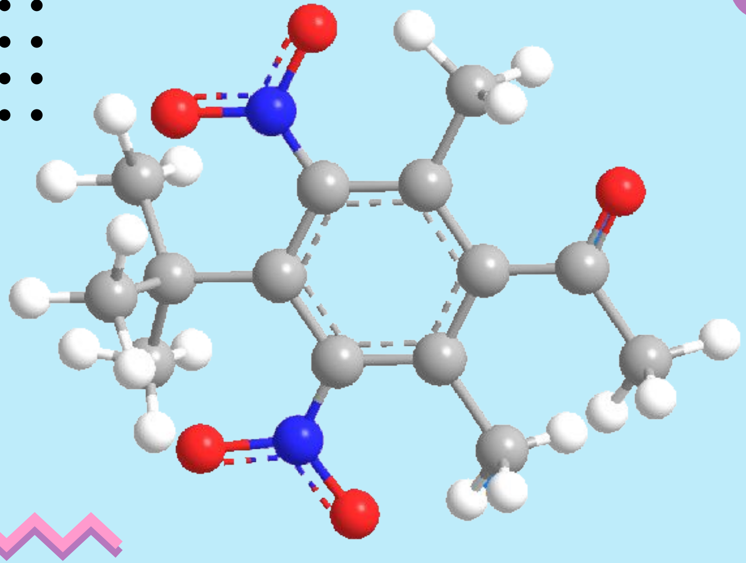


Chemical Classification

Musk OR Non-Musk

Table of Contents

01

Problem

Chemical classification

02

Analysis

What data is ?
How to measure ? etc

03

Create model

Created with ANN

04

Test & Save

Testing with data
Validation accuracy & loss

Introduction

The given dataset contains details about organic chemical compounds including their chemical features, isomeric conformation, names and the classes in which they are classified. The compounds are classified as either ‘Musk’ or ‘Non-Musk’ compounds. Your task is to build a classification model on the given data using any Deep Learning approach that you deem appropriate viz. Multi-Layer Perceptron, CNN, RNN, etc. or you could also use transfer learning approaches through selection of appropriate pre-trained model. The data has to be split in a 80:20 ratio for training and validation datasets. You can perform whatever preprocessing and post-processing operations on the data that may help you improve the performance of your model. You are required to report the performance measures of the model viz. Accuracy(Training and Validation) and Loss(Training and Validation) graphs, F1 score, precision, recall, etc. along with a well detailed report of what models, pre-processing, post-processing approaches you have used and why you chose to use these approaches.

Made in for this Company

We at Credicxo provide immediate monetary support to our customers in the form of instant Personal Loan with hassle-free processing and swift approval. Credicxo provides financial support to ambitious Entrepreneurs, Students, Salaried and Self-employed individuals. The eligibility assessment is nominal and we provide our service 24x7 with instant approval and quick disbursal.

Credicxo

Simple Swift **Money**

About Me

Kushal Dulani

Computer science Engineer

Mobile : +91 777-88-222-89
dulanikushal@gmail.com

Work Process

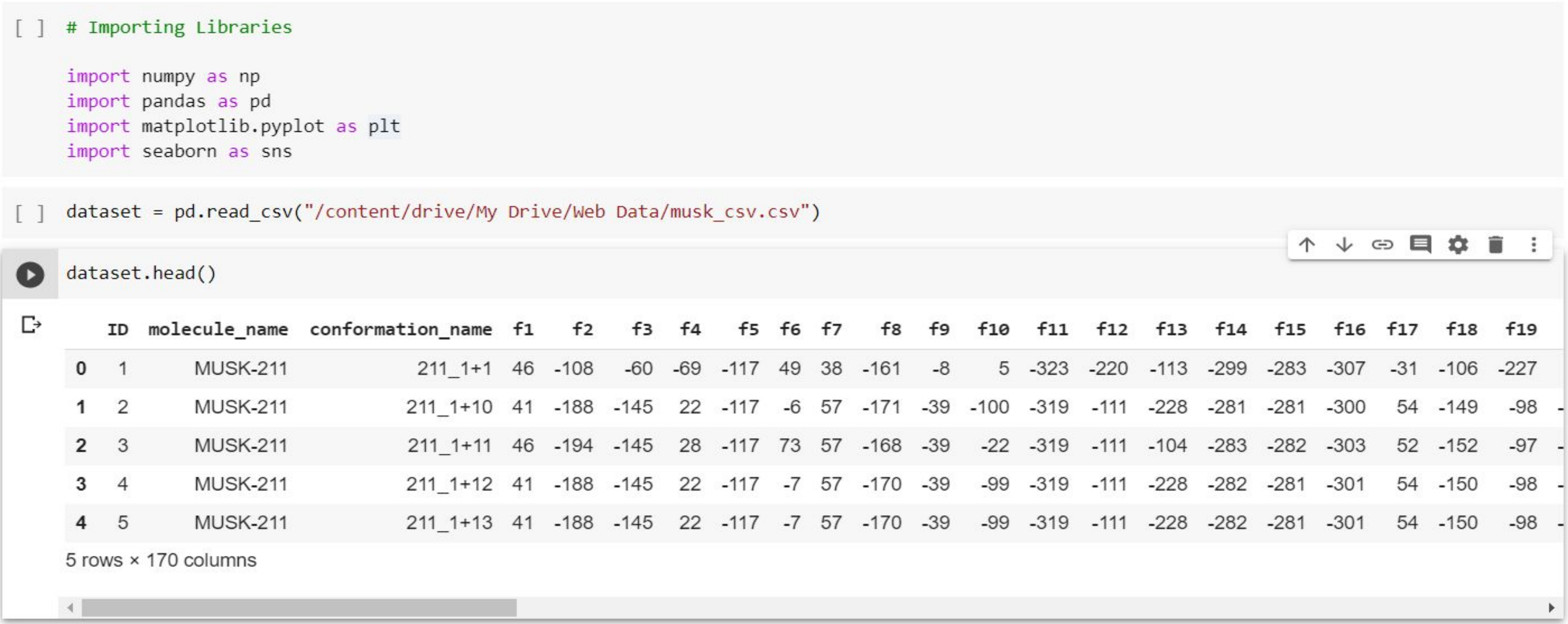
Analysis

Creating classifier

Training

Testing & Deploy

Code Demo



```
# Importing Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
dataset = pd.read_csv("/content/drive/My Drive/Web Data/musk_csv.csv")
```

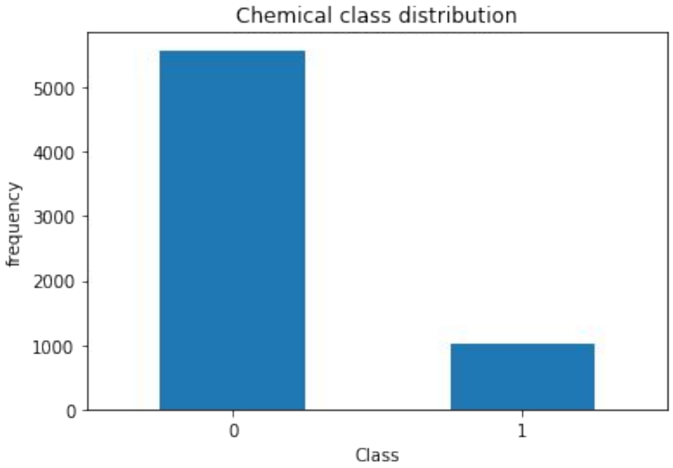
```
dataset.head()
```

	ID	molecule_name	conformation_name	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19
0	1	MUSK-211	211_1+1	46	-108	-60	-69	-117	49	38	-161	-8	5	-323	-220	-113	-299	-283	-307	-31	-106	-227
1	2	MUSK-211	211_1+10	41	-188	-145	22	-117	-6	57	-171	-39	-100	-319	-111	-228	-281	-281	-300	54	-149	-98
2	3	MUSK-211	211_1+11	46	-194	-145	28	-117	73	57	-168	-39	-22	-319	-111	-104	-283	-282	-303	52	-152	-97
3	4	MUSK-211	211_1+12	41	-188	-145	22	-117	-7	57	-170	-39	-99	-319	-111	-228	-282	-281	-301	54	-150	-98
4	5	MUSK-211	211_1+13	41	-188	-145	22	-117	-7	57	-170	-39	-99	-319	-111	-228	-282	-281	-301	54	-150	-98

5 rows × 170 columns

Analysis

There are 6597 total chemicals



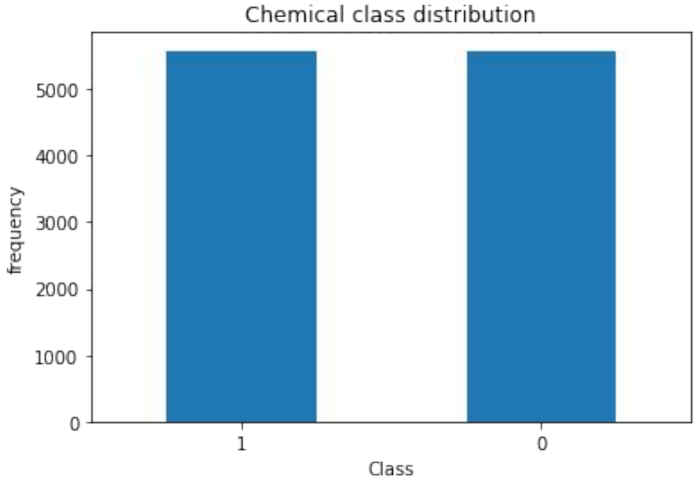
6597 Total

1017 Musk

5581 Non Musk

Analysis

Data Balancing



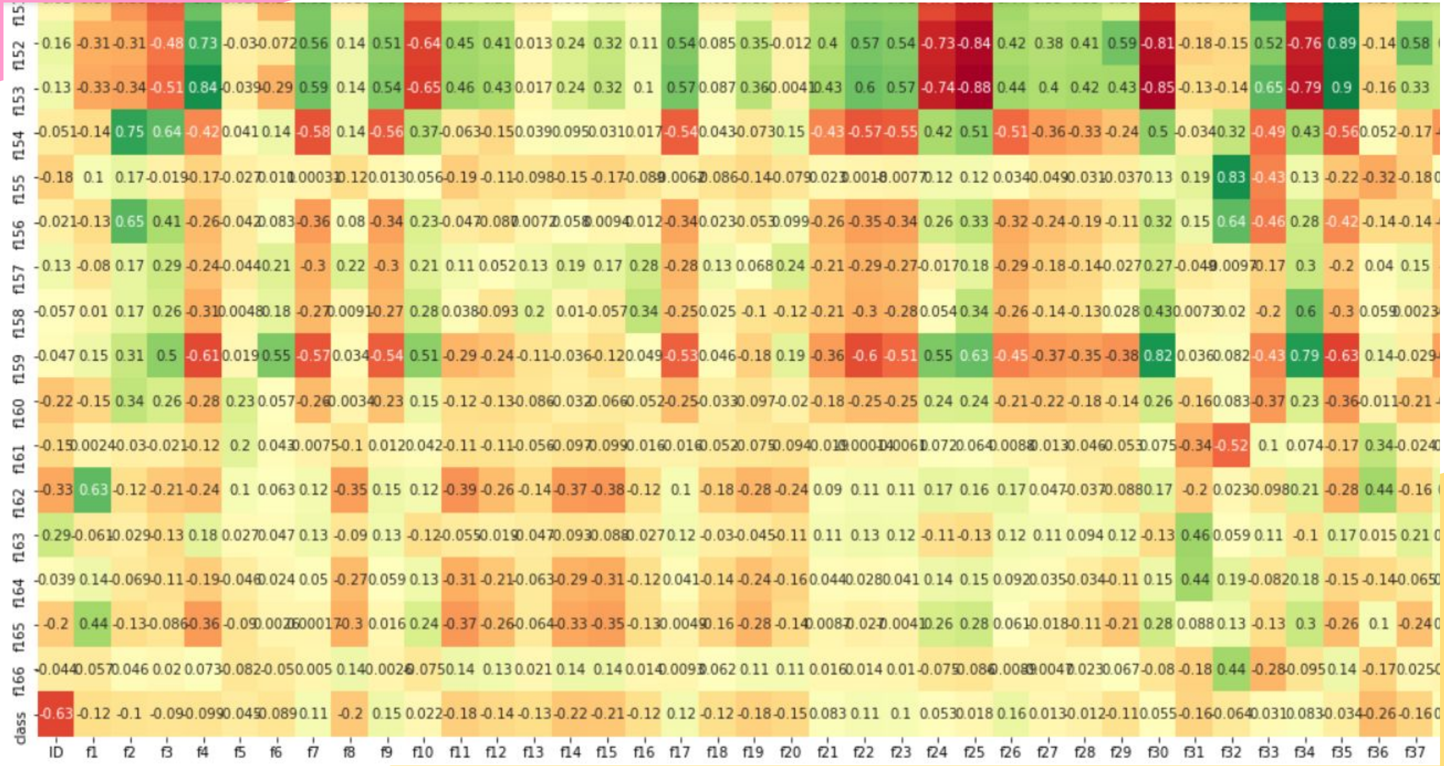
Combine Sampling Method

11162 Total

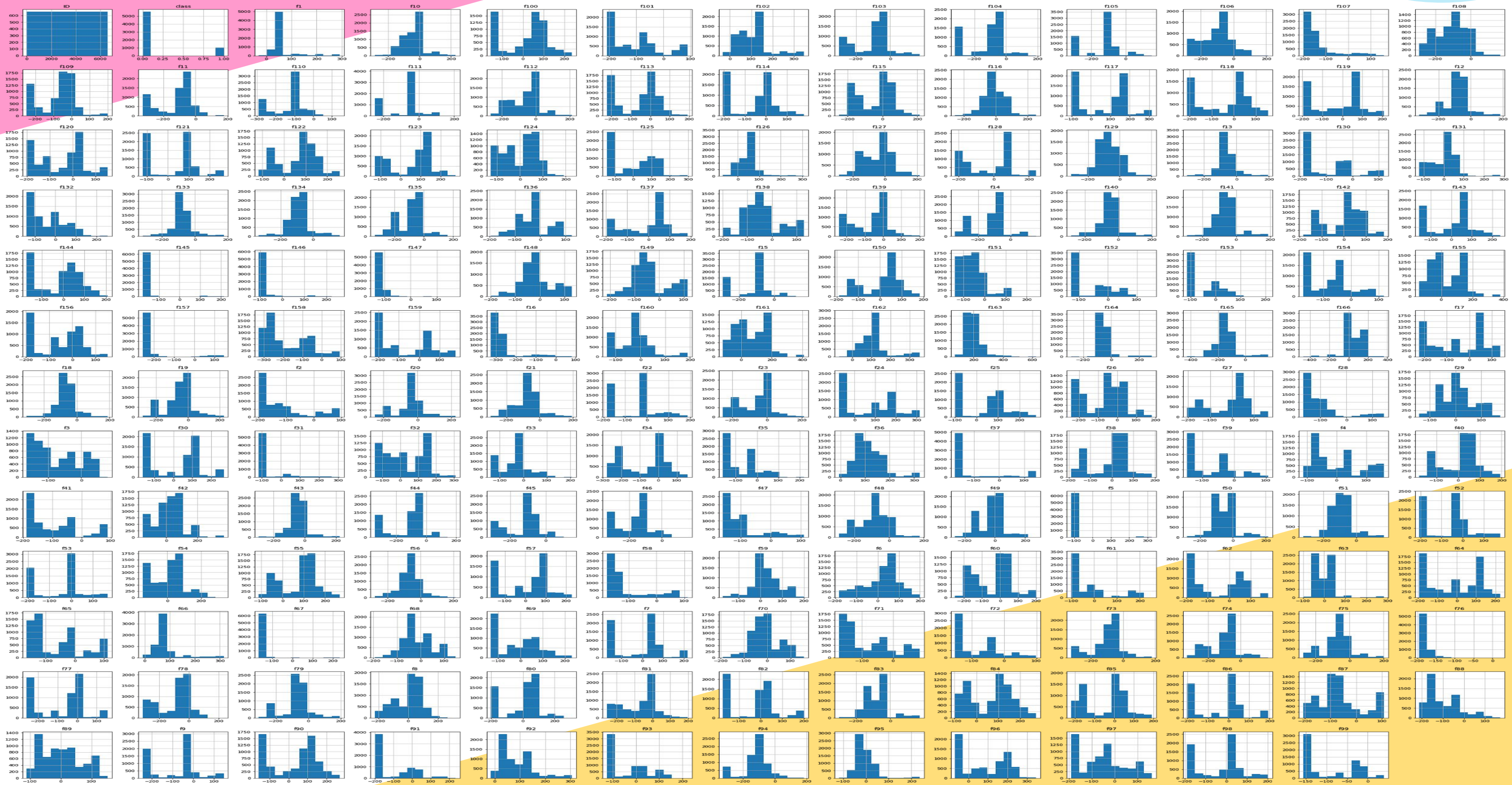
5581 Musk

5581 Non Musk

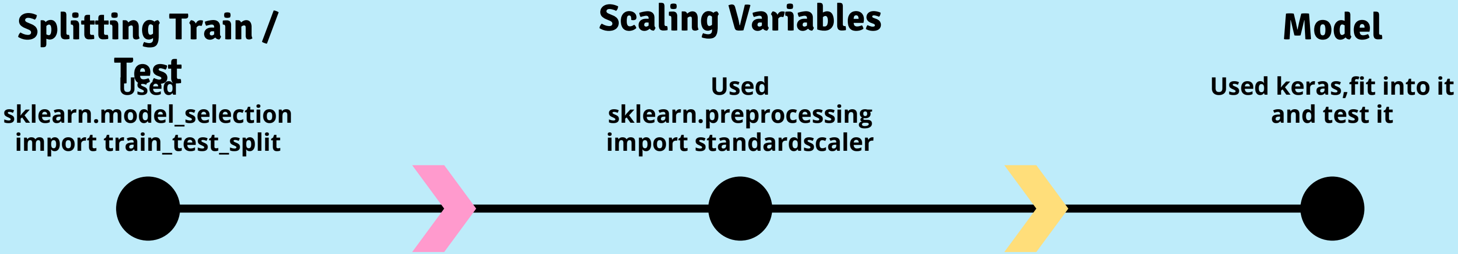
Samples of Heatmap



Samples of Histogram



Work Flow



Splitting Train / Test

Used sklearn.model_selection import train_test_split

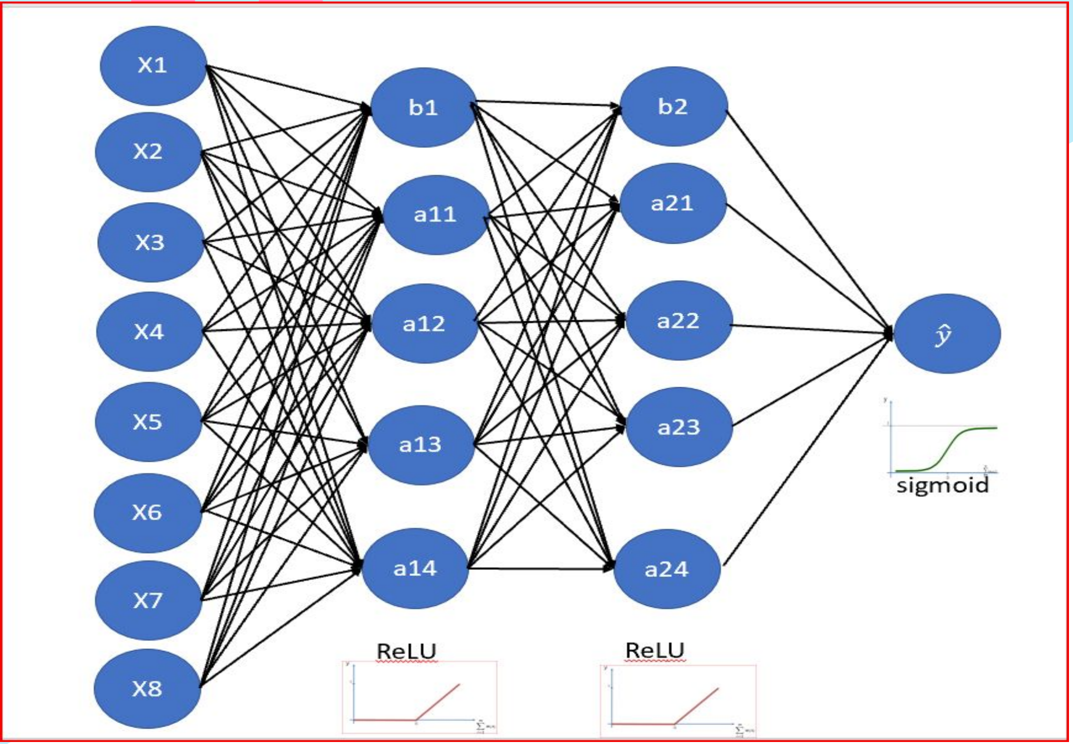
Scaling Variables

Used sklearn.preprocessing import standardscaler

Model

Used keras,fit into it and test it

Accuracy



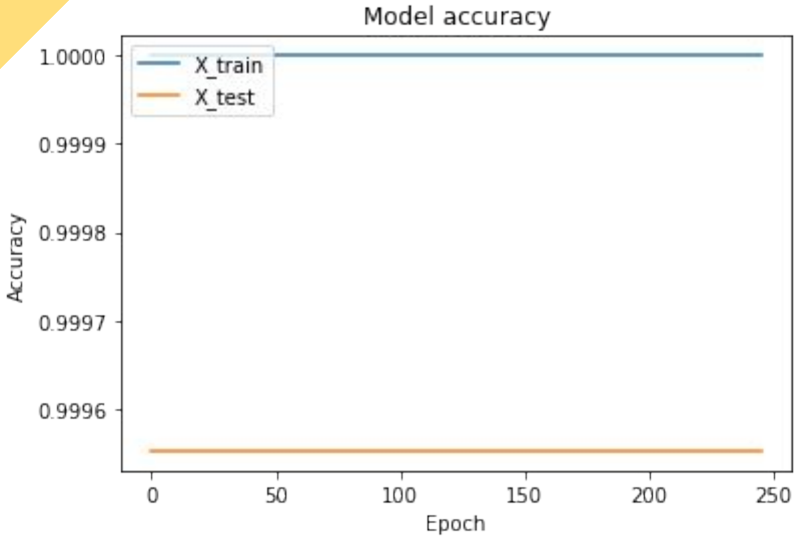
batch_size=40,epochs=246

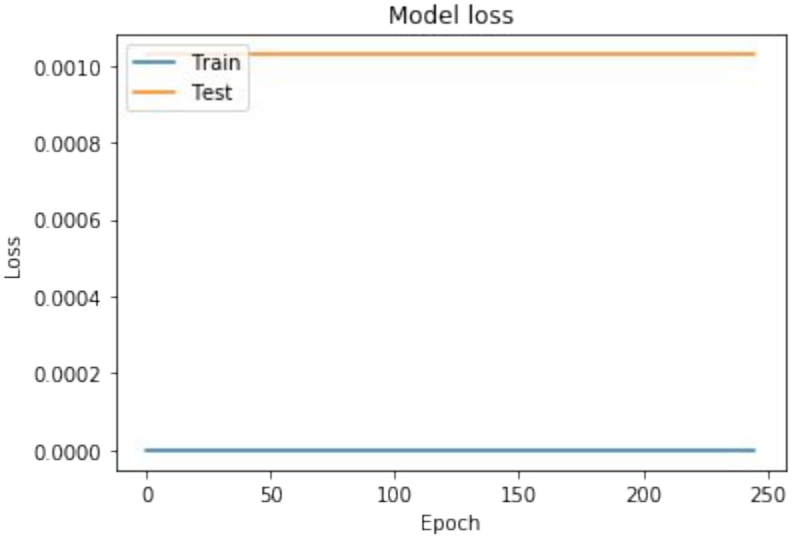
95.32%

batch_size=10,epochs=893

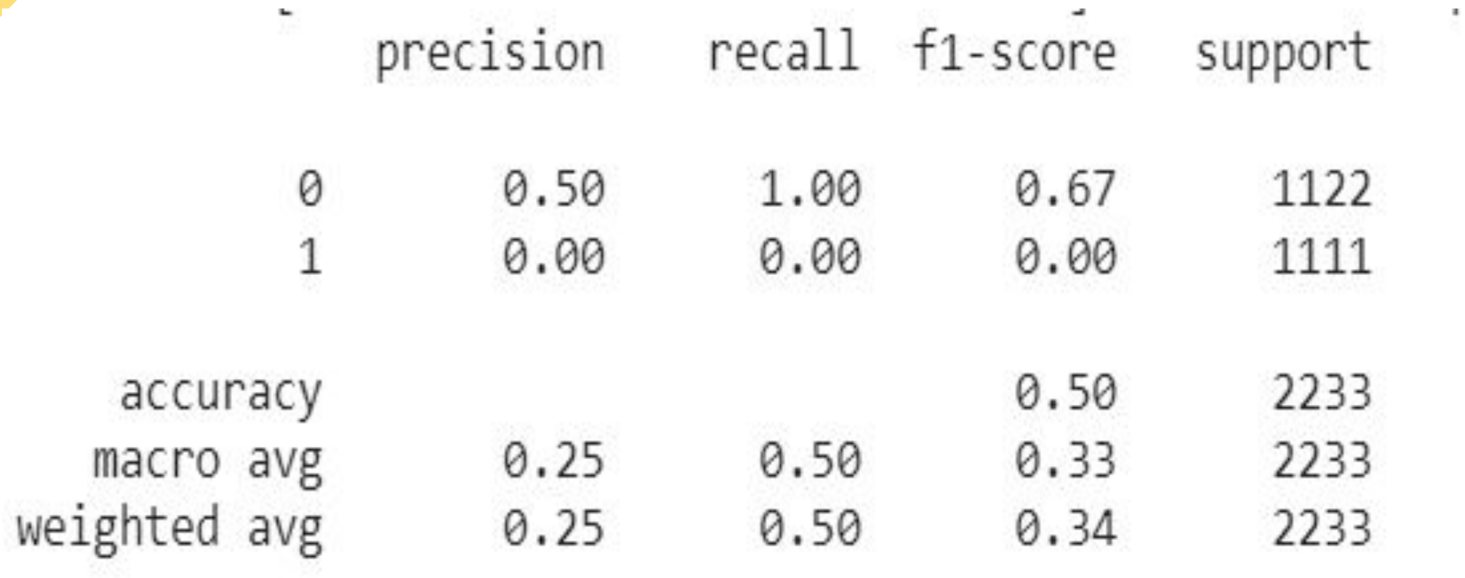
99.14%

Training & Validation





Recall & F-1 Score

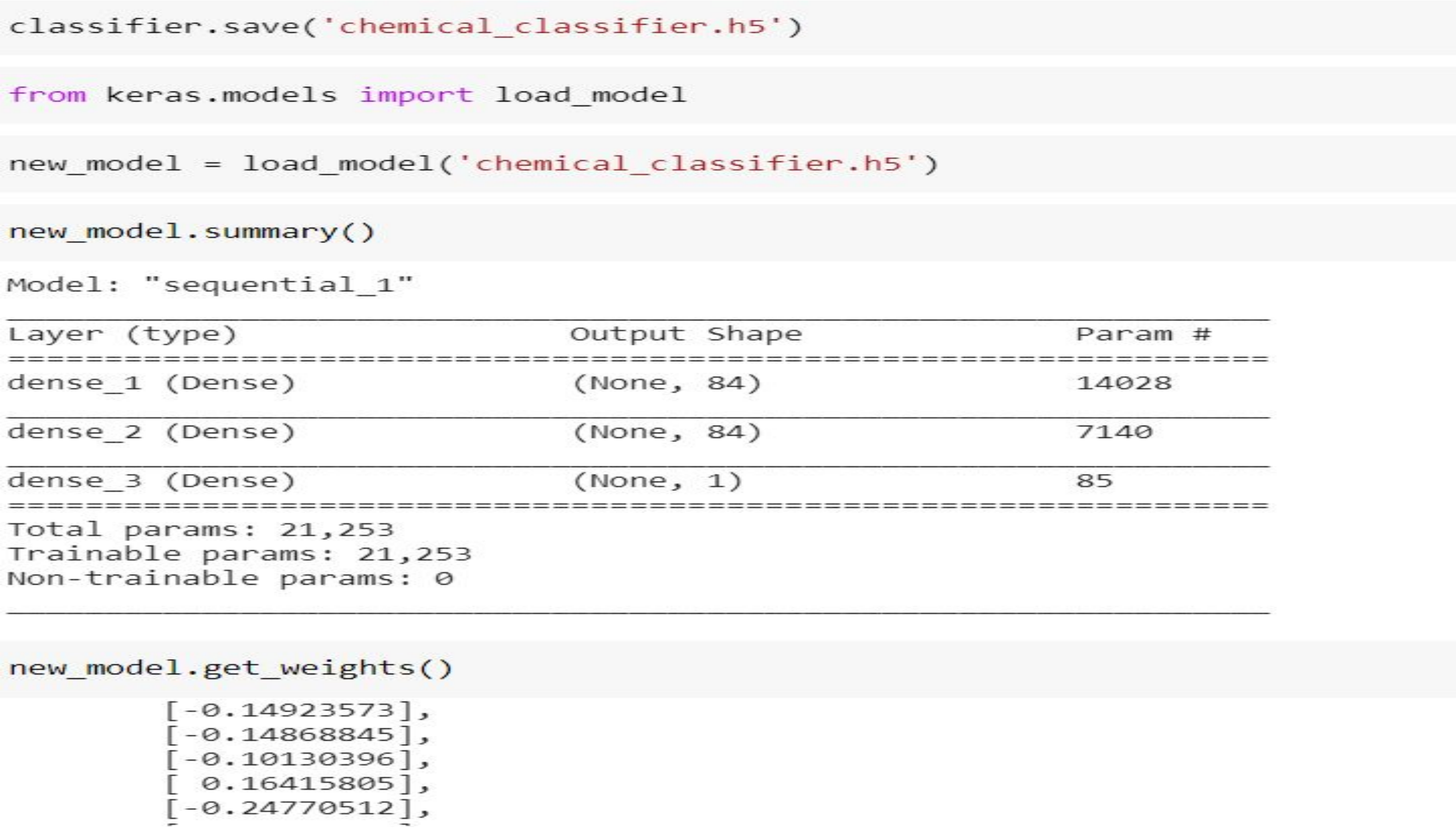


```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67      1122
           1       0.00      0.00      0.00      1111

    accuracy                           0.50      2233
   macro avg       0.25      0.50      0.33      2233
weighted avg       0.25      0.50      0.34      2233
```

H5 model trained



```
classifier.save('chemical_classifier.h5')
```

```
from keras.models import load_model
```

```
new_model = load_model('chemical_classifier.h5')
```

```
new_model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 84)                14028
_________________________________________________________________
dense_2 (Dense)              (None, 84)                7140
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 85
=================================================================
Total params: 21,253
Trainable params: 21,253
Non-trainable params: 0
_________________________________________________________________
```

```
new_model.get_weights()
```

```
        [-0.14923573],
        [-0.14868845],
        [-0.10130396],
        [ 0.16415805],
        [-0.24770512],
```

Used For Development & upload

Thank You