COLUMBIA UNIVERSITY

Spotify Predictive Analytics Competition
Applied Frameworks 1
Applied Analytics

Ishu JASWANI

December 8, 2022

Date Performed: November 18, 2022
Instructor: Professor VISHAL LALA

1 Introduction

The Spotify predictive Analytics competition from kaggle helped in becoming proficient in coding as all the regression models taught in class were used in the analysis. Performed Data wrangling to clean the data given by Spotify. In all this project helped develop skills such as :

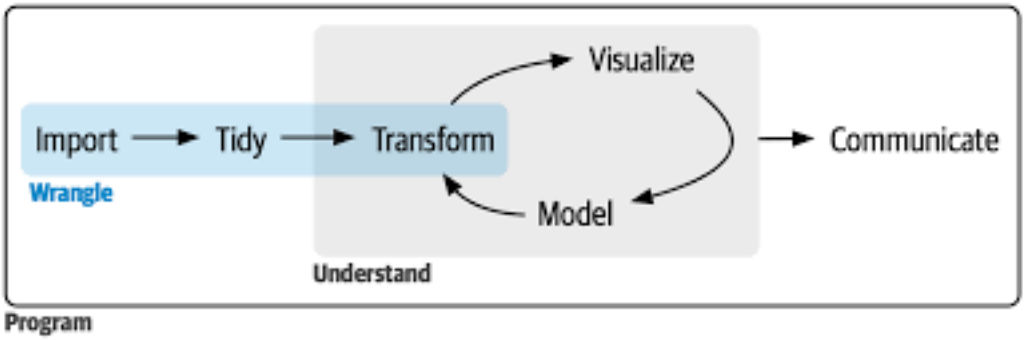
a. Data wrangling/cleaning

b. Dealing with missing Data

c. Data Visualisation

d. Data Modeling

e. Predicting and tuning models



2 Data Wrangling

The biggest task when it came to wrangling was to sort the genres and performers column.

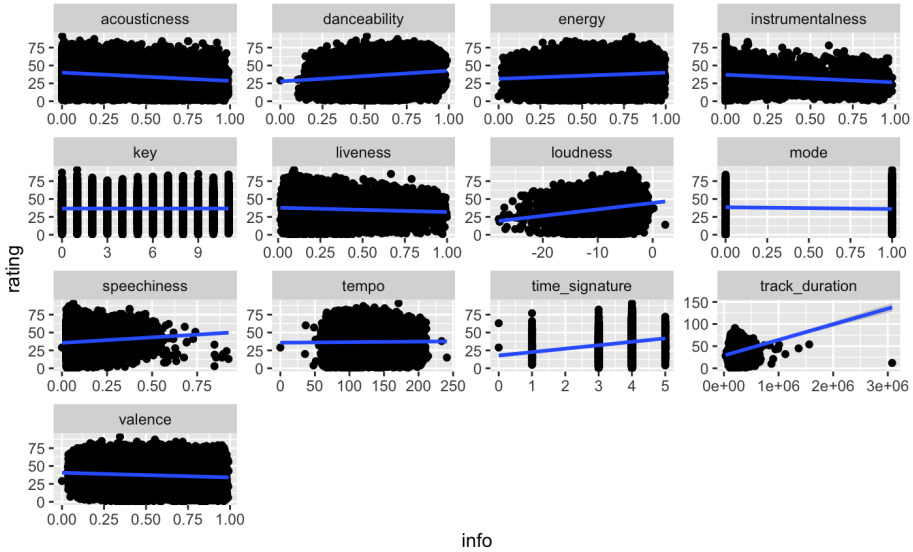
a. The genres column consisted of a list in every row in order to clean and wrangle it the following steps were taken:
   (a) The missing data was removed around 1% of the data was removed. This is fine as we are not removing a major chunk of the data and predicting categorical data especially lists would become tedious and impractical.
   (b) The next step was to split the lists into different rows and hence increase the number of columns (Train has 995 and test 628)
   (c) Then all the genres were one hot encoded which mean for each song there was a yes or no indicator for each genre
   (d) finally, I counted the highest number of yes of genres for each song and chose the top 10 one-hot encoded genres to use for predictions.

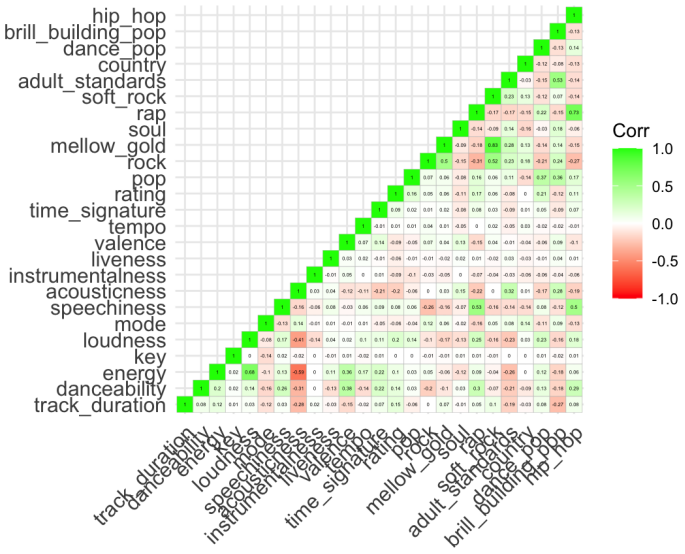
b. The Performers column was dropped as there was no practical reasoning to wrangle or clean the data. However, this might be something i would do differently if given more time for the project. I would have tried out ways in which i could bin the performer into shorter factors which would increase some predictive power.

Rest of the columns were kept the same!

3 Data Visualisation

To see how the data is correlated to ratings of the songs. I plotted graphs which helped assess the same.
I used ggplot and ggcorplot to plot the data.





4 Models

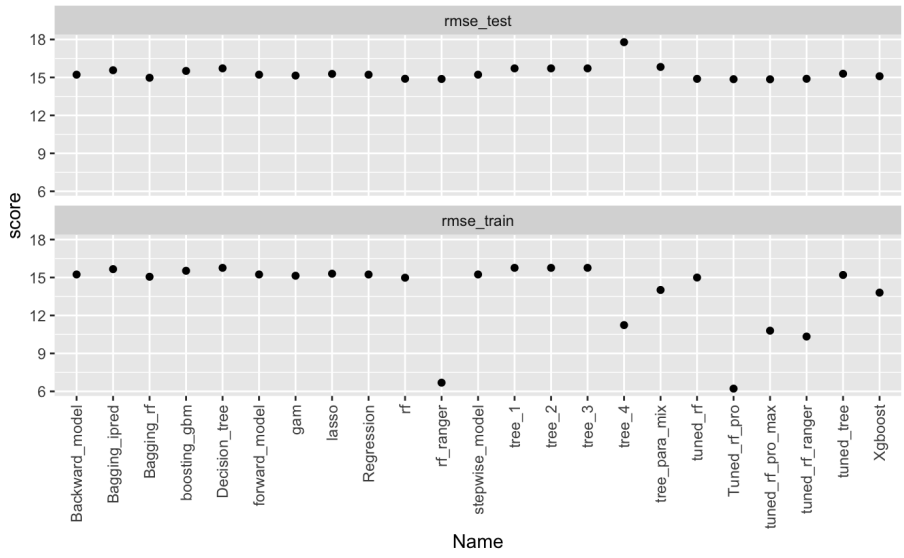
Using the there is no free lunch in this world theorem taught in class, I had to run all the models. since the data was too huge i split the data into train and test and ran models to see which model was giving the lowest RMSE.

a. Having run 23 models. I understood that either the hyper-parameters would further decrease the RMSE of the model or more exhaustive data wrangling would.

b. Another reason which could increase the score of the model is having more predictors and a faster laptop.

I found out that random-forest using the ranger package was giving the lowest RMSE ($Tunedrfpromax$ = 14.85370) in the test data.
The highest RMSE ($tree4$ = 17.78784) in the test data was given by a decision tree model whose complexity parameter was very low. This made me understand that there was an overfitting problems.
Once i understood that ranger was best predicting the ratings data I knew i had to tune it further. I tried out multiple variations such as : increasing number of trees, increasing number of Mtry's to consider, increased the number of min-node-size. On doing this i got a slightly better model. which i finally used to get the lowest rmse with the main scoringData. ($rmse$ = 14.78028)
All the other models used before gave sub par results. The ranger random-forest algorithm got me in the 14's.

Models List		
Model Name	RMSE Train	RMSE Test
Regression	15.237824	15.21449
forward model	15.239091	15.21729
Backward model	15.239091	15.21729
stepwise model	15.239091	15.21729
gam	15.138456	15.14827
lasso	15.301234	15.27686
Decision tree	15.762926	15.71594
tree1	15.762926	15.71594
tree2	15.762926	15.71594
tree3	15.762926	15.71594
tree4	11.238135	17.78784
tree para mix	14.013763	15.82891
tuned tree	15.198665	15.29275
Bagging ipred	15.661357	15.56915
Bagging rf	15.057197	14.97413
rf	14.981823	14.89492
tuned rf	14.995980	14.88767
rf ranger	6.686142	14.87837
tuned rf ranger	10.332721	14.89331
boosting gbm	15.529935	15.51316
Xgboost	13.803574	15.09583
Tuned rf pro	6.216722	14.86214
tuned rf pro max	10.795823	14.85370



4.1 Take-back from the models

Always wrangle the data in such a way it includes the essence of all the predictor values so that the model can have maximum predictive power.
Always run multiple models to see which model works better

4.2 Limitations of the models

The models took hours to run. Reducing the parameters through PCA or a shrinkage model and then using them in all the models would have reduced the time. However, from a machine learning engineers perspective any data even not significant can add come predictive power. Therefore, I have Included all the variables except ID and performer (why explained in the Data wrangling section)