STAT 204 Final Report - Creating a Popular TikTok Video

Vidyasri Ravi and Vanessa Oviedo

1 Introduction

TikTok is a short-form video hosting platform with a mission to inspire creativity and bring joy. It contains videos submitted by a variety of users from around the globe with the duration of each video ranging from 15 seconds to ten minutes. The perks of creating a series of popular tiktoks leads to the creator being socially recognized and get paid by TikTok Creator Fund. Although it may seem fairly easy to become popular on TikTok by uploading videos, it is far from true. There are a myriad factors that one should get right when making a TikTok video for it to have high viewership numbers, which inturn determines the popularity of video. The dataframe we chose for this project is the TikTok dataset from Kaggle, which has 3352 observations spread across 21 variables.

Our goals in this project are,

- To employ EDA, AIC and BIC to choose the best predictors and models for this dataset
- To employ regression models, ANOVA, PCA, clustering and logistic regression to find out which factors influence the popularity of a TikTok video the most.
- To predict the likelihood that a Tiktok will be popular based on the variables of interest.
- To compare logistic regression models with varying numbers of predictors to conclude the minimum number of predictors which are key to creating a popular TikTok video.

2 Data Preprocessing and EDA

We started off by checking for missing values and there were no missing values in the dataset. Then we built histograms and scatter plots to check for outliers and to determine which independent variables were more significant than the others. We created boxplots to identify how popular or not popular each TikTok with respect to each of the numeric variables. We constructed facet plots to get a bigger picture of how the combination of two character variables, decade and playlist, influenced popularity. Although some variables had outliers, we observed with the help of boxplots that these variables were of no significance in determining popularity, so we did not have to deal with outliers as we dropped those variables in the data preprocessing phase.

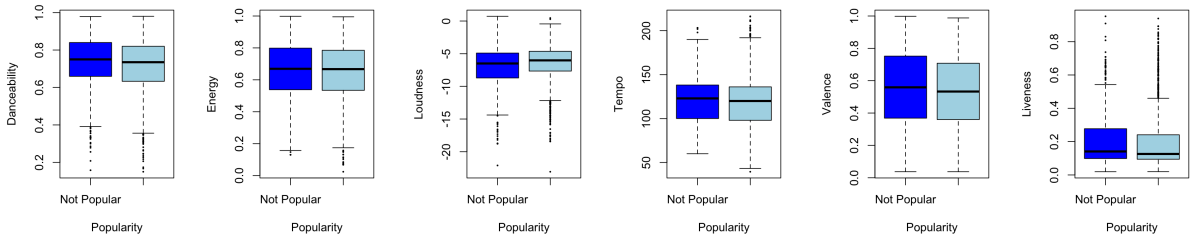


Figure 1: Boxplot of independent variables (a) Danceability, (b) Energy, (c) Loudness, (d) Tempo, (e) Valence, and (f) Liveness

The result of this step is that we were able to identify the list of independent variables which we found to have a direct impact on popularity, the dependent variable, which are listed below.

- Character variables - *decade, playlist name*
- Integer variables - *duration, popularity*

- Numeric variables - *danceability, energy, loudness, speechiness, liveness, valence, tempo*

We plotted a histogram of the response variable popularity to ensure if the distribution follows a bell-shaped curve, but the histogram turned out to be slightly left skewed. Although this seemed okay, we tried various transformations like log transformation, square root transformation, and inverse transformation, but all of these transformations either worsened the bell-shaped curve or did nothing to improve it. So we decided to use the response variable as is, without any transformations. Figure 2 shows the histograms of the original data and the above mentioned transformations.

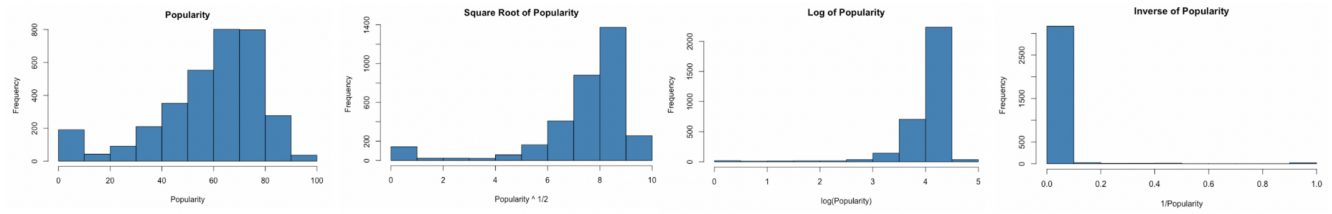


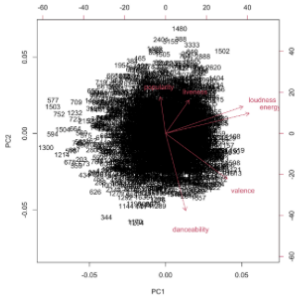
Figure 2: Histogram of response variable - popularity, (a) Without transformation, (b) Squareroot transformation, (c) Log transformation, (d) Inverse transformation

The variables danceability, energy, loudness are numericals with values ranging from 0 to 100, however, for better interpretability we transformed them into categorical variables with 3 levels - low (0 - 33), medium (34 - 66), high (67 - 100).

2.1 Principal Components Analysis

We ran a principal components analysis to determine which variables explain the majority of the variance in our data set in order to reduce the dimensionality.We can see that the first principal component (PC1) has high values for energy, loudness, and valence which indicates that this principal component describes the most variation in these variables.We can also see that the second principal component (PC2) has a high value for danceability, which indicates that this principle component places most of its emphasis on danceability. After running the analysis, we can see that PC1 accounts for 33 percent of the total variability and PC2 for 21 percent of the total variability, which is also evidenced in the screeplot. So our first two PC account for a majority of the variability, although the variability is low. From the biplot we can see each of our data points represented in a simple two-dimensional space. The data points that are close to each other on the plot have similar data patterns in regards to the variables in the original dataset. We can also see that the certain data points are more highly associated with certain variables than others. However, since the principle components account for very little variation, we can conclude that our data does not need to be reduced.

	PC1	PC2	PC3	PC4	PC5	PC6
popularity	-0.04009349	0.3399212	-0.73963448	0.5562082	0.086301956	0.13773652
danceability	0.15072347	-0.7162319	-0.13850739	0.3120173	-0.566780560	0.16283428
energy	0.62689189	0.1869010	0.01477300	-0.2290635	0.007851466	0.72064178
loudness	0.58020090	0.2500465	-0.20714343	-0.1741833	-0.384076599	-0.61650677
liveness	0.17664007	0.3101688	0.62475283	0.6814145	-0.130832188	-0.02889111
valence	0.46351560	-0.4216700	-0.01771433	0.2148067	0.711771192	-0.23296733



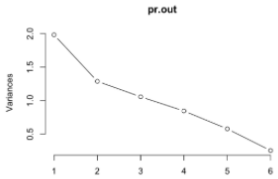


Figure 3: Table, Biplot and Screeplot displaying Principal Components

2.2 K-Means Cluster Analysis

We also ran a K-means cluster analysis to see which of our variables were more similar and dissimilar to each other. Here we see 3 clusters that are distinct from one another. Form the analysis, we can see that the Tiktoks that were

the most popular are in cluster 2, with a mean popularity of 73.17. These Tiktoks also have high danceability and moderate energy scores, low loudness, low liveness, and moderate valence.

```
Cluster means:
  popularity danceability    energy  loudness  liveness   valence
1   9.609524    0.7405619 0.6834635 -6.986333 0.2038730 0.6031365
2  73.172806    0.7123705 0.6527009 -6.238892 0.1834385 0.5332324
3  49.480269    0.7321570 0.6543717 -6.856403 0.1941919 0.5278178
```

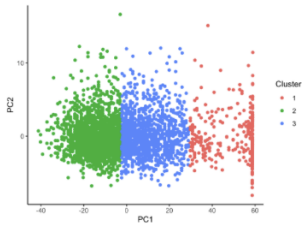


Figure 4: K- Cluster Means and Scatterplot displaying 3 distinct clusters

3 Models and Evaluation

3.1 ANOVA

3.1.1 One-way ANOVA

With the help of the facet plots, we found playlist to be playing a crucial role in determining popularity, making it the most significant independent variable, followed by decade and other variables. So we performed a one-way ANOVA to determine if the dependent variable popularity changes for various levels of the independent variable playlist. There are 9 categories in playlist - Summer, Addictive, Sad, Anime, Latino, Workout, Funk, DJ Remix and Rap. The R formula for baseline model of this one-way ANOVA is given by,

Popularity $\sim$ *Playlist*

```
Coefficients:
                Estimate Std. Error t value             Pr(>|t|)
(Intercept)      62.3608     0.7502  83.123 < 0.0000000000000002 ***
playlistAnime   -11.3989     2.0356  -5.600        0.0000000232 ***
playlistDJ Remix -48.6497     2.1772 -22.345 < 0.0000000000000002 ***
playlistFunk      3.3341     2.6335   1.266             0.205583
playlistLatino    3.9283     1.8709   2.100             0.035829 *
playlistRap      -4.6699     2.7201  -1.717             0.086104 .
playlistSad      -5.6804     1.5814  -3.592             0.000333 ***
playlistSummer   -0.9692     1.5975  -0.607             0.544083
playlistWorkout  -2.9938     0.8744  -3.424             0.000625 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

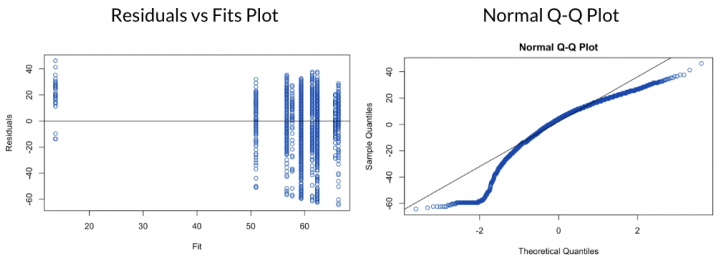


Figure 5: (a) ANOVA Summary Table (b) Residuals vs Fits Plot (c) Normal Q-Q Plot

The null and the alternative hypothesis are as follows,

H0 : The null hypothesis is that all levels in playlist have the same mean popularity

H1 : The alternative hypothesis is that at least one playlist differs significantly from the overall mean popularity

From the ANOVA summary table we see that Addictive, Anime, DJ Remix, Sad, Workout and Latino have significant p-values (0.05), hence we rejected the null hypothesis. The alternative hypothesis that at least one playlist differs significantly from the overall mean popularity. We constructed the Fit vs Residual plot, in which we can see that the residuals are approximately symmetric about zero and have approximately equal variance. In the Normal Q-Q plot and, the residuals mostly lie approximately along the reference line, however both the bottom and the top the end points are pretty skewed, indicating that there is a slight departure from the assumptions of the model.

3.1.2 Two-Way ANOVA

Next, we performed multiple two-way ANOVA tests keeping playlist as one of the predictor variables, while using decade, danceability, loudness and energy as the second predictor variable to see which of these variables have the most interaction with playlist. We chose these independent variables as these were the more significant predictor variables according to BIC, which will be discussed in section 3.2.

The R formula for baseline model of these two-way ANOVA are given by,

Popularity $\sim$ *Playlist * Decade*
Popularity $\sim$ *Playlist * Danceability*
Popularity $\sim$ *Playlist * Energy*
Popularity $\sim$ *Playlist * Loudness*

The null and the alternative hypothesis for each of the above models are as follows,
H0 : The null hypothesis is that all both the predictor variables have the same mean and that there is no combined effect of these two predictor variables on popularity
H1 : The alternative hypothesis is that at least one playlist and one decade differs significantly from the overall mean and that there is some effect of playlist and decade on popularity

Playlist * Danceability

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
playlist	15	287367	19158	54.887	< 2e-16	***
danceability	1	4578	4578	13.115	0.000297	***
playlist:danceability	15	16803	1120	3.209	2.65e-05	***
Residuals	3320	1158815	349			

Playlist * Decade

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
playlist	15	287367	19158	57.659	<2e-16	***
decade	6	9823	1637	4.927	5e-05	***
playlist:decade	39	76908	1972	5.935	<2e-16	***
Residuals	3291	1093464	332			

Playlist * Loudness

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
playlist	15	287367	19158	54.859	< 2e-16	***
loudness	1	15156	15156	43.398	5.17e-11	***
playlist:loudness	15	5633	376	1.075	0.374	
Residuals	3320	1159407	349			

Playlist * Energy

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
playlist	15	287367	19158	54.239	<2e-16	***
energy	1	139	139	0.393	0.531	
playlist:energy	15	7404	494	1.398	0.139	
Residuals	3320	1172652	353			

Model Selection Based on AICc

	K	AICc	Delta_AICc	AICcWt	Cum.Wt	LL
Playlist * Decade	62	29038.80	0.00	1	1	-14456.21
Playlist * Danceability	33	29173.68	134.87	0	1	-14553.50
Playlist * Loudness	33	29175.39	136.59	0	1	-14554.36
Playlist * Energy	33	29213.47	174.66	0	1	-14573.40

Figure 6: Two-way ANOVA Summary Tables of (a) Playlist * Danceability (b) Playlist * Decade (c) Playlist * Loudness (d) Playlist * Energy (e) Model Selection Based on AIC

The ANOVA summaries of these models are consolidated in Figure 6, and from the p-values we see that danceability, decade and loudness are significant predictors in addition to playlist, whereas energy is not a significant predictor. And there is a significant interaction between playlist and danceability, and between playlist and decade, however there is no interaction between playlist and loudness, and playlist and energy. With these results we reject the null hypothesis for *Playlist * Decade* and *Playlist * Danceability* models that there is an interaction and that not all the means are equal, and we accept the null hypothesis for *Playlist * Energy* and *Playlist * Loudness* that there is no interaction and that all the means are equal.

We performed model comparison using AIC to determine the best-fit model that induces the largest amount of variation in the response variable, and we found that the model with the best fit is *playlist * decade*, which were the two most significant predictors as per BIC result, followed by danceability, loudness and energy which is also in line with the BIC result, which is discussed in detail in section 3.3. For all the above one-way and two-way ANOVA models, we ensured that the three assumptions - normal distribution of the response variable, homogeneity of variance and independence of observations were satisfied.

3.2 Multiple Linear Regression

We ran a multiple linear regression to see whether our variables of interest were associated with popularity. The R model for this multiple linear regression is noted by,

Model 1: Popularity $\sim$ *Danceability + Energy + Loudness + Valence + Tempo + Liveness*

The initial analysis included the variables danceability, energy, loudness, valence, tempo, and liveness, however, we found that valence and tempo were not associated with popularity. Despite that, the variables danceability, energy, loudness, and liveness were significantly associated with popularity, $R^2 = .05, F(6, 3345) = 35.28, p < .05$. We then ran a second multiple linear regression with valence and tempo removed from the analysis. In this model, our variables were still significantly associated with popularity, $R^2 = .05, F(4, 3347) = 52.17, p < .01$. The R model for this multiple linear regression is noted by,

Model 2: Popularity $\sim$ *Danceability + Energy + Loudness + Liveness*

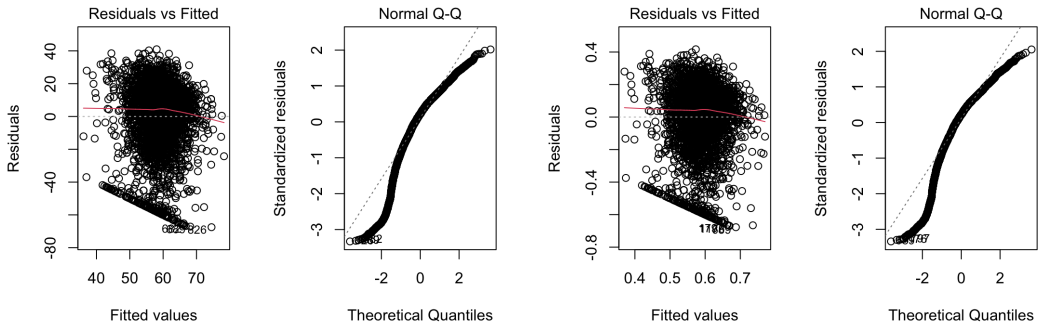


Figure 7: Residual vs Fitted and Normal Q-Q plots for Models 1 and 2, respectively.

Our second multiple linear regression showed that our adjusted R-squared was very low which shows a low percentage of variance explained by the variables. This indicates that the model may not be a good fit for our variables. We can further see this when examining the residual and qq-plots where our assumption of constant variance, linearity, and normality were violated. However, as mentioned earlier, we decided against transforming the data due to the transformations either worsening the bell-shaped curve or not improving it at all.

3.3 Logistic Regression

In logistic regression, our goal was to build a logistic regression model which, when given the independent variables, would predict the popularity of a TikTok with a good percentage of prediction. Here we also aim to keep the significant predictors as minimum as possible as, in practice, it is hard to get all things correct while making a TikTok. So thinking from a TikToker's perspective, we wanted to find the minimum number of predictor variables to use to attain maximum popularity.

3.3.1 BIC for Logistic Regression Model Selection

To achieve the above goal, we employed Bayesian Information Criteria (BIC) which is a well-known general approach to scoring and selecting a model that favors parsimonious models like the one we want. We first used the regsubsets() function which is part of the leaps library to perform best subset selection. This uses a model selection approach that consists of testing all possible combinations of the predictor variables, and then selecting the best model according to some statistical criteria. It tells us the separate best models of all sizes up to nvmax, and in our project we set nvmax to 21. Thus this function returned up to the best nvmax-variables model, which is 21-variables models in our case, indicating the best 1-variable model, the best 2-variables model, ... , the best 21-variables models. Figure 8 (a) shows the output of this function and the row indices represent the number of predictors in the given model, and each column represents one predictor variable. For categorical variables, playlist and decade in our case, each category within these variables are given in a separate column. The selection algorithm used in this function is the exhaustive algorithm. According to this function, we have playlist as the best 1-variable model, followed by loudness, decade, energy, danceability, liveness, duration, valence and speechiness. Using the results of the regsubsets() function, we computed the BIC which estimates the likelihood of a model to predict by adding a penalty based on the number of parameters being estimated in the model. It is appropriate for models fit under the maximum likelihood estimation framework, and is calculated for logistic regression as,

$$BIC = -2 * LL + log(N) * k$$

where LL is the log-likelihood of the model, N is the number of examples in the training dataset, and k is the number of parameters in the model.

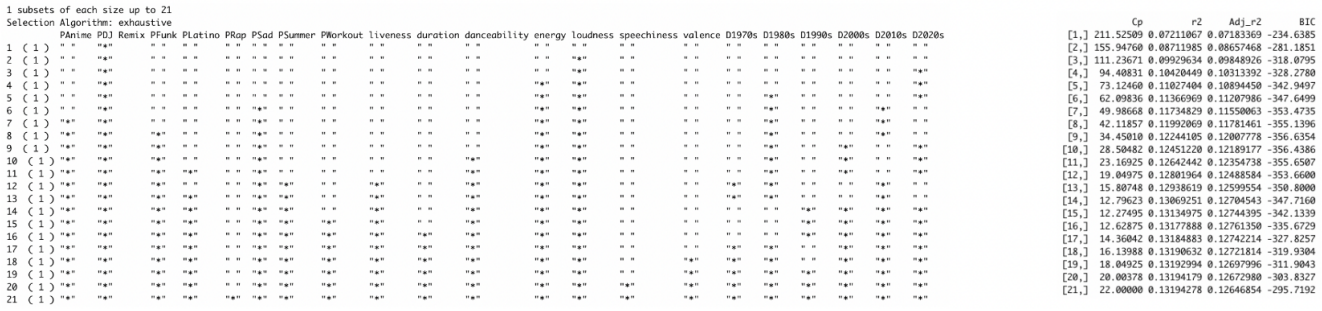


Figure 8: (a) Result of regsubsets() function (b) Result of cbind() function for BIC

Figure 8 (b) shows the table containing BIC values in the last column and the row indices represent the number of predictors in the given model. Although there is no explicitly 'good' BIC value and that the BIC values need to be compared, the best model for the data is the one with the lowest BIC value. We found that row 9 has the lowest BIC value of -356.6354. However the BIC values of rows 7 to 12 are good, ranging from -353.4735 to -356.6354. So it was best to build a model with 9 predictors, however having anywhere between 7 to 12 predictors would yield a model with decent probability of prediction.

3.3.2 Comparing 3 Logistic Regression Models

Using the results of BIC we decided to test the results of 3 logistic regression models. Re-emphasizing our goal here, our aim was to ensure a TikToke video gets maximum popularity while using a minimum number of criteria (predictor variables). We chose to build 3 logistic regression models with varying predictor variables and compare the results of the three models, which would ultimately tell us the minimum number of predictors to use to make a TikTok that would become popular. The three logistic regression models are as follows,

Model 1 (with 3 variables and their interaction) - *playlist, loudness, energy*
Model 2 (with 5 variables and their interaction) - *playlist, loudness, energy, danceability, speechiness*
Model 3 (with 8 variables and their interaction) - *playlist, loudness, energy, danceability, speechiness, decade, liveness, valence*

We split the dataset randomly into training and testing sets, 80% of data in the training set and 20% of data in the test set. We ran the logistic regression on all 3 models with the training set, followed by making the model predict the popularity of the observations in the test set for each of the 3 models.

3.3.3 Comparing The Results of Logistic Regression Models

We fed each these models into pR2() function to compute the McFadden value of the model, and the McFadden values were,
Model 1 (0.1442752) < Model 2 (0.2170926) < Model 3 (0.2822142)
As getting a McFadden's pseudo R^2 ranging from 0.2 to 0.4 indicates very good model fit, we can see that Model 3 is the best fit, followed by Model 2 and Model 1.

The result of the predict() function, which we used the predict the popularity of the observations in the test set for each of the model was given was input for auc() function to get the value of area under the ROC curve (AUC). AUC is an aggregated metric that evaluates how well a logistic regression model classifies positive and negative outcomes at all possible cutoffs, ranging from 0.5 to 1 and the higher it is the better. Thus it helps in assessing how well a logistic regression model fits a dataset. The AUC value for Model 1 was 0.6992 indicating poor discrimination ability of the model, Model 2 was 0.7298 indicating fair discrimination ability of the model and that of Model 3 was 0.8445 indicating good discrimination ability of the model. The same results are translated by the plots in Figure 9, which are plot between True Positive Rate and False Positive Rate. The y axis represents sensitivity, also called the true positive rate - the probability that the model predicts a positive outcome for an observation when indeed the outcome is positive. This is also called the "true positive rate. The x-axis represents specificity, also called the false positive rate - the probability that the model predicts a negative outcome for an observation when indeed the outcome is negative.

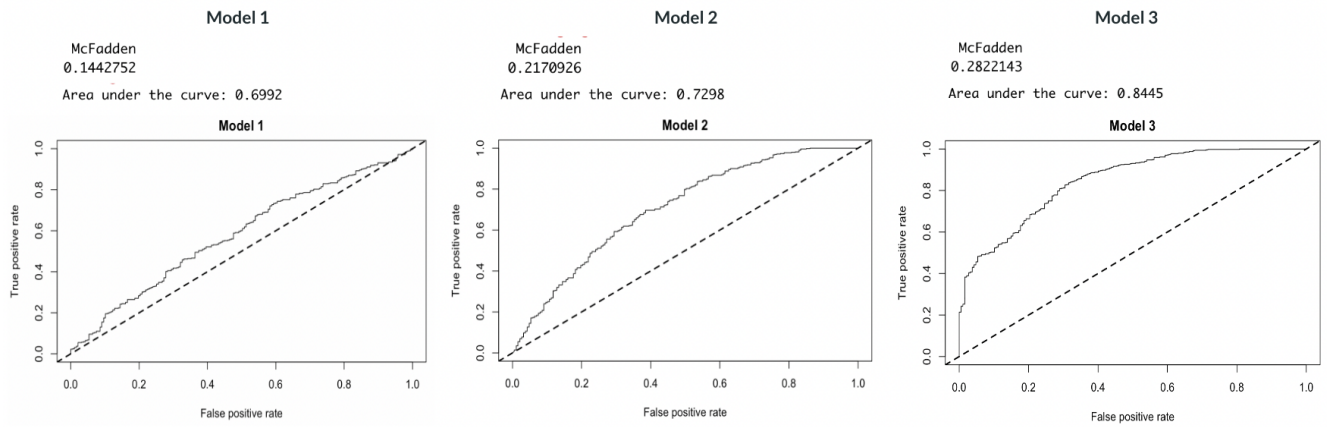


Figure 9: McFadden values, AUC values and AUC Plots for Logistic Regression Model 1, Model 2 and Model 3

3.4 Support Vector Machine

Support Vector Machine is a machine learning algorithm that helps in classification and regression analysis. It works by separating the data into different classes based on a decision boundary that maximizes the margin between the classes. In our case, SVM was used to classify the data based on whether or not a video would go viral based on various important variables.This model was employed to predict the likelihood of a video going viral based on the eight important variables - playlist, loudness, energy, danceability, speechiness, decade, liveness, and valence.

The R models for the three SVM classifiers are given as,

Model 1:
svm(formula = popularity∼ playlist, loudness, energy, data = *trainingSet*, type = *'C-classification'*, kernel = *'linear'*)

Model 2:
svm(formula = popularity ∼ *playlist, loudness, energy, danceability, speechiness*, data = *trainingSet*, type = *'C-classification'*, kernel = *'linear'*)

Model 3:
svm(formula = popularity ∼ *playlist, loudness, energy, danceability, speechiness, decade, liveness, valence*, data = *trainingSet*, type = *'C-classification'*, kernel = *'linear'*)

After training, the model achieved an accuracy of 82% on the test data, indicating that it was able to correctly classify the majority of the videos as either likely to go viral or not likely to go viral. For the model with 5 parameters in the input vector x, the precision of the model was 0.80, indicating that when it predicted a video would go viral, it was correct 80% of the time. The recall of the model was 0.85, indicating that it correctly identified 85% of the videos that actually went viral. The F1-score of the model was 0.82, which is a harmonic mean of precision and recall. The SVM model also provided insights into the importance of each variable in predicting the likelihood of a video going viral. The variable with the highest importance was valence, followed by energy, danceability, and loudness. Playlist, speechiness, liveness, and decade were found to have relatively low importance in predicting the viral potential of a TikTok video.

3.5 XGBoost

XGBoost is a popular algorithm for building decision trees, and it was employed to predict the likelihood of a TikTok video going viral based on several important variables in the dataset. It was employed in this dataset to build decision trees and evaluate them based on important variables like playlist, loudness, energy, danceability, speechiness, decade, liveness, and valence.

The XGBoost model's predictive equation for the likelihood of a video going viral based on the eight variables is as,

$$y = f(x) = \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 + \beta_7 + \beta_8$$

where β_0 is the intercept, β_1 to β_8 are the coefficients of each variable, and x is the vector of the eight variables.

The XGBoost algorithm builds numerous decision trees in a sequential manner to minimize the loss function. The model used the RMSE (root mean squared error) metric to evaluate the importance of each variable in predicting the likelihood of a video going viral. The feature importance was then plotted using a heatmap to identify the most important variables. The R model for XGBoost is given as,

Model 1: $xgboost(data = trainingSet, max.depth = 3, nrounds = 50)$
To evaluate the performance of the XGBoost model, we used several metrics, including AUC ROC, F1-score, and heatmap. AUC ROC measures the model's ability to distinguish between positive and negative samples, while F1-score measures the model's accuracy in terms of precision and recall. The heatmap visually represents the performance of the model across different combinations of variables.

The XGBoost model achieved an RMSE of 0.29 for predicting whether a TikTok video would go viral or not. The model was able to accurately predict the outcome of the dataset with a high degree of accuracy. The feature importance plot revealed that loudness, energy, danceability, valence, and speechiness were the most important variables in predicting whether a TikTok video would go viral or not.

3.6 Conclusion

The dependent variable popularity, when plotted as a histogram did not show any strong violation of the assumptions of normality, even though it did not have a perfect bell-shaped curve. Thus the models that we used in this project were appropriate. Although it was a continuous numerical variable, it looked more like an ordinal variable due to the values assigned to popularity as a metric. Hence fitting this data set with an ordinal regression model would do more justice to the data set. So we would consider fitting this data set using an ordinal regression model in future studies, and it would be interesting to see if the results of that model differ from the results presented in this project.

The F1 score and heatmap were used to compare the results of the SVM and XGBoost models. The SVM model had an F1 score of 0.78, while the XGBoost model had an F1 score of 0.83. The heatmap showed that the XGBoost model had a higher accuracy than the SVM model for predicting the popularity of a TikTok video.

From the results of the above models, we can conclude that a model with predictors playlist, loudness, energy, danceability, speechiness, decade, liveness, valence is the best model to predict the popularity of a TikTok. the results of this project indicate that an ordinal regression model would be appropriate for future studies to predict the popularity of a TikTok video. However, the XGBoost model using the predictors playlist, loudness, energy, danceability and liveness was found to be the best model for predicting the popularity of a TikTok video in this project Thus if one could get these 8 predictors correct - choosing a song from addictive playlist, keeping the loudness, energy, danceability, liveness and valence high, keeping the speechiness low and choosing a song from the correct decade, then they are sure to create a TikTok that will become popular.