

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

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion_matrix

import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: twitch = pd.read_csv("twitchdata-update.csv")
twitch = twitch.loc[0:500,]
twitch.head()
```

Out[2]:

	Channel	Watch time(Minutes)	Stream time(minutes)	Peak viewers	Average viewers	Followers	Followers gained	
0	xQcOW	6196161750	215250	222720	27716	3246298	1734810	9303
1	summit1g	6091677300	211845	310998	25610	5310163	1370184	8970
2	Gaules	5644590915	515280	387315	10976	1767635	1023779	1026
3	ESL_CSGO	3970318140	517740	300575	7714	3944850	703986	1065
4	Tfue	3671000070	123660	285644	29602	8938903	2068424	7895

```
In [3]: twitch2 = pd.get_dummies(twitch, columns = ["Language"], drop_first = False)
twitch2.head()
```

Out[3]:

	Channel	Watch time(Minutes)	Stream time(minutes)	Peak viewers	Average viewers	Followers	Followers gained	
0	xQcOW	6196161750	215250	222720	27716	3246298	1734810	93036735
1	summit1g	6091677300	211845	310998	25610	5310163	1370184	89705964
2	Gaules	5644590915	515280	387315	10976	1767635	1023779	102611607
3	ESL_CSGO	3970318140	517740	300575	7714	3944850	703986	106546942
4	Tfue	3671000070	123660	285644	29602	8938903	2068424	78998587

5 rows × 28 columns

```
In [4]: twitch2 = twitch2[['Language_English', "Followers", "Views gained"]]
twitch2.head()
```

Out[4]:

	Language_English	Followers	Views gained
0	1	3246298	93036735
1	1	5310163	89705964
2	0	1767635	102611607
3	1	3944850	106546942
4	1	8938903	78998587

```
In [5]: x = twitch2[["Followers", "Views gained"]]
x.head()
```

Out[5]:

	Followers	Views gained
0	3246298	93036735
1	5310163	89705964
2	1767635	102611607
3	3944850	106546942
4	8938903	78998587

```
In [6]: y = twitch2["Language_English"]  
y.head()
```

```
Out[6]: 0    1  
        1    1  
        2    0  
        3    1  
        4    1  
        Name: Language_English, dtype: uint8
```

```
In [7]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

```
In [8]: knn7 = KNeighborsClassifier(n_neighbors = 7)  
knn7.fit(x_train, y_train)
```

```
Out[8]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',  
                             metric_params=None, n_jobs=None, n_neighbors=7, p=2,  
                             weights='uniform')
```

```
In [9]: y_test_preds = knn7.predict(x_test)
```

```
In [10]: confusion_matrix(y_test, y_test_preds)
```

```
Out[10]: array([[30, 21],  
                [20, 30]])
```

```
In [11]: Sensitivity7 = 36/(23 + 36)  
Sensitivity7
```

```
Out[11]: 0.6101694915254238
```

```
In [12]: specificity7 = 27/(27 + 15)  
specificity7
```

```
Out[12]: 0.6428571428571429
```

DECISION TREE

```
In [13]: tree = DecisionTreeClassifier(max_depth = 4)  
tree.fit(x_train, y_train)
```

```
Out[13]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=4,  
                                max_features=None, max_leaf_nodes=None,  
                                min_impurity_decrease=0.0, min_impurity_split=None,  
                                min_samples_leaf=1, min_samples_split=2,  
                                min_weight_fraction_leaf=0.0, presort=False, random_state=None,  
                                splitter='best')
```

```
In [14]: y_test_preds_tree = tree.predict(x_test)
         confusion_matrix(y_test, y_test_preds_tree)
```

```
Out[14]: array([[40, 11],
               [31, 19]])
```

```
In [15]: Sensitivity4 = 28/(31 + 28)
         Sensitivity4
```

```
Out[15]: 0.4745762711864407
```

```
In [16]: specificity4 = 32/(32 + 10)
         specificity4
```

```
Out[16]: 0.7619047619047619
```