Parte 1 – Classificação de um Padrão de Voz

1. Correr o Ficheiro "Baseline"

Inicialmente correu-se o script Baseline fornecido pelo professor nas aulas de laboratório. Colocou-se todas as variáveis dos STEP[N] com N de 0 a 4 com valor positivo (true). Obteve-se uma percentagem inicial de **80,3653% (528/657)** de ficheiros corretamente classificados usando os comandos *"-t 0"* que corresponde a um Kernel linear. A maneira como se estruturou todo o processo, foi a seguinte: o nosso sistema retirou inicialmente alguns "features" dos ficheiros de áudio que nos eram fornecidos criando um modelo que utiliza a biblioteca de Support Vector Machines. Para que o sistema conseguisse aprender por si, foi-lhe fornecido um conjunto de ficheiros de treino, e este foi validado no conjunto de ficheiros de um ambiente de desenvolvimento. Conseguiu-se comprovar a qualidade do nosso classificador pois foram fornecidas labels para cada um dos ficheiros de treino e desenvolvimento.

A avaliação do nosso sistema será feita nos ficheiros de teste, com as labels que apenas os professores possuem.

2. Melhorar o Ficheiro "Baseline"

Inicialmente decidiu-se alterar os parâmetros do Passo 3 que corresponde a treinar o nosso modelo de Support Vector Machine. Para tal, fomos ler os parâmetros que podiam ser alterados e corremos cada um deles individualmente:

- *-t 1 (Polynomial)*
- *-t 2 (Radial Basis)*
- *-t 3 (Sigmoid)*
- *-s 0 C-SVC*
- *-s 1 nu-SVC*
- *-s 2 one-class SVM*
- *-s 3 epsilon-SVR*
- *-s 4 nu-SVR*

Em todas estas modificações o resultado foi idêntico com uma taxa de sucesso na identificação dos ficheiros do ambiente de desenvolvimento de cerca de **76,1035% (500/657)**. O que é ligeiramente inferior aos resultados anteriores que consideram um Kernel linear, no entanto todos estes realizavam muito menos iterações do que o Kernel linear. Tal se deve ao facto de termos poucos ficheiros para treinar o nosso sistema e para além disso cada ficheiro tem muitos parâmetros, ou seja features correspondentes.

De modo a aumentar um pouco a qualidade do classificador, e considerando desta vez o Kernel Linear por ter tido melhores resultados. Conseguiu-se detetar que grande parte do problema residia

na extração de features. Portanto foi feito o download através do site do *''open smile 2.3.0''* e guardados todos os ficheiros de configuração para extração de features. Decidiu-se então, substituir o método de extração de features pelo método Extended GeMAPS, cujos resultados podem ser encontrados abaixo:

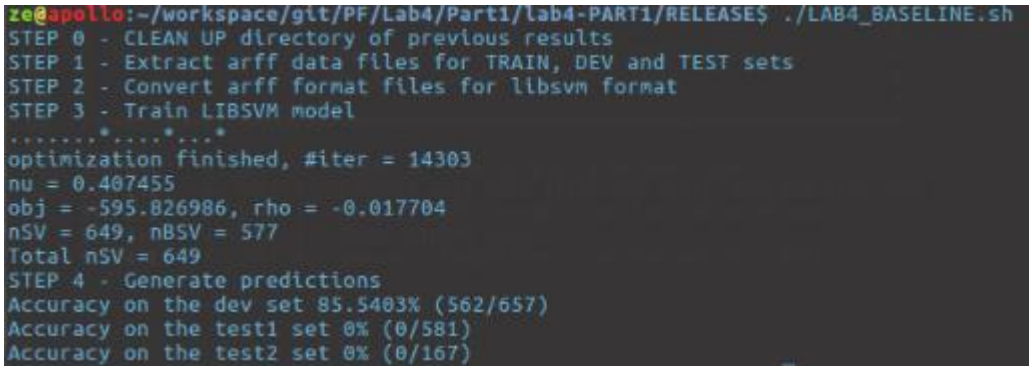

Figura 1 - Resultado de saída para Extended GeMAPS

Mesmo considerando que o resultado seria pior, testou-se a baseline com GeMAPS também e os resultados encontram-se ilustrados abaixo:

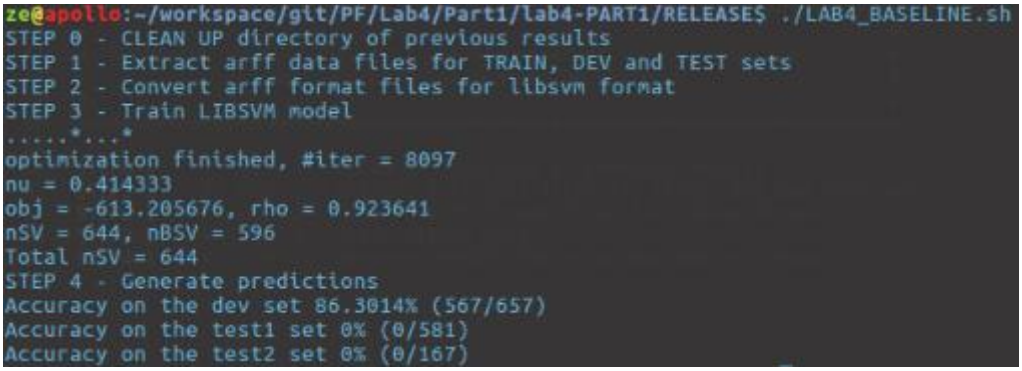

Figura 2 - Resultado de saída para GeMAPS

Dado que se tinham todos os outros ficheiros de configuração utilizados pelo open smile, foi criado um script para correr todos os estes ficheiros e imprimir qual o melhor resultado. Tanto o script **"runAllConfigs.sh"** como o output gerado **"resultsAllConfigs.txt"**, podem ser observados dentro do zip submetido. Uma nota a ter é que, certos ficheiros de configuração não se enquadram no método utilizado. Sendo por isso a razão de muitos deles obterem uma percentagem de accuracy final de 0.0%.

3. Decisões e Comentários Finais

Após terem sido corridos todos os testes com todas as suas variantes, o melhor resultado obtido foi, contrariamente ao esperado, kernel linear com método de extração de features de GeMAPS, com uma percentagem de accuracy no ambiente de desenvolvimento de **86,3014%**. A razão pela qual tanto o kernel radial como o método Extended GeMAPS não se revelaram superiores, deve-se ao facto de o conjunto de dados de treino ser muito reduzido, não sendo portanto uma amostra significativa e suficiente para correr estes dois métodos.