

# Name : SHEIKPAREETH

## Machine Learning, Supervised, classification, Naive baye's

In [1]:

```
#Import the libraries and put nicknames
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:

```
#Reading the dataset
dataset=pd.read_csv("fraud.csv")
```

In [3]:

```
#It have 16426 rows and 9 columns
dataset
```

Out[3]:

	step	type	amount	oldbalanceOrig	newbalanceOrig	oldbalanceDest	newbalar
	0	1	PAYMENT	9839.64	170136.00	160296.36	0.00
	1	1	PAYMENT	1864.28	21249.00	19384.72	0.00
	2	1	PAYMENT	11668.14	41554.00	29885.86	0.00
	3	1	PAYMENT	7817.71	53860.00	46042.29	0.00
	4	1	PAYMENT	7107.77	183195.00	176087.23	0.00
	...	...	...	...	...	...	...
	16421	743	CASH_OUT	339682.13	339682.13	0.00	0.00
	16422	743	TRANSFER	6311409.28	6311409.28	0.00	0.00
	16423	743	CASH_OUT	6311409.28	6311409.28	0.00	68488.84
	16424	743	TRANSFER	850002.52	850002.52	0.00	0.00
	16425	743	CASH_OUT	850002.52	850002.52	0.00	6510099.11

16426 rows × 9 columns

In [4]:

```
#Above the value is categorical value so i have used "one hot encoding method" (we cannot a
#I have removed the duplcates or dummy value
dataset=pd.get_dummies(dataset,drop_first=True)
```

In [ ]:

```
#provide input columns name for the easily without seeing the table  
dataset.columns
```

In [5]:

```
#Put the input and aouput column name  
indep=dataset[["step", "amount", "oldbalanceOrg", "newbalanceOrig", "oldbalanceDest", "newb  
dep=dataset[["isFlaggedFraud"]]
```

In [ ]:

```
#output values (dependent)  
dep
```

In [ ]:

```
#Input values (independent)  
indep
```

In [6]:

```
#split inti training set and test test  
#take 30% of sample  
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(indep,dep,test_size=0.3,random_state=0)
```

In [7]:

```

#model creation process for naive bayes classification
#Gaussian Naive Bayes
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)

```

	precision	recall	f1-score	support
0	1.00	0.46	0.63	4926
1	0.00	1.00	0.00	2
accuracy			0.46	4928
macro avg	0.50	0.73	0.32	4928
weighted avg	1.00	0.46	0.63	4928

```

[[2288 2638]
 [ 0 2]]

```

```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:
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().
y = column_or_1d(y, warn=True)

```

In [8]:

```
#Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:  
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().

```
y = column_or_1d(y, warn=True)
```

	precision	recall	f1-score	support
0	1.00	0.44	0.61	4926
1	0.00	1.00	0.00	2
accuracy			0.44	4928
macro avg	0.50	0.72	0.30	4928
weighted avg	1.00	0.44	0.61	4928

```
[[2149 2777]
 [ 0 2]]
```

In [9]:

```
#Bernoulli Naive Bayes
from sklearn.naive_bayes import BernoulliNB
classifier = BernoulliNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4926
1	0.00	0.00	0.00	2
accuracy			1.00	4928
macro avg	0.50	0.50	0.50	4928
weighted avg	1.00	1.00	1.00	4928

```
[[4926  0]
 [  2  0]]
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:  
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().

```
y = column_or_1d(y, warn=True)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1437: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.  
'precision', 'predicted', average, warn\_for)

In [11]:

```
#Complement Naive Bayes
from sklearn.naive_bayes import ComplementNB
classifier =ComplementNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

	precision	recall	f1-score	support
0	1.00	0.44	0.61	4926
1	0.00	1.00	0.00	2
accuracy			0.44	4928
macro avg	0.50	0.72	0.30	4928
weighted avg	1.00	0.44	0.61	4928

```
[[2148 2778]
 [  0    2]]
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:  
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().

```
y = column_or_1d(y, warn=True)
```