

# TRABAJO FINAL

Angelo Gabriel Manrique Rangel, Luis Rafael Reimóndez Blanco, Oscar Hugo Romero Samaniego y Asier Percaz Angós

# DERRAMES

- ¿QUE SON?
- ¿PORQUE OCURREN?
- ¿SON TRATABLES?
- ¿QUE SUPONEN?





# DATASET

<input checked="" type="checkbox"/> gender	# age	# hypertensi...	# heart_dise...
Male	67	0	1
Female	61	0	0

<input checked="" type="checkbox"/> ever_marri...	<input checked="" type="checkbox"/> work_type	<input checked="" type="checkbox"/> Residence...	# avg_gluco...	<input checked="" type="checkbox"/> bmi
Yes	Private	Urban	228.69	36.6
Yes	Self-employed	Rural	202.21	N/A

<input checked="" type="checkbox"/> bmi	<input checked="" type="checkbox"/> smoking_s...	# stroke
36.6	formerly smoked	1
N/A	never smoked	1



1	Male	67	0	1	1	Private	Urban	228.69	36.6	1	1
2	Female	61	0	0	1	Self-employed	Rural	202.21	0	0	1
3	Male	80	0	1	1	Private	Rural	105.92	32.5	0	1
4	Female	49	0	0	1	Private	Urban	171.23	34.4	2	1
5	Female	79	1	0	1	Self-employed	Rural	174.12	24	0	1
6	Male	81	0	0	1	Private	Urban	186.21	29	1	1



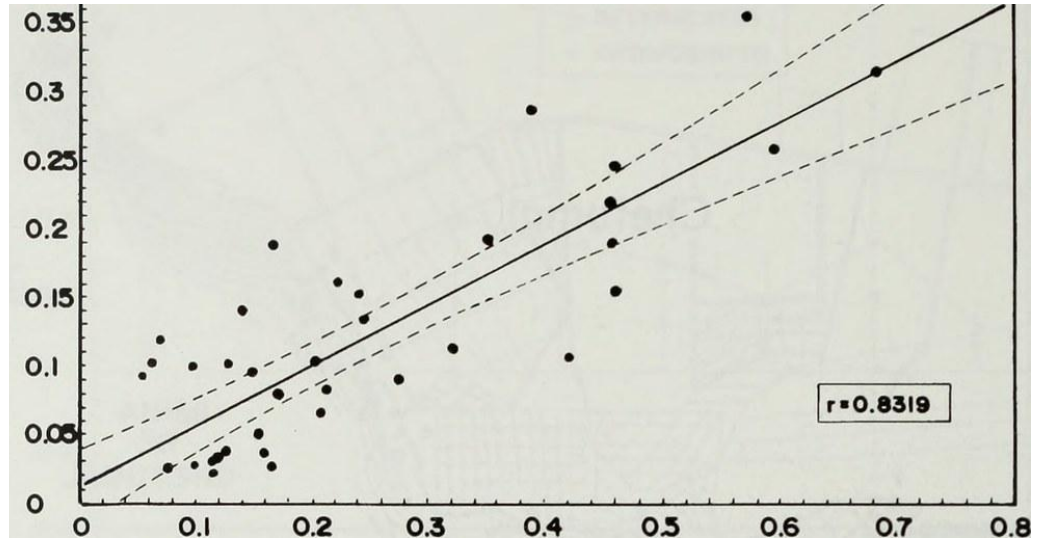
# NORMALIZAR

- FRAGMENTA
- .H
- .C
- FUNCIONES
- PROBLEMAS

```
void normalizeData(struct Patient pacientes[], int num_pacientes) {  
  
    MaxYMin maximos_minimos;  
    calcularMaxYMin(pacientes, &maximos_minimos, num_pacientes);  
    for (int i = 0; i < num_pacientes; i++) {  
        pacientes[i].glucose = (pacientes[i].glucose - maximos_minimos.min_glucose) / (maximos_minimos.max_glucose - maximos_minimos.min_glucose);  
        pacientes[i].bmi = (pacientes[i].bmi - maximos_minimos.min_bmi) / (maximos_minimos.max_bmi - maximos_minimos.min_bmi);  
        pacientes[i].age = (pacientes[i].age - maximos_minimos.min_age) / (maximos_minimos.max_age - maximos_minimos.min_age);  
        pacientes[i].smoking_status = (pacientes[i].smoking_status)/2;  
    }  
}
```

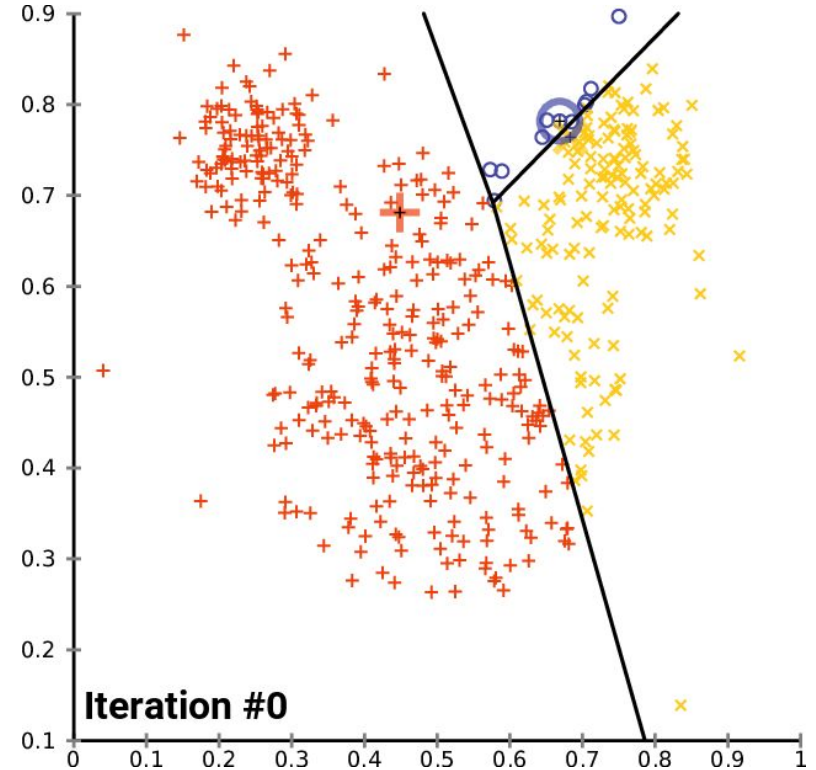
# DISTANCIAS MÍNIMAS


- FUNCIONAMIENTO
- .H
- .C
- EJECUTOR



# KNN

- ¿QUE ES?
- FUNCIONAMIENTO
- CODIFICACIÓN
- PROBLEMAS EN EL DESARROLLO






```
for (indice1 = 0; indice1 < num_pacientes; indice1++){
```

```
for (indice2 = 0; indice2 < num_pacientes; indice2++){
```

```
    dist_total = sqrt(dist_age * dist_age + dist_bmi * dist_bmi +
    elementoLista.distancia = dist_total;
    elementoLista.stroke = pacientes[indice2].stroke;
    insertar(&lista, elementoLista);
}
check = lista.ini->elemento.stroke;
```





```
check = lista.ini->elemento.stroke;
for (indice2 = 0; indice2 < k; indice2++){
    if(lista.ini->elemento.stroke == 1){
        contador1++;
    }
    else{
        contador0++;
    }
    eliminarMenor(&lista);
}
```

```
if (contador0 < contador1){
    prediccion = 1;
}
else if (contador0 > contador1){
    prediccion = 0;
}
else if (contador0 == contador1){
    prediccion = check;
}
if (prediccion == pacientes[indice1].stroke){
    contadorAciertos++;
}
```

```
alumno@eim-alu-69099:~/Escritorio/abb$ ./knn_wilson
Qué número de K quieres usar (1 <= K < 999): 1
Quieres usar Wilson si ejecutas otra vez knn s/n: n
La probabilidad de acierto para 999 casos con K = 1, con 0 wilson es: 72.27227%
numero de aciertos: 722    numero de datos: 999
```

```
alumno@eim-alu-69099:~/Escritorio/abb$ ./knn_wilson
Qué número de K quieres usar (1 <= K < 999): 6
Quieres usar Wilson si ejecutas otra vez knn s/n: n
La probabilidad de acierto para 999 casos con K = 6, con 0 wilson es: 75.67567%
numero de aciertos: 756    numero de datos: 999
Deseas ejecutar otra vez knn s/n: █
```

```
alumno@eim-alu-69099:~/Escritorio/abb$ ./knn_wilson
Qué número de K quieres usar (1 <= K < 999): 15
Quieres usar Wilson si ejecutas otra vez knn s/n: n
La probabilidad de acierto para 999 casos con K = 15, con 0 wilson es: 77.67768%
numero de aciertos: 776    numero de datos: 999
Deseas ejecutar otra vez knn s/n: █
```




# WILSON

- ¿QUE ES?
- FUNCIONAMIENTO
- CODIFICACIÓN
- PROBLEMAS EN EL DESARROLLO

```
else if(prediccion != pacientes[indice1].stroke && (resW == 's' || resW == 'S')){  
    contadorFallos++;  
    pacientes[indice1].num = NUM_FALLOS;
```

```
if(resW == 's' || resW == 'S'){  
    for(int i = 0; i < num_pacientes; i++){  
        if(pacientes[i].num == NUM_FALLOS){  
            pacientes[i].num = SALTAR_FALLOS;  
        }  
    }  
    contadorWilson++;
```



```
for (indice1 = 0; indice1 < num_pacientes; indice1++){
    contador1 = 0;
    contador0 = 0;
    while(pacientes[indice1].num == SALTAR_FALLOS){
        indice1++;
    }
    for (indice2 = 0; indice2 < num_pacientes; indice2++){
        while(indice1 == indice2 || pacientes[indice2].num == SALTAR_FALLOS){
            indice2++;
        }
    }
}
```

```
alumno@eim-alu-69099:~/Escritorio/abb$ ./knn_wilson
Qué número de K quieres usar ( $1 \leq K < 999$ ): 1
Quieres usar Wilson si ejecutas otra vez knn s/n: s
La probabilidad de acierto para 999 casos con  $K = 1$ , con 0 wilson es: 72.27227%
numero de aciertos: 722    numero de datos: 999
Deseas ejecutar otra vez knn s/n: s
Qué número de K quieres usar ( $1 \leq K < 722$ ): 1
Quieres usar Wilson si ejecutas otra vez knn s/n: n
La probabilidad de acierto para 722 casos con  $K = 1$ , con 1 wilson es: 97.50692%
numero de aciertos: 704    numero de datos: 722
Deseas ejecutar otra vez knn s/n: █
```

```
alumno@eim-alu-69099:~/Escritorio/abb$ ./knn_wilson
Qué número de K quieres usar ( $1 \leq K < 999$ ): 6
Quieres usar Wilson si ejecutas otra vez knn s/n: s
La probabilidad de acierto para 999 casos con  $K = 6$ , con 0 wilson es: 75.67567%
numero de aciertos: 756    numero de datos: 999
Deseas ejecutar otra vez knn s/n: s
Qué número de K quieres usar ( $1 \leq K < 756$ ): 6
Quieres usar Wilson si ejecutas otra vez knn s/n: n
La probabilidad de acierto para 756 casos con  $K = 6$ , con 1 wilson es: 94.70900%
numero de aciertos: 716    numero de datos: 756
Deseas ejecutar otra vez knn s/n: █
```