

# Reconocimiento de Emociones en Habla Utilizando SVM

Lia Natalia Chicue Garcia.

Machine Learning  
Maestría en Ingeniería de Sistemas y Computación  
Universidad Tecnológica de Pereira



Universidad Tecnológica  
de Pereira

Noviembre 21 / 2021

# Motivación

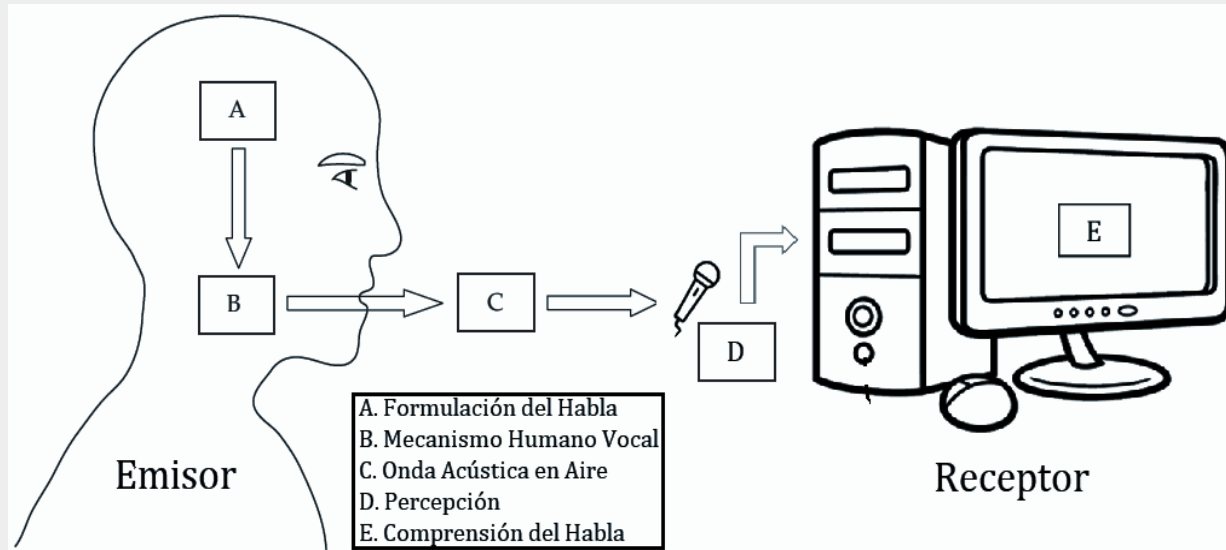
A horizontal row of 18 small red squares, evenly spaced, extending across the width of the slide.

## Desarrollo de HCIs:

- Asistentes virtuales.
- Sistemas de vigilancia sanitaria.
- Marketing y Publicidad

# El SER y sus Desafíos

## Modelo oculto de Markov (HMM)



**Clasificar emociones en habla a partir de la extracción de un grupo de características de sonido utilizando máquinas de soporte vectorial**

# Materiales



Base de Datos Características	Etiquetas	Features
<ol style="list-style-type: none"><li>1. <a href="#">Ravdess</a></li><li>2. <a href="#">Crema</a></li><li>3. <a href="#">Tess</a></li><li>4. <a href="#">Savee</a></li></ol>	<ul style="list-style-type: none"><li>● Happy</li><li>● Fear</li><li>● Sad</li><li>● Surprise</li><li>● Disgust</li><li>● Neutral</li><li>● Angry</li></ul>	<ul style="list-style-type: none"><li>● Energía</li><li>● Vector Chroma</li><li>● Espectrograma de Mel</li><li>● Coeficientes Cepstrales en las Frecuencias de Mel</li><li>● Raíz del error cuadrático medio</li><li>● Velocidad de cruce cero</li><li>● Centroide Espectral</li><li>● Ancho de Banda Espectral</li><li>● Contraste Espectral</li><li>● Coeficiente de Tonalidad o Planitud Espectral</li><li>● Frecuencia Espectral Rolloff</li><li>● Tonnetz</li></ul>

# Métodos



Extracción de Features	Método SVM
<p>Se creó una <a href="#">función</a> que extrae las características anteriormente nombradas.</p>	<p>Se utilizará el toolkit de sklearn llamado <a href="#">SVC</a>, el cual es capaz de clasificar bases de datos multi-clase apoyado en métodos de aprendizaje supervisado como lo son las máquinas de soporte vectorial.</p>

# Energía

```
energia = np.array([np.sum(np.abs(data[i:i+frame_length]**2))  
                    for i in range(0, len(data.shape[0]), hop_length)  
                    ]) / frame_length  
  
eng = np.mean(energia, axis=0)
```

# Vector Croma

```
stft = np.abs(librosa.stft(data))  
  
chroma_stft = np.mean(librosa.feature.chroma_stft(S=stft,  
sr=sample_rate).T, axis=0)
```

# Espectrograma de Mel

```
mel = np.mean(librosa.feature.melspectrogram(y=data,  
sr=sample_rate).T, axis=0)
```

# Coeficientes Cepstrales en las Frecuencias de Mel

```
mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T,  
axis=0)
```



# Raíz del error cuadrático medio

```
rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
```

# Velocidad de cruce cero

```
zcr = np.mean(librosa.feature.zero_crossing_rate(y=data).T,  
axis=0)
```

# Centroide Espectral

```
cent = np.mean(librosa.feature.spectral_centroid(y=data,  
sr=sample_rate).T,axis=0)
```

# Ancho de Banda Espectral

```
spec_bw = np.mean(librosa.feature.spectral_bandwidth(y=data, sr=sample_rate).T,  
axis=0)
```

# Contraste Espectral

```
S_c = np.abs(librosa.stft(data))  
  
contrast = np.mean(librosa.feature.spectral_contrast(S=S_c,  
sr=sample_rate).T, axis=0)
```

# Coeficiente de Tonalidad o Planitud Espectral

```
flatness = np.mean(librosa.feature.spectral_flatness(y=data).T,  
axis=0)
```

# Frecuencia Espectral de Desplazamiento

```
rolloff=np.mean(librosa.feature.spectral_rolloff(y=data,  
sr=sample_rate).T,axis=0)
```

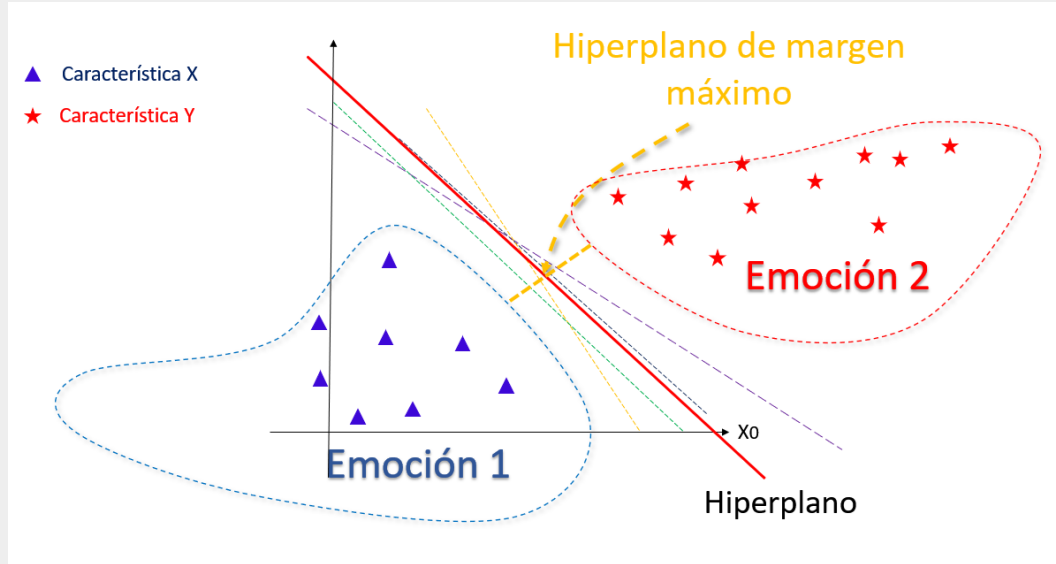
# Tonnetz

```
y1 = librosa.effects.harmonic(data)

tonnetz = np.mean(librosa.feature.tonnetz(y=y1,
sr=sample_rate).T,axis=0)
```

# Clasificación en SVM

1. Hiperplano separador
2. Hiperplano de Margen Máximo
3. Margen
4. Kernel



# Resultados y Discusiones

- Precisión de la clasificación

Accuracy Test (%) [Regularización $C=0,01$ ]				
test_size		0,2	0,3	0,5
Kernel	linear	56,568	55,492	53,792
	rbf	32,007	26,847	25,304
	poly	24,604	22,368	22,428
	sigmoid	29,937	24,508	21,640

Accuracy Test (%) [Regularización $C = 1$ ]				
test_size		0,2	0,3	0,5
Kernel	linear	58,280	56,662	55,966
	rbf	58,708	56,091	56,309
	poly	45,528	43,823	42,390
	sigmoid	37,484	38,088	40,370

Accuracy Test (%) [Regularización $C = 10$ ]				
test_size		0,2	0,3	0,5
Kernel	linear	57,938	56,234	54,888
	rbf	62,730	61,170	60,043
	poly	55,413	53,210	52,542
	sigmoid	32,092	32,354	32,649

Accuracy Test (%) [Regularización $C = 100$ ]				
test_size		0,2	0,3	0,5
Kernel	rbf	64,014	62,767	61,701
	poly	42,747	41,940	41,089
	sigmoid	30,894	31,583	31,056

Accuracy Test (%) [Regularización $C = 1000$ ]				
test_size		0,2	0,3	0,5
Kernel	rbf	61,575	60,114	58,534
	poly	50,107	49,443	49,375
	sigmoid	30,809	31,583	31,552

Accuracy Test (%) [Regularización $C = 1 \times 10^6$ ]				
test_size		0,2	0,3	0,5
Kernel	rbf	60,419	59,087	57,730
	poly	59,264	58,060	55,384
	sigmoid	30,766	31,555	31,450

Accuracy Test (%) [test_size = 0,3]							
Regularización		0,01	1	10	100	1000	$1 \times 10^6$
deg	3	22,368	43,823	53,210	41,940	49,443	58,060
	6	21,789	31,783	36,119	41,94	49,444	54,236
	10	21,74	27,389	30,442	32,781	35,521	45,221

- Classification Report

	precision	recall	f1-score	support
happy	0,67	0,71	0,69	567
fear	0,54	0,57	0,55	560
sad	0,63	0,55	0,59	566
surprise	0,57	0,55	0,56	539
disgust	0,65	0,66	0,66	560
neutral	0,62	0,66	0,6	535
angry	0,91	0,84	0,87	178

	precision	recall	f1-score	support
macro avg	0,66	0,65	0,65	3505
weigthed av	0,63	0,63	0,63	3505



# Conclusiones



El kernel de función de base radial resulta óptimo para el proceso de clasificación de grandes números de datos, como lo es la base de datos empleada en el presente proyecto, debido a que almacena los vectores de soporte únicamente durante el entrenamiento y no todo el conjunto de datos acelerando el proceso de ejecución y aumentando porcentaje de precisión. Esto último corresponde a que nuestra base de datos es un conjunto no linealmente separable.

# Trabajos Futuros



- La mayoría de las bases de datos en el estado del arte están en inglés. Se recomienda crear una base de datos en diferentes idiomas con el fin de verificar los valores de las características extraídas con las emociones de etiquetado.
- Se plantea utilizar otros medios de procesamiento de datos como Big data con el fin aumentar los tiempos de ejecución.

# Bibliografía



- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.
- Oppenheim, A. (1975). V. and Schafer, Ronald W. Discrete-Time Signal Processing.
- Lieskovská, E., Jakubec, M., Jarina, R., & Chmulík, M. (2021). A Review on Speech Emotion Recognition Using Deep Learning and Attention Mechanism. Electronics, 10(10), 1163.
- Selvaraj, Mahalakshmi & Bhuvana, R. & Karthik, S Padmaja. (2016). Human speech emotion recognition. 8. 311-323.

# Bibliografía



- Ramakrishnan, S., & El Emary, I. M. (2013). Speech emotion recognition approaches in human computer interaction. Telecommunication Systems, 52(3), 1467-1478.
- Karg, M. (2012). Pattern recognition algorithms for gait analysis with application to affective computing (Doctoral dissertation, Technische Universität München).