

Import reuired Packages

```
In [7]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Read Datasets

```
In [8]: df=pd.read_csv("eeg-eye-state_csv.csv")
```

Print the datasets

```
In [9]: df.head(10)
```

Out[9]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
0	4329.23	4009.23	4289.23	4148.21	4350.26	4586.15	4096.92	4641.03	4222.05	4238.46	4211.2
1	4324.62	4004.62	4293.85	4148.72	4342.05	4586.67	4097.44	4638.97	4210.77	4226.67	4207.6
2	4327.69	4006.67	4295.38	4156.41	4336.92	4583.59	4096.92	4630.26	4207.69	4222.05	4206.6
3	4328.72	4011.79	4296.41	4155.90	4343.59	4582.56	4097.44	4630.77	4217.44	4235.38	4210.7
4	4326.15	4011.79	4292.31	4151.28	4347.69	4586.67	4095.90	4627.69	4210.77	4244.10	4212.8
5	4321.03	4004.62	4284.10	4153.33	4345.64	4587.18	4093.33	4616.92	4202.56	4232.82	4209.7
6	4319.49	4001.03	4280.51	4151.79	4343.59	4584.62	4089.74	4615.90	4212.31	4226.67	4201.0
7	4325.64	4006.67	4278.46	4143.08	4344.10	4583.08	4087.18	4614.87	4205.64	4230.26	4195.9
8	4326.15	4010.77	4276.41	4139.49	4345.13	4584.10	4091.28	4608.21	4187.69	4229.74	4202.0
9	4326.15	4011.28	4276.92	4142.05	4344.10	4582.56	4092.82	4608.72	4194.36	4228.72	4212.8

Print the shape of Datasets

```
In [4]: df.shape
```

Out[4]: (14980, 15)

Here we can see total 14980 data instances are there in the dataset. It is also having 15 cloumns, in which one column is of target variable(Label/class) and remaining 14 are the features.

Print Unique Classes

```
In [12]: df.V1.unique()
```

```
Out[12]: array([ 4329.23,  4324.62,  4327.69,  4328.72,  4326.15,  4321.03,
 4319.49,  4325.64,  4316.92,  4308.21,  4312.82,  4313.85,
 4306.67,  4303.59,  4312.31,  4315.9 ,  4314.36,  4331.28,
 4326.67,  4321.54,  4317.44,  4311.79,  4308.72,  4311.28,
 4306.15,  4307.18,  4314.87,  4297.44,  4301.03,  4305.64,
 4296.92,  4298.97,  4302.56,  4315.38,  4317.95,  4316.41,
 4305.13,  4293.33,  4335.9 ,  4299.49,  4296.41,  4313.33,
 4320.  ,  4328.21,  4325.13,  4309.74,  4300.  ,  4292.82,
 4281.54,  4288.21,  4301.54,  4295.38,  4318.97,  4323.08,
 4318.46,  4309.23,  4304.62,  4300.51,  4298.46,  4297.95,
 4287.18,  4304.1 ,  4286.15,  4291.79,  4289.23,  4286.67,
 4293.85,  4289.74,  4295.9 ,  4303.08,  4351.79,  4375.38,
 4401.54,  4412.31,  4416.92,  4430.26,  4447.18,  4454.36,
 4458.46,  4472.31,  4480.  ,  4474.87,  4477.44,  4484.62,
 4483.59,  4487.69,  4490.77,  4481.54,  4482.56,  4477.95,
 4475.9 ,  4471.28,  4484.1 ,  4493.85,  4479.49,  4463.59,
 4461.03,  4457.44,  4450.77,  4448.72,  4454.87,  4465.13,
 4468.21,  4452.82,  4445.13,  4438.46,  4432.31,  4427.18,
 4426.67,  4427.69,  4419.49,  4412.82,  4411.28,  4403.59,
 4395.9 ,  4393.33,  4388.72,  4379.49,  4371.79,  4366.15,
 4362.56,  4363.08,  4366.67,  4361.54,  4348.21,  4347.69,
 4354.87,  4350.26,  4348.72,  4357.44,  4343.59,  4345.13,
 4340.51,  4322.56,  4288.72,  4294.87,  4307.69,  4292.31,
 4290.26,  4285.64,  4284.62,  4280.  ,  4283.59,  4284.1 ,
 4291.28,  4278.46,  4275.38,  4279.49,  4280.51,  4272.82,
 4277.95,  4276.92,  4282.56,  4338.46,  4344.1 ,  4341.54,
 4344.62,  4355.38,  4358.46,  4356.41,  4365.13,  4372.31,
 4367.69,  4362.05,  4356.92,  4338.97,  4337.44,  4354.36,
 4347.18,  4343.08,  4346.67,  4335.38,  4329.74,  4330.77,
 4337.95,  4342.05,  4332.82,  4332.31,  4333.85,  4360.  ,
 4351.28,  4352.31,  4361.03,  4349.74,  4359.49,  4367.18,
 4341.03,  4336.92,  4336.41,  4334.87,  4330.26,  4327.18,
 4331.79,  4323.59,  4322.05,  4324.1 ,  4310.77,  4294.36,
 4287.69,  4290.77,  4285.13,  4283.08,  4274.36,  4270.77,
 4281.03,  4278.97,  4264.1 ,  4256.41,  4270.26,  4282.05,
 4275.9 ,  4277.44,  4269.23,  4274.87,  4276.41,  4268.21,
 4271.28,  4269.74,  4267.69,  4271.79,  4265.13,  4339.49,
 4333.33,  4310.26,  4320.51,  4334.36,  4353.85,  4355.9 ,
 4372.82,  4369.23,  4360.51,  4358.97,  4349.23,  4352.82,
 4266.67,  4257.44,  4261.54,  4264.62,  4262.05,  4263.08,
 4254.36,  4246.67,  4249.23,  4253.33,  4253.85,  4256.92,
 4259.49,  4257.95,  4260.  ,  4255.38,  4258.46,  4250.26,
 4258.97,  4261.03,  4255.9 ,  4252.31,  7222.05,  4272.31,
 4266.15,  4273.85,  4262.56,  4273.33,  4248.21,  4242.05,
 4243.59,  4245.13,  4244.62,  4254.87,  4250.77,  4247.69,
 4247.18,  4246.15,  4252.82,  4251.79,  4302.05,  4388.21,
 4400.51,  4413.85,  4429.23,  4456.41,  4464.1 ,  4462.05,
 4460.51,  4455.9 ,  4461.54,  4470.26,  4473.85,  4476.41,
 4465.64,  4455.38,  4457.95,  4458.97,  4450.26,  4444.62,
 4446.67,  4449.74,  4443.08,  4431.79,  4420.  ,  4402.56,
 4386.67,  4378.97,  4376.41,  4374.36,  4369.74,  4364.62,
 4409.74,  4410.26,  4398.46,  4390.77,  4389.74,  4383.59,
 4378.46,  4240.51,  4239.49,  4242.56,  4248.72,  4244.1 ,
 4241.54,  4240.  ,  4263.59,  4267.18,  4265.64,  4373.33,
```

```

4374.87, 4370.77, 4346.15, 4340. , 4380. , 4389.23,
4407.18, 4396.41, 4409.23, 4429.74, 4435.9 , 4435.38,
4428.72, 4406.15, 4411.79, 4424.1 , 4451.79, 4422.05,
4420.51, 4392.31, 4387.18, 4415.9 , 4408.72, 4399.49,
4394.36, 4382.05, 4357.95, 4363.59, 4365.64, 4345.64,
4268.72, 4243.08, 4342.56, 4350.77, 4237.44, 4235.38,
4230.26, 4221.54, 4214.36, 4213.33, 4207.69, 4198.97,
4203.59, 4212.31, 4206.67, 4202.56, 4208.72, 4208.21,
4205.13, 4211.79, 4210.26, 4210.77, 4220. , 4211.28,
4209.74, 4213.85, 4217.95, 4215.9 , 4214.87, 4225.13,
4227.69, 4223.08, 4216.92, 4218.46, 4215.38, 4218.97,
4216.41, 4217.44, 4220.51, 4232.31, 4229.23, 4227.18,
4231.28, 4229.74, 4225.64, 4224.1 , 4234.36, 4232.82,
4238.97, 4245.64, 4236.41, 4241.03, 4235.9 , 4249.74,
4251.28, 4260.51, 4370.26, 4413.33, 4418.97, 4424.62,
4425.64, 4381.03, 4376.92, 4371.28, 4373.85, 4391.79,
4397.95, 4416.41, 4437.44, 4456.92, 4453.85, 4451.28,
4441.54, 4445.64, 4448.21, 4446.15, 4423.59, 4426.15,
4428.21, 4432.82, 4434.36, 4430.77, 4415.38, 4408.21,
4403.08, 4400. , 4395.38, 4397.44, 4404.62, 4396.92,
4393.85, 4384.1 , 4404.1 , 4405.64, 4394.87, 4390.26,
4383.08, 4382.56, 4384.62, 4386.15, 4381.54, 4380.51,
4377.95, 4375.9 , 4364.1 , 4368.21, 4398.97, 4406.67,
4402.05, 4392.82, 4233.85, 4238.46, 4234.87, 4221.03,
4233.33, 4237.95, 4385.64, 4421.03, 4421.54, 4417.95,
4442.05, 4440. , 4439.49, 4447.69, 4443.59, 4440.51,
4441.03, 4431.28, 4417.44, 4423.08, 4410.77, 4436.41,
4418.46, 4422.56, 4387.69, 4353.33, 4223.59, 4204.62,
4209.23, 4212.82, 4201.54, 4198.46, 4200. , 4207.18,
4204.1 , 4219.49, 4197.95, 4199.49, 4202.05, 4222.05,
4231.79, 4228.72, 4224.62, 4230.77, 4222.56, 4226.67,
4405.13, 4414.36, 4407.69, 4414.87, 4385.13, 4228.21,
4236.92, 4368.72, 4391.28, 7398.46, 4401.03, 4377.44,
4433.85, 4425.13, 309231. , 4438.97, 4453.33, 4459.49,
4466.15, 4482.05, 4488.72, 4499.49, 4504.1 , 4497.44,
4493.33, 4492.31, 4490.26, 4497.95, 4501.03, 4491.79,
4475.38, 4444.1 , 4433.33, 4434.87, 1030.77, 4437.95,
4226.15, 4442.56])

```

Count instances each class having

```
In [13]: df.V1.value_counts()
```

```
Out[13]: 4291.79    165
         4287.69    162
         4295.90    161
         4292.31    160
         4291.28    157
         ...
         4452.82     1
         4204.10     1
         4456.41     1
         4482.05     1
         4492.31     1
         Name: V1, Length: 548, dtype: int64
```

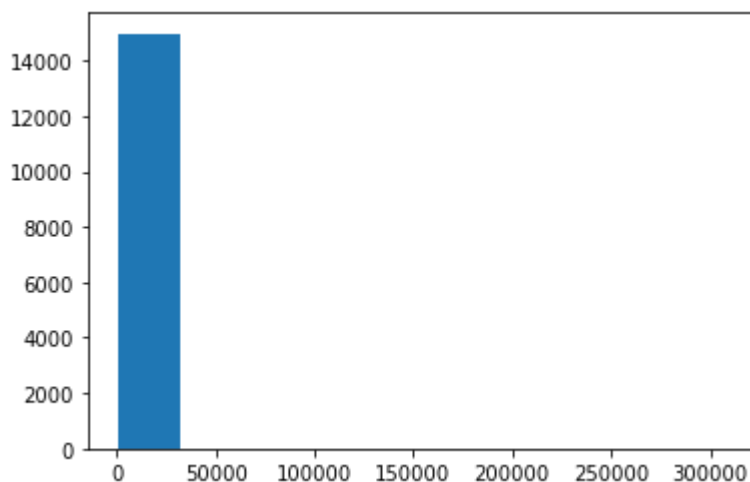
Print Percentage of each class

```
In [22]: round ((df.V1.value_counts()/df.shape[0])*100,2)
```

```
Out[22]: 4291.79    1.10
         4287.69    1.08
         4295.90    1.07
         4292.31    1.07
         4291.28    1.05
         ...
         4452.82    0.01
         4204.10    0.01
         4456.41    0.01
         4482.05    0.01
         4492.31    0.01
         Name: V1, Length: 548, dtype: float64
```

Plot Histogram

```
In [18]: plot =plt.hist(df['V1'])
```



This Datasets having 14980 instances of class V1 which is 1.10% of total instances.

Print datatypes of each column

```
In [23]: df.dtypes
```

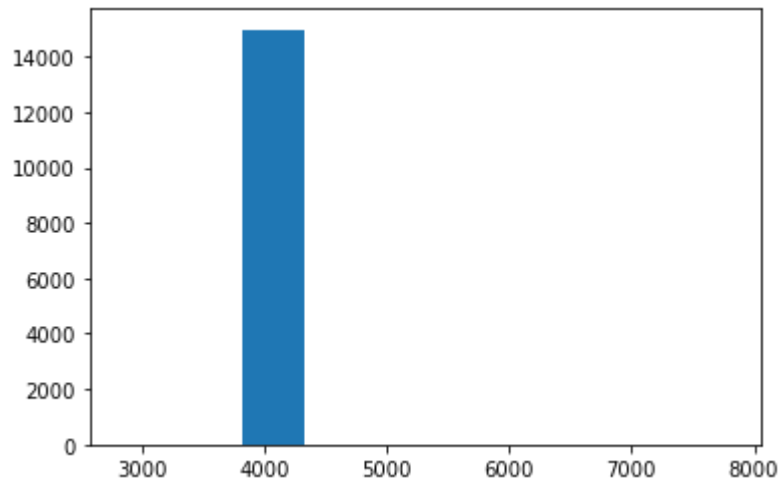
```
Out[23]: V1          float64  
V2          float64  
V3          float64  
V4          float64  
V5          float64  
V6          float64  
V7          float64  
V8          float64  
V9          float64  
V10         float64  
V11         float64  
V12         float64  
V13         float64  
V14         float64  
Class       int64  
dtype: object
```

Here we can see class having datatype as int54. Then V1,V2,V3,V4,V5,V6,V7,V8,V9.V10...etc have floar64 values

Plot hist for different features

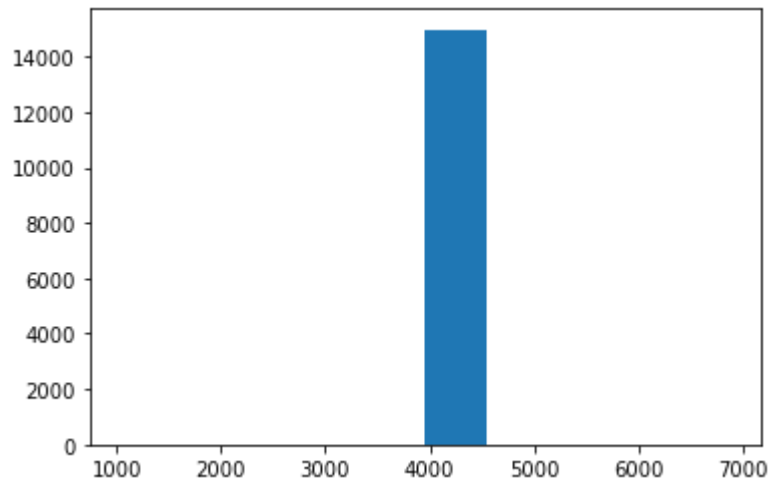
```
In [25]: print(df.V2.describe())  
  
plot = plt.hist(df['V2'])
```

```
count    14980.000000  
mean      4009.767694  
std        45.941672  
min       2830.770000  
25%       3990.770000  
50%       4005.640000  
75%       4023.080000  
max       7804.620000  
Name: V2, dtype: float64
```



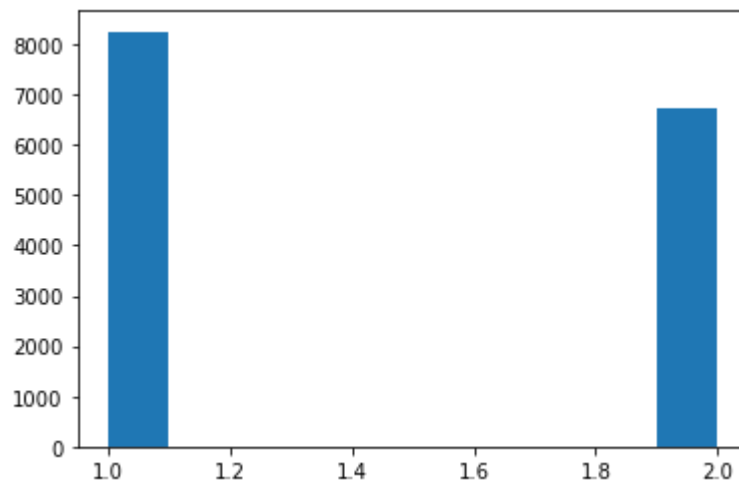
```
In [26]: print(df.V3.describe())  
  
plot = plt.hist(df['V3'])
```

```
count    14980.000000  
mean      4264.022433  
std        44.428052  
min       1040.000000  
25%       4250.260000  
50%       4262.560000  
75%       4270.770000  
max       6880.510000  
Name: V3, dtype: float64
```



```
In [28]: print(df.Class.describe())  
  
plot = plt.hist(df['Class'])
```

```
count    14980.000000  
mean       1.448798  
std        0.497388  
min        1.000000  
25%        1.000000  
50%        1.000000  
75%        2.000000  
max        2.000000  
Name: Class, dtype: float64
```



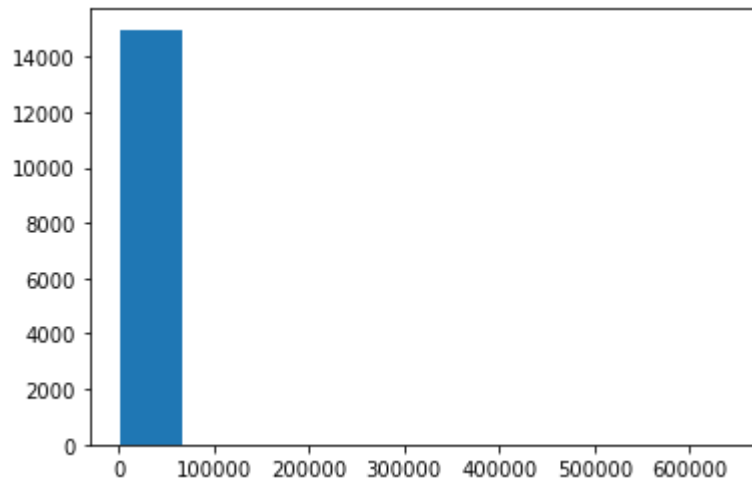
```
In [29]: df.V2.equals(df.V3)
```

```
Out[29]: False
```

Here V2 and V3 not having same values and it is not equally distributed over the range.

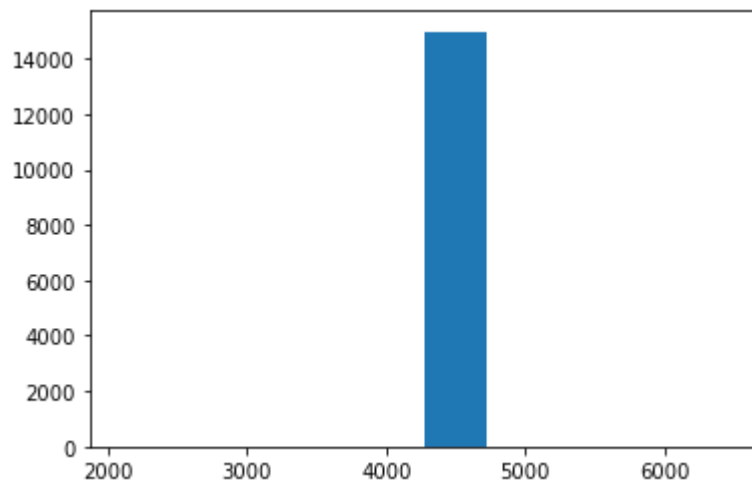

```
In [30]: print(df.V4.describe())  
  
plot = plt.hist(df['V4'])
```

```
count      14980.000000  
mean       4164.946326  
std        5216.404632  
min        2453.330000  
25%        4108.210000  
50%        4120.510000  
75%        4132.310000  
max        642564.000000  
Name: V4, dtype: float64
```



```
In [32]: print(df.V5.describe())  
  
plot = plt.hist(df['V5'])
```

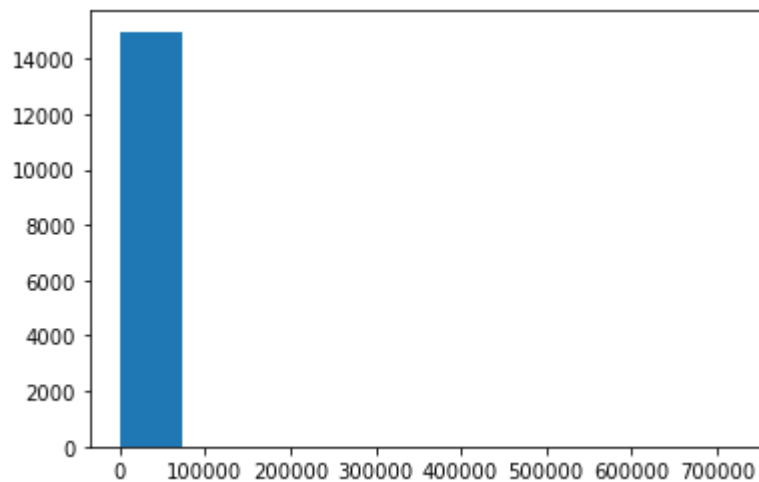
```
count    14980.000000  
mean      4341.741075  
std        34.738821  
min       2089.740000  
25%       4331.790000  
50%       4338.970000  
75%       4347.180000  
max       6474.360000  
Name: V5, dtype: float64
```



In V4 and V5, max values are different and also means are different. From plots of both we can conclude there distribution is also almost same

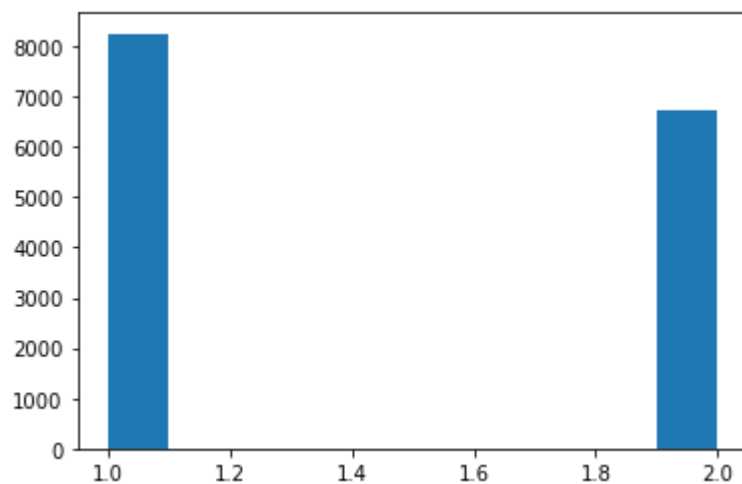
```
In [35]: print(df.V14.describe())  
  
plot = plt.hist(df['V14'])
```

```
count      14980.000000  
mean       4416.435832  
std        5891.285043  
min        1366.150000  
25%        4342.050000  
50%        4354.870000  
75%        4372.820000  
max        715897.000000  
Name: V14, dtype: float64
```



```
In [36]: print(df.Class.describe())  
  
plot = plt.hist(df['Class'])
```

```
count    14980.000000  
mean       1.448798  
std        0.497388  
min        1.000000  
25%        1.000000  
50%        1.000000  
75%        2.000000  
max        2.000000  
Name: Class, dtype: float64
```



Here values from 1.0 to 2.0 having frequency of atleast 90.