Project Title: A machine learning-based system for music genre classification using Spotify data.

Made by:

Kavita Nampoothiri (20BEC0464)
kavitapravin.n2020@vitstudent.ac.in

Anurag Chandra (20BEC0728)
anurag.chandra2020@vitstudent.ac.in

Varsha J Varma (20BEC0336)
varshavarma.j2020@vitstudent.ac.in

S. Vignesh (20BEC0323)
vignesh.s2020a@vitstudent.ac.in

1 INTRODUCTION -

1.1 **Overview** :

The goal of this project is to develop a machine learning-based system for music genre classification. The system will analyze the audio features of songs and accurately predict their genres. This project aims to enable music streaming platforms, like Spotify, to enhance their recommendation systems and improve user experience.

1.2 **Purpose**:

The purpose of this project is to leverage machine learning-based music genre classification to achieve several important outcomes in the music industry that can accurately categorize songs into specific genres. By achieving this, the project aims to enhance the music listening experience for users by enabling personalized music recommendations, facilitating music discovery, and promoting diversity in music consumption.
By implementing this project, several achievements can be realized:

- Music streaming platforms can enhance their recommendation systems, providing users with personalized song suggestions based on their genre preferences. This leads to a more engaging and satisfying music-listening experience.
- Music enthusiasts can discover new genres and artists, expanding their musical horizons and fostering a diverse music ecosystem.
- Music industry professionals, such as music supervisors and licensing agencies, can efficiently search for and license tracks that match specific genres required for various projects, streamlining the licensing process.
- The project enables researchers and analysts to gain insights into music consumption patterns, regional preferences, and cultural trends, empowering informed decision-making and targeted marketing strategies.
- The accurate genre classification can be applied in music recognition services, aiding in identifying the genre of recognized songs and providing relevant recommendations.

2. LITERATURE SURVEY

2.1 Existing problem;

There are several approaches and methods that can be used to solve music genre classification using Spotify data. Here are some common approaches:

1. Feature Extraction and Machine Learning:

- Extract audio features from the audio files using libraries like Librosa or Essentia. These features can include mel-frequency cepstral coefficients (MFCCs), spectral contrast, chroma feature, etc.
- Utilize the metadata provided by Spotify, such as artist information, track popularity, release date, etc.
- Combine the extracted audio features and metadata into a feature vector for each track.
- Train a machine learning algorithm, such as support vector machines (SVM), random forests, or deep neural networks, using the labeled data to classify the tracks into different genres.

2. Transfer Learning with Pretrained Models:

- Utilize pre-trained deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), that were trained on large-scale music datasets or general audio data.
- Fine-tune the pre-trained models using Spotify data by replacing the last layer(s) and training the model on the genre classification task.
- This approach benefits from the representations learned by the pre-trained models, which capture complex audio patterns and can help improve classification accuracy.

3. Hybrid Approaches:

- Combine both audio features and metadata features to form a fused representation of the tracks.
- Apply dimensionality reduction techniques like Principal Component Analysis (PCA) or t-SNE to reduce the dimensionality of the feature vectors.
- Train a classifier, such as SVM or neural networks, using the fused features to classify the tracks into different genres.

4. Deep Learning with Spectrograms:

- Convert the audio files into spectrograms, which provide a visual representation of the audio signal over time.
- Utilize Convolutional Neural Networks (CNNs) or other deep learning architectures to process the spectrograms.
- Train the network using labeled data to classify the spectrograms into different genres.

It's important to note that the performance of these approaches can vary depending on the dataset, feature selection, model architecture, and hyperparameter tuning. Experimentation and fine-tuning are crucial to achieving optimal results.

2.2 Proposed solution:

Developing a machine learning-based system for music genre classification using Spotify data typically involves several steps. Here is a high-level overview of the process:

1. **Data Collection:** We were provided a dataset containing Spotify data. The dataset contained 42,306 records spanning over 24 columns. These columns contain several audio features including tempo, energy, and danceability. We also have a genre column that serves as the label which has to be done.

2. **Data Preprocessing:** Clean and preprocess the collected data. This involves handling missing values, removing duplicates or outliers, and normalizing the features if necessary. Additionally, we converted categorical genre labels into numerical representations suitable for machine learning algorithms using encoding techniques.

3. **Feature Extraction**: Extract relevant features from the audio data. Spotify data provides a wide range of audio features, including tempo, loudness, valence, etc. These features (energy, loudness,speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms) will serve as input to the machine learning model. Whereas columns like id, uri, type, etc are dropped from the training dataset for optimized results.

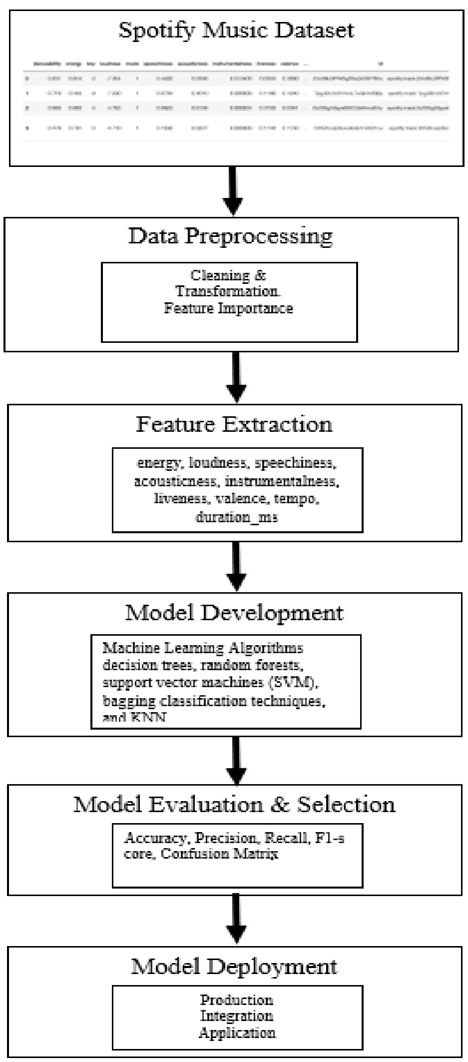
4. **Splitting the Dataset:** Divide the dataset into training, and test sets. The training set is used to train the machine learning model, and the test set is used for the final evaluation of the trained model. We split the dataset into 80:20, wherein 80% of the dataset is used for training whereas the rest of 20% is used to test the developed model.

5. **Model Selection and Training:** We analyzed several classification algorithms to find out the most accurate machine learning algorithm for the genre classification on the given dataset, such as decision trees, random forests, support vector machines (SVM), bagging classification techniques, and KNN. Then we trained the selected model using the training data.

6. **Model Evaluation:** We evaluated the trained model on the testing set to assess its performance. Common evaluation metrics for genre classification include accuracy, precision, recall, and F1 score. We analyzed the results and iterated on the model as necessary to improve performance. This step provides an unbiased assessment of how well the model generalizes to unseen data.

7. **Deployment and Application**: Once the model was deemed satisfactory, we deployed it to a production environment (flask deployment) to classify the genre of new, unseen music tracks. We then created a web application, integrating the model into a user interface for better usability.

Throughout the development process, we realized it's important to iterate, experiment, and fine-tune various components to achieve the best performance for your specific music genre classification task.



3. THEORETICAL ANALYSIS

3.1 Block diagram:

In this block diagram, the Spotify Music Data is collected using the Spotify data provided to us. The collected data then undergoes Data Preprocessing to clean, transform, and normalize it. Feature Extraction is performed to extract relevant audio features from the data.

The next step involves Model Development, where various Machine Learning Algorithms are employed for music genre classification. We tune the dataset to optimize the performance of the models.

Model Evaluation & Selection is carried out to assess the performance of the trained models using evaluation metrics such as accuracy, precision, recall, or F1 score. The best-performing model is selected for further use.

Finally, the selected model undergoes Model Deployment, where it is integrated into a production environment or application for music genre classification.

Note that this block diagram provides a high-level overview, and the actual implementation may involve more specific steps and components depending on the chosen algorithms and techniques.

3.2 Hardware / Software Designing

To develop and deploy a machine learning-based system for music genre classification using Spotify data, you will need specific hardware and software requirements. Here are the key requirements:

(i) Hardware Requirements:

1. Computer: A reasonably powerful computer with sufficient processing power and memory is recommended, especially for training complex machine-learning models and handling large datasets.
2. Storage: Sufficient storage space to store the Spotify data, extracted features, and trained models. The storage requirements will depend on the size of the dataset and the complexity of the models.
3. GPU (Optional): For training deep learning models, having a compatible GPU can significantly speed up the training process. GPUs with CUDA support are commonly used for accelerating deep-learning computations.

(ii) Software Requirements:

1. Programming Language: A programming language is needed that supports machine learning and data analysis libraries. Python is widely used in the field of machine learning, and it has a rich ecosystem of libraries such as NumPy, Pandas, and Scikit-learn.
2. Machine Learning Libraries: Libraries for machine learning model development and training are also needed. Popular libraries include Scikit-learn. Joblib etc are a must for the chosen operation. These libraries provide various algorithms and utilities for developing and training machine learning models.
3. Data Visualization Libraries: Matplotlib, Seaborn, or Plotly can be used to visualize the data, feature distributions, or evaluation metrics during the development and analysis phases.
4. Development Environment: An integrated development environment (IDE) like Jupyter Notebook, PyCharm, or Visual Studio Code can provide a convenient coding environment for building and testing your machine learning system.

Additionally, you may need to install and manage package dependencies using package managers like pip (for Python) or conda (for Python and R).

It's important to note that the specific software requirements may vary depending on the programming language, libraries, and frameworks you choose to work with. You should also ensure compatibility and version compatibility between different software components to avoid any conflicts or compatibility issues during development.

4. EXPERIMENTAL INVESTIGATIONS

We had all the steps laid out in front of us. Only we had to approach the solution stepwise.

1. Data preprocessing :

a) While preprocessing the data, we found out first that song id, uri, track_href, analysis_url, time_signature, song_name, unnamed:0, and title columns were insignificant in determining the genre of the song.

b) Then we had to make sure the data does not lead to overfitting. Overfitting happens when the model is trained with features that are more than needed. It leads to the model performing well during training but failing during prediction. To prevent this from happening, we plotted boxplots and distribution plots for each column with respect to each genre to differentiate between columns that are important and those that might not be much useful in predicting the genre. We inferred that key and mode might not be needed.

c) Now that the dataset was stripped of the insignificant columns, we had to check for outliers. For that, boxplots for each column were plotted. It turned out they all had outliers. So we took the quantile approach to handle them. We calculated the upper_limit and lower_limit for each column. Then we replaced the values that lay beyond these limits with the boundary value itself. Then again, boxplots were plotted to make sure columns were rid of the outliers.

d) After handling outliers, we plotted a heatmap of the correlation coefficients between the columns to make sure that the columns are relatively independent.

e) When everything in the data was checked, it was decided that ['key',' mode', 'time_signature', 'danceability'] were less likely to contribute to the prediction of music genre.

f) The next step was to check if the columns had any imbalance between them. We plotted a countplot to check that the data were in equal numbers in every column. Indeed, it was not the case. So we used the *imblearn* library in Python to handle the imbalance in the dataset. It contains a method called SMOTE(). SMOTE is basically Synthetic Minority Oversampling Technique. SMOTE is a technique in which the minority class is duplicated to make the count equal to the majority class.

2. Model Building: now that our data is preprocessed, we needed to build the model. There are several algorithms that can be used. The *sklearn* library contains various methods that implement these algorithms. For our purpose, we used RandomForestClassifier(), BaggingClassifier(), KNeighborsClassifier(), DecisionTreeClassifier(), SVC(). Upon building these models and testing, we found out that the RandomForestClassifier() performs best among others with an accuracy of 0.7726524822695036 which is approx. 77.27%.

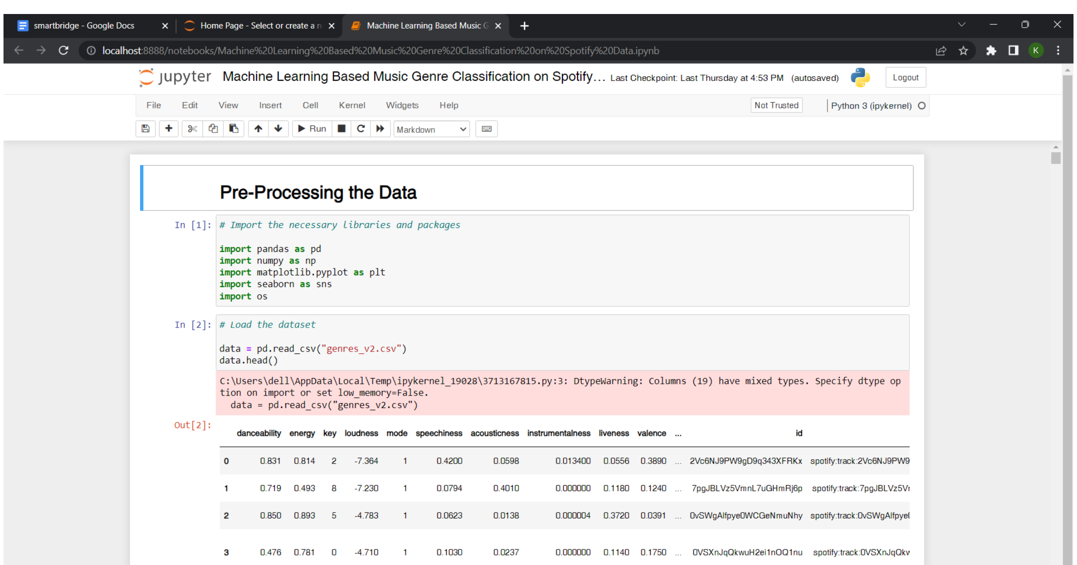
Algorithm	Accuracy(%)
Random Forest Classifier	77.27%
Bagging Classifier	76.09%
KNN Classifier	76.54%
Decision Tree Classifier	70.56%
Support Vector Classifier	66.4%

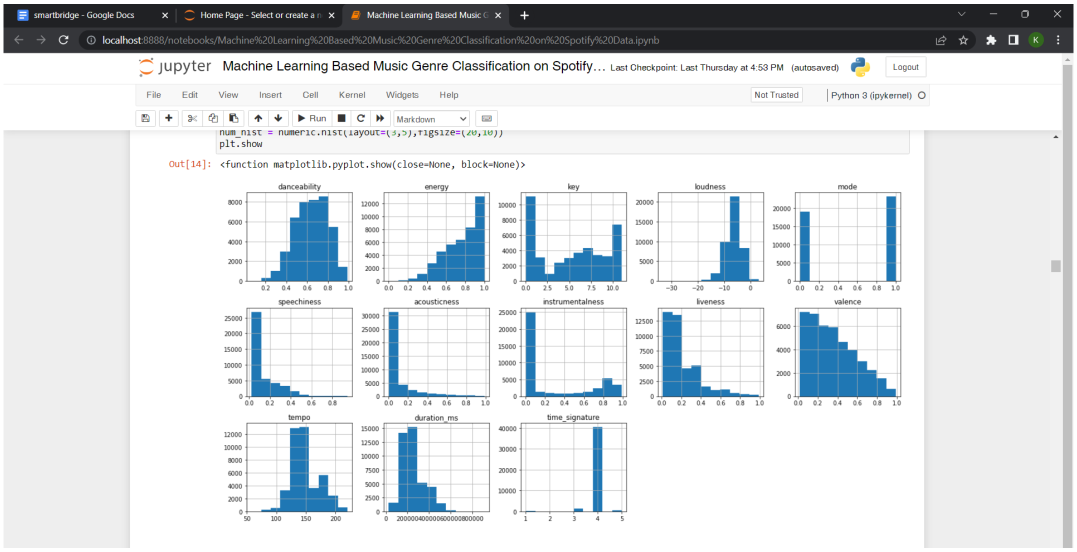
5. FLOWCHART

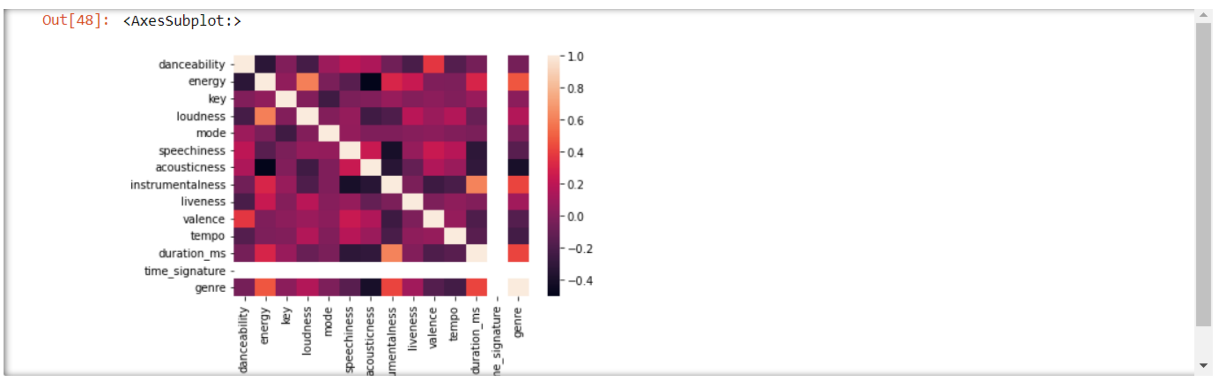
In this flowchart, the system starts by collecting the Spotify music dataset. The data is then preprocessed to clean and transform it as needed. Next, audio features are extracted from the data. The dataset is split into training and testing sets, where the training set is used to train the machine learning model. The model's performance is evaluated using appropriate evaluation metrics. If the performance is not satisfactory, the process loops back to retrain the model using a different algorithm or adjusting the parameters. Once the model's performance is acceptable, it is tested on the testing set, and its performance is evaluated. If the performance is not acceptable, further optimization or retraining may be required. Finally, when the model's performance is deemed acceptable, it is finalized and deployed for music genre classification.

6. RESULT

Upon testing all the models, we decided to go with the Random Forest classifier since it had the most accuracy rate among other algorithms.



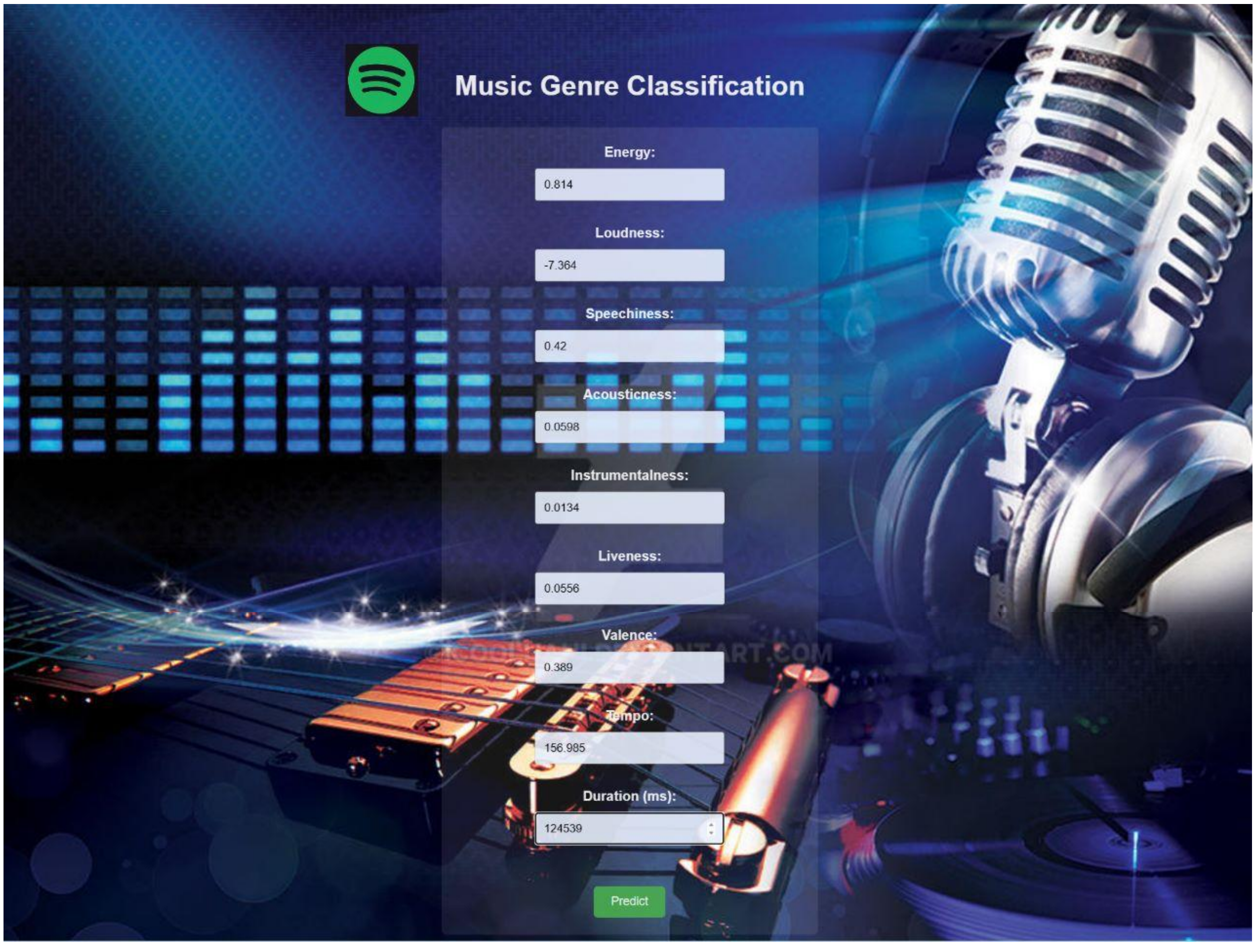


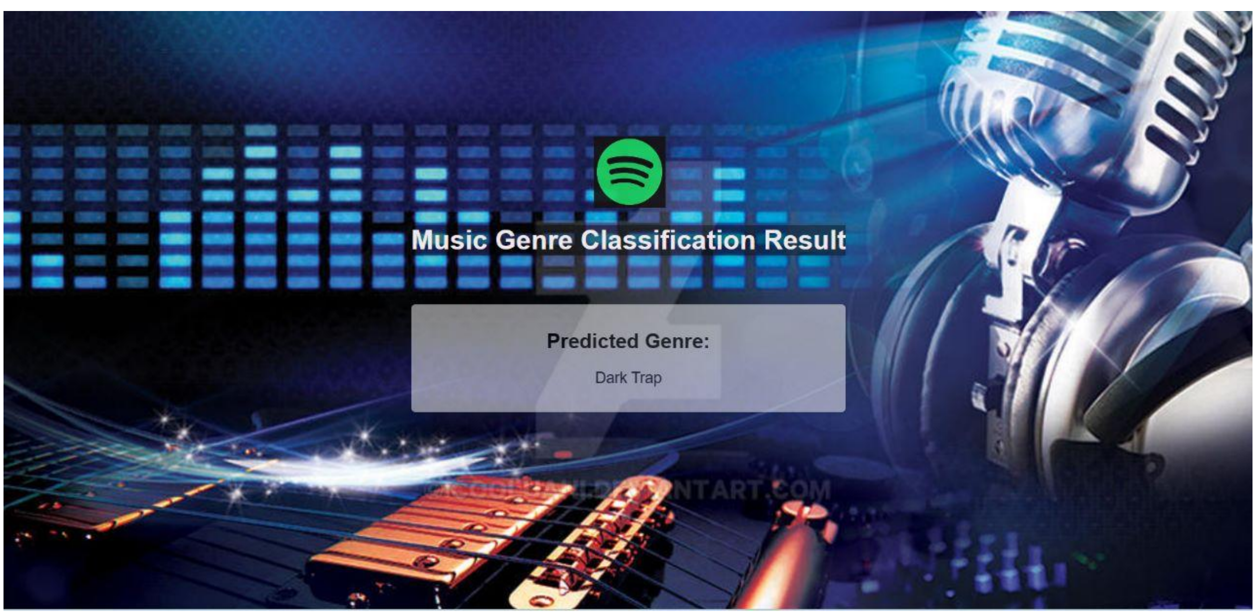


Finally selecting the most important features based on the correlation.

```
Evaluation results for Random Forest:
Accuracy: 0.7698723404255319
Precision: 0.7630532086402605
Recall: 0.7698723404255319
F1-score: 0.7660381783988871
Confusion Matrix:
[[ 553   17   40   11   38   30  144  171   13   17    9   15    9   60
    22]
 [  12  997   21   44    8   32    5   14    5   11    0    5    0    6
     7]
 [  64   41  700   24   74  119   29  148    5    0    0    1    0    2
     6]
 [  10   42   20 1018    7   20    1   10    3    4    0    4    0    5
     1]
 [  31   10   71    8  849   41   28  124    1    4    0    0    0    1
     1]
 [  32   46  124   59   53  777   11   57    1    3    0    3    0    2
     1]
 [ 122    9   18    7   30    7  803  154    0   12    0    0    0    6
    11]
 [ 243   23  193   20  178   72  184  283    2    2    0    1    1    2
     2]
 [   2    9    9    2    0    0    1    1 1163    0    0    0    0    0
     0]
 [   9    4    3    3    6    1    5    2    0 1104    3    0    0    0
    40]
 [  12    0    0    0    0    0    0    0    0    2 1119    0   14   28
     3]
 [   7    8    0    6    2    1    2    0    0    0    0 1099   47    7
     0]
 [  12    0    0    0    0    0    0    1    0    0   22   57 1076   31
     0]
 [  16    6    0    3    2    2    2    2    0    0   42   13   53  978
     4]
 [  29   17    5    2    2    1    7    5    1   56    5    0    0    2
  1050]]
```

Then a web application was created in Python using Flask. The model built was integrated with the web application and the genre of the song was predicted according to the song features provided by the application user.


Music Genre Classification
Energy:
0.814
Loudness:
-7.364
Speechiness:
0.42
Acousticness:
0.0598
Instrumentalness:
0.0134
Liveness:
0.0556
Valence:
0.389
Tempo:
156.985
Duration (ms):
124539
Predict


Music Genre Classification Result
Predicted Genre:
Dark Trap

7. ADVANTAGES & DISADVANTAGES -

Advantages of the proposed solution:

1. **Accuracy and Efficiency:** Machine learning-based music genre classification can achieve high accuracy in categorizing songs into specific genres. It can analyze a large volume of songs quickly and provide reliable genre predictions, enhancing the efficiency of music recommendation systems and other applications.

2. **Personalization:** By accurately identifying music genres, the system enables personalized music recommendations tailored to individual user preferences. This enhances the user experience by offering a curated selection of songs that align with their tastes and increases engagement with the music platform.

3. **Scalability:** The machine learning model can handle a vast amount of music data and scale efficiently to accommodate the growing volume of songs. This scalability is crucial for music streaming platforms that have millions of songs in their catalogs and need to provide real-time recommendations.

4. **Music Discovery and Diversity:** The genre classification system facilitates music discovery by exposing users to a wider range of genres and artists they may not have explored otherwise. It promotes diversity in music consumption and helps emerging artists gain exposure, contributing to a vibrant and inclusive music ecosystem.

5. **Insights and Analytics:** By analyzing the distribution of genres, the system provides valuable insights into music consumption patterns, regional preferences, and cultural trends. These insights can be used for market research, trend analysis, and targeted marketing strategies within the music industry.

Disadvantages of the proposed solution:

1. **Subjectivity and Ambiguity:** Music genre classification can be subjective, as different listeners may perceive the same song differently. Genre boundaries can be blurry, and some songs may exhibit elements from multiple genres, making classification challenging and potentially leading to misclassification.

2. **Lack of Contextual Understanding:** Genre classification solely based on audio features may overlook the contextual aspects of a song, such as cultural significance, lyrical themes, or historical context. This can result in oversimplification and may not capture the full essence of certain genres or subgenres.

3. **Bias and Representation:** There can be an inherent bias in the genre labeling process, as certain genres or subgenres may be underrepresented in the training data. This can lead to the misrepresentation or exclusion of certain music styles, potentially perpetuating biases and limiting diversity in recommendations.

4. **Evolution of Music Trends:** Music genres and styles constantly evolve, with new subgenres emerging over time. Machine learning models trained on existing data may struggle to adapt to emerging trends, leading to potential inaccuracies or outdated genre classifications.

5. **Data Limitations and Quality:** The accuracy of the genre classification system heavily relies on the quality and diversity of the training data. Incomplete or biased datasets may result in lower accuracy or limited coverage of niche genres, affecting the overall performance of the system.

It is important to address these disadvantages by continuously refining the classification algorithms, ensuring diverse and representative training datasets, incorporating user feedback, and considering context and cultural factors in the genre classification process.

8. APPLICATIONS -

The solution of machine learning-based music genre classification has several applications across various areas within the music industry. Some of the key applications include:

- **Music Streaming Platforms:** Music streaming platforms like Spotify, Apple Music, and SoundCloud can utilize genre classification to enhance their recommendation systems. By understanding users' genre preferences, these platforms can provide personalized playlists, curated recommendations, and improved discovery of new songs and artists.

- **Music Discovery and Recommendation Services:** Music discovery services and recommendation engines, both within music streaming platforms and standalone apps, can leverage genre classification to suggest songs, albums, or artists based on users' preferred genres. This helps users explore new music and expand their musical horizons.
- **Music Licensing and Rights Management:** Music genre classification can assist in music licensing and rights management processes. Music supervisors, licensing agencies, and production companies can use genre classification to efficiently search for songs that fit specific genres required for various media projects, such as films, commercials, or video games.

- **Music Recognition Services:** Music recognition apps like Shazam or SoundHound can benefit from genre classification to provide additional information about recognized songs. By identifying the genre of a recognized song, these services can offer genre-specific recommendations, create personalized playlists, or provide insights into users' music preferences.

- **Market Research and Trend Analysis:** Genre classification enables researchers and analysts to conduct market research and trend analysis in the music industry. By analyzing the distribution of genres, listening preferences, and cultural trends, valuable insights can be gained for marketing strategies, targeted promotions, and understanding consumer behavior.

- **Music Education and Training:** Genre classification can be applied in music education and training to categorize and organize music resources. It can assist students, educators, and musicians in exploring different genres, studying stylistic characteristics, and developing a comprehensive understanding of music genres and their historical context.

- **Music Genre Tagging and Metadata Management:** Genre classification can be used to tag music files with genre metadata. This enhances the organization and categorization of music libraries, making it easier for users to search, filter, and manage their music collections based on specific genres.

- **Music Recommendation for Events and Venues:** Event organizers, clubs, and venues can utilize genre classification to provide tailored music recommendations for specific events or target audiences. This ensures that the music played at events aligns with the desired genres and enhances the overall experience for attendees.

These are just a few examples of the applications of machine learning-based music genre classification. The solution can be further adapted and integrated into various areas within the music industry to enhance user experiences, streamline processes, and facilitate music exploration and discovery.

9. CONCLUSION

In conclusion, a machine learning-based system for music genre classification using Spotify data offers a promising approach to automatically categorizing music tracks into different genres. By leveraging the rich audio features provided by Spotify, along with machine learning algorithms, accurate genre classification can be achieved. Through the system's development process, various steps were undertaken. The Spotify music data was collected, preprocessed, and transformed to ensure its quality and suitability for analysis. Audio features were extracted to capture relevant characteristics of the music tracks. Machine learning algorithms were employed, and tuned to optimize the performance of the genre classification model. Evaluation metrics provided insights into the model's effectiveness and guided further refinement. The system's performance was thoroughly assessed using testing data, and iterations were made to enhance the model's accuracy and robustness. The final model, demonstrating satisfactory performance, was selected for deployment. By deploying the system, it becomes possible to classify music genres automatically and efficiently.

10. FUTURE SCOPE

The future scope of a machine learning-based system for music genre classification using Spotify data is promising and opens up several possibilities for advancement and expansion. Here are some potential future directions:

1. Enhanced Feature Extraction: Currently, the system relies on a set of predefined audio features provided by Spotify. Future research can explore the incorporation of more advanced feature extraction techniques, such as deep learning-based feature learning or combining audio features with lyrics, artist information, or user-generated content for improved genre classification accuracy.

2. Multi-label Classification: Music tracks often belong to multiple genres, and assigning a single genre label may not capture the full complexity of the music. Future systems can be developed to perform multi-label classification, allowing for more nuanced and accurate genre assignments.

3. Fine-Grained Genre Classification: While the current system focuses on broad music genres, future advancements can aim for finer-grained genre classification. This involves categorizing music into more specific sub-genres or even creating personalized genre taxonomies based on individual preferences.

4. Real-Time Genre Classification: Integrating the system into real-time music streaming platforms would enable on-the-fly genre classification, allowing for dynamic music recommendations and personalized experiences. This could involve developing lightweight models that can run efficiently on resource-constrained devices.

5. Active Learning and Incremental Learning: Incorporating active learning techniques can enable the system to iteratively and selectively acquire new training data to improve genre classification performance. Additionally, incremental learning approaches can facilitate updating the model with new data over time, allowing the system to adapt and stay up-to-date with evolving music trends and preferences.

6. User-Driven Genre Classification: Leveraging user feedback and preferences can enhance the system's accuracy by incorporating user-generated genre tags or user-specific genre preferences. Integrating user feedback mechanisms into the system can allow for active user participation and personalization.

7. Interpretability and Explainability: Providing interpretability and explainability for the genre classification decisions can help build trust and understanding among users. Future research can focus on developing methods to explain the reasoning behind the system's predictions, providing insights into how specific audio features contribute to genre classification.

Overall, the future scope of a machine learning-based system for music genre classification using Spotify data involves advancements in feature extraction, classification techniques, real-time applications, and incorporating user feedback. These advancements can lead to more accurate and personalized genre classification systems, further enhancing the music listening and discovery experiences for users.

11. BIBLIOGRAPHY

1. https://youtu.be/qBigTkBLU6g
2. https://youtu.be/xhB-dmKmzRk
3. https://youtu.be/v6VJ2RO66Ag
4. https://youtu.be/LbX4X71-TFI
5. https://youtu.be/5bHpPQ6_OU4
6. https://youtu.be/FndwYNcVe0U
7. https://youtube.com/playlist?list=PL-osiE80TeTs4UjLw5MM6OjgkjFeUxCYH
8. https://youtu.be/Z1RJmh_OqeA
9. https://www.geeksforgeeks.org/flask-tutorial/amp/
10. https://youtu.be/sugvnHA7EIY

APPENDIX

A. Source code for Machine Learning part.
B. Source Code for Flask app.
C. Source Code for index.html
D. Source Code for results.html

Source code for above listed files has been compiled below:

https://drive.google.com/drive/folders/1He4SLQRNIAmD3VLmuwcjD_nGtQUqXv03?usp=sharing