

```
features = ['B_R2', 'B_thrustBm', 'B_thrust0m', 'B_cosTBT0', 'B_daughterAngle', 'B_p', 'B_KSFWV_hso10', 'B_KSFWV_hso12',  
           'B_KSFWV_hso14', 'B_foxWolframR4', 'B_thrustAxisCosTheta', 'B_daughterAngle_CMS', 'B_daughterDiff_CMS_p',  
           'B_CMS_cosTheta', 'K_p', 'pi_CMS_pt', 'pi_CMS_p', 'pi_CMS_E']  
target = ['B_isContinuumEvent']
```

```
def build_model(n_hidden=3, n_neurons=128, learning_rate=0.001, input_shape=[len(features)]):  
    model = keras.models.Sequential()  
    model.add(keras.layers.InputLayer(input_shape=input_shape))  
  
    for layer in range(n_hidden):  
        model.add(keras.layers.Dense(n_neurons, activation="relu"))  
  
    model.add(keras.layers.Dense(1, activation="sigmoid"))  
    optimizer = keras.optimizers.Adam(learning_rate=learning_rate) # Change the optimizer here, if you don't want to use SGD  
    model.compile(loss="binary_crossentropy", optimizer=optimizer, metrics=["accuracy"])  
  
    return model
```

```
# train and save model  
reduce_lr = keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=0.00001)  
my_callbacks = [keras.callbacks.EarlyStopping(patience=10), reduce_lr]  
training_history = model.fit(X_train, y_train, epochs=100, validation_split=0.2, callbacks=[my_callbacks], verbose=2)  
model.save("cs_exercise_summer_school_2023")
```