Capstone Project - 3

HEALTH INSURANCE CROSS SELL PREDICTION

A project to predict how effective vehicle insurance will be for the customers of health insurance policy of the company.

Saurav Kumar

Index

➢ *Defining problem statement*

➢ *Data cleaning and feature engineering*

➢ *Feature Selection*

➢ *Prepare Dataset for modeling*

➢ *Apply Model*

➢ *Validation and Hyperparameter tuning*

➢ *Challenges*

➢ *Conclusion*

Problem Statement

First of all before going through any code or analysis we must know what is the reason for doing this analysis.

Data provided to us have many attributes which refer to a particular customer, our task is to remove noise from that data, and then find relation among the different attributes, to visualize the behavior of attribute or relation of two or more attributes using eda. To gain understanding from the data, we will use Python to undertake exploratory data analysis.

Our task involves to build 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

Data pipeline

- **Data Processing :** In this first we removed or replaced or the null and duplicate values as required, and removed the outliers
- **Feature Engineering :** In this we go through each feature, add new feature, changed the column which have object data type, do encoding on categorical variable, dropped the columns which are not necessary.
- **EDA:** In this part we do some exploratory data analysis on the feature which we occupied after feature engineering to explore some useful trends in our data.
- **Create a model :** In this First we created a baseline model, then slowly increased the model complexity for better performance, and then we performed hyperparameter tuning of models.

Data Summary

- The given dataset is of 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
- Important Features in HEALTH INSURANCE CROSS SELL PREDICTION dataset
  - 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

Data Summary

- Some important Features in HEALTH INSURANCE CROSS SELL PREDICTION dataset
  - 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

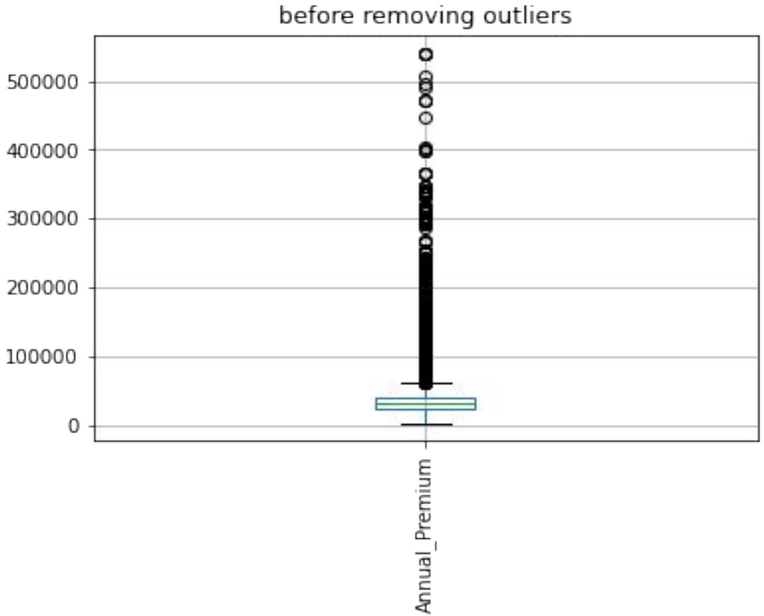
Data Cleaning and Feature engineering

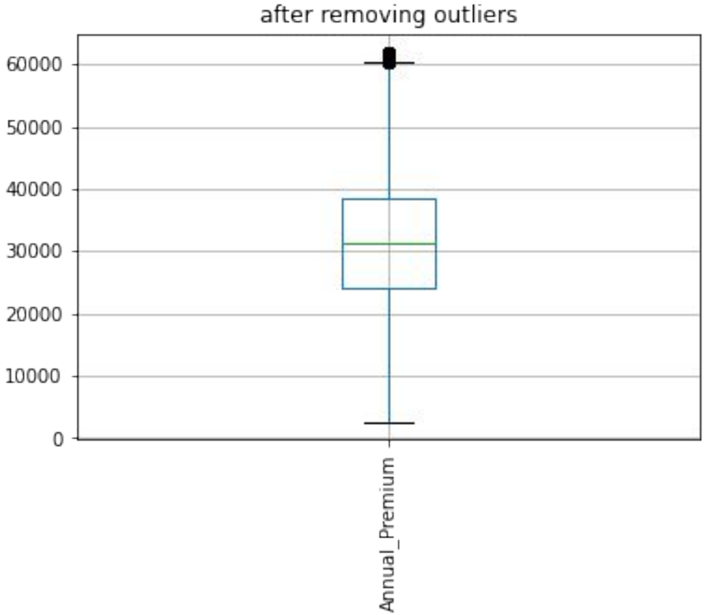
- Given dataset has neither have any null value nor any duplicate. We drop id column.
- **Encoding on categorical variable**
  - We do label encoding on gender and vehicle damage columns
  - For vehicle_age column, we have cars less than 1 year, between 1-2 year and greater than 2 years. So we encode it to 0, 1 and 2 respectively
- **Feature engineering on categorical variable**
  - **Policy Sales Channel :** From value counts we observe that some of the channels have very less contribution. so let's consider these all as 1 channel with index 0.0. Now we have added a new column named Channel Response which shows the percentage of customers for a given policy channel who takes vehicle insurance, as there are so many policy channels and we will drop policy channel column.
  - **Region_code :** There is no region code have very less value, so we cannot include others here. Here we also add new column Region Response as same done for Policy Sales Channel and drop Region_code column..

Data Cleaning and Feature engineering

- **Feature engineering on continuous variable**
    - **Age_section:** In age section we will reduce the cardinality by using binning method. We will divide age in three categories 20 to 40, 40 to 60 and above than 60, then we encode these categories to 0, 1 and 2 respectively.

Removing Outliers and boxplot

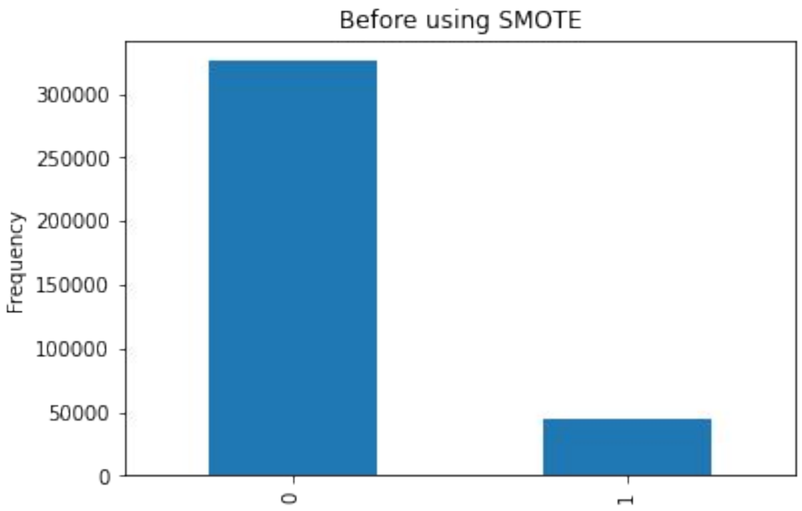


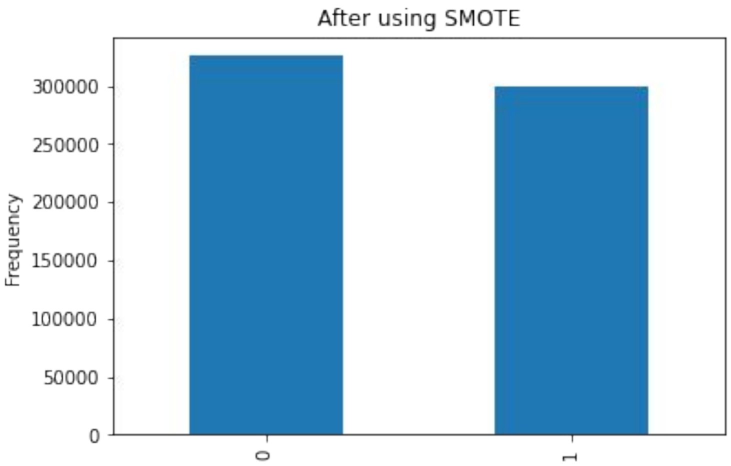


- We have outliers presents in Annual_Premium. So we remove it by using quantile method.

Dependent variable distribution

- Our target variable is Response column where we have very few data point who accept the insurance policy. we have only 12% entries for positive response, so our model will not be able to get trends.
- We will apply SMOTE to overcome the imbalance of dataset.

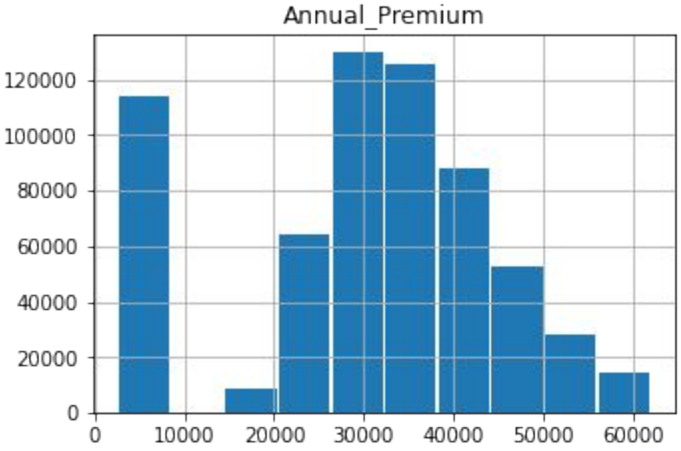


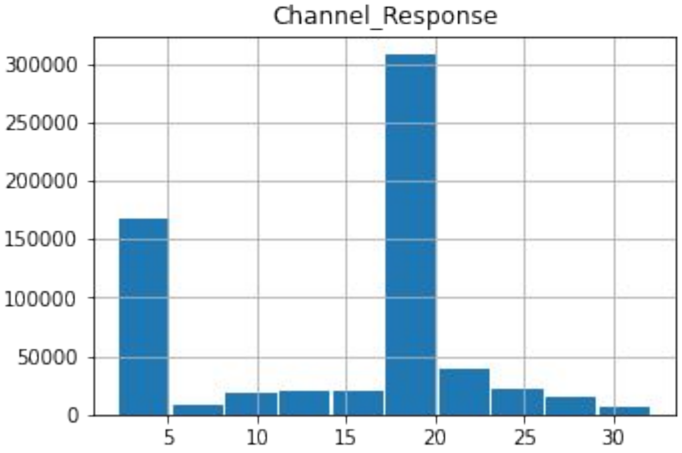


EDA

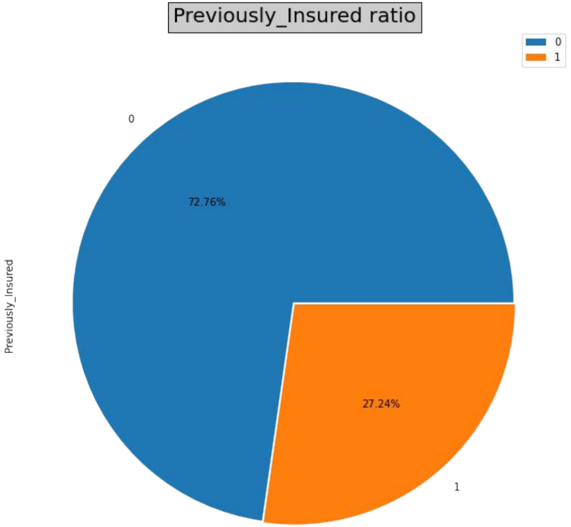
Independent_variable

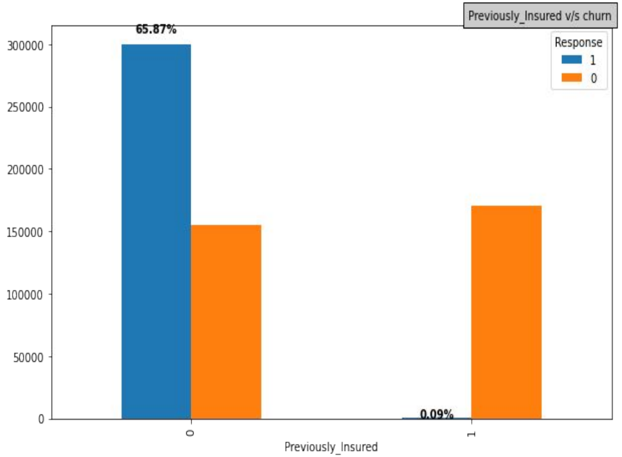
- From histogram we note that there are very less number of cars on which companies have charge very high annual premium
- In channel response either the percentage of response is mainly between 0-5 or between 15-20

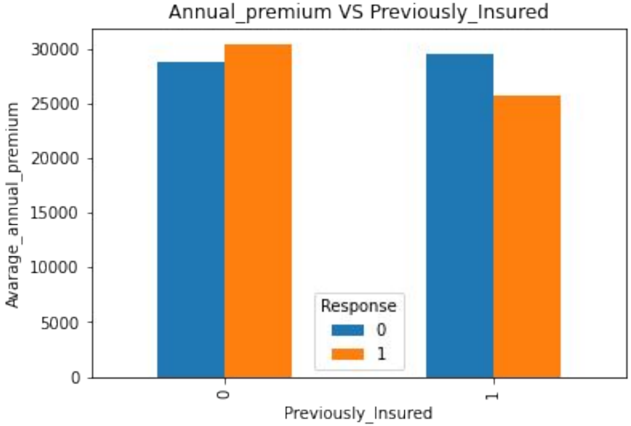




Insured previously

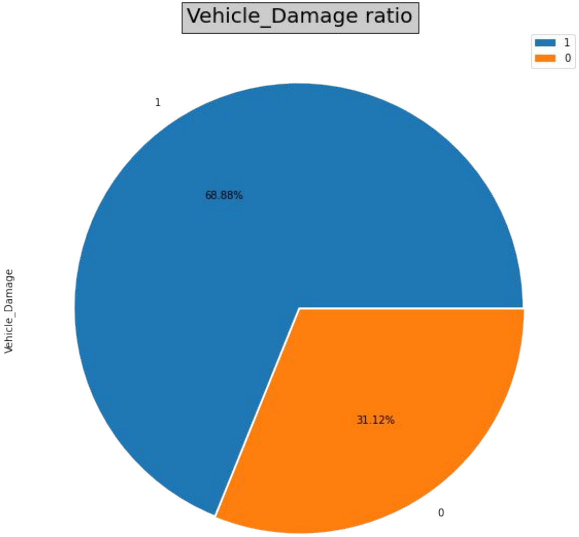


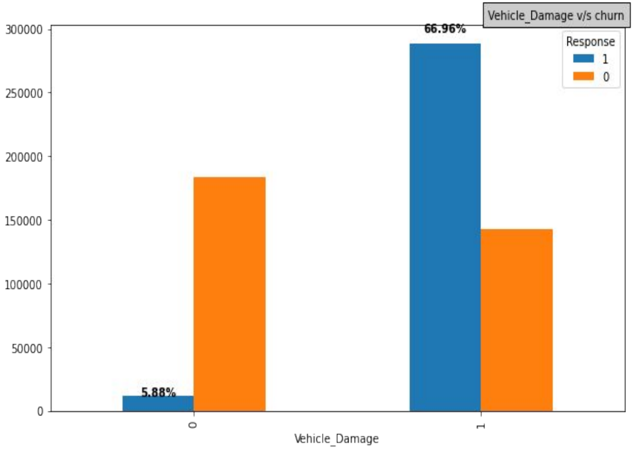


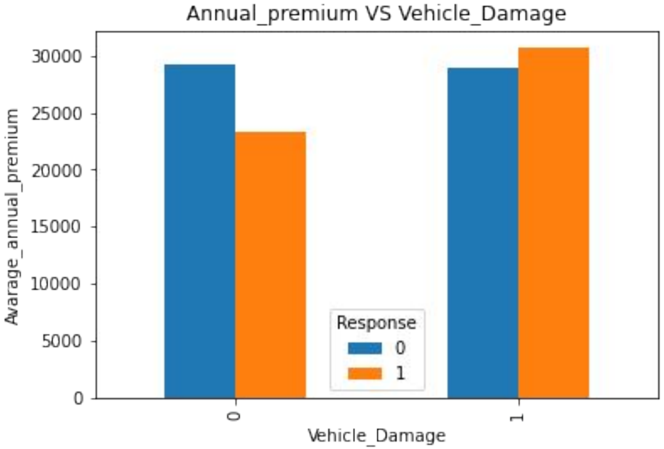


- There are only 26.17% of people who have insured previously
- Only 0.09% people take insurance while they have already insured there car.
- Average price is costly as people take insurance for first time. Average price is low for person who has already take a insurance for there car

Vehicle_damage Categorical Variable

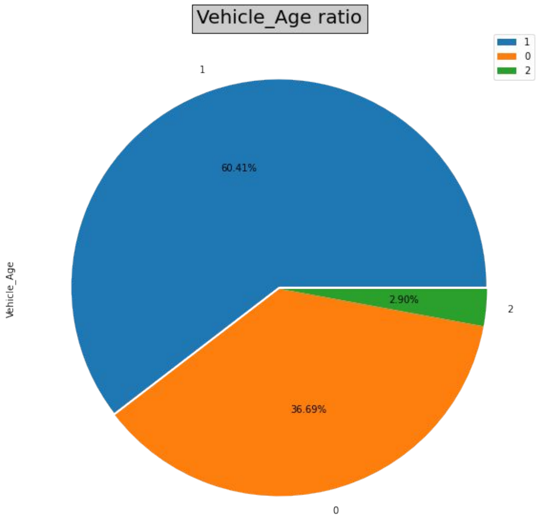


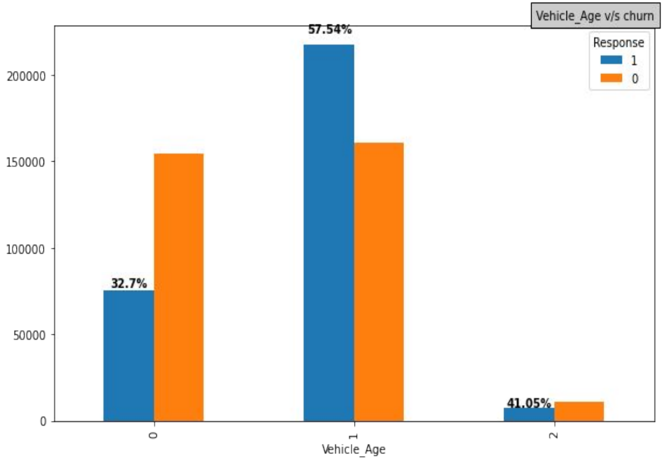


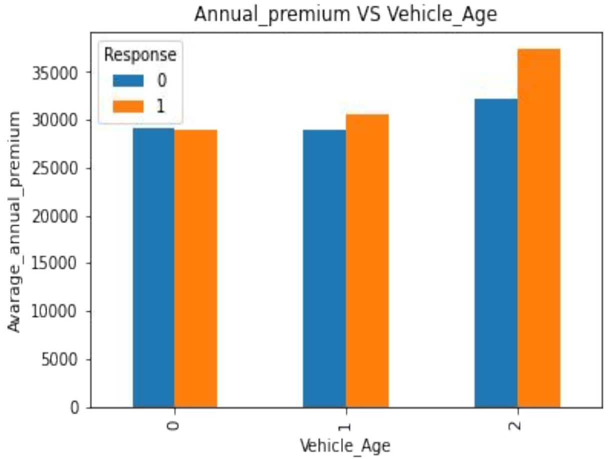


- We have 68% of data in which vehicle are damage. People generally takes insurance if there vehicle gets damage. 66% of people insured there car if they have damage vehicle.
- Approx 6% of people insured there car if there vehicle is not damage.
- People take insurance in lower premium rate if there car does not get damage, but for damage car people takes insurance at higher premium rate.

Vehicle_age Variable



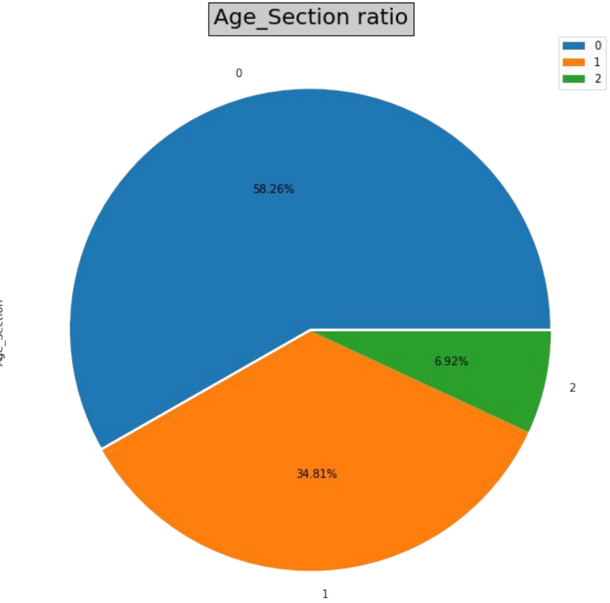


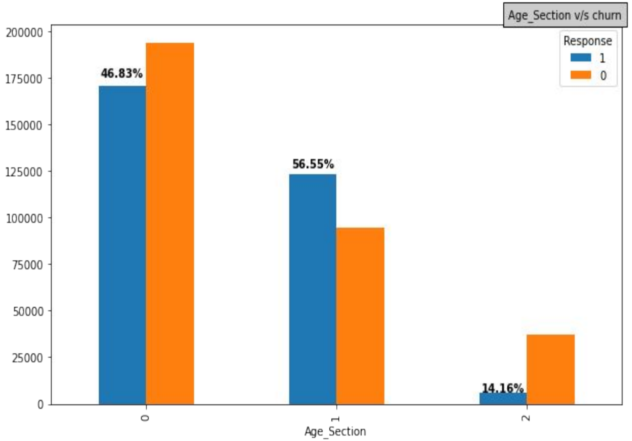


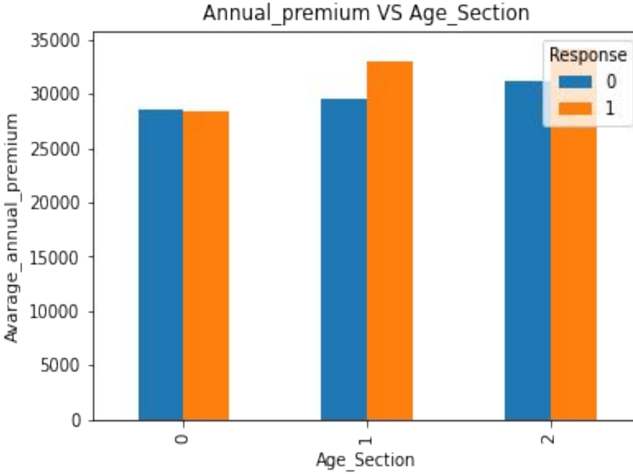
- About 57% people whose car age is between 1-2 years take the insurance. This percentage decreases for new car

- People takes costly insurance if there car is older than 2 years. For 1 year and less than 1 year insurance premium is approx same

Age_section Variable

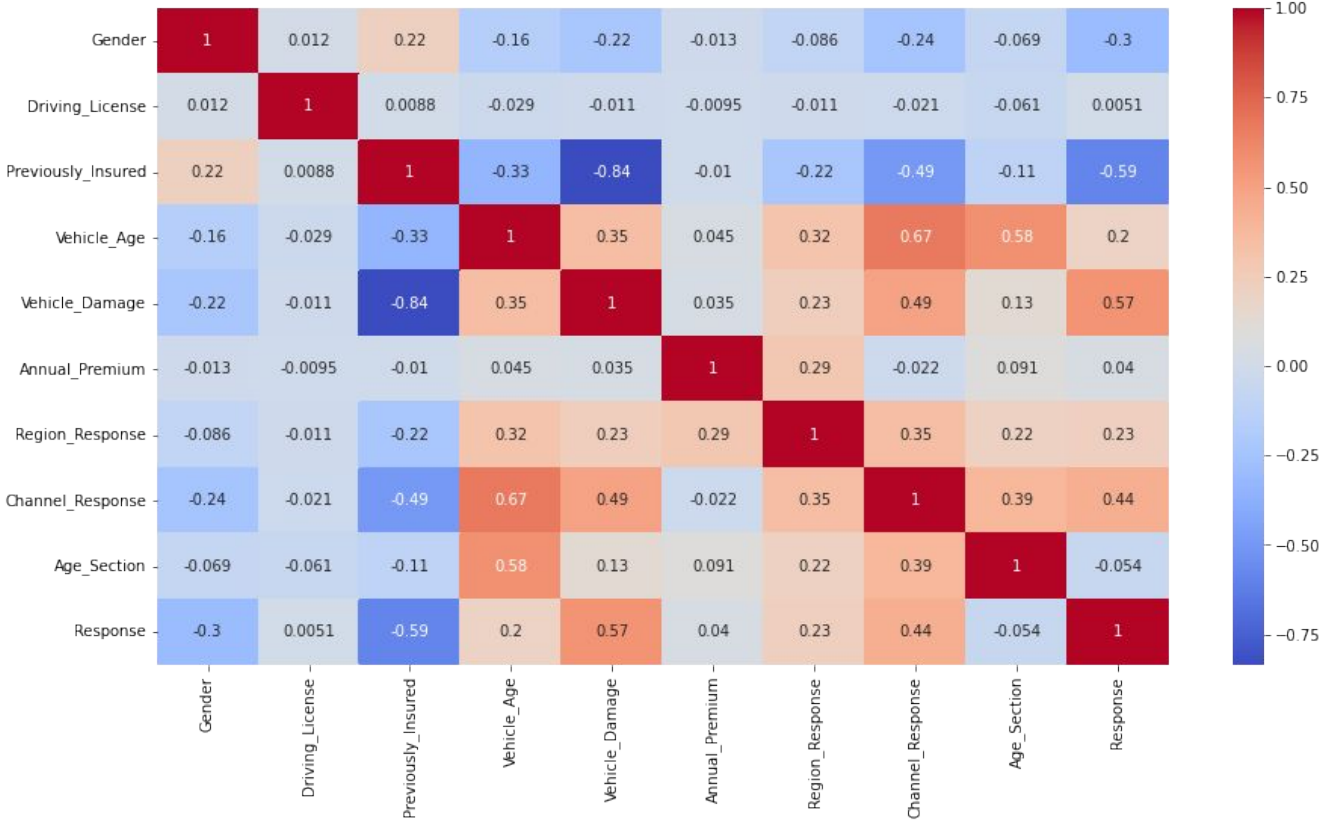






- About 58% of 40-60 age group people takes insurance with average annual premium is greater than 30,000.
- About 15% of 60 above age group people takes insurance with average annual premium is about 35,000.

Heat Map



	Gender	Driving_License	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Region_Response	Channel_Response	Age_Section	Response
Gender	1	0.012	0.22	-0.16	-0.22	-0.013	-0.086	-0.24	-0.069	-0.3
Driving_License	0.012	1	0.0088	-0.029	-0.011	-0.0095	-0.011	-0.021	-0.061	0.0051
Previously_Insured	0.22	0.0088	1	-0.33	-0.84	-0.01	-0.22	-0.49	-0.11	-0.59
Vehicle_Age	-0.16	-0.029	-0.33	1	0.35	0.045	0.32	0.67	0.58	0.2
Vehicle_Damage	-0.22	-0.011	-0.84	0.35	1	0.035	0.23	0.49	0.13	0.57
Annual_Premium	-0.013	-0.0095	-0.01	0.045	0.035	1	0.29	-0.022	0.091	0.04
Region_Response	-0.086	-0.011	-0.22	0.32	0.23	0.29	1	0.35	0.22	0.23
Channel_Response	-0.24	-0.021	-0.49	0.67	0.49	-0.022	0.35	1	0.39	0.44
Age_Section	-0.069	-0.061	-0.11	0.58	0.13	0.091	0.22	0.39	1	-0.054
Response	-0.3	0.0051	-0.59	0.2	0.57	0.04	0.23	0.44	-0.054	1

- From here we note that there is a good correlation between previously insurance and vehicle damage with target variable. so we conclude that people take insurance if there vehicle get damage.

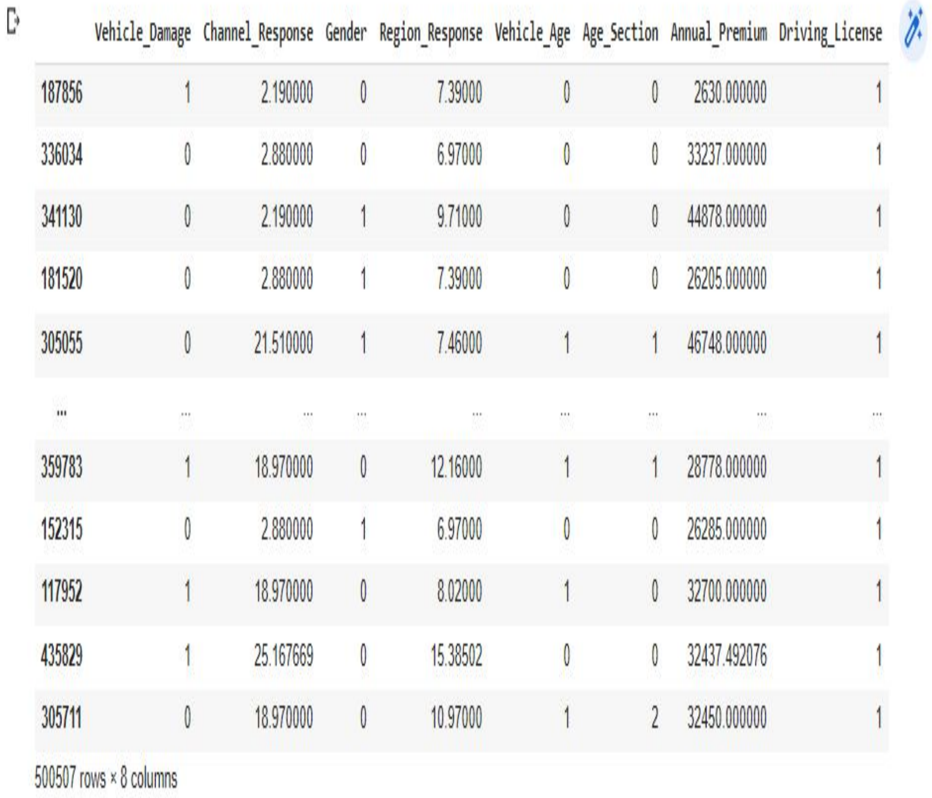
Preparing dataset for Modeling

Task : Binary Classification

Train_set : (500507, 8)

Test_set : (125127,8)

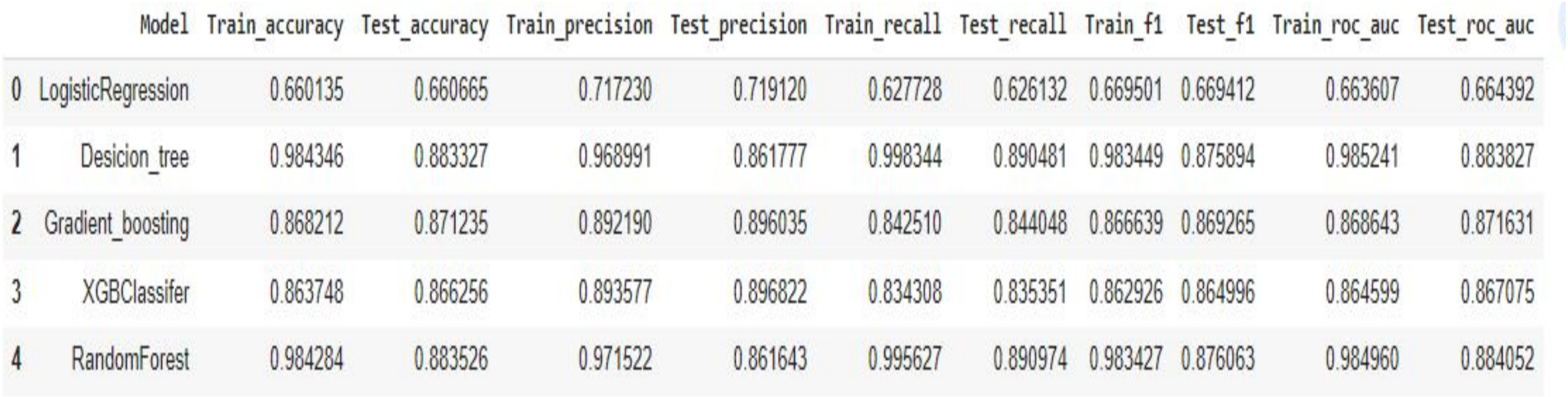
Response : 0/1



	Vehicle_Damage	Channel_Response	Gender	Region_Response	Vehicle_Age	Age_Section	Annual_Premium	Driving_License
187856	1	2.190000	0	7.39000	0	0	2630.000000	1
336034	0	2.880000	0	6.97000	0	0	33237.000000	1
341130	0	2.190000	1	9.71000	0	0	44878.000000	1
181520	0	2.880000	1	7.39000	0	0	26205.000000	1
305055	0	21.510000	1	7.46000	1	1	46748.000000	1
...	...	...	...	...	...	...	...	...
359783	1	18.970000	0	12.16000	1	1	28778.000000	1
152315	0	2.880000	1	6.97000	0	0	26285.000000	1
117952	1	18.970000	0	8.02000	1	0	32700.000000	1
435829	1	25.167669	0	15.38502	0	0	32437.492076	1
305711	0	18.970000	0	10.97000	1	2	32450.000000	1

500507 rows × 8 columns

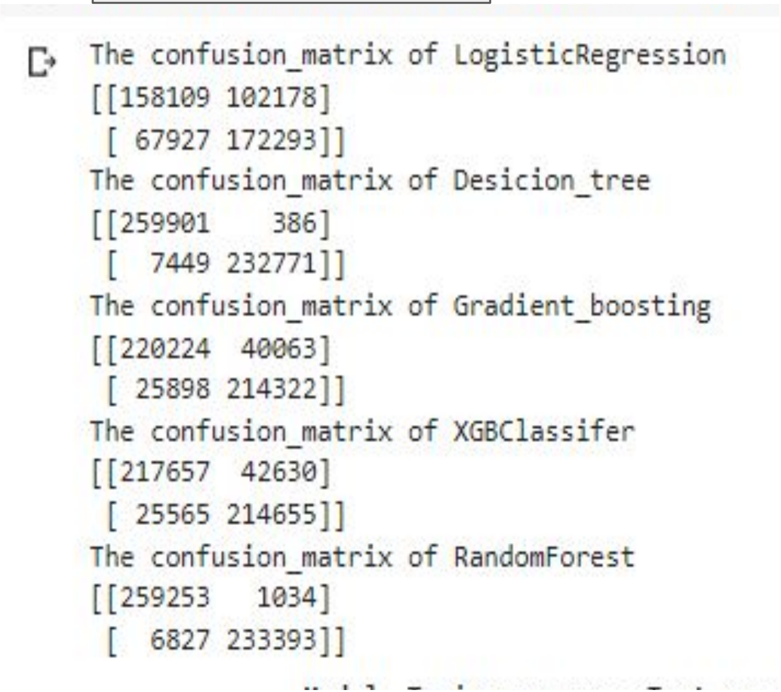
Model Validation and Selection



	Model	Train_accuracy	Test_accuracy	Train_precision	Test_precision	Train_recall	Test_recall	Train_f1	Test_f1	Train_roc_auc	Test_roc_auc
0	LogisticRegression	0.660135	0.660665	0.717230	0.719120	0.627728	0.626132	0.669501	0.669412	0.663607	0.664392
1	Desicion_tree	0.984346	0.883327	0.968991	0.861777	0.998344	0.890481	0.983449	0.875894	0.985241	0.883827
2	Gradient_boosting	0.868212	0.871235	0.892190	0.896035	0.842510	0.844048	0.866639	0.869265	0.868643	0.871631
3	XGBClassifer	0.863748	0.866256	0.893577	0.896822	0.834308	0.835351	0.862926	0.864996	0.864599	0.867075
4	RandomForest	0.984284	0.883526	0.971522	0.861643	0.995627	0.890974	0.983427	0.876063	0.984960	0.884052

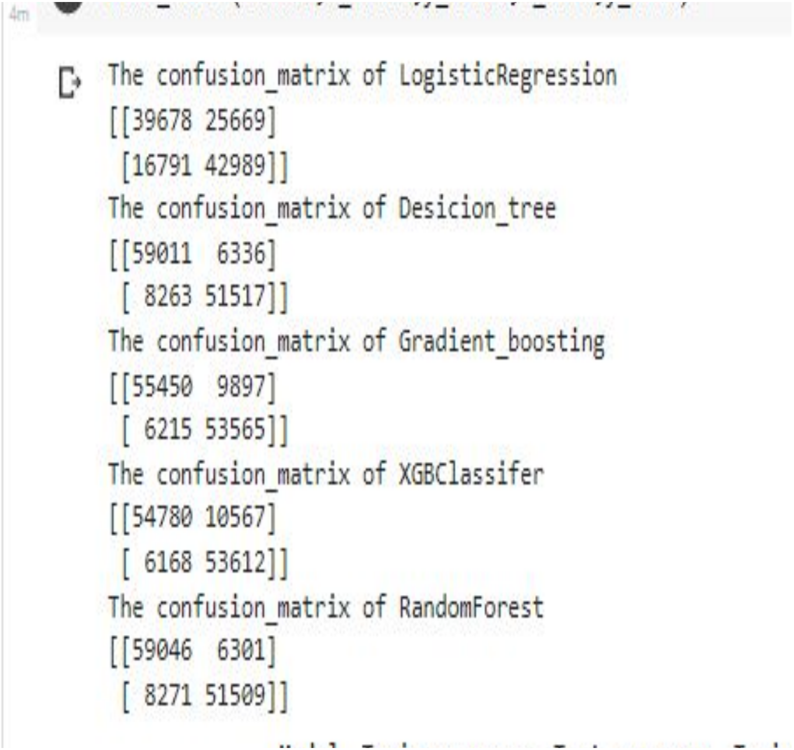
Confusion_matrix

Train_set



```
The confusion_matrix of LogisticRegression
[[158109 102178]
 [ 67927 172293]]
The confusion_matrix of Desicion_tree
[[259901    386]
 [  7449 232771]]
The confusion_matrix of Gradient_boosting
[[220224  40063]
 [ 25898 214322]]
The confusion_matrix of XGBClassifer
[[217657  42630]
 [ 25565 214655]]
The confusion_matrix of RandomForest
[[259253   1034]
 [  6827 233393]]
```

Test_set



```
The confusion_matrix of LogisticRegression
[[39678 25669]
 [16791 42989]]
The confusion_matrix of Desicion_tree
[[59011  6336]
 [ 8263 51517]]
The confusion_matrix of Gradient_boosting
[[55450  9897]
 [ 6215 53565]]
The confusion_matrix of XGBClassifer
[[54780 10567]
 [ 6168 53612]]
The confusion_matrix of RandomForest
[[59046  6301]
 [ 8271 51509]]
```

Observation 1

- As seen in the above table Logistic regression is not giving some great result. Accuracy and roc_auc score is less than 0.70.

- Decision tree and random forest looks overfitted. All training score is above than 0.98 and test score less than 0.88. To overcome this problem we will tuned the hyperparameter.

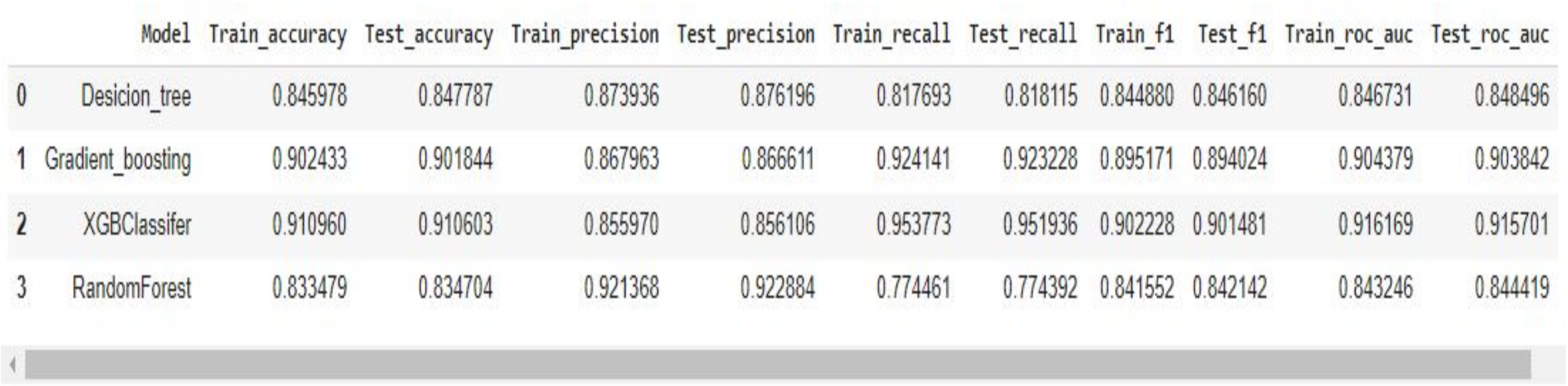
- XGB and GBM perform slightly better than all method without hyperparameter tuning

Observation 1

- For logistic regression we have lots of value in false positive and false negative in both training and testing data in the confusion matrix.

- Confusion matrix of random forest and decision tree classifier perform better for ture negative value in comparison to XGB and GBM classifier.

Model validation and tuning



	Model	Train_accuracy	Test_accuracy	Train_precision	Test_precision	Train_recall	Test_recall	Train_f1	Test_f1	Train_roc_auc	Test_roc_auc
0	Desicion_tree	0.845978	0.847787	0.873936	0.876196	0.817693	0.818115	0.844880	0.846160	0.846731	0.848496
1	Gradient_boosting	0.902433	0.901844	0.867963	0.866611	0.924141	0.923228	0.895171	0.894024	0.904379	0.903842
2	XGBClassifer	0.910960	0.910603	0.855970	0.856106	0.953773	0.951936	0.902228	0.901481	0.916169	0.915701
3	RandomForest	0.833479	0.834704	0.921368	0.922884	0.774461	0.774392	0.841552	0.842142	0.843246	0.844419

Observation 2

- On hyperparameter tuning decision tree and Random Forest classifier overcome with the overfitting problem. The training and testing roc_auc score of decision tree is 0.84. For Random forest classifier it is also 0.84.

- Best hyperparameter estimator for decision tree is, 'max_depth': 9 , 'max_leaf_nodes': 70, 'min_samples_leaf': 2, 'min_samples_split': 10. For random forest best estimator is, 'max_depth': 6, 'min_samples_split': 10

- XGB performance slightly better than GBM. GBM precision score perform slightly better than XGB precision score.

Observation 2

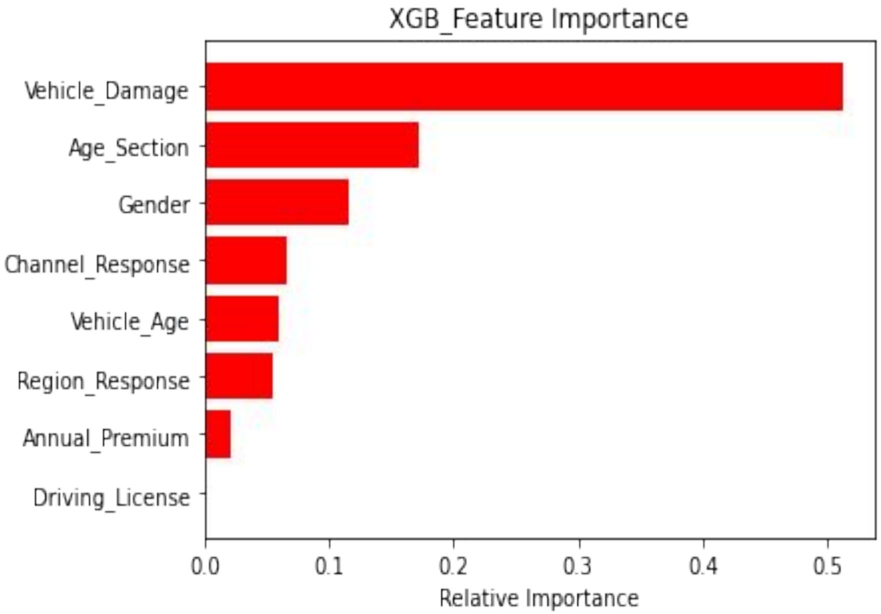
- Best hyperparameter estimator for XGB is, 'learning_rate': 0.5, 'max_depth': 5, 'subsample': 0.9 with roc_auc score is 0.91

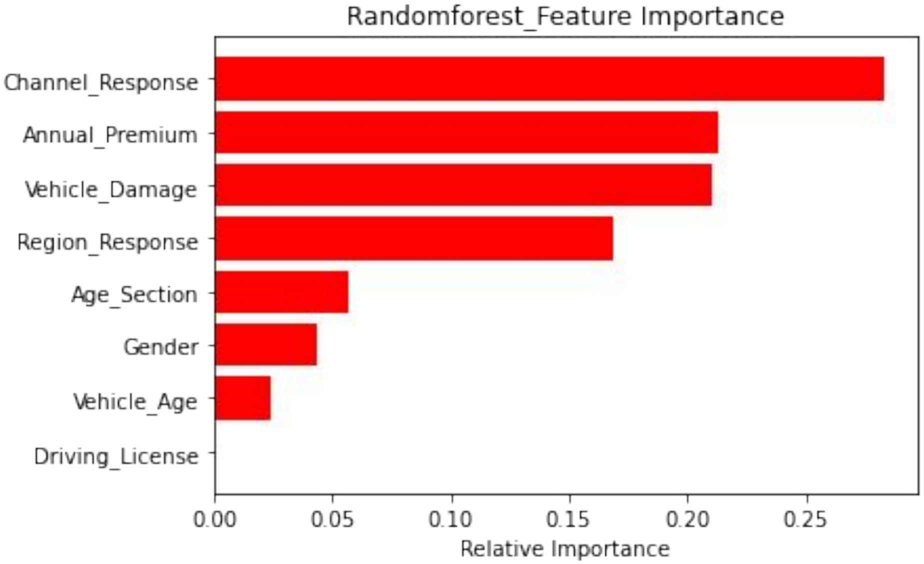
- Best hyperparameter estimator for GBM is, 'n_estimators': 80, 'max_depth': 8 with roc_auc score is 0.90

- XGB Classifier is performing better than other model.

- Finally we select XGB regressor having ROC_AUC score of nearly 0.91 for test data.

Feature Important





Challenges

- Big dataset, take very long time in computation.
- Imbalanced dataset leads model biased.
- Dealing with outliers is major challenge for us.

Conclusion

- Vehicle_damage, Vehicle_age and channel response feature were found to be most relevant for Classify the insurance response for company.
- From vehicle_damage and vehicle age, one can notice people have to pay higher premium for insurance if there car get damage previously or there car is older than 2 years. Also response rate of insurance is also high for such scenario.
- For random forest channel response, Annual premium and Vehicle_damage are important feature for classification.
- For XGB Vehicle_damage, Age section and gender are important feature for classification.

Conclusion

- Logistic regression model does not perform good in this dataset as very few dependent variable is strongly correlated to independent variable. The XGB and GBM provide substantial improvement in predicting the insurance response.The accuracy and roc_auc score is greater than 0.9 for both model.
- Random forest model precision score is 0.91 while recall score is 0.77. This means random forest gives good result for positive insurance response.
- So we used XGboost model for classification. This model can also improve by finer tuning of hyperparameters.

Thank you