# Name: SHEIKPAREETH

# Machine Learning, Supervised, classification, Random forest

# 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	oldbalanceOrg	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	33		
16422	743	TRANSFER	6311409.28	6311409.28	0.00	0.00			
16423	743	CASH_OUT	6311409.28	6311409.28	0.00	68488.84	637		
16424	743	TRANSFER	850002.52	850002.52	0.00	0.00			
16425	743	CASH_OUT	850002.52	850002.52	0.00	6510099.11	736		
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 account column name
indep=dataset[["step", "amount", "oldbalanceOrg", "newbalanceOrig", "oldbalanceDest", "newbalanceOrig", "newbalanceOrig", "oldbalanceDest", "newbalanceOrig", "newbala
```

#### 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 classification
#formul loaded this libraries and use fit method substitute the value
#finally create a model
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state
classifier.fit(X_train,y_train)
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\_launcher.py:6: 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().

#### Out[7]:

## In [8]:

```
#Evaluation metrics to use test set
#y_test output of prdicted value
y_pred=classifier.predict(X_test)
```

# In [9]:

```
#Calculate confusion matrix to evaluate the accuracy of a classification
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

# In [10]:

```
print(cm)

[[4926 0]
  [ 1 1]]
```

# In [11]:

```
#find clssification report (precision, recall, f1-score, accuracy )
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
```

#### In [12]:

# print(clf\_report)

	precision	recall	f1-score	support
0	1 00	1 00	1 00	4026
0	1.00	1.00	1.00	4926
1	1.00	0.50	0.67	2
accuracy			1.00	4928
macro avg	1.00	0.75	0.83	4928
weighted avg	1.00	1.00	1.00	4928