In [1]:

```
import pandas as pd
import datetime as dt
from datetime import datetime
import numpy as np

from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
import sys

BLACK, RED, GREEN, YELLOW, BLUE, MAGENTA, CYAN, WHITE = range(8)
```

In [2]:

```
myData = pd.read_excel('dataclean.xlsx')
```

In [3]:

```
myData.head()
```

Out[3]:

	insuranceID	fraud_claim_indicator	fraud_claim_reason	holder_ID	holder_name	hol
0	20322	1	expired policy	58765	Marianne	Brid
1	46374	1	inflated figure	72606	Ernest	Lore
2	46374	1	inflated figure	34665	Ernest	Lore
3	46374	1	inflated figure	57964	Yvonne	Reg
4	46374	1	inflated figure	13853	Frances	Sha

5 rows × 26 columns

PREDICTIONS BY AMOUNTS

In [4]:

```
methods = ['Decision Trees']
```

In [5]:

```
sumCld = myData[['sum_insured','claim_amount_paid','total_premium']]

fcI = myData[['fraud_claim_indicator']]
```

In [6]:

```
sumCld_fcI    = DecisionTreeClassifier()
```

In [7]:

```
sumCld_fcI    = sumCld_fcI.fit(sumCld,fcI)
```

```
In [12]:  sumCld_fcI_prediction    = sumCld_fcI.predict([700000,555000,15000])
```

```
C:\Anaconda\lib\site-packages\sklearn\utils\validation.py:395: DeprecationWar
ning: Passing 1d arrays as data is deprecated in 0.17 and will raise ValueErr
or in 0.19. Reshape your data either using X.reshape(-1, 1) if your data has
a single feature or X.reshape(1, -1) if it contains a single sample.
  DeprecationWarning)
```

```
In [13]:  sumCld_fcI_prediction[0]
```

```
Out[13]:  1
```

PREDICTION BY DATES

```
In [131]:  myDate = myData[['date_of_claim','policy_end','date_of_loss','policy_start']]
```

```
In [132]:  myDate['expired'] = (pd.to_datetime(myDate['date_of_claim']) - pd.to_datetime(
           myDate['policy_end'])).dt.days
           myDate['before_loss'] =(pd.to_datetime(myDate['date_of_claim']) - pd.to_dateti
           me(myDate['date_of_loss'])).dt.days
           myDate['has no policy']=(pd.to_datetime(myDate['date_of_claim']) - pd.to_datet
           ime(myDate['policy_start'])).dt.days
           exp1 = myDate[['expired']]
           myFinDate = myDate[['date_of_claim','expired','before_loss','has no policy']]
```

```
C:\Anaconda\lib\site-packages\ipykernel\__main__.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  if __name__ == '__main__':
C:\Anaconda\lib\site-packages\ipykernel\__main__.py:2: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  from ipykernel import kernelapp as app
C:\Anaconda\lib\site-packages\ipykernel\__main__.py:3: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  app.launch_new_instance()
```

#myFinDate # user inputs

```
In [134]: methods = ['Decision Trees']
```

```
In [135]: exp1_fcI = DecisionTreeClassifier()
```

```
In [136]: x_train = myFinDate[['expired','before_loss','has no policy']]
          exp1_fcI = exp_fcI.fit(x_train, fcI)
```

```
In [147]: ClaimD = '2015-06-22'
          expired = int((pd.to_datetime(ClaimD) - pd.to_datetime('2015-12-31')).days)
            #policy end date
          before_loss=int((pd.to_datetime(ClaimD) - pd.to_datetime('2015-06-21')).days)
            #date of loss
          has_no_policy=int((pd.to_datetime(ClaimD) - pd.to_datetime('2015-01-01')).days
          ) #policy start date
          testData = [expired,before_loss,has_no_policy]
```

```
In [148]: exp1_fcI_prediction = exp1_fcI.predict([[testData[0],testData[1],testData[2
          ]]])
```

```
In [208]: exp1_fcI_prediction[0]

Out[208]: 0
```

```
In [183]: overall = sumCld_fcI_prediction[0] + exp1_fcI_prediction[0]
```

```
In [203]:  if overall == 0:
              print("0: This is not fraud")
          else:
              print("[Warning!!!!] Fraudlent behaviour detected")

          [Warning!!!!] Fraudlent behaviour detected
```