In [60]:
```
#Using pandas for loading the csv file into a dataframe
import pandas as pd

#Using matplotlib library for basic graph plotting
import matplotlib.pyplot as plt

#Using seaborn library that allows us to optimize matplotlib's output
import seaborn as sns

##Using numpy library
import numpy as np
```

In [61]:
```
#Loading the csv file into the pandas dataframe
damage = pd.read_csv('aug_test.csv')
damage
```

Out[61]:

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage
0	57782	Female	34	1	39.0	1	1-2 Year	No	38244.0	124.0	146
1	286811	Female	55	1	28.0	0	> 2 Years	Yes	37577.0	122.0	109
2	117823	Male	39	1	28.0	1	1-2 Year	No	24578.0	26.0	63
3	213992	Male	28	1	50.0	1	1-2 Year	No	40507.0	8.0	129
4	324756	Female	24	1	10.0	0	< 1 Year	Yes	36783.0	152.0	201
...	...	...	...	...	...	...	...	...	...	...	...
78268	847	Male	43	1	39.0	0	1-2 Year	Yes	2630.0	124.0	26
78269	417524	Female	21	1	12.0	1	< 1 Year	No	32937.0	152.0	185
78270	188087	Male	48	1	29.0	1	1-2 Year	No	35247.0	124.0	101
78271	215680	Male	64	1	5.0	1	1-2 Year	No	25705.0	26.0	86
78272	138006	Female	25	1	41.0	1	< 1 Year	No	27752.0	152.0	235

78273 rows × 11 columns

In [62]:
```
#Checking if tnere are null values
damage.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 78273 entries, 0 to 78272
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   id                      78273 non-null  int64  
 1   Gender                  78273 non-null  object 
 2   Age                     78273 non-null  int64  
 3   Driving_License         78273 non-null  int64  
 4   Region_Code             78273 non-null  float64
 5   Previously_Insured      78273 non-null  int64  
 6   Vehicle_Age             78273 non-null  object 
 7   Vehicle_Damage          78273 non-null  object 
 8   Annual_Premium          78273 non-null  float64
 9   Policy_Sales_Channel    78273 non-null  float64
 10  Vintage                 78273 non-null  int64  
dtypes: float64(3), int64(5), object(3)
memory usage: 6.6+ MB
```

In [63]:
```
#Dropping unnecessary columns
damage.drop(['Annual_Premium','Policy_Sales_Channel', 'Vintage', 'Region_Code'], axis=1, inplace=True)
damage
```

Out[63]:

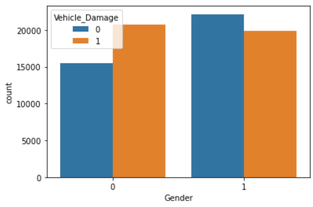
	id	Gender	Age	Driving_License	Previously_Insured	Vehicle_Age	Vehicle_Damage
0	57782	Female	34	1	1	1-2 Year	No
1	286811	Female	55	1	0	> 2 Years	Yes
2	117823	Male	39	1	1	1-2 Year	No
3	213992	Male	28	1	1	1-2 Year	No
4	324756	Female	24	1	0	< 1 Year	Yes
...	...	...	...	...	...	...	...
78268	847	Male	43	1	0	1-2 Year	Yes
78269	417524	Female	21	1	1	< 1 Year	No
78270	188087	Male	48	1	1	1-2 Year	No
78271	215680	Male	64	1	1	1-2 Year	No
78272	138006	Female	25	1	1	< 1 Year	No

78273 rows × 7 columns

In [64]:
```
#Changing categorical values to numerical values
damage['Gender'].replace({'Male':1, 'Female':0}, inplace=True)
damage['Vehicle_Age'].replace({'> 2 Years':1, '1-2 Year':0, '< 1 Year':0}, inplace=True)
damage['Vehicle_Damage'].replace({'No':1, 'Yes':0}, inplace=True)
```

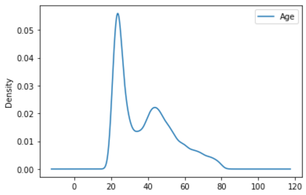
In [65]:
```
#A little bit of visualization
sns.countplot(x ='Gender', data = damage, hue = 'Vehicle_Damage')
```

Out[65]:
```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



In [66]:
```
#One more visual...
damage.plot(kind='kde', x = 'Vehicle_Damage', y = 'Age')
```

Out[66]:
```
<AxesSubplot:ylabel='Density'>
```



In [67]:
```
#Splitting the dataset into features and targets
y = damage[['Vehicle_Damage']]
X = damage[['Vehicle_Age', 'Previously_Insured']]
```

In [68]:
```
#Splitting the dataset into training and test sets in the ratio 70/30
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 10)
```

In [69]:
```
#Creating a Logistic Regression Model
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
```

In [70]:
```
#Training the Linear Regression model using the training data
lr.fit(X_train, y_train)
```

```
C:\Users\hagay\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[70]:
```
LogisticRegression()
```

In [71]:
```
#Make predictions using your test data
y_pred = lr.predict(X_test)
```

In [72]:
```
#Generating the confusion matrix using scikit-learn's confusion matrix method
from sklearn.metrics import confusion_matrix

confusion_matrix(y_test, y_pred)
```

Out[72]:
```
array([[10910,   531],
       [ 1223, 10818]], dtype=int64)
```

In [73]:
```
#Generate Classification Report
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.95      0.93     11441
           1       0.95      0.90      0.93     12041

    accuracy                           0.93     23482
   macro avg       0.93      0.93      0.93     23482
weighted avg       0.93      0.93      0.93     23482
```