**Preliminary stuff**

The regression task was built on a dataset consisting of 518 features extracted from 106,574 individual music tracks. Besides the feature data, 52 metadata features were available. The assignment itself consisted of the investigation and prediction of two metadata parameters for a given track, namely the number of times the track was listened, and the date on which the album came out. Data on these two outcomes was found in the columns *track\_listens* and *album\_date\_released* respectively. As feature data and metadata were stores in separate files, both were joined using a common id (column *track\_id*).

Before any further preprocessing was done, the complete dataset was split into a training and testing set, with the size of the training set being 80% of the size of the initial dataset. This process was done twice. Once for every outcome. We decided against using one common train-test split for both regression tasks, as the initial number of valid instances for the *album\_date\_released* variable was smaller than the initial number of valid instances for the *track\_listens*. More specific, an instance was deemed ‘valid’ when the outcome was not missing. The *track\_listens* variable contained no instances with missing data. As such, the training set consisted of 85259 instances. The *album\_date\_released* variable consisted of 36,280 instances with missing data (34.04% of full dataset). All instances with missing outcome data were removed, as the presence of outcome data is essential in the process of model building and evaluating. The training set for the *album\_date\_released* variable consisted of 56,235 instances. During this process – and technically being out of scope for this assignment- it was noted that missing outcome data within the *album\_date\_released* for a given track was defined on the album level, rather than on the track level. This meant that all tracks on a given album either had missing data for the outcome, or none of the tracks on an album had. This notion was further explored in the prediction phase.

The training set, consisting of predictor and outcome data were used to train and optimize various regression models and data reduction techniques. The test set on the other hand was only used in the final step of the project where we evaluate predictive capabilities of the various models. In the following sections, it will be explicitly stated if the test set played a relevant part in the section. In all other cases, it may be implicitly assumed that the training set was used.

Within the projects, main steps in the data analytical process were quite similar for both outcome variables. As such, we will report on both regression tasks simultaneously, pointing out differences between the two outcomes where relevant.

**Exploratory analysis and data cleaning**

A descriptive analysis was done on both variables to get to know the data and to serve as a starting point for further data cleaning. First, we assessed missing data in the dataset. By design, the outcomes did not have any missing data, as all ‘missing instances’ were removed. For the feature data, no report of any missing data could be made. The proposed imputation technique of multiple imputation by chained equations (MICE) was therefore ignored.

Next, we calculated summary statistics for the outcomes, resulting in both outcomes being transformed. The *track\_listens* outcome was characterized by a right skewed distribution with a very fine tail (See Appendix B: figure 1). The average amount of track listens ranged from XX to XX, with mean 2,326.23 and median 763.0 The outcome was subsequently log transformed in order to obtain a more Gaussian shaped distribution (Appendix B: figure 2). Before the transformation of the outcome, one instance with zero track listens was removed from the training set, as log transforming a value of zero is not possible.

Whereas the *track\_listens* outcome had a right skewed distribution shape, the opposite was true for *album\_date\_released* (Appendix B: Figure 3). The outcome was first transformed to a more numeric format by calculating the delta (in days) between the outcome and the arbitrarily chosen date of 1900-01-01. Mean delta was XX, median XX with minimum and maximum delta being XX and XX respectively. The outcome was consequently square transformed (Appendix B: Figure 4).

Outliers were assessed on a univariate and multivariate basis. In the univariate case, outliers were flagged using the IQR method. The IQR method was not implemented further, as nearly 99% of instances exceeded the IQR cut-off for at least 1 feature. For more information on the IQR method, see Appendix B: preprocess-IQR)

Next, the Mahalanobis distance between every instance and the relevant training set was calculated and evaluated through a chi-square test. Instances significant on the .05 level were removed. For the *track\_listens* variable, 15,401 instances were removed from the training set (18.06%), whereas 10442 training instances were removed for the *album\_date\_released* outcome (18.57%).

As a last step, we scaled both datasets using a min-max scaler. The min-max scaler was deemed most optimal out of an evaluation between three scalers: a min-max scaler, a standard scaler, and a robust scaler. To evaluate all scalers, 15 randomly chosen variables were scaled using all scalers. Next, boxplots for every scaled variable were assessed and evaluated (see Appendix B: figures 5 and 6 for the evaluated plots).

**Evaluative measures**

During the phases of dimension reduction and prediction, multiple models and workflows were assessed and evaluated. For the evaluation of algorithms, we consistently used three evaluative metrics, the R squared (R2), the root mean square error (RMSE) and the median absolute error (MAE). When evaluating models or optimizing hyperparameters, these metrics were used as evaluation metrics. However, an attentive reader will notice that the RMSE and MAE metrics are sometimes used in their negative counterparts. The reason for this is that the optimalization for MAE and RMSE in theory is a minimalization problem, whereas optimalization of R2 lies in maximizing R2. By calculating the negative RMSE and MAE respectively, the optimalization problem becomes a maximalization task for every metric rather than an optimization problem dependent of the metric used.

**Dataset reductions**

For each outcome variable, three distinct dimension reduction techniques were implemented, namely: principal component analysis (PCA), elastic net and local linear embedding (LLE). For each outcome variable and dimensionality reduction technique, a similar workflow was used. First, a reduction model was fit on the training set. Next, both the training and test sets were transformed using the reduction model originally fit on the training set. Resulting datasets were stored separately to be used further up in the project.

In context of PCA, two different thresholds were implemented for the number of variables to be retained. More specifically, a specific subset of variables was withheld so that the amount of explained variance was 95% in the first implementation and 99% in the second. The number of features retained differed slightly between outcomes. For the *track\_listens* variable, a subset of 160 features was enough to explain 95% of the variance within the original (cleaned) dataset. In the 99% explained variance case, a subset of 293 features had to be withheld. For the outcome of *album\_date\_released*, subsets of 164 and 297 features were obtained in order to explain respectively 95% and 99% of the variance within the original dataset.

Next, we reduced both datasets by use of elastic net regression. For each outcome variable, the transformed outcome was used as the dependent variable in the model fit. The hyperparameter for the mixing parameter (“l1\_ratio”) was optimized in a separate research project. More specifically, a cross validated grid search with 10 folds was performed on the training set for the *track\_listens* variable with transformed outcome. Four candidates for the hyperparameter were assessed. The grid search did not provide very clear result. As a consequence, we chose a value of 1 for the hyperparameter, which made the elastic net regression to function like a general lasso regression. To optimize runtime and probability of convergence, we defined the maximum number of iterations at 10,000 and the selection of variables within the model fit to be ‘random’ instead of ‘cyclic’. The elastic net reduction resulted in a number of subset features of 176 for the track\_listens outcome and 154 for the *album\_date\_released* outcome.

As a last method of dimensionality reduction, we researched and implemented a modified local linear embedding. We chose a dense solver to compute eigenvectors and standard dense matrix operations for decomposition of eigenvalues. The number of neighbors was chosen to be 16, identical to the number of neighbors used for the K-Nearest Neighbor prediction algorithm (see further).

**Regressors**

Four regressors were evaluated on each reducted dataset for each outcome variable. The four regressors used were: ordinary least square regression (OLS), a decisiontree, a K-Nearest Neighbors regressor and a random forest.

Hyperparameter optimization was done for a subset of hyperparameters in relevant places. A fixed process for the optimization of hyperparameters was used. In this process, we did a cross validated 10-fold grid search along a parameter grid. Results were stored, plotted, and evaluated. When informedly choosing hyperparameters, we tried to optimize prediction ‘power’ as well as computational load.

For the OLS regression, no hyperparameters were tuned, as OLS regression does not have any hyperparameters.

Appendix B:

Preprocess-IQR:

Outliers were flagged univariately using the IQR method. The IQR method was not implemented further, as nearly 99% of instances in both training sets exceeded the IQR cut-off for at least 1 feature.

For the *track\_listens* outcome, 99.01% of all instances in the training set exceeded the IQR cut-off for at least one feature (n=84415). On average, an instance exceeded IQR cut-offs for 21.23 features, with a standard deviation of 19.56.

For the *album\_dates\_released* outcome, 98.99% of all instances exceeded IQR cut-offs for at least one feature (n=55669). On average, an instance exceeded IQR cut-offs for 21.27 features, with a standard deviation of 19.81.