1. **AdaBoost in Music Emotion Recognition**

AdaBoost (Freund, Yoav; Schapire, Robert E (1997). "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences*. **55**: 119–139. [CiteSeerX](https://en.wikipedia.org/wiki/CiteSeerX_(identifier)" \o "CiteSeerX (identifier)) [10.1.1.32.8918](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.8918). [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1006/jcss.1997.1504](https://doi.org/10.1006%2Fjcss.1997.1504)) or Adaptive Boosting, is one of the popular boosting algorithms used in solving variety of machine learning problems. As the name implies, this algorithm relies on boosting methods were a sequence of weak learners collectively operate to generate an optimal prediction outcome. As a ensemble method, AdaBoost also offer flexibility to apply any classification algorithm as the underlying learning algorithm that forms backbone of the weak learners. Despite the popularity, flexibility and robustness, AdaBoost is not commonly used in Music Emotion Recognition (MER) to generate predictive emotional classifications. This chapter embellishes application of AdaBoost in MER and to use the model to determine emotion conveyed by Carnatic music compositions tuned to ragas Mayamalavagaula, Bhairavi, Kalyani, Kedaragaula, Khamboji, Mohanam, Shankarabharanam and Todi, as explained in section 4.8: Validation datasets of this document.

For the purpose to ascertain capability of AdaBoost, Decision Tree Classifier is used as the learning algorithm to evaluate performance of each classifier and to determine most suitable classifier for weak leaners.

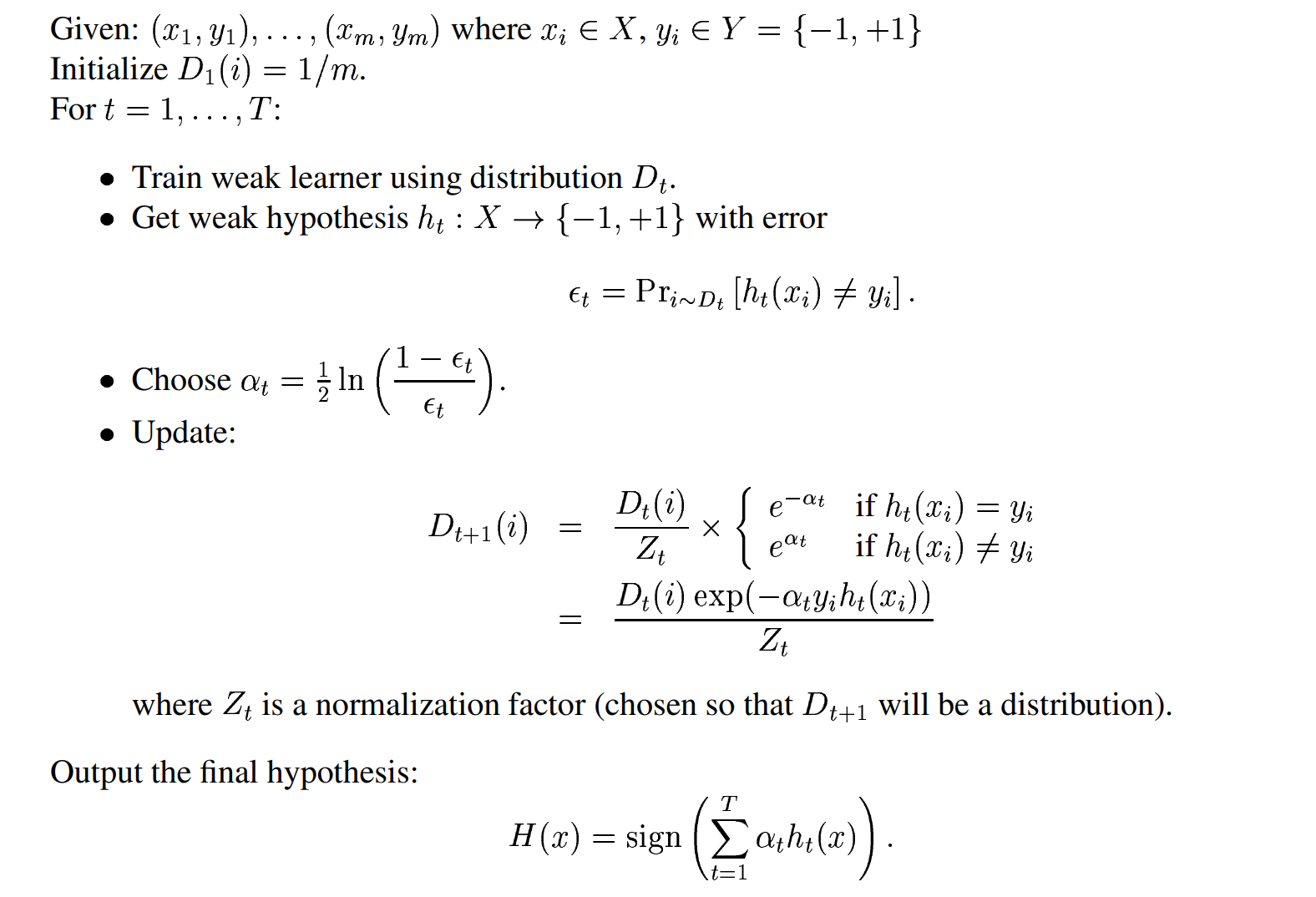
* 1. **AdaBoost Intuition**

Process of generating an outcome in AdaBoost consists of three stages viz – Scouting, Drafting and Weight assignment ( [Rojas, Raúl](https://en.wikipedia.org/wiki/Ra%C3%BAl_Rojas)*(2009).*["AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting"](http://www.inf.fu-berlin.de/inst/ag-ki/adaboost4.pdf)(Tech. Rep.)*. Freie University, Berlin.*

Scouting primarily involves testing classifiers by evaluating predictive outcome of each classifier. Correct and wrong predictions are penalized with a non-zero cost (β >0), e-β for correct predictions and eβ for wrong prediction so that wrong predictions are penalized heavily than right predictions. Once the classifiers are listed with befitting penalty, a pool of classifiers are available for AdaBoost to make a systematic prediction. AdaBoost then systematically proceeds through the pool by extracting one classifier from the pool during each iteration. The elements in the data set are weighted according to their current relevance (or urgency) at each iteration. At the beginning, all elements are assigned the same weight (just 1, or 1/N if we want to have a total sum of 1 for all weights). As the drafting progresses, the more difficult examples, that is, those where the committee still performs badly, are assigned larger and larger weights thereby reducing the overall predictive variance of the model.

Drafting involves ranking the classifiers and choosing the best classifier from the pool such that by the end of drafting process, a linear combination of classifiers is available to performing the prediction process. Once the drafting is complete each drafted classifier is assigned a weight.

The pseudocode of the AdaBoost Algorithm (Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999. (In Japanese, translation by Naoki Abe.) is represented below:

******

As per the algorithm, the training set is denoted by (xm, ym), drawn from a parent distribution where xm and ym are elements of a set containing natural numbers and with Dt (i) as the weight initialized to 1/m. During the beginning of training the weight remains the same and as the training progresses this weight will change with a higher value given to mis-classified samples.

During the initial step of the training a “weak classifier” is derived based on the distribution defined on the training set by the hardness weights, which is equal during the beginning. The classification error is then calculated based on the weak classifier model to adjust the weights assigned to tuples such that the mis-classified tuples gets higher weights. This adjustment of weights earned AdaBoost its name “Adaptive Boosting” where the weights adapts according to the training samples such that weak learners are generated each time a weight is changed. Over time, data that is consistently misclassified will receive a higher and higher weight. The algorithm is then incentivized to learn a weak classifier that is able to classify the difficult data. In this way, Adaboost makes sure that all data is covered. After T timesteps, weighted sum of all the weak classifiers are taken and is designated as final output.

**Explain about multi-class classification using AdaBoost and SAMME & SAMME.R boosting algorithms along with pseudocode.**

* 1. **Training, Test and Validation datasets**

Dataset obtained as a result of adopting influential dataset strategy, explained in chapter 4, was used to develop the MER model. Class variable in this dataset (N = 6287), *sound\_file\_class*, which is categorical variable, was converted into an ordinal variable and renamed as *sound\_file\_class\_num*. This conversion was specifically done to aid in MER modelling. Further to this, the dataset was divided into two sets viz. *model data set* and *validation data set* with partition ratio of 0.9 and stratified on *sound\_file\_class\_num* such that 90% of the observations are available for modelling along with 10% as validation set.

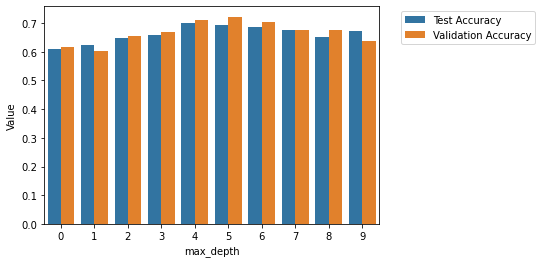
During the modelling process *model data set* is used and was split into *train* and *test* datasets with partition ratio of 0.8 and stratified on *sound\_file\_class\_num.*

* 1. **Appraising base estimator, Decision Tree Classifier**

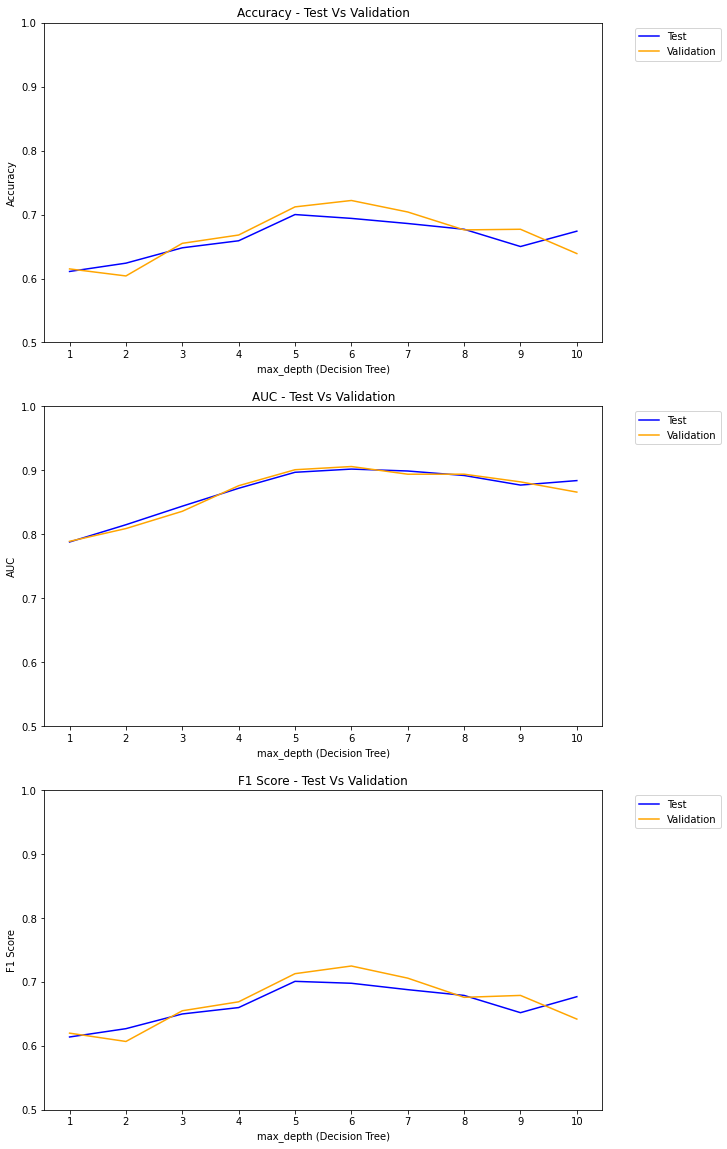
With appropriate datasets chosen and split into befitting train, test and validation datasets, the next logical step is the development of base model. Fundamental assumption taken during this step was that base estimator passed to AdaBoost would be Decision Tree Classifier. Moreover, during development of “influential dataset strategy”, it was observed that the optimal values of AdaBoost parameters – *n\_estimators* and *learning\_rate* would be 1500 and 0.1 respectively (Refer to section: tbd - Finding optimal dataset for modelling in chapter 4). The modelling process, explained here, utilizes this finding in further optimizing/fine tuning the MER model.

The assumption of using base estimator as Decision Tree Classifier invokes a critical task to be performed i.e. to determine the optimal depth of the Decision Tree so that it doesn’t overfit. Hence to determine the apt value for the parameter *max\_depth,,* a simulation of 10 trials was performed by passing various values to *max\_depth* parameter, starting from 1 to 10 during each trial, and by keeping all other parameters constant (default values as assigned by the Decision Tree Classifier of sklean library in Python). The training and test datasets were kept constant during each trial so as to avoid any sampling bias that may vary the optimal parameter determination. During each trial the model with varying max\_depth parameter for Decision Tree Classifier, was evaluated against the validation dataset – created and kept anonymous as hold out, as explained in section 5.2. The metrics considered for model evaluation and optimal parameter determination were accuracy, AUC and F1 Score (*elaborate: TBD*).

The bar char below indicates the variation of Test and Validation accuracies during each trial.



It can be observed that after trial 5 & 6 (where max\_depth = 5 and 6) the validation accuracies were dropping. Hence the max\_depth = 6 was chosen as the optimal value that would be passed to Decision Tree Classifier. The trend charts below, for accuracy, AUC and F1 Score, also supports the determination that the optimal parameter value for *max\_depth* as it can be observed visually that the validation accuracies diminish after the value of 6.

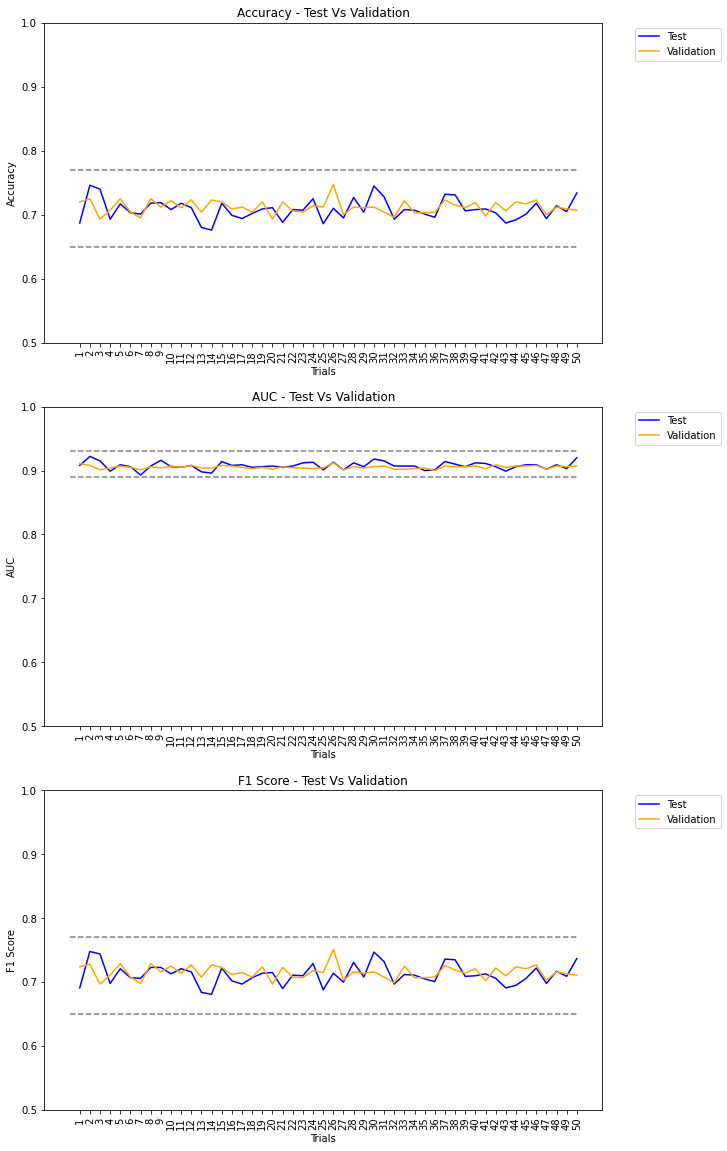


* 1. **Checking model robustness**

Having determined the optimal parameters for AdaBoost i.e. parameters that may potentially offer best model outcome viz. *base estimator* = Decision Tree Classifier with max\_depth = 6, *n\_estimators* = 1500 and *learning\_rate* = 0.1, its is pertinent to evaluate robustness of a model with these parameters by passing various samples of data to the model (contrary to earlier stage of determining optimal *max\_depth* parameter) and evaluating each model against the validation set generated earlier as explained in section 5.2. The evaluation, involving simulation containing 50 trials, was designed to develop separate model in each trial by performing a stratified random sampling on the model dataset to draw training and test data set. Each model was then evaluated against the validation data set to determine accuracy, AUC and F1 Score metrics as indicated in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Trial #** | **Accuracy** | | **AUC** | | **F1 Score** | |
| **Test** | **Validation** | **Test** | **Validation** | **Test** | **Validation** |
| 1 | 0.687 | 0.720 | 0.908 | 0.911 | 0.691 | 0.724 |
| 2 | 0.746 | 0.725 | 0.922 | 0.908 | 0.748 | 0.728 |
| 3 | 0.740 | 0.693 | 0.915 | 0.901 | 0.744 | 0.697 |
| 4 | 0.693 | 0.707 | 0.899 | 0.904 | 0.698 | 0.711 |
| 5 | 0.717 | 0.725 | 0.909 | 0.907 | 0.721 | 0.729 |
| 6 | 0.703 | 0.704 | 0.906 | 0.905 | 0.707 | 0.708 |
| 7 | 0.701 | 0.695 | 0.893 | 0.901 | 0.706 | 0.698 |
| 8 | 0.718 | 0.725 | 0.907 | 0.906 | 0.723 | 0.729 |
| 9 | 0.719 | 0.712 | 0.916 | 0.904 | 0.723 | 0.716 |
| 10 | 0.708 | 0.722 | 0.906 | 0.907 | 0.713 | 0.725 |
| 11 | 0.718 | 0.711 | 0.905 | 0.905 | 0.721 | 0.714 |
| 12 | 0.711 | 0.723 | 0.908 | 0.908 | 0.716 | 0.727 |
| 13 | 0.680 | 0.704 | 0.898 | 0.904 | 0.684 | 0.708 |
| 14 | 0.676 | 0.723 | 0.896 | 0.904 | 0.681 | 0.727 |
| 15 | 0.718 | 0.720 | 0.914 | 0.908 | 0.722 | 0.723 |
| 16 | 0.699 | 0.709 | 0.908 | 0.907 | 0.702 | 0.712 |
| 17 | 0.694 | 0.712 | 0.909 | 0.905 | 0.697 | 0.715 |
| 18 | 0.702 | 0.704 | 0.905 | 0.903 | 0.707 | 0.708 |
| 19 | 0.709 | 0.720 | 0.906 | 0.905 | 0.714 | 0.724 |
| 20 | 0.711 | 0.693 | 0.907 | 0.902 | 0.715 | 0.697 |
| 21 | 0.688 | 0.720 | 0.905 | 0.906 | 0.690 | 0.723 |
| 22 | 0.708 | 0.706 | 0.907 | 0.904 | 0.711 | 0.708 |
| 23 | 0.707 | 0.704 | 0.912 | 0.904 | 0.710 | 0.707 |
| 24 | 0.725 | 0.714 | 0.913 | 0.903 | 0.729 | 0.718 |
| 25 | 0.686 | 0.712 | 0.901 | 0.904 | 0.688 | 0.715 |
| 26 | 0.710 | 0.747 | 0.913 | 0.912 | 0.714 | 0.751 |
| 27 | 0.695 | 0.700 | 0.901 | 0.901 | 0.700 | 0.703 |
| 28 | 0.727 | 0.712 | 0.912 | 0.906 | 0.731 | 0.716 |
| 29 | 0.704 | 0.711 | 0.906 | 0.904 | 0.708 | 0.714 |
| 30 | 0.745 | 0.712 | 0.918 | 0.906 | 0.747 | 0.716 |
| 31 | 0.728 | 0.704 | 0.915 | 0.907 | 0.732 | 0.708 |
| 32 | 0.693 | 0.696 | 0.907 | 0.902 | 0.697 | 0.699 |
| 33 | 0.708 | 0.722 | 0.907 | 0.902 | 0.712 | 0.725 |
| 34 | 0.707 | 0.703 | 0.907 | 0.903 | 0.711 | 0.707 |
| 35 | 0.701 | 0.703 | 0.900 | 0.904 | 0.705 | 0.707 |
| 36 | 0.696 | 0.704 | 0.901 | 0.900 | 0.701 | 0.708 |
| 37 | 0.732 | 0.723 | 0.914 | 0.907 | 0.736 | 0.726 |
| 38 | 0.731 | 0.715 | 0.910 | 0.906 | 0.735 | 0.719 |
| 39 | 0.706 | 0.711 | 0.906 | 0.906 | 0.709 | 0.714 |
| 40 | 0.708 | 0.719 | 0.912 | 0.907 | 0.710 | 0.721 |
| 41 | 0.709 | 0.698 | 0.911 | 0.903 | 0.713 | 0.702 |
| 42 | 0.703 | 0.719 | 0.906 | 0.909 | 0.706 | 0.722 |
| 43 | 0.687 | 0.706 | 0.899 | 0.905 | 0.691 | 0.710 |
| 44 | 0.692 | 0.720 | 0.906 | 0.907 | 0.695 | 0.724 |
| 45 | 0.701 | 0.717 | 0.909 | 0.907 | 0.706 | 0.721 |
| 46 | 0.718 | 0.723 | 0.909 | 0.908 | 0.722 | 0.727 |
| 47 | 0.694 | 0.700 | 0.902 | 0.902 | 0.698 | 0.703 |
| 48 | 0.714 | 0.712 | 0.909 | 0.907 | 0.717 | 0.716 |
| 49 | 0.705 | 0.709 | 0.903 | 0.906 | 0.709 | 0.713 |
| 50 | 0.734 | 0.707 | 0.920 | 0.907 | 0.737 | 0.711 |

The trends of metrics considered is explained in the plots below.



The summary statistics of the metrics is show in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Summary  Statistic** | **Accuracy** | | **AUC** | | **F1 Score** | |
| **Test** | **Validation** | **Test** | **Validation** | **Test** | **Validation** |
| Mean (µ) | 0.708 | 0.712 | 0.908 | 0.905 | 0.712 | 0.715 |
| Std. Dev (σ) | 0.016 | 0.010 | 0.006 | 0.003 | 0.016 | 0.010 |
| CI – UB (µ+1.96 \* σ) | 0.740 | 0.732 | 0.919 | 0.910 | 0.743 | 0.736 |
| CI – LB (µ- 1.96 \* σ) | 0.677 | 0.692 | 0.896 | 0.900 | 0.681 | 0.695 |

* 1. **Choosing the boosting algorithm for AdaBoost**

**S**tagewise **A**dditive **M**odeling using a **M**ulti-class **E**xponential loss function (SAMME) (Zhu, H. Zou, S. Rosset, T. Hastie, “Multi-class AdaBoost”, 2009) and **S**tagewise **A**dditive **M**odeling using a **M**ulti-class **E**xponential loss function for **R**eal (SAMME.R) (Zhu, H. Zou, S. Rosset, T. Hastie, “Multi-class AdaBoost”, 2009) are commonly used multi-class classification algorithm used in boosting methods. AdaBoost can also be enhanced to perform multi-class classification using these algorithms.

In an effort to choose the best boosting algorithm to be selected two AdaBoost models were developed with the *algorithm* parameter as SAMME and SAMME.R respectively using a single trial. Outcome of both models is illustrated in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Algorithm** | **Metric** | **Emotion Class** | | | |
| **0 (Q1)** | **1 (Q2)** | **2 (Q3)** | **3 (Q4)** |
| Test | SAMME.R | Precision | 0.58 | 0.95 | 0.66 | 0.70 |
| Recall | 0.78 | 0.69 | 0.64 | 0.66 |
| F1 Score | 0.66 | 0.80 | 0.65 | 0.68 |
| Accuracy | 0.693 | | | |
| SAMME | Precision | 0.70 | 0.95 | 0.72 | 0.78 |
| Recall | 0.80 | 0.81 | 0.73 | 0.77 |
| F1 Score | 0.75 | 0.88 | 0.72 | 0.77 |
| Accuracy | 0.778 | | | |
| Validation | SAMME.R | Precision | 0.60 | 0.94 | 0.70 | 0.74 |
| Recall | 0.81 | 0.73 | 0.67 | 0.68 |
| F1 Score | 0.69 | 0.83 | 0.69 | 0.70 |
| Accuracy | 0.722 | | | |
| SAMME | Precision | 0.66 | 0.92 | 0.71 | 0.77 |
| Recall | 0.79 | 0.82 | 0.70 | 0.71 |
| F1 Score | 0.72 | 0.87 | 0.70 | 0.74 |
| Accuracy | 0.754 | | | |

* 1. **Modelling with the original emotion categorization (4 classes)**

With the analysis conducted so far, it was observed that the following combination of parameters and bossting algorithm would be most optimal while explaining MER using AdaBoost. Those are:

* n\_estimators = 1500
* learning\_rate = 0.1
* base\_estimator = Decision Tree Classifier
* boosting algorithm = SAMME

So as to develop a final model using these parameters a s step approach was taken which included:

1. Building the initial model with all features to visualize the feature importance
2. Developing an abstract model to chose top *N* features - choosing the best model that offers optimal accuracy from *n* simulations
3. Building the final model and making predictions
   * 1. **Building the initial model**

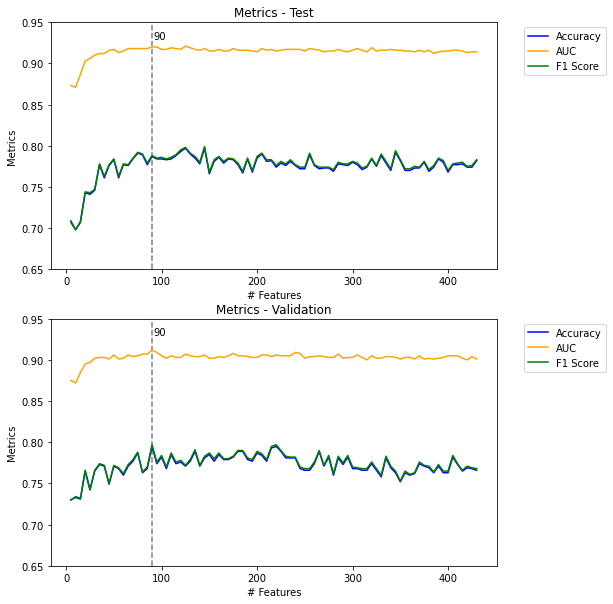
As adorned in *chapter 4:tbd*, the class variable, *sound\_file\_class\_num* which is an ordinal transformation of *sound\_file\_class* that contains the original emotion categorization viz Q1, Q2, Q3 and Q4 (refer to chapter 4: Data Processing), is used as the dependent variable to build the model. The classification report of the model outcome is depicted in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AdaBoostClassifier(algorithm='SAMME',  base\_estimator=DecisionTreeClassifier(max\_depth=6,  random\_state=1),  learning\_rate=0.1, n\_estimators=1500, random\_state=1) | | | | | |
| **Dataset** | **Metric** | **Emotion Class** | | | |
| **0 (Q1)** | **1 (Q2)** | **2 (Q3)** | **3 (Q4)** |
| Test | Precision | 0.72 | 0.94 | 0.74 | 0.77 |
| Recall | 0.83 | 0.86 | 0.73 | 0.71 |
| F1 Score | 0.77 | 0.90 | 0.73 | 0.74 |
| Accuracy | 0.78 | | | |
| AUC | 0.911 | | | |
| Validation | Precision | 0.67 | 0.94 | 0.73 | 0.79 |
| Recall | 0.80 | 0.83 | 0.71 | 0.75 |
| F1 Score | 0.73 | 0.88 | 0.72 | 0.77 |
| Accuracy | 0.778 | | | |
| AUC | 0.898 | | | |

It can be observed that the accuracy of the model – both test and validation are 0.78 & 0.778 respectively which is consistent with the accuracy measures observed during the selection of applicable boosting algorithm. In addition, these measures are significantly higher than the accuracy reported earlier by research on the same dataset (pandas et.al tbd) which was in the range of 0.67(+/- 5%) using SVM classifier as baseline model with 70 features. It was also reported that this accuracy was further improved to 0.764 (+/- 4%) using an SVM classifier that included baseline and novel features.

* + 1. **Choosing relevant features**

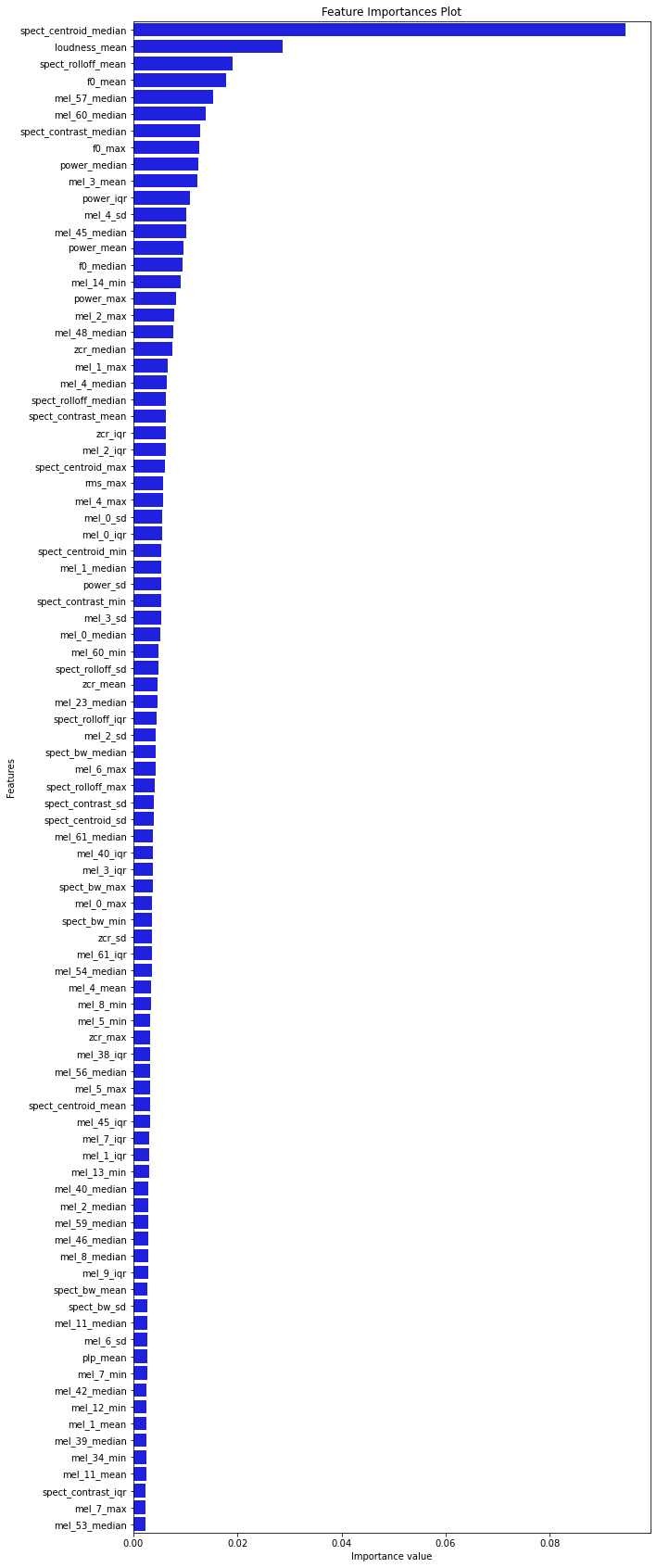
Having built the model with all features, the next step was to identify the optimal features needed so as to have an abstract model. In order to do that the feature importance’s, of the model with all features, were sorted and extracted in chunks of 5 features at a time to build models using those chunks of features in a recursive and cumulative fashion. Again, the same metrices – accuracy, AUC and F1S, were captured during each iteration to determine the optimal number of features by comparing these metrices and determining optimal cut off or number of features needed. The point at which the metrices hit a plateau was considered as the cut-off point. The train and test datasets passed to this model was kept constant to avoid sampling bias and the model was compared against validation dataset, kept as hold out, to evaluate validation metrices. The plot below explains the comparison of test and validation metrices of model trained.



It can be observed that all 3 metrices stabilizes beyond a certain set of features, somewhere less than 100, which acts as cut-off point. The maximum value among fist 100 observations of validation AUC is determined as the cut-off point as highlighted by the vertical line in the plot, which turned out to be 90. Hence the top 90 features within the features importance’s retrieved earlier (in 5.6.1) was considered as the optimal feature set for the AdaBoost model. The test and validation metrics observed at this point is shown in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **# Features** | **Accuracy** | | **AUC** | | **F1 Score** | |
| **Test** | **Validation** | **Test** | **Validation** | **Test** | **Validation** |
| 90 | 0.787 | 0.795 | 0.920 | 0.912 | 0.788 | 0.797 |

The barplot illustrating the feature importance of top 90 features is show below.



* + 1. **Developing final model**

Finally, based on the analysis performed so far, the final model is developed and the table below illustrates the classification report on test and validation datasets of final model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AdaBoostClassifier(algorithm='SAMME',  base\_estimator=DecisionTreeClassifier(max\_depth=6,  random\_state=1),  learning\_rate=0.1, n\_estimators=1500, random\_state=1) | | | | | |
| **Dataset** | **Metric** | **Emotion Class** | | | |
| **0 (Q1)** | **1 (Q2)** | **2 (Q3)** | **3 (Q4)** |
| Test | Precision | 0.74 | 0.95 | 0.74 | 0.78 |
| Recall | 0.80 | 0.85 | 0.74 | 0.80 |
| F1 Score | 0.77 | 0.9 | 0.74 | 0.79 |
| Accuracy | 0.798 | | | |
| AUC | 0.913 | | | |
| Validation | Precision | 0.72 | 0.94 | 0.7 | 0.73 |
| Recall | 0.82 | 0.84 | 0.73 | 0.68 |
| F1 Score | 0.77 | 0.89 | 0.71 | 0.71 |
| Accuracy | 0.768 | | | |
| AUC | 0.901 | | | |

The confusion matrix obtained through prediction on validation dataset is illustrated below.

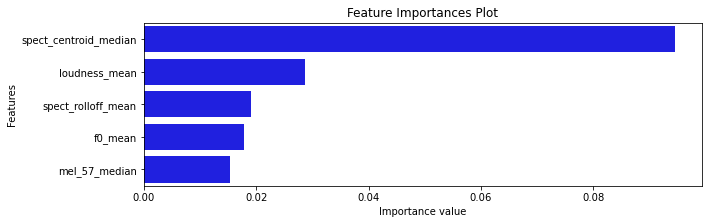
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Q1** | **Q2** | **Q3** | **Q4** |
| Q1 | 129 | 6 | 13 | 9 |
| Q2 | 18 | 133 | 3 | 4 |
| Q3 | 14 | 3 | 114 | 26 |
| Q4 | 17 | 0 | 33 | 107 |

The table below also illustrates various models developed during the analysis and its comparison using validation accuracies & AUC.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Model** | **# of features** | **Accuracy** | **AUC** |
| **1** | **Final Model** | **90** | **0.798** | **0.913** |
| 2 | Baseline Model | 431 | 0.778 | 0.898 |
| 3 | Initial model (SAMME.R) | 431 | 0.772 | 0.905 |
| 4 | Initial model (SAMME) | 431 | 0.773 | 0.898 |

* 1. **Inferences**

To draw relevant inferences and to deduce an abstract framework of musical features that induces emotion, top 5 features were evaluated. The bar plot below illustrates the significance of these features.



The selection of these features by model can be inferred from a musical standpoint.

1. **Spectral Centroid**

For any signal, the spectral centroid refers to the center mass of a frequency spectrum within which the signal belongs to. From a musical perspective, this center mass refers to the “brightness of the sound” or the tonal quality of sound which in turn is represented by the timbre that broadly refers to the number of harmonic elements a single frequency involved in the entire frequency spectrum of the music. This feature of music distinguishes musical sound – musical instrument vs voice, percussion vs string instruments, violin vs guitar etc. and hence AdaBoost model identifies this feature as the most important feature in MER. Human brain also relies on this feature to make the distinction while listening to music. As a result the variations in this component can influence the way a musical element is cherished by humans. As the model suggests, spectral centroid seems to be the critical element involved in inducing musical emotion with lower values indicating Valence on AV scale - emotions like Sad, Calm etc., and higher values indicating an Arousal – emotions like Happy, Anger, Fear, Disgust etc.)

1. **Loudness:**

Loudness refers to the intensity of sound in music which in turn is highly correlated with power of the sound. Based on the model outcome, the loudness seems to be directly proportional to the AV scale where in lower value of loudness indicating valence and higher values indicating Arousal. On evaluating the model outcome, the range of loudness associated with each quadrant may not be significantly different but seems to induce interaction effect with respect to the spectral centroid. As a result, loudness is highlighted as the second most important feature by AdaBoost model. From a musical stand point too, loudness is one of the critical elements in compositions as it can not only distinguish the genre (rock vs country) of the music but also can influence the melody involved.

1. **Spectral Rolloff:**

Spectral Rolloff refers to the frequency within the spectrogram where 85% of the power exists below that frequency. The model outcome (as illustrate in table below) indicates that this roll off point is higher for emotions attributing to Arousal and lower for emotions attributing to Valence. The mean values of this feature vary significantly, based on the model outcome, and is an import feature that distinguishes the emotion quadrant.

1. **Fundamental Frequency (F0):**

Fundamental frequency refers to the approximate frequency of the signal which in turn defines the number of oscillations per second expressed in Hertz. From a musical standpoint, F0 can be mapped to the musical key used in the composition. The model outcome illustrates that the variability in F0 has a significant effect MER and suggests F0 values trends lower for emotions associated with Valence and vice versa for Arousal on an AV scale.

1. **Mel Frequency band 57:**

The model indicates that the 57th component of 64 mel frequencies considered has a significant effect on distinguish musically induced emotion. On converting the values appearing in this feature to Hertz, it is observed that this band corresponds to extremely low frequencies in the frequency spectrum. The potential source of such frequencies might be bass instruments or low pitches instruments/percussions used in the musical compositions. As the dataset consists of monophonic and polyphonic sounds of various genres, this might be an important element that distinguishes emotional quadrants. In addition, the variability observed is not proportional to the AV scale (high value for emotions mapped to Q2 quadrant vs other quadrants), in contrast to other features explained above. In addition, the model outcome, as highlighted in table below, illustrates significant difference in mean values across all 4 quadrants, suggesting fact that the presence of base frequencies or low frequencies in the musical signal can influence musically induced emotion.

The table below illustrates the summary of potential range of values within each of the 5 features, considered and explained above, mapped to each emotion quadrant which in turn offers an indication into the composition of what an abstract emotion framework may look like. The values indicated are mean and the range (within braces) from the validation dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Emotion Class** | **Q1** | **Q2** | **Q3** | **Q4** |
| **Description** | **Happy, Surprise** | **Disgust, Anger, Fear** | **Sad** | **Calm** |
| **Spectral Centroid Range** | 1082.34 (344.8 to 1802.19 ) | 1211.09 (469.91 to 1885.44 ) | 765.54 (289.43 to 1553.36 ) | 735.56 (142.58 to 1563.31 ) |
| **Loudness Range** | -15.81 dB (-26.32 dB to -9.26 dB ) | -12.92 dB (-23.25 dB to -7.56 dB ) | -17.71 dB (-30.09 dB to -9.59 dB ) | -17.77 dB (-28.8 dB to -10.75 dB ) |
| **Spectral Rolloff** | 2329.39 (559.08 to 4205.72 ) | 2530.04 (929.92 to 3675.12 ) | 1662.56 (427.32 to 3611.56 ) | 1643.44 (169.49 to 3865.96 ) |
| **Fundamental Frequency Range** | 286.0 Hz (164.82 Hz to 566.52 Hz ) | 459.16 Hz (148.6 Hz to 963.27 Hz ) | 226.98 Hz (132.03 Hz to 495.34 Hz ) | 226.59 Hz (131.73 Hz to 466.46 Hz ) |
| **Mel Frequency band 57 Range** | 28.0 Hz (1.33 Hz to 168.0 Hz ) | 145.33 Hz (1.33 Hz to 638.0 Hz ) | 8.67 Hz (0.0 Hz to 48.67 Hz ) | 4.67 Hz (0.0 Hz to 32.67 Hz ) |

* 1. **Predictions**
     1. **Prediction based on validation dataset**

The predictions were performed using the validation data set and emotion plot, as demonstrated in chapter 4 were developed, to determine the emotion classification of the sound considered. For validation, 4 sound samples, belonging to quadrants Q1, Q2, Q3 and Q4 respectively were considered. Each of those samples were divided into 5 second window with 1 second overlap before extracting 90 musical features, as illustrated in section 5.6.2 above. The dataset was fed to the model such that the model generated emotion plots – trend and summary, for each of the sound sample. The output is illustrated below.

|  |
| --- |
| **Sound file name**: MT0001335920.mp3  **Actual classification**: Q1/Happy  **Model classification**: Q1/Happy |
|  |
| **Sound file name**: MT0001613887.mp3  **Actual classification**: Q2/ Anger, Disgust, Fear  **Model classification**: Q2/ Anger, Disgust, Fear |
|  |
| **Sound file name**: MT0000088320.mp3  **Actual classification**: Q3/ Sad  **Model classification**: Q3/ Sad |
|  |
| **Sound file name**: MT0000092267.mp3  **Actual classification**: Q4/ Calm  **Model classification**: Q4/ Calm |
|  |

* + 1. **Prediction using songs tuned to ragas Mayamalavagowla and Bhairavi**

Subsequent to the aforementioned validation, relevant features were extracted from sounds tuned to raga Mayamalavagowla and Bhairavi were fed to the model as input. The outcome is showed below.

|  |
| --- |
| **Raga:** Mayamalavagowla  **Sound file name:** 01-cintayEham\_jAnakIkAntam-mAyAmALavagauLa.mp3  **Model classification:** **Q1/Happy** |
|  |
| **Raga:** Mayamalavagowla  **Sound file name:** 01-dEvadEva\_kalayAmi-mAyAmALavagauLa-swAtitirunAL.mp3  **Model classification:** **Q1/Happy** |
|  |
| **Raga:** Mayamalavagowla  **Sound file name:** 01-dasharatha\_nandana\_disha-mAyAmALavagauLa.mp3  **Model classification:** **Q1/Happy** |
|  |
| **Raga:** Mayamalavagowla  **Sound file name:** 01-shrI\_nathAdi\_guruguhO-mAyAmALavagauLa.mp3  **Model classification:** **Q1/Sad** |
|  |
| **Raga** : Mayamalavagowla  **Sound file name:** 01-viribOni-varanam-bairavi-pachimiriyam\_Adiyappa.mp3  **Model classification:** **Q1/Happy** |
|  |
| **Raga:** Mayamalavagowla  **Sound file name:** 01-sarasijanAbhamurArE-mAyAmALavagauLa.mp3  **Model classification:** **Q1/Happy** |
|  |
| **Raga:** Bhairavi  **Sound file name:** 02-mahA\_tripurasundari-bhairavi.mp3  **Model classification:** **Q1/Happy** |
|  |
| **Raga:** Bhairavi  **Sound file name:** 02-nI\_pAdamulE\_gatiyani-bhairavi.MP3  **Model classification:** **Q3/Sad** |
|  |
| **Raga:** Bhairavi  **Sound file name:** 02-Amba-kamakshi-Bairavi-M.-Chapu-Syama-Sastri.mp3  **Model classification:** **Q4/Calm** |
|  |
| **Raga:** Bhairavi  **Sound file name:** 01-viribONi\_ninnE-VARNAM-bhairavi.mp3  **Model classification:** **Q3/Sad** |
|  |
|  |
|  |

* 1. ***Mayamalavagowla* – Does it induce “*Happy*” emotion musically ?**

***TBD***