Health Insurance Owners Cross Sell Project Built Engine-checkpoint-checkpoint

October 21, 2020

1 Health Insurance Owners Cross Sell Project

- Data Source:Kaggle
- Done by: Aravind R (J&J SD)
- Client Name : Analytics Vidya Health insurance Company Indirect Data Engineer ### Agenda:
- Exploritory data analysis
- Model development
- Model Tuning
- Model Deployement

In [1]:
```
# Necessary Imports
import pandas as pd
import dtale as dt
import numpy as np
import plotly.express as pl
import matplotlib.pyplot as plt
import seaborn as sns
from bokeh.plotting import figure,show,output_notebook
output_notebook()
# Model Related
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,BaggingClassifier,AdaBoostClassifi
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
from sklearn.linear_model import SGDClassifier;
from sklearn.kernel_approximation import RBFSampler;
# Evaluation metrics
from sklearn.metrics import accuracy_score,balanced_accuracy_score,cohen_kappa_score,r
# if there is imbalance in classes
from imblearn.over_sampling import SMOTE
# Model Tuning Engines
# Model Tuning Engine's and Validation Engine(Final)
from sklearn.model_selection import GridSearchCV,RandomizedSearchCV,cross_val_score,St
# Deploy Model to a file in binary format
```

```
import joblib as jb
plt.style.use('seaborn-whitegrid');
```

```
C:\Users\Aravind R\Anaconda3\lib\site-packages\dask\config.py:168: YAMLLoadWarning: calling yar
  data = yaml.load(f.read()) or {}
C:\Users\Aravind R\Anaconda3\lib\site-packages\distributed\config.py:20: YAMLLoadWarning: calli
  defaults = yaml.load(f)
```

In [2]:
```
# Importing the dataset

dataset_main = pd.read_csv('train.csv')

# Gathering information about the shape of the dataset

print(f"The No of rows is :{dataset_main.shape[0]} \nThe No Features is :{dataset_main
```

```
The No of rows is :381109
The No Features is :12
```

In [3]:
```
# Quite a big dataset lets gather the head of the dataset

dataset_main.head(8)
```

Out[3]:

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
0	1	Male	44	1	28.0	0	> 2 Years	Yes	40454.0	26.0	217	1
1	2	Male	76	1	3.0	0	1-2 Year	No	33536.0	26.0	183	0
2	3	Male	47	1	28.0	0	> 2 Years	Yes	38294.0	26.0	27	1
3	4	Male	21	1	11.0	1	< 1 Year	No	28619.0	152.0	203	0
4	5	Female	29	1	41.0	1	< 1 Year	No	27496.0	152.0	39	0
5	6	Female	24	1	33.0	0	< 1 Year	Yes	2630.0	160.0	176	0
6	7	Male	23	1	11.0	0	< 1 Year	Yes	23367.0	152.0	249	0
7	8	Female	56	1	28.0	0	1-2 Year	Yes	32031.0	26.0	72	1

```
2         1
3         0
4         0
5         0
6         0
7         1
```

```
In [4]: # Lets drop the id columns since its not necassary (we already have index)
        dataset_main.drop('id',axis=1,inplace=True)
        dataset_main.head(2)
```

```
Out[4]:   Gender  Age  Driving_License  Region_Code  Previously_Insured Vehicle_Age  \
        0   Male   44                1         28.0                   0   > 2 Years
        1   Male   76                1          3.0                   0    1-2 Year

          Vehicle_Damage  Annual_Premium  Policy_Sales_Channel  Vintage  Response
        0            Yes         40454.0                  26.0      217         1
        1             No         33536.0                  26.0      183         0
```

1.1 Exploritory Data Analysis

- Insights gathering

```
In [5]: # Features of the dataset
        from termcolor import colored
        print(colored("Feature Columns",color='blue'))
        for i,j in enumerate(dataset_main.columns):
            print(i+1,j)
```

```
Feature Columns
1 Gender
2 Age
3 Driving_License
4 Region_Code
5 Previously_Insured
6 Vehicle_Age
7 Vehicle_Damage
8 Annual_Premium
9 Policy_Sales_Channel
10 Vintage
11 Response
```

```
In [6]: # Lets get the information of the dataset
        dataset_main.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 381109 entries, 0 to 381108
Data columns (total 11 columns):
```

```
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Gender                381109 non-null  object
 1   Age                   381109 non-null  int64
 2   Driving_License       381109 non-null  int64
 3   Region_Code           381109 non-null  float64
 4   Previously_Insured    381109 non-null  int64
 5   Vehicle_Age           381109 non-null  object
 6   Vehicle_Damage        381109 non-null  object
 7   Annual_Premium        381109 non-null  float64
 8   Policy_Sales_Channel  381109 non-null  float64
 9   Vintage               381109 non-null  int64
 10  Response              381109 non-null  int64
dtypes: float64(3), int64(5), object(3)
memory usage: 32.0+ MB
```

In [7]:
```
# Summary Statistics of the dataset

dataset_main.describe()
```

Out[7]:

	Age	Driving_License	Region_Code	Previously_Insured	\
count	381109.000000	381109.000000	381109.000000	381109.000000	
mean	38.822584	0.997869	26.388807	0.458210	
std	15.511611	0.046110	13.229888	0.498251	
min	20.000000	0.000000	0.000000	0.000000	
25%	25.000000	1.000000	15.000000	0.000000	
50%	36.000000	1.000000	28.000000	0.000000	
75%	49.000000	1.000000	35.000000	1.000000	
max	85.000000	1.000000	52.000000	1.000000	

	Annual_Premium	Policy_Sales_Channel	Vintage	Response
count	381109.000000	381109.000000	381109.000000	381109.000000
mean	30564.389581	112.034295	154.347397	0.122563
std	17213.155057	54.203995	83.671304	0.327936
min	2630.000000	1.000000	10.000000	0.000000
25%	24405.000000	29.000000	82.000000	0.000000
50%	31669.000000	133.000000	154.000000	0.000000
75%	39400.000000	152.000000	227.000000	0.000000
max	540165.000000	163.000000	299.000000	1.000000

In [8]:
```
# Checking for null values

dataset_main.isna().sum()
```

Out[8]:
```
Gender                0
Age                   0
Driving_License       0
Region_Code           0
```

```
Previously_Insured      0
Vehicle_Age             0
Vehicle_Damage          0
Annual_Premium          0
Policy_Sales_Channel    0
Vintage                 0
Response                0
dtype: int64
```

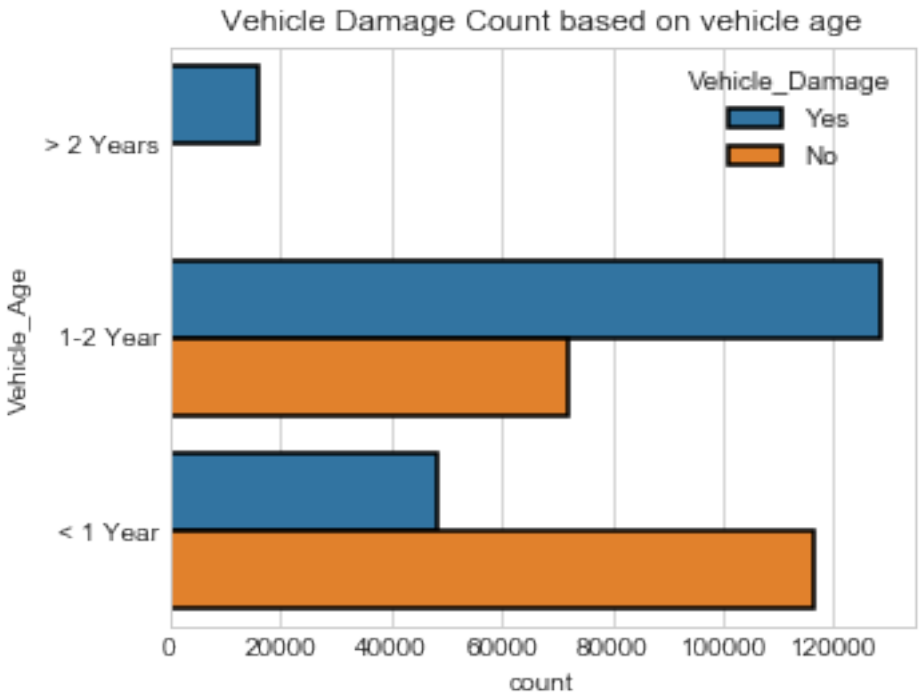
- No Null values are present

```
In [9]: # Check of class imbalance
        print("Class Distributions")
        print()
        print(dataset_main.Response.value_counts())
        print()
        print(f"Event Rate of Minority class(To be predicted): {(46710/381109)*100:.2f}%")
```

```
Class Distributions

0    334399
1     46710
Name: Response, dtype: int64

Event Rate of Minority class(To be predicted): 12.26%
```

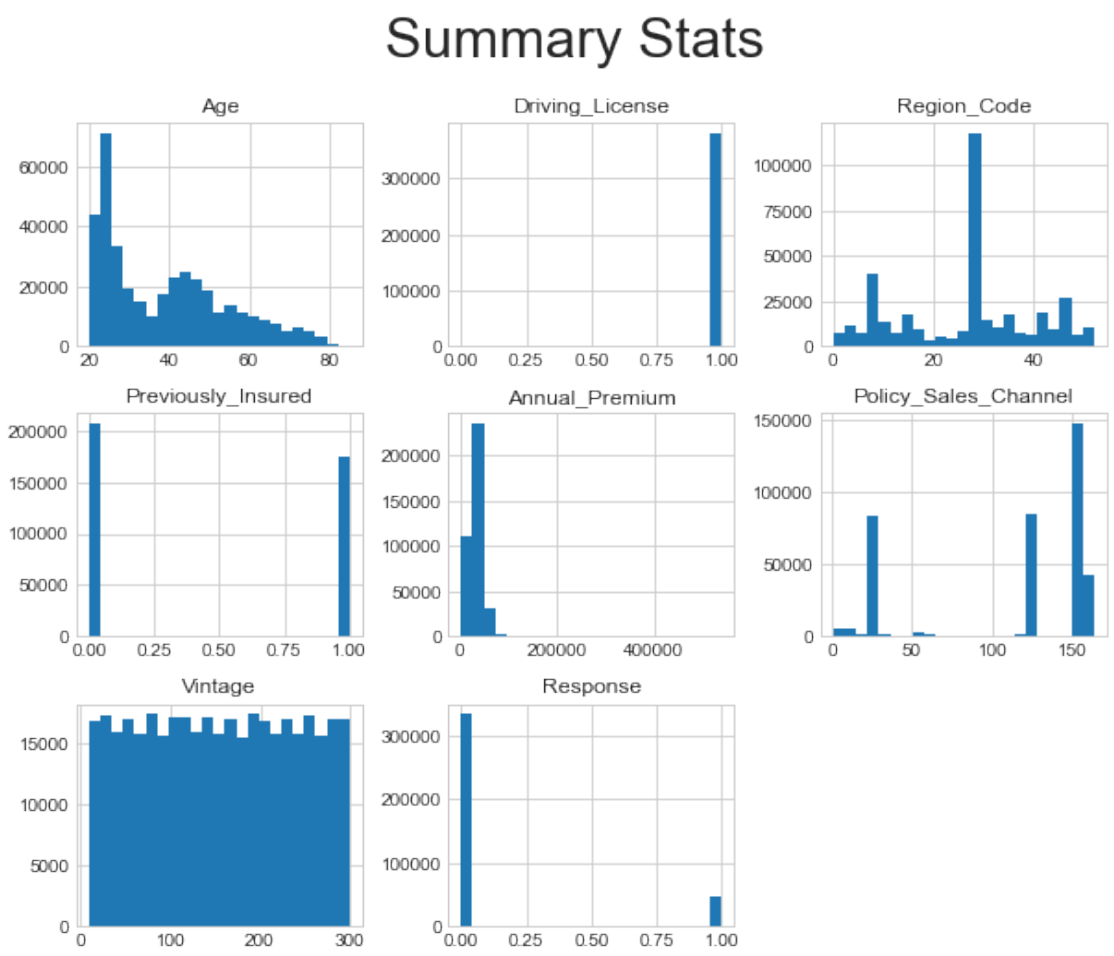
- Eventhough we have greater than 5% event rate its best we go over Over sampling because it increases Accuracy and Balanced Accuracy
- Any how we try both with oversampling and without oversampling and compare the results

```
In [10]: # Lets check customers vechile damage based on vehicle age
         plt.figure(figsize=(5,4))
         sns.countplot(y=dataset_main.Vehicle_Age,linewidth=1.6,edgecolor='black',hue=dataset_
         plt.title("Vehicle Damage Count based on vehicle age");
```

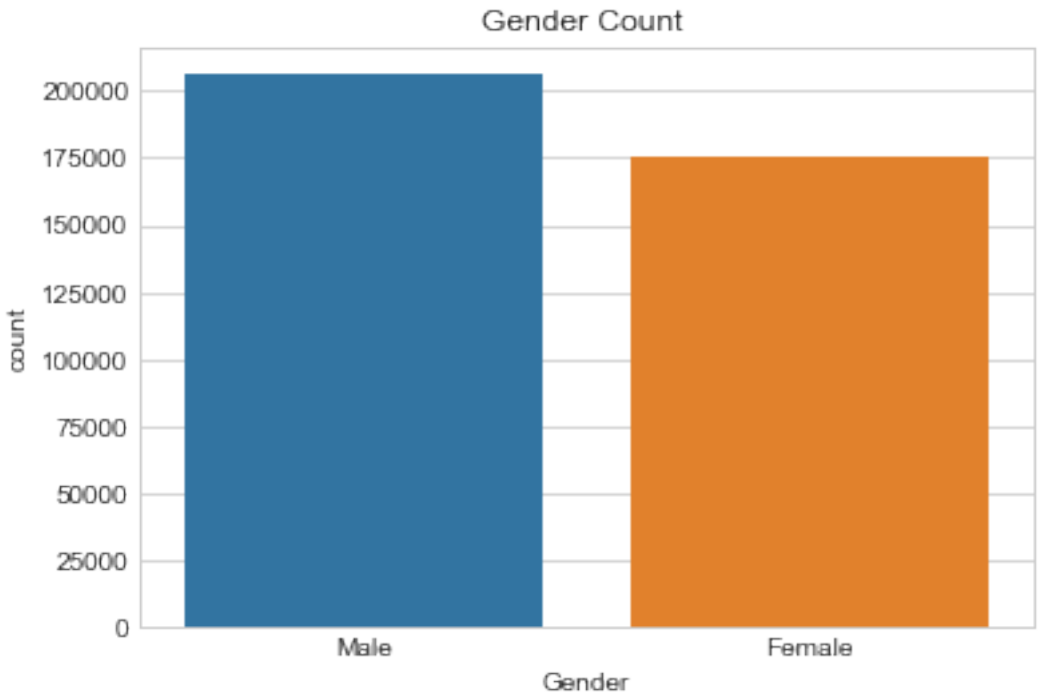


- Fron the above visualization vechile age between 1-2 are prone to vehicle damage as expected and have high chance to respond for insurance

```
In [11]: # Histogram plot
         ax = dataset_main.hist(figsize=(10,8),bins=23);
         plt.suptitle("Summary Stats",fontsize=30);
```

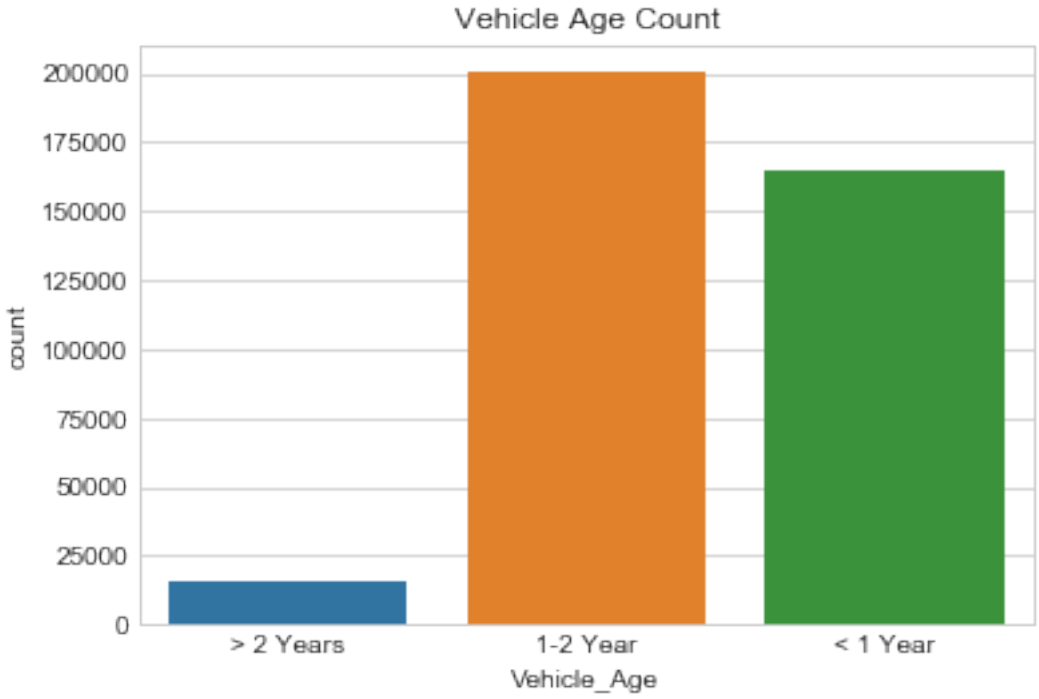


In [12]:
```
# Seeing Category distribution

sns.countplot(dataset_main.Gender);
plt.title("Gender Count ");
```



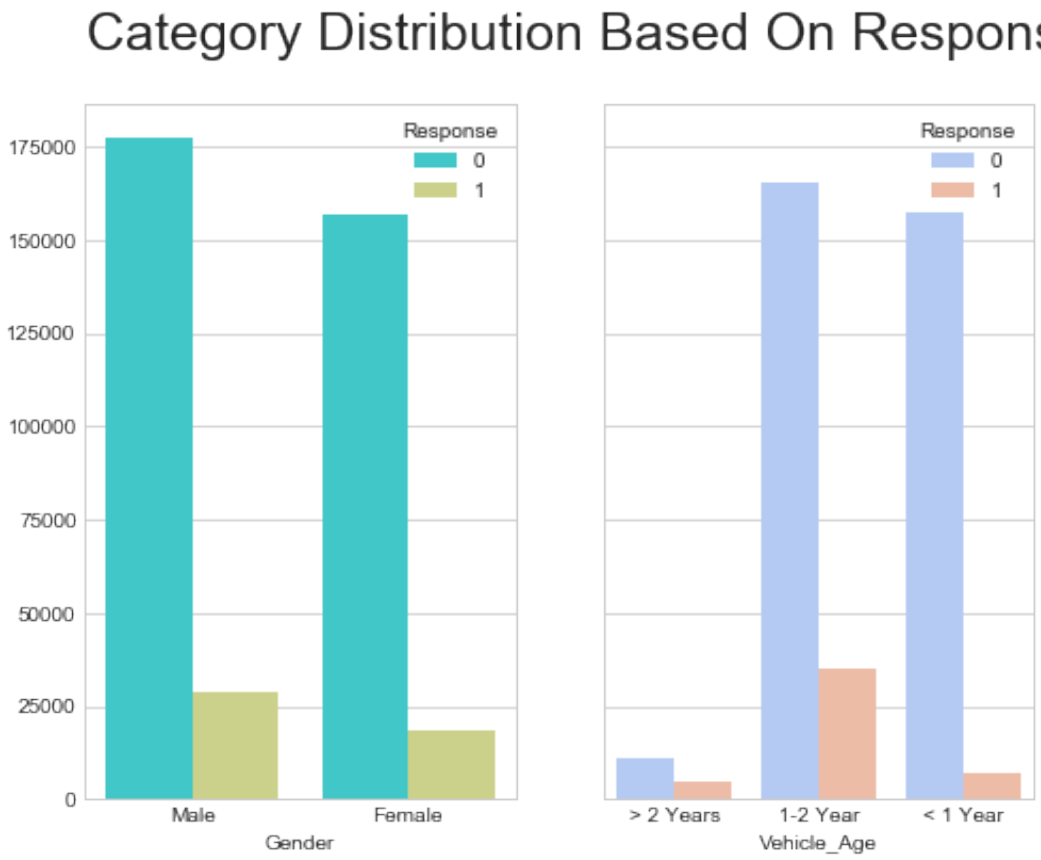
- More Male Samples compared to Female

```
In [13]: sns.countplot(dataset_main.Vehicle_Age)
         plt.title("Vehicle Age Count");
```

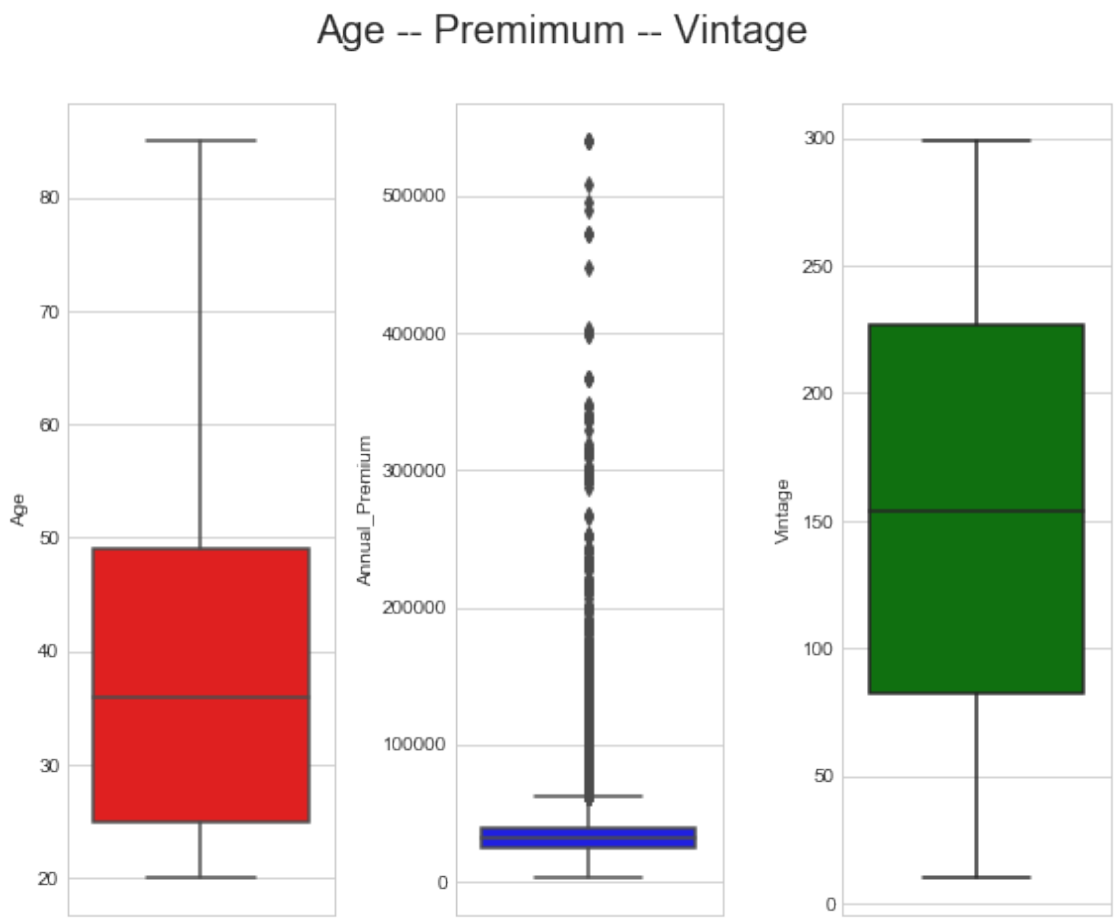


- There area more vehicles of the age 1-2

In [14]:
```
# Seeing Distribution based on response

fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(8,6),sharey=True)
sns.countplot(dataset_main.Gender,hue=dataset_main.Response,ax=ax[0],palette='rainbow
sns.countplot(dataset_main.Vehicle_Age,hue=dataset_main.Response,ax=ax[1],palette='co
ax[0].set_ylabel("");
ax[1].set_ylabel("");
fig.suptitle("      Category Distribution Based On Response",fontsize=25,horizontalal
```

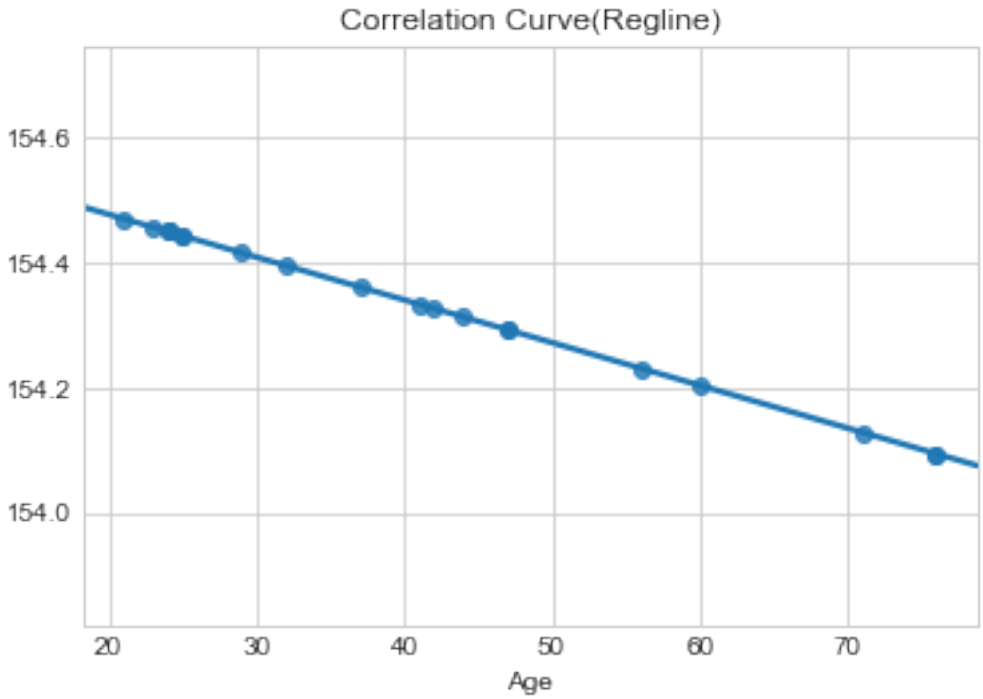


In [15]:
```
fig,ax = plt.subplots(nrows=1,ncols=3,figsize=(8,6))
sns.boxplot(dataset_main.Age,ax=ax[0],orient='v',color='r')
sns.boxplot(dataset_main.Annual_Premium,ax=ax[1],orient='v',color='blue')
sns.boxplot(dataset_main.Vintage,ax=ax[2],orient='v',color='g')
plt.tight_layout()
plt.suptitle("Age -- Premimum -- Vintage",y=1.08,fontsize=20);
```

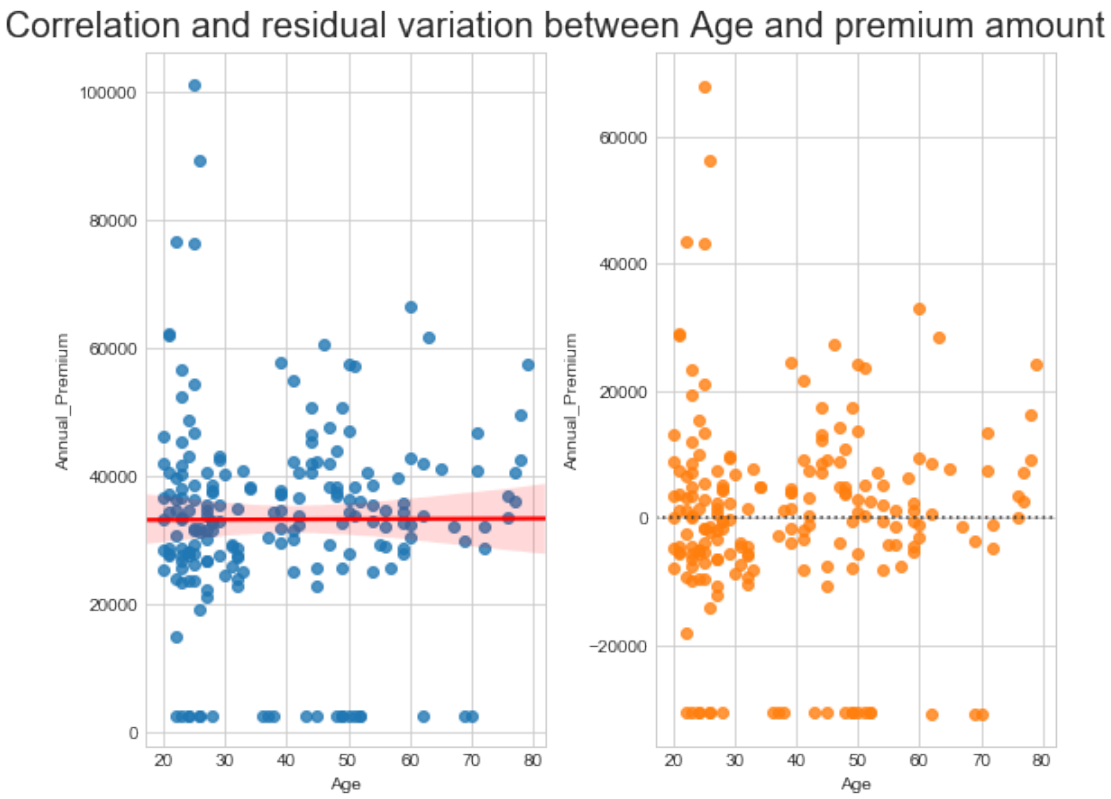



```
In [16]: # for plotting purposes
         No_response = dataset_main[dataset_main['Response']==0]
         response = dataset_main[dataset_main['Response']==1]

In [17]: m,c = np.polyfit(dataset_main['Age'],dataset_main['Vintage'],1)
         q = m*dataset_main['Age']+c

In [18]: # Imbalance Determination
         from bokeh.models import ColumnDataSource,CategoricalColorMapper
         from bokeh.layouts import gridplot
         source = ColumnDataSource(dict(x=[d for d in dataset_main.Age],y=[d for d in dataset_m
         color_map = CategoricalColorMapper(factors=['0','1'],palette=['red','blue'])

         p = figure(plot_width=500,plot_height=500,title='Class imbalance',tools="")
         p.circle(x='x',y='y',source=source,color={'field':'label','transform':color_map},legen
         d = figure(plot_width=200,plot_height=200,title='Regression Curve')
         show(p)

BokehDeprecationWarning: 'legend' keyword is deprecated, use explicit 'legend_label', 'legend_
```

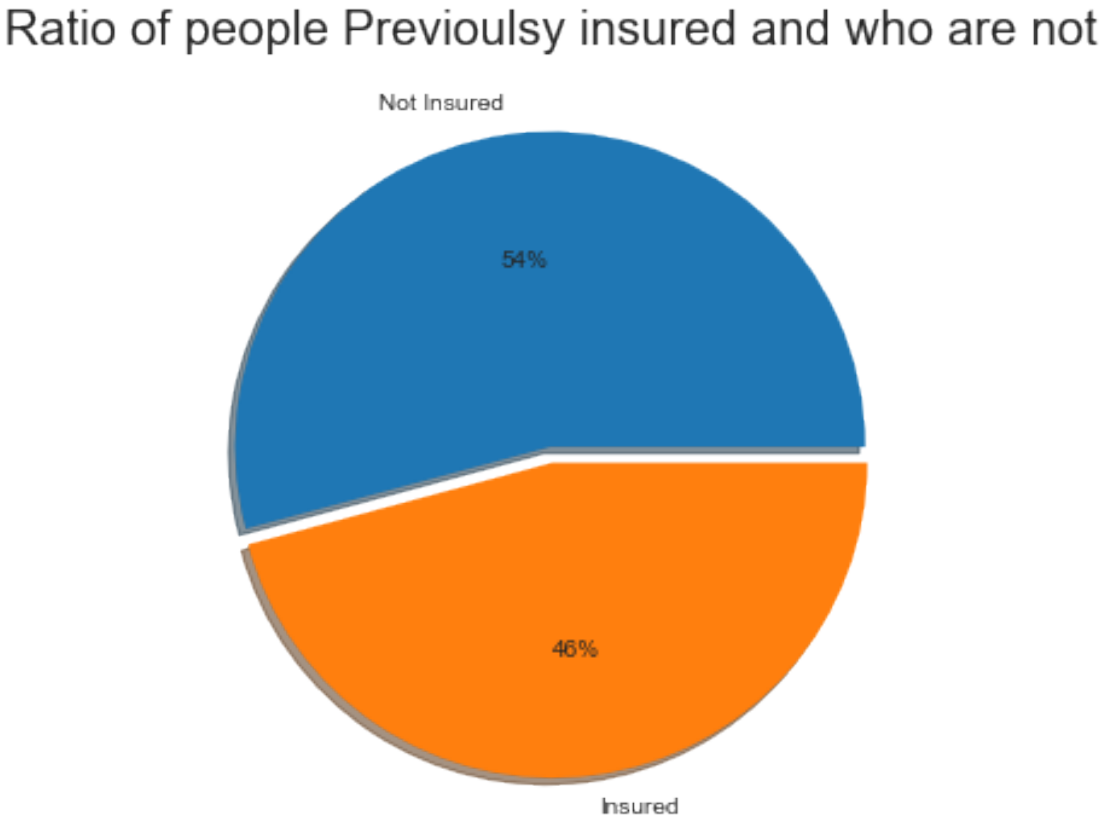
In [19]:
```
plt.title("Correlation Curve(Regline)")
sns.regplot(dataset_main['Age'][:20],q[:20]);
plt.ylabel("");
```



In [20]:
```
# Correlation and residual variation between Age and premium amount
fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(8,6),sharey=False)
sns.regplot(dataset_main['Age'][:200],dataset_main.Annual_Premium[:200],ax=ax[0],line
sns.residplot(dataset_main['Age'][:200],dataset_main.Annual_Premium[:200],ax=ax[1])
plt.suptitle("Correlation and residual variation between Age and premium amount",y=1.
plt.tight_layout()
```



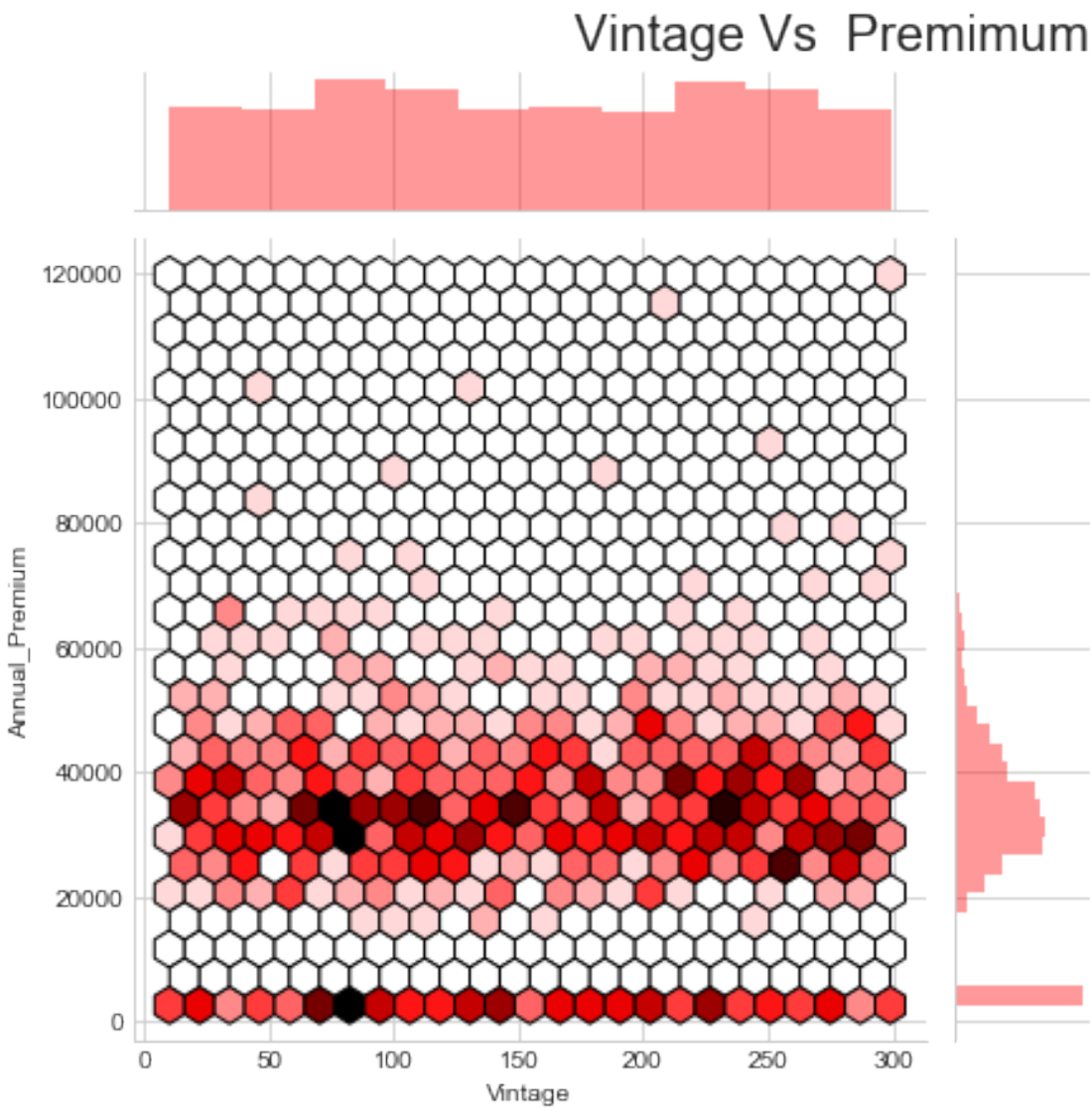
In [21]:
```
# Ratio of people Previoulsy insured and who are not:

R1,R2 = dataset_main['Previously_Insured'].value_counts()[0]/len(dataset_main),dataset
plt.figure(figsize=(6,6))
ax = plt.pie([R1,R2],labels=['Not Insured','Insured'],explode=[0.02,0.03],autopct="%.
plt.title("Ratio of people Previoulsy insured and who are not",fontsize=20);
```

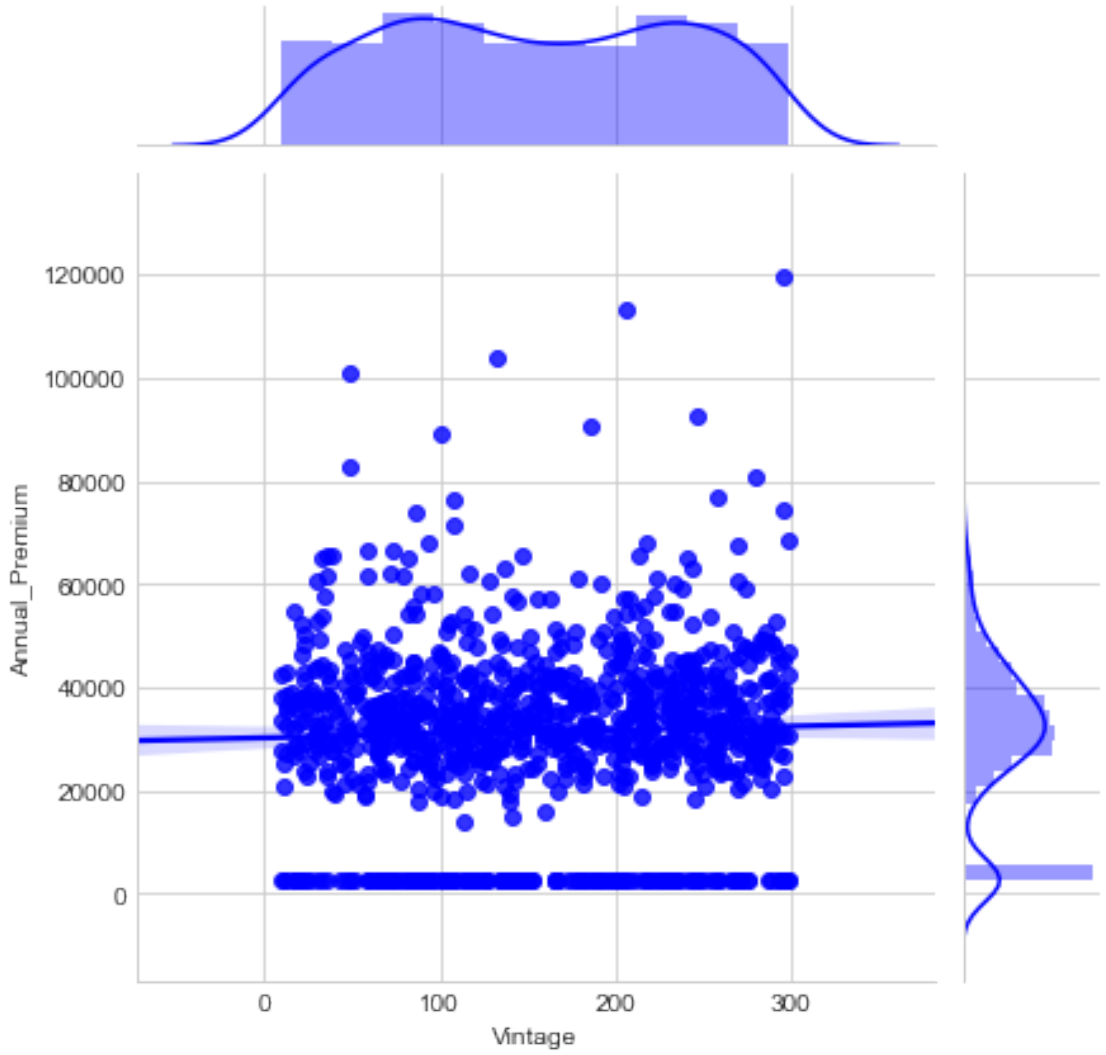


- From the above pie chart 46% of them have previously insured vehicle whereas the remaining part is the one where we must focus the 54% of customers

```
In [22]: # Lets goo deep and consider how does vehicle damage in the past impact on annual pre
         unique_values = ['No','Yes']
         mean_1 = dataset_main[dataset_main['Vehicle_Damage']=='No']['Annual_Premium'].mean()
         mean_2 = dataset_main[dataset_main['Vehicle_Damage']=='Yes']['Annual_Premium'].mean()

         alfa = figure(x_range=unique_values,plot_width=452,plot_height=300,tools="",toolbar_l
         alfa.vbar(x=unique_values,top=[mean_1,mean_2],width=0.5)
         show(alfa)
```

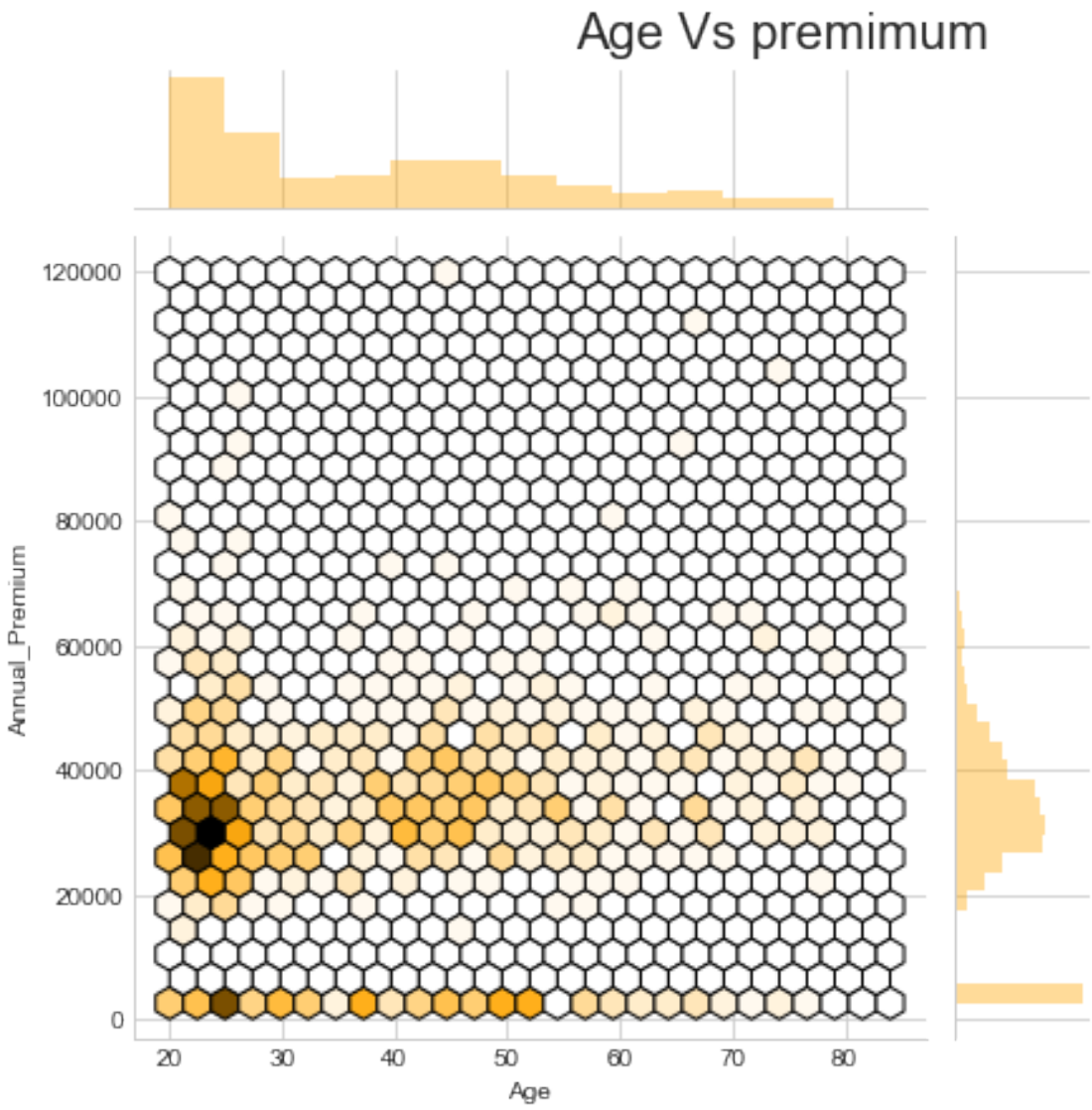
```
In [23]: # No of days with the company and Annual premimum
         p=sns.jointplot(x=dataset_main['Vintage'][:1000],y=dataset_main['Annual_Premium'][:10
         p.fig.suptitle("Vintage Vs  Premimum",y=1.03,fontsize=20,horizontalalignment='left')

         sns.jointplot(x=dataset_main['Vintage'][:1000],y=dataset_main['Annual_Premium'][:1000
```

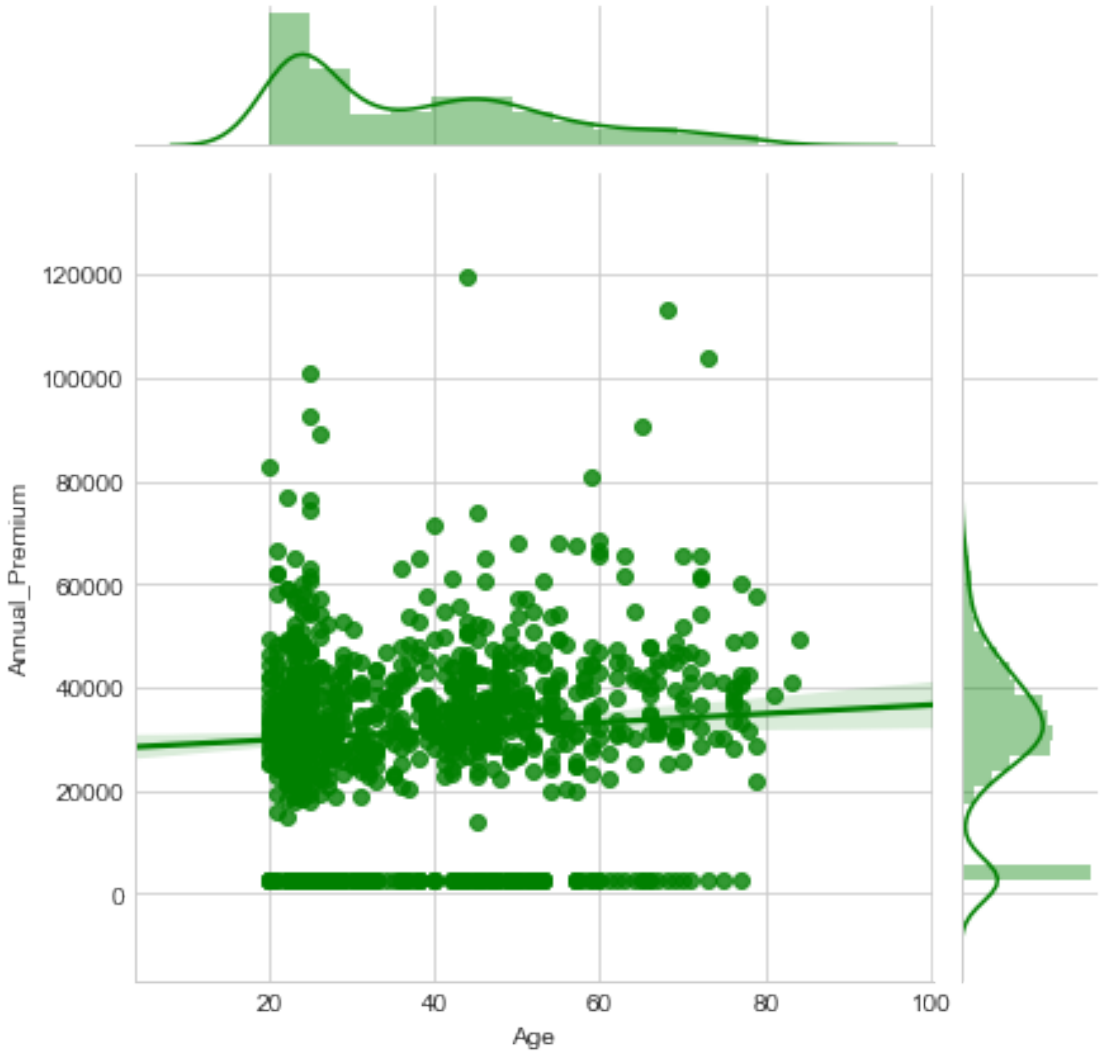

Vintage Vs Premimum
Annual_Premium
120000
100000
80000
60000
40000
20000
0
0
50
100
150
200
250
300
Vintage



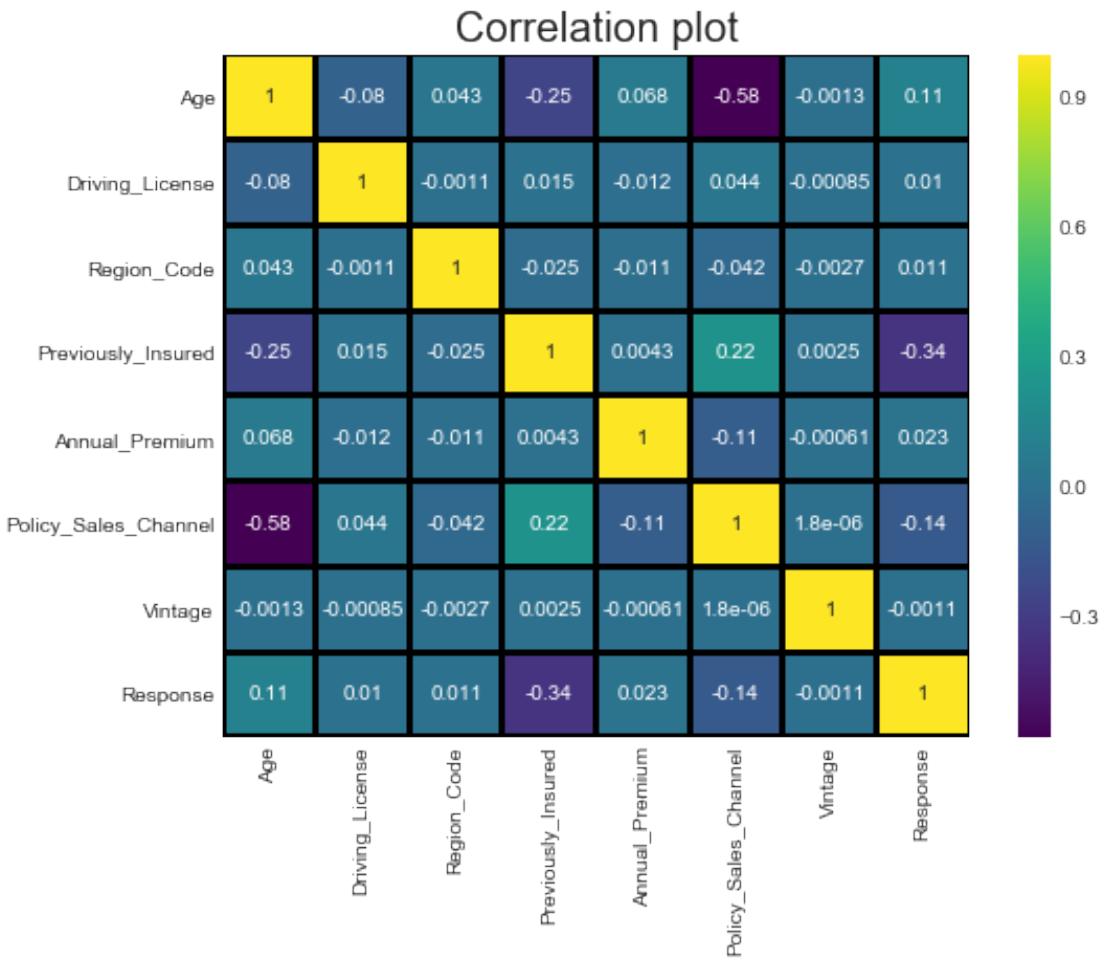
- Vintage days and Amount doesnot have any kind of correlation and Constant amount is maintained Below $60,000

```
In [24]: # Check the same for age and Premimum'
         p=sns.jointplot(x=dataset_main['Age'][:1000],y=dataset_main['Annual_Premium'][:1000],
         p.fig.suptitle("Age Vs premimum",y=1.03,fontsize=20,horizontalalignment='left')
         sns.jointplot(x=dataset_main['Age'][:1000],y=dataset_main['Annual_Premium'][:1000],kin
```


Age Vs premimum
Annual_Premium
120000
100000
80000
60000
40000
20000
0
20
30
40
50
60
70
80
Age



In [25]:
```
# Correlation plot
plt.figure(figsize=(8,6))
ax = sns.heatmap(dataset_main.corr(),annot=True,cmap='viridis',linecolor='black',linew
plt.title("Correlation plot",fontsize=20,y=1.01,)
bottom,top = ax.get_ylim()
ax.set_ylim(bottom+0.5,top-0.5);
```




```
In [26]: # Lets Check whether correlation is true for one entry as per heat map above
         from scipy import stats
         print(f"The Correlation value is:{stats.pearsonr(dataset_main.Age,dataset_main.Annual_
```

```
The Correlation value is:0.06750700155668278
The P-value is 0.0
```

From the above the correlation matrix is correct and Age implies a +ve increase in the gradient of Premimum

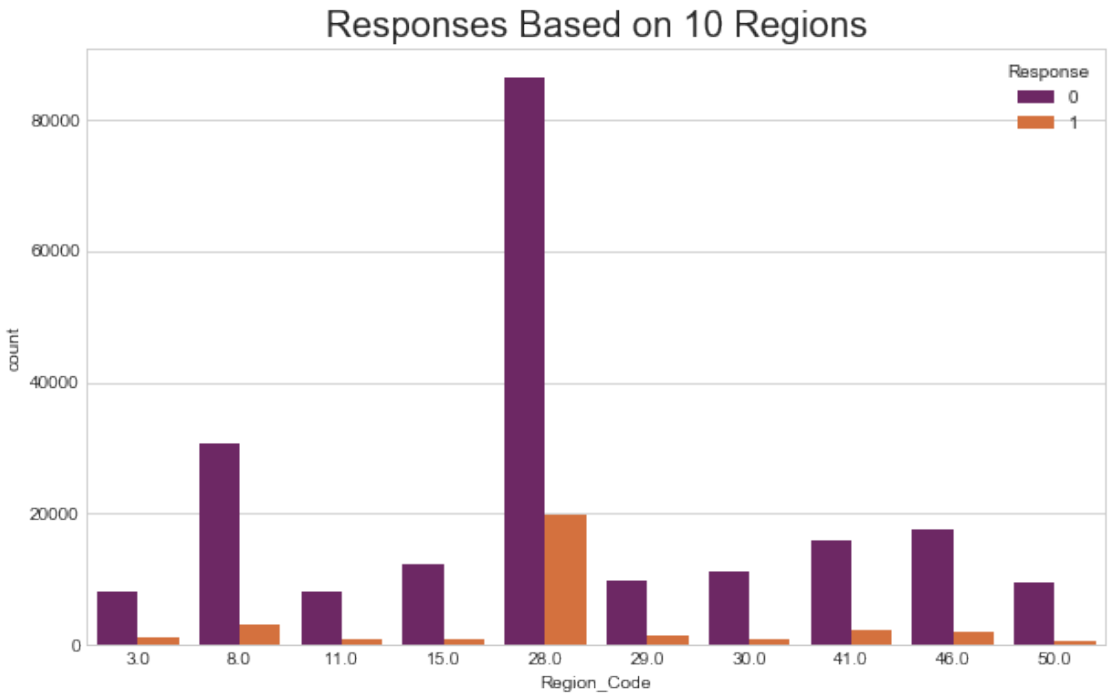
`Multidimensional Analysis`

- We are going to gather the details of responses for Top 10 Regions

```
In [27]: # Taking the top22 respones region code from the dataframe
         top22 = dataset_main.Region_Code.value_counts().index[:10]
```

```
indexs = []
for i in range(len(dataset_main)):
    for j in top22:
        if dataset_main.at[i,'Region_Code']==j:
            indexs.append(i)
```

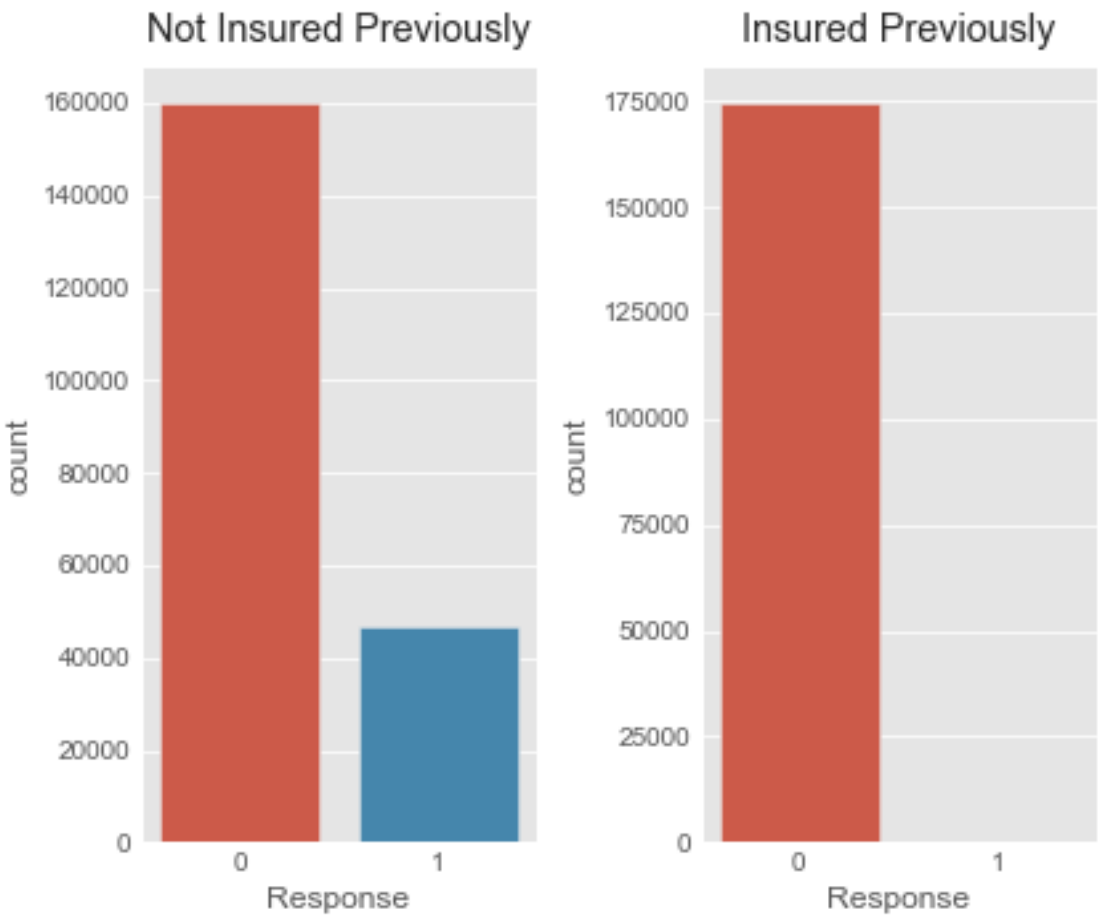
In [28]:
```
# Plotting the Requied
newFrame = dataset_main.iloc[indexs,:]
plt.figure(figsize=(10,6))
sns.countplot(newFrame['Region_Code'],hue=newFrame["Response"],palette='inferno');
plt.title("Responses Based on 10 Regions",fontsize=20);
sns.set_style("white");
```



- From the Above We can say Region code 28 has major responses and are in motive to apply for this Insurance(vehicle)
- Since we know that 46% of them have not previously insured lets consider them for the time being

In [29]:
```
# NOt insured customers data
plt.style.use("ggplot")
not_insured = dataset_main[dataset_main["Previously_Insured"]==0]
# Insured customers data
fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(6,5))
insured = dataset_main[dataset_main["Previously_Insured"]==1]
```

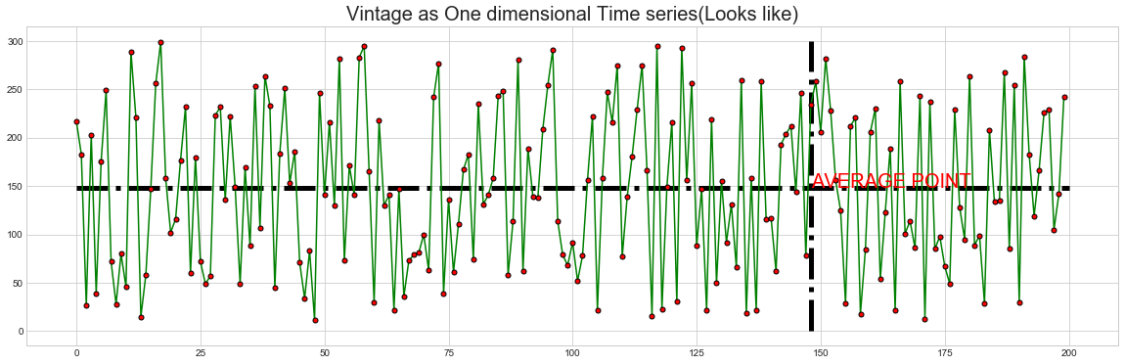
```
sns.countplot(not_insured['Response'],ax=ax[0])
sns.countplot(insured['Response'],ax=ax[1])
ax[0].set_title("Not Insured Previously",y=1.01)
ax[1].set_title("Insured Previously",y=1.01)
plt.tight_layout()
```



- Note a intresting thing 158 customers who have insured already have shown a postive response two things can be observed from this
- First and most it can be sampling error (error made during data gathering) possibility is high
- or May be because of any update in new insurance policy they might opt for it
- Let us consider the second scenario for the time being

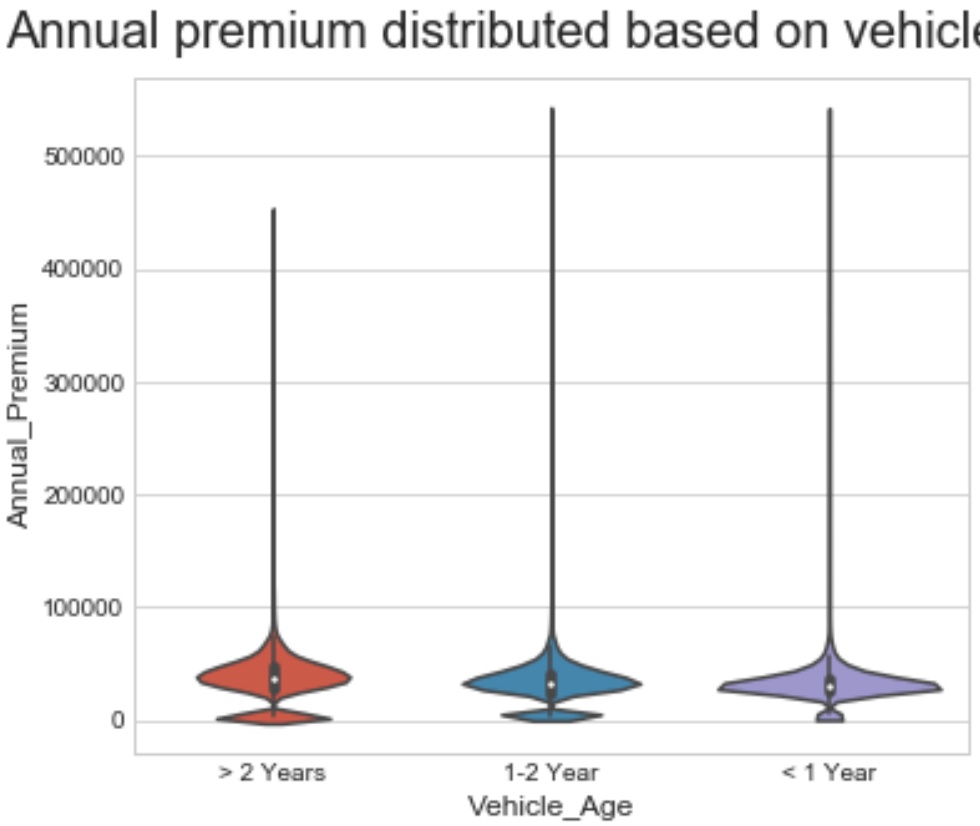
In [30]:
```
# just for fun time series but Vintage (doesnt make any sense anyhow lets see)
plt.style.use("seaborn-whitegrid")
plt.figure(figsize=(20,6))
plt.plot(dataset_main['Vintage'][:200],c='green',marker='.',markersize=10,markerfacec
```

```
plt.hlines(y=148,xmin=0,xmax=200,linestyles='dashdot',linewidth=5)
plt.vlines(x=148,ymin=0,ymax=300,linestyles='dashdot',linewidth=5)
plt.annotate(s="AVERAGE POINT",xy=(148,148),c='red',fontsize=20)
plt.title("Vintage as One dimensional Time series(Looks like)",fontsize=20);
```



- Acutally this analysis should not be undertaken but to show whether vintage point looks like time series signal just for some points we have emphazised this

In [31]:
```
# Annual premium distributed based on vehicle age
plt.figure(figsize=(6,5))
sns.violinplot(dataset_main['Vehicle_Age'],dataset_main['Annual_Premium'])
plt.title("Annual premium distributed based on vehicle age",fontsize=20,y=1.02);
```



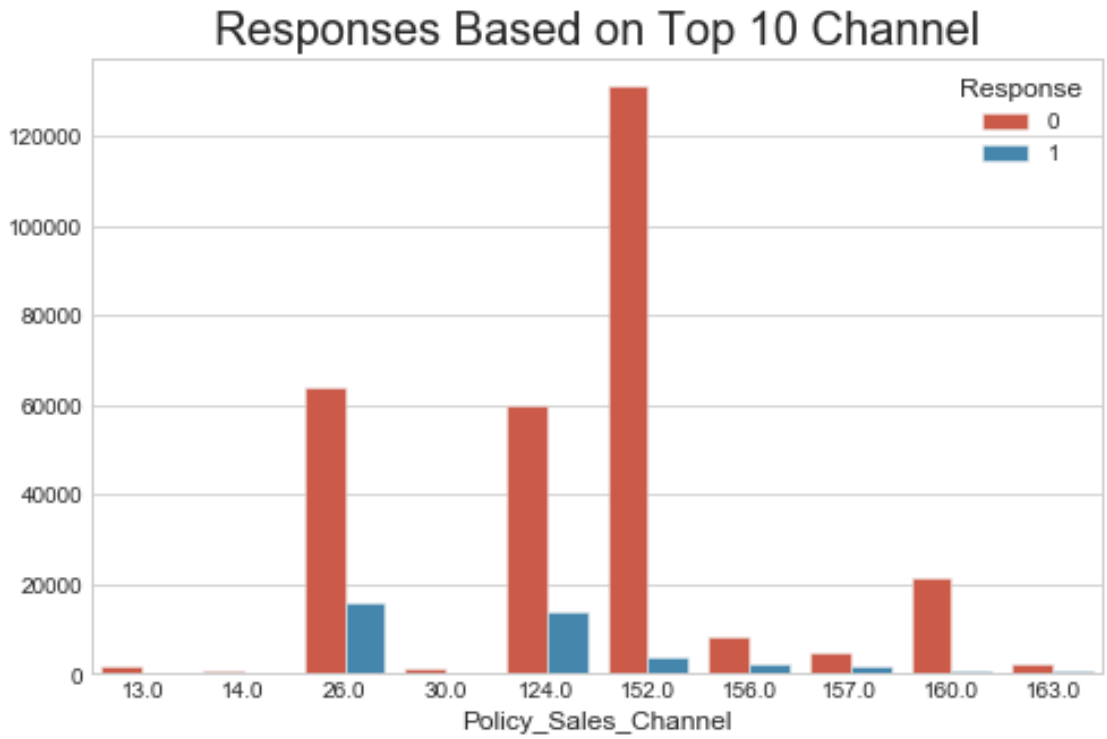

```
In [32]: dataset_main.head()
```

Out[32]:

	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
0	Male	44	1	28.0	0	> 2 Years	Yes	40454.0	26.0	217	1
1	Male	76	1	3.0	0	1-2 Year	No	33536.0	26.0	183	0
2	Male	47	1	28.0	0	> 2 Years	Yes	38294.0	26.0	27	1
3	Male	21	1	11.0	1	< 1 Year	No	28619.0	152.0	203	0
4	Female	29	1	41.0	1	< 1 Year	No	27496.0	152.0	39	0

```
In [33]: # Top 10 Policy channel based on response
         top_10_channel = dataset_main.Policy_Sales_Channel.unique()[:10]
         index=[]
         for i in range(len(dataset_main)):
             for j in top_10_channel:
```

```
            if dataset_main.at[i,"Policy_Sales_Channel"]==j:
                index.append(i)
        newframe2 = dataset_main.iloc[index,:]
        plt.figure(figsize=(8,5))
        sns.countplot(newframe2['Policy_Sales_Channel'],hue=dataset_main["Response"]);
        plt.title("Responses Based on Top 10 Channel",fontsize=20);
        plt.ylabel("");
```



In [5]:
```
# Since we gave imbalancing in the dataset we  can do smote oversampling technique to
```

```
# Storing backup
dataset_main2 = dataset_main.copy()
dataset_main2.head(5)
```

Out[5]:

	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
0	Male	44	1	28.0	0	> 2 Years	Yes	40454.0	26.0	217	1
1	Male	76	1	3.0	0	1-2 Year					
2	Male	47	1	28.0	0	> 2 Years					
3	Male	21	1	11.0	1	< 1 Year					
4	Female	29	1	41.0	1	< 1 Year					

```
1              No         33536.0                 26.0      183         0
2             Yes         38294.0                 26.0       27         1
3              No         28619.0                152.0      203         0
4              No         27496.0                152.0       39         0
```

```
In [6]: # Lets encode Gender Vehicle Age,Damage
        for i in dataset_main2.columns:
            if pd.api.types.is_string_dtype(dataset_main2[i]) or pd.api.types.is_object_dtype(
                dataset_main2[i] = pd.Categorical(dataset_main2[i]).codes
```

```
In [7]: dataset_main2.head()
```

```
Out[7]:    Gender  Age  Driving_License  Region_Code  Previously_Insured  Vehicle_Age  \
        0       1   44                1         28.0                   0            2
        1       1   76                1          3.0                   0            0
        2       1   47                1         28.0                   0            2
        3       1   21                1         11.0                   1            1
        4       0   29                1         41.0                   1            1

           Vehicle_Damage  Annual_Premium  Policy_Sales_Channel  Vintage  Response
        0               1         40454.0                  26.0      217         1
        1               0         33536.0                  26.0      183         0
        2               1         38294.0                  26.0       27         1
        3               0         28619.0                 152.0      203         0
        4               0         27496.0                 152.0       39         0
```

```
In [8]: # Encoded Dataset
        dataset_main2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 381109 entries, 0 to 381108
Data columns (total 11 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Gender                381109 non-null  int8
 1   Age                   381109 non-null  int64
 2   Driving_License       381109 non-null  int64
 3   Region_Code           381109 non-null  float64
 4   Previously_Insured    381109 non-null  int64
 5   Vehicle_Age           381109 non-null  int8
 6   Vehicle_Damage        381109 non-null  int8
 7   Annual_Premium        381109 non-null  float64
 8   Policy_Sales_Channel  381109 non-null  float64
 9   Vintage               381109 non-null  int64
 10  Response              381109 non-null  int64
dtypes: float64(3), int64(5), int8(3)
memory usage: 24.4 MB
```

```
In [9]: # Creating Datastructures
        X = dataset_main2.iloc[:,:-1].values
        y = dataset_main2.iloc[:,-1].values

        #Oversampling to reduce class imbalance
        sm = SMOTE()
        X,y = sm.fit_resample(X,y)
```

```
In [13]: # Before smote
         pl.scatter(dataset_main,x='Age',y='Vintage',color='Response')
```

```
In [11]: # After smote
         pl.scatter(x=X[:,1],y=X[:,9],color=y)
```

- From the above graph Nearest neighbours algorthim has been perfectly used by smote for oversampling and we have overcome class imbalances

1.2 Model Development

- Baseline Engine (Random Forest,catboost,.....)

```
In [14]: # Creating a Custom Evaluator

         # Model Performance Evaluator

         def Model_Performance(model,name):

             """
             Custom Performance Engine (For classification Problems) by Aravind R
                 1.accuracy_score
                 2.balanced_accuracy_score
                 3.plot_roc_curve and Score
                 4.plot_precision_recall_curve
                 5.cohen_kappa_score
                 6.log_loss
                 7.Confusion Matrix
                 8.Classification Report
                 Note: Dataset Must be splitted into X_train,X_test and y_train,y_test
             """

             print(f"\t\t {name} Performance Outline \t\t")
             pred_train = model.predict(X_train)
             pred_train_proba = model.predict_proba(X_train)

             pred_test = model.predict(X_test)
             pred_test_proba = model.predict_proba(X_test)

             print()
             print("\t Training Results\t")
```

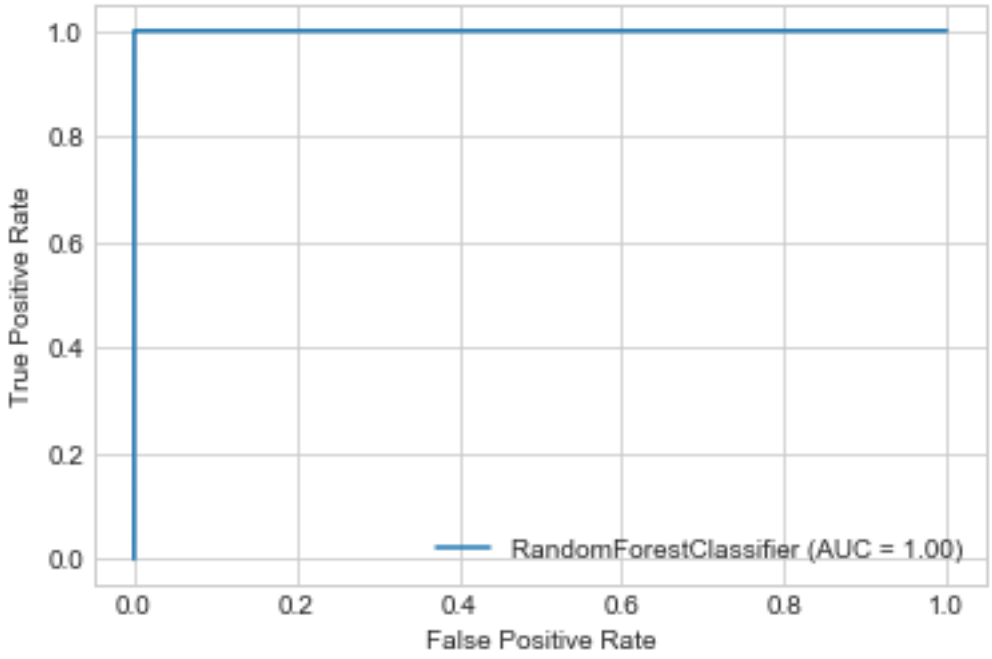
```
    print()
    print("Accuracy Score: ",accuracy_score(y_train,pred_train))
    print("Balanced Accuracy Score: ",balanced_accuracy_score(y_train,pred_train))
    print("Cohen Kappa Score: ",cohen_kappa_score(y_train,pred_train))
    print("Roc Score: ",roc_auc_score(y_train,pred_train_proba[:,1]))
    print("Confusion Matrix:\n",confusion_matrix(y_train,pred_train))
    print("Classification Report:\n",classification_report(y_train,pred_train))
    plot_roc_curve(model,X_train,y_train)
    print()


    print("\t Testing Results\t")
    print()
    print("Accuracy Score: ",accuracy_score(y_test,pred_test))
    print("Balanced Accuracy Score: ",balanced_accuracy_score(y_test,pred_test))
    print("Cohen Kappa Score: ",cohen_kappa_score(y_test,pred_test))
    print("Roc Score: ",roc_auc_score(y_test,pred_test_proba[:,1]))
    print("Confusion Matrix: \n",confusion_matrix(y_test,pred_test))
    print("Classification Report\n",classification_report(y_test,pred_test))
    plot_roc_curve(model,X_test,y_test)
```

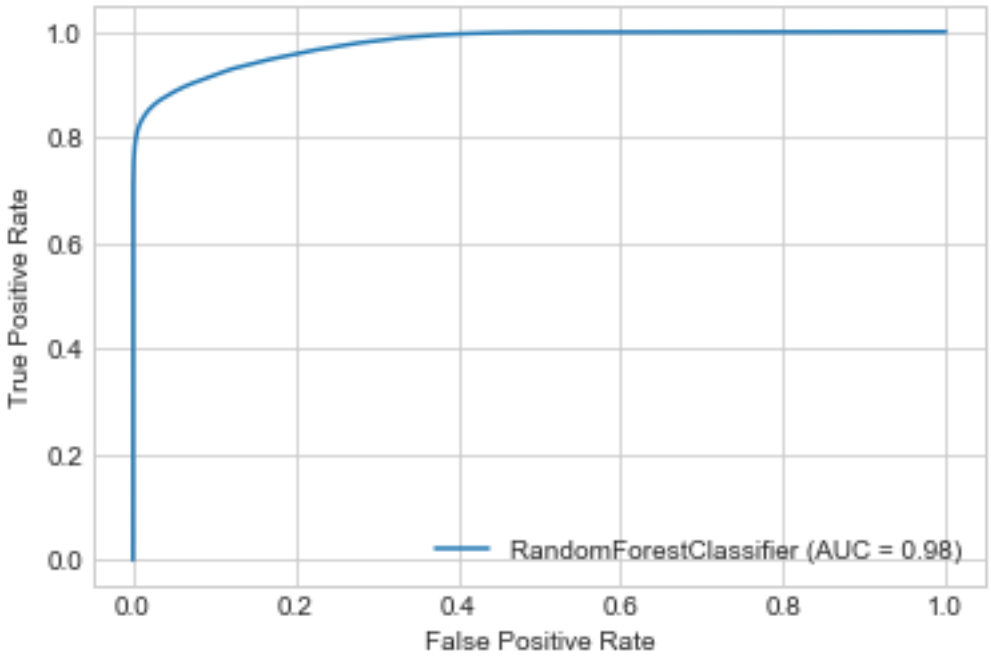
- Standardizing or normalizing the data did not provide better results
- Proceeding without Standardization

```
In [15]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=0)
```

```
In [49]: # Creating Baseline Random forest Engine
         baseline_random_forest = RandomForestClassifier(n_estimators=100)
         baseline_random_forest.fit(X_train,y_train)
         Model_Performance(baseline_random_forest,"Random Forest")

                 Random Forest Performance Outline

         Training Results

Accuracy Score:  0.9999166947910748
Balanced Accuracy Score:  0.9999166963240207
Cohen Kappa Score:  0.9998333895824658
Roc Score:  0.9999999464164917
Confusion Matrix:
 [[234062      9]
 [    30 234057]]
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00    234071
           1       1.00      1.00      1.00    234087
```

```
    accuracy                           1.00    468158
   macro avg       1.00      1.00      1.00    468158
weighted avg       1.00      1.00      1.00    468158


         Testing Results

Accuracy Score:  0.9156100478468899
Balanced Accuracy Score:  0.9156086888184838
Cohen Kappa Score:  0.8312196360470046
Roc Score:  0.977506306082538
Confusion Matrix:
 [[93571  6757]
 [10175 90137]]
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.93      0.92    100328
           1       0.93      0.90      0.91    100312

    accuracy                           0.92    200640
   macro avg       0.92      0.92      0.92    200640
weighted avg       0.92      0.92      0.92    200640
```





In [57]:
```
# Creating Baseline Catboost Classifier (Based on previous Experience)
baseline_catboost = CatBoostClassifier()
baseline_catboost.fit(X_train,y_train)
```

```
Learning rate set to 0.142296
0:        learn: 0.4827695        total: 273ms        remaining: 4m 33s
1:        learn: 0.3960941        total: 348ms        remaining: 2m 53s
2:        learn: 0.3477186        total: 413ms        remaining: 2m 17s
3:        learn: 0.3276000        total: 479ms        remaining: 1m 59s
4:        learn: 0.3195322        total: 552ms        remaining: 1m 49s
5:        learn: 0.3112146        total: 621ms        remaining: 1m 42s
6:        learn: 0.2957448        total: 695ms        remaining: 1m 38s
7:        learn: 0.2925019        total: 767ms        remaining: 1m 35s
8:        learn: 0.2898431        total: 831ms        remaining: 1m 31s
9:        learn: 0.2870312        total: 896ms        remaining: 1m 28s
10:       learn: 0.2832157        total: 976ms        remaining: 1m 27s
11:       learn: 0.2776796        total: 1.04s        remaining: 1m 26s
12:       learn: 0.2763014        total: 1.12s        remaining: 1m 24s
13:       learn: 0.2717114        total: 1.19s        remaining: 1m 23s
14:       learn: 0.2675487        total: 1.25s        remaining: 1m 21s
15:       learn: 0.2611152        total: 1.31s        remaining: 1m 20s
16:       learn: 0.2569796        total: 1.39s        remaining: 1m 20s
17:       learn: 0.2543589        total: 1.45s        remaining: 1m 19s
18:       learn: 0.2518683        total: 1.52s        remaining: 1m 18s
19:       learn: 0.2498062        total: 1.59s        remaining: 1m 17s
```

```
20:     learn: 0.2486897        total: 1.64s        remaining: 1m 16s
21:     learn: 0.2479099        total: 1.7s        remaining: 1m 15s
22:     learn: 0.2440608        total: 1.78s        remaining: 1m 15s
23:     learn: 0.2431318        total: 1.85s        remaining: 1m 15s
24:     learn: 0.2418041        total: 1.91s        remaining: 1m 14s
25:     learn: 0.2409767        total: 1.97s        remaining: 1m 13s
26:     learn: 0.2382270        total: 2.04s        remaining: 1m 13s
27:     learn: 0.2367039        total: 2.11s        remaining: 1m 13s
28:     learn: 0.2361702        total: 2.17s        remaining: 1m 12s
29:     learn: 0.2350157        total: 2.23s        remaining: 1m 12s
30:     learn: 0.2339778        total: 2.29s        remaining: 1m 11s
31:     learn: 0.2322143        total: 2.36s        remaining: 1m 11s
32:     learn: 0.2307163        total: 2.43s        remaining: 1m 11s
33:     learn: 0.2285193        total: 2.5s        remaining: 1m 11s
34:     learn: 0.2268118        total: 2.57s        remaining: 1m 10s
35:     learn: 0.2259196        total: 2.63s        remaining: 1m 10s
36:     learn: 0.2257011        total: 2.7s        remaining: 1m 10s
37:     learn: 0.2252429        total: 2.76s        remaining: 1m 9s
38:     learn: 0.2240899        total: 2.82s        remaining: 1m 9s
39:     learn: 0.2235541        total: 2.89s        remaining: 1m 9s
40:     learn: 0.2232160        total: 2.95s        remaining: 1m 9s
41:     learn: 0.2228721        total: 3.01s        remaining: 1m 8s
42:     learn: 0.2225464        total: 3.06s        remaining: 1m 8s
43:     learn: 0.2216330        total: 3.14s        remaining: 1m 8s
44:     learn: 0.2205560        total: 3.21s        remaining: 1m 8s
45:     learn: 0.2195217        total: 3.27s        remaining: 1m 7s
46:     learn: 0.2188822        total: 3.33s        remaining: 1m 7s
47:     learn: 0.2179138        total: 3.41s        remaining: 1m 7s
48:     learn: 0.2175785        total: 3.5s        remaining: 1m 7s
49:     learn: 0.2167971        total: 3.58s        remaining: 1m 7s
50:     learn: 0.2160070        total: 3.64s        remaining: 1m 7s
51:     learn: 0.2144169        total: 3.72s        remaining: 1m 7s
52:     learn: 0.2134352        total: 3.83s        remaining: 1m 8s
53:     learn: 0.2124690        total: 3.93s        remaining: 1m 8s
54:     learn: 0.2119436        total: 4.04s        remaining: 1m 9s
55:     learn: 0.2111846        total: 4.15s        remaining: 1m 9s
56:     learn: 0.2097505        total: 4.25s        remaining: 1m 10s
57:     learn: 0.2088536        total: 4.34s        remaining: 1m 10s
58:     learn: 0.2086160        total: 4.42s        remaining: 1m 10s
59:     learn: 0.2079391        total: 4.52s        remaining: 1m 10s
60:     learn: 0.2077142        total: 4.58s        remaining: 1m 10s
61:     learn: 0.2073102        total: 4.67s        remaining: 1m 10s
62:     learn: 0.2069401        total: 4.75s        remaining: 1m 10s
63:     learn: 0.2057845        total: 4.81s        remaining: 1m 10s
64:     learn: 0.2054172        total: 4.88s        remaining: 1m 10s
65:     learn: 0.2038755        total: 4.95s        remaining: 1m 9s
66:     learn: 0.2033773        total: 5.04s        remaining: 1m 10s
67:     learn: 0.2017947        total: 5.12s        remaining: 1m 10s
```

```
68:     learn: 0.2003845    total: 5.2s     remaining: 1m 10s
69:     learn: 0.1999200    total: 5.27s    remaining: 1m 10s
70:     learn: 0.1991886    total: 5.34s    remaining: 1m 9s
71:     learn: 0.1979457    total: 5.42s    remaining: 1m 9s
72:     learn: 0.1970520    total: 5.49s    remaining: 1m 9s
73:     learn: 0.1967020    total: 5.56s    remaining: 1m 9s
74:     learn: 0.1963212    total: 5.62s    remaining: 1m 9s
75:     learn: 0.1951749    total: 5.7s     remaining: 1m 9s
76:     learn: 0.1945314    total: 5.78s    remaining: 1m 9s
77:     learn: 0.1944119    total: 5.85s    remaining: 1m 9s
78:     learn: 0.1942449    total: 5.92s    remaining: 1m 8s
79:     learn: 0.1938264    total: 6s       remaining: 1m 9s
80:     learn: 0.1933242    total: 6.09s    remaining: 1m 9s
81:     learn: 0.1932490    total: 6.15s    remaining: 1m 8s
82:     learn: 0.1929290    total: 6.23s    remaining: 1m 8s
83:     learn: 0.1927395    total: 6.32s    remaining: 1m 8s
84:     learn: 0.1926635    total: 6.41s    remaining: 1m 8s
85:     learn: 0.1922030    total: 6.5s     remaining: 1m 9s
86:     learn: 0.1921134    total: 6.58s    remaining: 1m 9s
87:     learn: 0.1914389    total: 6.67s    remaining: 1m 9s
88:     learn: 0.1911441    total: 6.75s    remaining: 1m 9s
89:     learn: 0.1909674    total: 6.82s    remaining: 1m 8s
90:     learn: 0.1905390    total: 6.89s    remaining: 1m 8s
91:     learn: 0.1901359    total: 6.94s    remaining: 1m 8s
92:     learn: 0.1897817    total: 7.02s    remaining: 1m 8s
93:     learn: 0.1897195    total: 7.09s    remaining: 1m 8s
94:     learn: 0.1892581    total: 7.17s    remaining: 1m 8s
95:     learn: 0.1890609    total: 7.24s    remaining: 1m 8s
96:     learn: 0.1888900    total: 7.31s    remaining: 1m 8s
97:     learn: 0.1887393    total: 7.37s    remaining: 1m 7s
98:     learn: 0.1885731    total: 7.46s    remaining: 1m 7s
99:     learn: 0.1881856    total: 7.55s    remaining: 1m 7s
100:    learn: 0.1877934    total: 7.61s    remaining: 1m 7s
101:    learn: 0.1873638    total: 7.68s    remaining: 1m 7s
102:    learn: 0.1870651    total: 7.75s    remaining: 1m 7s
103:    learn: 0.1868119    total: 7.81s    remaining: 1m 7s
104:    learn: 0.1860680    total: 7.9s     remaining: 1m 7s
105:    learn: 0.1856807    total: 7.97s    remaining: 1m 7s
106:    learn: 0.1847564    total: 8.04s    remaining: 1m 7s
107:    learn: 0.1841205    total: 8.12s    remaining: 1m 7s
108:    learn: 0.1835795    total: 8.2s     remaining: 1m 7s
109:    learn: 0.1829759    total: 8.29s    remaining: 1m 7s
110:    learn: 0.1822266    total: 8.36s    remaining: 1m 6s
111:    learn: 0.1817173    total: 8.43s    remaining: 1m 6s
112:    learn: 0.1811733    total: 8.51s    remaining: 1m 6s
113:    learn: 0.1809977    total: 8.59s    remaining: 1m 6s
114:    learn: 0.1807972    total: 8.65s    remaining: 1m 6s
115:    learn: 0.1806725    total: 8.71s    remaining: 1m 6s
```

```
116:        learn: 0.1801931        total: 8.77s        remaining: 1m 6s
117:        learn: 0.1796655        total: 8.86s        remaining: 1m 6s
118:        learn: 0.1794828        total: 8.93s        remaining: 1m 6s
119:        learn: 0.1789826        total: 9s         remaining: 1m 5s
120:        learn: 0.1787845        total: 9.06s        remaining: 1m 5s
121:        learn: 0.1785497        total: 9.14s        remaining: 1m 5s
122:        learn: 0.1784588        total: 9.21s        remaining: 1m 5s
123:        learn: 0.1782408        total: 9.3s         remaining: 1m 5s
124:        learn: 0.1779094        total: 9.39s        remaining: 1m 5s
125:        learn: 0.1775000        total: 9.47s        remaining: 1m 5s
126:        learn: 0.1771318        total: 9.54s        remaining: 1m 5s
127:        learn: 0.1770402        total: 9.61s        remaining: 1m 5s
128:        learn: 0.1769605        total: 9.67s        remaining: 1m 5s
129:        learn: 0.1766468        total: 9.74s        remaining: 1m 5s
130:        learn: 0.1764522        total: 9.83s        remaining: 1m 5s
131:        learn: 0.1760739        total: 9.93s        remaining: 1m 5s
132:        learn: 0.1758491        total: 10s          remaining: 1m 5s
133:        learn: 0.1755882        total: 10.1s        remaining: 1m 5s
134:        learn: 0.1754646        total: 10.2s        remaining: 1m 5s
135:        learn: 0.1753270        total: 10.3s        remaining: 1m 5s
136:        learn: 0.1750133        total: 10.4s        remaining: 1m 5s
137:        learn: 0.1748172        total: 10.5s        remaining: 1m 5s
138:        learn: 0.1747579        total: 10.6s        remaining: 1m 5s
139:        learn: 0.1746546        total: 10.7s        remaining: 1m 5s
140:        learn: 0.1740435        total: 10.8s        remaining: 1m 5s
141:        learn: 0.1738424        total: 10.9s        remaining: 1m 5s
142:        learn: 0.1736564        total: 10.9s        remaining: 1m 5s
143:        learn: 0.1735647        total: 11s          remaining: 1m 5s
144:        learn: 0.1733860        total: 11.1s        remaining: 1m 5s
145:        learn: 0.1732356        total: 11.2s        remaining: 1m 5s
146:        learn: 0.1730022        total: 11.3s        remaining: 1m 5s
147:        learn: 0.1729019        total: 11.4s        remaining: 1m 5s
148:        learn: 0.1728344        total: 11.4s        remaining: 1m 5s
149:        learn: 0.1727153        total: 11.5s        remaining: 1m 5s
150:        learn: 0.1726798        total: 11.6s        remaining: 1m 5s
151:        learn: 0.1725136        total: 11.7s        remaining: 1m 5s
152:        learn: 0.1721646        total: 11.8s        remaining: 1m 5s
153:        learn: 0.1721289        total: 11.9s        remaining: 1m 5s
154:        learn: 0.1717666        total: 12s          remaining: 1m 5s
155:        learn: 0.1716670        total: 12.1s        remaining: 1m 5s
156:        learn: 0.1713786        total: 12.1s        remaining: 1m 5s
157:        learn: 0.1713332        total: 12.2s        remaining: 1m 5s
158:        learn: 0.1712250        total: 12.3s        remaining: 1m 5s
159:        learn: 0.1710993        total: 12.4s        remaining: 1m 4s
160:        learn: 0.1710474        total: 12.4s        remaining: 1m 4s
161:        learn: 0.1709487        total: 12.5s        remaining: 1m 4s
162:        learn: 0.1707465        total: 12.6s        remaining: 1m 4s
163:        learn: 0.1706921        total: 12.7s        remaining: 1m 4s
```

```
164:    learn: 0.1705604    total: 12.7s    remaining: 1m 4s
165:    learn: 0.1705051    total: 12.8s    remaining: 1m 4s
166:    learn: 0.1704675    total: 12.9s    remaining: 1m 4s
167:    learn: 0.1704077    total: 13s    remaining: 1m 4s
168:    learn: 0.1703769    total: 13.1s    remaining: 1m 4s
169:    learn: 0.1703213    total: 13.2s    remaining: 1m 4s
170:    learn: 0.1701658    total: 13.3s    remaining: 1m 4s
171:    learn: 0.1700015    total: 13.3s    remaining: 1m 4s
172:    learn: 0.1696796    total: 13.4s    remaining: 1m 4s
173:    learn: 0.1694866    total: 13.5s    remaining: 1m 3s
174:    learn: 0.1694613    total: 13.6s    remaining: 1m 3s
175:    learn: 0.1694150    total: 13.6s    remaining: 1m 3s
176:    learn: 0.1691478    total: 13.7s    remaining: 1m 3s
177:    learn: 0.1688773    total: 13.8s    remaining: 1m 3s
178:    learn: 0.1685706    total: 13.8s    remaining: 1m 3s
179:    learn: 0.1684061    total: 13.9s    remaining: 1m 3s
180:    learn: 0.1683270    total: 14s    remaining: 1m 3s
181:    learn: 0.1682360    total: 14.1s    remaining: 1m 3s
182:    learn: 0.1681788    total: 14.2s    remaining: 1m 3s
183:    learn: 0.1680954    total: 14.3s    remaining: 1m 3s
184:    learn: 0.1680628    total: 14.3s    remaining: 1m 3s
185:    learn: 0.1680210    total: 14.4s    remaining: 1m 3s
186:    learn: 0.1679423    total: 14.5s    remaining: 1m 3s
187:    learn: 0.1679048    total: 14.6s    remaining: 1m 3s
188:    learn: 0.1678282    total: 14.7s    remaining: 1m 2s
189:    learn: 0.1678004    total: 14.7s    remaining: 1m 2s
190:    learn: 0.1677637    total: 14.8s    remaining: 1m 2s
191:    learn: 0.1677360    total: 14.9s    remaining: 1m 2s
192:    learn: 0.1677126    total: 14.9s    remaining: 1m 2s
193:    learn: 0.1676113    total: 15s    remaining: 1m 2s
194:    learn: 0.1675874    total: 15s    remaining: 1m 2s
195:    learn: 0.1675371    total: 15.1s    remaining: 1m 1s
196:    learn: 0.1674330    total: 15.2s    remaining: 1m 1s
197:    learn: 0.1673875    total: 15.2s    remaining: 1m 1s
198:    learn: 0.1673075    total: 15.3s    remaining: 1m 1s
199:    learn: 0.1672873    total: 15.4s    remaining: 1m 1s
200:    learn: 0.1671282    total: 15.4s    remaining: 1m 1s
201:    learn: 0.1670411    total: 15.5s    remaining: 1m 1s
202:    learn: 0.1670153    total: 15.6s    remaining: 1m 1s
203:    learn: 0.1669743    total: 15.6s    remaining: 1m 1s
204:    learn: 0.1668803    total: 15.7s    remaining: 1m
205:    learn: 0.1668570    total: 15.8s    remaining: 1m
206:    learn: 0.1667904    total: 15.9s    remaining: 1m
207:    learn: 0.1667714    total: 16s    remaining: 1m
208:    learn: 0.1667484    total: 16s    remaining: 1m
209:    learn: 0.1664341    total: 16.1s    remaining: 1m
210:    learn: 0.1663941    total: 16.2s    remaining: 1m
211:    learn: 0.1663681    total: 16.3s    remaining: 1m
```

```
212:        learn: 0.1663152        total: 16.4s        remaining: 1m
213:        learn: 0.1662922        total: 16.4s        remaining: 1m
214:        learn: 0.1662696        total: 16.5s        remaining: 1m
215:        learn: 0.1662468        total: 16.6s        remaining: 1m
216:        learn: 0.1661362        total: 16.7s        remaining: 1m
217:        learn: 0.1661126        total: 16.7s        remaining: 60s
218:        learn: 0.1660901        total: 16.8s        remaining: 59.9s
219:        learn: 0.1658315        total: 16.9s        remaining: 59.8s
220:        learn: 0.1657441        total: 17s        remaining: 59.8s
221:        learn: 0.1657141        total: 17s        remaining: 59.7s
222:        learn: 0.1657004        total: 17.1s        remaining: 59.6s
223:        learn: 0.1656821        total: 17.2s        remaining: 59.5s
224:        learn: 0.1656645        total: 17.3s        remaining: 59.5s
225:        learn: 0.1656325        total: 17.3s        remaining: 59.4s
226:        learn: 0.1656130        total: 17.4s        remaining: 59.2s
227:        learn: 0.1655451        total: 17.5s        remaining: 59.1s
228:        learn: 0.1655016        total: 17.5s        remaining: 59.1s
229:        learn: 0.1653531        total: 17.6s        remaining: 59s
230:        learn: 0.1652874        total: 17.7s        remaining: 58.9s
231:        learn: 0.1650995        total: 17.8s        remaining: 58.9s
232:        learn: 0.1650733        total: 17.9s        remaining: 58.8s
233:        learn: 0.1650202        total: 18s        remaining: 58.8s
234:        learn: 0.1648768        total: 18s        remaining: 58.7s
235:        learn: 0.1647725        total: 18.1s        remaining: 58.6s
236:        learn: 0.1647277        total: 18.2s        remaining: 58.5s
237:        learn: 0.1646506        total: 18.2s        remaining: 58.3s
238:        learn: 0.1646299        total: 18.3s        remaining: 58.2s
239:        learn: 0.1644372        total: 18.4s        remaining: 58.2s
240:        learn: 0.1644142        total: 18.4s        remaining: 58.1s
241:        learn: 0.1643974        total: 18.5s        remaining: 58s
242:        learn: 0.1643555        total: 18.6s        remaining: 57.9s
243:        learn: 0.1643384        total: 18.7s        remaining: 57.8s
244:        learn: 0.1643094        total: 18.7s        remaining: 57.7s
245:        learn: 0.1642896        total: 18.8s        remaining: 57.7s
246:        learn: 0.1642597        total: 18.9s        remaining: 57.5s
247:        learn: 0.1642385        total: 18.9s        remaining: 57.4s
248:        learn: 0.1641993        total: 19s        remaining: 57.4s
249:        learn: 0.1640431        total: 19.1s        remaining: 57.3s
250:        learn: 0.1640201        total: 19.2s        remaining: 57.3s
251:        learn: 0.1640008        total: 19.3s        remaining: 57.2s
252:        learn: 0.1639805        total: 19.3s        remaining: 57.1s
253:        learn: 0.1639319        total: 19.4s        remaining: 57s
254:        learn: 0.1637783        total: 19.5s        remaining: 56.9s
255:        learn: 0.1637341        total: 19.6s        remaining: 56.8s
256:        learn: 0.1636830        total: 19.6s        remaining: 56.7s
257:        learn: 0.1636510        total: 19.7s        remaining: 56.6s
258:        learn: 0.1635435        total: 19.8s        remaining: 56.6s
259:        learn: 0.1635245        total: 19.8s        remaining: 56.5s
```

```
260:        learn: 0.1635096        total: 19.9s        remaining: 56.4s
261:        learn: 0.1634959        total: 20s        remaining: 56.3s
262:        learn: 0.1634431        total: 20s        remaining: 56.2s
263:        learn: 0.1633914        total: 20.1s        remaining: 56.1s
264:        learn: 0.1633372        total: 20.2s        remaining: 56s
265:        learn: 0.1632906        total: 20.3s        remaining: 55.9s
266:        learn: 0.1632698        total: 20.3s        remaining: 55.8s
267:        learn: 0.1632445        total: 20.4s        remaining: 55.7s
268:        learn: 0.1632253        total: 20.5s        remaining: 55.6s
269:        learn: 0.1631023        total: 20.5s        remaining: 55.5s
270:        learn: 0.1630423        total: 20.6s        remaining: 55.4s
271:        learn: 0.1630176        total: 20.7s        remaining: 55.3s
272:        learn: 0.1630003        total: 20.8s        remaining: 55.3s
273:        learn: 0.1629372        total: 20.8s        remaining: 55.2s
274:        learn: 0.1629093        total: 20.9s        remaining: 55.1s
275:        learn: 0.1628781        total: 21s        remaining: 55s
276:        learn: 0.1628593        total: 21.1s        remaining: 55s
277:        learn: 0.1628444        total: 21.1s        remaining: 54.9s
278:        learn: 0.1627833        total: 21.2s        remaining: 54.8s
279:        learn: 0.1627327        total: 21.3s        remaining: 54.7s
280:        learn: 0.1624714        total: 21.3s        remaining: 54.6s
281:        learn: 0.1623149        total: 21.4s        remaining: 54.5s
282:        learn: 0.1622784        total: 21.5s        remaining: 54.4s
283:        learn: 0.1621955        total: 21.6s        remaining: 54.4s
284:        learn: 0.1620622        total: 21.6s        remaining: 54.3s
285:        learn: 0.1620414        total: 21.7s        remaining: 54.2s
286:        learn: 0.1619760        total: 21.8s        remaining: 54.1s
287:        learn: 0.1619505        total: 21.9s        remaining: 54s
288:        learn: 0.1619053        total: 21.9s        remaining: 54s
289:        learn: 0.1618913        total: 22s        remaining: 53.9s
290:        learn: 0.1618771        total: 22.1s        remaining: 53.9s
291:        learn: 0.1618112        total: 22.2s        remaining: 53.8s
292:        learn: 0.1617375        total: 22.3s        remaining: 53.7s
293:        learn: 0.1616928        total: 22.3s        remaining: 53.6s
294:        learn: 0.1616755        total: 22.4s        remaining: 53.5s
295:        learn: 0.1616575        total: 22.5s        remaining: 53.4s
296:        learn: 0.1616404        total: 22.6s        remaining: 53.4s
297:        learn: 0.1615969        total: 22.6s        remaining: 53.3s
298:        learn: 0.1615786        total: 22.7s        remaining: 53.2s
299:        learn: 0.1615697        total: 22.8s        remaining: 53.1s
300:        learn: 0.1615471        total: 22.8s        remaining: 53s
301:        learn: 0.1615268        total: 22.9s        remaining: 52.9s
302:        learn: 0.1615085        total: 23s        remaining: 52.9s
303:        learn: 0.1613892        total: 23.1s        remaining: 52.8s
304:        learn: 0.1613576        total: 23.1s        remaining: 52.7s
305:        learn: 0.1612422        total: 23.2s        remaining: 52.7s
306:        learn: 0.1611822        total: 23.3s        remaining: 52.6s
307:        learn: 0.1611627        total: 23.4s        remaining: 52.6s
```

```
308:    learn: 0.1611397    total: 23.5s    remaining: 52.5s
309:    learn: 0.1611226    total: 23.5s    remaining: 52.4s
310:    learn: 0.1610705    total: 23.6s    remaining: 52.3s
311:    learn: 0.1610536    total: 23.7s    remaining: 52.2s
312:    learn: 0.1610273    total: 23.8s    remaining: 52.1s
313:    learn: 0.1610111    total: 23.9s    remaining: 52.1s
314:    learn: 0.1609803    total: 23.9s    remaining: 52s
315:    learn: 0.1609230    total: 24s    remaining: 52s
316:    learn: 0.1609025    total: 24.1s    remaining: 51.9s
317:    learn: 0.1608855    total: 24.2s    remaining: 51.8s
318:    learn: 0.1608654    total: 24.2s    remaining: 51.7s
319:    learn: 0.1608169    total: 24.3s    remaining: 51.6s
320:    learn: 0.1608010    total: 24.4s    remaining: 51.5s
321:    learn: 0.1607818    total: 24.4s    remaining: 51.5s
322:    learn: 0.1607174    total: 24.6s    remaining: 51.5s
323:    learn: 0.1606979    total: 24.6s    remaining: 51.4s
324:    learn: 0.1606876    total: 24.7s    remaining: 51.3s
325:    learn: 0.1606578    total: 24.8s    remaining: 51.2s
326:    learn: 0.1606396    total: 24.8s    remaining: 51.1s
327:    learn: 0.1606151    total: 24.9s    remaining: 51s
328:    learn: 0.1605937    total: 25s    remaining: 50.9s
329:    learn: 0.1605795    total: 25s    remaining: 50.8s
330:    learn: 0.1605605    total: 25.1s    remaining: 50.7s
331:    learn: 0.1604871    total: 25.2s    remaining: 50.6s
332:    learn: 0.1604706    total: 25.2s    remaining: 50.5s
333:    learn: 0.1604543    total: 25.3s    remaining: 50.4s
334:    learn: 0.1604399    total: 25.3s    remaining: 50.3s
335:    learn: 0.1603698    total: 25.4s    remaining: 50.2s
336:    learn: 0.1603556    total: 25.5s    remaining: 50.1s
337:    learn: 0.1603000    total: 25.5s    remaining: 50s
338:    learn: 0.1602803    total: 25.6s    remaining: 49.9s
339:    learn: 0.1602061    total: 25.6s    remaining: 49.8s
340:    learn: 0.1601843    total: 25.7s    remaining: 49.7s
341:    learn: 0.1601708    total: 25.8s    remaining: 49.6s
342:    learn: 0.1601593    total: 25.8s    remaining: 49.5s
343:    learn: 0.1601461    total: 25.9s    remaining: 49.4s
344:    learn: 0.1601297    total: 26s    remaining: 49.3s
345:    learn: 0.1601141    total: 26s    remaining: 49.2s
346:    learn: 0.1600960    total: 26.1s    remaining: 49.1s
347:    learn: 0.1600800    total: 26.2s    remaining: 49s
348:    learn: 0.1600668    total: 26.2s    remaining: 48.9s
349:    learn: 0.1600534    total: 26.3s    remaining: 48.8s
350:    learn: 0.1600412    total: 26.4s    remaining: 48.7s
351:    learn: 0.1599013    total: 26.4s    remaining: 48.7s
352:    learn: 0.1598824    total: 26.5s    remaining: 48.6s
353:    learn: 0.1598257    total: 26.6s    remaining: 48.5s
354:    learn: 0.1598016    total: 26.6s    remaining: 48.4s
355:    learn: 0.1597835    total: 26.7s    remaining: 48.3s
```

```
356:    learn: 0.1597707    total: 26.7s    remaining: 48.2s
357:    learn: 0.1597607    total: 26.8s    remaining: 48.1s
358:    learn: 0.1596866    total: 26.9s    remaining: 48s
359:    learn: 0.1596656    total: 26.9s    remaining: 47.9s
360:    learn: 0.1596435    total: 27s      remaining: 47.8s
361:    learn: 0.1596297    total: 27.1s    remaining: 47.7s
362:    learn: 0.1596127    total: 27.1s    remaining: 47.6s
363:    learn: 0.1595982    total: 27.2s    remaining: 47.5s
364:    learn: 0.1595831    total: 27.3s    remaining: 47.4s
365:    learn: 0.1595650    total: 27.3s    remaining: 47.3s
366:    learn: 0.1595441    total: 27.4s    remaining: 47.2s
367:    learn: 0.1595309    total: 27.4s    remaining: 47.1s
368:    learn: 0.1595048    total: 27.5s    remaining: 47s
369:    learn: 0.1594456    total: 27.6s    remaining: 46.9s
370:    learn: 0.1594273    total: 27.6s    remaining: 46.9s
371:    learn: 0.1594120    total: 27.7s    remaining: 46.8s
372:    learn: 0.1593957    total: 27.8s    remaining: 46.7s
373:    learn: 0.1593780    total: 27.8s    remaining: 46.6s
374:    learn: 0.1593627    total: 27.9s    remaining: 46.5s
375:    learn: 0.1593436    total: 28s      remaining: 46.4s
376:    learn: 0.1593206    total: 28s      remaining: 46.3s
377:    learn: 0.1593069    total: 28.1s    remaining: 46.2s
378:    learn: 0.1592570    total: 28.1s    remaining: 46.1s
379:    learn: 0.1591960    total: 28.2s    remaining: 46s
380:    learn: 0.1591313    total: 28.3s    remaining: 45.9s
381:    learn: 0.1591164    total: 28.3s    remaining: 45.8s
382:    learn: 0.1591021    total: 28.4s    remaining: 45.8s
383:    learn: 0.1590856    total: 28.5s    remaining: 45.7s
384:    learn: 0.1590704    total: 28.5s    remaining: 45.6s
385:    learn: 0.1590601    total: 28.6s    remaining: 45.5s
386:    learn: 0.1590483    total: 28.7s    remaining: 45.4s
387:    learn: 0.1590129    total: 28.7s    remaining: 45.3s
388:    learn: 0.1590038    total: 28.8s    remaining: 45.2s
389:    learn: 0.1589829    total: 28.9s    remaining: 45.1s
390:    learn: 0.1589288    total: 28.9s    remaining: 45.1s
391:    learn: 0.1589117    total: 29s      remaining: 45s
392:    learn: 0.1588885    total: 29.1s    remaining: 44.9s
393:    learn: 0.1588763    total: 29.1s    remaining: 44.8s
394:    learn: 0.1588555    total: 29.2s    remaining: 44.7s
395:    learn: 0.1588000    total: 29.3s    remaining: 44.6s
396:    learn: 0.1587844    total: 29.4s    remaining: 44.6s
397:    learn: 0.1587698    total: 29.5s    remaining: 44.6s
398:    learn: 0.1587601    total: 29.6s    remaining: 44.5s
399:    learn: 0.1586766    total: 29.6s    remaining: 44.5s
400:    learn: 0.1586609    total: 29.7s    remaining: 44.4s
401:    learn: 0.1586441    total: 29.8s    remaining: 44.4s
402:    learn: 0.1585904    total: 29.9s    remaining: 44.3s
403:    learn: 0.1585389    total: 30s      remaining: 44.2s
```

```
404:    learn: 0.1584998    total: 30.1s    remaining: 44.2s
405:    learn: 0.1584835    total: 30.2s    remaining: 44.1s
406:    learn: 0.1584494    total: 30.3s    remaining: 44.1s
407:    learn: 0.1584379    total: 30.4s    remaining: 44.1s
408:    learn: 0.1584235    total: 30.5s    remaining: 44s
409:    learn: 0.1584041    total: 30.5s    remaining: 43.9s
410:    learn: 0.1583835    total: 30.6s    remaining: 43.9s
411:    learn: 0.1582989    total: 30.7s    remaining: 43.8s
412:    learn: 0.1582828    total: 30.8s    remaining: 43.8s
413:    learn: 0.1582698    total: 30.9s    remaining: 43.7s
414:    learn: 0.1582540    total: 30.9s    remaining: 43.6s
415:    learn: 0.1582430    total: 31s      remaining: 43.5s
416:    learn: 0.1582240    total: 31.1s    remaining: 43.4s
417:    learn: 0.1582072    total: 31.1s    remaining: 43.3s
418:    learn: 0.1581918    total: 31.2s    remaining: 43.2s
419:    learn: 0.1581802    total: 31.3s    remaining: 43.2s
420:    learn: 0.1581684    total: 31.3s    remaining: 43.1s
421:    learn: 0.1581538    total: 31.4s    remaining: 43s
422:    learn: 0.1581407    total: 31.5s    remaining: 42.9s
423:    learn: 0.1581245    total: 31.5s    remaining: 42.8s
424:    learn: 0.1581094    total: 31.6s    remaining: 42.7s
425:    learn: 0.1580895    total: 31.7s    remaining: 42.7s
426:    learn: 0.1580804    total: 31.7s    remaining: 42.6s
427:    learn: 0.1580683    total: 31.8s    remaining: 42.6s
428:    learn: 0.1580538    total: 31.9s    remaining: 42.5s
429:    learn: 0.1580421    total: 32s      remaining: 42.4s
430:    learn: 0.1580283    total: 32s      remaining: 42.3s
431:    learn: 0.1580195    total: 32.1s    remaining: 42.2s
432:    learn: 0.1579894    total: 32.2s    remaining: 42.1s
433:    learn: 0.1578945    total: 32.2s    remaining: 42s
434:    learn: 0.1578856    total: 32.3s    remaining: 42s
435:    learn: 0.1578713    total: 32.4s    remaining: 41.9s
436:    learn: 0.1578540    total: 32.4s    remaining: 41.8s
437:    learn: 0.1578326    total: 32.5s    remaining: 41.7s
438:    learn: 0.1578161    total: 32.6s    remaining: 41.6s
439:    learn: 0.1578020    total: 32.6s    remaining: 41.5s
440:    learn: 0.1577840    total: 32.7s    remaining: 41.5s
441:    learn: 0.1577738    total: 32.8s    remaining: 41.4s
442:    learn: 0.1577599    total: 32.8s    remaining: 41.3s
443:    learn: 0.1577367    total: 32.9s    remaining: 41.2s
444:    learn: 0.1577157    total: 33s      remaining: 41.1s
445:    learn: 0.1577036    total: 33s      remaining: 41s
446:    learn: 0.1576957    total: 33.1s    remaining: 40.9s
447:    learn: 0.1576818    total: 33.2s    remaining: 40.9s
448:    learn: 0.1576551    total: 33.2s    remaining: 40.8s
449:    learn: 0.1576014    total: 33.3s    remaining: 40.7s
450:    learn: 0.1575844    total: 33.4s    remaining: 40.6s
451:    learn: 0.1575583    total: 33.4s    remaining: 40.5s
```

```
452:        learn: 0.1575416        total: 33.5s        remaining: 40.4s
453:        learn: 0.1575284        total: 33.5s        remaining: 40.3s
454:        learn: 0.1575156        total: 33.6s        remaining: 40.3s
455:        learn: 0.1575030        total: 33.7s        remaining: 40.2s
456:        learn: 0.1574907        total: 33.8s        remaining: 40.1s
457:        learn: 0.1574553        total: 33.8s        remaining: 40.1s
458:        learn: 0.1574407        total: 33.9s        remaining: 40s
459:        learn: 0.1574004        total: 34s        remaining: 39.9s
460:        learn: 0.1573862        total: 34.1s        remaining: 39.9s
461:        learn: 0.1573725        total: 34.2s        remaining: 39.8s
462:        learn: 0.1573437        total: 34.3s        remaining: 39.8s
463:        learn: 0.1573258        total: 34.3s        remaining: 39.7s
464:        learn: 0.1572597        total: 34.4s        remaining: 39.6s
465:        learn: 0.1572452        total: 34.5s        remaining: 39.6s
466:        learn: 0.1571863        total: 34.6s        remaining: 39.5s
467:        learn: 0.1571658        total: 34.7s        remaining: 39.4s
468:        learn: 0.1571537        total: 34.8s        remaining: 39.3s
469:        learn: 0.1571455        total: 34.8s        remaining: 39.3s
470:        learn: 0.1571309        total: 34.9s        remaining: 39.2s
471:        learn: 0.1570679        total: 35s        remaining: 39.1s
472:        learn: 0.1570517        total: 35s        remaining: 39s
473:        learn: 0.1570309        total: 35.1s        remaining: 38.9s
474:        learn: 0.1570160        total: 35.2s        remaining: 38.9s
475:        learn: 0.1570008        total: 35.2s        remaining: 38.8s
476:        learn: 0.1569884        total: 35.3s        remaining: 38.7s
477:        learn: 0.1569737        total: 35.4s        remaining: 38.6s
478:        learn: 0.1569585        total: 35.4s        remaining: 38.5s
479:        learn: 0.1569444        total: 35.5s        remaining: 38.5s
480:        learn: 0.1569290        total: 35.6s        remaining: 38.4s
481:        learn: 0.1569168        total: 35.6s        remaining: 38.3s
482:        learn: 0.1569078        total: 35.7s        remaining: 38.3s
483:        learn: 0.1568920        total: 35.8s        remaining: 38.2s
484:        learn: 0.1568801        total: 35.9s        remaining: 38.1s
485:        learn: 0.1568590        total: 36s        remaining: 38.1s
486:        learn: 0.1568514        total: 36.1s        remaining: 38s
487:        learn: 0.1568390        total: 36.2s        remaining: 38s
488:        learn: 0.1568253        total: 36.3s        remaining: 37.9s
489:        learn: 0.1568133        total: 36.4s        remaining: 37.9s
490:        learn: 0.1568021        total: 36.5s        remaining: 37.8s
491:        learn: 0.1567884        total: 36.6s        remaining: 37.8s
492:        learn: 0.1567760        total: 36.7s        remaining: 37.7s
493:        learn: 0.1567664        total: 36.8s        remaining: 37.7s
494:        learn: 0.1567549        total: 36.9s        remaining: 37.6s
495:        learn: 0.1567389        total: 37s        remaining: 37.6s
496:        learn: 0.1567248        total: 37s        remaining: 37.5s
497:        learn: 0.1567091        total: 37.1s        remaining: 37.4s
498:        learn: 0.1566999        total: 37.2s        remaining: 37.3s
499:        learn: 0.1566812        total: 37.2s        remaining: 37.2s
```

```
500:    learn: 0.1566680    total: 37.3s    remaining: 37.2s
501:    learn: 0.1566547    total: 37.4s    remaining: 37.1s
502:    learn: 0.1566469    total: 37.4s    remaining: 37s
503:    learn: 0.1566352    total: 37.5s    remaining: 36.9s
504:    learn: 0.1566057    total: 37.6s    remaining: 36.8s
505:    learn: 0.1565949    total: 37.6s    remaining: 36.7s
506:    learn: 0.1565822    total: 37.7s    remaining: 36.6s
507:    learn: 0.1565192    total: 37.7s    remaining: 36.6s
508:    learn: 0.1565055    total: 37.8s    remaining: 36.5s
509:    learn: 0.1564918    total: 37.9s    remaining: 36.4s
510:    learn: 0.1564574    total: 38s      remaining: 36.4s
511:    learn: 0.1564473    total: 38.1s    remaining: 36.3s
512:    learn: 0.1564368    total: 38.1s    remaining: 36.2s
513:    learn: 0.1563777    total: 38.2s    remaining: 36.1s
514:    learn: 0.1563011    total: 38.3s    remaining: 36s
515:    learn: 0.1562697    total: 38.3s    remaining: 36s
516:    learn: 0.1562408    total: 38.4s    remaining: 35.9s
517:    learn: 0.1562111    total: 38.5s    remaining: 35.8s
518:    learn: 0.1561927    total: 38.6s    remaining: 35.8s
519:    learn: 0.1561380    total: 38.6s    remaining: 35.7s
520:    learn: 0.1561229    total: 38.7s    remaining: 35.6s
521:    learn: 0.1561063    total: 38.8s    remaining: 35.5s
522:    learn: 0.1560398    total: 38.9s    remaining: 35.5s
523:    learn: 0.1560257    total: 39s      remaining: 35.4s
524:    learn: 0.1560103    total: 39s      remaining: 35.3s
525:    learn: 0.1559982    total: 39.1s    remaining: 35.2s
526:    learn: 0.1559839    total: 39.2s    remaining: 35.1s
527:    learn: 0.1559599    total: 39.2s    remaining: 35.1s
528:    learn: 0.1559449    total: 39.3s    remaining: 35s
529:    learn: 0.1559332    total: 39.4s    remaining: 34.9s
530:    learn: 0.1559264    total: 39.4s    remaining: 34.8s
531:    learn: 0.1559136    total: 39.5s    remaining: 34.7s
532:    learn: 0.1558983    total: 39.6s    remaining: 34.7s
533:    learn: 0.1558878    total: 39.7s    remaining: 34.6s
534:    learn: 0.1558726    total: 39.7s    remaining: 34.5s
535:    learn: 0.1558615    total: 39.8s    remaining: 34.4s
536:    learn: 0.1558493    total: 39.9s    remaining: 34.4s
537:    learn: 0.1558380    total: 39.9s    remaining: 34.3s
538:    learn: 0.1558277    total: 40s      remaining: 34.2s
539:    learn: 0.1558083    total: 40.1s    remaining: 34.1s
540:    learn: 0.1557956    total: 40.2s    remaining: 34.1s
541:    learn: 0.1557778    total: 40.2s    remaining: 34s
542:    learn: 0.1557446    total: 40.3s    remaining: 33.9s
543:    learn: 0.1557361    total: 40.3s    remaining: 33.8s
544:    learn: 0.1557260    total: 40.4s    remaining: 33.8s
545:    learn: 0.1557062    total: 40.5s    remaining: 33.7s
546:    learn: 0.1556908    total: 40.6s    remaining: 33.6s
547:    learn: 0.1556795    total: 40.6s    remaining: 33.5s
```

```
548:    learn: 0.1556631    total: 40.7s    remaining: 33.5s
549:    learn: 0.1556521    total: 40.8s    remaining: 33.4s
550:    learn: 0.1556175    total: 40.9s    remaining: 33.3s
551:    learn: 0.1555588    total: 41s    remaining: 33.2s
552:    learn: 0.1555451    total: 41s    remaining: 33.2s
553:    learn: 0.1555302    total: 41.1s    remaining: 33.1s
554:    learn: 0.1555137    total: 41.2s    remaining: 33s
555:    learn: 0.1554978    total: 41.3s    remaining: 33s
556:    learn: 0.1554808    total: 41.4s    remaining: 32.9s
557:    learn: 0.1553958    total: 41.4s    remaining: 32.8s
558:    learn: 0.1553836    total: 41.5s    remaining: 32.7s
559:    learn: 0.1553734    total: 41.6s    remaining: 32.7s
560:    learn: 0.1553586    total: 41.7s    remaining: 32.6s
561:    learn: 0.1553426    total: 41.7s    remaining: 32.5s
562:    learn: 0.1553304    total: 41.8s    remaining: 32.5s
563:    learn: 0.1553235    total: 41.9s    remaining: 32.4s
564:    learn: 0.1553123    total: 42s    remaining: 32.3s
565:    learn: 0.1552952    total: 42s    remaining: 32.2s
566:    learn: 0.1552829    total: 42.1s    remaining: 32.2s
567:    learn: 0.1552678    total: 42.2s    remaining: 32.1s
568:    learn: 0.1552547    total: 42.3s    remaining: 32s
569:    learn: 0.1552450    total: 42.4s    remaining: 32s
570:    learn: 0.1552303    total: 42.5s    remaining: 31.9s
571:    learn: 0.1552170    total: 42.5s    remaining: 31.8s
572:    learn: 0.1551840    total: 42.6s    remaining: 31.7s
573:    learn: 0.1551705    total: 42.7s    remaining: 31.7s
574:    learn: 0.1551560    total: 42.7s    remaining: 31.6s
575:    learn: 0.1551412    total: 42.8s    remaining: 31.5s
576:    learn: 0.1551286    total: 42.9s    remaining: 31.4s
577:    learn: 0.1551159    total: 43s    remaining: 31.4s
578:    learn: 0.1551070    total: 43s    remaining: 31.3s
579:    learn: 0.1550762    total: 43.1s    remaining: 31.2s
580:    learn: 0.1550646    total: 43.2s    remaining: 31.2s
581:    learn: 0.1550511    total: 43.3s    remaining: 31.1s
582:    learn: 0.1550384    total: 43.4s    remaining: 31s
583:    learn: 0.1550246    total: 43.5s    remaining: 31s
584:    learn: 0.1550118    total: 43.5s    remaining: 30.9s
585:    learn: 0.1550015    total: 43.6s    remaining: 30.8s
586:    learn: 0.1549893    total: 43.7s    remaining: 30.8s
587:    learn: 0.1549773    total: 43.8s    remaining: 30.7s
588:    learn: 0.1549640    total: 43.9s    remaining: 30.7s
589:    learn: 0.1549497    total: 44s    remaining: 30.6s
590:    learn: 0.1549365    total: 44.1s    remaining: 30.5s
591:    learn: 0.1549256    total: 44.2s    remaining: 30.4s
592:    learn: 0.1549114    total: 44.3s    remaining: 30.4s
593:    learn: 0.1549019    total: 44.4s    remaining: 30.3s
594:    learn: 0.1548904    total: 44.5s    remaining: 30.3s
595:    learn: 0.1548801    total: 44.5s    remaining: 30.2s
```

```
596:    learn: 0.1548669    total: 44.6s    remaining: 30.1s
597:    learn: 0.1548599    total: 44.7s    remaining: 30s
598:    learn: 0.1548518    total: 44.8s    remaining: 30s
599:    learn: 0.1548388    total: 44.8s    remaining: 29.9s
600:    learn: 0.1548250    total: 44.9s    remaining: 29.8s
601:    learn: 0.1548116    total: 45s      remaining: 29.7s
602:    learn: 0.1547989    total: 45s      remaining: 29.6s
603:    learn: 0.1547908    total: 45.1s    remaining: 29.6s
604:    learn: 0.1547769    total: 45.2s    remaining: 29.5s
605:    learn: 0.1547676    total: 45.3s    remaining: 29.4s
606:    learn: 0.1547540    total: 45.4s    remaining: 29.4s
607:    learn: 0.1547421    total: 45.5s    remaining: 29.3s
608:    learn: 0.1547336    total: 45.6s    remaining: 29.2s
609:    learn: 0.1546935    total: 45.6s    remaining: 29.2s
610:    learn: 0.1546806    total: 45.8s    remaining: 29.1s
611:    learn: 0.1546616    total: 45.8s    remaining: 29.1s
612:    learn: 0.1546507    total: 45.9s    remaining: 29s
613:    learn: 0.1546315    total: 46s      remaining: 28.9s
614:    learn: 0.1546205    total: 46.1s    remaining: 28.8s
615:    learn: 0.1546161    total: 46.1s    remaining: 28.8s
616:    learn: 0.1546027    total: 46.2s    remaining: 28.7s
617:    learn: 0.1545951    total: 46.3s    remaining: 28.6s
618:    learn: 0.1545760    total: 46.3s    remaining: 28.5s
619:    learn: 0.1545638    total: 46.4s    remaining: 28.4s
620:    learn: 0.1545525    total: 46.5s    remaining: 28.4s
621:    learn: 0.1545416    total: 46.5s    remaining: 28.3s
622:    learn: 0.1545315    total: 46.6s    remaining: 28.2s
623:    learn: 0.1545235    total: 46.7s    remaining: 28.1s
624:    learn: 0.1545153    total: 46.7s    remaining: 28s
625:    learn: 0.1544946    total: 46.8s    remaining: 28s
626:    learn: 0.1544820    total: 46.9s    remaining: 27.9s
627:    learn: 0.1544584    total: 46.9s    remaining: 27.8s
628:    learn: 0.1544468    total: 47s      remaining: 27.7s
629:    learn: 0.1544392    total: 47.1s    remaining: 27.7s
630:    learn: 0.1544167    total: 47.2s    remaining: 27.6s
631:    learn: 0.1543790    total: 47.3s    remaining: 27.5s
632:    learn: 0.1543659    total: 47.3s    remaining: 27.4s
633:    learn: 0.1543579    total: 47.4s    remaining: 27.4s
634:    learn: 0.1543469    total: 47.5s    remaining: 27.3s
635:    learn: 0.1543359    total: 47.5s    remaining: 27.2s
636:    learn: 0.1542795    total: 47.6s    remaining: 27.1s
637:    learn: 0.1542571    total: 47.7s    remaining: 27.1s
638:    learn: 0.1542313    total: 47.7s    remaining: 27s
639:    learn: 0.1542230    total: 47.8s    remaining: 26.9s
640:    learn: 0.1542150    total: 47.9s    remaining: 26.8s
641:    learn: 0.1542061    total: 48s      remaining: 26.7s
642:    learn: 0.1541971    total: 48s      remaining: 26.7s
643:    learn: 0.1541895    total: 48.1s    remaining: 26.6s
```

```
644:    learn: 0.1541773    total: 48.2s    remaining: 26.5s
645:    learn: 0.1541649    total: 48.3s    remaining: 26.4s
646:    learn: 0.1541526    total: 48.3s    remaining: 26.4s
647:    learn: 0.1541286    total: 48.4s    remaining: 26.3s
648:    learn: 0.1541137    total: 48.5s    remaining: 26.2s
649:    learn: 0.1541048    total: 48.5s    remaining: 26.1s
650:    learn: 0.1540958    total: 48.6s    remaining: 26.1s
651:    learn: 0.1540530    total: 48.7s    remaining: 26s
652:    learn: 0.1540482    total: 48.8s    remaining: 25.9s
653:    learn: 0.1540364    total: 48.9s    remaining: 25.9s
654:    learn: 0.1540235    total: 49s      remaining: 25.8s
655:    learn: 0.1540108    total: 49.1s    remaining: 25.8s
656:    learn: 0.1540010    total: 49.2s    remaining: 25.7s
657:    learn: 0.1539920    total: 49.3s    remaining: 25.6s
658:    learn: 0.1539744    total: 49.4s    remaining: 25.6s
659:    learn: 0.1539668    total: 49.5s    remaining: 25.5s
660:    learn: 0.1539582    total: 49.6s    remaining: 25.4s
661:    learn: 0.1539469    total: 49.6s    remaining: 25.3s
662:    learn: 0.1539384    total: 49.7s    remaining: 25.3s
663:    learn: 0.1539315    total: 49.8s    remaining: 25.2s
664:    learn: 0.1539215    total: 49.8s    remaining: 25.1s
665:    learn: 0.1539075    total: 49.9s    remaining: 25s
666:    learn: 0.1538893    total: 50s      remaining: 25s
667:    learn: 0.1538760    total: 50.1s    remaining: 24.9s
668:    learn: 0.1538693    total: 50.2s    remaining: 24.8s
669:    learn: 0.1538574    total: 50.3s    remaining: 24.8s
670:    learn: 0.1538472    total: 50.4s    remaining: 24.7s
671:    learn: 0.1538388    total: 50.4s    remaining: 24.6s
672:    learn: 0.1538320    total: 50.5s    remaining: 24.5s
673:    learn: 0.1538186    total: 50.6s    remaining: 24.5s
674:    learn: 0.1537906    total: 50.6s    remaining: 24.4s
675:    learn: 0.1537542    total: 50.7s    remaining: 24.3s
676:    learn: 0.1537438    total: 50.8s    remaining: 24.3s
677:    learn: 0.1537379    total: 50.9s    remaining: 24.2s
678:    learn: 0.1537293    total: 51s      remaining: 24.1s
679:    learn: 0.1537208    total: 51.1s    remaining: 24s
680:    learn: 0.1537081    total: 51.2s    remaining: 24s
681:    learn: 0.1536954    total: 51.3s    remaining: 23.9s
682:    learn: 0.1536654    total: 51.4s    remaining: 23.8s
683:    learn: 0.1536555    total: 51.5s    remaining: 23.8s
684:    learn: 0.1536492    total: 51.5s    remaining: 23.7s
685:    learn: 0.1536399    total: 51.6s    remaining: 23.6s
686:    learn: 0.1536277    total: 51.7s    remaining: 23.5s
687:    learn: 0.1535964    total: 51.8s    remaining: 23.5s
688:    learn: 0.1535831    total: 51.8s    remaining: 23.4s
689:    learn: 0.1535603    total: 51.9s    remaining: 23.3s
690:    learn: 0.1535172    total: 52s      remaining: 23.2s
691:    learn: 0.1535022    total: 52s      remaining: 23.2s
```

```
692:    learn: 0.1534905    total: 52.1s    remaining: 23.1s
693:    learn: 0.1534620    total: 52.2s    remaining: 23s
694:    learn: 0.1534494    total: 52.2s    remaining: 22.9s
695:    learn: 0.1534370    total: 52.3s    remaining: 22.8s
696:    learn: 0.1534268    total: 52.4s    remaining: 22.8s
697:    learn: 0.1533782    total: 52.4s    remaining: 22.7s
698:    learn: 0.1533669    total: 52.5s    remaining: 22.6s
699:    learn: 0.1533556    total: 52.6s    remaining: 22.5s
700:    learn: 0.1533387    total: 52.6s    remaining: 22.5s
701:    learn: 0.1533276    total: 52.7s    remaining: 22.4s
702:    learn: 0.1533126    total: 52.8s    remaining: 22.3s
703:    learn: 0.1533032    total: 52.8s    remaining: 22.2s
704:    learn: 0.1532891    total: 52.9s    remaining: 22.1s
705:    learn: 0.1532768    total: 53s      remaining: 22.1s
706:    learn: 0.1532624    total: 53s      remaining: 22s
707:    learn: 0.1532523    total: 53.1s    remaining: 21.9s
708:    learn: 0.1532420    total: 53.2s    remaining: 21.8s
709:    learn: 0.1532318    total: 53.2s    remaining: 21.7s
710:    learn: 0.1532145    total: 53.3s    remaining: 21.7s
711:    learn: 0.1532053    total: 53.4s    remaining: 21.6s
712:    learn: 0.1531916    total: 53.4s    remaining: 21.5s
713:    learn: 0.1531869    total: 53.5s    remaining: 21.4s
714:    learn: 0.1531779    total: 53.6s    remaining: 21.4s
715:    learn: 0.1531694    total: 53.7s    remaining: 21.3s
716:    learn: 0.1531566    total: 53.8s    remaining: 21.2s
717:    learn: 0.1531374    total: 53.9s    remaining: 21.2s
718:    learn: 0.1530928    total: 53.9s    remaining: 21.1s
719:    learn: 0.1530840    total: 54s      remaining: 21s
720:    learn: 0.1530696    total: 54.1s    remaining: 20.9s
721:    learn: 0.1530566    total: 54.1s    remaining: 20.8s
722:    learn: 0.1530493    total: 54.2s    remaining: 20.8s
723:    learn: 0.1530381    total: 54.3s    remaining: 20.7s
724:    learn: 0.1530277    total: 54.3s    remaining: 20.6s
725:    learn: 0.1530173    total: 54.4s    remaining: 20.5s
726:    learn: 0.1530072    total: 54.5s    remaining: 20.5s
727:    learn: 0.1529946    total: 54.5s    remaining: 20.4s
728:    learn: 0.1529817    total: 54.6s    remaining: 20.3s
729:    learn: 0.1529401    total: 54.7s    remaining: 20.2s
730:    learn: 0.1529296    total: 54.8s    remaining: 20.1s
731:    learn: 0.1529173    total: 54.8s    remaining: 20.1s
732:    learn: 0.1528702    total: 54.9s    remaining: 20s
733:    learn: 0.1528603    total: 55s      remaining: 19.9s
734:    learn: 0.1528424    total: 55.1s    remaining: 19.9s
735:    learn: 0.1528355    total: 55.2s    remaining: 19.8s
736:    learn: 0.1528244    total: 55.3s    remaining: 19.7s
737:    learn: 0.1528144    total: 55.3s    remaining: 19.6s
738:    learn: 0.1527997    total: 55.4s    remaining: 19.6s
739:    learn: 0.1527895    total: 55.5s    remaining: 19.5s
```

```
740:    learn: 0.1527775    total: 55.5s    remaining: 19.4s
741:    learn: 0.1527577    total: 55.6s    remaining: 19.3s
742:    learn: 0.1527446    total: 55.7s    remaining: 19.3s
743:    learn: 0.1527335    total: 55.8s    remaining: 19.2s
744:    learn: 0.1527257    total: 55.9s    remaining: 19.1s
745:    learn: 0.1527154    total: 55.9s    remaining: 19s
746:    learn: 0.1527052    total: 56s      remaining: 19s
747:    learn: 0.1526971    total: 56.1s    remaining: 18.9s
748:    learn: 0.1526890    total: 56.1s    remaining: 18.8s
749:    learn: 0.1526787    total: 56.2s    remaining: 18.7s
750:    learn: 0.1526659    total: 56.3s    remaining: 18.7s
751:    learn: 0.1526558    total: 56.3s    remaining: 18.6s
752:    learn: 0.1526312    total: 56.4s    remaining: 18.5s
753:    learn: 0.1526184    total: 56.5s    remaining: 18.4s
754:    learn: 0.1526072    total: 56.5s    remaining: 18.3s
755:    learn: 0.1525736    total: 56.6s    remaining: 18.3s
756:    learn: 0.1525577    total: 56.7s    remaining: 18.2s
757:    learn: 0.1525490    total: 56.8s    remaining: 18.1s
758:    learn: 0.1525387    total: 56.8s    remaining: 18s
759:    learn: 0.1525256    total: 56.9s    remaining: 18s
760:    learn: 0.1524989    total: 57s      remaining: 17.9s
761:    learn: 0.1524930    total: 57s      remaining: 17.8s
762:    learn: 0.1524811    total: 57.1s    remaining: 17.7s
763:    learn: 0.1524743    total: 57.2s    remaining: 17.7s
764:    learn: 0.1524669    total: 57.3s    remaining: 17.6s
765:    learn: 0.1524591    total: 57.3s    remaining: 17.5s
766:    learn: 0.1524500    total: 57.4s    remaining: 17.4s
767:    learn: 0.1524374    total: 57.5s    remaining: 17.4s
768:    learn: 0.1524244    total: 57.5s    remaining: 17.3s
769:    learn: 0.1524109    total: 57.6s    remaining: 17.2s
770:    learn: 0.1523855    total: 57.7s    remaining: 17.1s
771:    learn: 0.1523538    total: 57.7s    remaining: 17.1s
772:    learn: 0.1523398    total: 57.8s    remaining: 17s
773:    learn: 0.1523042    total: 57.9s    remaining: 16.9s
774:    learn: 0.1522973    total: 57.9s    remaining: 16.8s
775:    learn: 0.1522895    total: 58s      remaining: 16.7s
776:    learn: 0.1522828    total: 58.1s    remaining: 16.7s
777:    learn: 0.1522722    total: 58.1s    remaining: 16.6s
778:    learn: 0.1522338    total: 58.2s    remaining: 16.5s
779:    learn: 0.1522207    total: 58.3s    remaining: 16.4s
780:    learn: 0.1522105    total: 58.3s    remaining: 16.4s
781:    learn: 0.1521984    total: 58.4s    remaining: 16.3s
782:    learn: 0.1521874    total: 58.5s    remaining: 16.2s
783:    learn: 0.1521745    total: 58.5s    remaining: 16.1s
784:    learn: 0.1521637    total: 58.6s    remaining: 16.1s
785:    learn: 0.1521544    total: 58.7s    remaining: 16s
786:    learn: 0.1521383    total: 58.7s    remaining: 15.9s
787:    learn: 0.1521286    total: 58.8s    remaining: 15.8s
```

```
788:    learn: 0.1521243    total: 58.9s    remaining: 15.7s
789:    learn: 0.1521132    total: 58.9s    remaining: 15.7s
790:    learn: 0.1521004    total: 59s      remaining: 15.6s
791:    learn: 0.1520907    total: 59.1s    remaining: 15.5s
792:    learn: 0.1520827    total: 59.1s    remaining: 15.4s
793:    learn: 0.1520723    total: 59.2s    remaining: 15.4s
794:    learn: 0.1520606    total: 59.3s    remaining: 15.3s
795:    learn: 0.1520526    total: 59.3s    remaining: 15.2s
796:    learn: 0.1520385    total: 59.4s    remaining: 15.1s
797:    learn: 0.1520220    total: 59.5s    remaining: 15.1s
798:    learn: 0.1520107    total: 59.5s    remaining: 15s
799:    learn: 0.1520013    total: 59.6s    remaining: 14.9s
800:    learn: 0.1519933    total: 59.6s    remaining: 14.8s
801:    learn: 0.1519801    total: 59.7s    remaining: 14.7s
802:    learn: 0.1519724    total: 59.8s    remaining: 14.7s
803:    learn: 0.1519636    total: 59.9s    remaining: 14.6s
804:    learn: 0.1519544    total: 59.9s    remaining: 14.5s
805:    learn: 0.1519171    total: 1m       remaining: 14.4s
806:    learn: 0.1519036    total: 1m       remaining: 14.4s
807:    learn: 0.1518935    total: 1m       remaining: 14.3s
808:    learn: 0.1518816    total: 1m       remaining: 14.2s
809:    learn: 0.1518729    total: 1m       remaining: 14.1s
810:    learn: 0.1518656    total: 1m       remaining: 14.1s
811:    learn: 0.1518519    total: 1m       remaining: 14s
812:    learn: 0.1518444    total: 1m       remaining: 13.9s
813:    learn: 0.1518347    total: 1m       remaining: 13.8s
814:    learn: 0.1518251    total: 1m       remaining: 13.8s
815:    learn: 0.1518189    total: 1m       remaining: 13.7s
816:    learn: 0.1518054    total: 1m       remaining: 13.6s
817:    learn: 0.1517977    total: 1m       remaining: 13.5s
818:    learn: 0.1517845    total: 1m       remaining: 13.4s
819:    learn: 0.1517731    total: 1m       remaining: 13.4s
820:    learn: 0.1517651    total: 1m       remaining: 13.3s
821:    learn: 0.1517610    total: 1m 1s    remaining: 13.2s
822:    learn: 0.1517499    total: 1m 1s    remaining: 13.1s
823:    learn: 0.1517359    total: 1m 1s    remaining: 13.1s
824:    learn: 0.1517248    total: 1m 1s    remaining: 13s
825:    learn: 0.1517184    total: 1m 1s    remaining: 12.9s
826:    learn: 0.1517091    total: 1m 1s    remaining: 12.8s
827:    learn: 0.1517020    total: 1m 1s    remaining: 12.8s
828:    learn: 0.1516925    total: 1m 1s    remaining: 12.7s
829:    learn: 0.1516881    total: 1m 1s    remaining: 12.6s
830:    learn: 0.1516789    total: 1m 1s    remaining: 12.5s
831:    learn: 0.1516684    total: 1m 1s    remaining: 12.4s
832:    learn: 0.1516584    total: 1m 1s    remaining: 12.4s
833:    learn: 0.1516402    total: 1m 1s    remaining: 12.3s
834:    learn: 0.1516267    total: 1m 1s    remaining: 12.2s
835:    learn: 0.1516174    total: 1m 1s    remaining: 12.1s
```

```
836:    learn: 0.1516073    total: 1m 1s    remaining: 12.1s
837:    learn: 0.1515934    total: 1m 2s    remaining: 12s
838:    learn: 0.1515798    total: 1m 2s    remaining: 11.9s
839:    learn: 0.1515719    total: 1m 2s    remaining: 11.8s
840:    learn: 0.1515590    total: 1m 2s    remaining: 11.8s
841:    learn: 0.1515314    total: 1m 2s    remaining: 11.7s
842:    learn: 0.1515232    total: 1m 2s    remaining: 11.6s
843:    learn: 0.1515100    total: 1m 2s    remaining: 11.5s
844:    learn: 0.1515021    total: 1m 2s    remaining: 11.5s
845:    learn: 0.1514945    total: 1m 2s    remaining: 11.4s
846:    learn: 0.1514830    total: 1m 2s    remaining: 11.3s
847:    learn: 0.1514779    total: 1m 2s    remaining: 11.3s
848:    learn: 0.1514615    total: 1m 2s    remaining: 11.2s
849:    learn: 0.1514505    total: 1m 3s    remaining: 11.1s
850:    learn: 0.1514380    total: 1m 3s    remaining: 11.1s
851:    learn: 0.1514287    total: 1m 3s    remaining: 11s
852:    learn: 0.1514187    total: 1m 3s    remaining: 10.9s
853:    learn: 0.1514119    total: 1m 3s    remaining: 10.8s
854:    learn: 0.1514033    total: 1m 3s    remaining: 10.8s
855:    learn: 0.1513925    total: 1m 3s    remaining: 10.7s
856:    learn: 0.1513766    total: 1m 3s    remaining: 10.6s
857:    learn: 0.1513722    total: 1m 3s    remaining: 10.6s
858:    learn: 0.1513611    total: 1m 3s    remaining: 10.5s
859:    learn: 0.1513542    total: 1m 3s    remaining: 10.4s
860:    learn: 0.1513432    total: 1m 4s    remaining: 10.3s
861:    learn: 0.1513323    total: 1m 4s    remaining: 10.3s
862:    learn: 0.1513192    total: 1m 4s    remaining: 10.2s
863:    learn: 0.1513091    total: 1m 4s    remaining: 10.1s
864:    learn: 0.1512990    total: 1m 4s    remaining: 10s
865:    learn: 0.1512891    total: 1m 4s    remaining: 9.95s
866:    learn: 0.1512772    total: 1m 4s    remaining: 9.88s
867:    learn: 0.1512672    total: 1m 4s    remaining: 9.8s
868:    learn: 0.1512625    total: 1m 4s    remaining: 9.73s
869:    learn: 0.1512509    total: 1m 4s    remaining: 9.65s
870:    learn: 0.1512452    total: 1m 4s    remaining: 9.58s
871:    learn: 0.1512350    total: 1m 4s    remaining: 9.5s
872:    learn: 0.1512225    total: 1m 4s    remaining: 9.43s
873:    learn: 0.1512142    total: 1m 4s    remaining: 9.35s
874:    learn: 0.1511975    total: 1m 4s    remaining: 9.27s
875:    learn: 0.1511890    total: 1m 4s    remaining: 9.2s
876:    learn: 0.1511372    total: 1m 5s    remaining: 9.12s
877:    learn: 0.1511117    total: 1m 5s    remaining: 9.05s
878:    learn: 0.1511009    total: 1m 5s    remaining: 8.97s
879:    learn: 0.1510925    total: 1m 5s    remaining: 8.9s
880:    learn: 0.1510878    total: 1m 5s    remaining: 8.83s
881:    learn: 0.1510788    total: 1m 5s    remaining: 8.75s
882:    learn: 0.1510719    total: 1m 5s    remaining: 8.68s
883:    learn: 0.1510608    total: 1m 5s    remaining: 8.6s
```

```
884:    learn: 0.1510498    total: 1m 5s    remaining: 8.52s
885:    learn: 0.1510402    total: 1m 5s    remaining: 8.45s
886:    learn: 0.1510285    total: 1m 5s    remaining: 8.37s
887:    learn: 0.1510205    total: 1m 5s    remaining: 8.3s
888:    learn: 0.1510108    total: 1m 5s    remaining: 8.22s
889:    learn: 0.1510027    total: 1m 5s    remaining: 8.15s
890:    learn: 0.1509940    total: 1m 5s    remaining: 8.07s
891:    learn: 0.1509861    total: 1m 6s    remaining: 8s
892:    learn: 0.1509691    total: 1m 6s    remaining: 7.92s
893:    learn: 0.1509635    total: 1m 6s    remaining: 7.85s
894:    learn: 0.1509526    total: 1m 6s    remaining: 7.77s
895:    learn: 0.1509413    total: 1m 6s    remaining: 7.7s
896:    learn: 0.1509298    total: 1m 6s    remaining: 7.62s
897:    learn: 0.1509232    total: 1m 6s    remaining: 7.55s
898:    learn: 0.1509125    total: 1m 6s    remaining: 7.47s
899:    learn: 0.1509038    total: 1m 6s    remaining: 7.4s
900:    learn: 0.1508922    total: 1m 6s    remaining: 7.32s
901:    learn: 0.1508803    total: 1m 6s    remaining: 7.25s
902:    learn: 0.1508690    total: 1m 6s    remaining: 7.17s
903:    learn: 0.1508558    total: 1m 6s    remaining: 7.1s
904:    learn: 0.1508392    total: 1m 6s    remaining: 7.03s
905:    learn: 0.1508303    total: 1m 6s    remaining: 6.95s
906:    learn: 0.1508194    total: 1m 7s    remaining: 6.88s
907:    learn: 0.1508080    total: 1m 7s    remaining: 6.8s
908:    learn: 0.1507921    total: 1m 7s    remaining: 6.73s
909:    learn: 0.1507840    total: 1m 7s    remaining: 6.65s
910:    learn: 0.1507718    total: 1m 7s    remaining: 6.58s
911:    learn: 0.1507503    total: 1m 7s    remaining: 6.51s
912:    learn: 0.1507399    total: 1m 7s    remaining: 6.44s
913:    learn: 0.1507317    total: 1m 7s    remaining: 6.36s
914:    learn: 0.1507227    total: 1m 7s    remaining: 6.29s
915:    learn: 0.1507116    total: 1m 7s    remaining: 6.22s
916:    learn: 0.1507028    total: 1m 7s    remaining: 6.15s
917:    learn: 0.1506914    total: 1m 7s    remaining: 6.07s
918:    learn: 0.1506774    total: 1m 8s    remaining: 6s
919:    learn: 0.1506646    total: 1m 8s    remaining: 5.92s
920:    learn: 0.1506568    total: 1m 8s    remaining: 5.85s
921:    learn: 0.1506449    total: 1m 8s    remaining: 5.77s
922:    learn: 0.1506360    total: 1m 8s    remaining: 5.7s
923:    learn: 0.1506260    total: 1m 8s    remaining: 5.63s
924:    learn: 0.1506098    total: 1m 8s    remaining: 5.55s
925:    learn: 0.1505792    total: 1m 8s    remaining: 5.48s
926:    learn: 0.1505674    total: 1m 8s    remaining: 5.4s
927:    learn: 0.1505562    total: 1m 8s    remaining: 5.33s
928:    learn: 0.1505446    total: 1m 8s    remaining: 5.26s
929:    learn: 0.1505327    total: 1m 8s    remaining: 5.18s
930:    learn: 0.1505135    total: 1m 8s    remaining: 5.11s
931:    learn: 0.1505038    total: 1m 9s    remaining: 5.04s
```

```
932:    learn: 0.1504902    total: 1m 9s    remaining: 4.96s
933:    learn: 0.1504807    total: 1m 9s    remaining: 4.89s
934:    learn: 0.1504716    total: 1m 9s    remaining: 4.81s
935:    learn: 0.1504643    total: 1m 9s    remaining: 4.74s
936:    learn: 0.1504529    total: 1m 9s    remaining: 4.67s
937:    learn: 0.1504326    total: 1m 9s    remaining: 4.59s
938:    learn: 0.1504210    total: 1m 9s    remaining: 4.52s
939:    learn: 0.1504006    total: 1m 9s    remaining: 4.44s
940:    learn: 0.1503610    total: 1m 9s    remaining: 4.37s
941:    learn: 0.1503527    total: 1m 9s    remaining: 4.29s
942:    learn: 0.1503418    total: 1m 9s    remaining: 4.22s
943:    learn: 0.1503310    total: 1m 9s    remaining: 4.15s
944:    learn: 0.1503223    total: 1m 10s    remaining: 4.08s
945:    learn: 0.1503183    total: 1m 10s    remaining: 4s
946:    learn: 0.1503086    total: 1m 10s    remaining: 3.93s
947:    learn: 0.1502849    total: 1m 10s    remaining: 3.86s
948:    learn: 0.1502763    total: 1m 10s    remaining: 3.79s
949:    learn: 0.1502628    total: 1m 10s    remaining: 3.71s
950:    learn: 0.1502497    total: 1m 10s    remaining: 3.64s
951:    learn: 0.1502372    total: 1m 10s    remaining: 3.57s
952:    learn: 0.1502266    total: 1m 10s    remaining: 3.49s
953:    learn: 0.1502164    total: 1m 10s    remaining: 3.42s
954:    learn: 0.1502015    total: 1m 11s    remaining: 3.35s
955:    learn: 0.1501943    total: 1m 11s    remaining: 3.27s
956:    learn: 0.1501870    total: 1m 11s    remaining: 3.2s
957:    learn: 0.1501802    total: 1m 11s    remaining: 3.13s
958:    learn: 0.1501532    total: 1m 11s    remaining: 3.05s
959:    learn: 0.1501396    total: 1m 11s    remaining: 2.98s
960:    learn: 0.1501290    total: 1m 11s    remaining: 2.9s
961:    learn: 0.1501227    total: 1m 11s    remaining: 2.83s
962:    learn: 0.1501147    total: 1m 11s    remaining: 2.76s
963:    learn: 0.1501054    total: 1m 11s    remaining: 2.68s
964:    learn: 0.1500759    total: 1m 11s    remaining: 2.61s
965:    learn: 0.1500610    total: 1m 11s    remaining: 2.53s
966:    learn: 0.1500463    total: 1m 12s    remaining: 2.46s
967:    learn: 0.1500412    total: 1m 12s    remaining: 2.38s
968:    learn: 0.1500359    total: 1m 12s    remaining: 2.31s
969:    learn: 0.1500252    total: 1m 12s    remaining: 2.23s
970:    learn: 0.1500182    total: 1m 12s    remaining: 2.16s
971:    learn: 0.1499856    total: 1m 12s    remaining: 2.09s
972:    learn: 0.1499705    total: 1m 12s    remaining: 2.01s
973:    learn: 0.1499609    total: 1m 12s    remaining: 1.94s
974:    learn: 0.1499456    total: 1m 12s    remaining: 1.86s
975:    learn: 0.1499346    total: 1m 12s    remaining: 1.79s
976:    learn: 0.1499232    total: 1m 12s    remaining: 1.72s
977:    learn: 0.1499112    total: 1m 13s    remaining: 1.64s
978:    learn: 0.1498996    total: 1m 13s    remaining: 1.57s
979:    learn: 0.1498887    total: 1m 13s    remaining: 1.49s
```

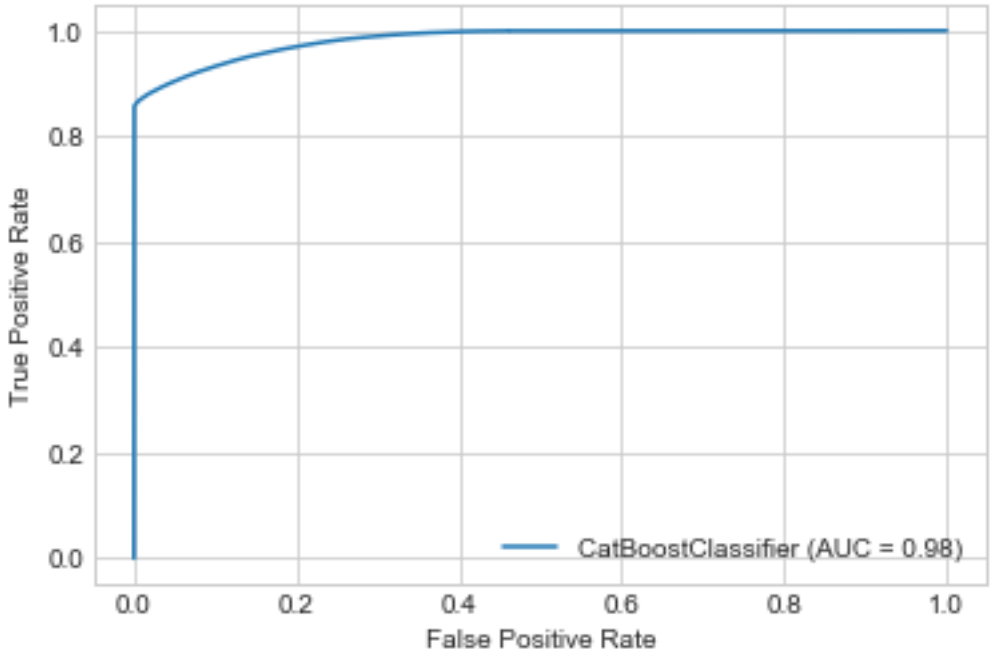
```
980:        learn: 0.1498834        total: 1m 13s        remaining: 1.42s
981:        learn: 0.1498647        total: 1m 13s        remaining: 1.34s
982:        learn: 0.1498589        total: 1m 13s        remaining: 1.27s
983:        learn: 0.1498539        total: 1m 13s        remaining: 1.2s
984:        learn: 0.1498494        total: 1m 13s        remaining: 1.12s
985:        learn: 0.1498440        total: 1m 13s        remaining: 1.05s
986:        learn: 0.1498383        total: 1m 13s        remaining: 971ms
987:        learn: 0.1498307        total: 1m 13s        remaining: 896ms
988:        learn: 0.1498205        total: 1m 13s        remaining: 822ms
989:        learn: 0.1498128        total: 1m 13s        remaining: 747ms
990:        learn: 0.1498045        total: 1m 14s        remaining: 672ms
991:        learn: 0.1497923        total: 1m 14s        remaining: 598ms
992:        learn: 0.1497797        total: 1m 14s        remaining: 523ms
993:        learn: 0.1497748        total: 1m 14s        remaining: 448ms
994:        learn: 0.1497670        total: 1m 14s        remaining: 373ms
995:        learn: 0.1497595        total: 1m 14s        remaining: 299ms
996:        learn: 0.1497481        total: 1m 14s        remaining: 224ms
997:        learn: 0.1497324        total: 1m 14s        remaining: 149ms
998:        learn: 0.1497226        total: 1m 14s        remaining: 74.7ms
999:        learn: 0.1497028        total: 1m 14s        remaining: 0us
```

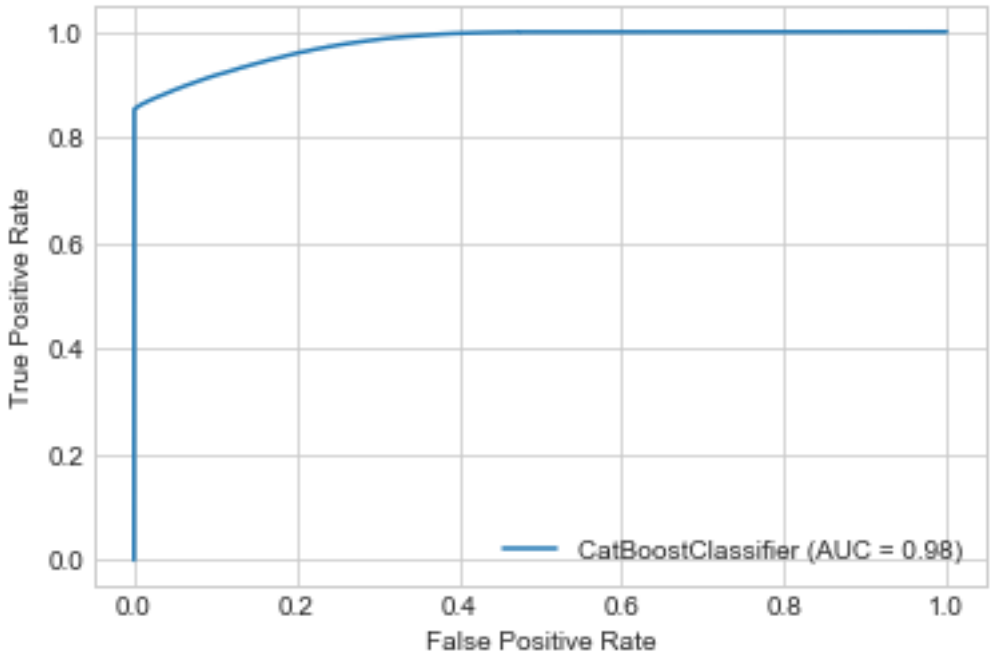
```
Out[57]: <catboost.core.CatBoostClassifier at 0x2ab8ce14860>
```

```
In [58]: Model_Performance(baseline_catboost,"Catboost Engine")
```

```
                Catboost Engine Performance Outline

         Training Results

Accuracy Score:  0.9303098526565817
Balanced Accuracy Score:  0.9303120626393908
Cohen Kappa Score:  0.8606203212264419
Roc Score:  0.9829784681444549
Confusion Matrix:
 [[232895   1176]
 [ 31450 202637]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.99      0.93    234071
           1       0.99      0.87      0.93    234087

    accuracy                           0.93    468158
   macro avg       0.94      0.93      0.93    468158
weighted avg       0.94      0.93      0.93    468158
```

```
         Testing Results

Accuracy Score:  0.9266397527910686
Balanced Accuracy Score:  0.9266345969715704
Cohen Kappa Score:  0.853277991841622
Roc Score:  0.9792018529808397
Confusion Matrix:
 [[99454   874]
 [13845 86467]]
Classification Report
               precision    recall  f1-score   support

           0       0.88      0.99      0.93    100328
           1       0.99      0.86      0.92    100312

    accuracy                           0.93    200640
   macro avg       0.93      0.93      0.93    200640
weighted avg       0.93      0.93      0.93    200640
```



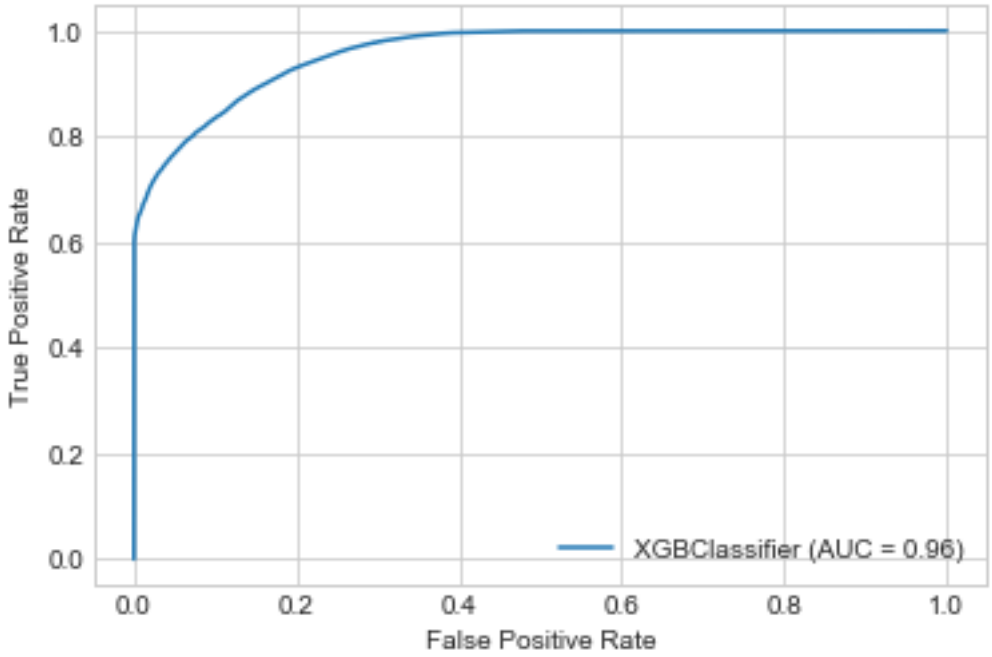


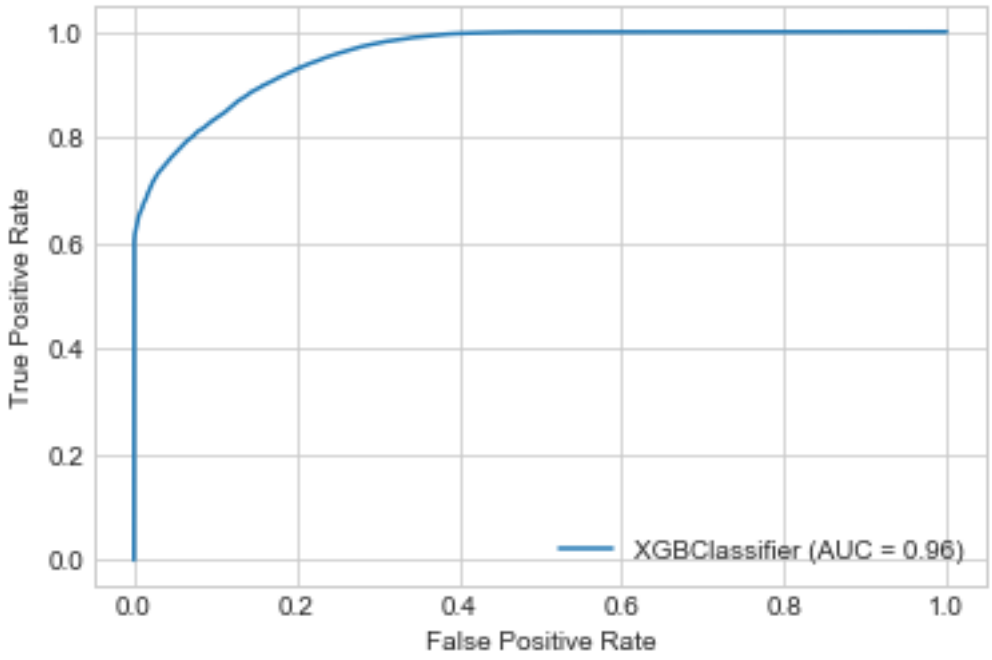
In [59]: *# Lets Create Baseline Xgboost*

```
baseline_Xgboost = XGBClassifier()
baseline_Xgboost.fit(X_train,y_train)
Model_Performance(baseline_Xgboost,"XGboost Engine")
```

```
        XGboost Engine Performance Outline

Training Results

Accuracy Score:  0.8655005361437805
Balanced Accuracy Score:  0.8654983525682213
Cohen Kappa Score:  0.7309998972937818
Roc Score:  0.9588602471430142
Confusion Matrix:
 [[187633  46438]
 [ 16529 217558]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.80      0.86    234071
           1       0.82      0.93      0.87    234087

    accuracy                           0.87    468158
   macro avg       0.87      0.87      0.86    468158
```

```
weighted avg       0.87      0.87      0.86    468158


         Testing Results

Accuracy Score:  0.8649272328548644
Balanced Accuracy Score:  0.8649324143756401
Cohen Kappa Score:  0.7298572639562915
Roc Score:  0.9589183831606062
Confusion Matrix:
 [[80258 20070]
 [ 7031 93281]]
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.80      0.86    100328
           1       0.82      0.93      0.87    100312

    accuracy                           0.86    200640
   macro avg       0.87      0.86      0.86    200640
weighted avg       0.87      0.86      0.86    200640
```



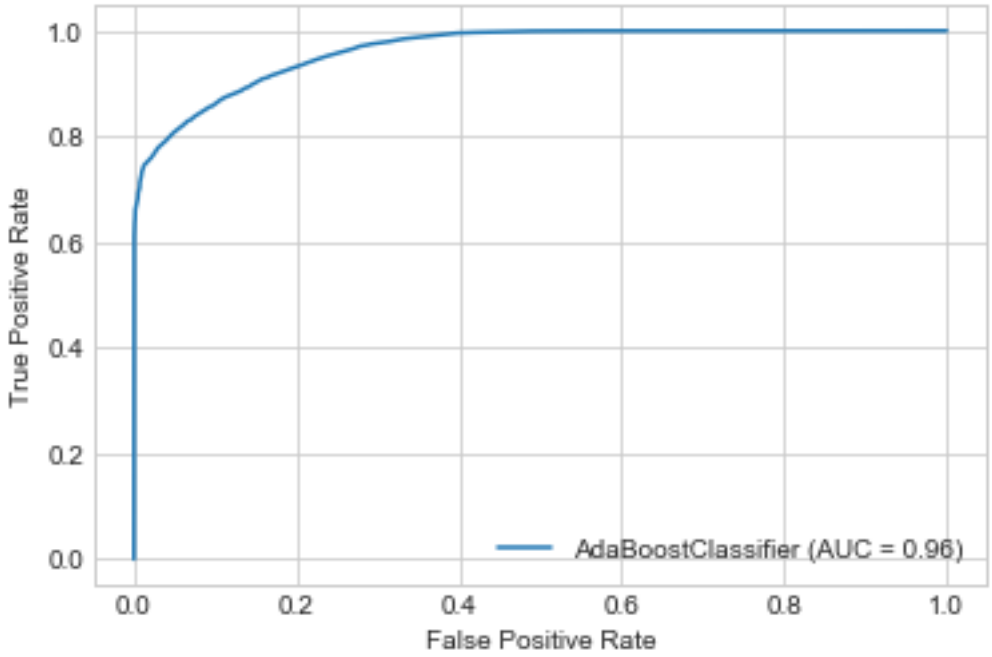


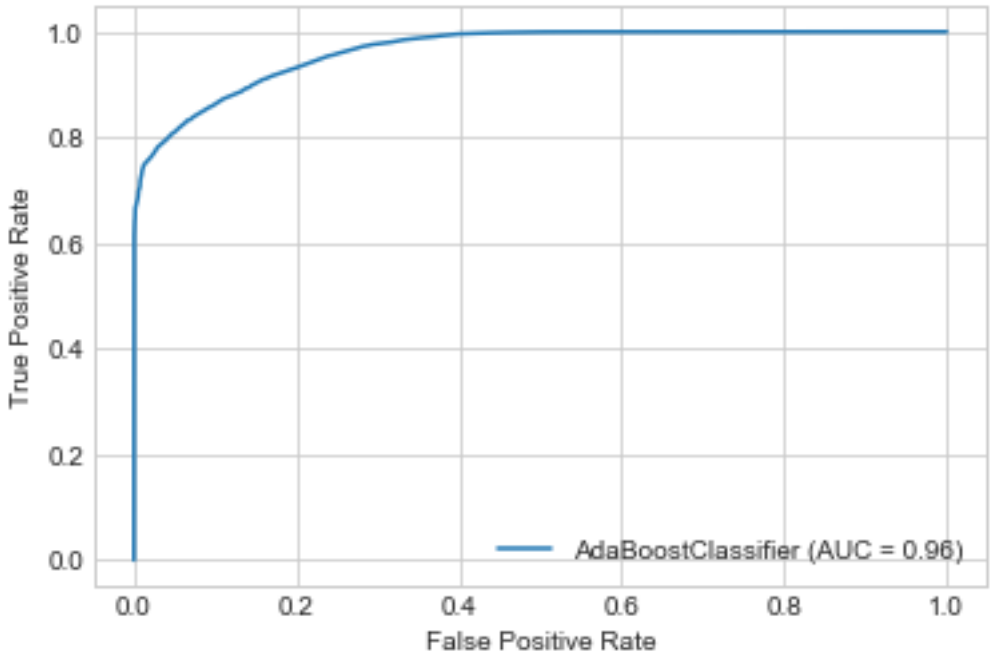
In [14]: *# Lets Create adaboost Engine*

```
baseline_adaboost = AdaBoostClassifier()
baseline_adaboost.fit(X_train,y_train)
Model_Performance(baseline_adaboost,"AdaBoost Engine")
```

```
        AdaBoost Engine Performance Outline

Training Results

Accuracy Score:  0.8754523045638438
Balanced Accuracy Score:  0.8754511407656131
Cohen Kappa Score:  0.7509040291143105
Roc Score:  0.9639459338362951
Confusion Matrix:
 [[196947  37124]
 [ 21184 212903]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.84      0.87    234071
           1       0.85      0.91      0.88    234087

    accuracy                           0.88    468158
   macro avg       0.88      0.88      0.88    468158
```

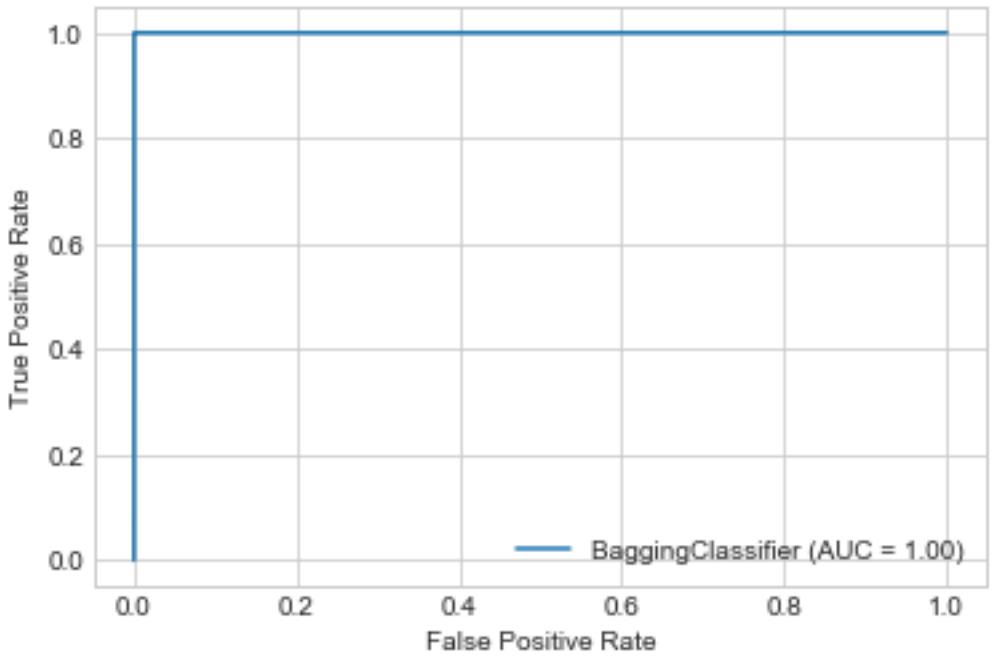
```
weighted avg       0.88      0.88      0.88    468158


         Testing Results

Accuracy Score:  0.8754734848484849
Balanced Accuracy Score:  0.8754762618502849
Cohen Kappa Score:  0.7509483517413785
Roc Score:  0.9643567189577512
Confusion Matrix:
 [[84341 15987]
 [ 8998 91314]]
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.84      0.87    100328
           1       0.85      0.91      0.88    100312

    accuracy                           0.88    200640
   macro avg       0.88      0.88      0.88    200640
weighted avg       0.88      0.88      0.88    200640
```

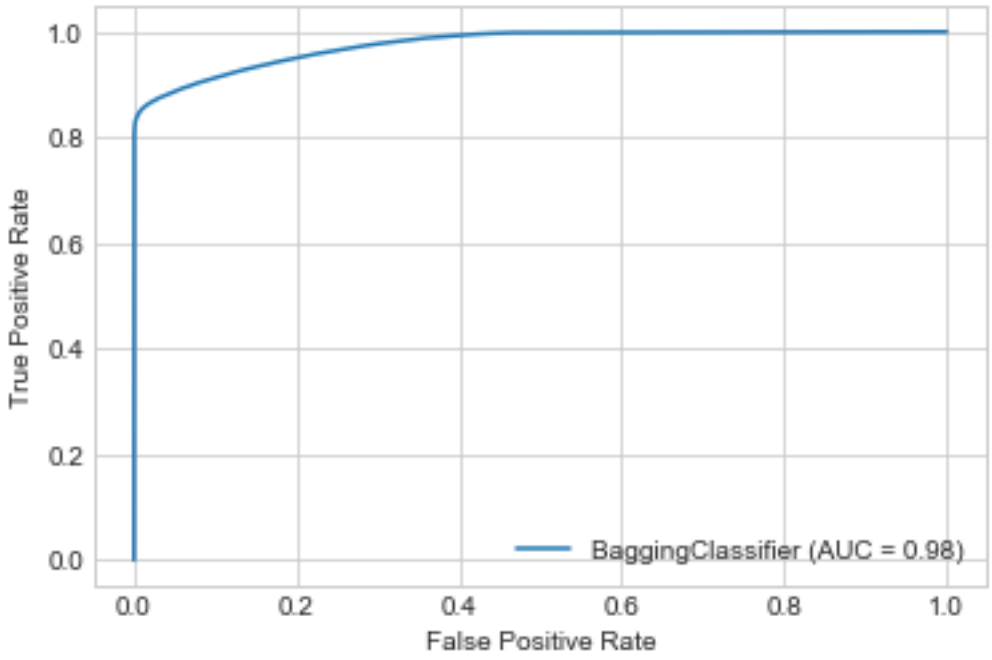





```
In [60]: # lets create bagging classifier
         baseline_bagging = BaggingClassifier(n_estimators=100)
         baseline_bagging.fit(X_train,y_train)
         Model_Performance(baseline_bagging,"Bagging Engine")

                 Bagging Engine Performance Outline

         Training Results

Accuracy Score:  0.9999081506670825
Balanced Accuracy Score:  0.9999081516160019
Cohen Kappa Score:  0.9998163013342991
Roc Score:  0.9999999582245741
Confusion Matrix:
 [[234056     15]
 [    28 234059]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    234071
           1       1.00      1.00      1.00    234087

    accuracy                           1.00    468158
   macro avg       1.00      1.00      1.00    468158
weighted avg       1.00      1.00      1.00    468158
```

```
         Testing Results

Accuracy Score:  0.9173245614035088
Balanced Accuracy Score:  0.9173224376878991
Cohen Kappa Score:  0.8346484196099156
Roc Score:  0.9759758738605895
Confusion Matrix:
 [[94705  5623]
 [10965 89347]]
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.94      0.92    100328
           1       0.94      0.89      0.92    100312

    accuracy                           0.92    200640
   macro avg       0.92      0.92      0.92    200640
weighted avg       0.92      0.92      0.92    200640
```

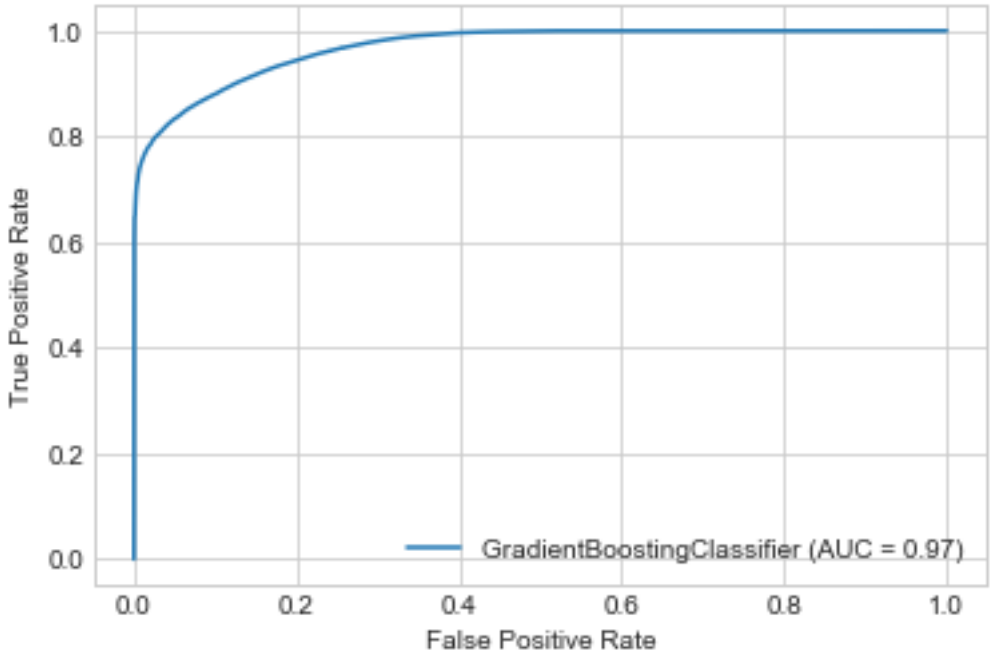


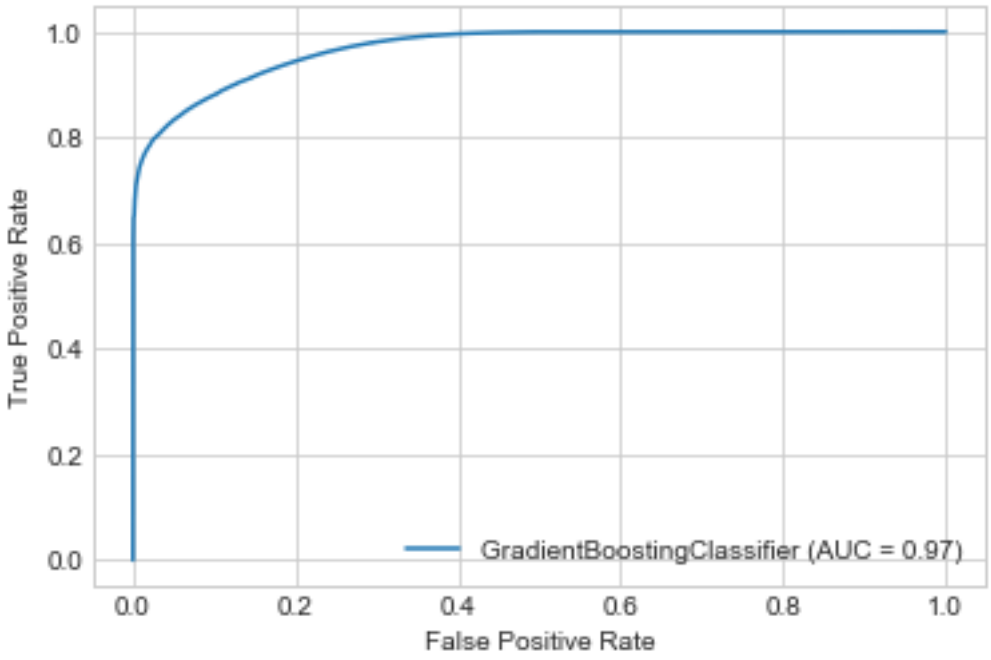


In [17]:
```
# lets create baseline gradient boosting

baseline_gradient = GradientBoostingClassifier()
baseline_gradient.fit(X_train,y_train)
Model_Performance(baseline_gradient,"Gradient Boosting Engine")
```

```
        Gradient Boosting Engine Performance Outline

Training Results

Accuracy Score:  0.8886871526279589
Balanced Accuracy Score:  0.8886869162632922
Cohen Kappa Score:  0.7773741998120279
Roc Score:  0.9689089957750263
Confusion Matrix:
 [[206397  27674]
 [ 24438 209649]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.88      0.89    234071
           1       0.88      0.90      0.89    234087

    accuracy                           0.89    468158
   macro avg       0.89      0.89      0.89    468158
```

```
weighted avg       0.89      0.89      0.89    468158


         Testing Results

Accuracy Score:  0.8883522727272727
Balanced Accuracy Score:  0.8883528646185659
Cohen Kappa Score:  0.7767048086845927
Roc Score:  0.9687916135474399
Confusion Matrix:
 [[88382 11946]
 [10455 89857]]
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.88      0.89    100328
           1       0.88      0.90      0.89    100312

    accuracy                           0.89    200640
   macro avg       0.89      0.89      0.89    200640
weighted avg       0.89      0.89      0.89    200640
```






```
In [21]: # Lets create a stochiastic gradient descent engine with kernel approximation
         Sampler =
         baseline_sgd  =  SGDClassifier(loss='hinge',n_jobs=-1)
         baseline_sgd.fit(X_train,y_train)

Out[21]: SGDClassifier(n_jobs=-1)

In [23]: print(baseline_sgd.score(X_train,y_train))
         print(baseline_sgd.score(X_test,y_test))

0.540975482636204
0.5404954146730463


In [24]: baseline_sgd.coef_

Out[24]: array([[-1.48345161e+00,  1.73247997e+02,  5.10255873e+03,
                  5.51444368e+01,  1.13874819e+03, -1.15700932e+03,
                 -2.53962057e+02,  1.25225847e+03,  2.60110169e+00,
                 -2.19695586e+03,  1.88504387e+01]])

In [25]: baseline_sgd.intercept_

Out[25]: array([40.45447831])

In [50]: # Deploying Random Forest Baseline
         jb.dump(baseline_random_forest,"BaselineRandomForestEngine.sav")
```

```
Out[50]: ['BaselineRandomForestEngine.sav']
```

- Baseline Random Forest is Saved

```
In [56]: # Just for Check
         baseline_random_forest.predict([[1, 44, 1, 28.0,
                 0, 2, 1, 40454.0, 26.0,
                 217]])

Out[56]: array([1], dtype=int64)

In [61]: # Deploying Baseline Bagging
         jb.dump(baseline_bagging,"BaselineBaggingEngine.sav")

Out[61]: ['BaselineBaggingEngine.sav']
```

- Baseline Bagging Engine is Saved

```
In [62]: # Deploying catboost model also

         jb.dump(baseline_catboost,"Catboost Engine.sav")

Out[62]: ['Catboost Engine.sav']
```

- Baseline catboost is deployed

1.3 Model Tuning

- Note: Model Tuning is not required because we have an optimal accuracy and the resulting Tuning can decrease the performance
- Lets jump to model evaluation

1.4 Model Evaluation

```
In [16]: model_evaluated  = cross_val_score(RandomForestClassifier(n_estimators=100),X,y,cv=10

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   7 out of  10 | elapsed:  6.6min remaining:  2.8min
[Parallel(n_jobs=-1)]: Done  10 out of  10 | elapsed:  8.9min finished


In [21]: print(f"The Average Cross validated Accuarcy for a Random Forest model is: {model_eval
         print(f"The Standard Deviation is:{model_evaluated.std()}")

The Average Cross validated Accuarcy for a Random Forest model is: 90.98%
The Standard Deviation is:0.119055220858802


In [ ]:
```