**Genre Genius: Applied ML Group Project**



By:

James Fordham

Patrick Le

Philippe Louis Jr

Frank Prescott Williams

**Introduction**

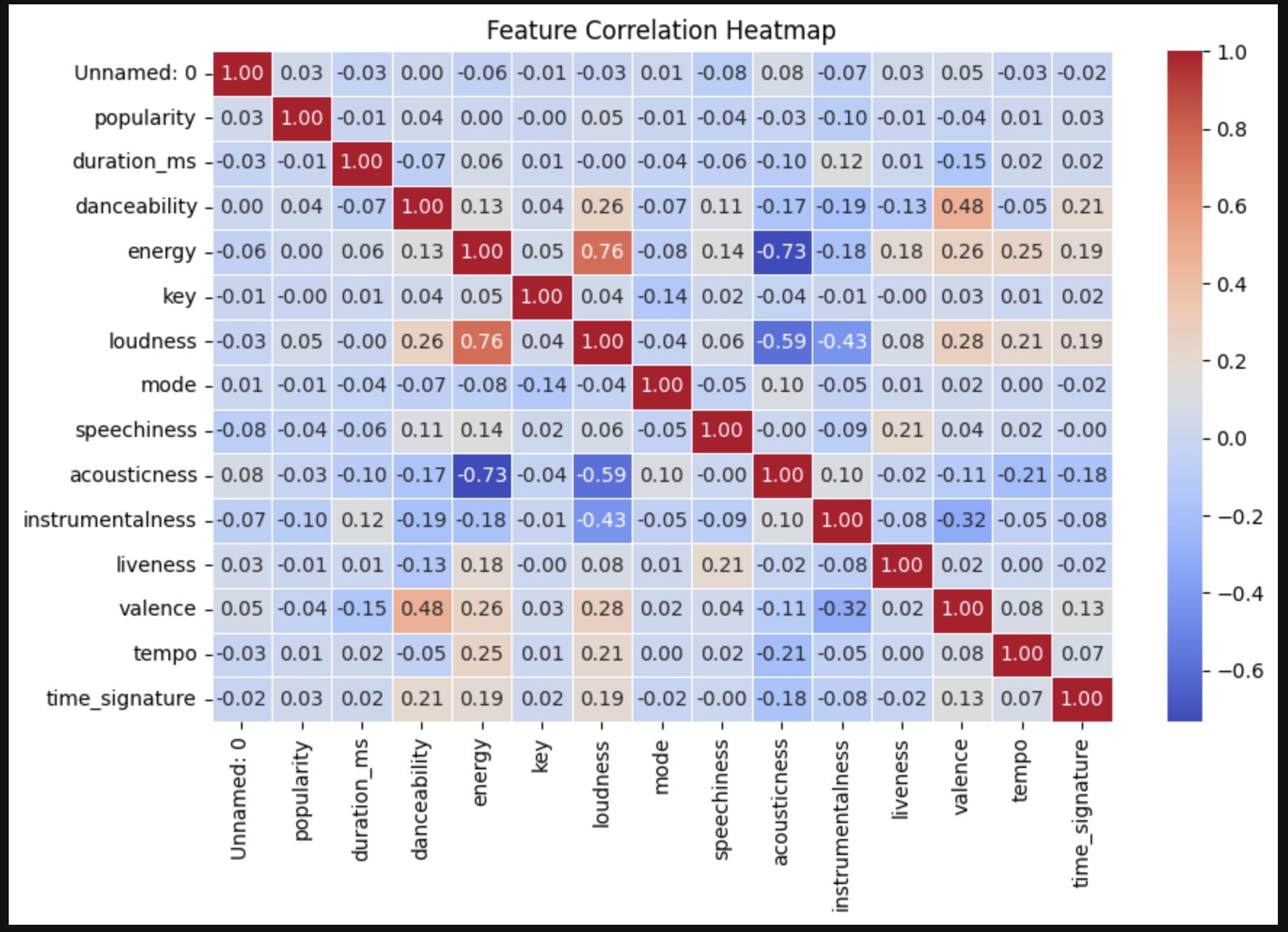
In the era of digital music, streaming platforms like Spotify have revolutionized the way we access, enjoy, and explore music. These platforms house vast, eclectic libraries of tracks, encompassing a wide range of genres and artists. With this wealth of data, there emerges an opportunity to delve deeper into understanding the intricate characteristics of music and developing sophisticated models that can predict and classify various attributes of a song, such as its genre.

**Literature Review**

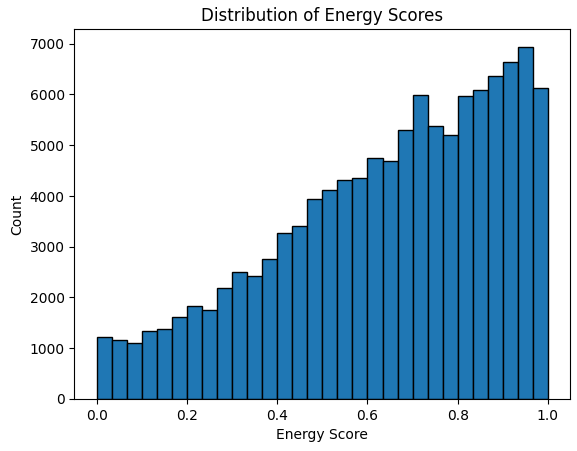
The primary problem this project addresses is the challenge of automatically classifying songs into their respective genres based solely on intrinsic musical features. Genre classification is inherently complex due to the subtle and overlapping characteristics that different genres often share, and traditional approaches typically rely on manual tagging, metadata analysis, or simplistic models that fail to capture these nuances. Our stakeholders include Spotify and Spotify artists who seek better tools for identifying their ideal market. The goal of this project is to leverage data science to predict a song’s genre using a rich Spotify dataset containing detailed musical attributes. While most applications of this dataset focus on predicting a song’s popularity, our approach is unique in using these features to classify genre. Unlike typical genre classification methods that rely on raw audio input, such as Shazam, we aim to achieve similar outcomes through structured data, potentially improving recommendation engines and aiding artists in better understanding their musical identity.

**Data**

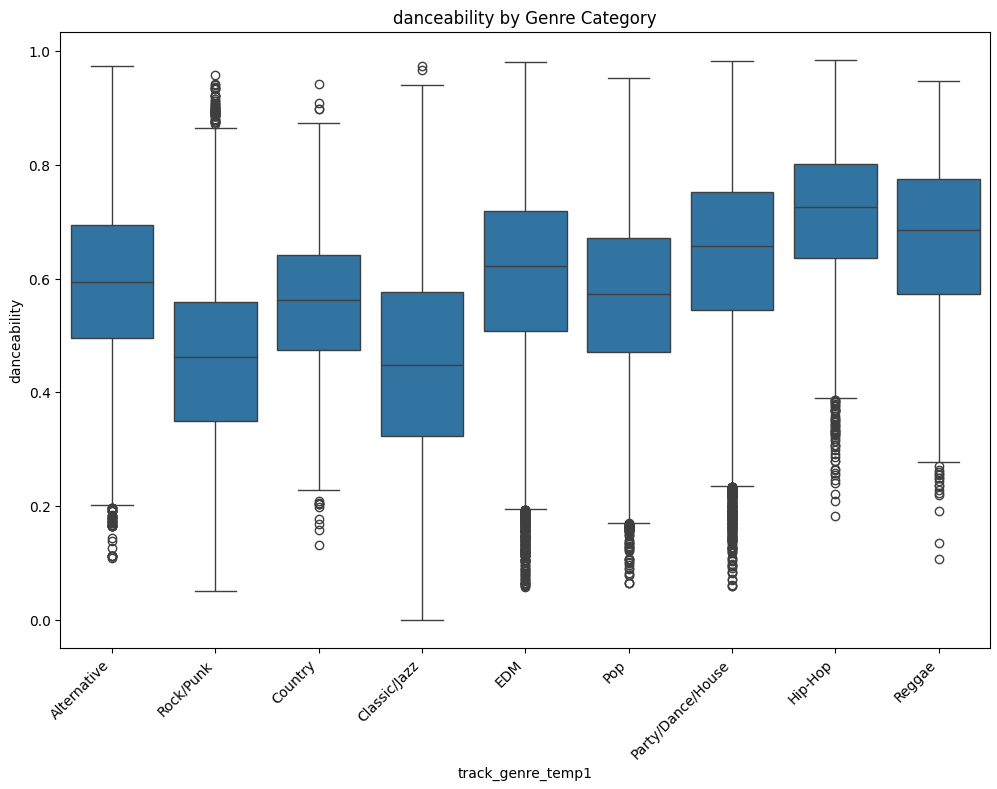
There are 11400 rows and 21 columns. This dataset contains integer, float, object (string), and boolean types. Most of the numerical columns use int64 or float64, while categorical values are stored as objects. Artist, Album\_Name and Track\_Name each have 1 missing value but all other columns have no missing values. Given the structure, this dataset appears to be derived from Spotify's API, it is likely reliable. The dataset does not explicitly include metadata such as source attribution, collection date, or methodology. There were 1000 data points for each genre and there was an imbalance once we grouped genres which we overcame using SMOTE.



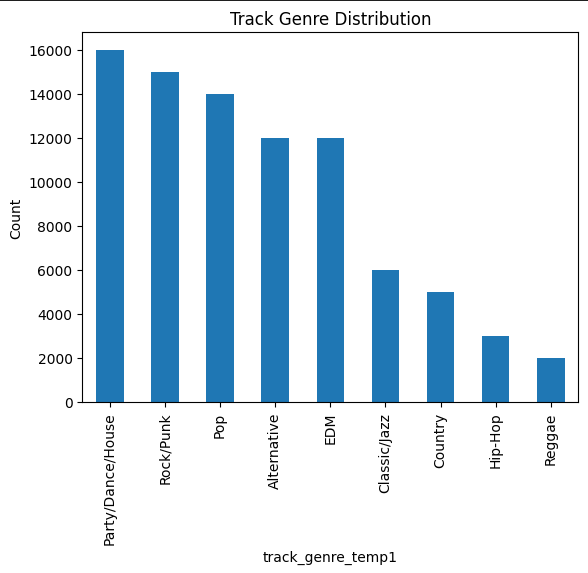
Songs with higher energy are typically louder (0.76), and danceable songs tend to be moderately happier (0.48). In contrast, As a songs' energy rises, the “acousticness” of said song goes down (-0.73) and as “loudness” increases the “acousticness” tends to go down(-0.59). Popularity shows little correlation with musical characteristics, suggesting that factors like marketing, playlist placements, and artist reputation play a more significant role in a song’s success.



We also plotted distributions of multiple features to see if we can gather any insights on features by themselves. One big finding was that there were more songs with higher energy scores than lower. This helped us conclude that this dataset may be skewed with a lot of high energy songs that relate more towards EDM.



Looking into the genres, we noticed that there were multiple features with a lot of outliers for each of the genres. For example, looking at danceability to genre, we saw a lot of outliers in every genre of music.

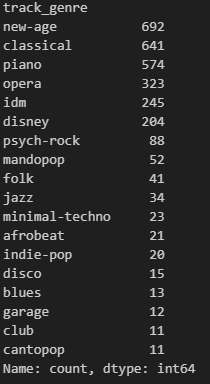


After consolidating all genres into broader categories mentioned in the methods section, we found that there was a distinct difference in the amount of songs between different genres. This helped us in realizing that there were big class imbalances and also in deciding to combine certain genres.

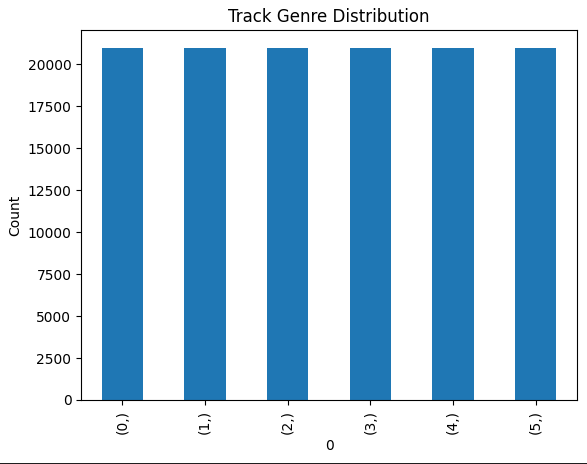
**Methods**

There was an issue of having a multitude of distinct music genres and this had negatively impacted our model’s accuracy. We consolidated similar subgenres into broader, more general categories to reduce the number of output classes. Additionally, we removed mislabeled entries such as “Iranian” and “Spanish,” which refer to languages rather than musical genres. After redefining genre labels, we conducted exploratory data analysis to understand attribute distributions within each group.

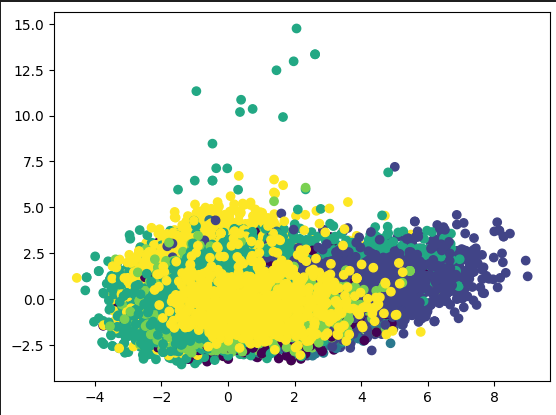
Outlier handling was performed using z-score analysis across relevant audio features such as “loudness” and “speechiness”. We first made boxplots shown above in the EDA, and then printed out the count of songs per specific genre that were outliers.. Data points with z-scores exceeding ±3 were considered outliers and removed. Certain genres, such as “New Age” and “Piano,” consistently displayed extreme values across multiple features and were therefore excluded from the dataset to improve overall model performance.

“Instrulmentalness” outliers “loudness” outliers

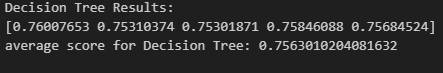
For feature scaling, we applied StandardScaler rather than MinMaxScaler, as the standardization approach better preserved the distributional characteristics of our audio features. Following preprocessing, we observed significant class imbalance among the remaining genre categories (e.g., EDM/House vs. Hip-Hop). To address this, we employed SMOTEto generate synthetic samples for underrepresented genres and balance the dataset. The picture below is to show that the classes are balanced.

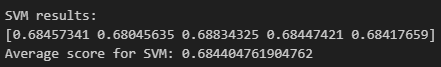


An unsuccessful method we attempted involved using PCA to visualize potential linear separability among genres. However, the resulting plots showed significant overlap between classes, indicating that PCA was not effective for our use case. This was meant to see if we were able to separate the genres well enough. However, this shows that based on all of our features and the way we grouped the genres, it is not enough to separate each of the genres.

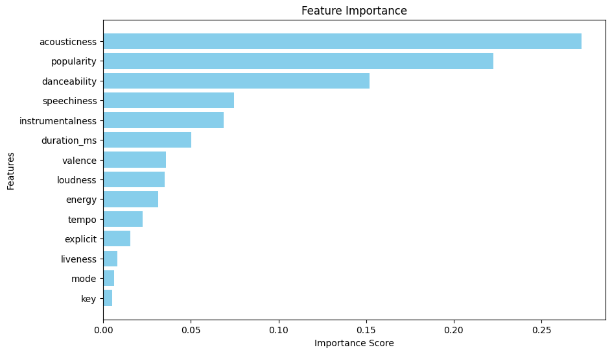


**Results**

For our classical machine learning models, we used Decision Tree, SVM, and XGboost. We found the natural hyperparameters to work the best after using grid search, so we kept the base models. We used 5 fold cross validation for each model, and here were the results:



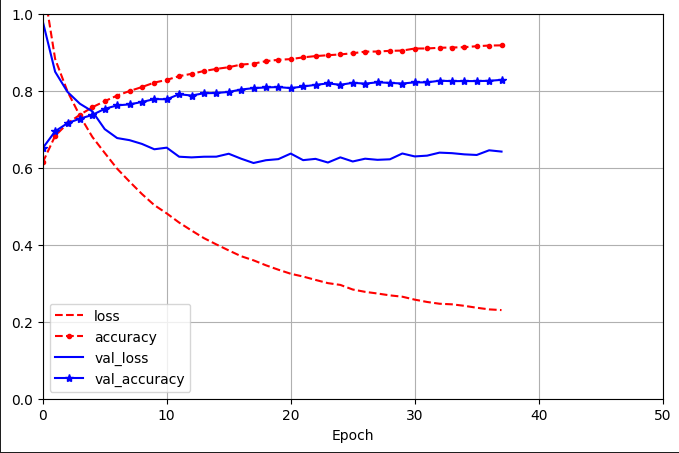
Overall, the XGboost model worked the best with an average of 80.1% accuracy, SVM performed the worst with 68.4% accuracy, and Decision Tree somewhere in the middle with 75.6% accuracy. We also created a feature importance visual after training our decision tree.



We found that the top 3 features were “acousticness”, popularity, and danceability.

Overall, our classical models did pretty well compared to what we were expecting considering that our PCA visualization made us assume otherwise.

We also implemented a feed-forward neural network. This neural network comprises 1 input layer, 3 hidden layers, and 1 output layer. The hidden layers used a RELU activation function. We also included batch normalization to avoid vanishing gradients. Lastly, we had early stopping and used the adam optimizer.



Our neural network was a step above our classical ml models, having an accuracy of 94.72% and a validation accuracy of 86.52%.

**Discussion**

Our model achieved an 87% accuracy by consolidating 114 musical genres into six broader categories, guided by research and a music specialist's input. This grouping was necessary as initial attempts to classify all genres yielded low accuracy. While this approach limits specificity, it offers practical applications, such as aiding platforms like Spotify in general genre identification. However, dataset inconsistencies, including misclassifications like linguistic labels (e.g., "Spanish" or "Iranian"), likely impacted reliability. Time constraints prevented manual reclassification, but with additional resources, a more rigorous labeling process could improve data quality and model performance.

**Limitations**

One limitation of the dataset was the inconsistency in genre classification. Several entries labeled as genres were, in fact, languages—such as Spanish or Iranian—which do not accurately represent musical genres. This misclassification likely impacted the reliability of our genre-based analysis. There are also major differences even in between genres which could cause issues, such as EDM having 22 different subgenres, all with different energy, “instrulmentalness”, etc. Lastly, our genres could also have hidden sub genres that are not accurately reflected such as a song being primarily rock, but have a subgenre of jazz. This was proven by our PCA visualization where genres blended together.

Additionally, our dataset is missing crucial data that would help with identifying which genres that a song should fall into. For example, a key identifier would be having a list of all the instruments used in the song. A song that contains instruments like an electric guitar and drums would probably lead to a prediction of a rock song. Lastly, our project was constrained by time, limiting our ability to manually verify and reclassify each song. While ideally we could have individually reviewed each track to ensure accurate genre categorization, this approach would have been highly impractical. However, with additional resources, such as funding to outsource this task, a more precise genre labeling process could be implemented to enhance data quality.

**Future work**

Although the current model demonstrates solid performance in classifying genres using Spotify’s audio features, there are numerous opportunities to enhance this project in future iterations.

**1. Integrate Lyrics-Based Features**

Incorporating Natural Language Processing (NLP) techniques to analyze song lyrics could provide deeper insight into mood, themes, or sentiment, all of which may influence genre and popularity. Sentiment scores, word frequency, or topic modeling could add additional predictive power to our existing feature set.

**2. Use Listener and Behavioral Data**

Spotify offers metadata such as user skip rates, playlist adds, geographic location, and demographic details. These features can help model popularity with greater context and accuracy. For example, a song’s popularity might vary dramatically across regions or age groups, which is not currently reflected in the model.

**3. Implement Deep Learning on Raw Audio**

Instead of relying solely on engineered Spotify features, future work could explore the use of **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)** to extract patterns directly from raw audio files or spectrograms. Previous studies have shown this to be effective in genre classification and emotion detection.

**5. Unsupervised Learning for Subgenre Discovery**

Clustering methods like K-Means, DBSCAN, or Hierarchical Clustering can be used to discover emergent subgenres or acoustic groupings that aren’t represented in current taxonomies. Dimensionality reduction techniques (PCA, t-SNE) can also be used to visualize song groupings and genre overlaps.

**Citations**

**Spotify Developers.** *Spotify Web API Documentation*. Spotify,<https://developer.spotify.com/documentation/web-api/>.

**Spotify Track Dataset.** *Spotify Dataset 1921–2020, 160k+ Tracks*. Kaggle,<https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset>.

**Choi, Keunwoo, György Fazekas, and Mark Sandler.** "Automatic Tagging Using Deep Convolutional Neural Networks." *arXiv*, 2016,<https://arxiv.org/abs/1606.00298>.

**Tzanetakis, George, and Perry Cook.** "Musical Genre Classification of Audio Signals." *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, 2002, pp. 293–302. IEEE,<https://doi.org/10.1109/TSA.2002.800560>.

**McFee, Brian, et al.** "librosa: Audio and Music Signal Analysis in Python." *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–25,<https://doi.org/10.25080/Majora-7b98e3ed-003>.

**Naranpanawa, Surani.** *Spotify Track Genre Prediction*. Kaggle,<https://www.kaggle.com/code/suraninaranpanawa/spotify-track-genre-prediction#4)-Data-Preprocessing>.