```
In [32]:
#importing the libraries
import pandas as pd
```

import matplotlib.pyplot as plt import seaborn as sns import pickle import warnings from sklearn import metrics

warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split from sklearn.ensemble import RandomForestClassifier from sklearn.model_selection import cross_val_score from sklearn.metrics import classification_report, accuracy_score

Load the Exploratory Data Analysis pickle file and build the model

df_eda = pickle.load(open('C:\\Users\\acer\\Desktop\\Vechicle_Insurance_DataSet\\models\\ExploratoryDataAnalysis.pkl','rb'))

df_eda.shape

(370789, 12)

df_eda.size

4449468

0

df_eda.isnull().sum().sum()

sns.set(rc={'figure.facecolor':'lightgreen'}) plt.figure(figsize=(15,5))

sns.heatmap(df_eda.corr(),annot=True)

<axessubplot:></axessubplot:>													
Age	1	-0.079	0.043	-0.25	0.051	-0.58	-0.0016	0.11	0.15	-0.79	0.22	0.27	- 1.00
Driving_License	-0.079	1	-0.0012	0.014	-0.01	0.043	-0.00051	0.0096	-0.018	0.04	-0.0073	-0.016	- 0.75
Region_Code	0.043	-0.0012	1	-0.024	-0.0024	-0.043	-0.0029	0.0095	0.0011	-0.044	0.015	0.027	
Previously_Insured	-0.25	0.014	-0.024	1	0.014	0.22	0.0029	-0.34	-0.083	0.36	-0.19	-0.82	- 0.50
Annual_Premium	0.051	-0.01	-0.0024	0.014	1	-0.11	-0.00088	0.019	0.0015	-0.0061	0.054	0.00039	- 0.25
Policy_Sales_Channel	-0.58	0.043	-0.043	0.22	-0.11	1	9.9e-06	-0.14	-0.11	0.57	-0.14	-0.22	- 0.23
Vintage	-0.0016	-0.00051	-0.0029	0.0029	-0.00088	9.9e-06	1	-0.0016	-0.0026	0.0028	0.00019	-0.0024	- 0.00
Response	0.11	0.0096	0.0095	-0.34	0.019	-0.14	-0.0016	1	0.052	-0.21	0.11	0.35	
Gender_Male	0.15	-0.018	0.0011	-0.083	0.0015	-0.11	-0.0026	0.052	1	-0.17	0.044	0.092	- -0.25
Vehicle_Age_< 1 Year	-0.79	0.04	-0.044	0.36	-0.0061	0.57	0.0028	-0.21	-0.17	1	-0.18	-0.37	0.50
Vehicle_Age_> 2 Years	0.22	-0.0073	0.015	-0.19	0.054	-0.14	0.00019	0.11	0.044	-0.18	1	0.2	
Vehicle_Damage_Yes	0.27	-0.016	0.027	-0.82	0.00039	-0.22	-0.0024	0.35	0.092	-0.37	0.2	1	- -0.75
	Age	Driving_License	Region_Code	Previously_Insured	Annaal_Premium	Policy_Sales_Channel	Vintage	Response	Gender_Male	Vehicle_Age_< 1 Year	Vehicle_Age_> 2 Years	Vehicle_Damage_Yes	

In [43]:

In [44]:

In [33]:

In [34]:

Out[34]:

In [35]:

Out[35]:

In [36]:

Out[36]:

In [38]:

Out[38]:

plt.figure(figsize=(30,5)) # sns.boxplot(data=df eda)

In [45]:

#pd.concat([X,y],axis=1)

```
X=df_eda.drop(columns=['Response'],axis=1)
y=df_eda['Response']
                                                                                                                                                    In [48]:
#splitting the data into training and testing data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=1)
                                                                                                                                                    In [49]:
#size of the train and test data
X_train.shape,X_test.shape,y_train.shape,y_test.shape
                                                                                                                                                   Out[49]:
((259552, 11), (111237, 11), (259552,), (111237,))
                                                                                                                                                    In [51]:
#building a Random Forest classifier model
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
                                                                                                                                                   Out[51]:
▼ RandomForestClassifier
RandomForestClassifier()
                                                                                                                                                    In [55]:
#predicting the outcome value
y_pred=rfc.predict(X_test)
y_pred
                                                                                                                                                   Out[55]:
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
                                                                                                                                                    In [53]:
#evaluating the performance of the training data
rfc.score(X_train,y_train)
                                                                                                                                                   Out[53]:
0.9998921218098878
                                                                                                                                                    In [56]:
#evaluating the performance of the test data/predicting data
accuracy_score(y_test,y_pred)
                                                                                                                                                   Out[56]:
0.867472154049462
                                                                                                                                                    In [58]:
print(classification_report(y_test,y_pred))
metrics.confusion_matrix
        precision recall f1-score support
            0.89
                    0.97
                            0.93
                                   97751
            0.36
                                   13486
                    0.12
                            0.18
      1
                           0.87 111237
  accuracy
                0.62
                        0.54
                                0.55 111237
 macro avg
weighted avg
                 0.82
                         0.87
                                 0.84 111237
                                                                                                                                                    In [62]:
print(sns.heatmap(metrics.confusion_matrix(y_test,y_pred),annot=True))
```

#independent and dependent variable

AxesSubplot(0.125,0.125;0.62x0.755)

- 80000
- 9.5e+04
- 2.8e+03
- 60000
- 40000
- 20000

1

Loading the test model prediction data in pickle file

0

with open(r'C:\Users\acer\Desktop\Vechicle_Insurance_DataSet\models\test_model.pkl','wb') as file: # pickle.dump(rfc,file) In [65]: