MUSK CLASSIFICATION PROBLEM

1.INTRODUCTION

This is basically a classification problem of organical chemical compounds and we have to classify an organical compound as Musk or Non-Musk depending on the data given.We build a classification model using Artificial neural network on given data.This model predict the class of an organic compound either Musk or Non-Musk.

2.Data Preprocessing

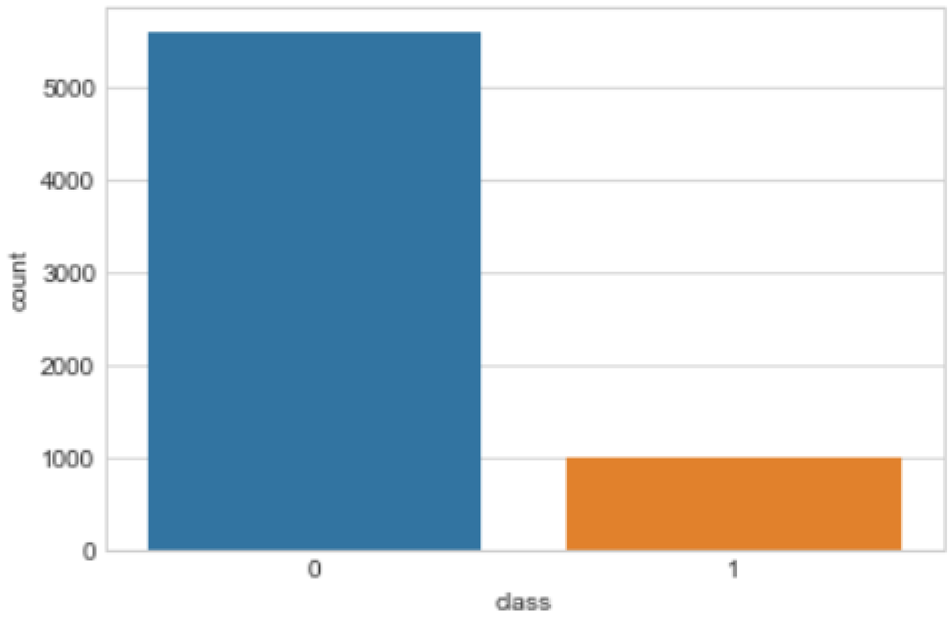
For every Machine Learning model, the importance of data rules above every other factor. I cannot stress this enough… LOOK AT YOUR DATA!!!. Data might give some explanation while training.

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.

We plot a graph that show the count of organic compound belonging to each class.

In the below graph 0 indicates Non-musk class and 1 indicatest Musk class.

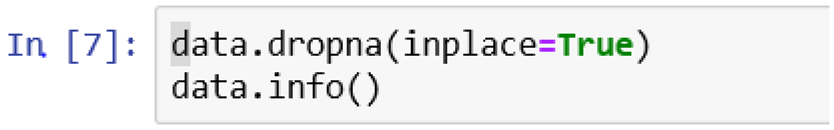
Count of Non-Musk organic compound are more than the musk organic compound as shown in graph.



The preprocessing I have done in this project are

1. Handling Null Values

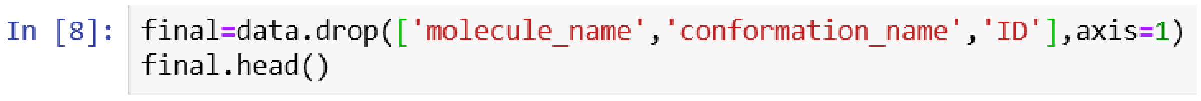
   Check the dataframe for null values,if present then drop that row from the data frame.



```
In [7]: data.dropna(inplace=True)
        data.info()
```

2. Removing unnecessary detail

   Unecessary coloumns are removed from the dataframe.The unnecessary details in the data are molecule name, conformation name and ID,they can,t affect the predicton.So we drop that coloumns

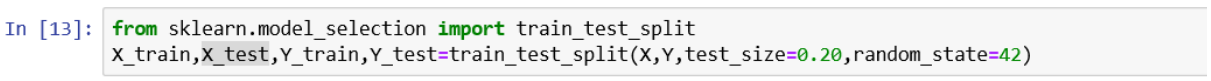


```
In [8]: final=data.drop(['molecule_name','conformation_name','ID'],axis=1)
        final.head()
```

We can do some more preprocessing but this are enough to get a good accuracy.

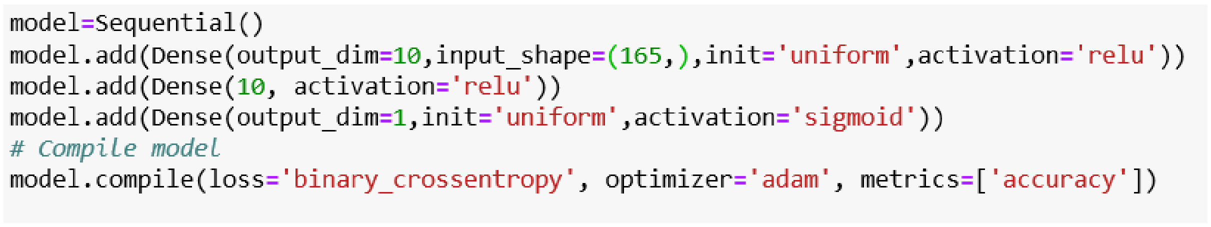
3.Training and Testing

We split our data in two parts that are trained data and test data using train_test_split function.

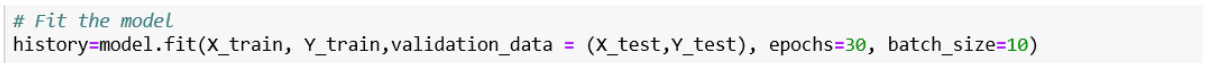


```
In [13]: from sklearn.model_selection import train_test_split
         X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.20,random_state=42)
```

After split 80% of data belongs to trained data and 20% data belongs to test data.Afterthat we trained our model using the trained data.



```
model=Sequential()
model.add(Dense(output_dim=10,input_shape=(165,),init='uniform',activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(output_dim=1,init='uniform',activation='sigmoid'))
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```



```
# Fit the model
history=model.fit(X_train, Y_train,validation_data = (X_test,Y_test), epochs=30, batch_size=10)
```

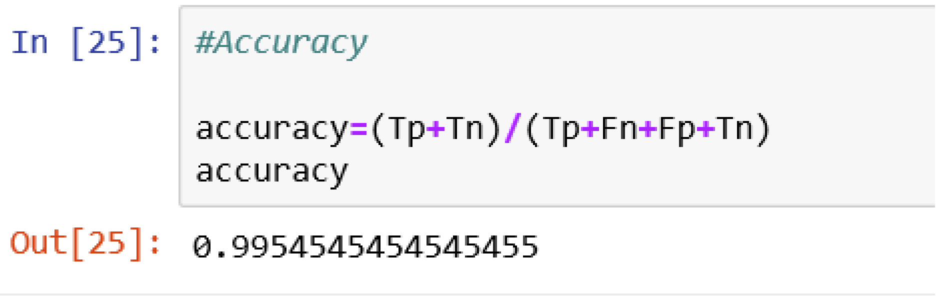
Neural network is run for 30 epochs for getting a good accuracy and validation accuracy.

ANN(Artificial neural network) learn on every epoch and improve its accuracy epoch by epoch.

4.Result

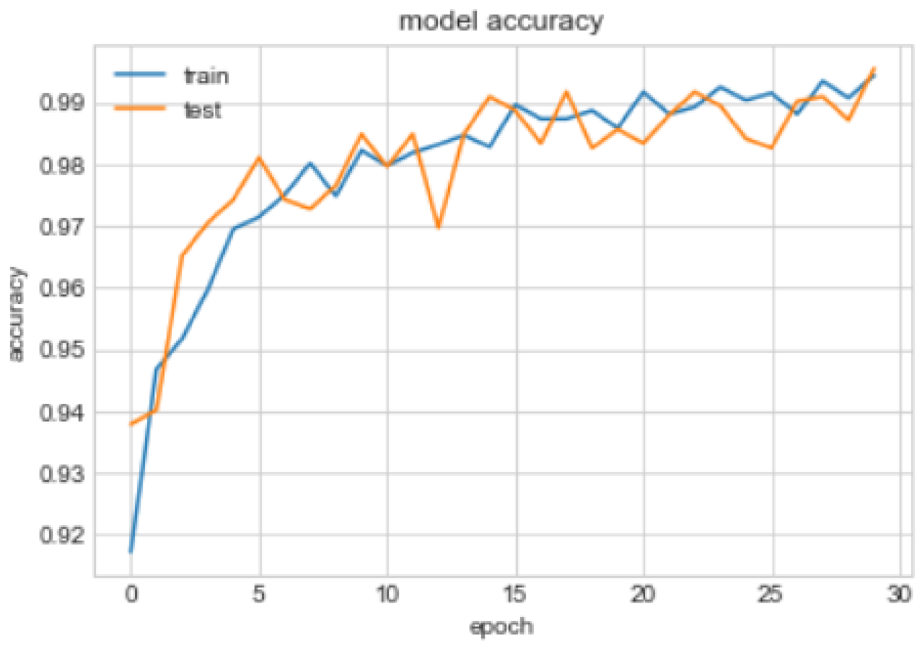
Accuracy : 0.9954

We got an accuracy of 99% from this model using given data



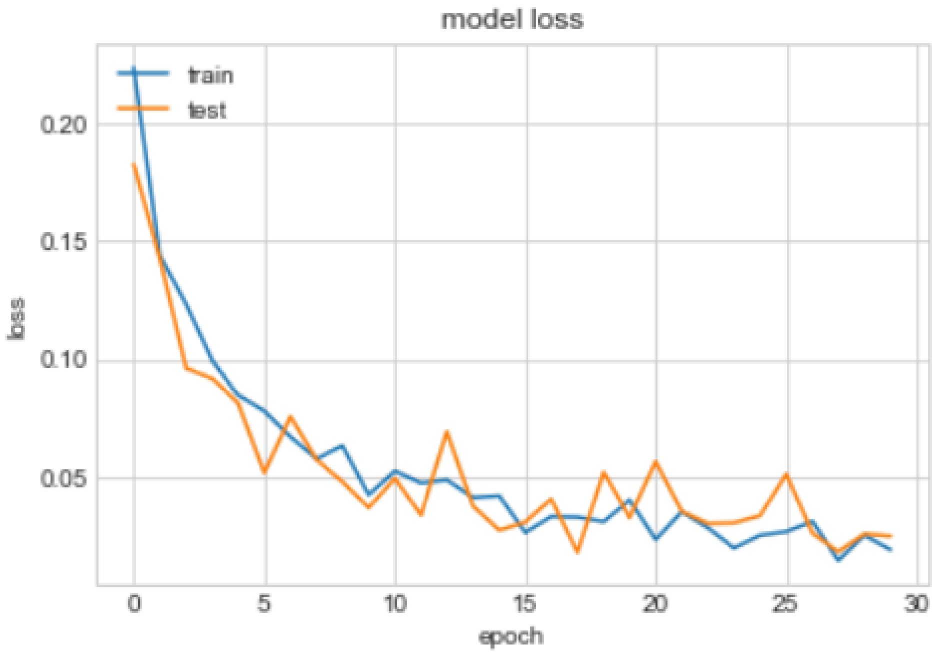
```
In [25]: #Accuracy

         accuracy=(Tp+Tn)/(Tp+Fn+Fp+Tn)
         accuracy

Out[25]: 0.9954545454545455
```

Graph showing the variation in accuracy along with validation accuracy with respect to number of epochs.



Loss : We get Mean squared error =0.0045

Graph showing the variation in loss and validation loss with respect to epochs.



Precision : 0.9990

Recall : 0.9954

F1_score:0.9972