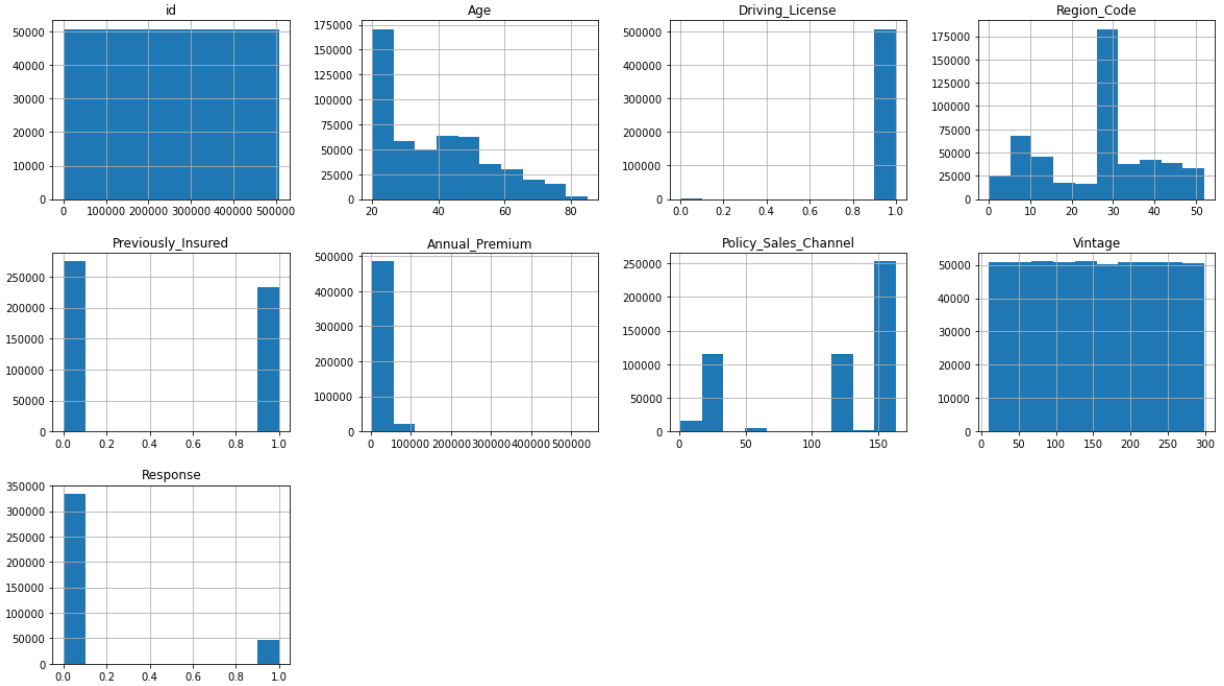
Cross Prediction Analysis

Characteristics of dataset:

- No Null Values
- Data types -
  - Numerical:

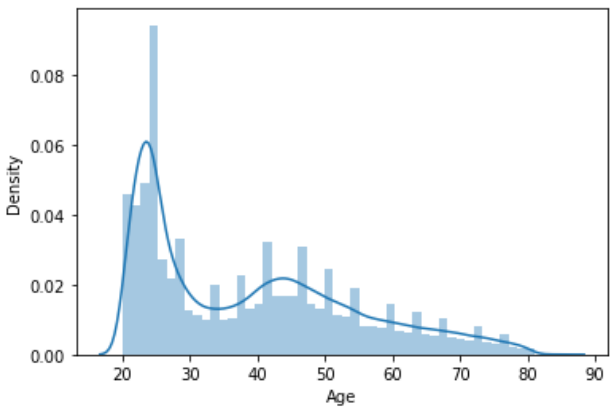
    'id', 'Age', 'Driving_License', 'Region_Code', 'Previously_Insured', 'Annual_Premium', Policy_Sales_Channel', 'Vintage', 'Response'
  - Categorical:

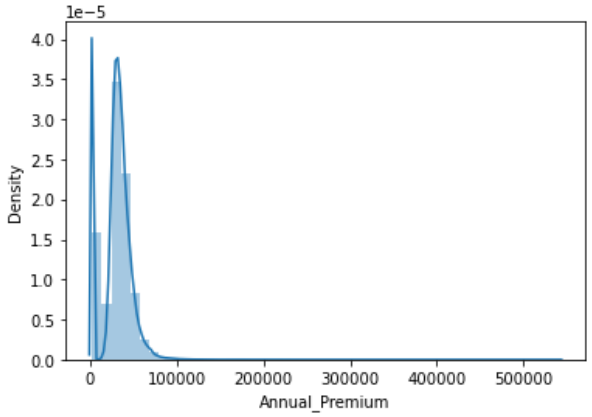
    'Gender', 'Vehicle_Age', 'Vehicle_Damage'

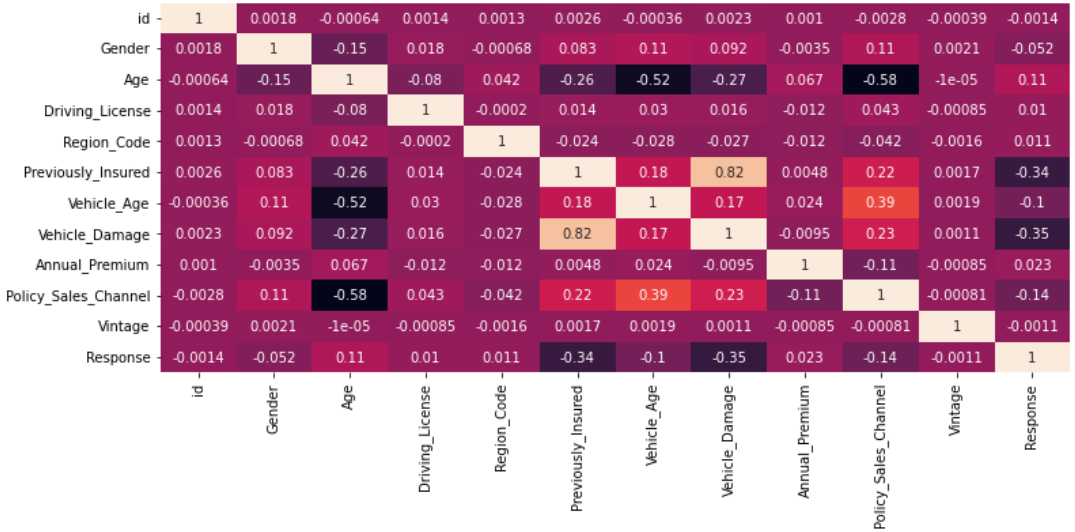


First look insights of data

- Age and Annual premium are skewed
- Driving license doesn't add any value, can be dropped
- Gender, Vehicle age, damage, previously insured to be encoded
- Age, Vintage and Annual Premium can be binned
- After binning, vehicle_Damage', 'Vehicle_Age', 'Age','Gender', 'Previously_Insured', 'Vintage', 'Annual_Premium' can be one hot encoded

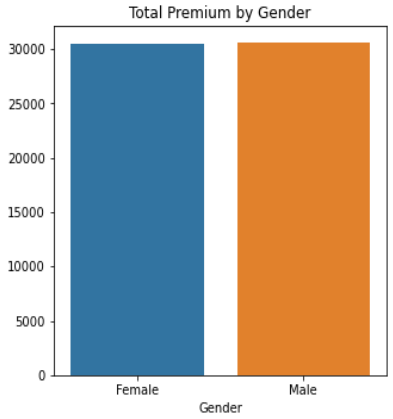




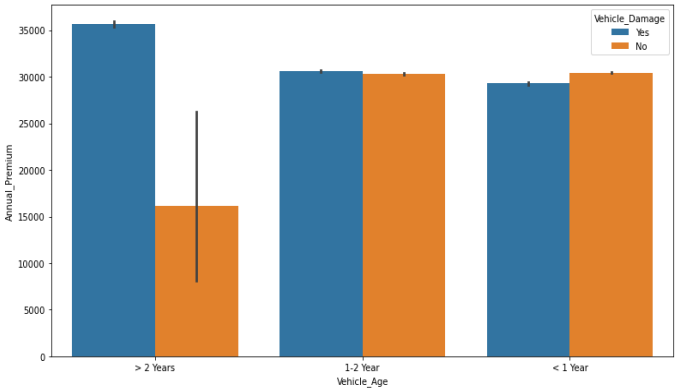


Story board

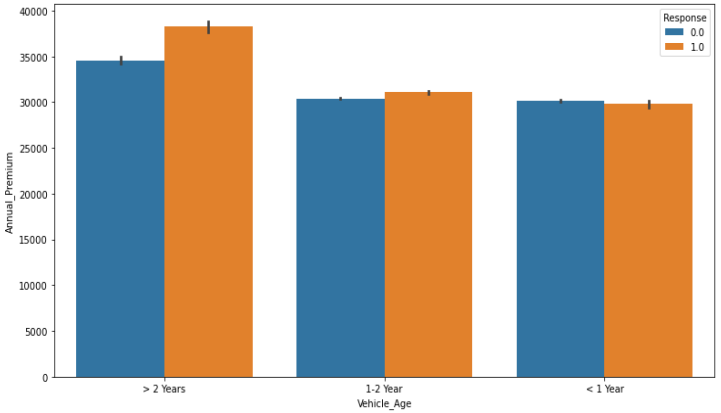
Male population opts for slightly high annual premium



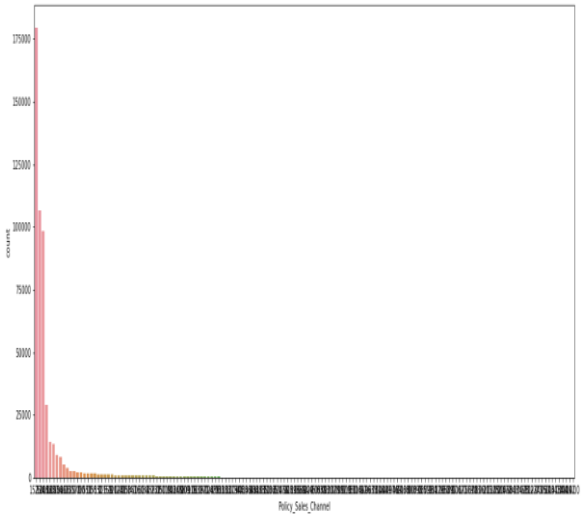
People who own vehicle > 2 years with damage pays for high annual premium



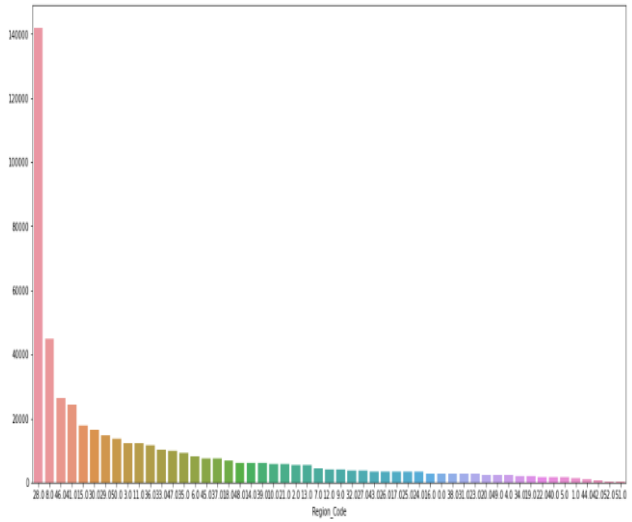
Damaged Vehicle with 1-2 years old and who are insured already are more likely to buy insurance again



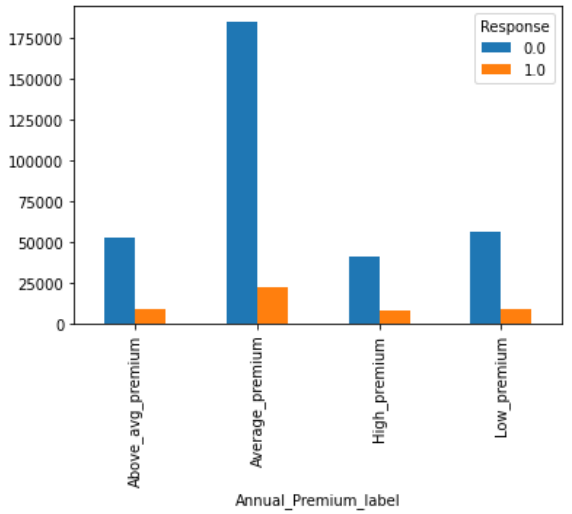
Only few policy channels get more customers



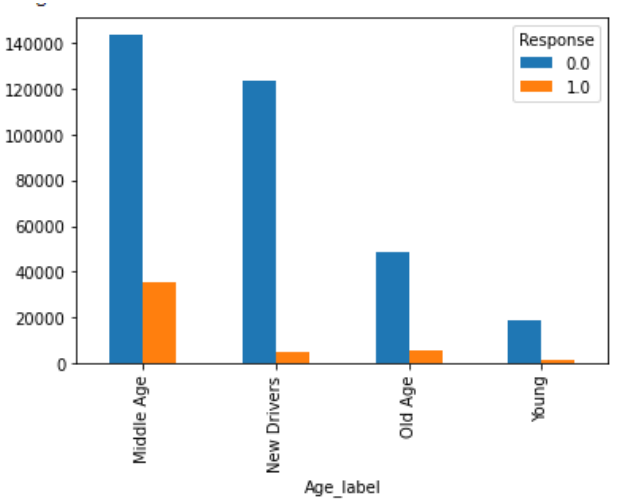
Region code 28 gets the more customers



Generally people do not prefer purchasing high premium



Middle age people are likely to buy premium and own most of the premium



Approach

1. ***Data Cleaning:***
    - Convert all the categorical values to numerical values for some features
2. ***Feature Engineering:***
    - Binning/Discretization –
        1. Pick out the features to perform data binning
            1. Age
            2. Vintage
            3. Annual Premium
        2. Find the best depth of tree to perform binning using decision tree classifier
        3. Check if the bin's relationship with target is linear
        4. Find min and max of bin categories
        5. Label the bins into a new discrete column and assign numerical values to each bins
    - Encoding –
        1. Encode the features with less unique values using one hot encoding
        2. Remove the original values
3. ***Model Selection:*** Try running the classifier models
4. ***Error Metrics***: ROC AUC for all models
5. ***Hyper parameter tuning***: Use Grid search for choosing the right parameters
6. ***Summary:*** Load all the Error metrics to a dataframe for better decision making to choose right model
7. ***Best Model:*** After checking the ROC AUC for all models, it is evident that **CATBOOST** works well for this model, both with and without hypertuning parameters, giving a final result of **85.2** for Test dataset

Result

	Model	Accuracy Score	F1 Score	Precision Score	Recall Score	ROC AUC
0	Logistic Regression	0.878495	0.000144	1.000000	0.000072	0.837371
1	Naive Bayes CB	0.718734	0.425894	0.283184	0.858562	0.835353
2	Random Forest	0.871087	0.107539	0.338673	0.063917	0.829578
3	Decision Tree	0.871131	0.095074	0.323985	0.055712	0.806867
4	AdaBoost	0.878487	0.000000	0.000000	0.000000	0.846477
5	CatBoost	0.877752	0.053113	0.451613	0.028216	0.852118
6	Logistic Regression - Tuned	0.878495	0.000144	1.000000	0.000072	0.837371
7	Cat Boost - Tuned	0.877752	0.053113	0.451613	0.028216	0.852118