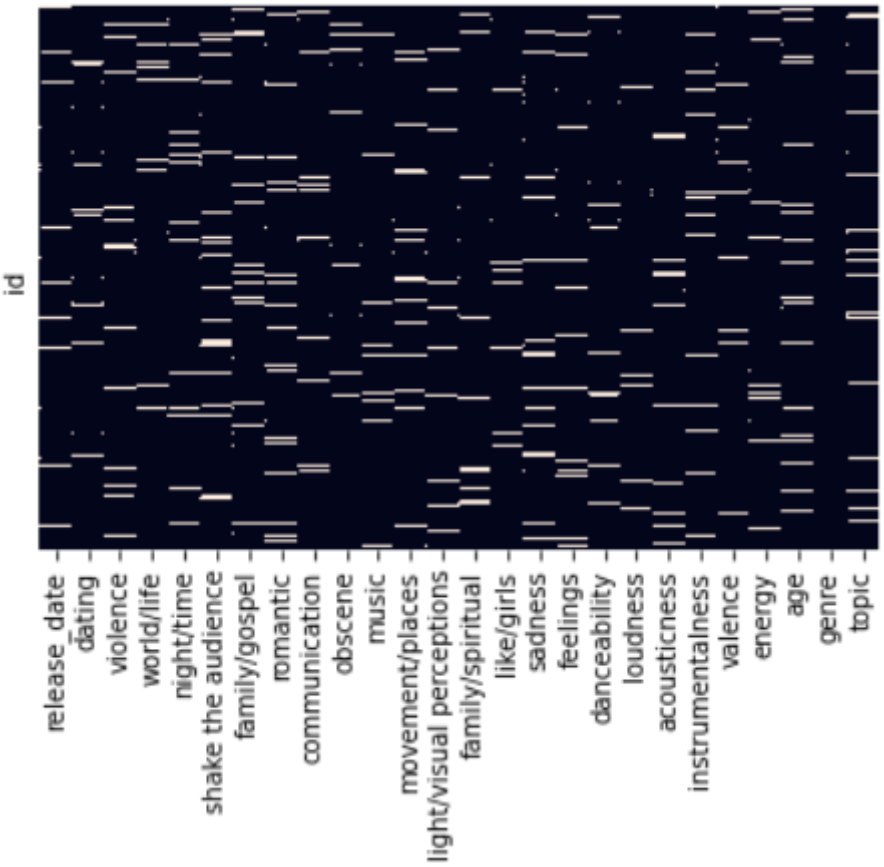
BEGINNER'S HYPOTHESIS SOLUTION DOCUMENTATION

Torture the data, and it will confess to anything. This was an amazing quote that was present on Cerebro's Homepage which inspired me. And I think I have done my fair share of work over the past few days to honour this quote.

Data Cleaning

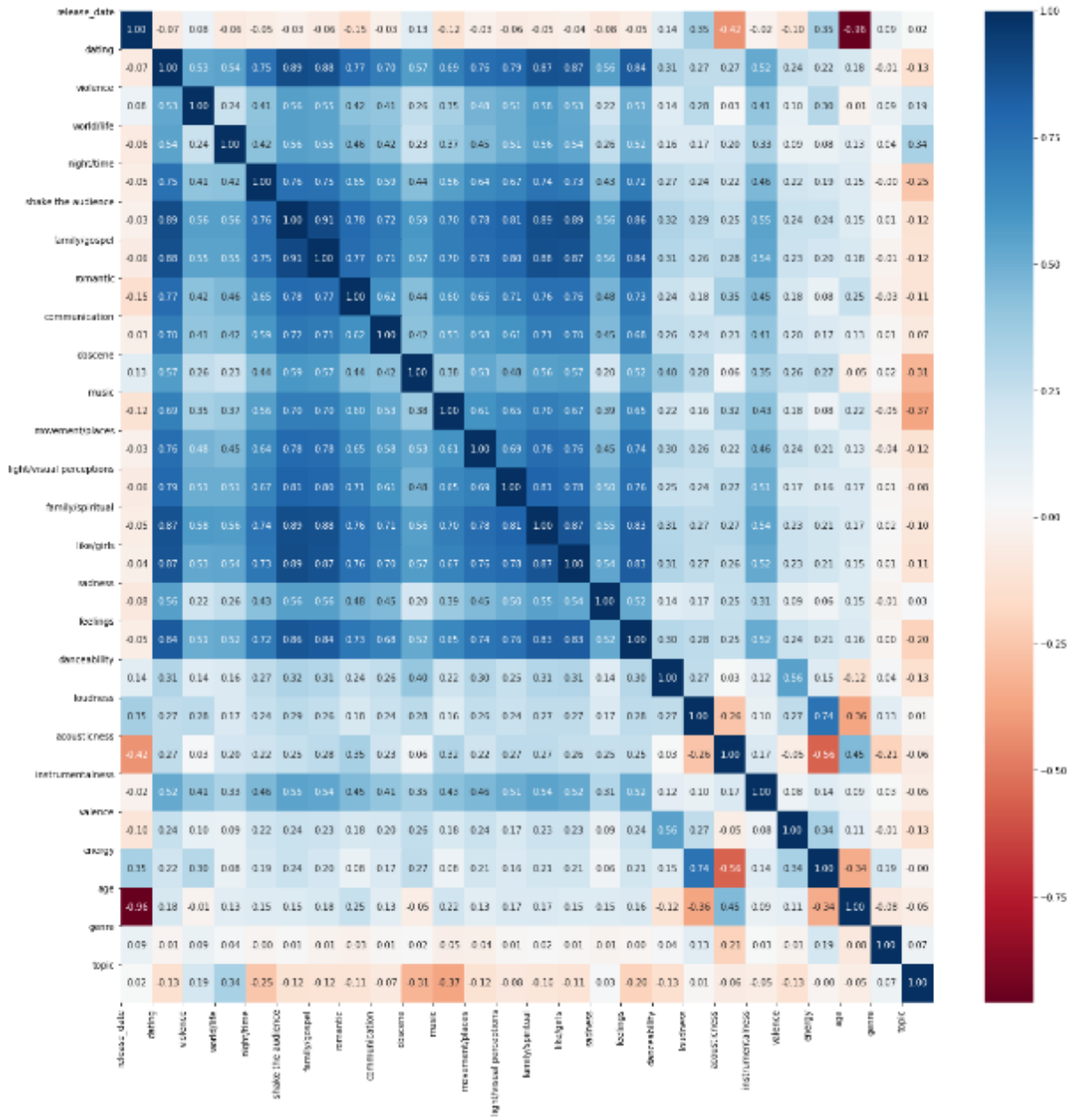
After obtaining the dataset, we must first find out the presence of the null values. The presence of missing data will affect the working of the model. We can find the number of rows using the basic 'isnull().sum()'. We can then use a heatmap to visualise which rows are affected.



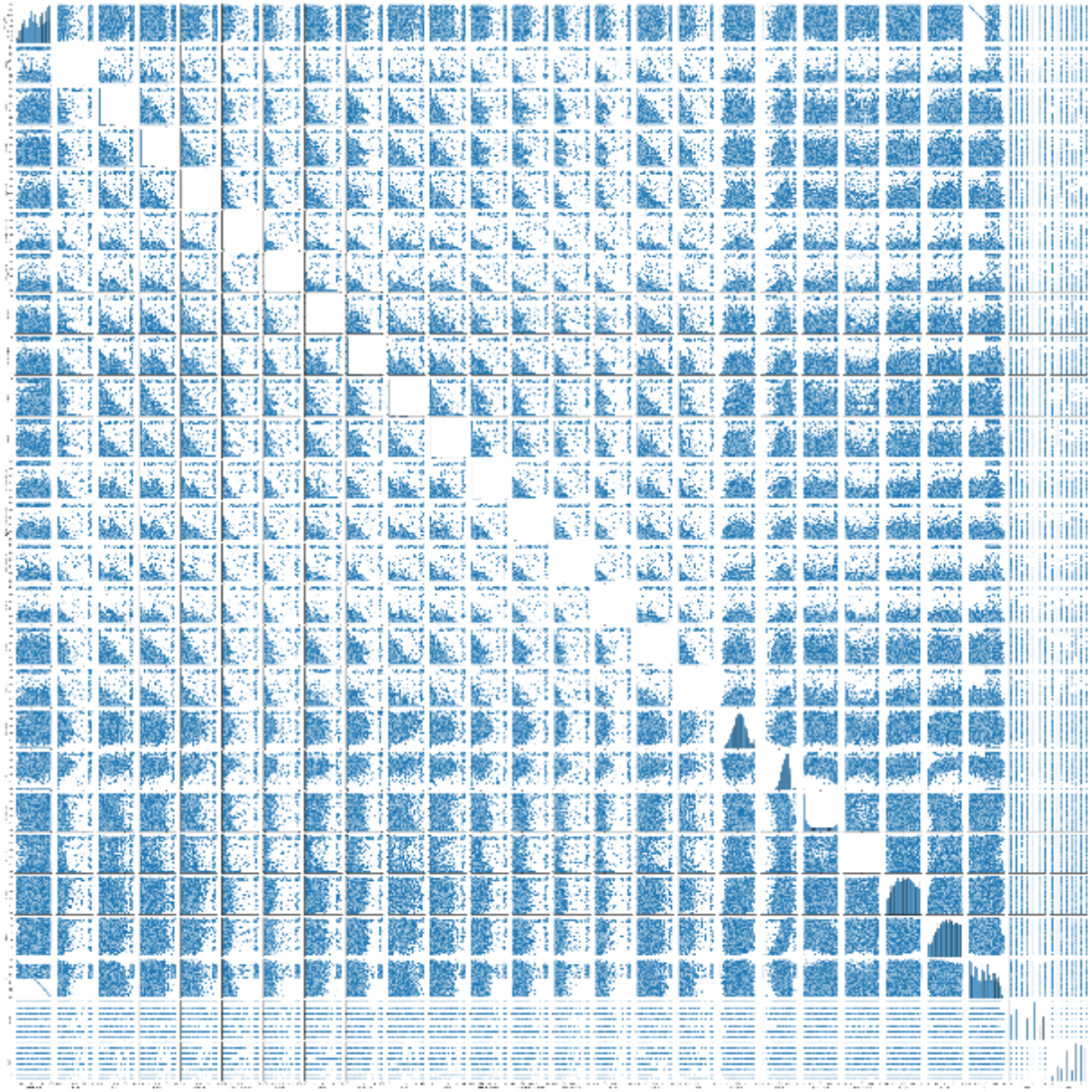
Since we can see that many rows have missing values, we cannot drop all the rows with missing data because it will lead to the loss of a large amount of data. So, we need to try and fill these null values with certain appropriate values based on statistical methods such as mean, median (for numerical variables), mode (for categorical variables) etc. Usually, median is preferred over mean when the data has too many outliers. But I have chosen to fill the null values with the mean of the respective column because it produced better results in the final model.

Data Visualisation

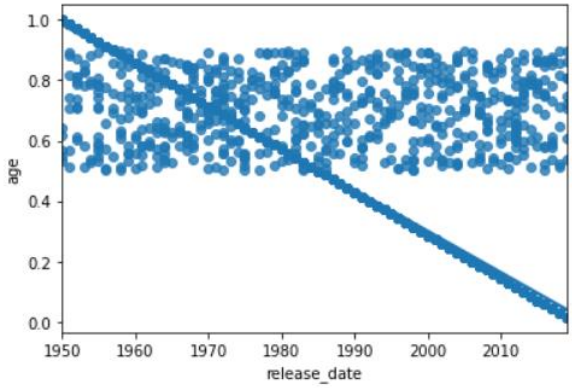
The next important task in Data Analysis is Exploratory Data Analysis. I have first made the heatmap of the correlation data from the dataset. The correlation data helps us to understand how different columns are related to each other.



The correlation value of the target column 'genre' and the remaining columns is very close to 0, which suggest a weak relationship between the target column and the remaining columns. Then I have pair plotted the data to get a better understanding.

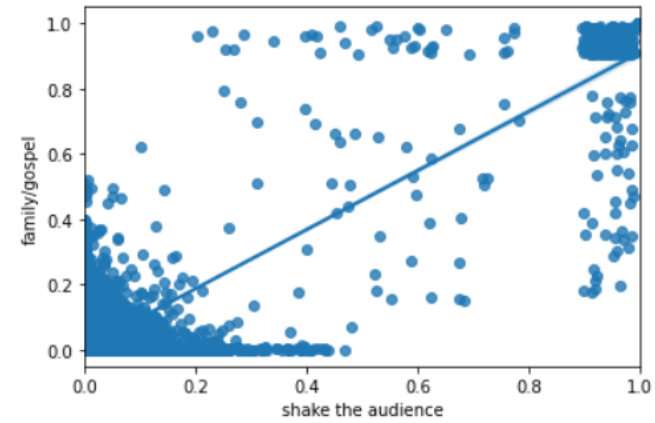


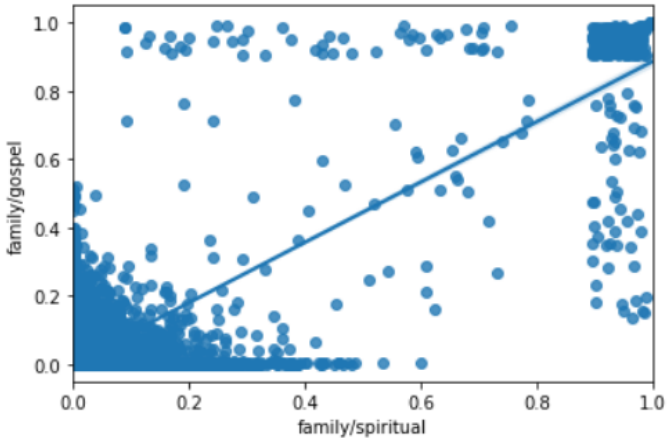
I have observed a very strong negative relationship between the 'age' and 'release_date' columns. So I made a regression plot for these two columns.

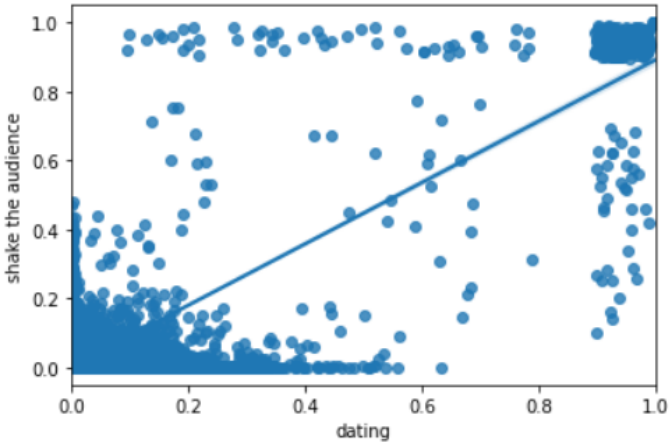


We can observe that the regression line is almost along the data points. I have also observed a strong positive relationship (corr ≈ 0.9) between some other columns. I made some regression plots for these columns as well such as

'shake the audience'-'family/gospel', 'family/gospel'-'family/spiritual' and 'dating'-'shake the audience'. The regression line with a positive slope in these graphs explains the strong positive correlation value







The code for the graphs is available in the 'Visualisation.ipynb' file of the GitHub repository given below. GitHub is unable to render this file properly. To view this file, please download the repository.

StandardScaler()

Before jumping into the model, I have made use of the StandardScaler() on the data. The reason why I have used it in the first place was that when I was exploring different models, Logistic Regression was throwing a warning and asked me to make use of the StandardScaler function. After making use of the StandardScaler, I observed a mild increase in performance of the RandomForestClassifier (which is the final model) as well. Upon investigating the dataset, I observed that some of the columns were marked on a scale between 0 and 1 while others (excluding the 'release_date' column) using a different scale. I believed that this discrepancy may have caused a mild dip in the performance of the model. So to avoid this, I have made use of the StandardScaler function to normalize the data before fitting it into the model. This helps to improve the performance of the data.

Model Selection

Model selection is the next important step in the process. I first took the dataset (after cleaning and normalising), I experimented by applying different classification algorithms to it and analysing the accuracy score (using the accuracy_score() function in sklearn) of each raw model (model without changing the hyperparameters). I have used Logistic Regression, Naive Byes, K Nearest Neighbors, Support Vector Machine and Random Forest. The Naive Bayes model gave a very low accuracy score of 0.30. The Logistic Regression gave a better result of 0.37. The K Nearest Neighbors Classifier also gave a score similar to Logistic Regression. I have tried to increase the number of neighbours and get a better result but it was not successful. The Support Vector Classifier showed varying success based on the kernel used. Out of the different kernels available, the 'rbf' kernel gave the highest accuracy score of 0.40. Finally, the Random Forest Classifier gave good accuracy with the default arguments. So, I increased the n_estimators by putting it in a for loop and checking accuracy for different values. Due to limited RAM on my laptop, I have run some of the notebooks in Colaboratory and Kaggle. Random Forest Classifier showed a huge increase in accuracy as I increased the number of decision trees. Upon doing some research showed me why Random Forest worked better. One of the biggest reasons why Random Forest gives a better result was because it is a very versatile algorithm. Random Forest consists of many decision trees. It reduces the overfitting of data by training different datasets. But the biggest problem with Random Forest algorithms is that it is computationally intensive and requires a lot of RAM. I got a memory error

when I tried to increase the estimators. This was the reason why I had to move to Colaboratory and Kaggle. Due to these reasons, I have used the Random Forest Classifier in my final model.
I then tried to remove some columns which already have a high correlational value with other columns. But it started decreasing the accuracy of the model. So, I have decided against dropping any columns.

Hyperparameters

Then finally, to try and improve the model I tried to tweak the hyperparameters of the model using the help of GridSearchCV. I tried to vary different parameters such as n_estimators, bootstrap, min_samples_split and max_features and max_depth. bootstrap controls whether the entire dataset is used to fit the model or only a part of it is used. max_features controls the number of features to be considered when splitting the branches. min_sample_split controls the minimum number of samples required to split an internal node. max_depth controls the maximum depth of the tree. By making small changes I have arrived at the final model.

The final code is available on 'finalnotebook.ipynb' file and the final csv is available on 'finaldata.csv' file on the GitHub repository given below.

By

Rohith Varma Buddaraju

20116080

ECE First Year

rohith_vb@ece.iitr.ac.in

GitHub Repository - https://github.com/rohithvarma3000/beghyposolution