

# CS 4361 – MACHINE LEARNING: EMOTION CLASSIFICATION NEURAL-NETWORK

Team Members: Adrian Martinez, Angel Armendariz, Joshua Salas

# INTRO

- The goal of our project was to predict the emotion or mood a song is portraying.
- We trained our model on data from Spotify playlists associated with four specific emotions: happy, sad, hype, and calm.
- This work has potential applications in music recommendation systems, music therapy, and personalized playlists.

# DATA COLLECTION

- We collected data from Spotify playlists using the Spotify API.
- For each song, we collected features such as tempo, loudness, energy, danceability, key, instrumentalness, and valence.

```
# Function to convert playlist data to CSV
def playlists_to_csv(playlist):
    # Get Spotify API token
    token = spotify_integration.get_token()
    # Get playlist URI from playlist URL
    pl_uri = spotify_integration.get_uri(playlist[1])
    # Get tracks from playlist
    tracks = spotify_integration.get_playlist_tracks(token,
pl_uri)
    # Format track data for CSV
    formatted_dataframe =
spotify_integration.format_track_data_for_csv(token, tracks)
    # Add emotion column to dataframe
    formatted_dataframe['emotion'] = playlist[0]
    # Define CSV filename
    csv_df_filename = playlist[0] + "Data.csv"
    # Save dataframe to CSV
    formatted_dataframe.to_csv(csv_df_filename)

# Convert playlist to CSV
for playlist in emotion_playlists_https:
    playlists_to_csv(playlist)
```

# DATA PREPROCESSING

- We preprocessed the data by performing one-hot encoding for the 'key' feature and label encoding for the 'emotion' feature.
- The "emotion" feature was label encoded, converting the emotion categories into numerical labels
- We then normalized the features using the StandardScaler from sklearn to ensure that all features contribute equally to the model's predictions.

```
# Perform one-hot encoding on the 'key' column
combined_data = pd.get_dummies(combined_data, columns=['key'])

# Perform label encoding on the 'emotion' column
label_encoder = LabelEncoder()
combined_data['emotion'] =
label_encoder.fit_transform(combined_data['emotion'])
```

# MODEL BUILDING AND TRAINING

- We chose the Multi-Layer Perceptron (MLP) classifier as our machine learning model.
- The MLP had a single hidden layer of size 100.
- We trained the model with a maximum of 2000 iterations for the solver to converge.
- The learning rate was set to be adaptive, the activation function was logistic (sigmoid), and the alpha (regularization parameter) was 0.00

```
# Define the MLP classifier
mlp = MLPClassifier(hidden_layer_sizes=(100), # one hidden layer of 100 neurons
                    learning_rate='adaptive', # adaptive learning rate
                    activation='logistic',     # logistic (sigmoid) activation function
                    random_state=42,          # random state for reproducibility
                    max_iter=2000,            # maximum of 2000 iterations for the solver
                    alpha=0.001)              # regularization parameter

# Train the model on the scaled features and labels
mlp.fit(X_scaled, y)
```

# MODEL EVALUATION

- The model's performance was evaluated on both the training and testing data.
- The training score achieved was 0.895734, indicating that the model was able to accurately predict the emotion of a song in approximately 89.57% of cases in the training set.
- The testing score was 0.823899, meaning the model accurately predicted the emotion of a song in approximately 82.39% of cases in the testing set.

```
# Evaluate the model
train_score = mlp.score(X_train, y_train)
test_score = mlp.score(X_test, y_test)

print(f"Training score: {train_score}\nTesting score: {test_score}")
```

# CONCLUSION AND FUTURE WORK

- Our project successfully built a machine learning model that can predict the emotion or mood a song is portraying with a high degree of accuracy.
- The model was trained on data from Spotify playlists, each associated with a specific emotion: happy, sad, hype, and calm.
- The training score and the testing score indicate that the model is performing well and is able to generalize to unseen data.
- For future work, we could consider expanding the range of emotions, using more diverse datasets, or experimenting with different machine learning models to further improve the model's performance.