

In [1]:

```
import pandas as pd
```

In [2]:

```
data = pd.read_csv('musk_csv.csv')
```

▶

In [3]:

```
data.head()
```

Out[3]:

	ID	molecule_name	conformation_name	f1	f2	f3	f4	f5	f6	f7	...	f158	f159
0	1	MUSK-211	211_1+1	46	-108	-60	-69	-117	49	38	...	-308	52
1	2	MUSK-211	211_1+10	41	-188	-145	22	-117	-6	57	...	-59	-2
2	3	MUSK-211	211_1+11	46	-194	-145	28	-117	73	57	...	-134	-154
3	4	MUSK-211	211_1+12	41	-188	-145	22	-117	-7	57	...	-60	-4
4	5	MUSK-211	211_1+13	41	-188	-145	22	-117	-7	57	...	-60	-4

5 rows × 170 columns



In [4]:

```
data.shape
```

Out[4]:

```
(6598, 170)
```

In [5]:

```
data['class'].value_counts()
```

Out[5]:

```
0    5581
1    1017
Name: class, dtype: int64
```

In [6]:

```
target = data['class']  
del data['class']
```

In [7]:

```
from sklearn.model_selection import train_test_split  
x_train,x_val,y_train,y_val= train_test_split(data,target,test_size=0.2,random_stat
```

In [8]:

```
print('train data shape: ',x_train.shape)  
print('test data shape: ',x_val.shape )
```

```
train data shape:  (5278, 169)  
test data shape:  (1320, 169)
```

Important observation

When molecule_name starts with MUSK class is 1

When molecule_name starts with NON-MUSK class is 0

In [9]:

```
def predict(x):  
    if(x[:3]=='MUS'):  
        return(1)  
    elif(x[:3]=='NON'):  
        return(0)
```

In [10]:

```
y_train_predict = x_train['molecule_name'].apply(lambda x:predict(x))  
y_val_predict = x_val['molecule_name'].apply(lambda x:predict(x))
```

Calculating accuracy

In [11]:

```
from sklearn.metrics import accuracy_score
train_acc = accuracy_score(y_train, y_train_predict)*100
val_acc = accuracy_score(y_val, y_val_predict)*100
print('train accuracy: ',train_acc)
print('val accuracy: ',val_acc)
```

```
train accuracy: 100.0
val accuracy: 100.0
```

Calculating precision, recall, f1 score

In [12]:

```
from sklearn.metrics import precision_score, recall_score, f1_score
train_precision = precision_score(y_train, y_train_predict)
val_precision = precision_score(y_val, y_val_predict)

train_recall = recall_score(y_train, y_train_predict)
val_recall = recall_score(y_val, y_val_predict)

train_f1 = f1_score(y_train, y_train_predict)
val_f1 = f1_score(y_val, y_val_predict)
print('train_precision : ', train_precision, 'val_precision : ', val_precision)
print('train_recall : ', train_recall, 'val_recall : ', val_recall)
print('train_f1 : ', train_f1, 'val_f1 : ', val_f1)
```

```
train_precision : 1.0 val_precision : 1.0
train_recall : 1.0 val_recall : 1.0
train_f1 : 1.0 val_f1 : 1.0
```

train accuracy = 100%

Val accuracy = 100%

train f1 score = 1

val f1 score = 1