Problem Statement

Your client is an Insurance company that has provided Health Insurance to its customers. Now they need your 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.

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 a company bear such a high hospitalisation cost when it charges a premium of only Rs. 5000/-, that is where the concept of probabilities comes into 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.

Just like medical insurance, there is vehicle insurance where every year a customer needs to pay a premium of a certain amount to the insurance provider company so that in case of an 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 optimise its business model and revenue.

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

Train.csv

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	Anonymised 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

Test.csv

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	Anonymised 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

Evaluation Metric : The evaluation metric for this hackathon is ROC_AUC score.

Data Findings for any underlying discrepancies:

- No data type discrepancy found in train and test set
- No missing values in train and test set
- Data Quality Check :
    - Proportion of Gender variable is almost similar in training and test set, i.e 54% to 46% is the ratio of Male is to Female
    - Proportion of Having Driving Licence is almost similar to 99.9%:.1% of having licence to not having licence
    - Proportion of previously not insured to insured is of 54% to 46% in both the set
    - Proportion of Vehicle Age is 53%:43%:4% of 1-2 Years : < 1 Years : >2 Years in both the set
    - Proportion of vehicle being damage to not damage : 51%:49% in both the set
    - Distribution of Age is same in train and test set
    - Distribution of Annual Premium is same in train and test set
    - Distribution of Vintage days is also approximately same in train and test
- Hence, we can conclude that there is no data shift in between the training and test set. Test set very well represents the training set over which model will be developed. Though there lies skewness in the continuous variables like age, annual premium,etc.
- We see that there lies outliers in the annual premium column. Hence, using inbuilt outlier_detection[1]
- There is a huge class imbalance for Not Interested to Interested ratio is 87% : 13%

Insights From Data

- We see that out of all 53% males only 7% of them have shown their interest in vehicle insurance. Whereas, of all 47% females only 4% have shown their interest

```
pd.crosstab(train['Gender'],train['Response'],normalize=True).sort_values(by=[1],ascending=False)
```

Response	0	1
Gender		
Male	0.465914	0.074847
Female	0.411523	0.047716

- We see that out of all 99.9% driving license holders only 12% have shown their interest. Whereas, of all .1% non holders , .01% have shown their interest

```
pd.crosstab(train['Driving_License'],train['Response'],normalize=True).sort_values(by=[1],ascending=False)
```

Response	0	1
Driving_License		
1	0.875414	0.122456
0	0.002023	0.000108

- We see that of all 54% who have availed vehicle insurance previously only 12% have shown their interest. Whereas of all 46% who have not got vehicle insurance previously, only .04% have shown their interest .

```
pd.crosstab(train['Previously_Insured'],train['Response'],normalize=True).sort_values(by=[1],ascending=False)
```

Response	0	1
Previously_Insured		
0	0.419641	0.122149
1	0.457796	0.000415

- Of all 53% of vehicles whose age is between 1-2 year, only 9% have shown interest in vehicle insurance, whereas of all 43% of vehicles whose age is less than 1 year, only 1% vehicles have shown their interest. Whereas, of all 4% vehicles whose age is greater than 2 years, only 1% has shown interest.

```
pd.crosstab(train['Vehicle_Age'],train['Response'],normalize=True).sort_values(by=[1],ascending=False)
```

Response	0	1
Vehicle_Age		
1-2 Year	0.434285	0.091328
< 1 Year	0.413488	0.018897
> 2 Years	0.029663	0.012338

- Of all 51% damaged vehicles, only 11% have got interest in insurance. Whereas, of all 49% damaged vehicles, only .2% have got interest in insurance.

```
pd.crosstab(train['Vehicle_Damage'],train['Response'],normalize=True).sort_values(by=[1],ascending=False)
```

Response	0	1
Vehicle_Damage		
Yes	0.384890	0.119987
No	0.492547	0.002577

- On an average, for both the response class and for both genders, it has been 154 days of association with the company.

```
pd.crosstab(train['Gender'],train['Response'],values=train['Vintage'],aggfunc='mean').sort_values(by=[1],ascending=False)
```

Response	0	1
Gender		
Male	154.128551	154.307555
Female	154.665202	153.805884

- Those who have driving licence and have shown interest in vehicle insurance, are associated with the company for 154 days. There is no difference in the number of days of association for those who didn't show interest. Where those who do not have a driving licence and showed interest, they are 143 days. But those who did not show interest are for more days in association with the company as compared, i.e for 156 days

```
pd.crosstab(train['Driving_License'],train['Response'],values=train['Vintage'],aggfunc='mean').
```

Response	0	1
Driving_License		
1	154.375292	154.121258
0	156.522698	143.853659

- We see eventual difference in the number of days of association for those who were previously insured and in between their response type.

```
d.crosstab(train['Previously_Insured'],train['Response'],values=train['Vintage'],aggfunc='mean')
```

Response	0	1
Previously_Insured		
1	154.576684	156.253165
0	154.165942	154.104979

- For all the vehicles of different ages, who had shown interest are associated with the company for 154 days appx.

```
pd.crosstab(train['Vehicle_Age'],train['Response'],values=train['Vintage'],aggfunc='mean').
```

Response	0	1
Vehicle_Age		
> 2 Years	154.296418	155.286261
1-2 Year	154.129557	154.179021
< 1 Year	154.649552	153.023049

- We did not see any difference in average days of association for damaged vehicle categories and in between their response type. While for not damaged vehicles , those who showed interest were for 152 days of association whereas those who did not show their interest, were for 154 days.

```
pd.crosstab(train['Vehicle_Damage'],train['Response'],values=train['Vintage'],aggfunc='mean')
```

Response	0	1
Vehicle_Damage		
Yes	154.188179	154.138383
No	154.530328	152.895112

- For both males and females we see a difference in average annual premium respective to the response type.

```
pd.crosstab(train['Gender'],train['Response'],values=train['Annual_Premium'],aggfunc='mean').
```

Response	0	1
Gender		
Male	32610.22613	33694.979720
Female	32232.28201	33615.168628

- For people with license and without license, their average annual premium differed with respect to the response type

```
pd.crosstab(train['Driving_License'],train['Response'],values=train['Annual_Premium'],aggfunc='mean').
```

Response	0	1
Driving_License		
0	36581.781453	35227.463415
1	32423.380496	33662.534273

- Average annual premium differs for response type for those who were insured before and as well as those who were not.

```
pd.crosstab(train['Previously_Insured'],train['Response'],values=train['Annual_Premium'],aggfunc='mean')
```

Response	0	1
Previously_Insured		
0	32689.674443	33676.470828
1	32197.656889	29962.455696

- With respect to response type, the average annual premium differs for each vehicle age category

```
pd.crosstab(train['Vehicle_Age'],train['Response'],values=train['Annual_Premium'],aggfunc='mean')
```

Response	0	1
Vehicle_Age		
> 2 Years	36560.379213	39159.552425
1-2 Year	32952.652402	33295.238005
< 1 Year	31591.047032	31857.660164

- With respect to response type, the average annual premium differs for damaged and not damaged vehicles.

```
pd.crosstab(train['Vehicle_Damage'],train['Response'],values=train['Annual_Premium'],aggfunc='mean').
```

Response	0	1
Vehicle_Damage		
Yes	32867.682081	33778.990345
No	32093.270590	28304.956721

- With respect to response type, mean age for damaged and non damaged vehicles were almost similar

```
pd.crosstab(train['Vehicle_Damage'],train['Response'],values=train['Age'],aggfunc='mean').sort_values(by=[1],ascending=False)
```

Response	0	1
Vehicle_Damage		
Yes	42.725282	43.595893
No	34.625031	35.969450

- With respect to response type, average age slightly differs for each vehicle age category. Majorly for vehicle age of between 1 to 2 years.

```
pd.crosstab(train['Vehicle_Age'],train['Response'],values=train['Age'],aggfunc='mean').sort_values(by=[1],ascending=False)
```

Response Vehicle_Age	0	1
> 2 Years	56.046351	53.069332
1-2 Year	49.761410	45.572344
< 1 Year	24.730594	26.819217

- With respect to response type, average age was similar for previously insured and non previously insured.

```
pd.crosstab(train['Previously_Insured'],train['Response'],values=train['Age'],aggfunc='mean').sort_values(by=[1],ascending=False)
```

Response Previously_Insured	0	1
0	42.162760	43.461828
1	34.525781	35.696203

- With respect to response type , we can see a difference in age in people with license and without license.

```
pd.crosstab(train['Driving_License'],train['Response'],values=train['Age'],aggfunc='mean').sort_values(by=[1],ascending=False)
```

Response Driving_License	0	1
0	65.952010	59.073171
1	38.114043	43.421822

- With respect to response type , we can see differences in age in Males and Females.

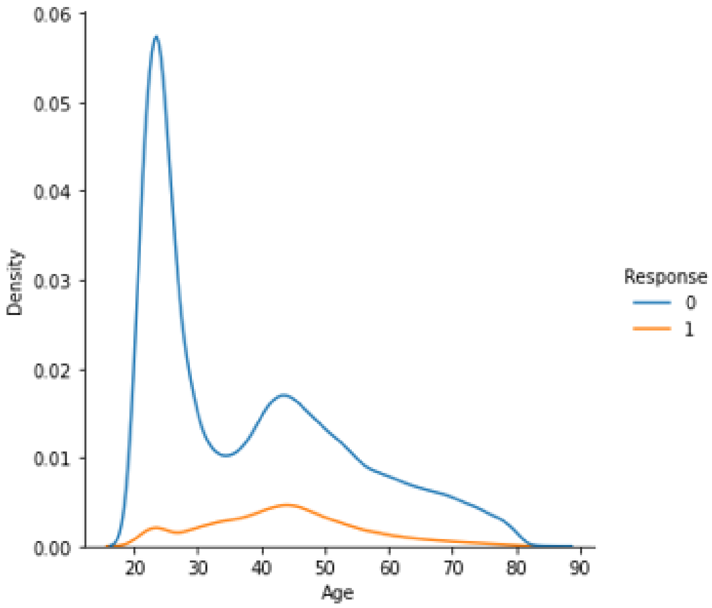
```
pd.crosstab(train['Gender'],train['Response'],values=train['Age'],aggfunc='mean').sort_values(by=[1],ascending=False)
```

Response Gender	0	1
Male	40.373375	44.200526
Female	35.692945	42.235634

- From the Age we can infer the following:
  - There is presence of data from at least two different clusters
  - Highest probability is for age group between 20 to 30
  - Distribution of age with respect to Response==1 is short and broad because :
    - Less data points , it comprises of only 10% approx of the whole data
    - High Variance
  - Age is right skewed

```
sns.displot(data=train,x='Age',kind='kde',hue='Response')
```

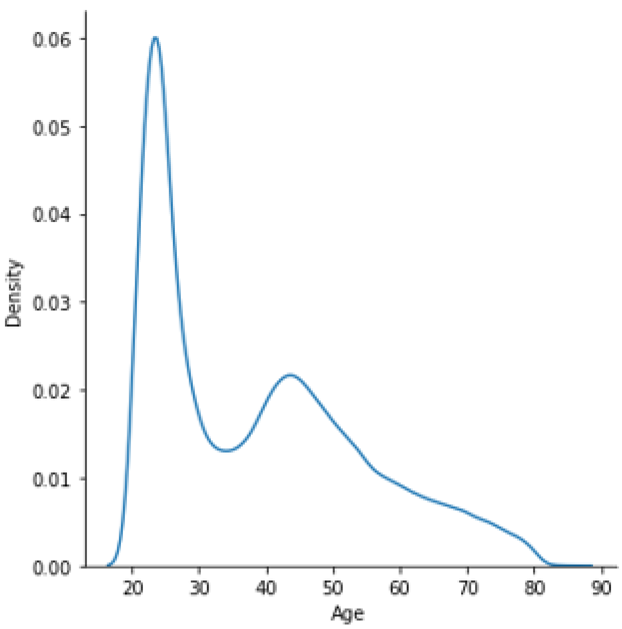
<seaborn.axisgrid.FacetGrid at 0x14013890490>



-

```
sns.displot(data=train,x='Age',kind='kde')
```

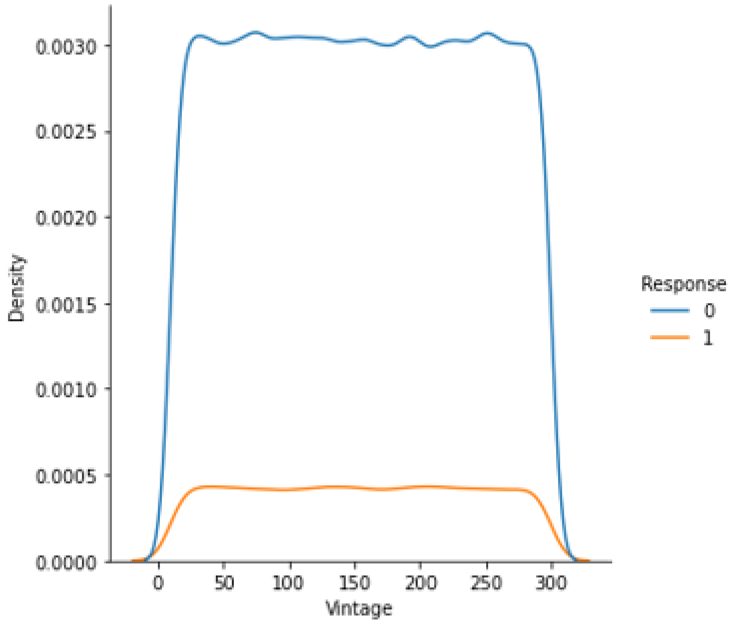
<seaborn.axisgrid.FacetGrid at 0x14014548040>



- The distribution of Vintage for both the type appears to be more of uniform type

```
sns.displot(data=train,x='Vintage',kind='kde',hue="Response")
```

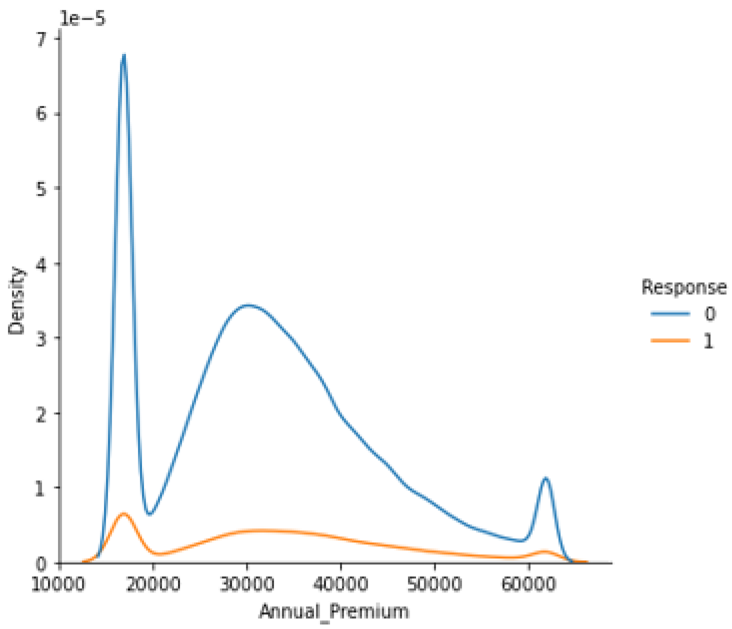
```
<seaborn.axisgrid.FacetGrid at 0x1401c60e190>
```



- Annual premium indicates presence of data from at least three different gaussians. There is low variance in the distribution, hence you can find the curve narrow and tall. It is positive skewed data. To address skewness we have used box-cox transformation.

```
sns.displot(data=train,x='Annual_Premium',kind='kde',hue="Response")
```

```
<seaborn.axisgrid.FacetGrid at 0x140177cea60>
```

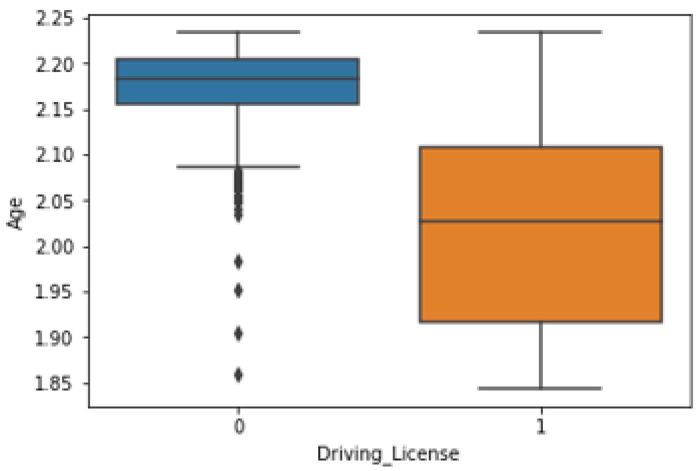


- All the transformation does not make the distribution perfectly normal, nor skewless, it just reduces the magnitude!
- It is also imperative to notice that there is no overlapping in distribution of variables depending on the type of response, the reason is imbalance in classes. The samples for interested type are less significantly from not interested type.

- While checking the distribution of Age column with respect to Driving License and Vehicle Age columns , we found there are outliers , hence we have treated the same.

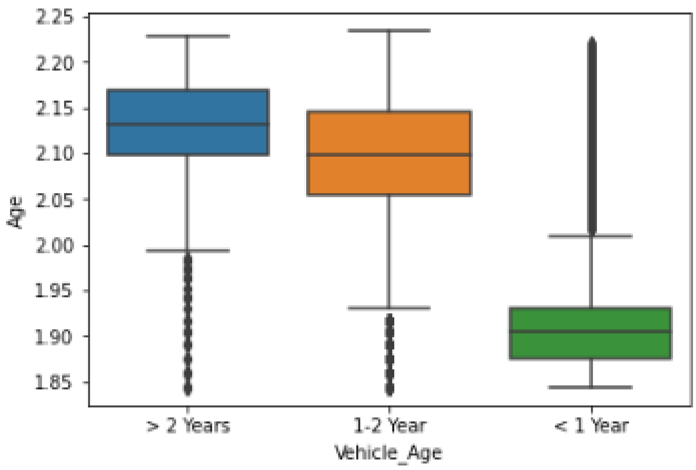
```
sns.boxplot(x=train['Driving_License'],y=train['Age'])
```

```
<AxesSubplot:xlabel='Driving_License', ylabel='Age'>
```



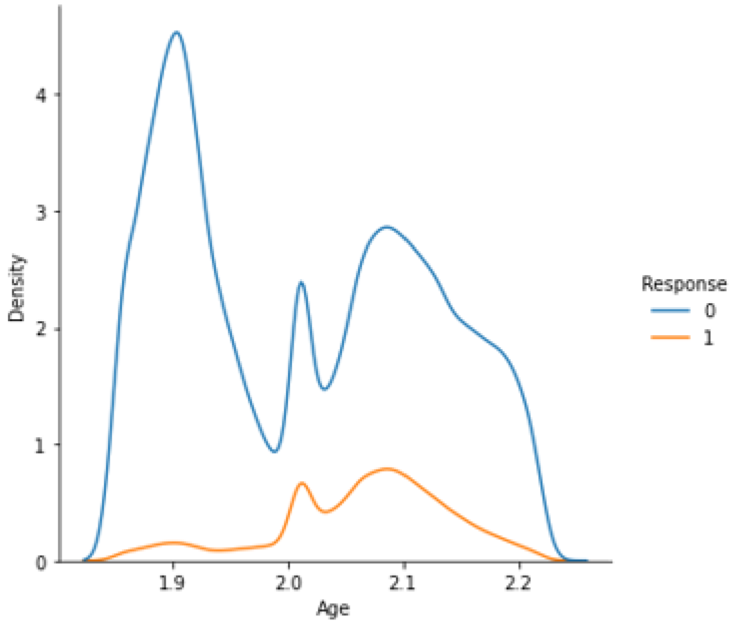

```
sns.boxplot(x=train['Vehicle_Age'],y=train['Age'])
```

```
<AxesSubplot:xlabel='Vehicle_Age', ylabel='Age'>
```



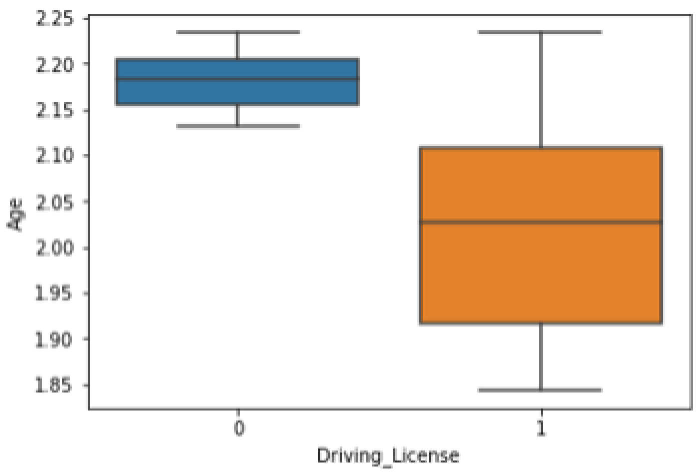
- Now after treating the Age column for outliers, the distribution shows presence of at least 3 gaussians.

```
sns.displot(data=train,x='Age',kind='kde',hue='Response')
<seaborn.axisgrid.FacetGrid at 0x29bc65f2130>
```



- Distribution of Age for people who do not have driving licence is between normalized value : 2.12 and 2.25, whereas for those having driving licence their age is between 1.85 and 2.25

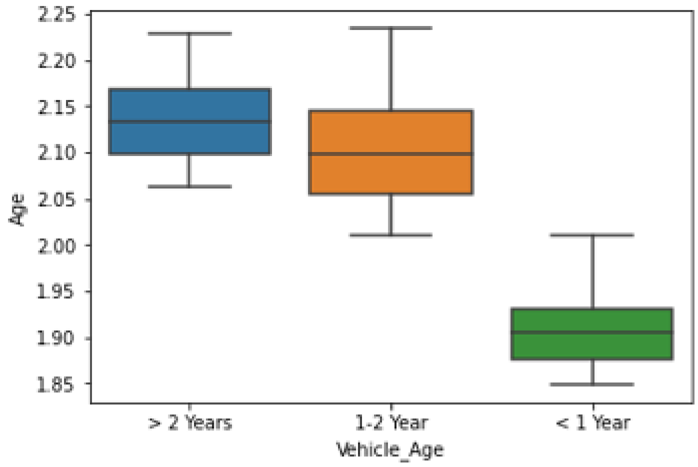
```
sns.boxplot(x=train['Driving_License'],y=train['Age'])
<AxesSubplot:xlabel='Driving_License', ylabel='Age'>
```



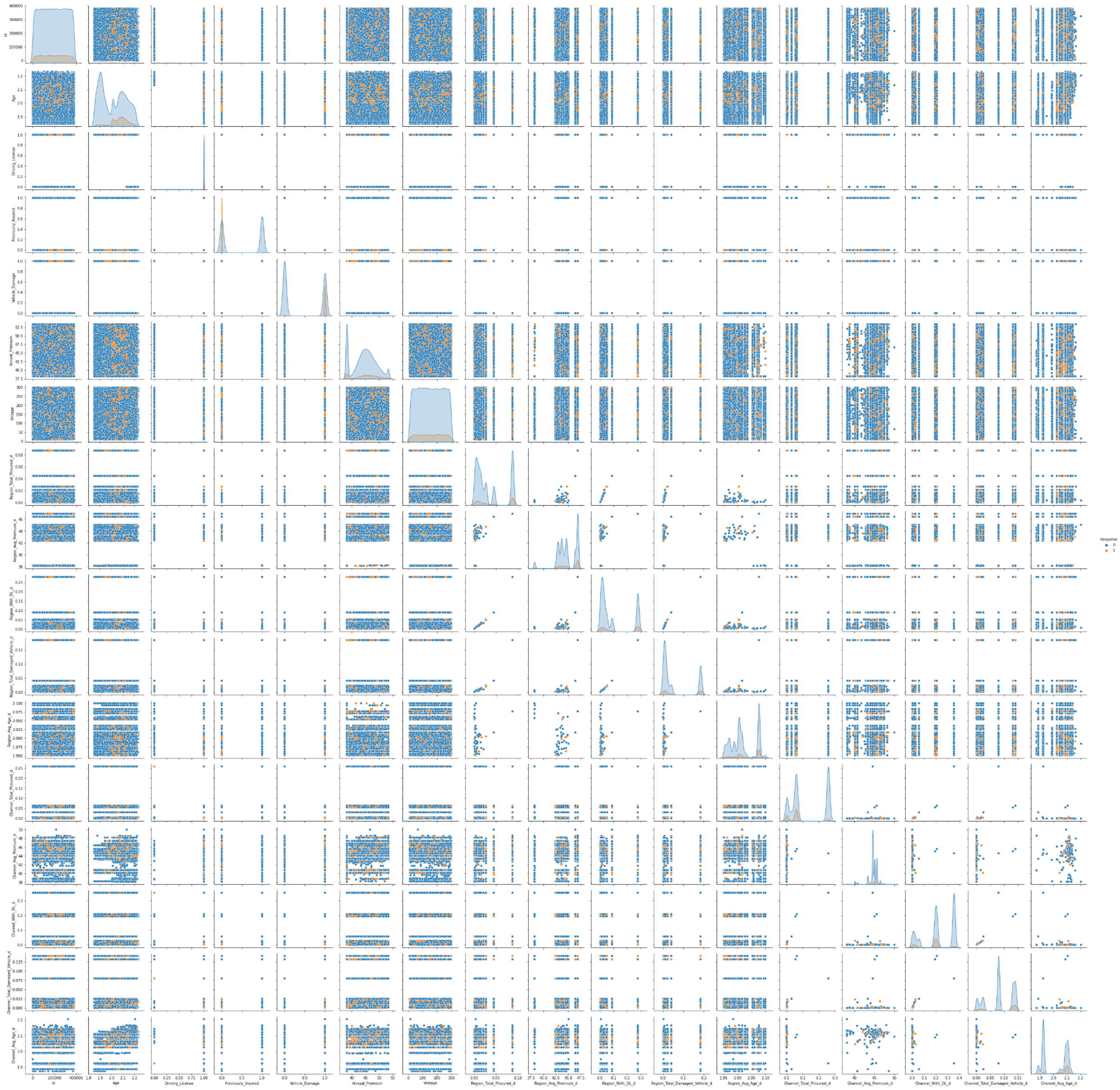
- Age distribution for vehicle owners whose vehicle ages less than a year are young comparatively, they lie in the range of 1.85 to 2

```
sns.boxplot(x=train['Vehicle_Age'],y=train['Age'])
```

```
<AxesSubplot:xlabel='Vehicle_Age', ylabel='Age'>
```



- From the below pairplot we can comprehend that algorithms like logistic regression can not separate data points with respect to the target variable. We have to check for algorithms like KNN, Decision Trees, and other ensemble models. SVM could have been a good model , but our data is large .



Model Building

Name	Train ROC	Val ROC	Overfitting	Underfitting	Avg CV ROC-Train	Avg-CV ROC-Test
DTC Baseline	.997	.54	1	-	-	-
DTC Tuned	.74	.71	-	-	.745	.714

XGB-Baseline	.887	.842	1	-	-	-
XGB-CV	.839	.832	-	-	.841	.8331

XGB-CV Parameters :

- Objective -> binary:logistic
- eta -> .05
- max_depth -> 4
- scale_pos_weight -> 1
- subsample -> .8
- colsample_bytree -> .35
- reg_lambda -> 1.2

```
Fold : 0
ROC-AUC | Fold : 0 | Train Fold Score  : 0.8219113925327202
ROC-AUC | Fold : 0 | Test Fold Score  : 0.8182211194775018
ROC-AUC | Val Score  : 0.8189599435262658
ROC-AUC | Train Score  : 0.8219113925327202
Fold : 1
ROC-AUC | Fold : 1 | Train Fold Score  : 0.8530343933406696
ROC-AUC | Fold : 1 | Test Fold Score  : 0.8402815536415489
ROC-AUC | Val Score  : 0.8403616476602085
ROC-AUC | Train Score  : 0.8530343933406696
Fold : 2
ROC-AUC | Fold : 2 | Train Fold Score  : 0.8294729589605998
ROC-AUC | Fold : 2 | Test Fold Score  : 0.8219550126830806
ROC-AUC | Val Score  : 0.8257003079398924
ROC-AUC | Train Score  : 0.8294729589605998
Fold : 3
ROC-AUC | Fold : 3 | Train Fold Score  : 0.8479370446354942
ROC-AUC | Fold : 3 | Test Fold Score  : 0.840706374143713
ROC-AUC | Val Score  : 0.8387981147662816
ROC-AUC | Train Score  : 0.8479370446354942
Fold : 4
ROC-AUC | Fold : 4 | Train Fold Score  : 0.8294096399705965
ROC-AUC | Fold : 4 | Test Fold Score  : 0.8275708944796354
ROC-AUC | Val Score  : 0.8257653759187698
ROC-AUC | Train Score  : 0.8294096399705965
Fold : 5
ROC-AUC | Fold : 5 | Train Fold Score  : 0.8298462559488985
ROC-AUC | Fold : 5 | Test Fold Score  : 0.8191239448680101
ROC-AUC | Val Score  : 0.8253875366308769
ROC-AUC | Train Score  : 0.8298462559488985
Fold : 6
ROC-AUC | Fold : 6 | Train Fold Score  : 0.8456940743701806
ROC-AUC | Fold : 6 | Test Fold Score  : 0.8449192797156697
ROC-AUC | Val Score  : 0.8380572752978939
ROC-AUC | Train Score  : 0.8456940743701806
Fold : 7
ROC-AUC | Fold : 7 | Train Fold Score  : 0.8478579036106351
ROC-AUC | Fold : 7 | Test Fold Score  : 0.8391226908911118
ROC-AUC | Val Score  : 0.8382281224006046
ROC-AUC | Train Score  : 0.8478579036106351
Fold : 8
ROC-AUC | Fold : 8 | Train Fold Score  : 0.8560871695354477
ROC-AUC | Fold : 8 | Test Fold Score  : 0.8416288532918683
ROC-AUC | Val Score  : 0.8403288349932304
ROC-AUC | Train Score  : 0.8560871695354477
Fold : 9
ROC-AUC | Fold : 9 | Train Fold Score  : 0.8450086842943915
ROC-AUC | Fold : 9 | Test Fold Score  : 0.8380444678122043
ROC-AUC | Val Score  : 0.8372768993595223
ROC-AUC | Train Score  : 0.8450086842943915
```