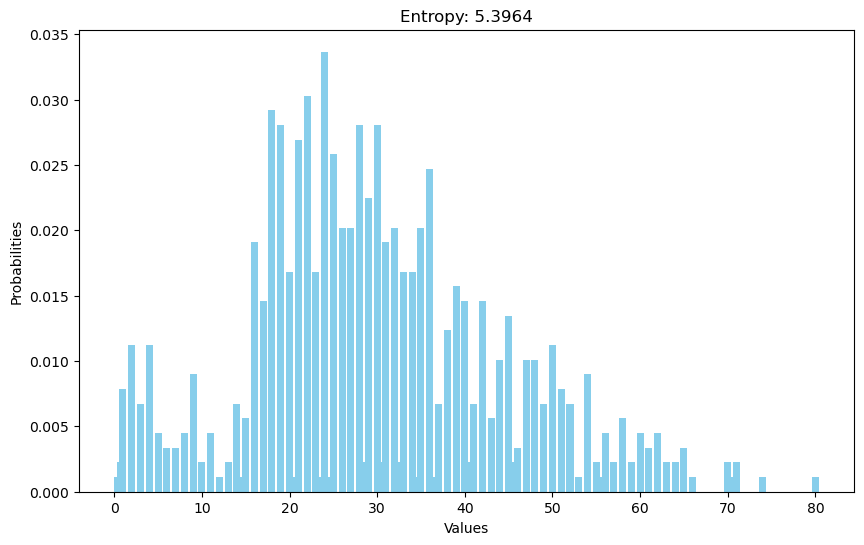
Decision Tree?

Logistic regression

Gradient descent



Entropy (‘Age’ column)

1. Use logistic regression to predict the survivors
2. Run diagnostics on my results

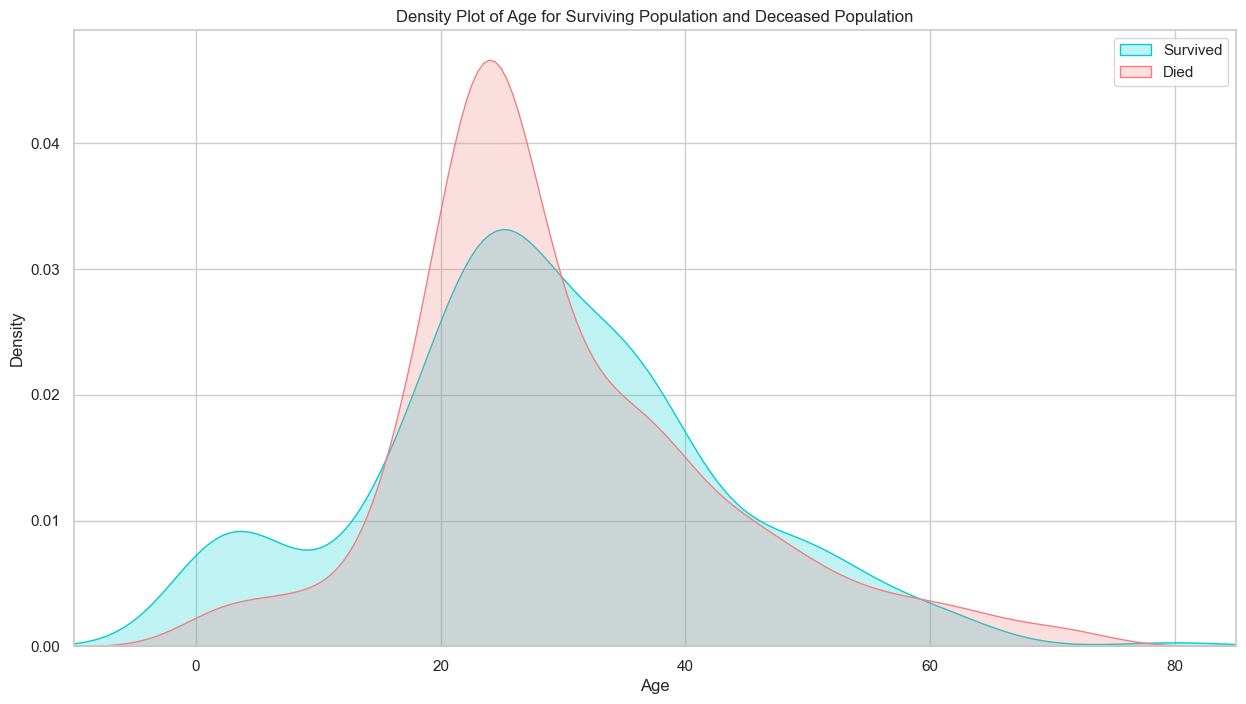
Logistic regression - using scikit learn

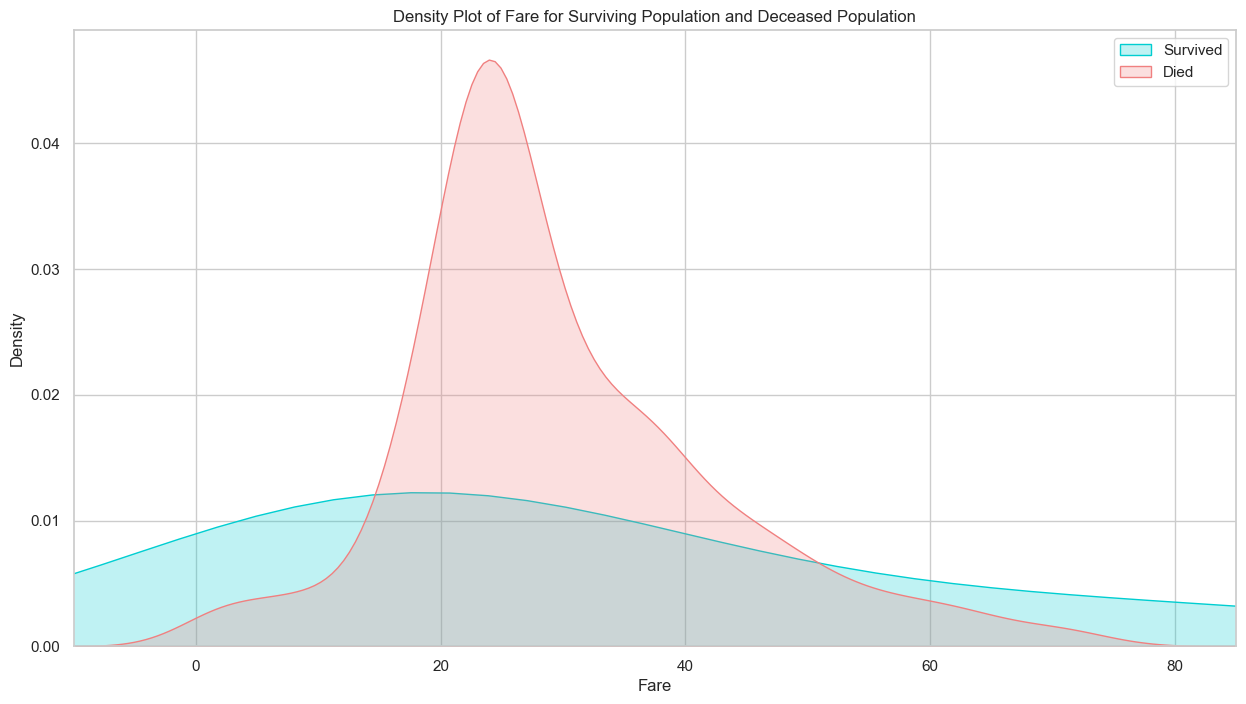
1. Data preparation
   1. Load the data (Done)
   2. Handle the missing values (Done)
   3. Feature engineering
   4. Encode categorical variables
2. Train-test split

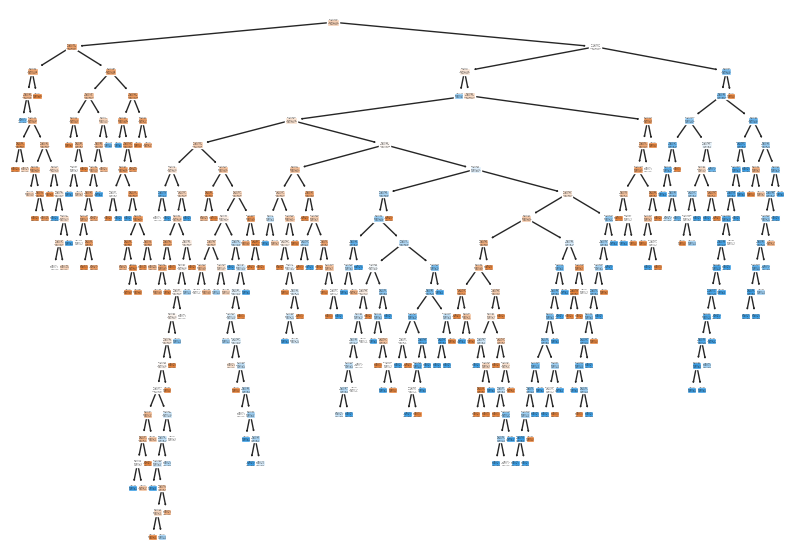
This week:

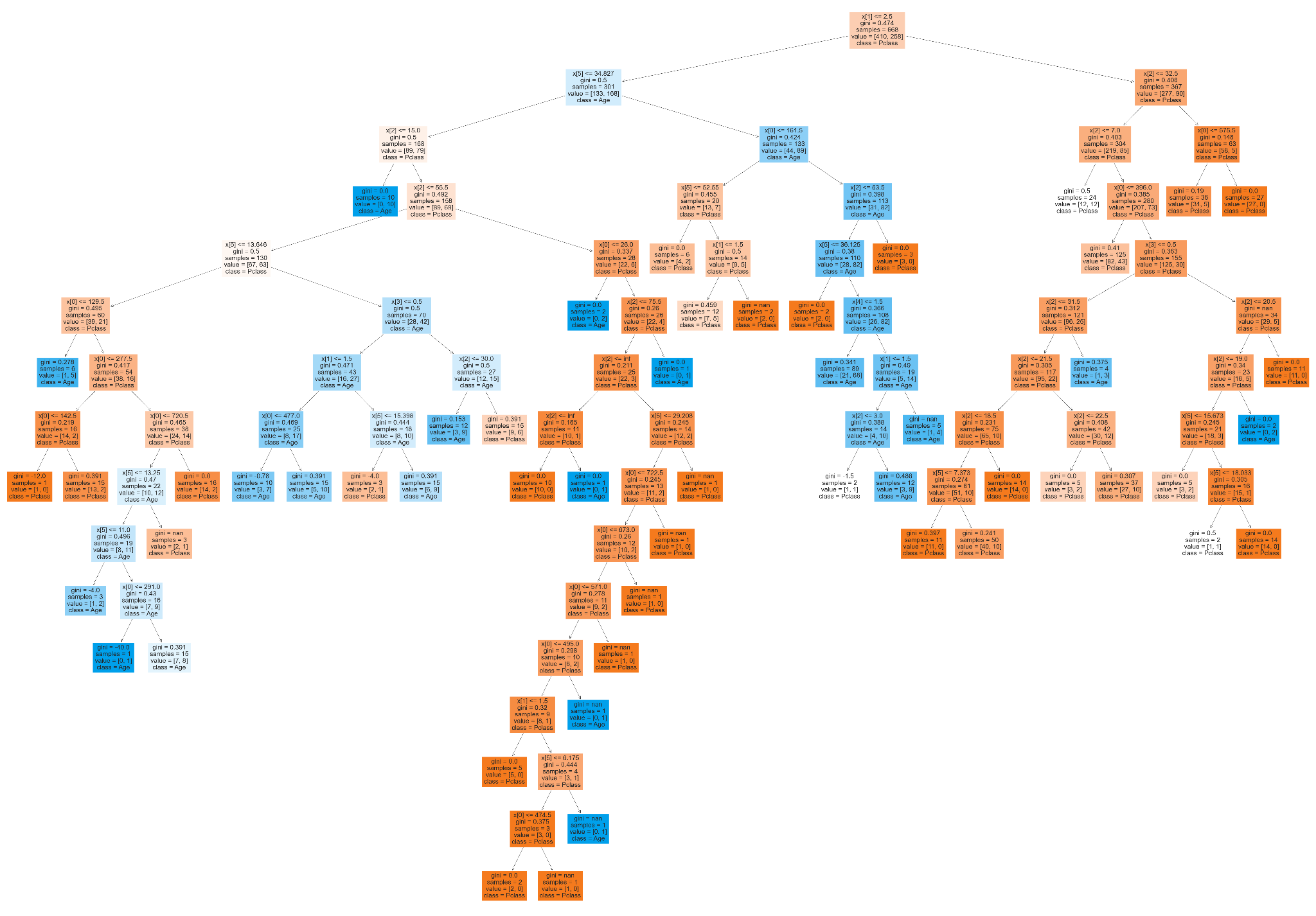
**Implemented logistic regression in python using pandas and numpy**

Graphs:









-fit/predict → tested on the test\_dataset and the results were disappointing

cols=["Age", "Fare","Pclass"]

train\_data = pd.read\_csv('/Users/annikaseo-yeonkim/Desktop/MongooseAI/1주차/titanic/datasets/train (1).csv')

test\_data = pd.read\_csv('/Users/annikaseo-yeonkim/Desktop/MongooseAI/1주차/titanic/datasets/test.csv')

X\_DT=train\_data[cols]

Y\_DT=train\_data['Survived']

X2= test\_data[cols]

# Y\_d = test\_data['Survived']

tree1.fit(X\_DT, Y\_DT)

# tree1.predict(X\_DT)

newList = list(tree1.predict(X2))

newList2 = list(Y\_DT)

print(newList)

false\_neg = 0

false\_pos = 0

for i in range(len(newList)):

if newList[i] != newList2[i]:

if newList[i] == 0:

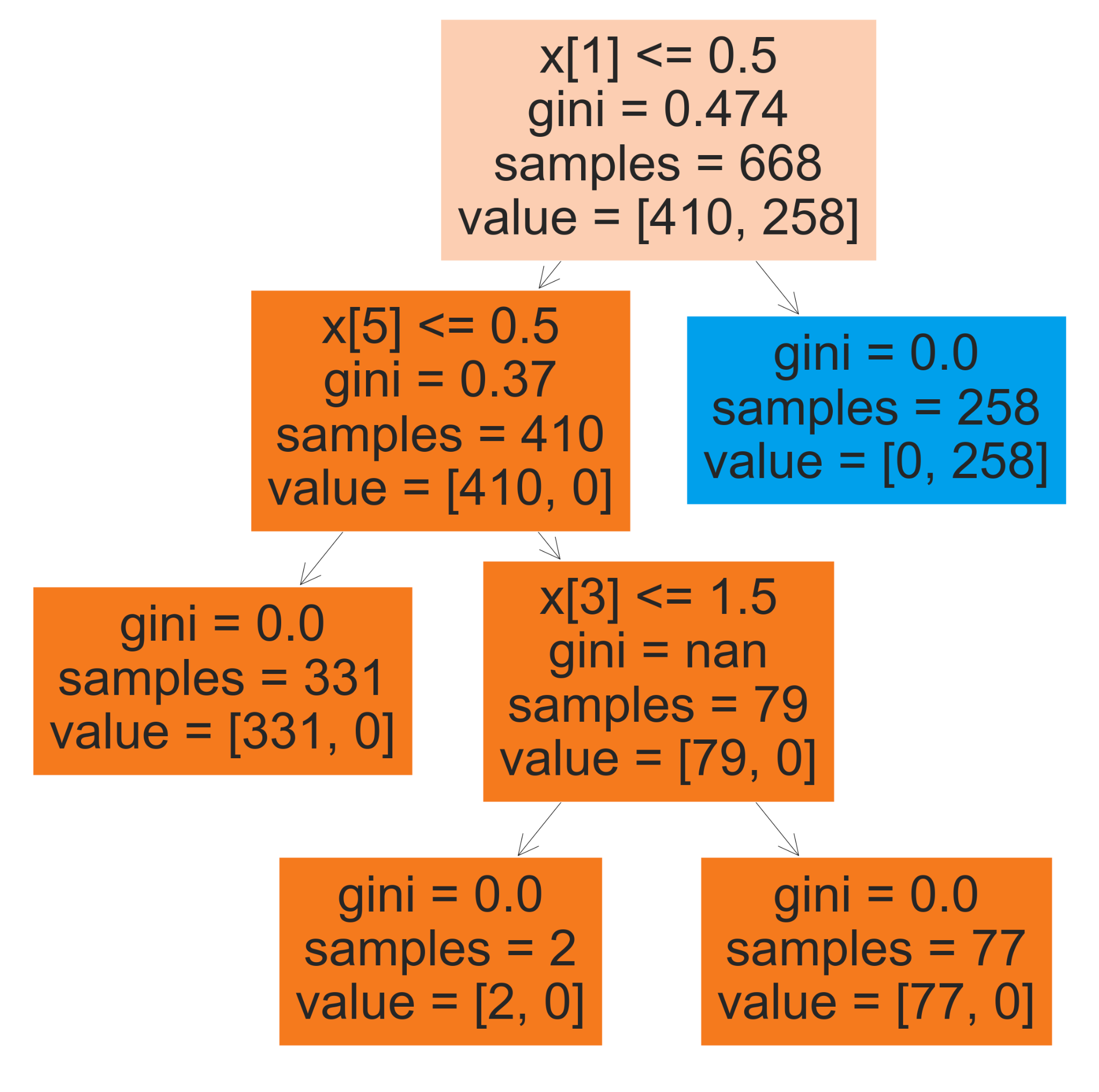
false\_neg+=1

elif newList[i] == 1:

false\_pos+=1

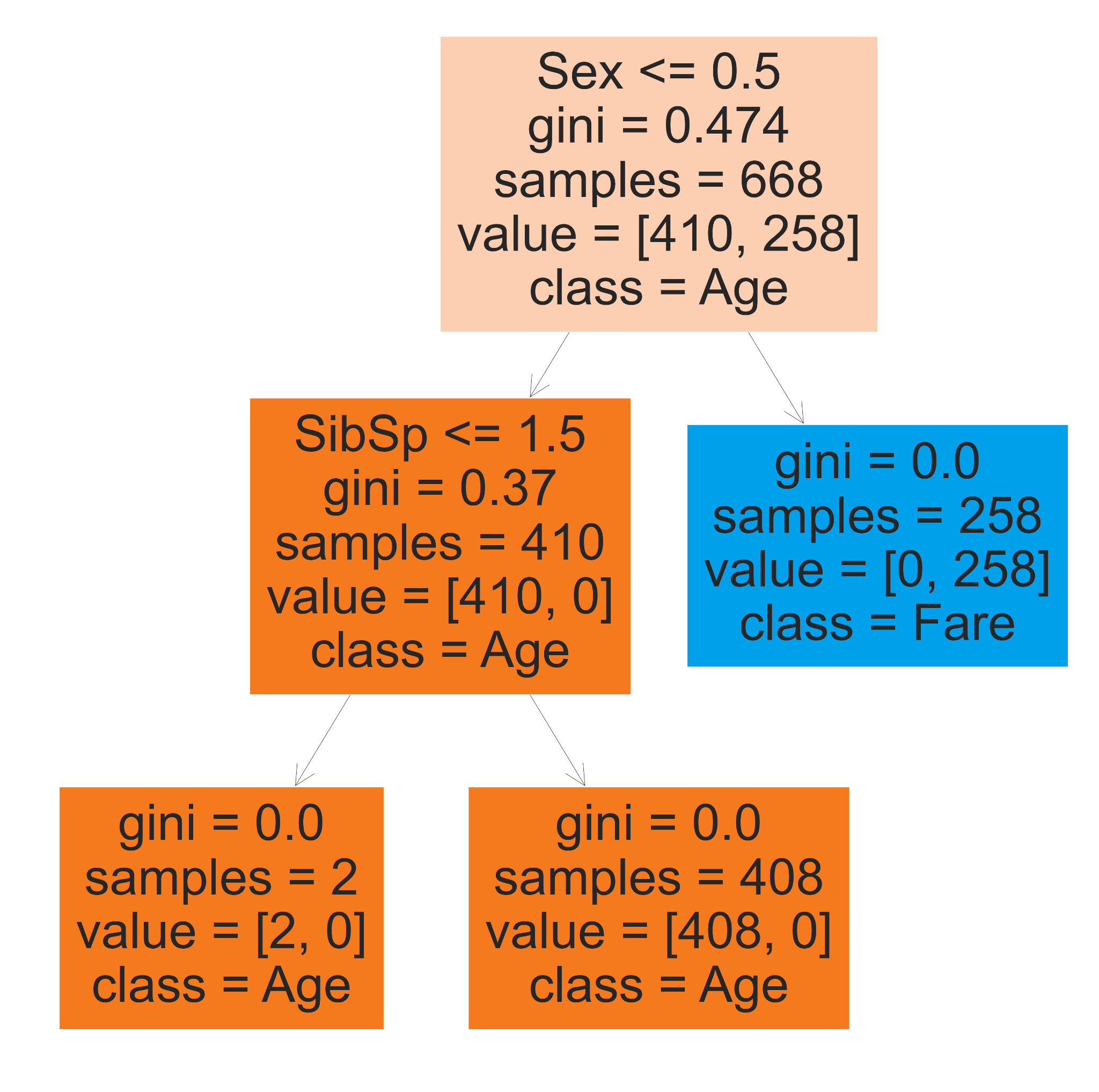
print(f"there are {false\_neg} false negatives and {false\_pos} false positives")

print(f"total predictions: {len(newList)}")



(truncated)

Need to prune my tree



Confusion Matrix - how well our model works

[[445 104]

[151 191]]