In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
a=pd.read_csv(r"C:\All Datasets\bangalore-cas-alerts.csv")
a
```

Out[2]:

| e_location_longitude | deviceCode_location_wardName | deviceCode_pyld_alarmType | deviceCode_pyld_sp |
|---|---|---|---|
| 77.744087 | Kadugodi | PCW | |
| 77.744087 | Kadugodi | PCW | |
| 77.741119 | Garudachar Playa | FCW | |
| 77.741119 | Garudachar Playa | FCW | |
| 77.740051 | Hudi | Overspeed | |
| ... | ... | ... | |
| 77.741516 | Kadugodi | UFCW | |
| 77.745117 | Kadugodi | UFCW | |
| 77.749886 | Hagadur | Overspeed | |
| 77.746841 | Hagadur | FCW | |
| 77.744125 | Hagadur | PCW | |

In [3]:
```
a.isnull().sum()
```

Out[3]:
```
deviceCode_deviceCode                  0
deviceCode_location_latitude           0
deviceCode_location_longitude          0
deviceCode_location_wardName           0
deviceCode_pyld_alarmType              0
deviceCode_pyld_speed                  0
deviceCode_time_recordedTime_$date     0
dtype: int64
```

In [4]:
```
a["deviceCode_pyld_alarmType"].value_counts()
```

Out[4]:
```
UFCW         82425
HMW          36143
FCW          35000
Overspeed    27440
PCW          24003
LDWL          1412
LDWR          1194
Name: deviceCode_pyld_alarmType, dtype: int64
```

In [5]:

```
from sklearn.preprocessing import LabelEncoder
l=LabelEncoder()
a["deviceCode_location_wardName"]=l.fit_transform(a["deviceCode_location_wardName
a["deviceCode_pyld_alarmType"]=l.fit_transform(a["deviceCode_pyld_alarmType"])
```

In [6]:

```
a
```

Out[6]:

| | deviceCode_deviceCode | deviceCode_location_latitude | deviceCode_location_longitude | devic |
|---|---|---|---|---|
| **0** | 864504031502210 | 12.984595 | 77.744087 | |
| **1** | 864504031502210 | 12.984595 | 77.744087 | |
| **2** | 864504031502210 | 12.987233 | 77.741119 | |
| **3** | 864504031502210 | 12.987233 | 77.741119 | |
| **4** | 864504031502210 | 12.987503 | 77.740051 | |
| **...** | ... | ... | ... | |
| **207612** | 864504031432707 | 12.976435 | 77.741516 | |
| **207613** | 864504031035658 | 12.986425 | 77.745117 | |
| **207614** | 863977033828919 | 12.969396 | 77.749886 | |
| **207615** | 863977033715231 | 12.974123 | 77.746841 | |
| **207616** | 863977033827523 | 12.975480 | 77.744125 | |

207617 rows × 7 columns

In [7]:

```
del a["deviceCode_time_recordedTime_$date"]
```

In [8]:
```
a
```

Out[8]:

| | deviceCode_deviceCode | deviceCode_location_latitude | deviceCode_location_longitude | devic |
|---|---|---|---|---|
| **0** | 864504031502210 | 12.984595 | 77.744087 | |
| **1** | 864504031502210 | 12.984595 | 77.744087 | |
| **2** | 864504031502210 | 12.987233 | 77.741119 | |
| **3** | 864504031502210 | 12.987233 | 77.741119 | |
| **4** | 864504031502210 | 12.987503 | 77.740051 | |
| **...** | ... | ... | ... | |
| **207612** | 864504031432707 | 12.976435 | 77.741516 | |
| **207613** | 864504031035658 | 12.986425 | 77.745117 | |
| **207614** | 863977033828919 | 12.969396 | 77.749886 | |
| **207615** | 863977033715231 | 12.974123 | 77.746841 | |
| **207616** | 863977033827523 | 12.975480 | 77.744125 | |

207617 rows × 6 columns

In [9]:
```
x=a.iloc[:,[0,1,2,3,5]].values
y=a.iloc[:,-2].values
```

In [10]:
```
from sklearn.preprocessing import StandardScaler
s=StandardScaler()
x1=s.fit_transform(x)
```

In [11]:
```
from imblearn.over_sampling import SMOTE
sm=SMOTE()
X,Y=sm.fit_resample(x1,y)
```

In [12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=40,test_size=0.2)
```

In [13]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

In [14]:
```
kn=KNeighborsClassifier(n_neighbors=5,metric="minkowski",p=2)
dt=DecisionTreeClassifier(criterion="entropy")
rf=RandomForestClassifier(n_estimators=5)
```

In [15]:

```
from sklearn.ensemble import VotingClassifier
vc=VotingClassifier(estimators=[("knn",kn),("Dec.tree",dt),("RandomForest",rf)])
vc.fit(x_train,y_train)
```

Out[15]:

```
VotingClassifier(estimators=[('knn', KNeighborsClassifier()),
                             ('Dec.tree',
                              DecisionTreeClassifier(criterion='entropy')),
                             ('RandomForest',
                              RandomForestClassifier(n_estimators=5))])
```

In [16]:

```
pred=vc.predict(x_test)
```

In [17]:

```
pred
```

Out[17]:

```
array([0, 5, 0, ..., 3, 5, 1])
```

In [18]:

```
from sklearn.metrics import accuracy_score
accuracy_score(pred,y_test)*100
```

Out[18]:

```
87.02456778889899
```

In [20]:

```
from sklearn.model_selection import StratifiedKFold
st=StratifiedKFold(n_splits=5,random_state=30,shuffle=True)
st.get_n_splits(x_train,y_train)
```

Out[20]:

```
5
```

In [21]:

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import accuracy_score
```

In [22]:

```
scores=cross_val_score(vc,x_train,y_train)
pred2=cross_val_predict(vc,x_test,y_test)
print((scores)*100)
print(pred2)
accuracy_score(pred2,y_test)*100
```

```
[85.12933836 85.233329   85.34923524 85.2484943  85.3178214 ]
[1 6 1 ... 3 5 2]
```

Out[22]:

```
74.9789852246631
```

In [ ]: