

```
In [1]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
```

```
In [2]: # Import data set csv
data_set = pd.read_csv('filtered_data_set.csv')
```

```
In [3]: # Set X and y columns
X = data_set[['valence', 'acousticness', 'danceability', 'duration_ms', 'energy', 'exp',
              'instrumentalness', 'key', 'liveness', 'loudness', 'mode', 'speechiness',
              'tempo', 'time_signature']]
y = data_set['popularity'].values
```

```
In [4]: # Create the X training and testing set, and Y training and testing set where 70%
# are for the training set and the rest to the testing set.
x_testing_set, x_training_set, y_testing_set, y_training_set = train_test_split(X, y,
```

```
In [5]: # Create model we still construct sequentially
model = Sequential()

# Add dense (every input connected to all units in hidden layer)
# Activation - sigmoid maps between 0 and 1. relu maps to 0 or 1
model.add(Dense(15, input_dim=13, activation='relu'))
model.add(Dense(18, activation='relu'))
model.add(Dense(13, activation='relu'))
model.add(Dense(10, activation='relu'))

# Output Layer
model.add(Dense(1, activation='sigmoid'))
```

WARNING:tensorflow:From D:\Anaconda\lib\site-packages\tensorflow\python\ops\init\_ops.py:1251: calling VarianceScaling.\_\_init\_\_ (from tensorflow.python.ops.init\_ops) with dtype is deprecated and will be removed in a future version.  
Instructions for updating:  
Call initializer instance with the dtype argument instead of passing it to the constructor

```
In [6]: # Compile the model.
# Optimizer - Adam is an efficient optimize to apply gradient descent to the model
# Metrics - want the accuracy on how the model predicts
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

WARNING:tensorflow:From D:\Anaconda\lib\site-packages\tensorflow\python\ops\nn\_impl.py:180: add\_dispatch\_support.<locals>.wrapper (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.  
Instructions for updating:  
Use tf.where in 2.0, which has the same broadcast rule as np.where

```
In [7]: model.fit(x_training_set,y_training_set,epochs=1000, batch_size=64)
```

```
6910/6910 [=====] - 0s 10us/sample - loss: 0.4464 - acc: 0.7973
Epoch 996/1000
6910/6910 [=====] - 0s 10us/sample - loss: 0.4425 - acc: 0.8001
Epoch 997/1000
6910/6910 [=====] - 0s 10us/sample - loss: 0.4445 - acc: 0.7948
Epoch 998/1000
6910/6910 [=====] - 0s 10us/sample - loss: 0.4421 - acc: 0.8001
Epoch 999/1000
6910/6910 [=====] - 0s 10us/sample - loss: 0.4411 - acc: 0.7990
Epoch 1000/1000
6910/6910 [=====] - 0s 10us/sample - loss: 0.4458 - acc: 0.7919
```

```
Out[7]: <tensorflow.python.keras.callbacks.History at 0x1d59fdd1608>
```

```
In [8]: # Get the predicted values with the testing set
test_predictions = model.predict(x_testing_set)
```

```
In [9]: # Resize to series
test_predictions = pd.Series(test_predictions.reshape(2961,))
```

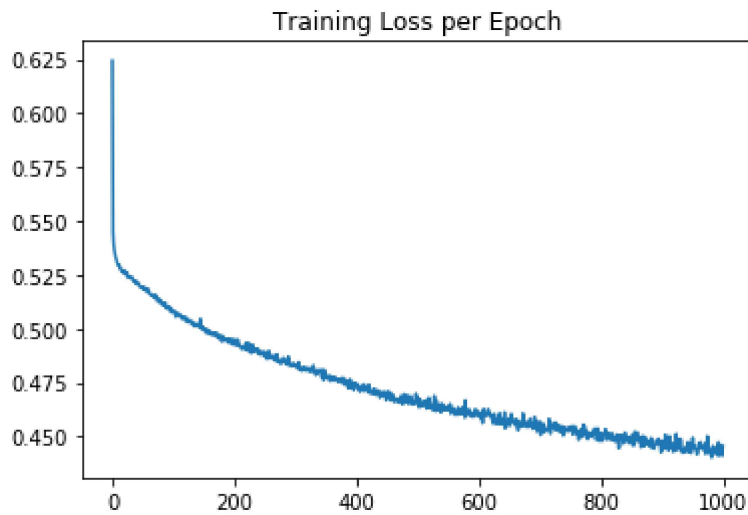
```
In [10]: training_score = model.evaluate(x_training_set,y_training_set)
test_score = model.evaluate(x_testing_set,y_testing_set)
print(training_score)
print(test_score)
```

```
6910/6910 [=====] - 0s 12us/sample - loss: 0.4377 - acc: 0.7988
2961/2961 [=====] - 0s 9us/sample - loss: 0.6067 - acc: 0.7413
[0.4377477920995607, 0.79884225]
[0.6066685957165434, 0.7413036]
```

```
In [11]: # Find predict y values with the x testing set and find accuracy
ynew = model.predict_classes(x_testing_set)
correct=0
for i in range(0,len(ynew)):
    if(ynew[i]==y_testing_set[i]):
        correct = correct + 1
print("Accuracy=", correct/len(test_predictions))
```

Accuracy= 0.7413036136440392

```
In [12]: loss = model.history.history['loss']
sns.lineplot(x=range(len(loss)),y=loss)
plt.title("Training Loss per Epoch");
```



In [ ]: