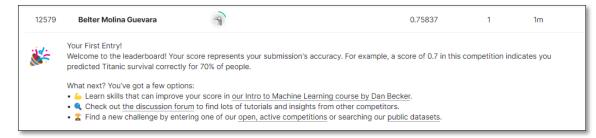
Nombre: Belter Molina Guevara

Carnet: 21002925
MODELOS GENERADOS

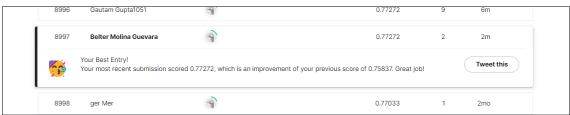
## Arquitectura de Red 1 y posición en Kaggle Modelo 1





### Arquitectura de Red 2 y posición en Kaggle Modelo 2





# Arquitectura de Red 3 y posición en Kaggle Modelo 3

```
clasificador = Sequential()
# Primera Capa
clasificador.add(Dense(7, activation = 'relu', input_shape=(7,),kernel_initializer='uniform'))
clasificador.add(Dense(50, activation = "relu",kernel_initializer='uniform'))

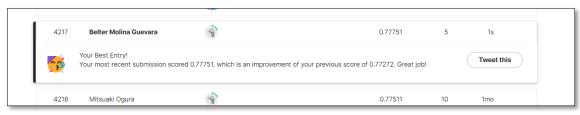
# Capa de Salida
# utilizamos la activacion Softmax, esta devolvera una probabilidad para cada resultado y elegiremos la probabilidad mas alta como nuestra prediccion
clasificador.add(Dense(2, activation = "softmax",kernel_initializer='uniform'))

[495] # parametros de optimizacion
clasificador.compile(loss-keras.losses.sparse_categorical_crossentropy, optimizer = "adam", metrics = ['accuracy'])
```

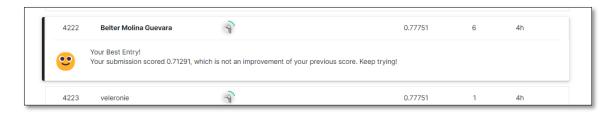


### Arquitectura de Red 4 y posición en Kaggle Modelo 4





#### Arquitectura de Red 5 y posición en Kaggle Modelo 5



```
clasificador = Sequential()

# Primera Capa
clasificador.add(Dense(7, activation = 'relu', input_shape=(7,),kernel_initializer='random_normal'))
clasificador.add(Dropout(8.2))

# Segunda Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation = "relu",kernel_initializer='random_normal'))
clasificador.add(Dropout(8.2))

# Tercera Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation = "relu",kernel_initializer='random_normal'))
clasificador.add(Dense(50, activation = "relu",kernel_initializer='random_normal'))

# Capa de Salida

# utilizanos la activacion sigmoid, esta devolvera una probabilidad para cada resultado y elegiremos la probabilidad mas alta como nuestra prediccion
clasificador.add(Dense(2, activation = "sigmoid",kernel_initializer='random_normal'))

[614] # parametros de optimizacion
clasificador.compile(loss=keras.losses.sparse_categorical_crossentropy, optimizer = "SGD", metrics = ['accuracy'])
```

#### **CONCLUSION:**

Vemos que de las diferentes configuraciones de red neuronal en las que se incluye, por ejemplo, cantidad de neuronas, activación, el kernel initializer y diferentes algoritmos de activación como lo son Adam, SGD, softmax, Sigmoid, el un número determinado de épocas.

La configuración que mas se adapto fue utilizando los parámetros de optimizador Adam, algoritmo de activación Sigmoid en la ultima capa y en las primeras capas se utilizó el algoritmo Relu, trabajando con una cantidad de épocas de 400, con una configuración de redes neuronales de 50. Podemos observar que el algoritmo de Adam trabaja mejor ya que no importa si los datos son pequeños o grandes.