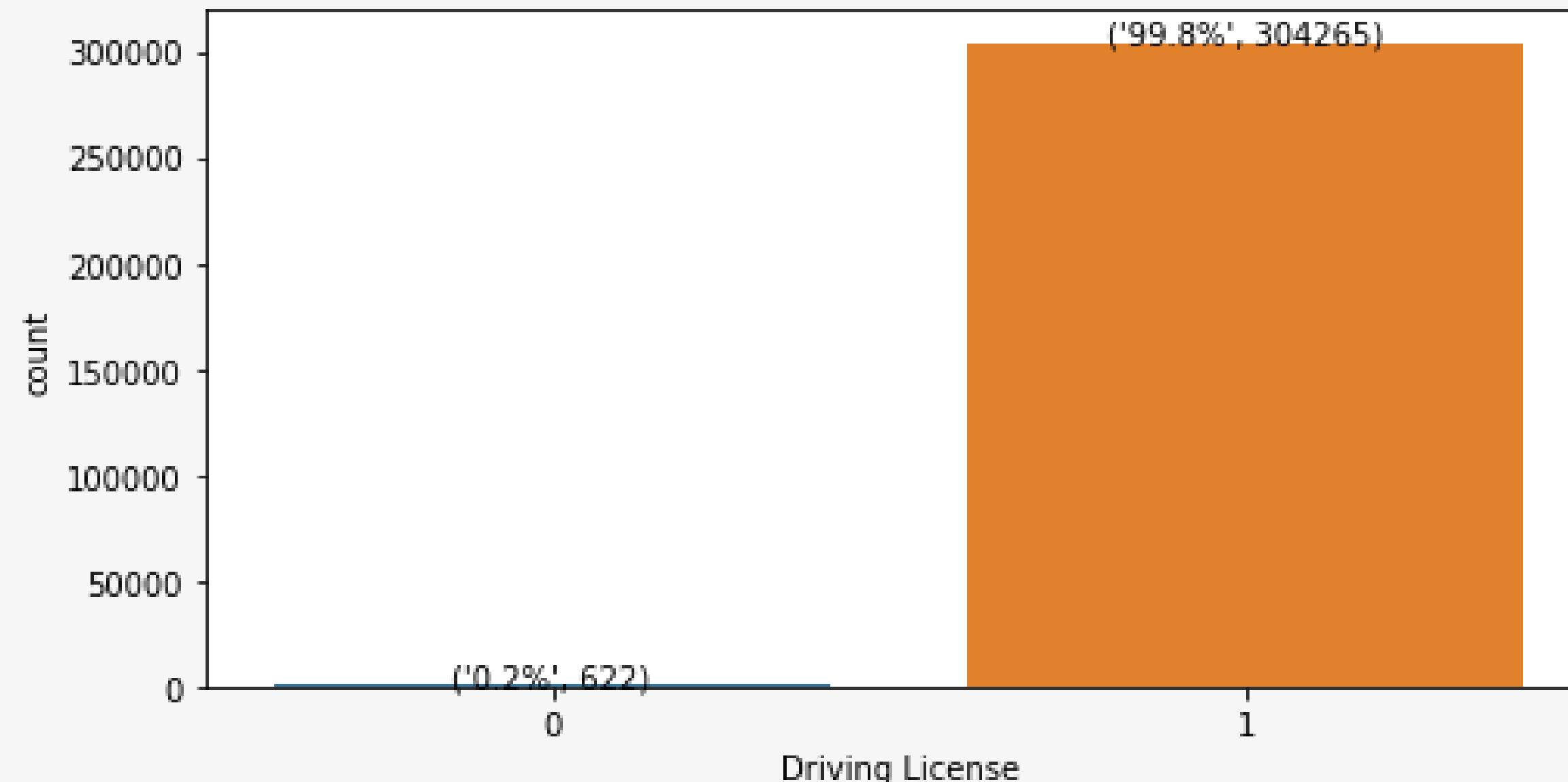
Remove Unwanted Feature

The feature Policy_Sales_Channel and Region_Code are not relevant to real world applications.

Policy_Sales_Channel are anonymized saling code of the insurance company. Hence, we can't use it as we don't know what was the meaning of the values behind the feature. On the other hand, Region_Code represents 53++ distinct values, whereas the Government India stated only 35 state codes in India

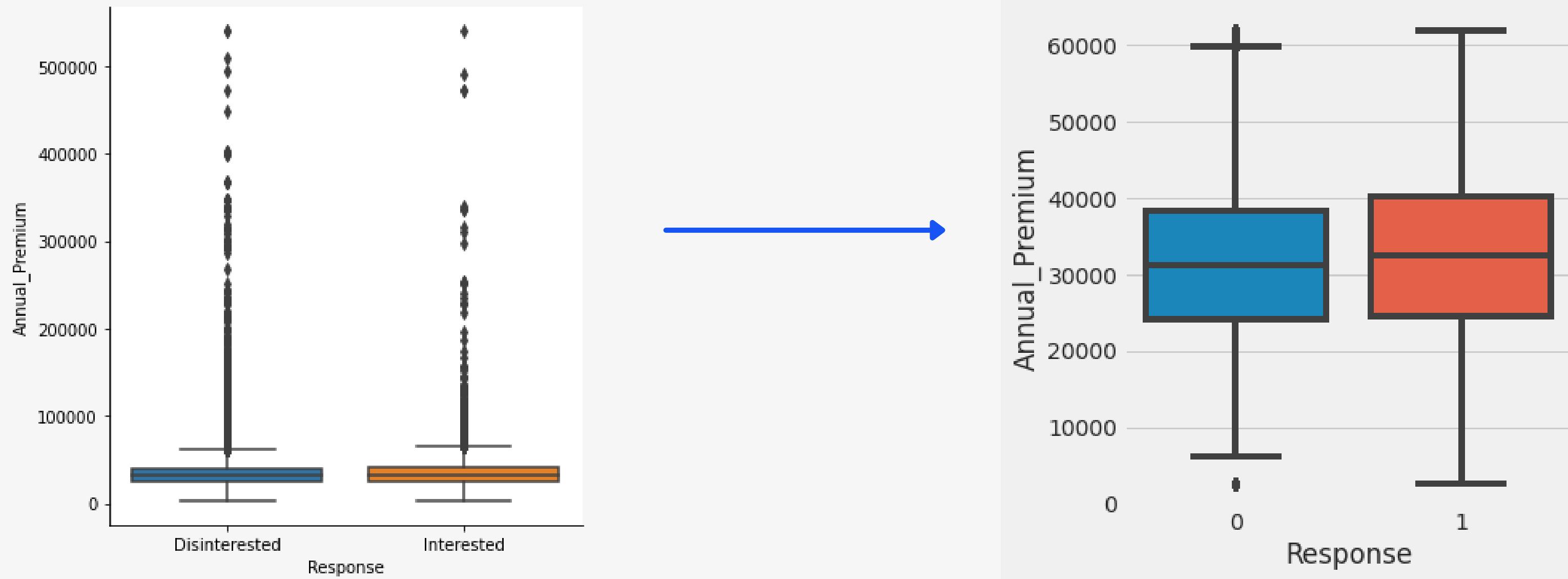
Encode Categorical Columns

Remove Rows with no Driving License, then Remove the Column



The countplot shows that the number of people with no Driving License is really small, with a 6.2%. Hence, we decide to delete the row of customer without any Driving license, assuming that all of the customers who buy insurance vehicle must have a driving license.

Remove Outlier on Annual_Premium column



We will remove the outliers using the Inter Quartile Range(IQR) method.

Normalized Numerical Columns

	<code>id</code>	<code>Age</code>	<code>Previously_Insured</code>	<code>Vehicle_Age</code>	<code>Vehicle_Damage</code>	<code>Annual_Premium</code>	<code>Vintage</code>	<code>Response</code>	0	1
0	138389	0.415385		0	0.5	1.0	0.526802	0.560554	0	0.0 1.0
1	149691	0.384615		0	0.5	1.0	0.000000	0.024221	0	0.0 1.0
2	213565	0.046154		1	0.0	0.0	0.541090	0.401384	0	0.0 1.0
3	278877	0.046154		1	0.0	0.0	0.460329	0.197232	0	0.0 1.0
4	181394	0.292308		0	0.5	1.0	0.557421	0.020761	0	1.0 0.0

Normalizing numerical columns using
MinMaxScaler, which has proven to improve
our model by 23%

Modeling and Evaluation



Making Baseline Value

Making baseline value is important, so that we can use it as an measurement to improve and get our best model. In this case, we used LogisticRegression as our baseline model

Baseline LogisticRegression Model without Normalization ROC-AUC Score:
0.6712248106072092

Baseline LogisticRegression After Normalization Model ROC-AUC Score:
0.8305344294005266

Split dataset and test over 10 models

	Model	Accuracy	F1	Precision	Recall	ROC
0	XGBoost	0.878081	0.821079	0.892945	0.878081	0.846362
1	Gradient Boosting	0.878081	0.821079	0.892945	0.878081	0.846156
2	LGBM	0.878115	0.821455	0.838209	0.878115	0.846132
3	CatBoost	0.877405	0.822915	0.818824	0.877405	0.842939
4	Logistic Regression	0.878013	0.821045	0.771019	0.878013	0.829902
5	LDA	0.875800	0.825152	0.816336	0.875800	0.828957
6	Gaussian	0.639716	0.700895	0.903455	0.639716	0.821410
7	Random Forest	0.854312	0.832645	0.818916	0.854312	0.810628
8	KNN	0.844852	0.831341	0.820965	0.844852	0.693938
9	Decision Tree	0.822046	0.822967	0.823902	0.822046	0.595975
10	SGD	0.878081	0.821079	0.892945	0.878081	0.000000

From the testing, we got XGBoostClassifier as our best model. Hence, we will use it for further evaluation

Parameter tuning using GridSearchCV

What we do next is parameter tuning. In this case, we will use a library called GridSearchCV from SKLearn. Parameter tuning is used to improve our model and get better score than non-tuned model

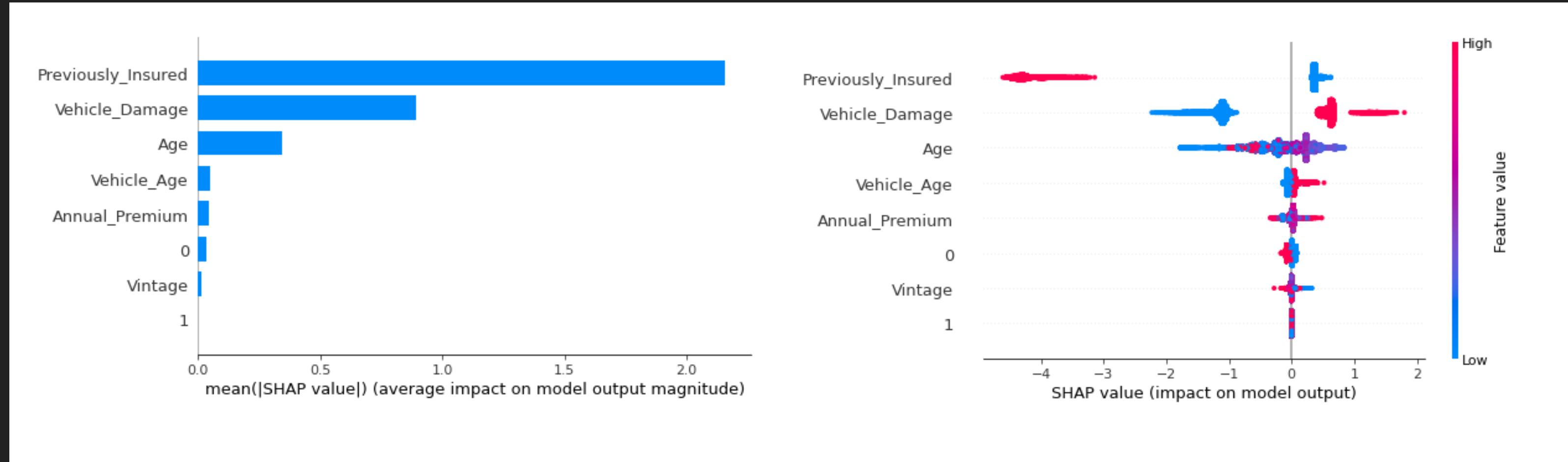


Evaluate the model

```
Accuracy score : 0.8780809189965368
F1 score : 0.8210786793124588
Precision score : 0.8929451813092658
Recall score : 0.8780809189965368
ROC score : 0.8467748807167663
precision      recall   f1-score   support
0             1.00     0.88      0.94    59195
1             0.00     0.00      0.00       0
accuracy                           0.88    59195
macro avg       0.50     0.44      0.47    59195
weighted avg    1.00     0.88      0.94    59195
```

We then fit our model and test it with our test dataset. The precision and recall is still 0, it is because we still used the default threshold value, 0.5

Feature Importance



Both SHAP value charts explained that `Previously_Insured` and `Vehicle_Age` was the most important feature, based on their feature importance

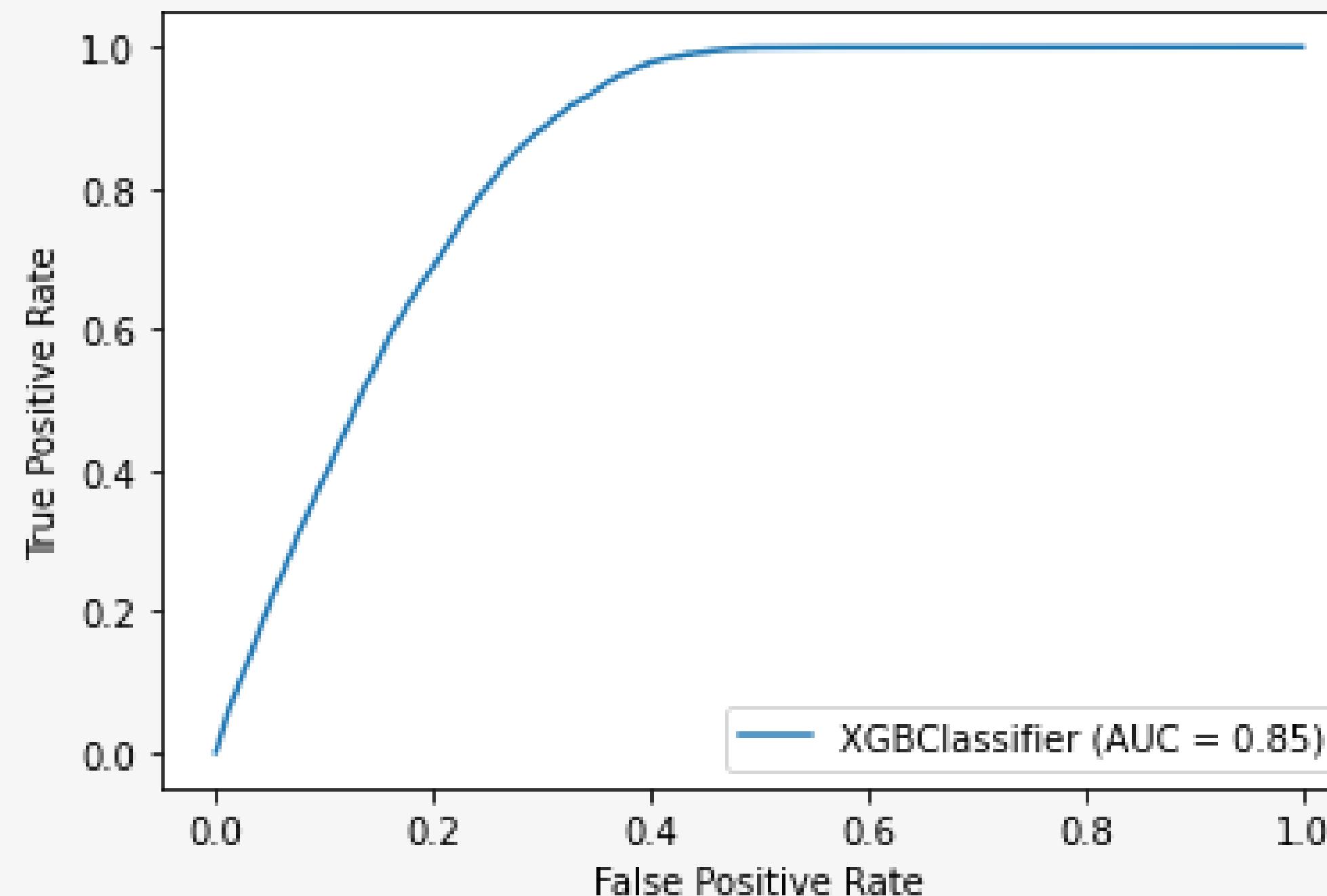
Prediction on Unseen Dataset

We then use the data than we have splitted for final prediction. The model has never seen this data yet. Hence, it is perfect for evaluation

```
Accuracy score : 0.8780830429262231
F1 score : 0.8210817228542153
Precision score : 0.8929467873483522
Recall score : 0.8780830429262231
ROC score : 0.8482817855576683
```

We still get a good score, for our model to predict unseen data

Prediction on Unseen Dataset



Using ROC_AUC as new threshold

We will use our new threshold 0.85, to predict the new dataset. Then, we get the classification report as below

	precision	recall	f1-score	support
0	0.98	0.70	0.81	65008
1	0.29	0.89	0.44	9026
accuracy			0.72	74034
macro avg	0.63	0.79	0.62	74034
weighted avg	0.89	0.72	0.77	74034

We can see that the recall has increased into 0.89, by changing the threshold

Confusion Matrix

		True Class	
		Positive	Negative
Predicted Class	Positive	TP 45300	FP 19708
	Negative	FN 1015	TN 8011

Business Case Study



Business Case Study

By Using Machine Learning, the insurance company can save up to 62.56% in saling expenses(Reaching out customer, operation expenses, etc.

The insurance company can also save up time from contacting every single policyholders, assuming that they use the model prediction output to reach out customers

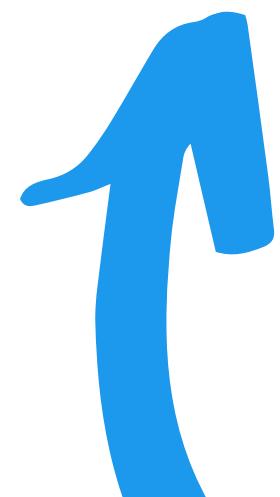
Conclusion and Reccomendation



Conclusion

Supervised Machine Learning has been proven to be able to predict the customer's response with 0.848 recall score

It also helps the company to reduce expenses up to 62% from reaching out customers and sales



Reccomendation

Get a deeper domain knowledge regarding previous column that got dropped, which can be used to improve the model

Add more rows and features to the model, so to reduce underfitting and improve model variation

Focus to look reach out policyholders who have ever had their vehicle damaged



Contact Us



EMAIL ADDRESS

darnellkikoo@gmail.com

PHONE NUMBER

+6281218222211

CURRENT PROJECT GITHUB LINK:

https://github.com/darnellkikoo/AlgoBC_FProject



THANK

YOU