Music & Speech Detection in Radio Broadcasts

A brief look on correlation

- Pearson correlation has a range of values from +1 to -1.
- Zero indicates no association between two variables, whereas +1 and -1 indicate positive association and negative association respectively.
- As the value gets closer to +/- 1, the association gets stronger.
- In a positive association, when the value of one variable increases, the value of the other also increases. While, In a negative association, they go in opposite directions.

- For the Speech data, we can see a positive correlation among the features, from feature 43 to the last one (Figure 1).

- For the Music data, we can see positive and negative correlation, with positive in higher number. However, it is not in a specific region as in Speech data (Figure 2).

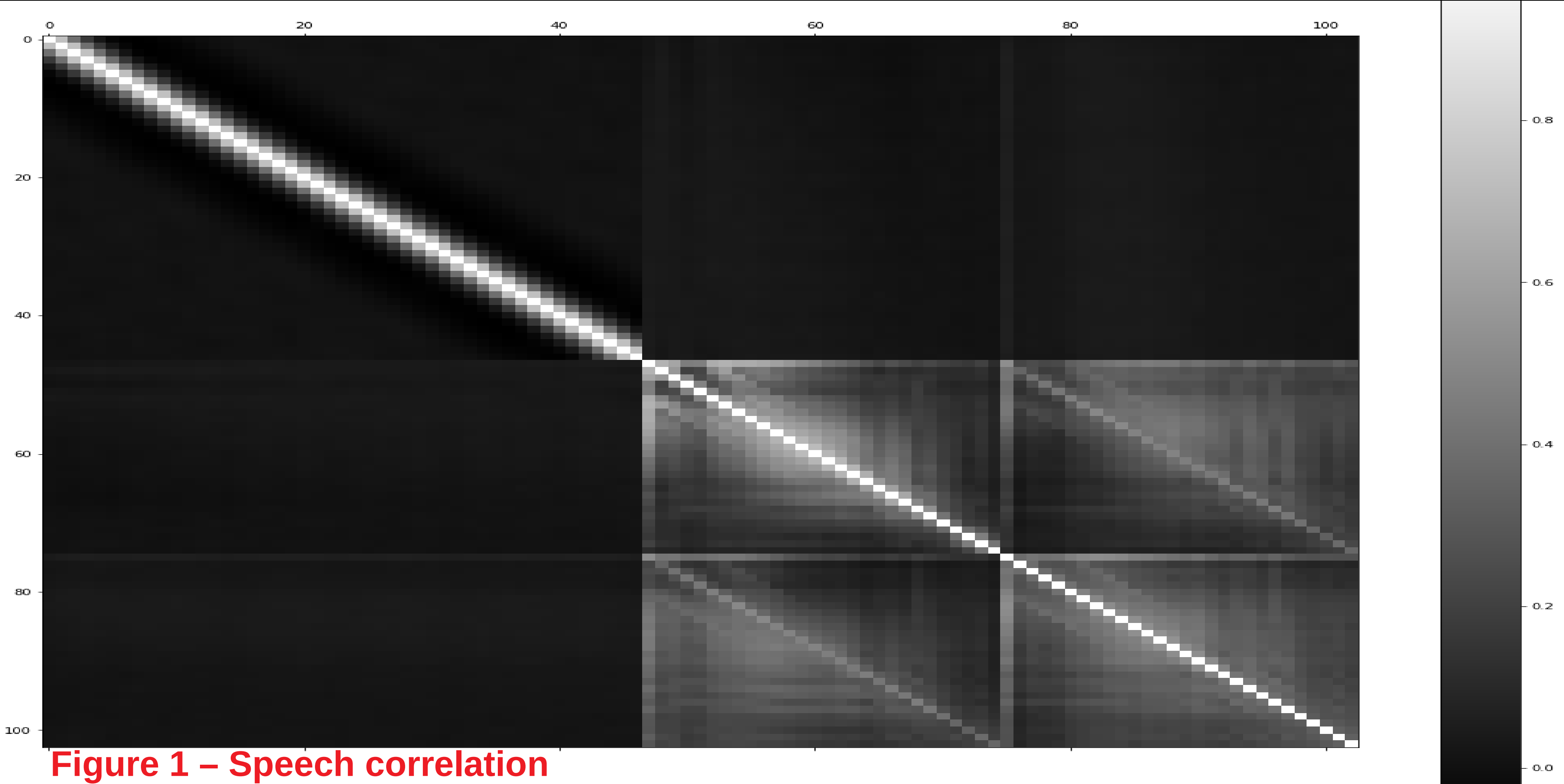


Figure 1 – Speech correlation

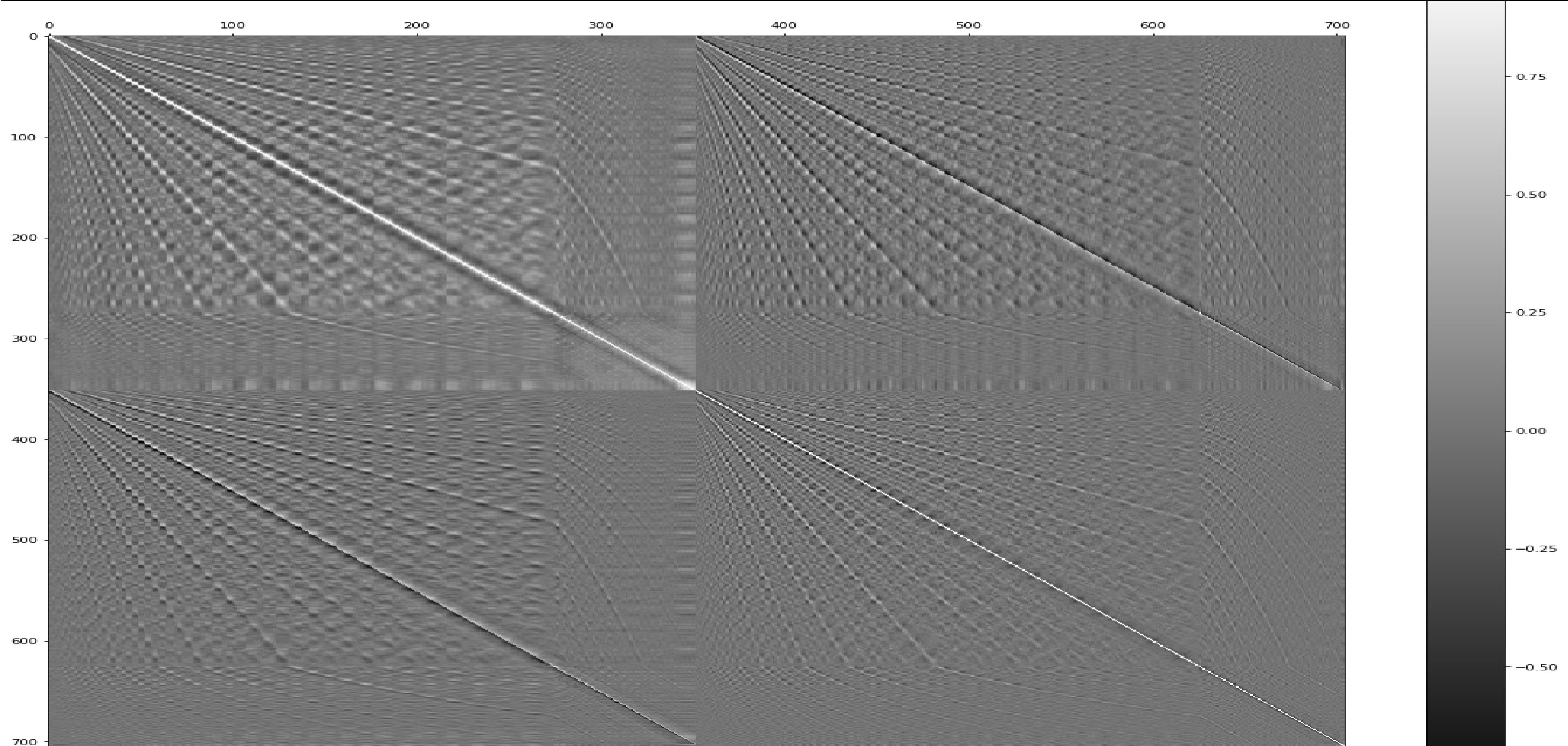
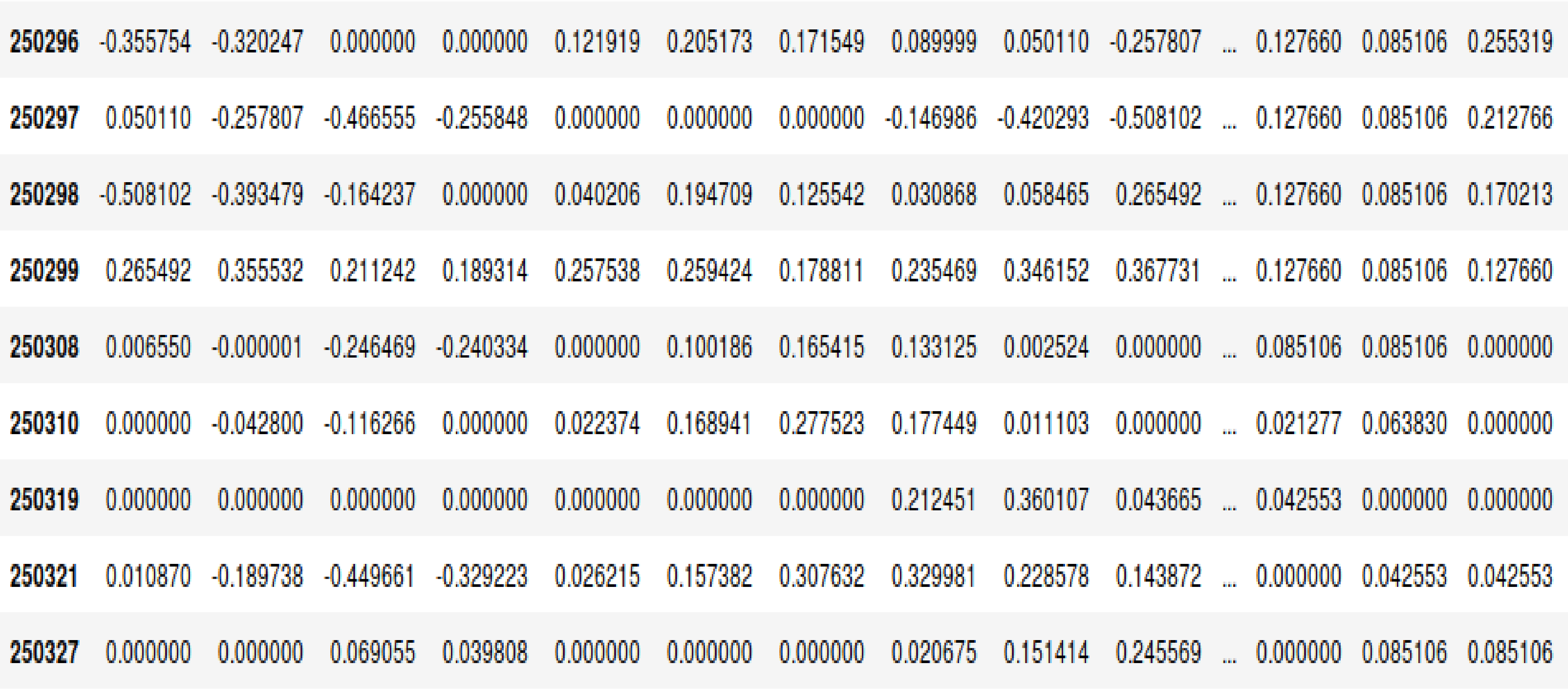


Figure 2 – Music correlation

Detecting outliers

- The Isolation Forest Algorithm was used to detect outliers, it is an ensemble method, based on decision trees.
- The algorithm first randomly selects a feature and then randomly selects a split value between the maximum and minimum values of the selected feature.

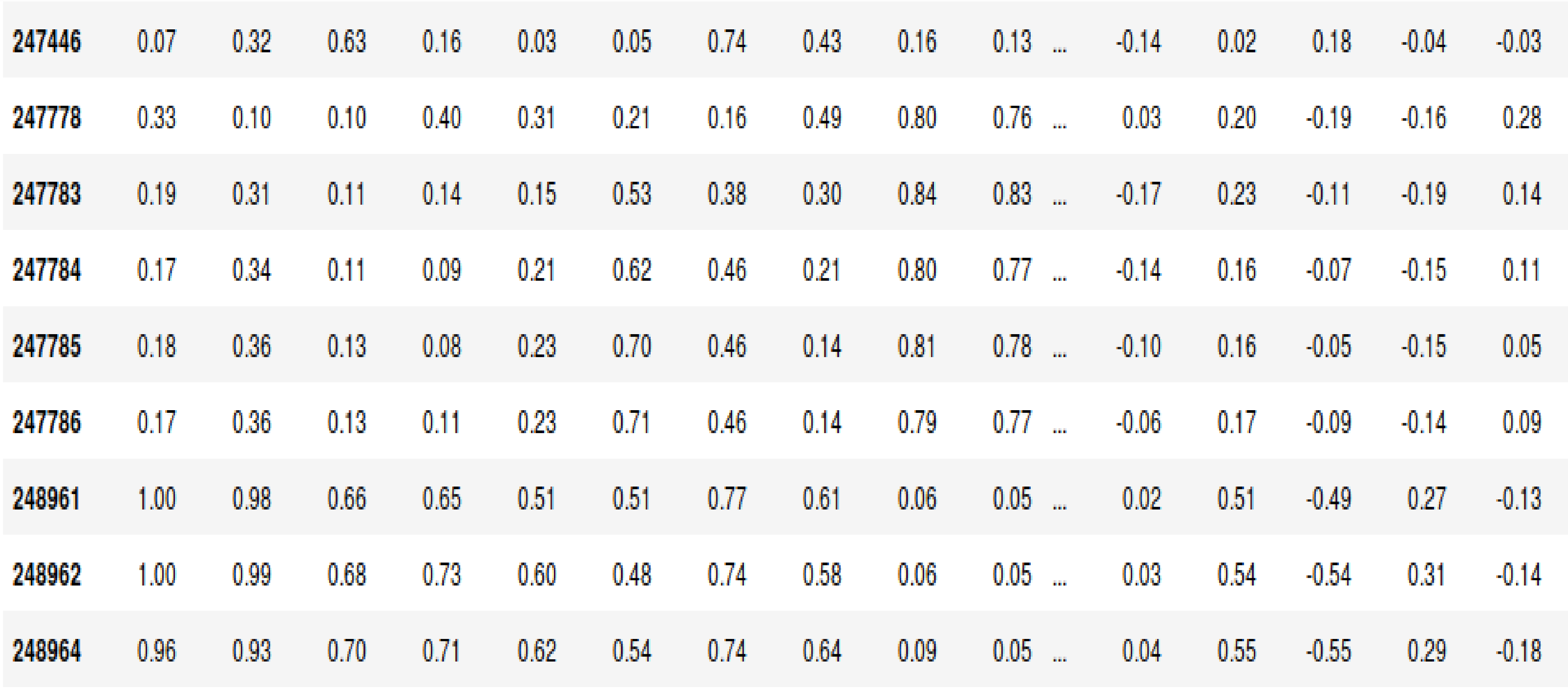
- For the Speech data, 12854 rows were detected as outliers, and then eliminated from the dataset (Figure 3).
- For the Music data, 4466 rows were detected as outliers, and then eliminated from the dataset (Figure 4).



250296	-0.355754	-0.320247	0.000000	0.000000	0.121919	0.205173	0.171549	0.089999	0.050110	-0.257807	...	0.127660	0.085106	0.255319
250297	0.050110	-0.257807	-0.466555	-0.255848	0.000000	0.000000	0.000000	-0.146986	-0.420293	-0.508102	...	0.127660	0.085106	0.212766
250298	-0.508102	-0.393479	-0.164237	0.000000	0.040206	0.194709	0.125542	0.030868	0.058465	0.265492	...	0.127660	0.085106	0.170213
250299	0.265492	0.355532	0.211242	0.189314	0.257538	0.259424	0.178811	0.235469	0.346152	0.367731	...	0.127660	0.085106	0.127660
250308	0.006550	-0.000001	-0.246469	-0.240334	0.000000	0.100186	0.165415	0.133125	0.002524	0.000000	...	0.085106	0.085106	0.000000
250310	0.000000	-0.042800	-0.116266	0.000000	0.022374	0.168941	0.277523	0.177449	0.011103	0.000000	...	0.021277	0.063830	0.000000
250319	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.212451	0.360107	0.043665	...	0.042553	0.000000	0.000000
250321	0.010870	-0.189738	-0.449661	-0.329223	0.026215	0.157382	0.307632	0.329981	0.228578	0.143872	...	0.000000	0.042553	0.042553
250327	0.000000	0.000000	0.069055	0.039808	0.000000	0.000000	0.000000	0.020675	0.151414	0.245569	...	0.000000	0.085106	0.085106

12854 rows × 104 columns

Figure 3 – Speech Outliers



247446	0.07	0.32	0.63	0.16	0.03	0.05	0.74	0.43	0.16	0.13	...	-0.14	0.02	0.18	-0.04	-0.03
247778	0.33	0.10	0.10	0.40	0.31	0.21	0.16	0.49	0.80	0.76	...	0.03	0.20	-0.19	-0.16	0.28
247783	0.19	0.31	0.11	0.14	0.15	0.53	0.38	0.30	0.84	0.83	...	-0.17	0.23	-0.11	-0.19	0.14
247784	0.17	0.34	0.11	0.09	0.21	0.62	0.46	0.21	0.80	0.77	...	-0.14	0.16	-0.07	-0.15	0.11
247785	0.18	0.36	0.13	0.08	0.23	0.70	0.46	0.14	0.81	0.78	...	-0.10	0.16	-0.05	-0.15	0.05
247786	0.17	0.36	0.13	0.11	0.23	0.71	0.46	0.14	0.79	0.77	...	-0.06	0.17	-0.09	-0.14	0.09
248961	1.00	0.98	0.66	0.65	0.51	0.51	0.77	0.61	0.06	0.05	...	0.02	0.51	-0.49	0.27	-0.13
248962	1.00	0.99	0.68	0.73	0.60	0.48	0.74	0.58	0.06	0.05	...	0.03	0.54	-0.54	0.31	-0.14
248964	0.96	0.93	0.70	0.71	0.62	0.54	0.74	0.64	0.09	0.05	...	0.04	0.55	-0.55	0.29	-0.18

Figure 4 – Music Outliers

4466 rows × 706 columns

Feature selection

- Feature selection was made using backward elimination with linear regression (ordinary least squares model).
- It iteratively removes the worst performing features. If the pvalue is above 0.05 then it is removed, else it keeps it.

- For the Speech data, 70 features were kept.
- For the Speech data, 506 features were kept.

Feature Selection – Speech and Music

- ['f000002', 'f000003', 'f000010', 'f000014', 'f000018', 'f000022', 'f000026', 'f000027', 'f000029', 'f000032', 'f000035', 'f000038', 'f000041', 'f000044', 'f000046', 'f000048', 'f000049', 'f000050', 'f000051', 'f000052', 'f000053', 'f000054', 'f000055', 'f000057', 'f000058', 'f000059', 'f000060', 'f000061', 'f000062', 'f000063', 'f000064', 'f000065', 'f000066', 'f000067', 'f000068', 'f000069', 'f000070', 'f000071', 'f000072', 'f000073', 'f000074', 'f000075', 'f000076', 'f000077', 'f000078', 'f000079', 'f000080', 'f000081', 'f000082', 'f000083', 'f000084', 'f000085', 'f000086', 'f000087', 'f000089', 'f000090', 'f000091', 'f000092', 'f000093', 'f000094', 'f000095', 'f000096', 'f000097', 'f000098', 'f000099', 'f000100', 'f000101', 'f000102', 'f000103']

- ['f000001', 'f000002', 'f000003', 'f000004', 'f000005', 'f000006', 'f000007', 'f000008', 'f000009', 'f000010', 'f000013', 'f000014', 'f000015', 'f000016', 'f000017', 'f000018', 'f000020', 'f000021', 'f000022', 'f000023', 'f000024', 'f000026', 'f000027', 'f000028',

  ...

  'f000672', 'f000673', 'f000674', 'f000675', 'f000676', 'f000677', 'f000681', 'f000682', 'f000684', 'f000686', 'f000687', 'f000689', 'f000691', 'f000692', 'f000693', 'f000694', 'f000696', 'f000699', 'f000700', 'f000701', 'f000702', 'f000703', 'f000704', 'f000705']

Hyperparameter tuning and Cross validation

- A Randomized Search Cross Validation was used.
- It does not try out all parameter values, a fixed number of parameter settings is sampled from the specified distributions.

- Logistic Regression and Random Forest were chosen after the validation made on Task 1.
- Gradient Boosting was included after validation, the accuracy achieved was similar to Logistic Regression and Random Forest.

Hyperparameter tuning and Cross validation - Speech

- Logistic Regression:

    Best parameters {'solver': 'liblinear', 'penalty': 'l1', 'C': 5}

    Best score 0.8980091691264209

- Random Forest:

    Best parameters {'n_estimators': 200, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': 50, 'bootstrap': False}

    Best score 0.931111180891373

- Gradient Boosting:

    Best parameters {'colsample_bytree': 0.86870234448516, 'min_child_samples': 341, 'min_child_weight': 0.01, 'num_leaves': 26, 'reg_alpha': 2, 'reg_lambda': 100, 'subsample': 0.25372980851750904}

    Best score 0.9163369549289287

Hyperparameter tuning and Cross validation - Music

- Logistic Regression:

  Best parameters {'solver': 'liblinear', 'penalty': 'l1', 'C': 1000}

  Best score 0.941158625989706

- Random Forest:

  Best parameters {'n_estimators': 2000, 'min_samples_split': 2, 'min_samples_leaf': 2, 'max_features': 'auto', 'max_depth': None, 'bootstrap': False}

  Best score 0.9881993669926081

- Gradient Boosting:

  Best parameters {'colsample_bytree': 0.5699223466222688, 'min_child_samples': 228, 'min_child_weight': 10.0, 'num_leaves': 48, 'reg_alpha': 0, 'reg_lambda': 1, 'subsample': 0.914229204723856}

  Best score 0.9750589020456453

Classification score

- After the Hyperparameter tuning and cross validation, it was calculated the score for each classifier using test data.

- Speech:

  Logistic Regression: 0.8984487847767381

  Random Forest: 0.9363813351755322

  Gradient Boosting: 0.9166404990684335

- Music:

  Logistic Regression: 0.9398143466742168

  Random Forest: 0.9913847149473173

  Gradient Boosting: 0.9753473415980747

Ensemble Methods

- Voting, Bagging, Stacking and Boosting were used.
- A bagging classifier was created based on Logistic Regression.

- Voting and Stacking classifiers were created based on Bagging classifier (Logistic Regression), Random Forest and Gradient Boosting.
- Gradient Boosting was used as a Boosting method

Ensemble Methods - Stacking

- Stacking classifier was trained using the whole speech dataset.

- estimator 0: [BC: BaggingClassifier]

```
fold  0:  [0.90066780]
fold  1:  [0.89692060]
fold  2:  [0.89782076]
fold  3:  [0.89606230]
MEAN:     [0.89786786] + [0.00173199]
```

- estimator 1: [RF: RandomForestClassifier]

```
fold  0:  [0.93202705]
fold  1:  [0.92999644]
fold  2:  [0.93162930]
fold  3:  [0.92712847]
MEAN:     [0.93019531] + [0.00192721]
```

- estimator 2: [GB: LGBMClassifier]

```
fold  0:  [0.91940380]
fold  1:  [0.91425401]
fold  2:  [0.91804308]
fold  3:  [0.91285143]
MEAN:     [0.91613808] + [0.00267608]
```

Ensemble Methods - Stacking

- Stacking classifier was trained using the whole music dataset.

- estimator 0: [BC: BaggingClassifier]

```
fold  0:  [0.94201873]
fold  1:  [0.94264566]
fold  2:  [0.94039963]
fold  3:  [0.93904461]
MEAN:     [0.94102716] + [0.00140775]
```

- estimator 1: [RF: RandomForestClassifier]

```
fold  0:  [0.98746132]
fold  1:  [0.98810848]
fold  2:  [0.98758241]
fold  3:  [0.98634874]
MEAN:     [0.98737524] + [0.00064064]
```

- estimator 2: [GB: LGBMClassifier]

```
fold  0:  [0.97554958]
fold  1:  [0.97647987]
fold  2:  [0.97524572]
fold  3:  [0.97423452]
MEAN:     [0.97537742] + [0.00080134]
```

Fitting model and classification score

- The Stacking classifier was used to convert data to meta-features format, i.e predicted class labels from Bagging, Random Forest and Gradient Boosting.
- The Voting ensemble model was trained using meta-features.

- VotingClassifier(estimators=[('BC', BaggingClassifier(base_estimator=LogisticRegression(C=1000, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100, multi_class='warn', penalty='l1', random_state=None, solver='liblinear', tol=0.0001, verbose=0,...ent=True, subsample=0.914229204723856, subsample_for_bin=200000, subsample_freq=0))], flatten_transform=None, voting='soft', weights=None)
- Score calculated on test dataset:

Score: 0.9883713875462616

Speech and Music prediction

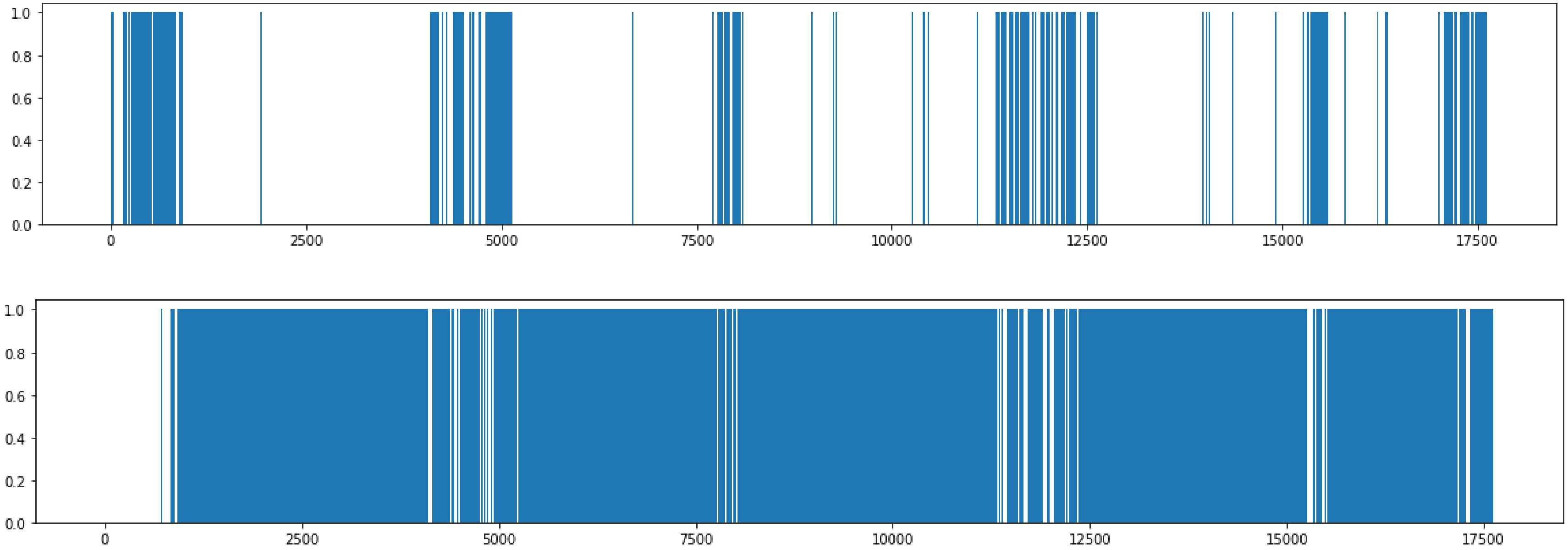


Figure 5 – 16.speech.arff
16.music.arff

Music prediction correction

- For music dataset, the method which predict probabilities was used. This approach returns the probability of the data instance belonging to each class.
- As misclassification cost for music is high, thus a threshold of 80% was chosen to correct the predictions, only probabilities higher than 0.8 were classified as music (1).

Final Gain

- 270465 rows predicted.
- 188060 were predicted as music.
- 105948 were predicted as speech.
- **129500.80** is the final value to receive, after the discount of music and speech overlap.