

Prediction of a Chemical whether it is musk or non musk, using Artificial Neural Network

# Resources:

- Dataset: <https://drive.google.com/file/d/1pZhzZnaPi74aKCQImSPzrTxWzVeE0qv/view?usp=sharing>
- Anaconda navigator 2019.10 :<https://www.anaconda.com/distribution/> 3.7 :
- Spyder: <https://www.spyder-ide.org>
- pandas: <https://pandas.pydata.org/pandas-docs/stable/>
- Numpy: <https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>
- Scikit-learn: <http://scikit-learn.org/stable/>
- Matplotlib: <https://matplotlib.org/2.1.0/index.html>
- Keras: <https://keras.io/>

# Tools Used:

- Data Analysis and Preprocessing:
  - Pandas, Numpy and Scikit-learn
- Data Visualization:
  - Matplotlib
- Deep Learning Library:
  - Keras (with Tensorflow backend)

# We will divide the project into 2 parts:

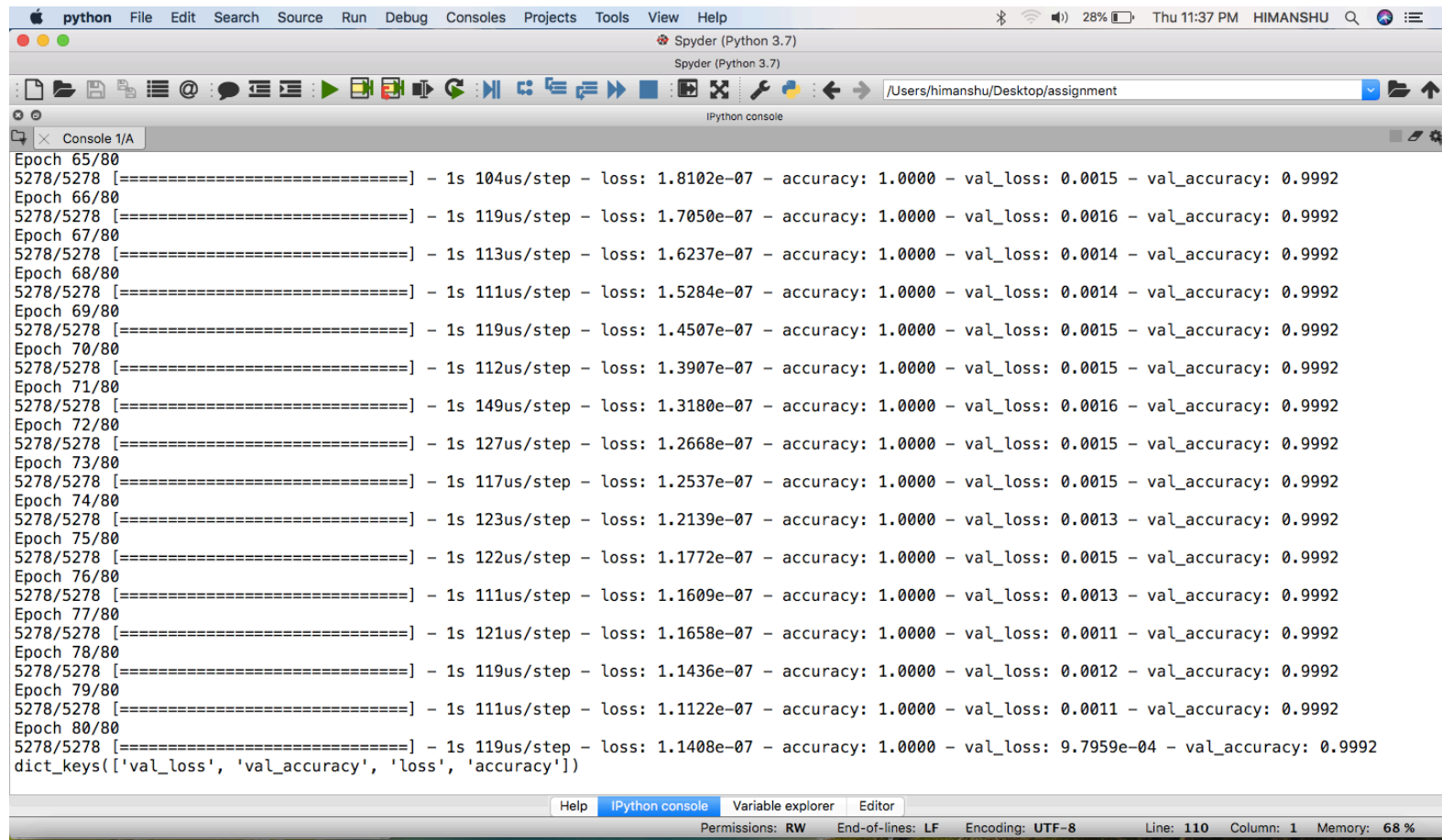
1. Data Exploration and Data Preprocessing:
  1. Description of Data
  2. Feature scaling with class StandardScaler
  3. Splitting data into training and test set
  4. Using matplotlib to visualization

# Approach:

## Network Architecture:

- Input Layer with 84 Neurons(average of input and output nodes)
- First Hidden Layer with 84 Neurons and “Rectified” Activation Function
- Second Hidden Layer with 84 Neurons and “Rectified” Activation Function
- Output Layer with 1 Neurons and “Sigmoid” Activation Function
- Number of epochs 80 and batch size 20

# Run time :



The screenshot shows the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The title bar indicates 'Spyder (Python 3.7)'. The main editor area shows a file path: '/Users/himanshu/Desktop/assignment'. Below the editor is a Jupyter console window titled 'Console 1/A'. The console displays the output of a training process over 80 epochs. Each epoch's output includes the number of steps, time per step, loss, accuracy, validation loss, and validation accuracy. The loss decreases from approximately 1.81e-07 to 1.14e-07, while accuracy remains at 1.0000. The validation loss fluctuates slightly, ending at 9.7959e-04, and validation accuracy remains at 0.9992. The final line of the console output is a dictionary: `dict_keys(['val_loss', 'val_accuracy', 'loss', 'accuracy'])`. The bottom status bar shows 'Permissions: RW', 'End-of-lines: LF', 'Encoding: UTF-8', 'Line: 110', 'Column: 1', and 'Memory: 68 %'.

```
Epoch 65/80
5278/5278 [=====] - 1s 104us/step - loss: 1.8102e-07 - accuracy: 1.0000 - val_loss: 0.0015 - val_accuracy: 0.9992
Epoch 66/80
5278/5278 [=====] - 1s 119us/step - loss: 1.7050e-07 - accuracy: 1.0000 - val_loss: 0.0016 - val_accuracy: 0.9992
Epoch 67/80
5278/5278 [=====] - 1s 113us/step - loss: 1.6237e-07 - accuracy: 1.0000 - val_loss: 0.0014 - val_accuracy: 0.9992
Epoch 68/80
5278/5278 [=====] - 1s 111us/step - loss: 1.5284e-07 - accuracy: 1.0000 - val_loss: 0.0014 - val_accuracy: 0.9992
Epoch 69/80
5278/5278 [=====] - 1s 119us/step - loss: 1.4507e-07 - accuracy: 1.0000 - val_loss: 0.0015 - val_accuracy: 0.9992
Epoch 70/80
5278/5278 [=====] - 1s 112us/step - loss: 1.3907e-07 - accuracy: 1.0000 - val_loss: 0.0015 - val_accuracy: 0.9992
Epoch 71/80
5278/5278 [=====] - 1s 149us/step - loss: 1.3180e-07 - accuracy: 1.0000 - val_loss: 0.0016 - val_accuracy: 0.9992
Epoch 72/80
5278/5278 [=====] - 1s 127us/step - loss: 1.2668e-07 - accuracy: 1.0000 - val_loss: 0.0015 - val_accuracy: 0.9992
Epoch 73/80
5278/5278 [=====] - 1s 117us/step - loss: 1.2537e-07 - accuracy: 1.0000 - val_loss: 0.0015 - val_accuracy: 0.9992
Epoch 74/80
5278/5278 [=====] - 1s 123us/step - loss: 1.2139e-07 - accuracy: 1.0000 - val_loss: 0.0013 - val_accuracy: 0.9992
Epoch 75/80
5278/5278 [=====] - 1s 122us/step - loss: 1.1772e-07 - accuracy: 1.0000 - val_loss: 0.0015 - val_accuracy: 0.9992
Epoch 76/80
5278/5278 [=====] - 1s 111us/step - loss: 1.1609e-07 - accuracy: 1.0000 - val_loss: 0.0013 - val_accuracy: 0.9992
Epoch 77/80
5278/5278 [=====] - 1s 121us/step - loss: 1.1658e-07 - accuracy: 1.0000 - val_loss: 0.0011 - val_accuracy: 0.9992
Epoch 78/80
5278/5278 [=====] - 1s 119us/step - loss: 1.1436e-07 - accuracy: 1.0000 - val_loss: 0.0012 - val_accuracy: 0.9992
Epoch 79/80
5278/5278 [=====] - 1s 111us/step - loss: 1.1122e-07 - accuracy: 1.0000 - val_loss: 0.0011 - val_accuracy: 0.9992
Epoch 80/80
5278/5278 [=====] - 1s 119us/step - loss: 1.1408e-07 - accuracy: 1.0000 - val_loss: 9.7959e-04 - val_accuracy: 0.9992
dict_keys(['val_loss', 'val_accuracy', 'loss', 'accuracy'])
```

# Performance:

● Accuracy: 0.9984

● F1 score :0.9950

```

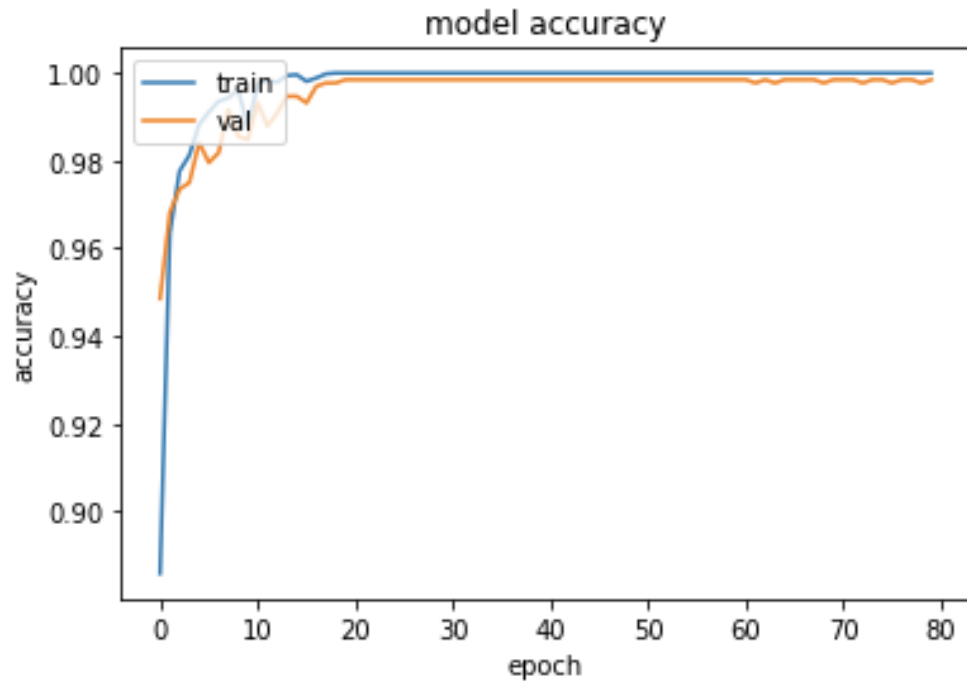
                                epocn
Accuracy 0.9984848484848485
f1 score 0.9950248756218906
      precision    recall  f1-score   support

     0           1.00      1.00      1.00       1120
     1           0.99      1.00      1.00        200

   accuracy                   1.00       1320
  macro avg           1.00      1.00      1.00       1320
 weighted avg           1.00      1.00      1.00       1320
```

In [5]:

## Accuracy graph:





# Loss Graph:

