

16	EP01_05		66	106	156		4+R12	happiness
16	EP01_08		73	94	110		17	repression
16	EP01_09f		51	76	163		4+12	others
16	EP04_02f		181	269	271		R12	happiness

Supponiamo di concentrarci sul training del soggetto 16. A sinistra, la sezione del file excel casme coding relativa al soggetto 16; in seconda colonna abbiamo la sottocartella di riferimento mentre le colonne successive rappresentano l'onset, l'apex e l'offset frame della microespressione.

AU number ↕	FACS name ↕	Muscular basis ↕
0	Neutral face	
1	Inner brow raiser	frontalis (pars medialis)
2	Outer brow raiser	frontalis (pars lateralis)
4	Brow lowerer	depressor glabellae, depressor supercilii, corrugator supercilii
5	Upper lid raiser	levator palpebrae superioris, superior tarsal muscle
6	Cheek raiser	orbicularis oculi (pars orbitalis)
7	Lid tightener	orbicularis oculi (pars palpebralis)
8	Lips toward each other	orbicularis oris
9	Nose wrinkler	levator labii superioris alaeque nasi
10	Upper lip raiser	levator labii superioris, caput infraorbitalis
11	Nasolabial deepener	zygomaticus minor
12	Lip corner puller	zygomaticus major
13	Sharp lip puller	levator anguli oris (also known as caninus)
14	Dimpler	buccinator
15	Lip corner depressor	depressor anguli oris (also known as triangularis)
16	Lower lip depressor	depressor labii inferioris
17	Chin raiser	mentalis

La penultima colonna indica l'AU che ci serve per individuare la regione del viso interessata dalla microespressione.

Supponiamo di occuparci del file raw relativo alla bocca ( datax\_0 ), notiamo, grazie alla tabella delle AU qui a sinistra e consultabile online, che le microespressioni con AU 12 coinvolgono la bocca ( quelle con AU 4 coinvolgono anche le sopracciglia e quindi dovranno essere etichettate negli altri file raw) .

Supponiamo inoltre di adottare una soglia di true positive dell' **80%**, ovvero la finestra corrente contiene effettivamente un esempio positivo se contiene almeno l'80% dei frame della microespressione segnalati dal file excel.

Vediamo adesso come etichettare correttamente il file.

16	EP01_05		66	106	156		4+R12	happiness
16	EP01_08		73	94	110		17	repression
16	EP01_09f		51	76	163		4+12	others
16	EP04_02f		181	269	271		R12	happiness

```

datax_0_raw - Blocco note di Windows
File Modifica Formato Visualizza ?
+2 1:0.0 2:0.05 3:0.6 4:0.67 5:0.01 6:0.0 7:0.0
4:0.0 135:0.0 136:0.0 137:0.04 138:2.19 139:0.0
AAA end window
+2 1:1.31 2:0.01 3:0.0 4:0.0 5:0.0 6:0.0 7:0.0
134:0.0 135:0.0 136:0.11 137:0.0 138:0.0 139:0.0
AAA end window
END
+2 1:0.0 2:0.0 3:0.0 4:0.0 5:0.01 6:0.39 7:1.02
:1.02 132:0.8 133:0.0 134:0.0 135:0.0 136:0.0
AAA end window

```



```

datax_0_labelled - Blocco note di Windows
File Modifica Formato Visualizza ?
+2 1:0.0 2:0.05 3:0.6 4:0.67 5:0.01 6:0.0 7:0.0
4:0.0 135:0.0 136:0.0 137:0.04 138:2.19 139:0.0
+1 1:1.31 2:0.01 3:0.0 4:0.0 5:0.0 6:0.0 7:0.0
134:0.0 135:0.0 136:0.11 137:0.0 138:0.0 139:0.0
+2 1:0.0 2:0.0 3:0.0 4:0.0 5:0.01 6:0.39 7:1.02
:1.02 132:0.8 133:0.0 134:0.0 135:0.0 136:0.0

```

Sopra a sinistra abbiamo una sezione del file raw generato dall' esecuzione del programma mentre a destra una sezione del file correttamente etichettato e pronto ad essere usato per l'addestramento.

I descrittori sono in formato LibSVM per cui abbiamo un'etichetta iniziale (+1 o +2) e una serie di parametri 1: x 2: x 3: x....

Come indicato, il carattere "AAA" indica la fine della finestra mentre "END" la fine della sottocartella.

In questo caso nella prima cartella sono individuate 2 finestre (una che va dal frame 0 a 100 e una da 60 a 160); come notiamo dalla tabella excel la microespressione della bocca della prima cartella è contenuta completamente nella seconda finestra e per soli 24/90 frame nella prima. Per la soglia imposta, la microespressione deve essere segnalata solo nella seconda finestra cambiando l'etichetta in +1; l'etichetta della prima finestra rimane invariata ma bisogna procedere a rimuovere i caratteri di delimitazione come è possibile vedere nel file etichettato.

Si procede poi con lo stesso ragionamento per la seconda sottocartella che avrà la prima finestra successiva al carattere END.

16	EP01_05		66	106	156		4+R12	happiness
16	EP01_08		73	94	110		17	repression
16	EP01_09f		51	76	163		4+12	others
16	EP04_02f		181	269	271		R12	happiness

```

datax_0_raw - Blocco note di Windows
File Modifica Formato Visualizza ?
END
+2 1:0.0 2:0.04 3:2.77 4:0.79 5:0.0 6:0.0 7:0.0
05 133:0.02 134:0.02 135:0.01 136:0.0 137:0.0
AAA end window
+2 1:0.14 2:0.03 3:0.02 4:0.03 5:0.05 6:0.03
134:0.05 135:2.63 136:0.37 137:0.0 138:0.0 139:0.0
AAA end window
+2 1:0.0 2:0.0 3:0.0 4:0.01 5:0.95 6:1.15 7:0.0
2 135:0.29 136:0.15 137:0.0 138:0.0 139:0.0 140:0.0
AAA end window
+2 1:0.02 2:0.55 3:0.87 4:0.07 5:0.0 6:0.0 7:0.0
0.0 133:0.01 134:0.03 135:0.07 136:0.31 137:0.0
AAA end window
+2 1:0.03 2:4.39 3:0.16 4:0.01 5:0.0 6:0.0 7:0.0
01 135:0.11 136:3.19 137:0.0 138:0.0 139:0.0
AAA end window
+2 1:0.01 2:0.0 3:0.0 4:0.0 5:0.0 6:0.02 7:0.0
.01 134:0.01 135:0.41 136:2.76 137:1.04 138:0.0
AAA end window
+2 1:0.0 2:0.0 3:0.01 4:0.74 5:0.59 6:0.04 7:0.0
0.16 133:0.08 134:0.01 135:0.0 136:0.01 137:0.0
AAA end window
+2 1:0.0 2:0.0 3:0.0 4:0.0 5:0.01 6:0.81 7:0.0

```



```

datax_0_labelled - Blocco note di Windows
File Modifica Formato Visualizza ?
+2 1:0.0 2:0.04 3:2.77 4:0.79 5:0.0 6:0.0 7:0.0
05 133:0.02 134:0.02 135:0.01 136:0.0 137:0.0
+2 1:0.14 2:0.03 3:0.02 4:0.03 5:0.05 6:0.03
134:0.05 135:2.63 136:0.37 137:0.0 138:0.0 139:0.0
+2 1:0.0 2:0.0 3:0.0 4:0.01 5:0.95 6:1.15 7:0.0
2 135:0.29 136:0.15 137:0.0 138:0.0 139:0.0 140:0.0
+1 1:0.02 2:0.55 3:0.87 4:0.07 5:0.0 6:0.0 7:0.0
0.0 133:0.01 134:0.03 135:0.07 136:0.31 137:0.0
+2 1:0.03 2:4.39 3:0.16 4:0.01 5:0.0 6:0.0 7:0.0
01 135:0.11 136:3.19 137:0.0 138:0.0 139:0.0
+2 1:0.01 2:0.0 3:0.0 4:0.0 5:0.0 6:0.02 7:0.0
.01 134:0.01 135:0.41 136:2.76 137:1.04 138:0.0
+2 1:0.0 2:0.0 3:0.01 4:0.74 5:0.59 6:0.04 7:0.0
0.16 133:0.08 134:0.01 135:0.0 136:0.01 137:0.0
+2 1:0.0 2:0.0 3:0.0 4:0.0 5:0.01 6:0.81 7:0.0
32:0.14 133:0.07 134:0.01 135:0.01 136:0.08 137:0.0
+2 1:0.0 2:0.0 3:0.0 4:0.0 5:0.01 6:0.49 7:0.0
0.0 133:0.11 134:3.66 135:0.28 136:0.28 137:0.0
+2 1:0.45 2:0.99 3:1.22 4:0.63 5:0.23 6:0.15
41 129:0.31 130:0.84 131:1.51 132:1.16 133:0.0
+2 1:1.43 2:0.01 3:0.0 4:0.0 5:0.0 6:0.0 7:0.0
:0.0 133:0.02 134:0.02 135:0.11 136:1.8 137:0.0

```

Altro esempio.

Siamo arrivati alla quarta sottocartella; questa contiene molte finestre in quanto il video è particolarmente lungo.

Abbiamo dunque le finestre 0-100, 60-160, 120-220, 180-280, 240-340 ecc..

La microespressione è completamente contenuta nella finestra 4 mentre ha 39/90 frame nella finestra 3 e 31/90 frame nella finestra 5.

Come possiamo vedere dal file etichettato solo la finestra 4 avrà etichetta +1.

Se si pone una soglia molto più bassa, ad esempio 40%, anche la finestra 3 dovrà avere etichetta +1.

Il file raw e quello etichettato sono consultabili inoltre per intero sempre nella cartella training\_example.






Eseguito questo procedimento anche per i file raw relativi anche alle regioni delle sopracciglia e ripetendo per altri soggetti si possono combinare i vari esempi in 3 unici file che rappresentano i 3 dataset per la detection di ME di bocca, sopracciglia dx e sx (vedere i file DatasetX presenti in source).

Dato a questo punto uno di questi file ( si può usare anche solo il file etichettato del soggetto 16 ma contiene solo 3 esempi positivi), si può inserire il path e chiamare la funzione SVM\_model di svm.py.

Ex. con "datax\_0" presente in source:

```
SVM_model("datax_0.txt")
```

Vengono poi generati i file in figura in basso; il file modello e il file range serviranno poi per le predizioni future.

 datax_0	30/07/2020 23:36	Documento di testo	23 KB
 datax_0.txt	30/07/2020 23:36	File MODEL	27 KB
 datax_0.txt	30/07/2020 23:36	File RANGE	4 KB
 datax_0.txt	30/07/2020 23:36	File SCALE	28 KB
 datax_0.txt.scale	30/07/2020 23:36	File OUT	4 KB

```
def SVM_predict(folder, start_frame, end_frame, type):
    svmscale_exe = r".\windows\svm-scale.exe"
    svmpredict_exe = r".\windows\svm-predict.exe"

    assert os.path.exists(svmscale_exe), "svm-scale executable not found"
    assert os.path.exists(svmpredict_exe), "svm-predict executable not found"

    testPath = "testData" + str(type) + ".txt"

    file_name = "testData" + str(type)
    model_file = "DatasetX" + str(type) + ".txt.model"
    scaled_test_file = file_name + ".scale"
    predict_test_file = file_name + ".predict"
    range_file = "DatasetX" + str(type) + ".txt.range"

    cmd = '{0} -r "{1}" "{2}" > "{3}"'.format(svmscale_exe, range_file, testPath, scaled_test_file)
    #print('Scaling testing data...')
    Popen(cmd, shell=True, stdout=PIPE).communicate()
```

Supponendo di aver svolto il procedimento anche per datax\_1 e datax\_2 e di avere i file nella cartella source, se nella funzione SVM\_predict sostituiamo "DatasetX\_" con "datax\_" il predict punterà ai nuovi modelli e le predizioni future saranno basate su questi.