Big Data Analysis and Data Science Master
Machine Learning Project
Speaker Accent Detection using SVMs

Roberto Gallea

October 27, 2020

Abstract

This report summarizes a study aimed to the application and evaluation of multi-class Support Vector Machine algorithm to a Speaker Accent Recognition Data set. Speakers' mother-language detection is performed from feature based on Mel-frequencies Cepstrum of 1 second of reading words from audio samples.
General model applicability is firstly evaluated on a mathematical model based on the pure definition of dichotomic SVM optimization algorithm for classification. Then, more in-depth implementation and testing is conducted for 6-classes categorization, using single SVM and Bagging SVM approaches.

1 Introduction and Overview

Support Vector Machines (SVMs) are a well-established method used for both binary and multi-class classification, firstly described and introduced in their formal definition in [3], and further developed on both mathematical and statistical basis (for example, see [6] and [17]). It was also proved that they are equivalent to Regularization Neural Networks [2].
In the last decades they were succesfully applied to audio signals problems, for example speech recognition and segmentation [15], [10], multimedia content-retrieval [19] and sounds classification [1], [7].
In this report, it is shown how it can be applied to speaker accent detection. The problem is to detect speaker mother language from some audio samples. The same approach was originally took in [13]. Here the concept is further investigated with an in-depth dissertation and testing.

1.1 Dataset description

A total of 329 signal data were collected from the voice of 22 speakers, 11 female and 11 male, from an internet source. Because of the method used for collecting the data, there is no background noise in any of the sound tracks. 15 words were assigned to each voice. Notwithstanding the sound tracks have lengths of only around 1 second, with a sampling rate of 44,100 Hz, each sound track vector on the time domain has more than 30,000 entries. For each set of samples the ground truth of the speaker langauge is known.
In order to drastically reduce the huge dimensionality of the data, MFCC was used as feature extraction method [9]. The main idea of MFCC is to transform the signal from time domain to frequency domain and then map the transformed signal in hertz onto Mel-scale [18] which takes into account human hearing frequency range. The calculation of MFCCs includes the following steps:

- Pre-emphasis filtering;
- Pre-Take the absolute value of the short time Fourier transformation using windowing;
- Warp to auditory frequency scale (Mel-scale);
- Take the discrete cosine transformation of the log-auditory-spectrum;

- Return the first q MFCCs.

In this context, $q = 12$. Roughly speaking, MFCCs are values that summarize the frequency content of the audio sample in the human hearing frequency range (20hz-20khz).

2 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are a consolidated method aimed to both binary and multi-class data classification. The original idea came within the area of statistical learning theory and structural risk minimization. However SVMs have demonstrated to work successfully on various classification and forecasting problems. They were also been used in many pattern recognition and regression estimation problems and have been exploited for addressing problems of dependency estimation, forecasting and constructing intelligent machines.

An SVM is defined as the maximum margin hyperplane that lies in the data features space such that classifies data by separating corresponding vectors using linear or non-linear optimal boundaries. After construction of the hyperplanes, the SVM discovers the boundaries between the input classes and the input elements defining the boundaries (i.e. the support vectors). Such hyperplane parameters are chosen so that the distance between the resulting hyperplane and the margin defined by support vectors is maximized. If data is not separable and such hyperplane does not exist, a minimum error hyperplane is found.

Such formulation leads to a quadratic programming problem with linear constraints, which, in its dual form is represented as follows:

$$maximize \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \phi(x_i) \phi(x_j) \tag{1}$$

$$w.r.t. \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{2}$$

Where the dot product $\phi(x_i)\phi(x_j) = K(x_i, x_j)$ in (1) is called kernel matrix or Gram matrix, and represents a vector distance matrix of the mapping of feature data in a different feature space (with arbitrary number of dimensions). Any function could be used for this purpose with the only constraints, as a result of Mercer's theorem [14], is that kernel matrix must be *symmetric* and *positive semi-definite*.

Popular kernels are:

- **Linear kernel**: $K(x_i, x_j) = x_i \cdot x_j$
- **Polynomial kernel**: $K(x_i, x_j) = (x_i \cdot x_j + 1)^p$
- **Gaussian (radial-basis function) kernel**: $K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}$
- **Sigmoid kernel**: $K(x_i, x_j) = \tanh(\eta x_i \cdot x_j + \upsilon)$

3 Algorithms Implementation

SVMs are intrinsically binary classifiers, though there are several extensions to multi-class classification, such as *one-vs-one* [11] and *one-vs-all* [8]. However, before applying multi-class SVMs directly, the conceptual applicability and feasibility of the problem was tested with a binary classifier based on its exact formal definition. Once model suitability was proven, state-of-art algorithms were applied to the complete problem. The available dataset was splitted in two subsets, a training group of 180 samples, and a test group of 129 samples, distributed as shown in Table 1.

	Language					
	US	**UK**	**IT**	**FR**	**ES**	**GE**
Training set	99	21	17	14	14	15
Test set	66	24	13	16	15	15
Total	165	45	30	30	29	30

Table 1: Dataset summary and division in training and test set.

3.1 Mathematical problem solution

The feasibility of SVMs applicability was tested by solving SVM problem on a two-classes classification. SVM model was applied to discriminate Italian language speech from non-Italian language speech. The model was defined and solved using GAMS non-linear solver. The model, after receiving feature vectors and precomputed distance matrix from an external file, determines actual K matrix using rbf, and find optimal hyperplane parameters by solving the quadratic model. The value for σ was empirically tuned and set to 10 (which is equivalent to set $\gamma = 0.005$), which provides a reduced set of support vector. Code for this optimization is shown in supplementary material at https://github.com/unipa-bigdata/gallea-svm-accent-detection

3.2 State-of-art problem solution

After validating formal applicability, state of art algorithm implementations were applied to the actual binary and multi-class problems. For this purpose *scikit-learn* python library [16] was used.

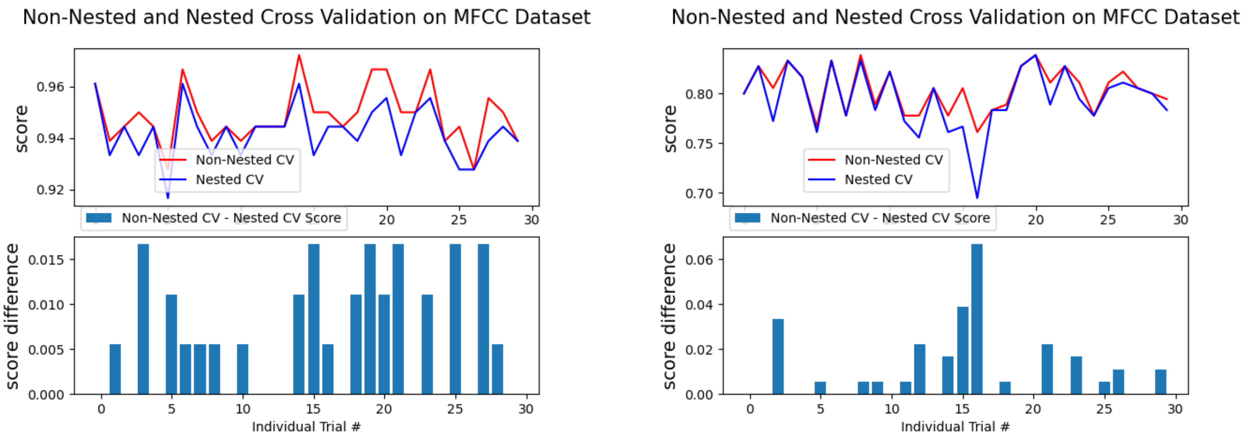
Prior model general performance was evaluated using cross-validation. However, flat cross-validation can introduce an optimistic bias into the performance estimate as it uses the same data for both model evaluation and hyperparameters tuning [5]. For this reason nested cross-validation approach was applied. In this scheme, an outer cross-validation procedure is performed to provide a performance estimate used to select the optimal model. In each fold of the outer cross-validation, the hyperparameters of the model are tuned independently to minimise an inner cross-validation estimate of generalisation performance. This mitigates the bias introduced by the flat cross-validation procedure as the test data in each iteration of the outer cross-validation has not been used to optimise the performance of the model in any way, and may therefore provide a more reliable criterion for choosing the best model. The computational expense of nested cross-validation, however, is substantially higher.
The procedure was applied using 4 outer cross-validation folds and 4 inner-validation folds. Thus, each inner group model was trained on $180/4/4 \approx 22$ samples.

The procedure was applied to both two-classes (IT, non-IT language) and multi-classes classification and mean estimated model performance was evaluated for 30 trials. Figures 1a-b show quantitative measure of the bias reduction provided by nested-cv, for binary and multi-class cases respectively.

From this analysis also emerged that *rbf* kernel has a sligtly best performances over other kernel types.

Actual model performance evaluation on the test dataset was performed in a similar way. The model was trained on the full train dataset and a grid search approach was applied to find the best hyperparameters for rbf function in terms of *gamma* and *c* values. Also for this cases the results for both binary and multi-class approach were assessed. Tables 2-5 show the respective results using not normalized (first row) and normalized (second row) features data. Figures 2a-d depict the resulting confusion matrices. In addition, ROC and precision-recall curves are shown for binary case, Figures 3a-d.

The last improvement attempted is related to the usage of the Bagging approach [4]. Bagging is a form of ensemble classification method, where several instances of a black-box estimator are built on random subsets of the original training set and then aggregate their individual predictions to form a final prediction. Such procedure provides monotonically increasing score improvement as the number of estimators increases. In this context were used 30 estimators. Results are shown in Tables 6 and 7. Figures 2e-f depict the resulting confusion matrices. In addition, ROC and precision-recall curves are shown for binary case Figures 3e-f.



(a) Binary classification.

(b) Multi-class classification.

Figure 1: Bias reduction using nested cross-validation for binary classification (a) and multi-class classification (b). Top plot shows nested-cv score (in blue) and flat-cv score (in red), for each trial. Bottom plot shows differences between scores, which is a quantitative measure of removed bias.

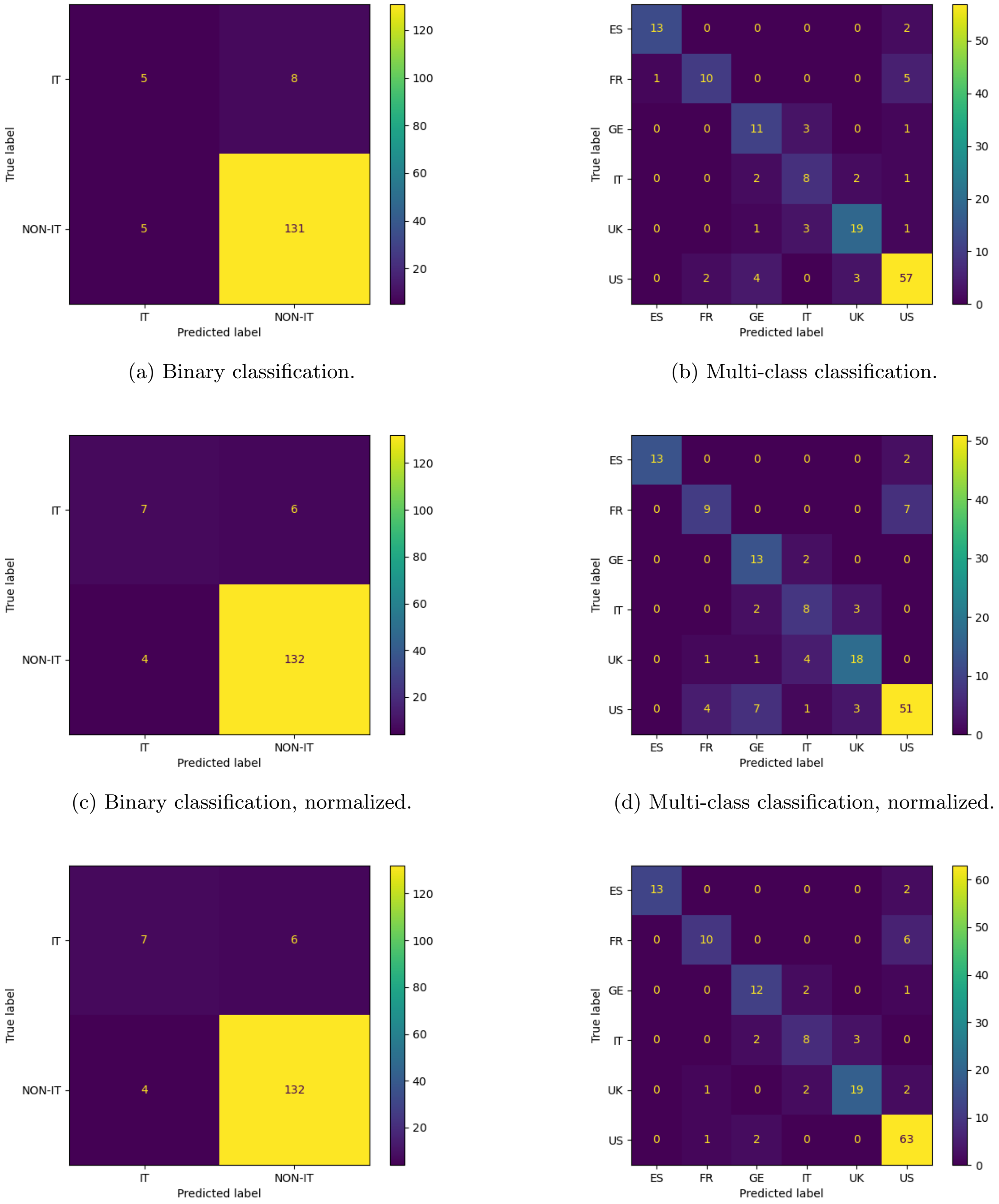
The results point out that Italian and German language are the hardest to detect, while other languages (Spanish, French and both American and UK English) have much higher rate, in particular, Spanish language has a very high performance. This could be partially due to an unbalance of samples ($\sim 10\%$ of total). However, French and Spanish languages have the same number of samples and an higher performance.
Using normalized features provides better results than raw features in the binary case, while in multi-class case some scores are lower. However, using bagging this scores increase again. This could be a clue that better performance without normalization may be related to chance due to the low number of samples.
The overall performance is quite high, since accuracy is $\sim 93\%$ for binary classification and $\sim 84\%$ for multi-class classification. Lowest performance is related to recall but this could not be a problem, since, in a real case scenario, many 1 second audio clips can be recorded during a normal speech and many classifications could be attempted to detect the correct speaker's accent.

3.3 Model explanation

After evaluation model performance, an additional analysis was made to address model explanation. This is meant as a way of understand what features influences the most the predictions operated by the model. For this purpose the SHAP (SHapley Additive exPlanations) framework was used [12]. SHAP assigns each feature an importance value for a particular prediction. The results of this analysis for both not-normalized and normalized feature vectors are show in Figures 4 and 5 respectively. Additionally, Figure 6 show a plot that summarize the previous ones. Recalling that features Xi represent are related to frequencies in increasing order, These results point out that the most emphasized frequency by the model are the mid-low ones for GE, ES, and US language, while mid-frequencies are most important for UK and FR language. IT language gives similar importance to a broader span of frequencies (Class 5 in Figure 5), and this could explain why this is the hardest language to detect.

4 Summary and Conclusions

An SVM-based approach was proposed for speaker's accent detection. The analysis was conducted on a dataset of 329 audio samples. Features used for training are extracted from MFCC analysis on the original audio samples. After an SVM definition test, several models of increasing complexity have been developed and described. The overall results are positive and promising. Further development of the method could be to use voting classifiers leveraging other classification methods, such as clustering algorithms, decision trees, etc.



(a) Binary classification.

(b) Multi-class classification.

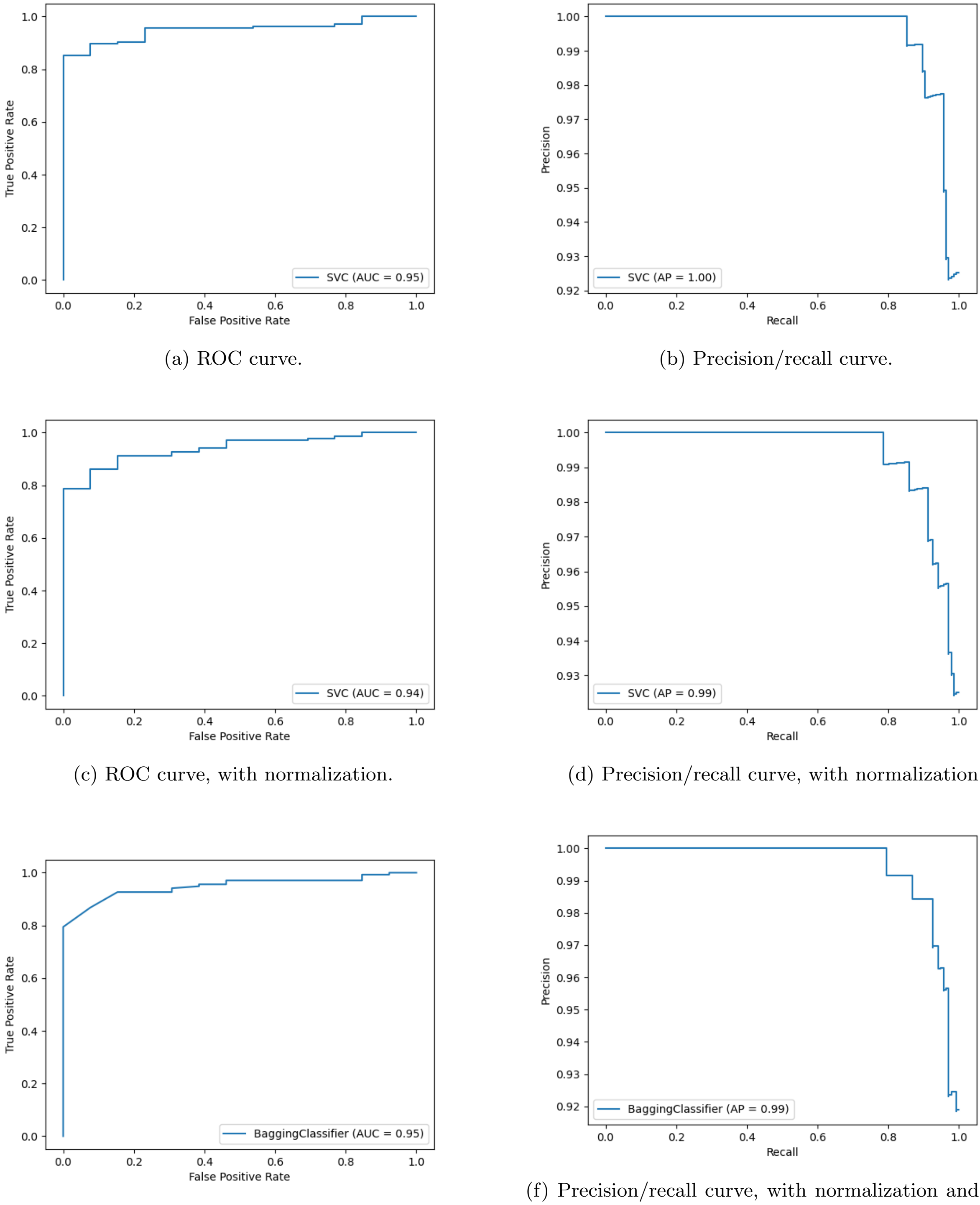
(c) Binary classification, normalized.

(d) Multi-class classification, normalized.

(e) Binary classification, normalized, with bagging.

(f) Multi-class classification, normalized, with bagging.

Figure 2: Confusion matrices for binary (a) and multi-class (b) classification.



(a) ROC curve.

(b) Precision/recall curve.

(c) ROC curve, with normalization.

(d) Precision/recall curve, with normalization.

(e) ROC curve, with normalization and bagging.

(f) Precision/recall curve, with normalization and bagging.

Figure 3: ROC (a), (c) and (e) and Precision/recall (b), (d) and (f) curves for binary classification problem, with normalization and bagging respectively.

	Prec.	**Recall**	**f1-score**	**Supp.**
IT	0.50	0.38	0.43	13
NON-IT	0.94	0.96	0.95	136
accuracy	-	-	0.91	149
macro avg	0.72	0.67	0.69	149
wgh. avg	0.90	0.91	0.91	149

Table 2: Performance summary for binary classification, without normalized features.

	Prec.	**Recall**	**f1-score**	**Supp.**
ES	0.93	0.87	0.90	15
FR	0.83	0.62	0.71	16
GE	0.61	0.73	0.67	15
IT	0.57	0.62	0.59	13
UK	0.79	0.79	0.79	24
US	0.85	0.86	0.86	66
accuracy	-	-	0.79	149
macro avg	0.76	0.75	0.75	149
wgh. avg	0.80	0.79	0.79	149

Table 3: Performance summary for multi-class classification, without normalized features.

	Prec.	**Recall**	**f1-score**	**Supp.**
IT	0.64	0.54	0.58	13
NON-IT	0.96	0.97	0.96	136
accuracy	-	-	0.91	149
macro avg	0.80	0.75	0.77	149
wgh. avg	0.93	0.93	0.93	149

Table 4: Performance summary for binary classification, with normalized features.

	Prec.	**Recall**	**f1-score**	**Supp.**
ES	1.00	0.87	0.93	15
FR	0.64	0.56	0.60	16
GE	0.57	0.87	0.68	15
IT	0.53	0.62	0.57	13
UK	0.75	0.75	0.75	24
US	0.85	0.77	0.81	66
accuracy	-	-	0.79	149
macro avg	0.72	0.74	0.72	149
wgh. avg	0.77	0.75	0.76	149

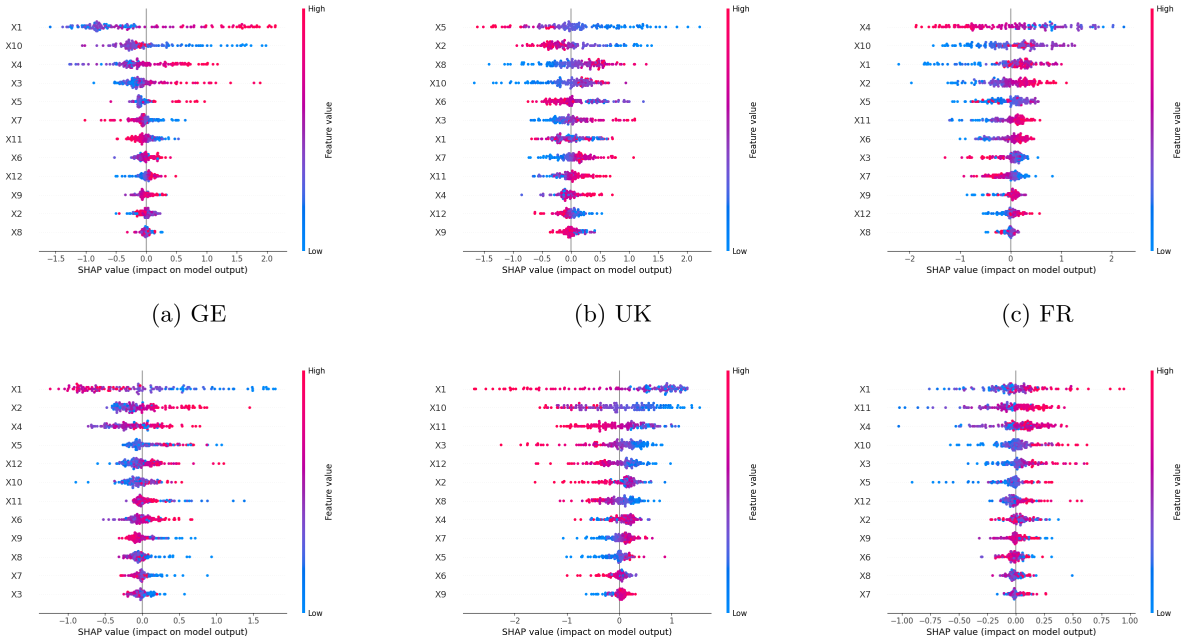
Table 5: Performance summary for multi-class classification, with normalized features.

	Prec.	**Recall**	**f1-score**	**Supp.**
IT	0.64	0.54	0.58	13
NON-IT	0.96	0.97	0.96	136
accuracy	-	-	0.91	149
macro avg	0.80	0.75	0.77	149
wgh. avg	0.93	0.93	0.93	149

Table 6: Performance summary for binary classification, with normalized features and bagging.

	Prec.	**Recall**	**f1-score**	**Supp.**
ES	1.00	0.87	0.93	15
FR	0.83	0.62	0.71	16
GE	0.75	0.80	0.77	15
IT	0.67	0.62	0.64	13
UK	0.86	0.79	0.83	24
US	0.85	0.95	0.90	66
accuracy	-	-	0.84	149
macro avg	0.83	0.78	0.80	149
wgh. avg	0.84	0.84	0.84	149

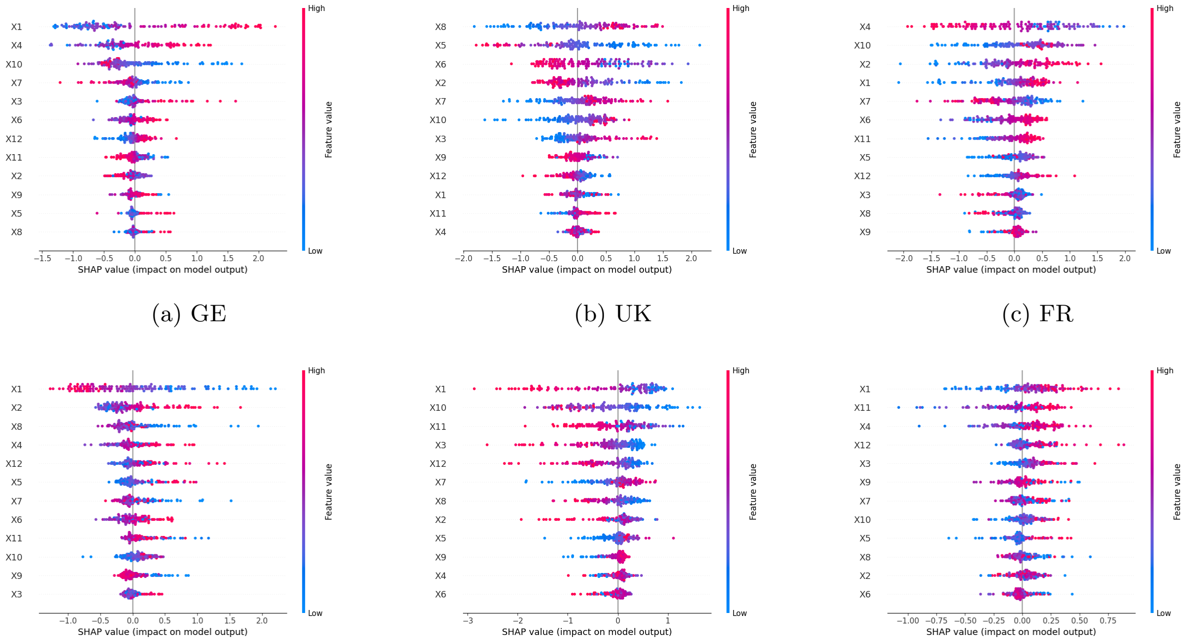
Table 7: Performance summary for multi-class classification, with normalized features and bagging.



(a) GE (b) UK (c) FR

(d) ES (e) US (f) IT

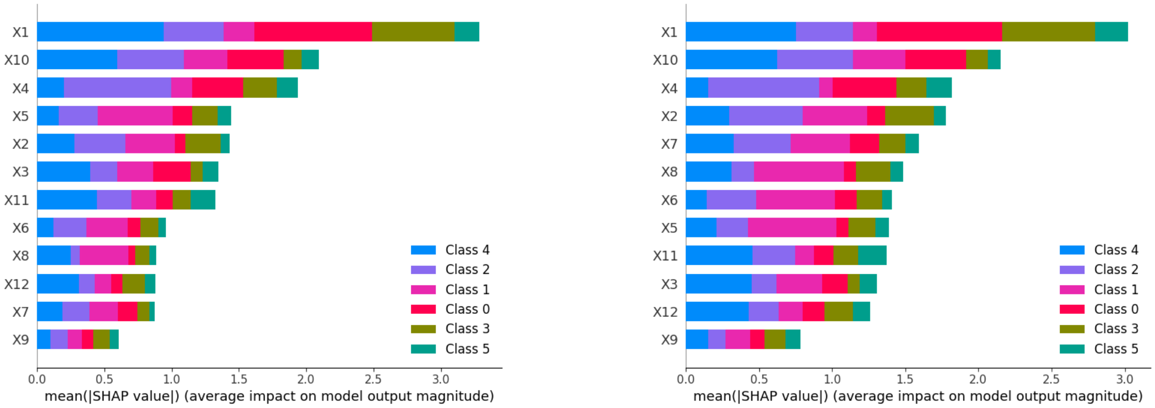
Figure 4: Shap analysis plots for not-normalized features. Each plot shows the importance of each feature in assigning the sample to one of the six language classes.



(a) GE (b) UK (c) FR

(d) ES (e) US (f) IT

Figure 5: Shap analysis plots for normalized features. Each plot shows the importance of each feature in assigning the sample to one of the six language classes.



(a) Not-normalized summary

(b) Normalized summary

Figure 6: Shap analysis plots for not-normalized features. Each plot shows the importance of each feature in assigning the sample to one of the six language classes.

References

[1] Shahzad Ahmed, Hyun In Jo, and Jin Yong Jeon. “Classification of human sounds using support vector machine with psycho-acoustic data”. In: July 2018.

[2] Peter Andras. “The Equivalence of Support Vector Machine and Regularization Neural Networks”. In: *Neural Processing Letters* 15 (Apr. 2002), pp. 97–104. DOI: 10.1023/A:1015292818897.

[3] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT ’92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401. URL: https://doi.org/10.1145/130385.130401.

[4] Leo Breiman. “Bagging Predictors”. In: *Mach. Learn.* 24.2 (Aug. 1996), pp. 123–140. ISSN: 0885-6125. DOI: 10.1023/A:1018054314350. URL: https://doi.org/10.1023/A:1018054314350.

[5] Gavin C. Cawley and Nicola L.C. Talbot. “On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation”. In: *J. Mach. Learn. Res.* 11 (Aug. 2010), pp. 2079–2107. ISSN: 1532-4435.

[6] Corinna Cortes and Vladimir Vapnik. “Support-Vector Networks”. In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411.

[7] Daghan Dogan. “Road-types classification using audio signal processing and SVM method”. In: June 2017. DOI: 10.1109/SIU.2017.7960154.

[8] Reza Entezari-Maleki, Arash Rezaei, and Behrouz Minaei. “Comparison of Classification Methods Based on the Type of Attributes and Sample Size”. In: *JCIT* 4 (Sept. 2009), pp. 94–102. DOI: 10.4156/jcit.vol4.issue3.14.

[9] Xuedong Huang et al. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. 1st. USA: Prentice Hall PTR, 2001. ISBN: 0130226165.

[10] A. Juneja and C. Espy-Wilson. “Segmentation of continuous speech using acoustic-phonetic parameters and statistical learning”. In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP ’02.* Vol. 2. 2002, 726–730 vol.2. DOI: 10.1109/ICONIP.2002.1198153.

[11] S. Knerr, L. Personnaz, and G. Dreyfus. “Single-layer learning revisited: a stepwise procedure for building and training a neural network”. In: *Neurocomputing*. Ed. by Françoise Fogelman Soulié and Jeanny Hérault. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 41–50. ISBN: 978-3-642-76153-9.

[12] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 4765–4774. URL: `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`.

[13] Zichen Ma and Ernest Fokou’e. “A Comparison of Classifiers in Performing Speaker Accent Recognition Using MFCCs”. In: *CoRR* [Web Link] (2015). arXiv: `1501.07866`. URL: `[Web%20Link]`.

[14] J. Mercer. “Functions of positive and negative type, and their connection with the theory of integral equations”. In: *Philosophical Transactions of the Royal Society, London* 209 (1909), pp. 415–446.

[15] J. Padrell-Sendra, D. Martín-Iglesias, and F. Díaz-de-María. “Support Vector Machines for continuous speech recognition”. In: *2006 14th European Signal Processing Conference*. 2006, pp. 1–4.

[16] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[17] J Shawe-Taylor and N Cristianini. “2 - Margin Distribution and Soft Margin”. English. In: *Advances in Large Margin Classifiers*. Ed. by Bertlett, B Schoelkopf, and C B Schuurmans. United States: Massachusetts Institute of Technology (MIT) Press, 1999.

[18] S.S. Stevens, J. Volkmann, and E.B. Newman. *A scale for the measurement of the psychological magnitude pitch*. 1937. URL: `https://books.google.it/books?id=9SCWoAEACAAJ`.

[19] Yingying Zhu, Zhong Ming, and Qiang Huang. “SVM-Based Audio Classification for Content- Based Multimedia Retrieval”. In: *Multimedia Content Analysis and Mining*. Ed. by Nicu Sebe et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 474–482. ISBN: 978-3-540-73417-8.