

In [6]:



**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**import** warnings

​

**%**matplotlib inline

sns.set()

warnings.simplefilter('ignore')

In [7]:



data **=** pd.read\_csv(r'C:\Users\sumanth\Downloads\Credit-Card-Fraud-Detection-using-Machine-Learning-master\creditcard.csv')

df **=** data.copy() *# To keep the data as backup*

df.head()

Out[7]:

|  | **Time** | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** | **V9** | **...** | **V21** | **V22** | **V23** | **V24** | **V25** | **V26** | **V27** | **V28** | **Amount** | **Class** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0.0 | -1.359807 | -0.072781 | 2.536347 | 1.378155 | -0.338321 | 0.462388 | 0.239599 | 0.098698 | 0.363787 | ... | -0.018307 | 0.277838 | -0.110474 | 0.066928 | 0.128539 | -0.189115 | 0.133558 | -0.021053 | 149.62 | 0 |
| **1** | 0.0 | 1.191857 | 0.266151 | 0.166480 | 0.448154 | 0.060018 | -0.082361 | -0.078803 | 0.085102 | -0.255425 | ... | -0.225775 | -0.638672 | 0.101288 | -0.339846 | 0.167170 | 0.125895 | -0.008983 | 0.014724 | 2.69 | 0 |
| **2** | 1.0 | -1.358354 | -1.340163 | 1.773209 | 0.379780 | -0.503198 | 1.800499 | 0.791461 | 0.247676 | -1.514654 | ... | 0.247998 | 0.771679 | 0.909412 | -0.689281 | -0.327642 | -0.139097 | -0.055353 | -0.059752 | 378.66 | 0 |
| **3** | 1.0 | -0.966272 | -0.185226 | 1.792993 | -0.863291 | -0.010309 | 1.247203 | 0.237609 | 0.377436 | -1.387024 | ... | -0.108300 | 0.005274 | -0.190321 | -1.175575 | 0.647376 | -0.221929 | 0.062723 | 0.061458 | 123.50 | 0 |
| **4** | 2.0 | -1.158233 | 0.877737 | 1.548718 | 0.403034 | -0.407193 | 0.095921 | 0.592941 | -0.270533 | 0.817739 | ... | -0.009431 | 0.798278 | -0.137458 | 0.141267 | -0.206010 | 0.502292 | 0.219422 | 0.215153 | 69.99 | 0 |

5 rows × 31 columns

In [8]:



df.shape

Out[8]:

(284807, 31)

In [9]:



df.isnull().sum()

Out[9]:

Time 0

V1 0

V2 0

V3 0

V4 0

V5 0

V6 0

V7 0

V8 0

V9 0

V10 0

V11 0

V12 0

V13 0

V14 0

V15 0

V16 0

V17 0

V18 0

V19 0

V20 0

V21 0

V22 0

V23 0

V24 0

V25 0

V26 0

V27 0

V28 0

Amount 0

Class 0

dtype: int64

In [10]:



df.dtypes

Out[10]:

Time float64

V1 float64

V2 float64

V3 float64

V4 float64

V5 float64

V6 float64

V7 float64

V8 float64

V9 float64

V10 float64

V11 float64

V12 float64

V13 float64

V14 float64

V15 float64

V16 float64

V17 float64

V18 float64

V19 float64

V20 float64

V21 float64

V22 float64

V23 float64

V24 float64

V25 float64

V26 float64

V27 float64

V28 float64

Amount float64

Class int64

dtype: object

In [11]:



df.Time.tail(15)

Out[11]:

284792 172774.0

284793 172775.0

284794 172777.0

284795 172778.0

284796 172780.0

284797 172782.0

284798 172782.0

284799 172783.0

284800 172784.0

284801 172785.0

284802 172786.0

284803 172787.0

284804 172788.0

284805 172788.0

284806 172792.0

Name: Time, dtype: float64

In [12]:



df.describe()

Out[12]:

|  | **Time** | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** | **V9** | **...** | **V21** | **V22** | **V23** | **V24** | **V25** | **V26** | **V27** | **V28** | **Amount** | **Class** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 284807.000000 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | ... | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 284807.000000 | 284807.000000 |
| **mean** | 94813.859575 | 3.919560e-15 | 5.688174e-16 | -8.769071e-15 | 2.782312e-15 | -1.552563e-15 | 2.010663e-15 | -1.694249e-15 | -1.927028e-16 | -3.137024e-15 | ... | 1.537294e-16 | 7.959909e-16 | 5.367590e-16 | 4.458112e-15 | 1.453003e-15 | 1.699104e-15 | -3.660161e-16 | -1.206049e-16 | 88.349619 | 0.001727 |
| **std** | 47488.145955 | 1.958696e+00 | 1.651309e+00 | 1.516255e+00 | 1.415869e+00 | 1.380247e+00 | 1.332271e+00 | 1.237094e+00 | 1.194353e+00 | 1.098632e+00 | ... | 7.345240e-01 | 7.257016e-01 | 6.244603e-01 | 6.056471e-01 | 5.212781e-01 | 4.822270e-01 | 4.036325e-01 | 3.300833e-01 | 250.120109 | 0.041527 |
| **min** | 0.000000 | -5.640751e+01 | -7.271573e+01 | -4.832559e+01 | -5.683171e+00 | -1.137433e+02 | -2.616051e+01 | -4.355724e+01 | -7.321672e+01 | -1.343407e+01 | ... | -3.483038e+01 | -1.093314e+01 | -4.480774e+01 | -2.836627e+00 | -1.029540e+01 | -2.604551e+00 | -2.256568e+01 | -1.543008e+01 | 0.000000 | 0.000000 |
| **25%** | 54201.500000 | -9.203734e-01 | -5.985499e-01 | -8.903648e-01 | -8.486401e-01 | -6.915971e-01 | -7.682956e-01 | -5.540759e-01 | -2.086297e-01 | -6.430976e-01 | ... | -2.283949e-01 | -5.423504e-01 | -1.618463e-01 | -3.545861e-01 | -3.171451e-01 | -3.269839e-01 | -7.083953e-02 | -5.295979e-02 | 5.600000 | 0.000000 |
| **50%** | 84692.000000 | 1.810880e-02 | 6.548556e-02 | 1.798463e-01 | -1.984653e-02 | -5.433583e-02 | -2.741871e-01 | 4.010308e-02 | 2.235804e-02 | -5.142873e-02 | ... | -2.945017e-02 | 6.781943e-03 | -1.119293e-02 | 4.097606e-02 | 1.659350e-02 | -5.213911e-02 | 1.342146e-03 | 1.124383e-02 | 22.000000 | 0.000000 |
| **75%** | 139320.500000 | 1.315642e+00 | 8.037239e-01 | 1.027196e+00 | 7.433413e-01 | 6.119264e-01 | 3.985649e-01 | 5.704361e-01 | 3.273459e-01 | 5.971390e-01 | ... | 1.863772e-01 | 5.285536e-01 | 1.476421e-01 | 4.395266e-01 | 3.507156e-01 | 2.409522e-01 | 9.104512e-02 | 7.827995e-02 | 77.165000 | 0.000000 |
| **max** | 172792.000000 | 2.454930e+00 | 2.205773e+01 | 9.382558e+00 | 1.687534e+01 | 3.480167e+01 | 7.330163e+01 | 1.205895e+02 | 2.000721e+01 | 1.559499e+01 | ... | 2.720284e+01 | 1.050309e+01 | 2.252841e+01 | 4.584549e+00 | 7.519589e+00 | 3.517346e+00 | 3.161220e+01 | 3.384781e+01 | 25691.160000 | 1.000000 |

8 rows × 31 columns

**Checking the frequency of frauds before moving forward**

In [13]:



df.Class.value\_counts()

Out[13]:

0 284315

1 492

Name: Class, dtype: int64

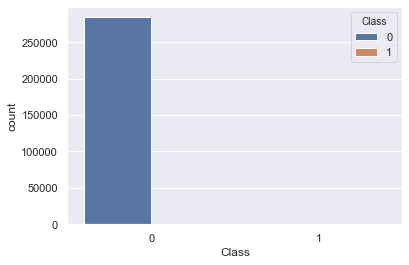
In [14]:



sns.countplot(x**=**df.Class, hue**=**df.Class)

Out[14]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1cff769a248>



**Checking the distribution of amount**

In [15]:

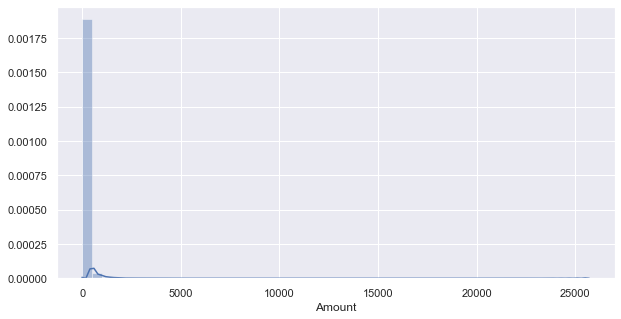


plt.figure(figsize**=**(10, 5))

sns.distplot(df.Amount)

Out[15]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1cf817fd308>



In [16]:



df['Amount-Bins'] **=** ''

Since, it is a little difficult to see. Let's engineer a new feature of bins.

In [17]:



**def** make\_bins(predictor, size**=**50):

'''

Takes the predictor (a series or a dataframe of single predictor) and size of bins

Returns bins and bin labels

'''

bins **=** np.linspace(predictor.min(), predictor.max(), num**=**size)

​

bin\_labels **=** []

​

*# Index of the final element in bins list*

bins\_last\_index **=** bins.shape[0] **-** 1

​

**for** id, val **in** enumerate(bins):

**if** id **==** bins\_last\_index:

**continue**

val\_to\_put **=** str(int(bins[id])) **+** ' to ' **+** str(int(bins[id **+** 1]))

bin\_labels.append(val\_to\_put)

**return** bins, bin\_labels

In [18]:



bins, bin\_labels **=** make\_bins(df.Amount, size**=**10)

Now, adding bins in the column Amount-Bins.

In [19]:



df['Amount-Bins'] **=** pd.cut(df.Amount, bins**=**bins,

labels**=**bin\_labels, include\_lowest**=True**)

df['Amount-Bins'].head().to\_frame()

Out[19]:

|  | **Amount-Bins** |
| --- | --- |
| **0** | 0 to 2854 |
| **1** | 0 to 2854 |
| **2** | 0 to 2854 |
| **3** | 0 to 2854 |
| **4** | 0 to 2854 |

Let's plot the bins.

In [20]:



df['Amount-Bins'].value\_counts()

Out[20]:

0 to 2854 284484

2854 to 5709 285

5709 to 8563 28

8563 to 11418 4

11418 to 14272 3

17127 to 19982 2

22836 to 25691 1

19982 to 22836 0

14272 to 17127 0

Name: Amount-Bins, dtype: int64

In [21]:



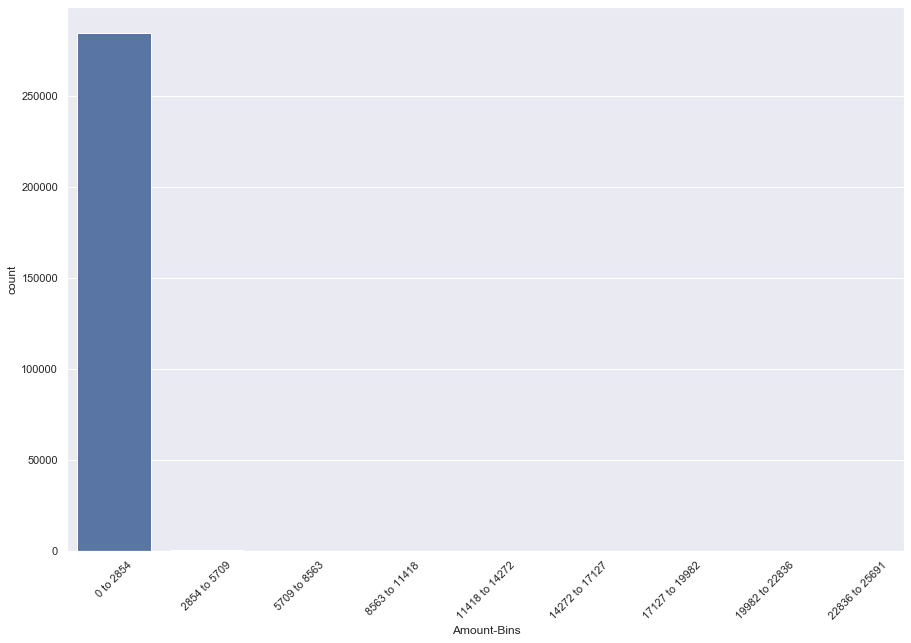
plt.figure(figsize**=**(15, 10))

sns.countplot(x**=**'Amount-Bins', data**=**df)

plt.xticks(rotation**=**45)

Out[21]:

(array([0, 1, 2, 3, 4, 5, 6, 7, 8]), <a list of 9 Text xticklabel objects>)



Since, count of values of Bins other than '0 to 2854' are difficult to view. Let's not insert the first one.

In [22]:



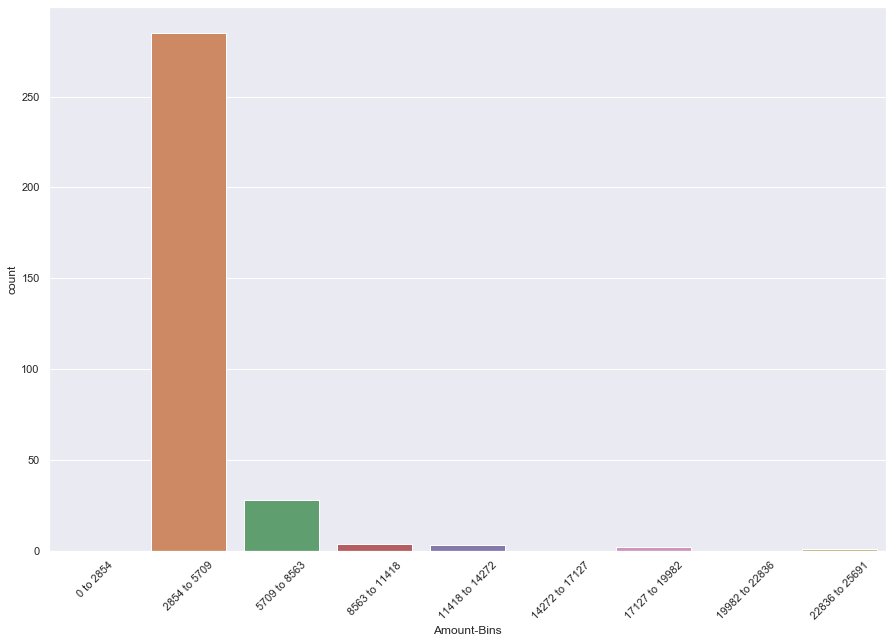
plt.figure(figsize**=**(15, 10))

sns.countplot(x**=**'Amount-Bins', data**=**df[**~**(df['Amount-Bins'] **==** '0 to 2854')])

plt.xticks(rotation**=**45)

Out[22]:

(array([0, 1, 2, 3, 4, 5, 6, 7, 8]), <a list of 9 Text xticklabel objects>)



We can see that mostly the amount is between 0 and 2854 euros.