

Nombre: Belter Molina Guevara  
Carnet: 21002925  
MODELOS GENERADOS

Arquitectura de Red 1 y posición en Kaggle Modelo 1

```
clasificador = Sequential()
# Primera Capa
clasificador.add(Dense(7, activation = 'relu', input_shape=(7,)))
clasificador.add(Dropout(0.2))

# agregamos 1 capa mas con 4 neuronas y una activacion relu
clasificador.add(Dense(4, activation = "relu"))

#clasificador.add(Dropout(0.2))
#clasificador.add(Dense(50, activation = "relu"))

# agregamos nuestra 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"))

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

# Entrenando nuestro modelo
clasificador.fit(X, y, batch_size=1, epochs=100, validation_split = 0.2, verbose = 0)
```

12579

Belter Molina Guevara

0.75837

1

1m

Your First Entry!

Welcome to the leaderboard! Your score represents your submission's accuracy. For example, a score of 0.7 in this competition indicates you predicted Titanic survival correctly for 70% of people.

What next? You've got a few options:

- 👉 Learn skills that can improve your score in our [Intro to Machine Learning course](#) by Dan Becker.
- 💬 Check out the [discussion forum](#) to find lots of tutorials and insights from other competitors.
- 🏆 Find a new challenge by entering one of our [open, active competitions](#) or searching our [public datasets](#).

Arquitectura de Red 2 y posición en Kaggle Modelo 2

```
clasificador = Sequential()
# Primera Capa
clasificador.add(Dense(7, activation = 'relu', input_shape=(7,),kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# Segunda Capa con 10 neuronas y una activacion relu
clasificador.add(Dense(10, activation = "relu",kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# Tercera Capa con 10 neuronas y una activacion relu
clasificador.add(Dense(10, activation = "relu",kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# 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'))
```

8996	Gautam Gupta1051		0.77272	9	6m
8997	Belter Molina Guevara		0.77272	2	2m
<div><div></div><div><p>Your Best Entry!</p><p>Your most recent submission scored 0.77272, which is an improvement of your previous score of 0.75837. Great job!</p><div>Tweet this</div></div></div>					
8998	ger Mer		0.77033	1	2mo

### 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(Dropout(0.2))


# Segunda Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation='relu', kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# Tercera Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation='relu', kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# 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'])
```

9002	Belter Molina Guevara	0.77272	4	1s
------	-----------------------	---------	---	----

 Your Best Entry!  
Your most recent submission scored 0.77272, which is the same as your previous score. Keep trying!

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

```
# establecemos nuestra semilla aleatoria
tf.random.set_seed(42)

527] clasificador = Sequential()
# Primera Capa
clasificador.add(Dense(7, activation='relu', input_shape=(7,)), kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# Segunda Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation='relu', kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))


# Tercera Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation='relu', kernel_initializer='uniform'))
clasificador.add(Dropout(0.2))

# 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='sigmoid', kernel_initializer='uniform'))

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




529] # Entrenando nuestro modelo
clasificador.fit(X, y, batch_size=1, epochs=400, validation_split=0.2, verbose=0)
```

4217	Belter Molina Guevara	0.77751	5	1s
------	-----------------------	---------	---	----

 Your Best Entry!  
Your most recent submission scored 0.77751, which is an improvement of your previous score of 0.77272. Great job! [Tweet this](#)

4218	Mitsuaki Ogura	0.77511	10	1mo
------	----------------	---------	----	-----

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

4222	Belter Molina Guevara		0.77751	6	4h
 Your Best Entry! Your submission scored 0.71291, which is not an improvement of your previous score. Keep trying!					
4223	veleronie		0.77751	1	4h

```
clasificador = Sequential()
# Primera Capa
clasificador.add(Dense(7, activation='relu', input_shape=(7,), kernel_initializer='random_normal'))
clasificador.add(Dropout(0.2))

# Segunda Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation='relu', kernel_initializer='random_normal'))
clasificador.add(Dropout(0.2))

# Tercera Capa con 50 neuronas y una activacion relu
clasificador.add(Dense(50, activation='relu', kernel_initializer='random_normal'))
clasificador.add(Dropout(0.2))

# Capa de Salida
# utilizamos 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.