NN Model Design

# Objective:

Develop a NN that is at least as good as a logistic model using cosine similarity as input (AUC = 0.891).

# Procedure:

## Start with a single, small dense layer. Increase as follows:

12, 16, 30, 60, 100, 200, 300, 600, 1000

# Model 1: Dense(12, activation="relu")

lstm\_1 = common\_embed(input\_1)

lstm\_2 = common\_embed(input\_2)

d1 = Dense(size, activation="relu")

vector\_11n = d1(lstm\_1)

vector\_11n = Flatten()(vector\_11n)

vector\_12n = d1(lstm\_2)

vector\_12n = Flatten()(vector\_12n)

conc = Concatenate(axis = -1)([vector\_11n, vector\_12n])

x = Dense(size, activation="relu", name='conc\_layer')(conc)

#x = Dropout(0.2)(x)''

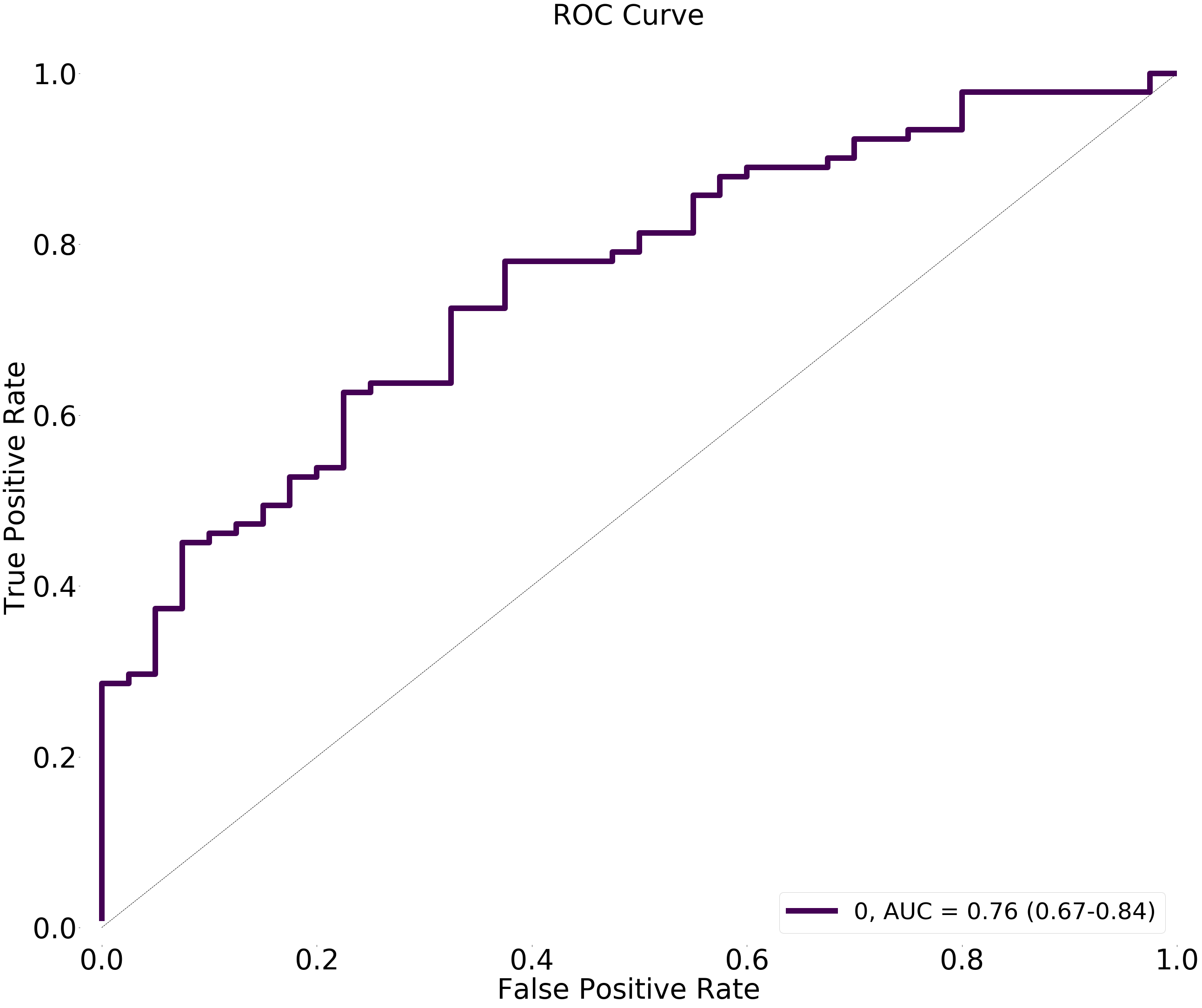
out = Dense(1, activation="sigmoid", name = 'out')(x)

model = Model([input\_1, input\_2], out)

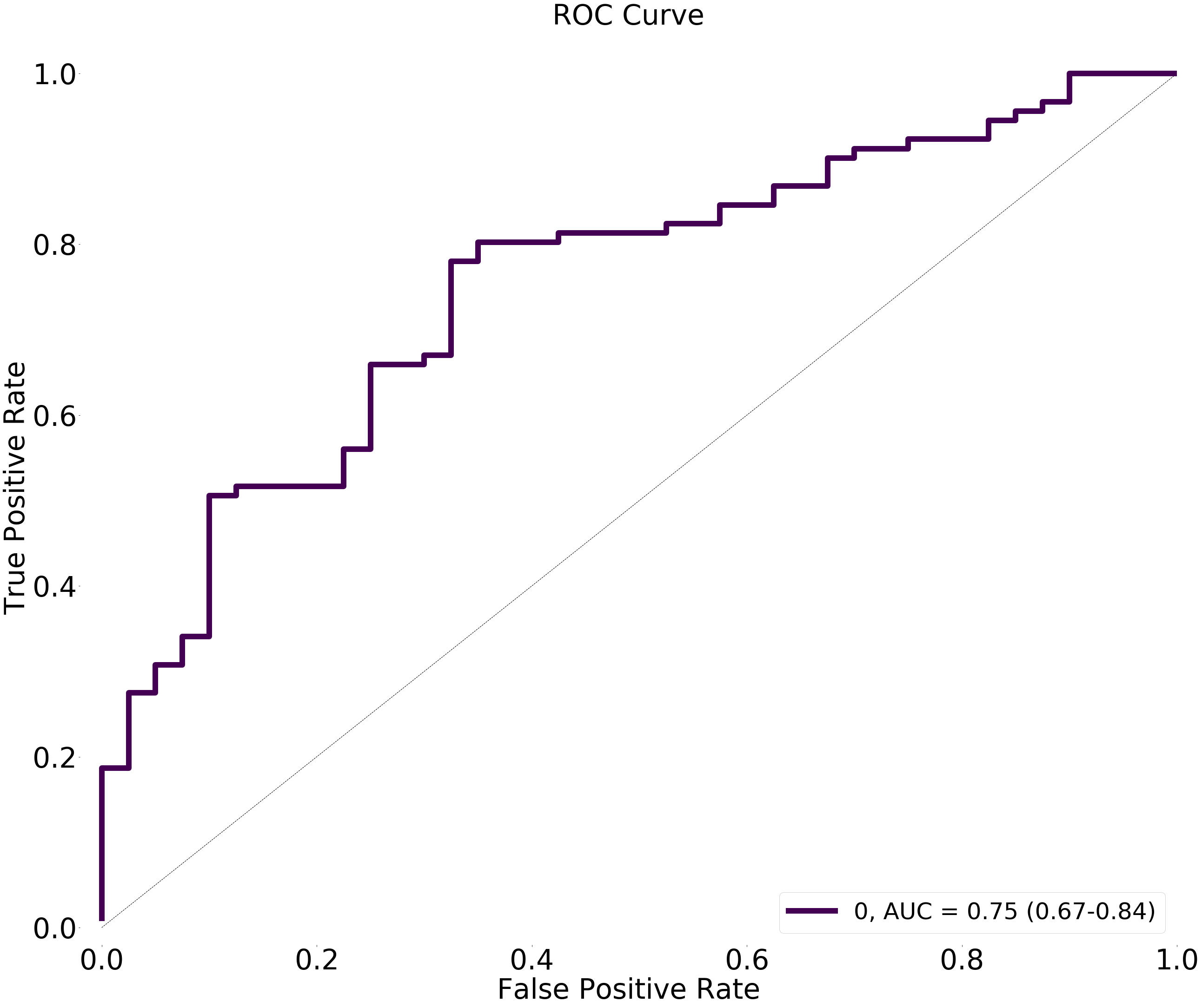
model.compile(loss= "binary\_crossentropy", metrics=['acc', keras.metrics.AUC()],

optimizer = Adam()) # Tried Adadelta as well

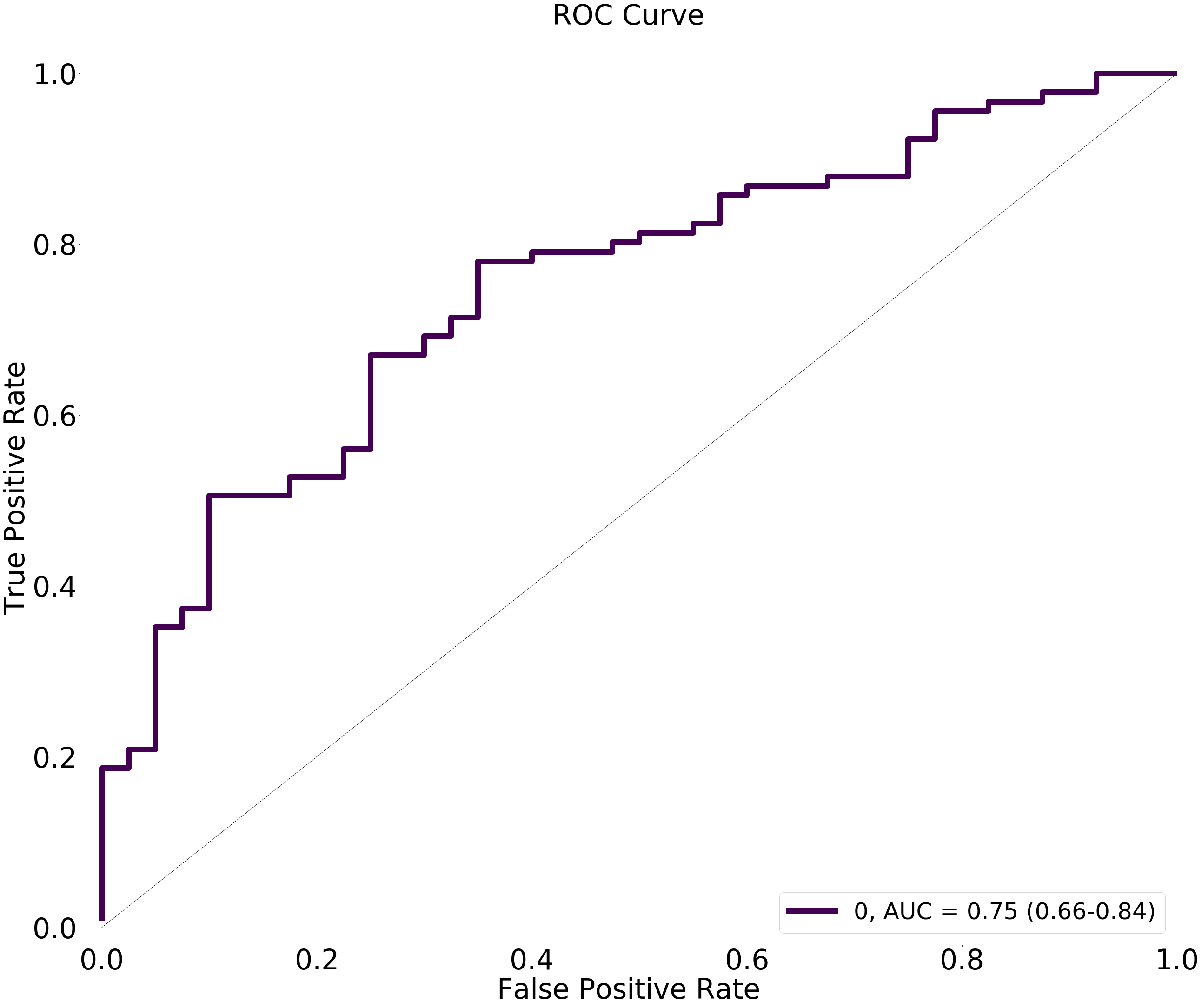
## Results:



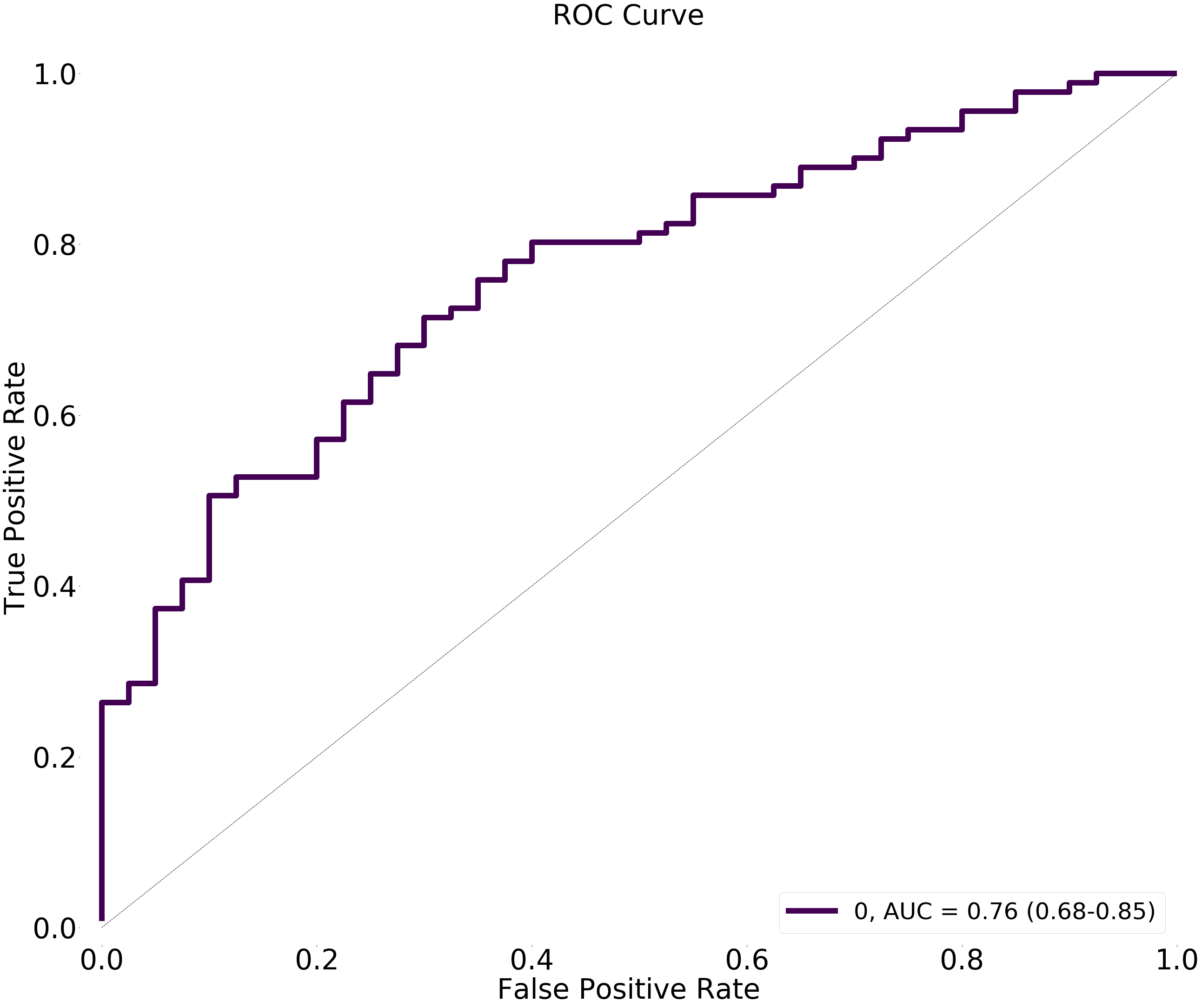
# Model 2: Dense(16, activation="relu")



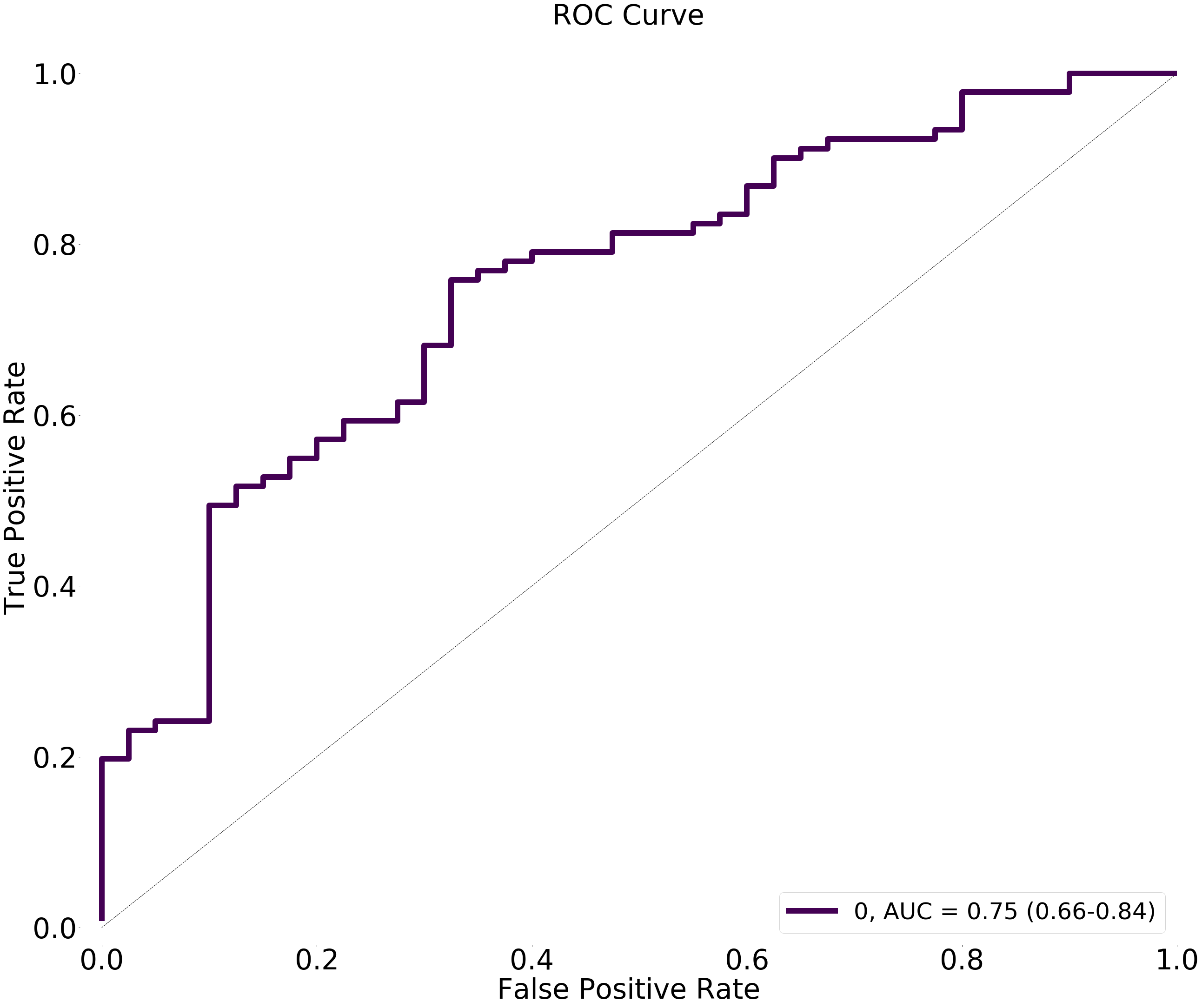
# Model 3: Dense(30, activation="relu")



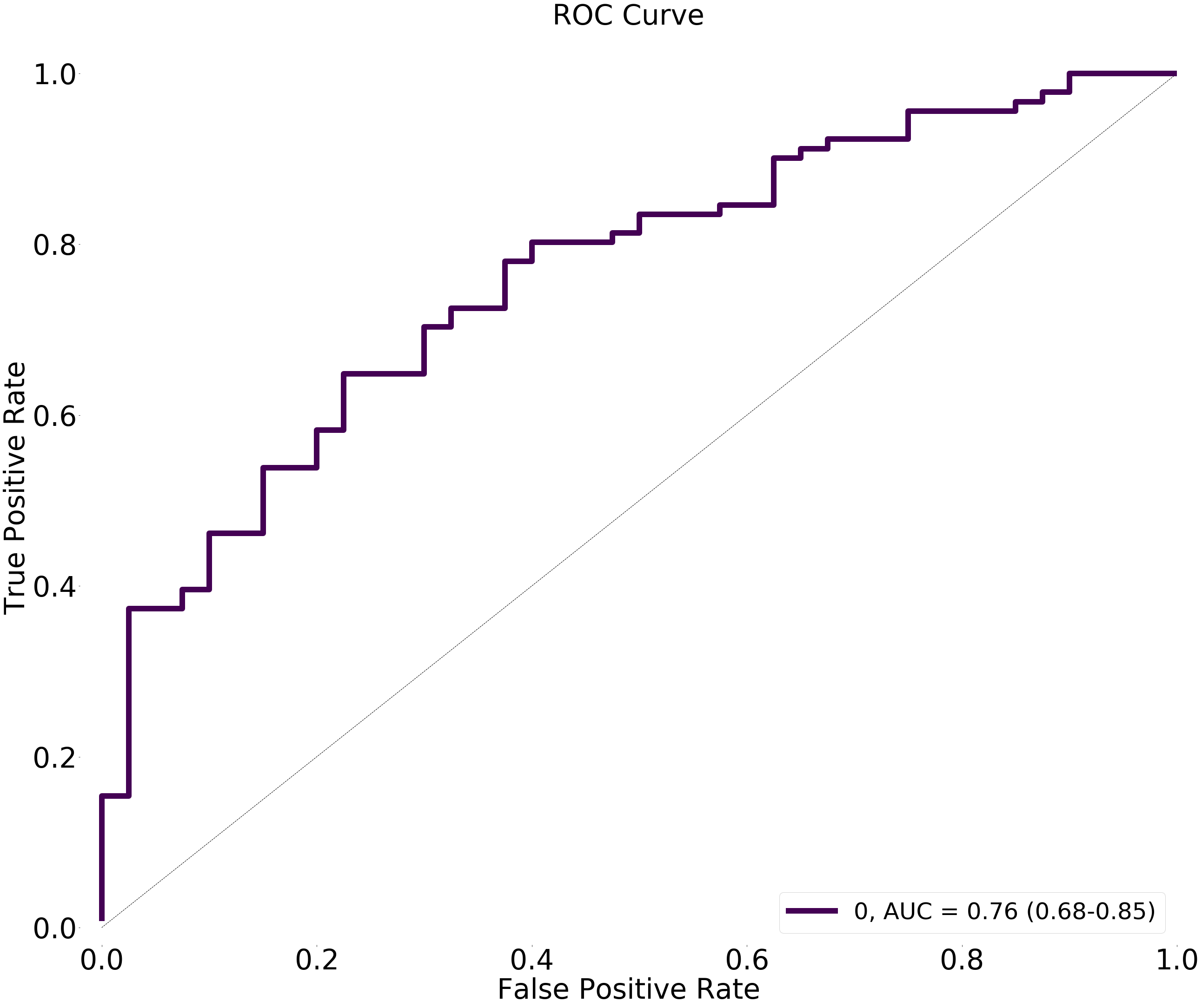
# Model 4: Dense(60, activation="relu")



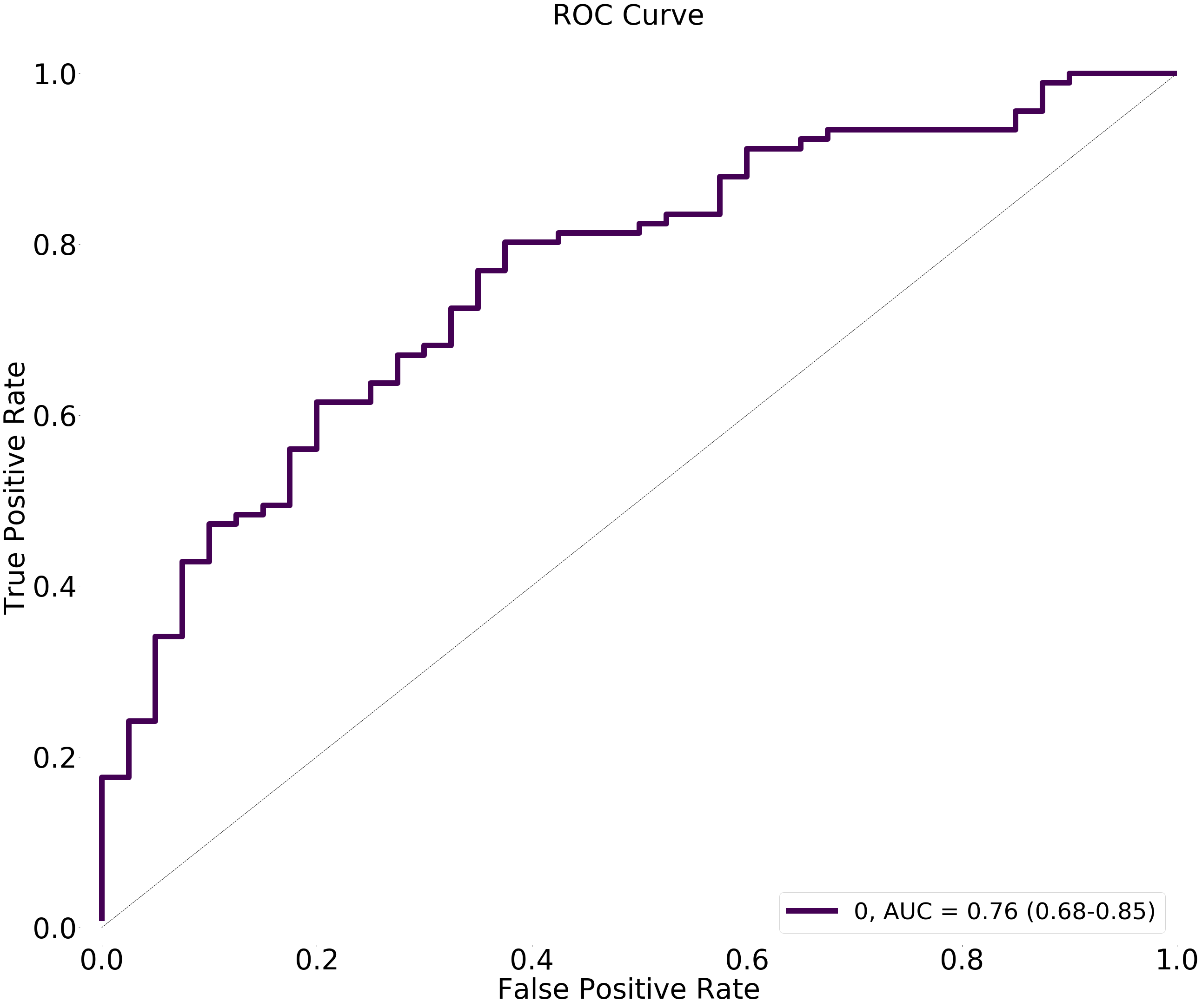
# Model 5: Dense(100, activation="relu")



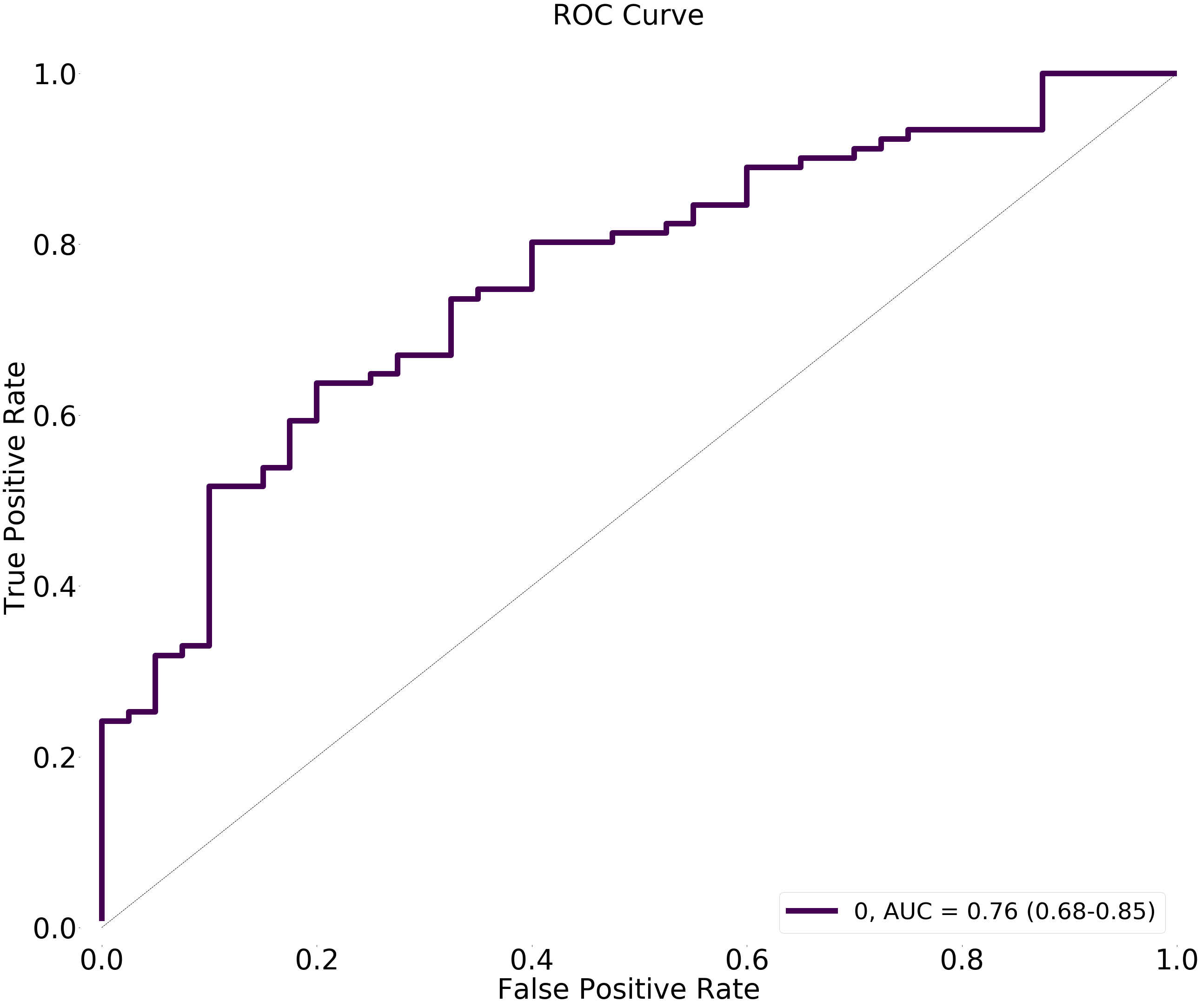
# Model 6: Dense(200, activation="relu")



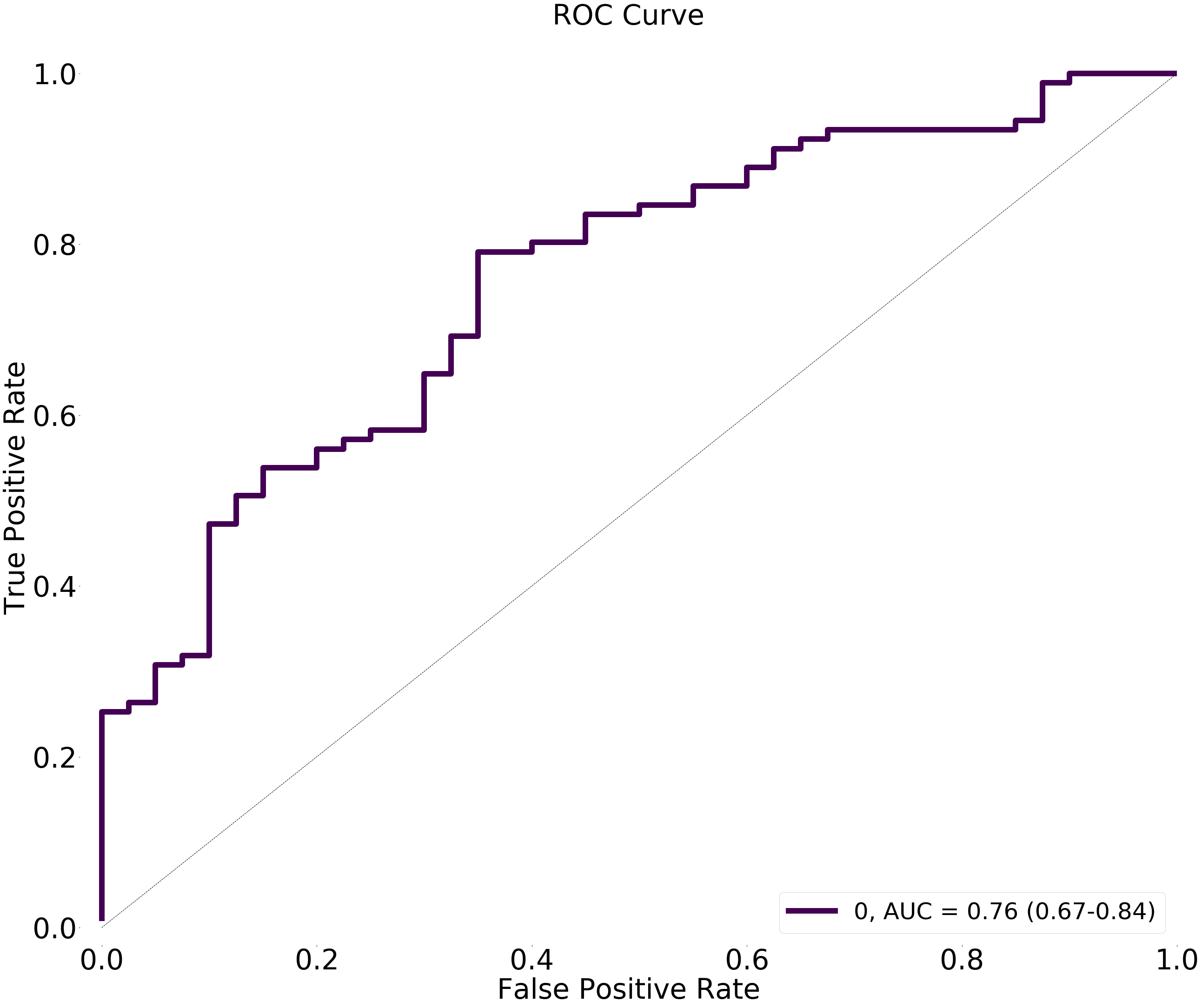
# Model 7: Dense(300, activation="relu")



# Model 8: Dense(600, activation="relu")



# Model 9: Dense(1000, activation="relu")

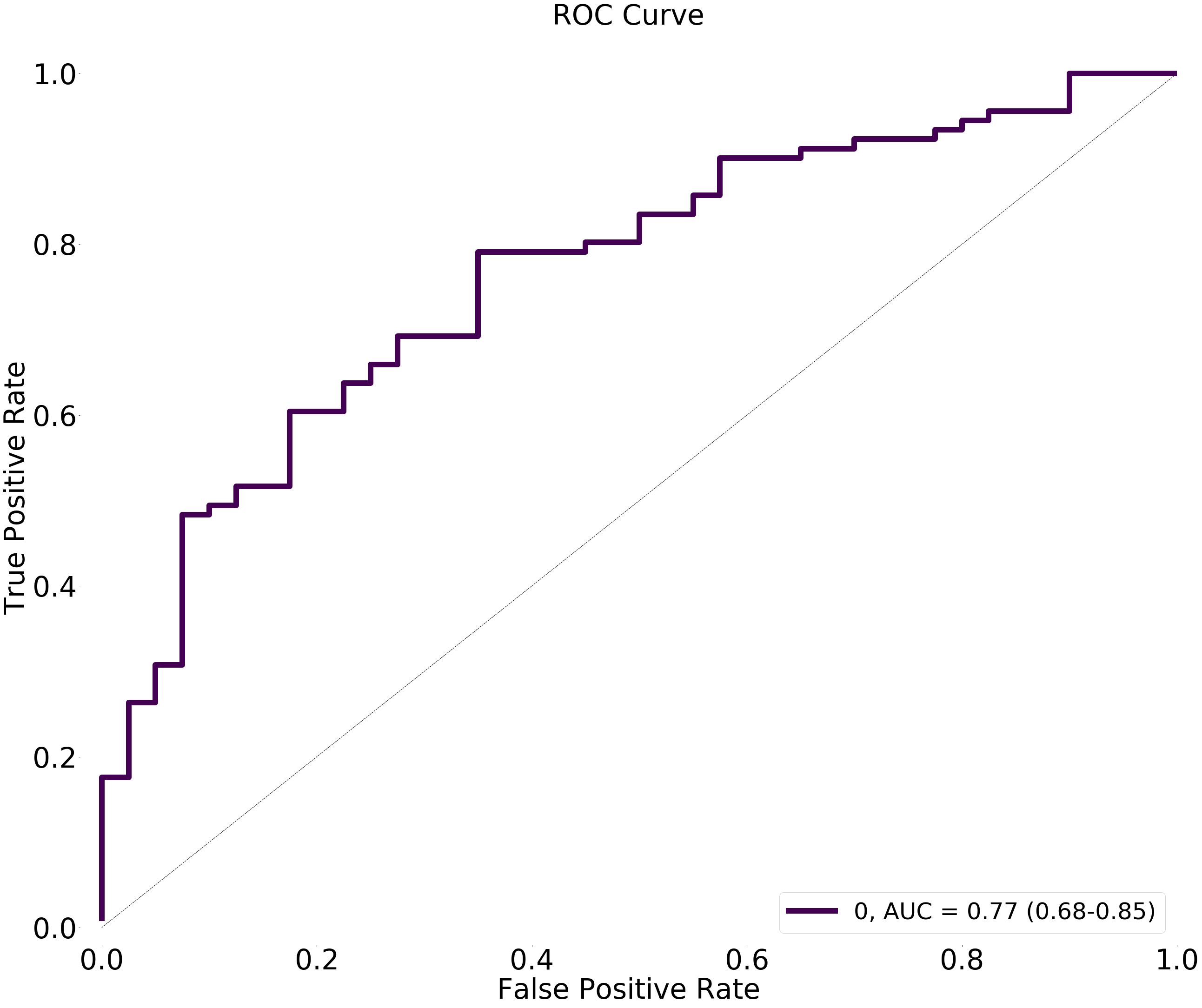


# Conclusion part 1:

increasing the size of the 2 layer from 12 to 1000 makes no diff.

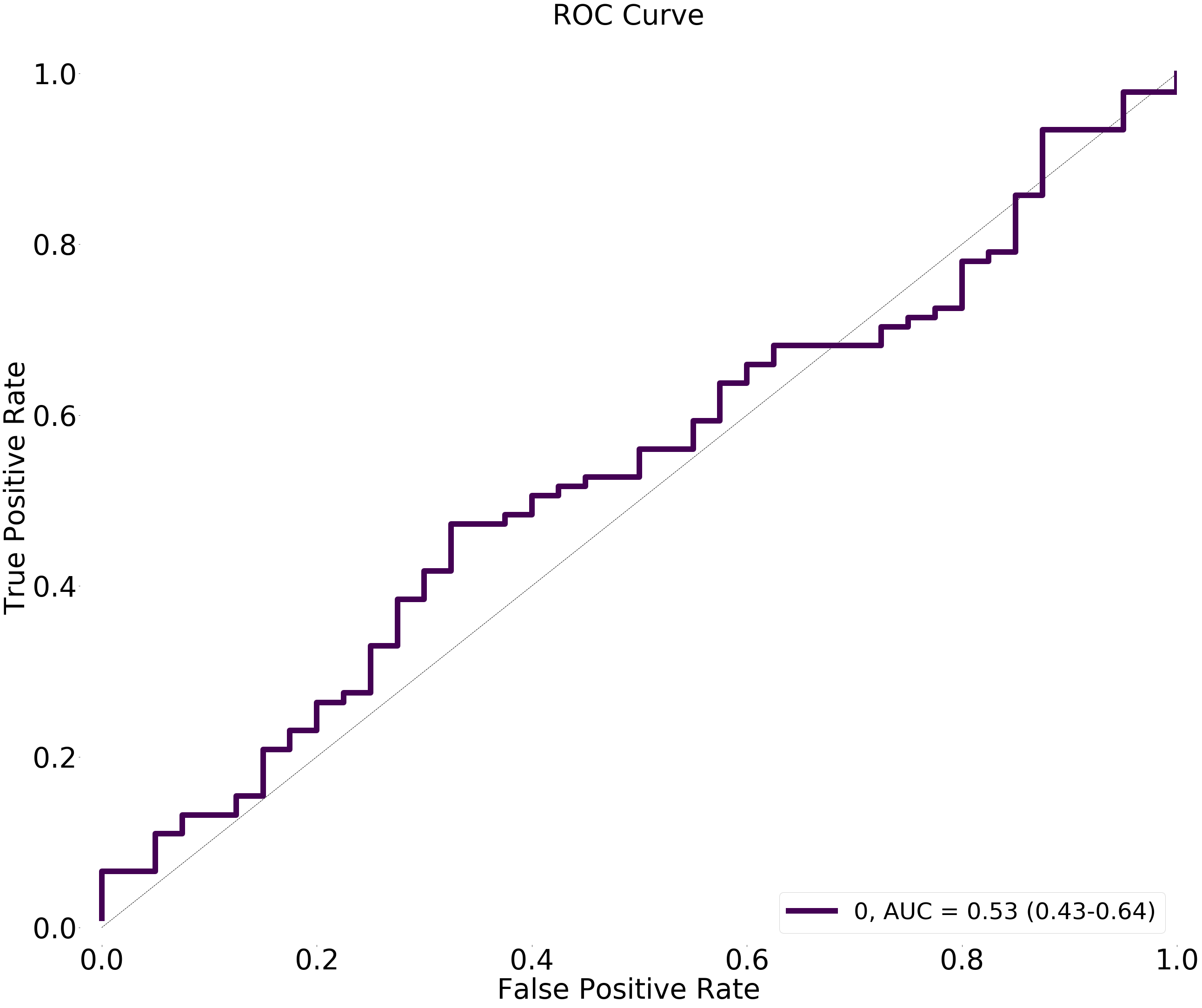
# Model 10: Dense(1000, activation="tanh")

Figure size 612x252 with 0 Axes>



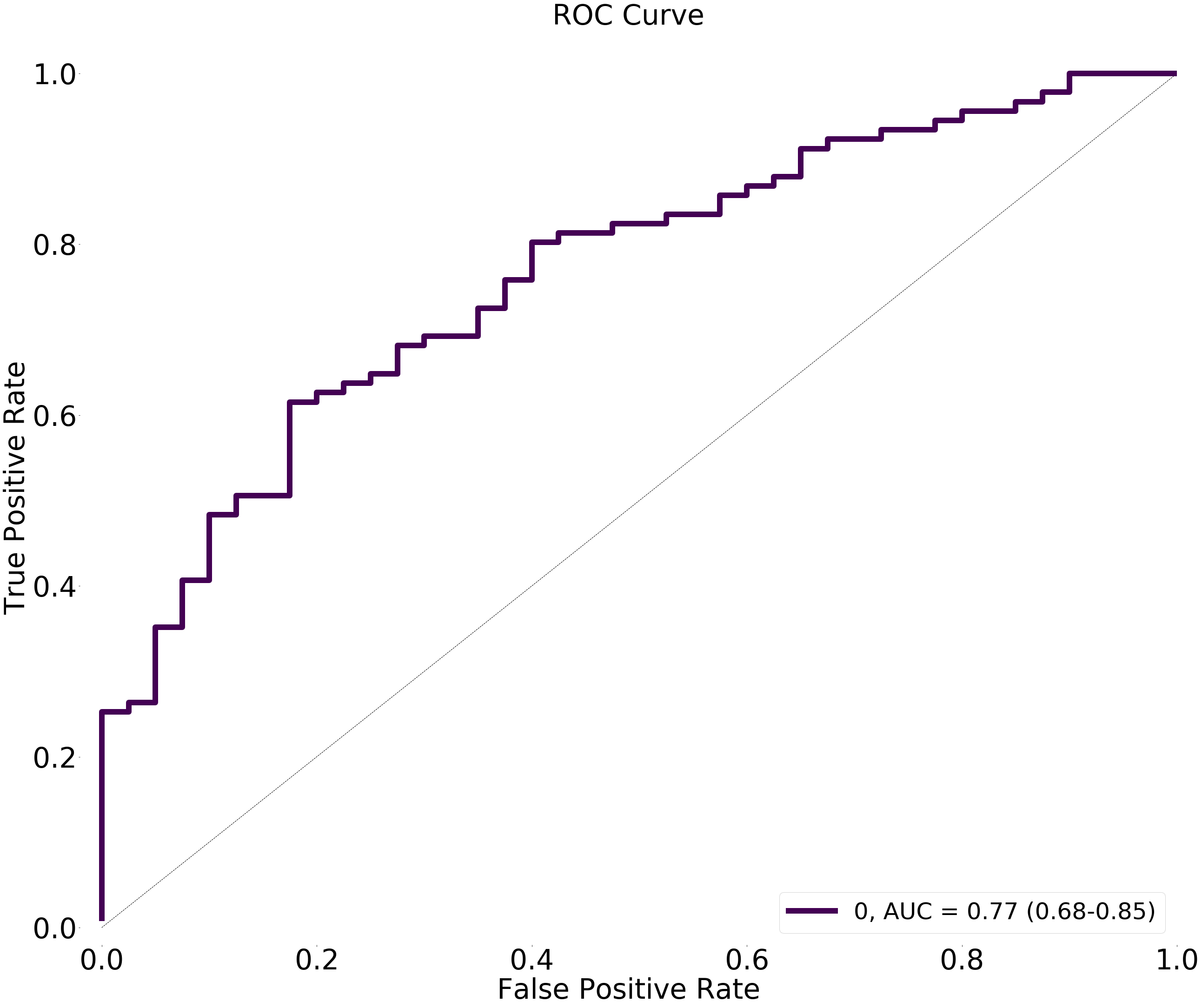
# Model 11: Dense(12, activation\_l1="tanh")

>

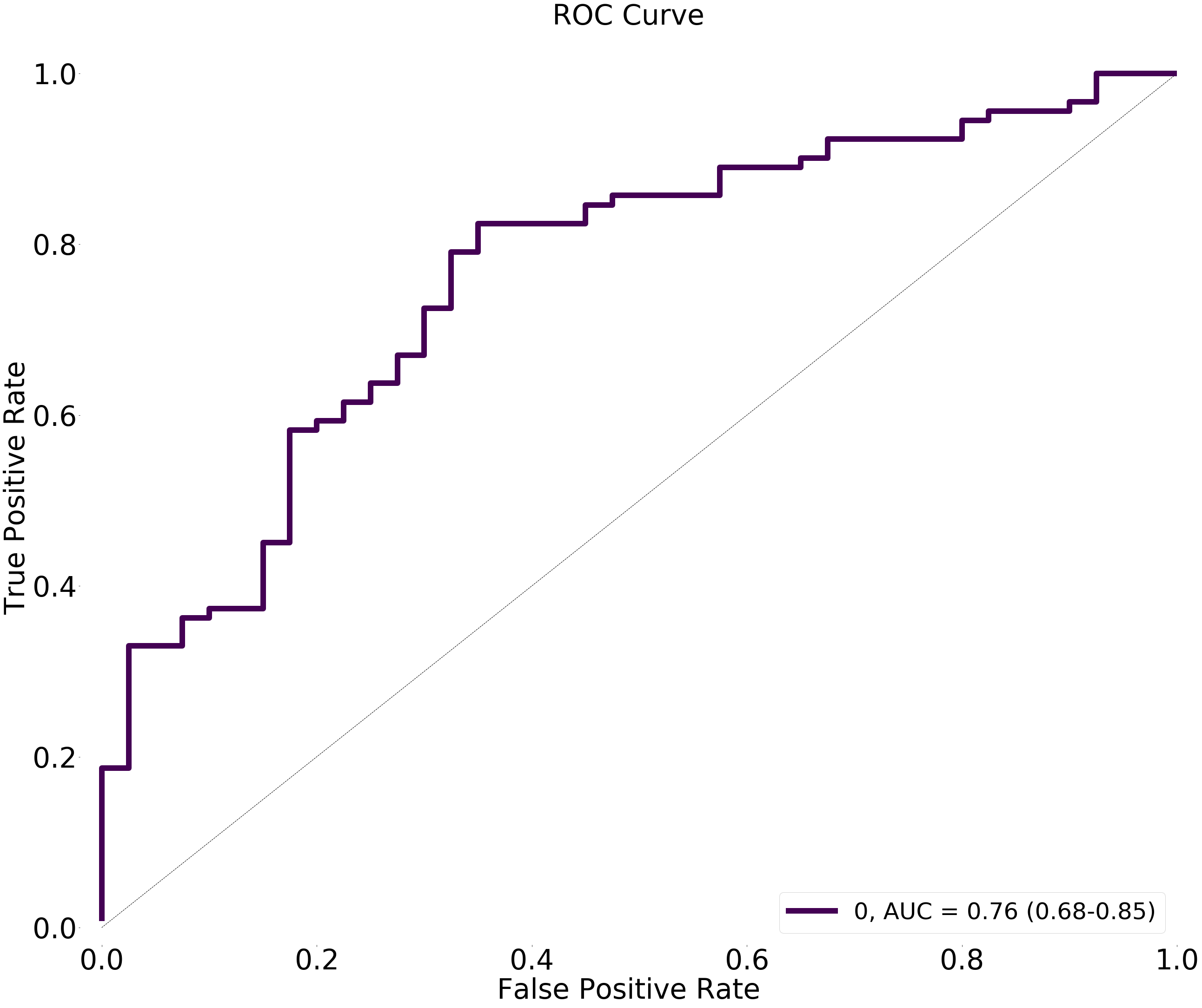


# Model 12: Dense(1000, activation\_l1="tanh")

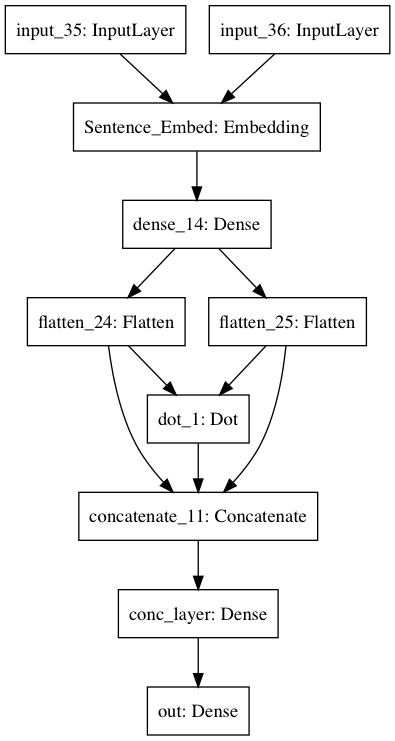
>

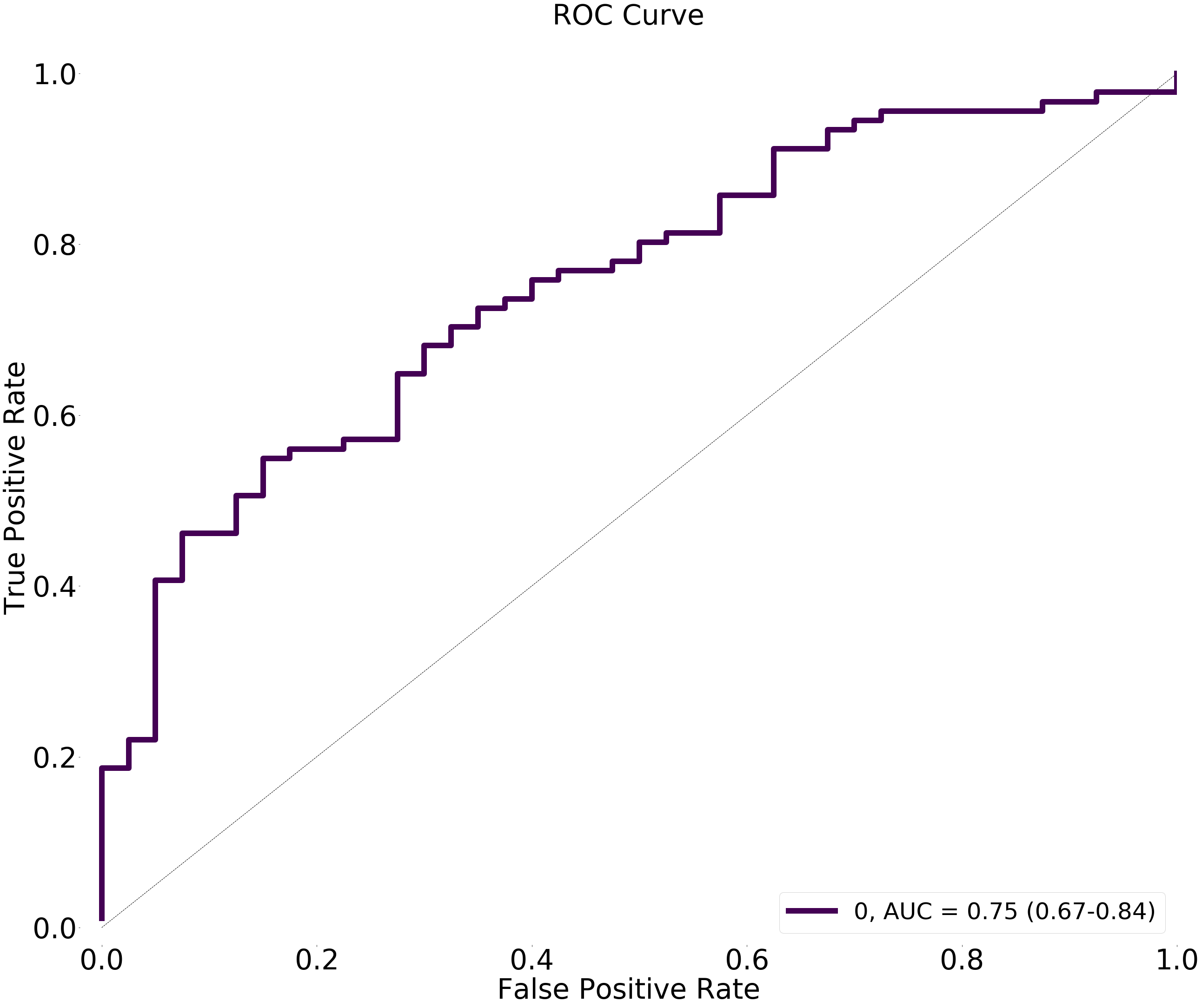


# Model 12: Dense(1000, activation\_l1="tanh", activation\_l2="tanh")

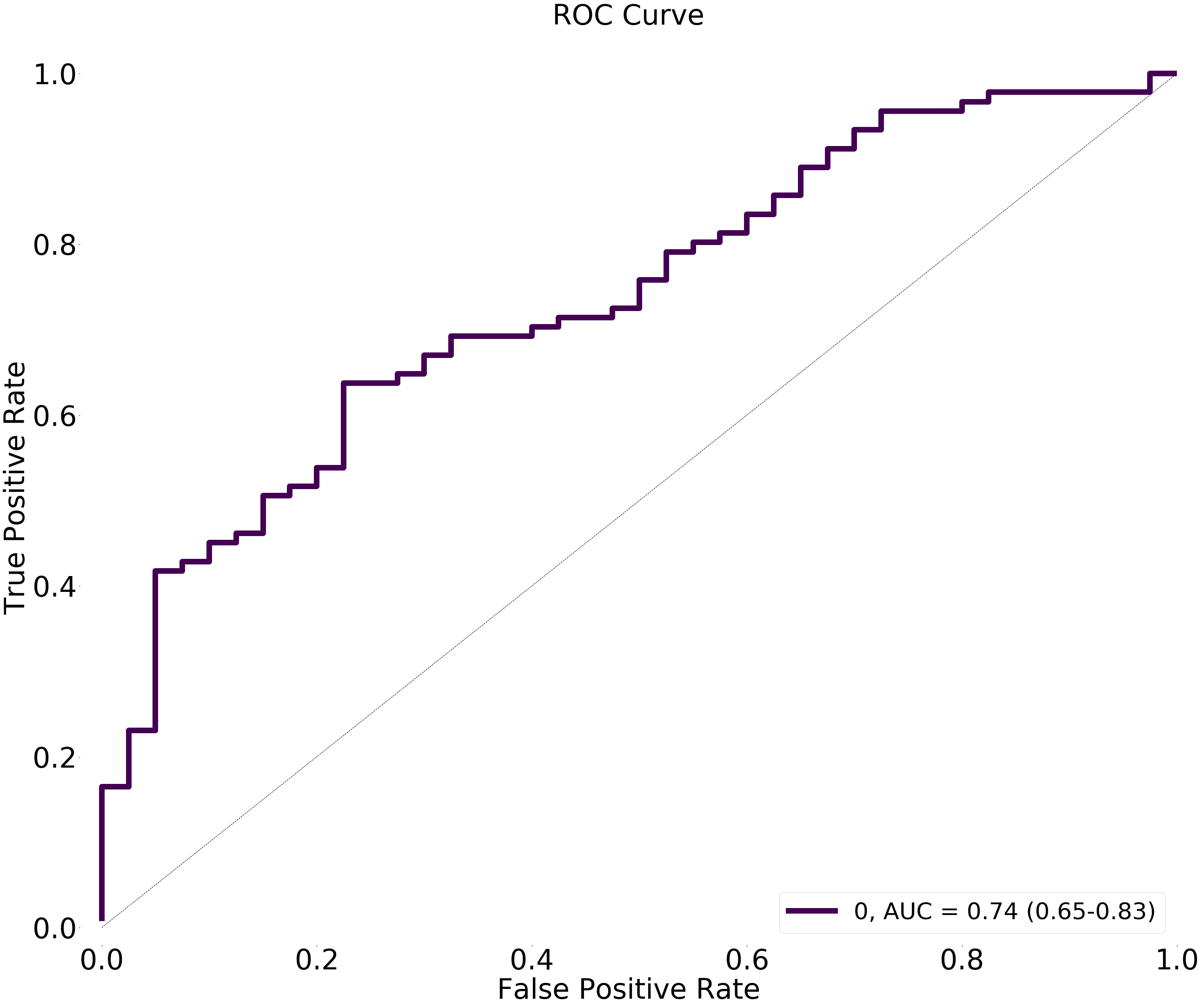


# Model 12: Dense(13, activation\_l1="relu", activation\_l2="relu"), Dot

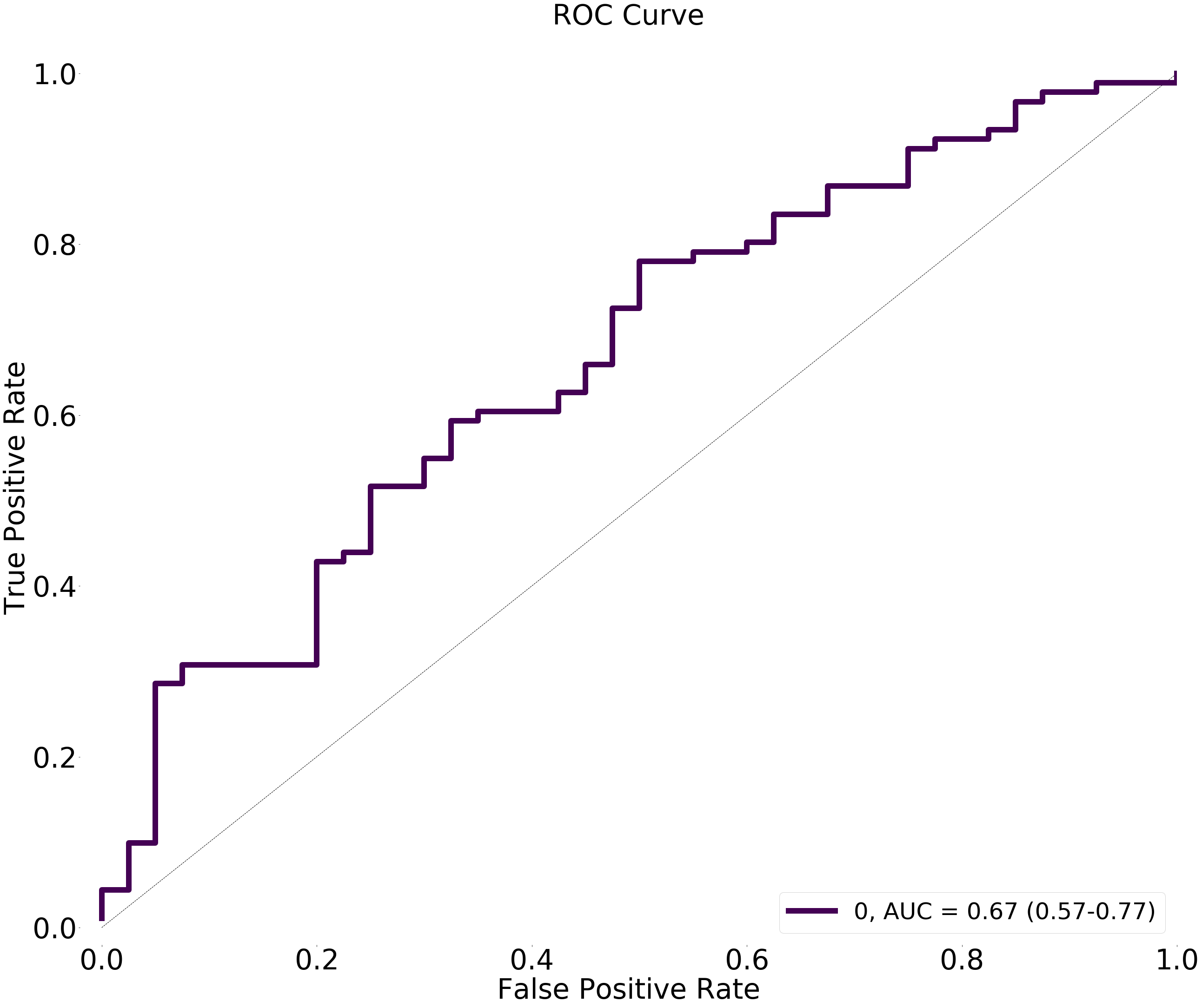




# Model 13: Dense(12, activation\_l1="relu", activation\_l2="relu"), Dot, BatchNorm, Adam(lr = 0.001)



# Model 14: Dense(12, activation\_l1="relu", activation\_l2="relu"), Dot, LayerNorm, Adam(lr = 0.001)



# Model 15: Dense(13, activation\_l1="relu", activation\_l2="relu"), Dot, LayerNorm, Adam(lr = 0.001)