Search

Home

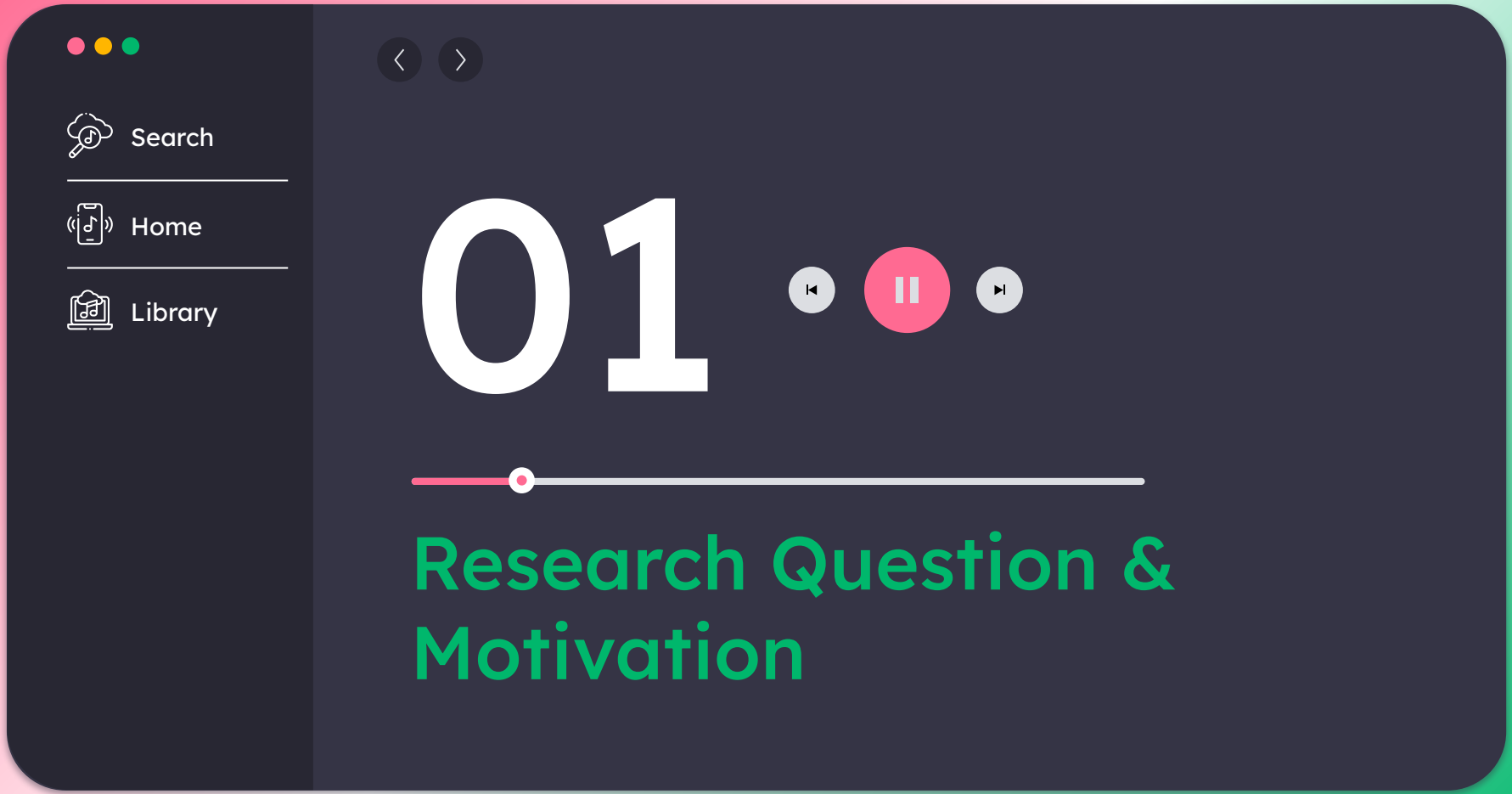
Library

Decoding Emotion in Music:

Analyzing the Correlation between Audio Features and Emotions Using ML

MACS 30100 Jiaming (Vera) Mao

Link to Video Presentation


Search
Home
Library
01
Research Question &
Motivation

Related Literatures

Chanda & Levitin (2013):

- Decreases Anxiety: Lowers Cortisol
- Improves Mood
- Boots Immunity: Associated with immunoglobulin, an antibody linked to immunity, as well as higher counts of cells that fight germs and bacteria.

Kraus & Chandrasekaran (2010):

- 15 months of intense music training improved auditory and motor skills.
- Children who are musically trained have a better vocabulary in their native language and a greater reading ability.

Related Literatures

Table 2. Efficiency of various regression algorithms.

	MAE of valence	MAE of arousal	Ac of valence (%)	Ac of arousal (%)	Ac of classification (%)
MARS	0.2826	0.2776	61.4	70	56
SVM	0.2333	0.2173	71	82	63
RBFR	0.1987	0.1803	72.4	80	64

Xia, Yu, and Fumei Xu (2022):

Music Emotion Database: Built using Thayer's two-dimensional emotion plane, categorizing music into five emotion types with continuous perception.

Regression-Based Classification: Uses artificial labeling and regression methods, with training and testing phases.

Algorithm Performance: Support vector machine, fuzzy K-neighborhood, and Fisher linear discrimination exceed 80% accuracy.

Improved Accuracy: The "mixed classifier" achieves 84.9% accuracy, outperforming traditional methods.

Research Question

Can machine learning accurately classify songs' emotional category based on its musical features, and which features contribute most to this prediction?


Search
Home
Library
02
Data Collection &
Data Wrangling

Data Collection Progress

Download

Connect to `tags.db` and extract matching **Tags' last.fmID** based on emotion mapping

Word2Vc optimization for **emotional words mapping**

last.fm +

Extract matching tracks from `metadata.db` using last.fmID, getting **tracks' titles and artists' names**

Download

Spotify for Developers

Search SpotifyIDs by artists and titles and Store Spotify **SpotifyIDs with labeled emotions**

Collect

Generate Spotify track ids and audio features dataset from `spotify.sqlite` by matching SpotifyIDs.

kaggle

Features: track_id, acousticness, danceability, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time_signature, valence

Word2Vec Optimization

Objectives:

- Extract meaningful emotion-related words from music tags.
- Use Word2Vec embeddings to measure similarity to core emotions.

Steps:

1. Retrieve music tags from the database.
2. Load Word2Vec GoogleNews model for word embeddings.
3. Compute tag embeddings by averaging word vectors.
4. Compute the cosine similarity between each tag and the five core emotions: happy, sad, angry, fear, surprise.
5. Perform word frequency analysis on the selected tags to generate raw emotion mappings

Connect & Extract TagID using Emotional Mapping

```
emotion_mapping = {
    "happy": {
        "love", "good", "great", "nice", "happy", "awesome", "beautiful", "fun", "amazing", "dance",
        "sweet", "better", "smile", "fantastic", "joy", "wonderful", "cheerful", "pleasure",
        "upbeat", "cool", "positive", "perfect", "uplifting", "exciting", "alive", "hopeful",
        "bright", "sunny", "laugh", "merry", "delight", "lucky", "sunshine", "enjoy", "bliss",
        "cheery", "satisfying", "warm", "friendly", "peaceful", "thrilled", "comforting",
        "relaxing", "soothing", "loved", "calm", "bubbly", "grateful", "lovely", "harmonious",
        "glorious", "cheer", "exhilarating", "carefree", "serene", "jolly", "radiant",
        "heartwarming", "charming", "affectionate", "giddy", "ecstatic", "gleeful", "rejoice",
        "euphoria", "miracle", "pleasant", "refreshing", "heavenly", "blessed", "celebrate",
        "adore", "cheers", "fascinating", "playful", "motivating", "appreciate", "energetic",
        "laughing", "wholesome", "enchanting", "lively", "optimistic", "thrill", "joyful",
        "excited", "electrifying", "joyful", "interested", "proud", "accepted", "powerful",
        "peaceful", "intimate", "optimistic"
    },
    "sad": {
        "sad", "sorrow", "heartbreak", "melancholy", "melancholic", "tragic", "pain", "painful",
        "heartache", "bittersweet", "mournful", "grief", "depressed", "depression", "lonely",
        "loneliness", "despair", "emptiness", "longing", "misery", "regrets", "hopeless", "hurting",
        "unbearable", "aching", "suffer", "desolate", "disappointment", "devastating",
        "tortured", "lament", "lamentation", "despairing", "dismal", "weep", "weeping", "tear",
        "tears", "tearful", "crying", "sob", "sobbing", "heartbroken", "loss", "shattered",
        "mourn", "mourning", "heartwrenching", "wistful", "nostalgia", "nostalgic", "bleak",
        "devastation", "solemn", "ache", "agony", "pity", "gloomy", "grief-stricken",
        "somber", "unsettling", "troubling", "poignant", "wretched", "despondent", "mournfully",
        "sorrowing", "dejected", "disheartened", "dispirited", "guilty", "abandoned", "despair",
        "depressed", "lonely", "bored"
    },
    "angry": {
        "hate", "die", "bitch", "damn", "kill", "mad", "angry", "brutal", "rage", "killing", "riot",
        "revenge", "sucks", "scream", "brutality", "furious", "fucked", "fury", "chaotic", "violent",
        "attack", "wild", "aggressive", "harsh", "fight", "evil", "cruel", "sadistic", "hopeless",
        "suicidal", "shout", "dumb", "stupid", "sick", "drown", "bleeding", "nasty", "hatred",
        "insane", "venom", "hell", "bloody", "anarchy", "homicidal", "pissed", "burn", "sinister",
        "wretched", "hateful", "frenzy", "menacing", "crushing", "grudge", "screeching", "intense",
        "threatened", "hateful", "mad", "aggressive", "frustrated", "distant", "critical"
    },
    "surprise" : {
        "unexpected", "unusual", "shock", "astonishing", "unexpectedly", "amazing",
        "surprised", "stunned", "startling", "unexpectedly", "mindblowing", "unexpectedness",
        "unbelievable", "jaw-dropping", "spectacular", "incredible", "unforeseen",
        "sudden", "remarkable", "miraculous", "outstanding", "eye-opening", "breathtaking",
        "thrilling", "wonder", "wow", "unanticipated", "curious", "unexpectedness",
        "surprise", "blow", "crazy", "fantastic", "wonderful", "magic", "mystery",
        "strange", "unfamiliar", "spontaneous", "jumps", "wild", "shockingly",
        "astonished", "fluke", "random", "serendipity", "striking", "flabbergasted",
        "explosive", "spine-tingling", "extraordinary", "eye-popping", "triumph",
        "bizarre", "peculiar", "unexplored", "fresh", "mystifying", "phenomenal",
        "wondrous", "outlandish", "peculiarity", "wonderstruck", "jawdrop", "awe",
        "unorthodox", "intriguing", "offbeat", "quirky", "mesmerizing", "unprecedented",
        "unconventional", "twist", "unusualness", "baffling", "unpredictable",
        "adventure", "unaccustomed", "groundbreaking", "different", "enlightening",
        "curiosity", "discover", "elating", "suddenness", "mesmerize", "twisting",
        "unreal", "blowmind", "wowed", "unexpectedly", "overwhelming", "ecstatic",
        "amusement", "unknown", "newfound", "epiphany", "exploring", "realization"
    },
    "fear": {
        "doom", "kill", "mad", "alone", "scream", "dead", "evil", "terror", "dark", "nightmare",
        "violence", "destroy", "slowly", "dying", "bleed", "ghost", "panic", "enemy",
        "misery", "heartache", "revenge", "despair", "insane", "tremble", "satan",
        "grief", "madness", "rage", "dread", "thrill", "night", "haunting", "afraid",
        "shock", "suffocate", "danger", "disturbing", "creepy", "brutal", "eerie",
        "frightened", "shadows", "worry", "guilt", "paranoia", "creatures", "chaos",
        "distress", "weird", "horrifying", "sinister", "anxiety", "tension", "freak",
        "gloom", "shiver", "unnerving", "nervous", "uneasy", "fearful", "menacing",
        "dreadful", "apprehension", "humiliated", "rejected", "submissive", "insecure",
        "anxious", "scared"
    }
}
```

Used ChatGPT for cleaning the raw emotion by telling it to remove the words that is irrelevant to happy/sad/angry/fear/surprise and put them in emotional mapping format.

Steps:

1. Query TrackIDs (tid) based on emotion-tag mappings.

2. Join tid_tag, tids, tags to match tags with their respective track IDs.

3. LIKE query for partial tag matches.

4. Maps extracted track IDs to emotions.

tags.db last.fm

```
with sqlite3.connect('datasets/tags.db') as conn:
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM tags LIMIT 20')
    data = cursor.fetchall()
    for row in data:
        print(row[0])
conn.close()
```

✓ 0.0s

```
classic rock
Progressive rock
blues
memphis slim
pop
70s
Middle of the road
Bonjour ca va
Tony Levin
instrumental
Zappa Related
Bozzio Levin Stevens
Steve Stevens
groovy
cool
experimental
Experimental Rock
jazz fusion
hard progressive rock
Black Light Syndrome
```

Match & Match & Match

Track Metadata last.fm

- Maps Songs to Emotion Categories:
  - `{emotion: [(title, artist)]}`

Spotify API Spotify

- Search Spotify track IDs by artists and titles and Store Spotify track ids to emotion in a dataframe and CSV file

Spotify Track IDs & Audio Features

8+ M. Spotify Tracks, Genre, Audio Features

SQLite db of millions of Spotify tracks (Audio Features, Genre, Artist etc)

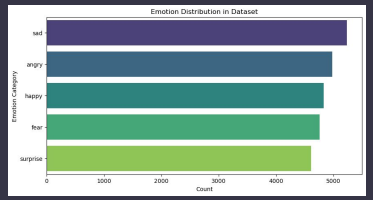
kaggle

```
track_id_to_audio_features_pd = pd.read_csv('datasets/spotify_tracks_with_features.csv')
track_id_to_audio_features_pd.head()
```

✓ 8.5s

	track_id	acousticness	danceability	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
0	2jKoVIU7VAmExKJ1Jh3w9P	0.1800	0.893	0.514	0.000000	11	0.0596	-5.080	1	0.283	95.848000	4	0.787
1	4JYUDRtPZuVNi7FAnbHyux	0.2720	0.520	0.847	0.000000	9	0.3250	-5.300	1	0.427	177.371002	4	0.799
2	6YjKAkDYmlasMqYw73iB0w	0.0783	0.918	0.586	0.000000	1	0.1450	-2.890	1	0.133	95.516998	4	0.779
3	2YlvHjDb4TyxI4A1lcDhAe	0.5840	0.877	0.681	0.000000	1	0.1190	-6.277	0	0.259	94.834999	4	0.839
4	3UOuBNEin5peSRqdzvlnWM	0.1700	0.814	0.781	0.000518	11	0.0520	-3.330	1	0.233	93.445000	4	0.536

Final Dataset



Data Size: (24411, 14)

Target Variable Distribution

- Sad: 5234
- Angry: 4978
- Happy: 4831
- Fear: 4759
- Surprise: 4609

Audio Feature	Description
Danceability	Danceability describes how suitable a track is for dancing.
Energy	Energizing music has a quick, loud, and raucous feeling. For example, death metal has high energy.
Key	the key that the song is in. Using the conventional pitch class notation, integers correspond to pitches.
Loudness	The intensity level of the sound of the track decibels (dB).
Mode	Mode indicates the modality (major or minor) of a track.
Speechiness	Speechiness identifies if lyrics are being sung on a track.
Acousticness	Acousticness refers to the confidence measure of a track, wherein a track with a CM of 0.0 to 1.0 is acoustic.
Instrumentalness	Predicts whether a track contains no vocals.
Liveness	Detects if a live audience is present on a track.
Valence	Valence refers to the positive tone of the track.
Tempo	The tempo refers to the number of beats per minute (BPM) of a track.
Time Signature	An estimated overall time signature of a track.

```
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   track_id          24411 non-null  object
 1   emotion           24411 non-null  object
 2   acousticness      24411 non-null  float64
 3   danceability      24411 non-null  float64
 4   energy            24411 non-null  float64
 5   instrumentalness  24411 non-null  float64
 6   key               24411 non-null  int64
 7   liveness          24411 non-null  float64
 8   loudness          24411 non-null  float64
 9   mode              24411 non-null  int64
 10  speechiness       24411 non-null  float64
 11  tempo             24411 non-null  float64
 12  time_signature    24411 non-null  int64
 13  valence           24411 non-null  float64
dtypes: float64(9), int64(3), object(2)
memory usage: 2.6+ MB
```

Numerical Features (Continuous & Discrete)
- Music Features (except Key, Mode, Time: categorical variables represented as discrete numerical values) are continuous values, following a standard numerical scale.

Categorical Features
- emotion (Target variable, encoded as a categorical label)
- track_id (Unique identifier, categorical but not used as a feature)

Search

Home

Library

Our Music Playlists

03

Data Transformation & Representation

Data Preprocessing - Cleaning and Splitting

Dropped irrelevant/missing data

- Removed time_signature, track IDs, and invalid tempo values (tempo = 0)

Split dataset

- 80% training, 20% testing, ensuring stratified class distribution.

Encoded target variable (emotion)

- Converted categorical labels into numerical codes.

Feature scaling

- Standardized numerical features using StandardScaler.

Search

Home

Library

Our Music Playlists

04

Model Exploration

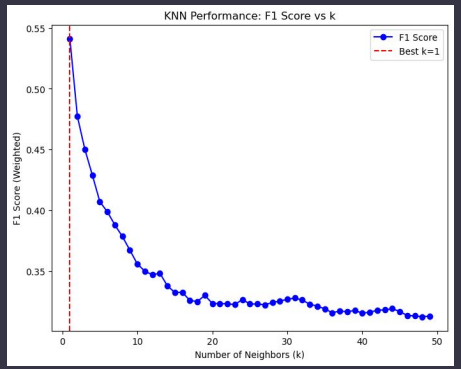
KNN

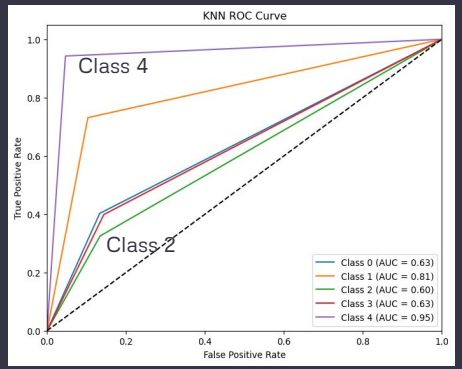
```python
# Step 1: Try different k values (1 to 50) and evaluate using F1 Score
k_values = list(range(1, 50))
best_k, best_knn_f1 = -1, -1
knn_f1_scores = []

for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred_knn = knn.predict(X_test)
    f1_knn = f1_score(y_test, y_pred_knn, average="weighted")  # Weighted F1 Score
    knn_f1_scores.append(f1_knn)

    if f1_knn > best_knn_f1:
        best_knn_f1 = f1_knn
        best_k = k

print(f"Best KNN Test F1 Score: {best_knn_f1:.4f} with k={best_k}")
```

✓ 13.0s

```
Best KNN Test F1 Score: 0.5412 with k=1
```

```
Confusion Matrix:
[[401 133 192 211  59]
 [ 73 696  81  91  11]
 [219 134 314 235  64]
 [207 135 237 417  50]
 [ 19   4  16  14 868]]
Class 0: TP=401, FP=518, FN=595, TN=3367
Class 1: TP=696, FP=406, FN=256, TN=3523
Class 2: TP=314, FP=526, FN=652, TN=3389
Class 3: TP=417, FP=551, FN=629, TN=3284
Class 4: TP=868, FP=184, FN=53, TN=3776
```

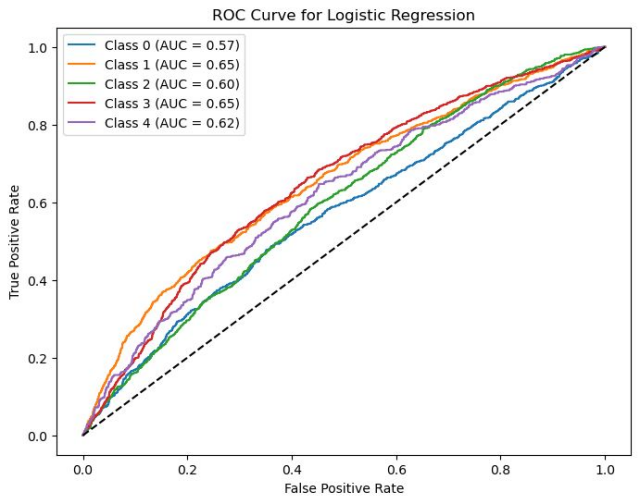




Key Takeaways:

1. Great on Class 4: Surprise (AUC 0.95)
2. Especially weak on Class 2: Angry and Happy
3. Happy (Low TP, High FN, AUC 0.60)

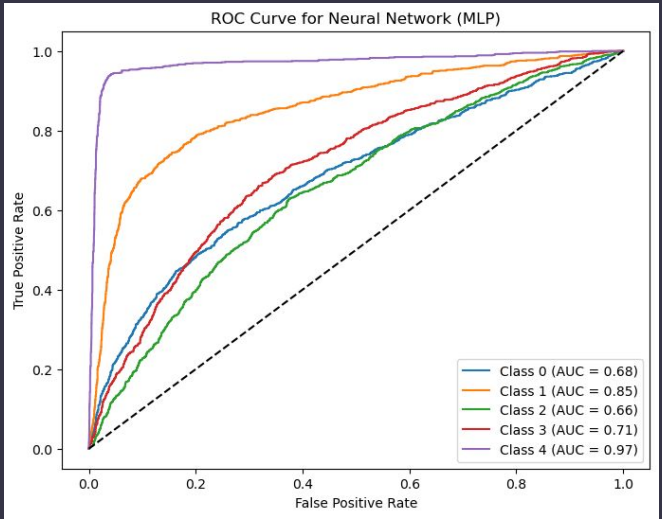
Logistic Regression Model Analysis



Key Takeaways:

1. All AUC are lower than 0.70
   a. Weak classification ability
2. Class 1 (fear) & Class 3 (sad): Highest AUC=0.65
   a. Still limited
3. Class 0 (Angry): Lowest AUC = 0.5

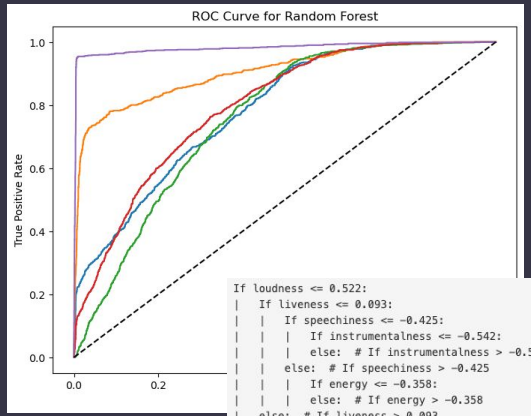
Multi-layer perceptron (MLP), NN



Key Takeaways:

1. MLP performs worse than Random Forest in some classes (especially Class 2 and 3)
2. Class 4 (surprise): AUC = 0.97
3. Class 2 (happy) remains difficult to classify.

Random Forest Model Analysis




```
If loudness <= 0.522:
|   If liveness <= 0.093:
|   |   If speechiness <= -0.425:
|   |   |   If instrumentalness <= -0.542:
|   |   |   else:  # If instrumentalness > -0.542
|   |   else:  # If speechiness > -0.425
|   |   |   If energy <= -0.358:
|   |   |   else:  # If energy > -0.358
|   else:  # If liveness > 0.093
|   |   If acousticness <= -0.762:
|   |   |   If energy <= 1.145:
|   |   |   else:  # If energy > 1.145
|   |   else:  # If acousticness > -0.762
|   |   |   If speechiness <= -0.443:
|   |   |   else:  # If speechiness > -0.443
else:  # If loudness > 0.522
|   If energy <= 0.880:
|   |   If energy <= -1.313:
|   |   |   --> Predict: sad
|   |   else:  # If energy > -1.313
|   |   |   If tempo <= -0.883:
|   |   |   else:  # If tempo > -0.883
|   else:  # If energy > 0.880
|   |   If danceability <= -0.018:
|   |   |   If instrumentalness <= -0.542:
|   |   |   else:  # If instrumentalness > -0.542
|   |   else:  # If danceability > -0.018
|   |   |   If instrumentalness <= -0.542:
|   |   |   else:  # If instrumentalness > -0.542
```

Key Takeaways:

1. Outperforms Logistic Regression in all categories, with particularly notable improvements in Class 1 (fear) and Class 4 (surprise).
2. Class 4 (surprise): AUC = 0.98
3. Class 2 (happy): AUC = 0.76

MODEL SELECTION

	Model	Macro-Average AUC Score
0	KNN	0.723871
1	Logistic Regression	0.617250
2	Random Forest	0.846089
3	Neural Network (MLP)	0.774887

0 = angry 1 = fear 2 = happy 3 = sad 4 = surprise
Random Forest is the best choice for emotion classification.

```
KNN Evaluation:
Precision: 0.5342
Recall: 0.5523
F1 Score: 0.5412

Logistic Regression Evaluation:
Precision: 0.3009
Recall: 0.3128
F1 Score: 0.2948

Random Forest Evaluation:
Precision: 0.5987
Recall: 0.6007
F1 Score: 0.5984

Neural Network (MLP) Evaluation:
Precision: 0.5206
Recall: 0.5368
F1 Score: 0.5270
```

Classification Report – KNN:

	precision	recall	f1-score	support
0	0.28	0.19	0.23	996
1	0.36	0.38	0.37	952
2	0.28	0.20	0.23	966
3	0.31	0.39	0.35	1046
4	0.35	0.45	0.39	921
accuracy			0.32	4881
macro avg	0.32	0.32	0.31	4881
weighted avg	0.32	0.32	0.31	4881

Classification Report – Logistic Regression:

	precision	recall	f1-score	support
0	0.25	0.12	0.16	996
1	0.33	0.45	0.38	952
2	0.28	0.18	0.22	966
3	0.33	0.49	0.40	1046
4	0.30	0.32	0.31	921
accuracy			0.31	4881
macro avg	0.30	0.31	0.29	4881
weighted avg	0.30	0.31	0.29	4881

Classification Report – Random Forest:

	precision	recall	f1-score	support
0	0.47	0.41	0.44	996
1	0.73	0.72	0.73	952
2	0.41	0.38	0.39	966
3	0.48	0.57	0.52	1046
4	0.94	0.95	0.95	921
accuracy			0.60	4881
macro avg	0.61	0.61	0.60	4881
weighted avg	0.60	0.60	0.60	4881

Classification Report – Neural Network (MLP):

	precision	recall	f1-score	support
0	0.43	0.39	0.41	996
1	0.60	0.69	0.64	952
2	0.35	0.29	0.32	966
3	0.42	0.41	0.41	1046
4	0.84	0.94	0.89	921
accuracy			0.54	4881
macro avg	0.53	0.54	0.53	4881
weighted avg	0.52	0.54	0.53	4881

Fine-tune Random Forest Hyperparameters

```python
# Define hyperparameter grid
param_grid = {
    'n_estimators': [50, 100, 200],  # Number of trees in the forest
    'max_depth': [10, 20, None],  # Maximum depth of the trees
    'min_samples_split': [2, 5, 10],  # Minimum number of samples required to split a node
    'min_samples_leaf': [1, 2, 4],  # Minimum number of samples required at a leaf node
    "max_features": ["sqrt", "log2"],  # Number of features to consider at each split
    "bootstrap": [True, False]  # Whether bootstrap samples are used when building trees
}
```

```python
# Perform GridSearchCV for hyperparameter tuning
grid_search = GridSearchCV(
    estimator=rf,
    param_grid=param_grid,
    scoring='f1_weighted',  # Optimize for weighted F1-score
    cv=5,  # 5-fold cross-validation
    verbose=2,
    n_jobs=-1  # Use all available processors
)
```

Best Parameters: {'bootstrap': False, 'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}

Mean CV F1 Score (Validation-Like Performance): 0.5835
The CV F1 Score (0.5835) and Test F1 Score (0.60) are similar, meaning the model is stable.

```python
final_rf = RandomForestClassifier(
    bootstrap=False,
    n_estimators=200,
    max_features="sqrt",
    min_samples_split=2,
    min_samples_leaf=1,
    max_depth=None,
    random_state=42
)
```

Model	Precision	Recall	F1 Score
Original RF	0.5987	0.6007	0.5984
Final RF	**0.6064**	**0.6011**	**0.6032**

- **Disabling bootstrap (bootstrap=False) removes randomness from sampling, making trees more correlated.**
- **Increasing n_estimators to 200 improves stability but adds computation time.**
- **Setting max_depth=None allows trees to grow fully, increasing risk of overfitting.**

Model Comparison - MLP & RF

Metrics Wise:

- Random Forest is better-performing in this case with higher F1 score
  - MLP could be improved with hyperparameter tuning (e.g., adjusting hidden layers, learning rate).

Class Wise: Both models struggle with Class 2 (Happy).

	Model	Precision	Recall	F1 Score
0	MLP	0.520567	0.536775	0.526978
1	Random Forest	0.606410	0.601106	0.603223

Model	Strengths	Weaknesses
Random Forest	✅ Handles structured numerical features well (like tempo, loudness, energy). ✅ Provides better performance for most emotion classes. ✅ More robust to noisy or redundant features.	❌ Struggles with complex decision boundaries (e.g., subtle variations in "happy" vs. "sad"). ❌ Can be computationally expensive with large trees.
MLP (Neural Network)	✅ Captures non-linear relationships (useful if emotions have complex dependencies). ✅ Potentially better with more training data and hyperparameter tuning.	❌ Underperforms on small datasets (not enough training samples to learn deep patterns). ❌ More sensitive to feature scaling and hyperparameters.

Search

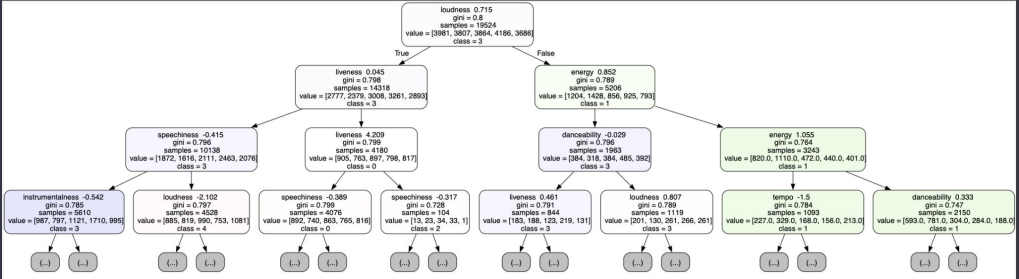
Home

Library

05

Model Performance

Model Performance - Decision Tree




```
If loudness <= 0.715:
|   If liveness <= 0.045:
|   |   If speechiness <= -0.415:
|   |   |   If instrumentalness <= -0.542:
|   |   |   else:  # If instrumentalness > -0.542
|   |   else:  # If speechiness > -0.415
|   |   |   If loudness <= -2.102:
|   |   |   else:  # If loudness > -2.102
|   else:  # If liveness > 0.045
|   |   If liveness <= 4.209:
|   |   |   If speechiness <= -0.389:
|   |   |   else:  # If speechiness > -0.389
|   |   else:  # If liveness > 4.209
|   |   |   If speechiness <= -0.317:
|   |   |   else:  # If speechiness > -0.317
else:  # If loudness > 0.715
|   If energy <= 0.852:
|   |   If danceability <= -0.029:
|   |   |   If liveness <= 0.461:
|   |   |   else:  # If liveness > 0.461
|   |   else:  # If danceability > -0.029
|   |   |   If loudness <= 0.807:
|   |   |   else:  # If loudness > 0.807
|   else:  # If energy > 0.852
|   |   If energy <= 1.055:
|   |   |   If tempo <= -1.500:
|   |   |   else:  # If tempo > -1.500
|   |   else:  # If energy > 1.055
|   |   |   If danceability <= 0.333:
|   |   |   else:  # If danceability > 0.333
```

- Loudness: first split (high-energy | low-energy)

- Liveness and speechiness (studio recordings | live)

- Energy and danceability (upbeat | intense | calm music)

- Tempo (fast-paced happy | slower sad tracks)

- Some features (e.g., loudness & energy) may be redundant, and interactions between features (e.g., speechiness + liveness) could improve classification.

- ❌Feature Selection - after removing, the accuracy score is lower


Search
Home
Library
06
Error Analysis &
Model Comparison

Error Samples - Random Forest

	True Label	Predicted Label	Index
1612	1	2	3979
1403	0	3	3444
974	2	0	2354
1055	0	3	2549
307	3	0	715
432	2	0	1042
1464	2	3	3583
598	0	3	1453
1087	2	3	2626
305	0	3	710
367	2	0	863
1610	3	2	3974
1804	3	0	4533
1480	2	3	3628
1107	3	2	2672
1355	3	0	3328
69	2	3	137
1606	2	0	3960
438	0	2	1052
135	2	3	296

Class 3 (Sad) -> Class 2 (Happy) and Class 0 (Angry)

- Energy and valence overlap, meaning sad songs with rhythmic elements or higher tempo get mistaken for happier or aggressive tracks.

Class 2 (Happy) -> Class 0 (Angry) and Class 3 (Sad)

- High-energy happy songs may resemble intense angry music, while low-energy happy songs resemble melancholic ones.

Class 1 (Fear) -> Class 2 (Happy)

- Some fear-inducing tracks have instrumental and tempo similarities to neutral/happy music.

Why These Errors Occur?

- Feature Overlap in Emotional Classes
- Limited Depth of Decision Trees
- Hard Class Boundaries

Error Samples - MLP

	True Label	Predicted Label	Index
1576	3	2	3386
771	4	0	1602
1192	3	0	2510
1137	2	0	2383
56	2	3	100
845	0	3	1774
1041	1	2	2180
1319	4	2	2803
111	1	3	222
705	1	3	1486
1283	3	0	2724
700	2	1	1474
196	3	2	386
719	3	2	1514
1053	2	4	2209
2070	3	2	4489
1814	2	1	3884
1829	1	3	3933
350	2	1	702
838	2	0	1761

Class 3 (Sad) -> Class 2 (Happy) and Class 0 (Angry)
Class 2 (Happy) -> Class 0 (Angry) and Class 3 (Sad)
Class 1 (Fear) -> Multiple Class
Why These Errors Occur?

- MLP Struggles with Feature Weighting
  - MLP learns non-linear patterns, but it may not prioritize the right features.
- Feature Overlap & Hidden Representation Limitations
  - Some emotions share similar tempo, valence, and danceability, making it harder for MLP to form distinct feature representations.
- Lack of Sequential Understanding
  - MLP treats input as static, meaning it doesn't account for how emotions evolve within a song, unlike an RNN-based model.

Shared Errors

emotion	Random Forest Errors	MLP Errors	Shared Errors
0	577	602	457
1	257	291	218
2	624	713	487
3	498	605	381
4	47	64	45

Observation	MLP (Neural Network) Misclassification Trend	Random Forest Misclassification Trend
Class 3 (Sad) Misclassification	Frequently misclassified as Class 2 (Happy)	More evenly distributed errors, but often confused with Class 0 (Angry)
Class 2 (Happy) Misclassification	Often misclassified as Class 0 (Angry) or Class 4 (Surprise)	More frequently mistaken for Class 0 (Angry)
Class 1 (Fear) Misclassification	Spread across multiple classes (Happy, Sad, Surprise)	Primarily misclassified as Class 2 (Happy)
Class 4 (Surprise) Misclassification	Mistaken for Class 0 (Angry) due to high energy/loudness	More confidently classified, fewer errors
Class 0 (Angry) Misclassification	Confused with both Class 3 (Sad) and Class 2 (Happy)	More structured misclassification, often confused with Class 3 (Sad)

Why Differences?

Factor	MLP (Neural Network)	Random Forest
Decision Boundaries	Learns complex, non-linear patterns	Creates clear, hierarchical rules based on feature importance
Flexibility	Adapts to complex data but may overfit	More robust to noise but struggles with high-dimensional interactions
Feature Interpretability	Harder to interpret weight contributions	Easier to see which features drive decisions
Handling of Overlapping Features	More likely to mix emotions due to feature similarity	Struggles with subtle variations but avoids extreme misclassifications
Error Distribution	More spread-out errors , meaning it struggles to form well-defined class separations	Errors are structured but might be biased by the model's rigid splits

How to Improve Model Performance?

For MLP:

- **Improve Feature Engineering** → Add melodic progression, spectral contrast, and chord transitions to differentiate similar emotions.
- **Optimize Hyperparameters** → Adjust hidden layers, dropout, and activation functions to refine feature representation
- **Use Hybrid Models** → Combine MLP with CNNs or attention-based networks for better pattern extraction.

For Random Forest:

- **Optimize Feature Selection** → Reduce redundant features (e.g., loudness vs. energy) to refine decision boundaries.
- **Tune Hyperparameters** → Increase max_depth, min_samples_split, or try XGBoost for better adaptability.
- **Use Post-Processing Techniques** → Apply error correction algorithms or ensemble MLP+RF for balanced performance.

Search

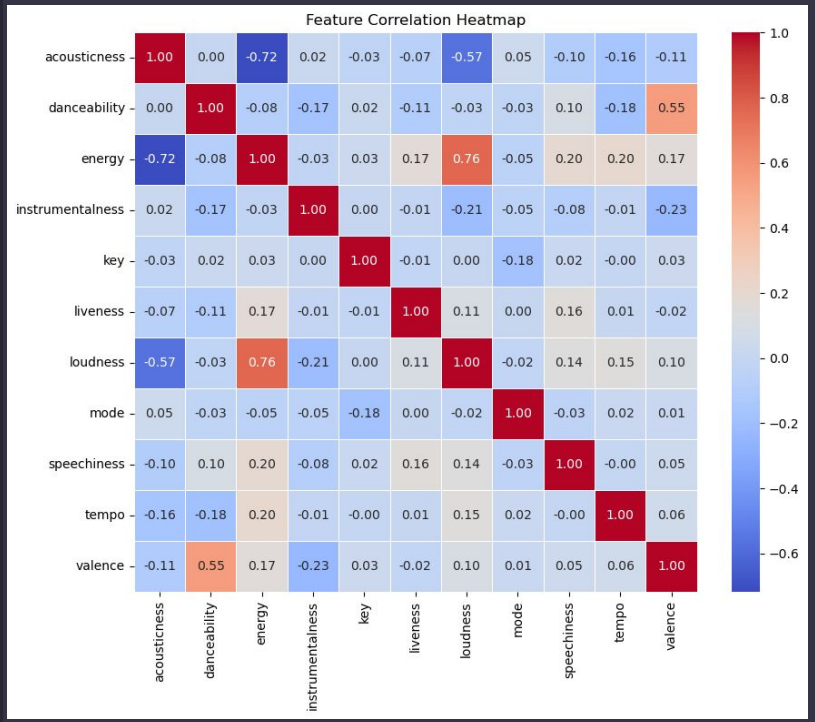
Home

Library

06

Result Interpretation

Feature Correlation



Loudness & Energy (+0.76)
- Louder songs tend to have higher energy, reinforcing their role as key emotion predictors.

Danceability & Valence (+0.55)
- More danceable songs are often happier, confirming their importance in classification.

Acousticness & Energy (-0.72)
- Acoustic tracks are usually lower-energy, aligning with calmer emotions.

Key & Mode Have Weak Correlation with Emotion
- Explains why they ranked low in feature importance across models.

Feature Importance - Comparison

	Feature	Importance Score
3	instrumentalness	0.089558
7	mode	0.093586
8	speechiness	0.114996
5	liveness	0.121817
0	acousticness	0.133088
9	tempo	0.150516
4	key	0.176275
6	loudness	0.179379
2	energy	0.181330
1	danceability	0.181354
10	valence	0.189988

Permutation Based Feature Importance for MLP

- Higher importance to key and liveness - suggests a focus on abstract musical relationships rather than structured feature splits.
- Speechiness - emphasizes how vocal presence affects emotion, aligning with non-linear dependencies.
- Instrumentalness - captures subtle effects distinguishing instrument-heavy vs. vocal-heavy music, reinforcing MLP's reliance on complex feature interactions.

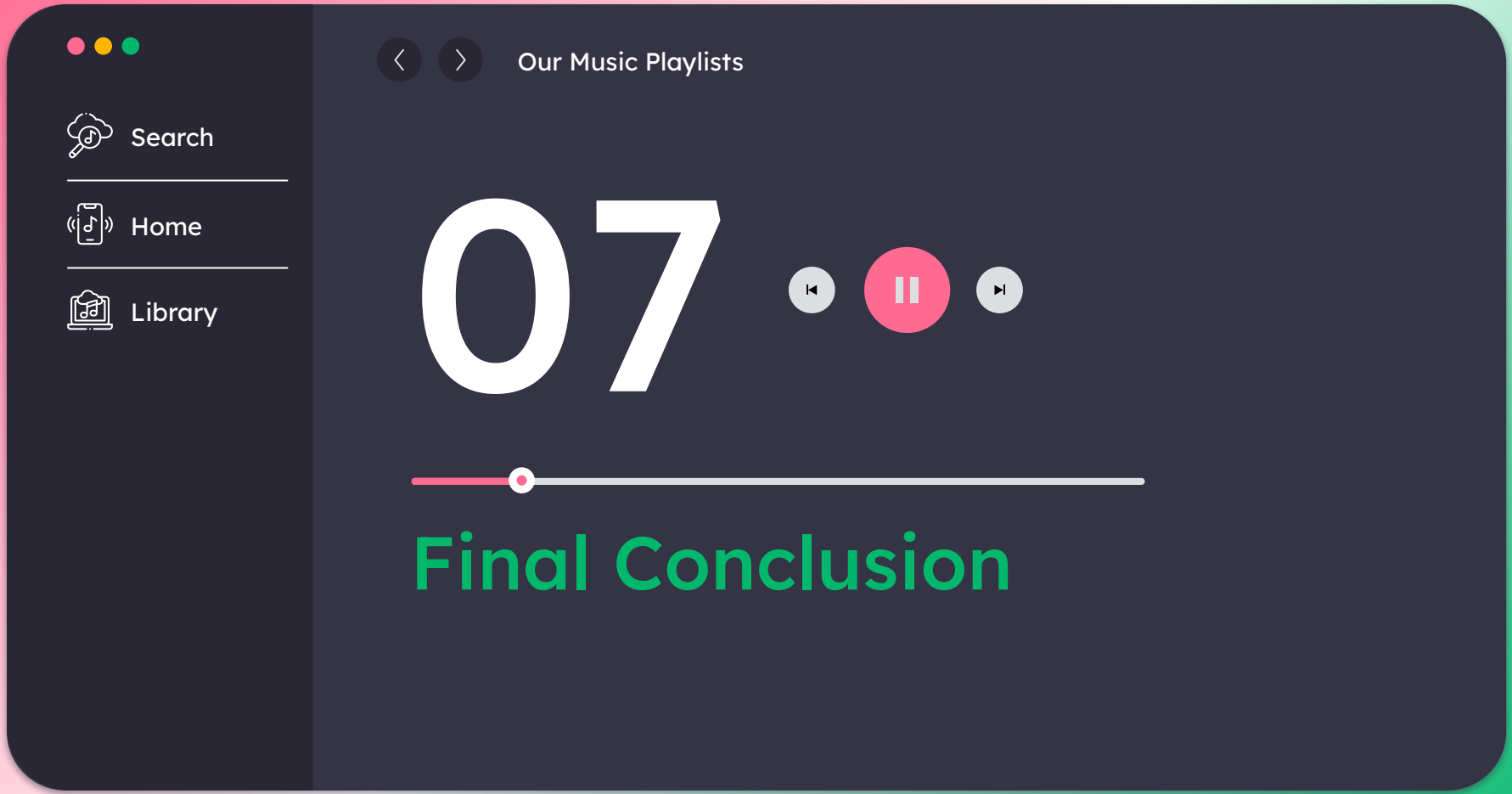
	Feature	Importance Score
7	mode	0.016996
4	key	0.062349
3	instrumentalness	0.085662
5	liveness	0.098874
9	tempo	0.099834
10	valence	0.104498
1	danceability	0.104863
2	energy	0.104942
6	loudness	0.104969
8	speechiness	0.107132
0	acousticness	0.109881

Feature Importance for Random Forest

- Favors tempo and mode - prioritizes structured, rule-based splits, aligning with its hierarchical decision-making process.
- Tempo & Mode - key differentiators in emotion classification, reinforcing RF's reliance on explicit, interpretable thresholds.

Key Feature Influence in Different Models

Feature	RF Impact	MLP Impact	Key Observations
Energy ⚡	**High**	**Highest**	Most predictive feature, strongly linked to excitement/intensity.
Valence 🎭	Moderate	**High**	Helps distinguish positive (happy, surprise) vs. negative (sad, fear) emotions.
Danceability 💃	**High**	Moderate	RF uses it for rule-based classification, MLP finds non-linear trends.
Loudness 🎚️	**High**	Moderate	Strongly affects angry, happy, surprise, but overlaps with energy.
Speechiness 🗣️	Moderate	**High**	Helps distinguish spoken-word music, but MLP uses it more flexibly.
Acousticness 🎼	Low	Moderate	Weak predictor alone, but important in multi-feature interactions.
Mode (Major/Minor) 🎵	Low	Low	Has some effect, but doesn't strongly determine emotion alone.
Tempo (BPM) 🥁	Low	Low	Faster songs aren’t always happier–ML struggles with this nuance.


Search
Home
Library
Our Music Playlists
07
Final Conclusion

Final Analysis: Confirmed vs. Challenged Theories

Confirmed Theories

Loudness & Energy are Key Predictors

- Feature importance and correlation confirmed that louder, high-energy tracks align with stronger emotions (e.g., happy, angry, surprise).

Danceability & Valence Influence Emotional Perception

- Correlation analysis showed danceability positively relates to happiness, validated by feature importance in MLP.
- MLP leveraged valence more effectively, capturing subtler distinctions than RF's rule-based approach.

Challenged Theories

Feature Importance is Model-Dependent

- RF prioritized tempo & mode, while MLP weighted speechiness & valence higher–showing different models extract meaning in unique ways.

Fear & Sadness Are Not Always Distinct

- Both RF & MLP misclassified fear into multiple categories (happy, sad, surprise).

Happy & Angry Songs Share More Similarities Than Expected

- High-energy happy tracks were frequently confused with angry ones, especially in MLP.

Final Analysis: Unexpected & Key Takeaways

Interesting Findings

MLP Captures Complex Emotional Blends

- Unlike RF, MLP doesn't rely on rigid splits, making it better at handling songs that blend multiple emotions.

Shared Errors Reveal Overlapping Features

- Both models struggled with Happy vs. Sad and Angry vs. Surprise, reinforcing that some emotional categories lack clear musical separation.

Key Takeaways:

Random Forest is the Best Performing Model

- Highest F1 score, best feature interpretability, and strong rule-based classification make RF the most reliable model.

Feature Selection & Model Choice Matter

- Structured features (tempo, mode) favor RF, while abstract relationships (valence, speechiness) benefit MLP.

Energy & Valence are Key, but Not Always Enough

- Emotion perception is multi-dimensional–adding lyrical content or contextual metadata may improve classification.

Connect & Extract TagID using Emotional Mapping

```
emotion_mapping = {
    "happy": {
        "love", "good", "great", "nice", "happy", "awesome", "beautiful", "fun", "amazing", "dance",
        "sweet", "better", "smile", "fantastic", "joy", "wonderful", "cheerful", "pleasure",
        "upbeat", "cool", "positive", "perfect", "uplifting", "exciting", "alive", "hopeful",
        "bright", "sunny", "laugh", "merry", "delight", "lucky", "sunshine", "enjoy", "bliss",
        "cheery", "satisfying", "warm", "friendly", "peaceful", "thrilled", "comforting",
        "relaxing", "soothing", "loved", "calm", "bubbly", "grateful", "lovely", "harmonious",
        "glorious", "cheer", "exhilarating", "carefree", "serene", "jolly", "radiant",
        "heartwarming", "charming", "affectionate", "giddy", "ecstatic", "gleeful", "rejoice",
        "euphoria", "miracle", "pleasant", "refreshing", "heavenly", "blessed", "celebrate",
        "adore", "cheers", "fascinating", "playful", "motivating", "appreciate", "energetic",
        "laughing", "wholesome", "enchanting", "lively", "optimistic", "thrill", "joyful",
        "excited", "electrifying", "joyful", "interested", "proud", "accepted", "powerful",
        "peaceful", "intimate", "optimistic"
    },
    "sad": {
        "sad", "sorrow", "heartbreak", "melancholy", "melancholic", "tragic", "pain", "painful",
        "heartache", "bittersweet", "mournful", "grief", "depressed", "depression", "lonely",
        "loneliness", "despair", "emptiness", "longing", "misery", "regrets", "hopeless", "hurting",
        "unbearable", "aching", "suffer", "desolate", "disappointment", "devastating",
        "tortured", "lament", "lamentation", "despairing", "dismal", "weep", "weeping", "tear",
        "tears", "tearful", "crying", "sob", "sobbing", "heartbroken", "loss", "shattered",
        "mourn", "mourning", "heartwrenching", "wistful", "nostalgia", "nostalgic", "bleak",
        "devastation", "solemn", "ache", "agony", "pity", "gloomy", "grief-stricken",
        "somber", "unsettling", "troubling", "poignant", "wretched", "despondent", "mournfully",
        "sorrowing", "dejected", "disheartened", "dispirited", "guilty", "abandoned", "despair",
        "depressed", "lonely", "bored"
    },
    "angry": {
        "hate", "die", "bitch", "damn", "kill", "mad", "angry", "brutal", "rage", "killing", "riot",
        "revenge", "sucks", "scream", "brutality", "furious", "fucked", "fury", "chaotic", "violent",
        "attack", "wild", "aggressive", "harsh", "fight", "evil", "cruel", "sadistic", "hopeless",
        "suicidal", "shout", "dumb", "stupid", "sick", "drown", "bleeding", "nasty", "hatred",
        "insane", "venom", "hell", "bloody", "anarchy", "homicidal", "pissed", "burn", "sinister",
        "wretched", "hateful", "frenzy", "menacing", "crushing", "grudge", "screeching", "intense",
        "threatened", "hateful", "mad", "aggressive", "frustrated", "distant", "critical"
    },
    "surprise" : {
        "unexpected", "unusual", "shock", "astonishing", "unexpectedly", "amazing",
        "surprised", "stunned", "startling", "unexpectedly", "mindblowing", "unexpectedness",
        "unbelievable", "jaw-dropping", "spectacular", "incredible", "unforeseen",
        "sudden", "remarkable", "miraculous", "outstanding", "eye-opening", "breathtaking",
        "thrilling", "wonder", "wow", "unanticipated", "curious", "unexpectedness",
        "surprise", "blow", "crazy", "fantastic", "wonderful", "magic", "mystery",
        "strange", "unfamiliar", "spontaneous", "jumps", "wild", "shockingly",
        "astonished", "fluke", "random", "serendipity", "striking", "flabbergasted",
        "explosive", "spine-tingling", "extraordinary", "eye-popping", "triumph",
        "bizarre", "peculiar", "unexplored", "fresh", "mystifying", "phenomenal",
        "wondrous", "outlandish", "peculiarity", "wonderstruck", "jawdrop", "awe",
        "unorthodox", "intriguing", "offbeat", "quirky", "mesmerizing", "unprecedented",
        "unconventional", "twist", "unusualness", "baffling", "unpredictable",
        "adventure", "unaccustomed", "groundbreaking", "different", "enlightening",
        "curiosity", "discover", "elating", "suddenness", "mesmerize", "twisting",
        "unreal", "blowmind", "wowed", "unexpectedly", "overwhelming", "ecstatic",
        "amusement", "unknown", "newfound", "epiphany", "exploring", "realization"
    },
    "fear": {
        "doom", "kill", "mad", "alone", "scream", "dead", "evil", "terror", "dark", "nightmare",
        "violence", "destroy", "slowly", "dying", "bleed", "ghost", "panic", "enemy",
        "misery", "heartache", "revenge", "despair", "insane", "tremble", "satan",
        "grief", "madness", "rage", "dread", "thrill", "night", "haunting", "afraid",
        "shock", "suffocate", "danger", "disturbing", "creepy", "brutal", "eerie",
        "frightened", "shadows", "worry", "guilt", "paranoia", "creatures", "chaos",
        "distress", "weird", "horrifying", "sinister", "anxiety", "tension", "freak",
        "gloom", "shiver", "unnerving", "nervous", "uneasy", "fearful", "menacing",
        "dreadful", "apprehension", "humiliated", "rejected", "submissive", "insecure",
        "anxious", "scared"
    }
}
```

Used ChatGPT for cleaning the raw emotion by telling it to remove the words that is irrelevant to happy/sad/angry/fear/surprise and put them in emotional mapping format.

last.fm
tags.db

```
tag_to_track_ids = {}
with sqlite3.connect("datasets/tags.db") as conn:
    cursor = conn.cursor()
```

Query TrackIDs (tid) based on emotion-tag mappings.

```
with sqlite3.connect('datasets/tags.db') as conn:
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM tags LIMIT 20')
    data = cursor.fetchall()
    for row in data:
        print(row[0])
conn.close()
```

✓ 0.0s

```
classic rock
Progressive rock
blues
memphis slim
pop
70s
Middle of the road
Bonjour ca va
Tony Levin
instrumental
Zappa Related
Bozzio Levin Stevens
Steve Stevens
groovy
cool
experimental
Experimental Rock
jazz fusion
hard progressive rock
Black Light Syndrome
```

```
emotion_mapping.items():
= tid_tag.tid
= tags.ROWID
gs.tag LIKE ?" for _ in potential_names])})
{word}%" for word in potential_names])
[row[0] for row in data]
_to_track_ids.items()})
ngry': 281298, 'surprise': 128654, 'fear': 167259}
```

Join tid_tag, tids, tags to match tags with their respective track IDs.

LIKE query for partial tag matches.

Maps extracted track IDs to emotions.

Word2Vc Optimization

Compute the cosine similarity between each tag and the five core emotions: happy, sad, angry, fear, surprise.

```python
# Calculate the similarity between tags and 5 emotion core words
emotion_categories = ["happy", "sad", "angry", "fear", "surprise"]
emotion_vectors = {emo: model[emo].reshape(1, -1) for emo in emotion_categories}

# Calculate the similarity score
similarities = {emo: {} for emo in emotion_categories}
for tag, embedding in tag_embeddings.items():
    for emo, emo_vector in emotion_vectors.items():
        similarity = cosine_similarity(emo_vector, embedding.reshape(1, -1))[0][0]
        similarities[emo][tag] = similarity

# Take out the top 50000 tags for each emotion category
top_n = 50000
top_tags_per_emotion = {}
for emo in emotion_categories:
    sorted_tags = sorted(similarities[emo], key=similarities[emo].get, reverse=True)[:top_n]
    top_tags_per_emotion[emo] = sorted_tags
```

Generated raw emotion mappings for classification.

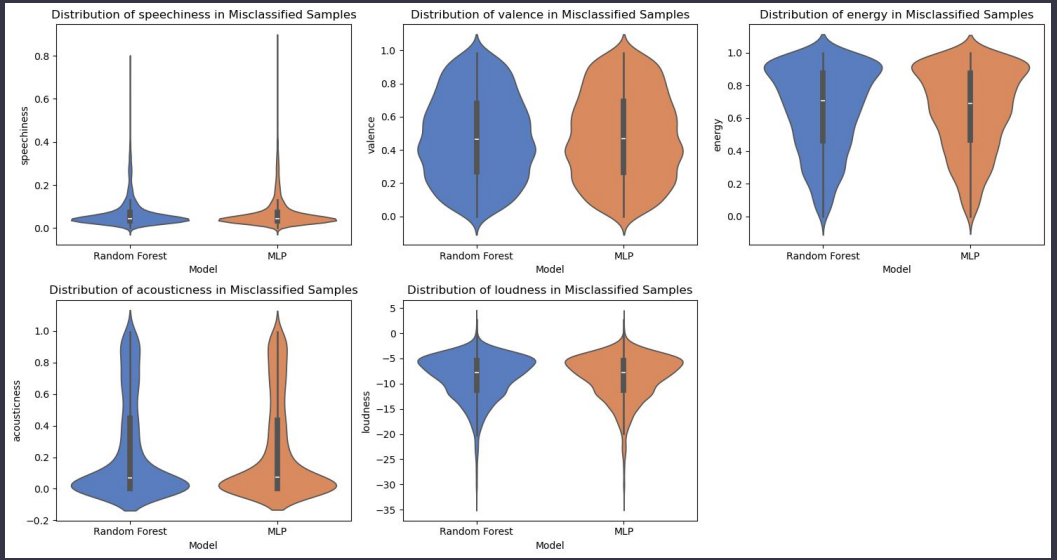
```
happy: ('love', 'I', 'good', 'song', 'like', 'songs', '-', 'great', 'want',
sad: ('love', 'I', 'song', 'like', 'good', 'songs', '-', 'music', 'great', '
angry: ('love', 'I', 'song', 'songs', 'like', '-', 'good', 'fucking', 'happy
fear: ('love', 'word', '-', 'I', 'like', 'song', 'songs', 'want', 'music', '
surprise: ('song', 'good', 'love', 'I', '-', 'like', 'one', 'ever', 'nice',
```

Perform word frequency analysis on the selected tags and remove irrelevant stopwords.

```python
# Calculate the most common words for each emotion category, and remove stopwords
filtered_top_tags = {}
for emo, tags in top_tags_per_emotion.items():
    words = [word for tag in tags for word in tag.split()]  # Split tag to single words
    filtered_words = [word for word in words if word not in stop_words]  # remove stopwords
    most_common_words = [word for word, _ in Counter(filtered_words).most_common(5000)]  # Pick 5000 high frequency words
    filtered_top_tags[emo] = most_common_words

# Generate raw emotion mapping
raw_emotion_mapping = {emo: tuple(filtered_top_tags[emo]) for emo in emotion_categories}

print("\**Generated Raw Emotion Mapping:**")
for emo, words in raw_emotion_mapping.items():
    print(f"{emo}: {words}")
```

Compare the Distribution of Errors Per Feature

- **Speechiness & Valence** → Similar misclassification patterns in both models.
- **Energy & Loudness** → MLP struggles more with high-energy, loud samples.
- **Acousticness** → Random Forest misclassifies more acoustic-heavy tracks.



Failure case analysis & Unique Misclassification for each model

Failure Case Summary:

- High energy, low valence → Anger misclassified as sadness/fear.
- Acoustic tracks → Sadness & fear confusion.
- High danceability → Surprise mistaken for happy.
- MLP struggles with instrumentals → Anger misclassified.
- Shared errors → Anger & surprise overlap.

```
(      speechiness  acousticness  valence  energy  danceability   loudness  \
 4304       0.0355         0.8130    0.253   0.238         0.199 -14.847000
 4617       0.0344         0.0216    0.707   0.822         0.301  -0.699000
 2741       0.0433         0.4390    0.671   0.220         0.614 -18.761999
 212        0.0421         0.1340    0.509   0.387         0.769  -9.905000
 3705       0.0293         0.0203    0.547   0.738         0.382  -5.118000

       instrumentalness  True Emotion  RF Predicted Emotion
 4304          0.000008             2                     3
 4617          0.002020             0                     3
 2741          0.793000             0                     2
 212           0.001430             3                     2
 3705          0.000000             0                     2  ,
       speechiness  acousticness  valence  energy  danceability  loudness  \
 691        0.0365       0.14500    0.436   0.904         0.474    -4.151
 3974       0.0476       0.01560    0.654   0.878         0.668   -10.217
 4717       0.0343       0.85200    0.258   0.276         0.596   -11.732
 819        0.0381       0.01610    0.466   0.730         0.729   -11.922
 757        0.0312       0.00371    0.495   0.776         0.484    -6.480

       instrumentalness  True Emotion  MLP Predicted Emotion
 691            0.00643             0                     3
 3974           0.68000             2                     1
 4717           0.25400             3                     0
 819            0.17000             1                     3
 757            0.10900             3                     0  )
```

	speechiness	acousticness	valence	energy	danceability	loudness	instrumentalness	True Emotion	RF Predicted Emotion	MLP Predicted Emotion
533	0.0829	0.093000	0.546	0.871	0.635	-7.045000	0.000006	1	3	3
1433	0.0314	0.046100	0.375	0.460	0.472	-9.715000	0.001780	2	0	3
376	0.2420	0.006690	0.649	0.511	0.862	-17.691999	0.000008	1	2	2
1725	0.1700	0.001280	0.459	0.935	0.427	-7.101000	0.178000	0	2	2
3109	0.0361	0.000002	0.129	0.867	0.244	-4.754000	0.001110	2	0	0

Model Performance - Linear Regression

	Feature	Coefficient
0	energy	0.185253
1	danceability	0.163019
2	valence	0.140516
3	speechiness	0.128275
4	loudness	0.091452
5	acousticness	0.090008
6	mode	0.067023
7	instrumentalness	0.061842
8	tempo	0.020973
9	liveness	0.017250
10	key	0.004028

Energy (0.185) & Danceability (0.163): strongest

- high-energy, danceable songs are more likely to be happy.

Valence (0.141): strong positive impact

- makes sense since valence measures positivity in music.

Speechiness (0.128) plays a key role

- spoken-word or rap content impacts emotional perception.

Loudness (0.091) & Acoustics (0.090): contribute moderately

- louder, less acoustic songs -> more intense and emotional.

Mode (0.067): smaller impact

- major/minor scales alone don't fully determine emotion.

Tempo (0.020) and Liveness (0.017): minimal effects

- speed and live performance setting don't heavily dictate emotions in this dataset.

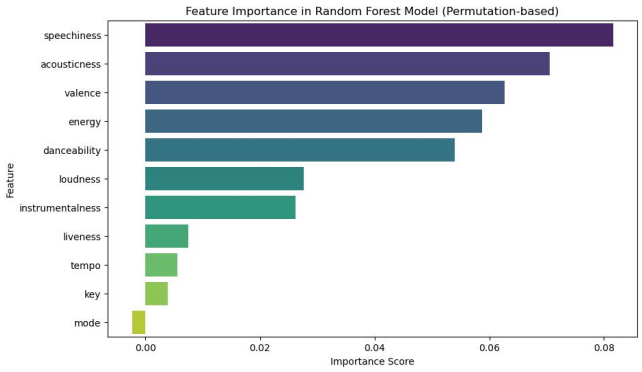
Key (0.004) is the least influential,

- specific pitch relationships aren't strongly linked to emotion classification.

Permutation Based Importance Comparison

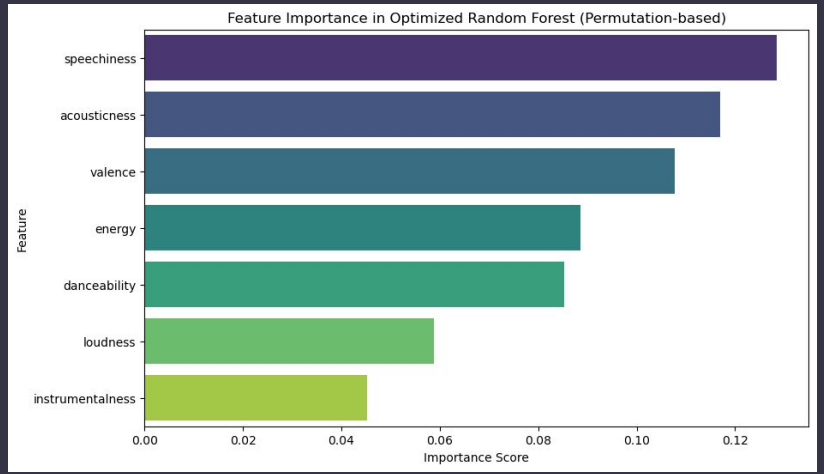
Permutation Feature Importance for Random Forest:

	Feature	Importance
8	speechiness	0.081620
0	acousticness	0.070582
10	valence	0.062667
2	energy	0.058745
1	danceability	0.053932
6	loudness	0.027586
3	instrumentalness	0.026237
5	liveness	0.007539
9	tempo	0.005582
4	key	0.003961
7	mode	-0.002340



Permutation Feature Importance – Optimized Random Forest:

	Feature	Importance
0	speechiness	0.128513
1	acousticness	0.116986
2	valence	0.107696
3	energy	0.088511
4	danceability	0.085288
5	loudness	0.058816
6	instrumentalness	0.045202



Investigate Feature Relationships and Model Understanding of Features

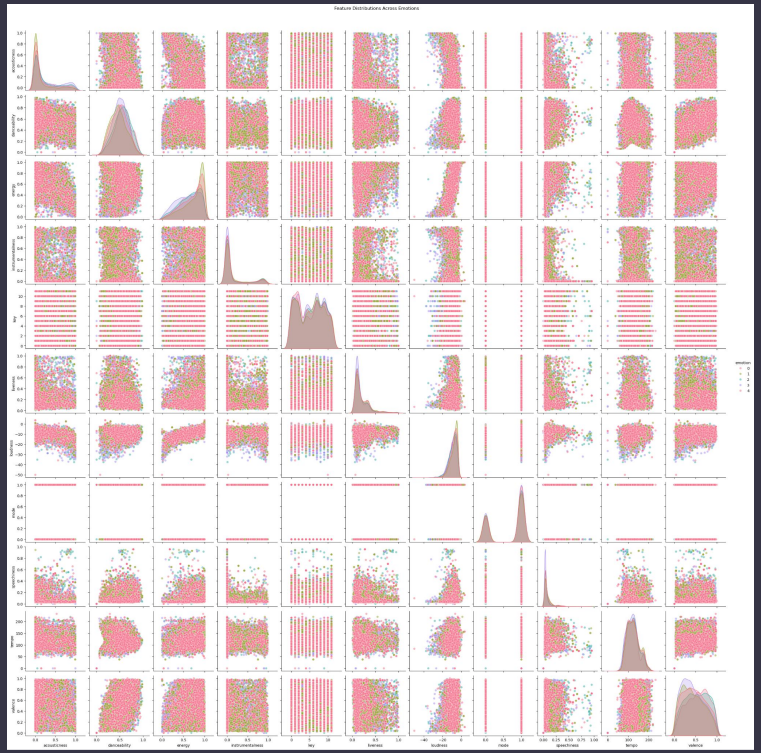
Speechiness and acousticness : distinct separation patterns

Energy and valence : clear clustering trends for high-energy emotions like anger or happiness.

Instrumentalness and loudness : diffuse relationship with emotion categories.

Danceability & Valence : evenly spread across emotions

Key and mode : variation across emotions, aligning with their low feature importance rankings.



Investigate Feature Relationships and Model Understanding of Features

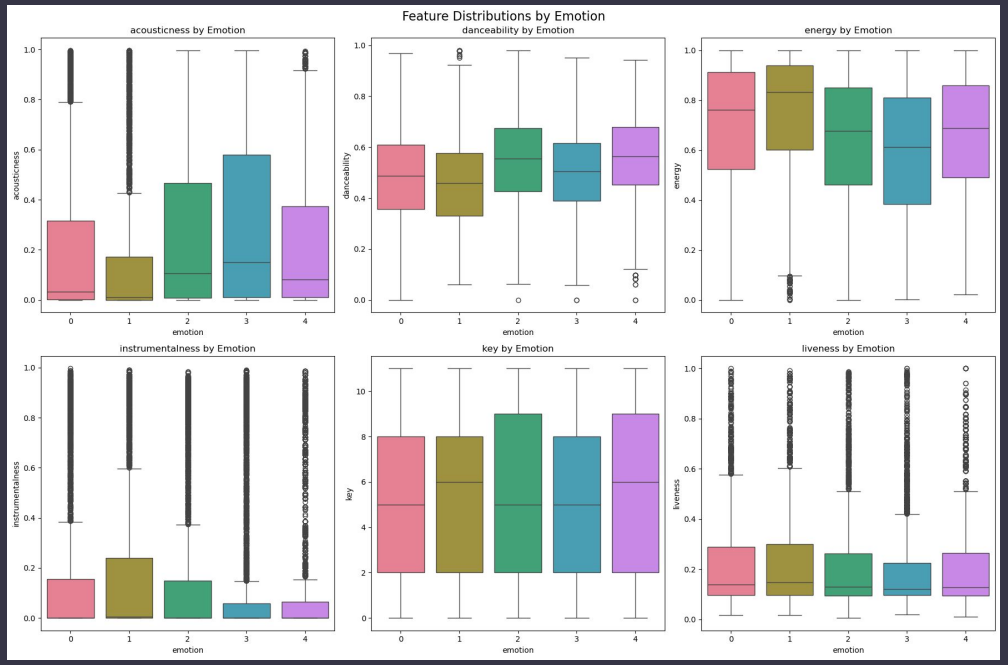
Acousticness: higher for emotions like sadness and surprise.

Energy: elevated for emotions like anger and happiness.

Danceability: stable across emotions

Instrumentalness: broader spread but is generally lower for emotions associated with speech-heavy tracks.

Liveness and key: minimal variation.



Emotion-Specific Feature Analysis

Key Takeaways

- Loudness, speechiness, and valence drive classification.
- Overlap issues exist, especially between happiness-anger and sadness-fear.

Summary:

- Happiness:
  - High energy, loudness, and valence. Misclassified if low valence or too acoustic.
- Sadness:
  - High acousticness, low energy, and valence. Overlaps with soft happy or strange fear tracks.
- Anger:
  - High speechiness, energy, and loudness. Confused with happy if valence is high.
- Fear:
  - Low loudness, high instrumentalness, and acousticness. May resemble strange sadness.
- Surprise:
  - Highly variable features. Often overlaps with happiness or fear.

Feature Importance Differences Between MLP & Random Forest

Speechiness & Acousticness:

- Crucial for Random Forest, less for MLP.

Valence & Energy:

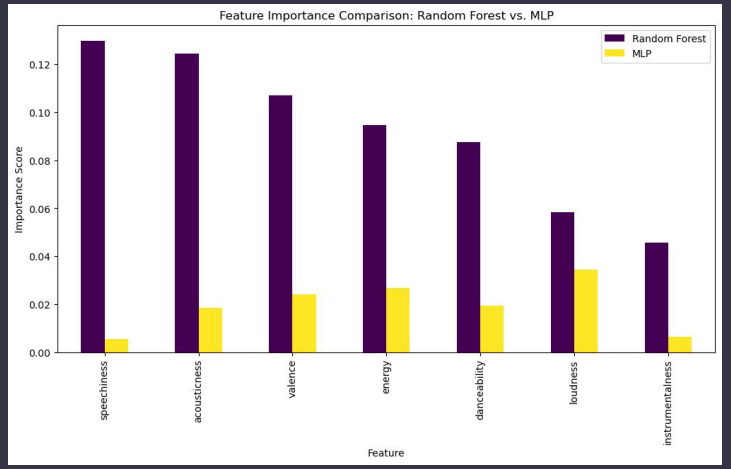
- Important for both, MLP emphasizes loudness more.

Loudness:

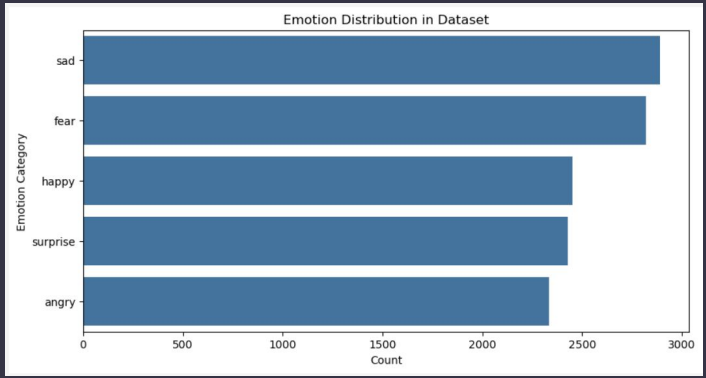
- MLP relies on it more than Random Forest.

Feature Distribution:

- Random Forest focuses on key features, MLP spreads importance.



Compare The Progress



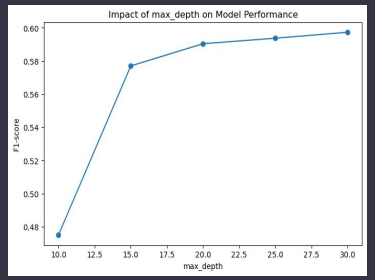

```
KNN Evaluation:
Precision: 0.2863
Recall: 0.2869
F1 Score: 0.2819

Logistic Regression Evaluation:
Precision: 0.3004
Recall: 0.3009
F1 Score: 0.2921

Random Forest Evaluation:
Precision: 0.2408
Recall: 0.2436
F1 Score: 0.2411

Neural Network (MLP) Evaluation:
Precision: 0.3030
Recall: 0.3101
F1 Score: 0.3017
```

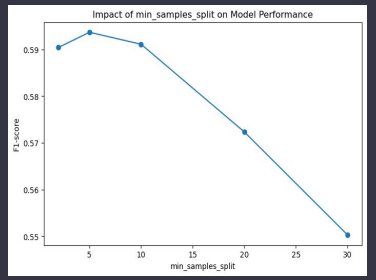
Classification Report - KNN:

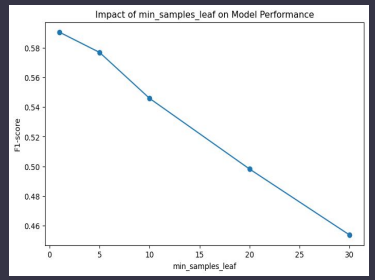
	precision	recall	f1-score	support
0	0.28	0.22	0.25	467
1	0.28	0.37	0.32	564
2	0.27	0.22	0.24	491
3	0.31	0.38	0.34	578
4	0.30	0.21	0.25	486
accuracy			0.29	2586
macro avg	0.29	0.28	0.28	2586
weighted avg	0.29	0.29	0.28	2586

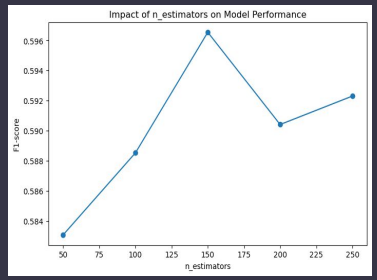
Best KNN Test F1 Score: 0.2819 with k=40

Model Justification Fine-tune Random Forest Hyperparameters










```
max_features=sqrt: F1 Score = 0.5904
max_features=log2: F1 Score = 0.5904
max_features=None: F1 Score = 0.5139
Max Features Hyperparameter Tuning Results:
max_features=sqrt: 0.5904
max_features=log2: 0.5904
max_features=None: 0.5139
```

```
n_estimators=50, F1 Score = 0.5833
n_estimators=100, F1 Score = 0.5866
n_estimators=200, F1 Score = 0.5898
n_estimators=300, F1 Score = 0.5886
n_estimators=400, F1 Score = 0.5887
n_estimators=500, F1 Score = 0.5897
```

```
min_samples_split=2, min_samples_leaf=1, F1 Score = 0.5898
min_samples_split=2, min_samples_leaf=2, F1 Score = 0.5880
min_samples_split=2, min_samples_leaf=5, F1 Score = 0.5588
min_samples_split=2, min_samples_leaf=10, F1 Score = 0.5126
min_samples_split=5, min_samples_leaf=1, F1 Score = 0.5884
min_samples_split=5, min_samples_leaf=2, F1 Score = 0.5860
min_samples_split=5, min_samples_leaf=5, F1 Score = 0.5588
min_samples_split=5, min_samples_leaf=10, F1 Score = 0.5126
min_samples_split=10, min_samples_leaf=1, F1 Score = 0.5836
min_samples_split=10, min_samples_leaf=2, F1 Score = 0.5800
min_samples_split=10, min_samples_leaf=5, F1 Score = 0.5588
min_samples_split=10, min_samples_leaf=10, F1 Score = 0.5126
min_samples_split=20, min_samples_leaf=1, F1 Score = 0.5504
min_samples_split=20, min_samples_leaf=2, F1 Score = 0.5438
min_samples_split=20, min_samples_leaf=5, F1 Score = 0.5316
min_samples_split=20, min_samples_leaf=10, F1 Score = 0.5126

Best min_samples_split: 2, Best min_samples_leaf: 1, Best F1 Score: 0.5898
```

```
max_depth=10, F1 Score = 0.4672
max_depth=20, F1 Score = 0.5931
max_depth=30, F1 Score = 0.5912
max_depth=40, F1 Score = 0.5904
max_depth=None, F1 Score = 0.5898

Best max_depth: 20, Best F1 Score: 0.5931
```

max_depth = 20, min_samples_split = 2, min_samples_leaf = 1
SAME AS the Fine-tuned Random Forest