#Source Model

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential,Model

from sklearn.metrics import accuracy\_score

from keras.layers import Dense

from sklearn.metrics import confusion\_matrix

#Reading the dataset

dataset = pd.read\_csv(r'C:\Users\saikr\Desktop\Covid Dataset.csv')

print(dataset.head(3))

number=LabelEncoder()

for col in dataset.columns:

dataset[col]=number.fit\_transform(dataset[col].astype('str'))

print(dataset.head(3))

#feature transformation

dataset['kidney disease']=dataset['Gastrointestinal']

dataset['travel-history']=dataset['Abroad travel']+dataset['Contact with COVID Patient']+dataset['Attended Large Gathering']+dataset['Visited Public Exposed Places']

dataset['safety']=dataset['Wearing Masks']+dataset['Sanitization from Market']

i=2

for i in range(5):

dataset['travel-history'].replace(to\_replace=i,value=1,inplace=True)

dataset['safety'].replace(to\_replace=1,value=2,inplace=True)

dataset['travel-history']=dataset['travel-history']-dataset['safety']

j=-2

for j in range(0):

dataset['travel-history'].replace(to\_replace=j,value=0,inplace=True)

dataset['lung disease']=dataset['Asthma']+dataset['Chronic Lung Disease']

dataset['lung disease'].replace(to\_replace=2,value=1,inplace=True)

dataset['aches-pains']=dataset['Headache']

dataset['Corona result']=dataset['COVID-19']

dataset=dataset.drop(columns=['COVID-19','Family working in Public Exposed Places','Gastrointestinal','safety','Abroad travel','Contact with COVID Patient','Asthma','Chronic Lung Disease','Attended Large Gathering','Visited Public Exposed Places','Wearing Masks','Sanitization from Market','Running Nose','Headache'])

print(dataset['travel-history'].value\_counts())

print(dataset.head(3))

#printing the column names

for col in dataset.columns:

print(col)

#training the model

X = dataset.values[:,0:12]

Y=dataset['Corona result']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,Y, test\_size=0.3)

model =Sequential()

model.add(Dense(8, input\_dim=X\_train.shape[1], activation='relu'))

model.add(Dense(4, activation='relu'))

model.add(Dense(2, activation='relu'))

model.add(Dense(1, activation='softmax'))

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=100,batch\_size=5435)

y\_pred = model.predict(X\_test)

c=confusion\_matrix(y\_test,y\_pred)

print(c)

print("Accuracy of source:",accuracy\_score(y\_test, y\_pred)\*100)

model.save('model.h5')

#Target model

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from sklearn.metrics import accuracy\_score

from keras.layers import Dense

from keras.models import load\_model

from sklearn.metrics import confusion\_matrix

#Reading dataset

dataset = pd.read\_excel(r'C:\Users\saikr\Desktop\COVID-19.xlsx')

print(dataset.head(3))

#Feature transformation

dataset.loc[dataset['body temperature']>98.6,'Fever']=1

dataset.loc[dataset['body temperature']<=98.6,'Fever']=0

dataset['Fever']=dataset['Fever'].astype(int)

print(dataset['Fever'].head(3))

dataset['Sore throat']=dataset['sour throat']

dataset['Breathing Problem']=dataset['breathing problem']

dataset['Fatigue']=dataset['drowsiness']

dataset['Heart Disease']=dataset['heart disease']

dataset['Diabetes']=dataset['diabetes']

dataset['Hyper Tension']=dataset['high blood pressue']

dataset['travel-history']=dataset['travel history to infected countries']

dataset['aches-pains']=dataset['Loss of sense of smell']

#Remove the comment if you wanna try with 3 labels

#dataset['Corona result'].replace(to\_replace=2,value=1,inplace=True)

dataset=dataset.drop(columns=['Sno','body temperature','sour throat','breathing problem','drowsiness','heart disease','diabetes','high blood pressue','Loss of sense of smell','travel history to infected countries','age','gender','weakness','pain in chest','stroke or reduced immunity','symptoms progressed','change in appetide'])

dataset=dataset.reindex(columns=['Breathing Problem','Fever','Dry Cough','Sore throat','Heart Disease','Diabetes','Hyper Tension','Fatigue','kidney disease','travel-history','lung disease','aches-pains','Corona result'])

for col in dataset.columns:

print(col)

X = dataset.values[:,0:12]

Y=dataset['Corona result']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,Y, test\_size=0.3)

#train model normally

model =Sequential()

model.add(Dense(8, input\_dim=X\_train.shape[1], activation='relu'))

model.add(Dense(4, activation='relu'))

model.add(Dense(1, activation='softmax'))

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=100,batch\_size=5435)

y\_pred = model.predict(X\_test)

c=confusion\_matrix(y\_test,y\_pred)

print(c)

print("Accuracy of source:",accuracy\_score(y\_test, y\_pred)\*100)

#training model with transfer learning

model =load\_model('model.h5')

#remove the comments if you want freezing layers

#model.layers[1].trainable=False

#model.layers[2].trainable=False

#model.layers[3].trainable=False

#model.layers[0].trainable=False

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=10,batch\_size=125)

y\_pred = model.predict(X\_test)

c=confusion\_matrix(y\_test,y\_pred)

print(c)

print("Accuracy with transfer learning :",accuracy\_score(y\_test, y\_pred)\*100)

#to print the graphs the code try this with tl model and without model to plot different accuracy graphs

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,Y, test\_size=0.3)

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_train, y\_train, test\_size=0.2, random\_state=1)

model.compile(loss='categorical\_crossentropy', optimizer='sgd', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=10,batch\_size=15,validation\_data=(X\_val, y\_val))

print(history.history.keys())

loss\_train = history.history['accuracy']

loss\_val = history.history['val\_accuracy']

epochs = range(1,11)

plt.plot(epochs, loss\_train, 'g', label='Training accuracy')

plt.plot(epochs, loss\_val, 'b', label='validation accuracy')

plt.title('Training and Validation accuracy with model')

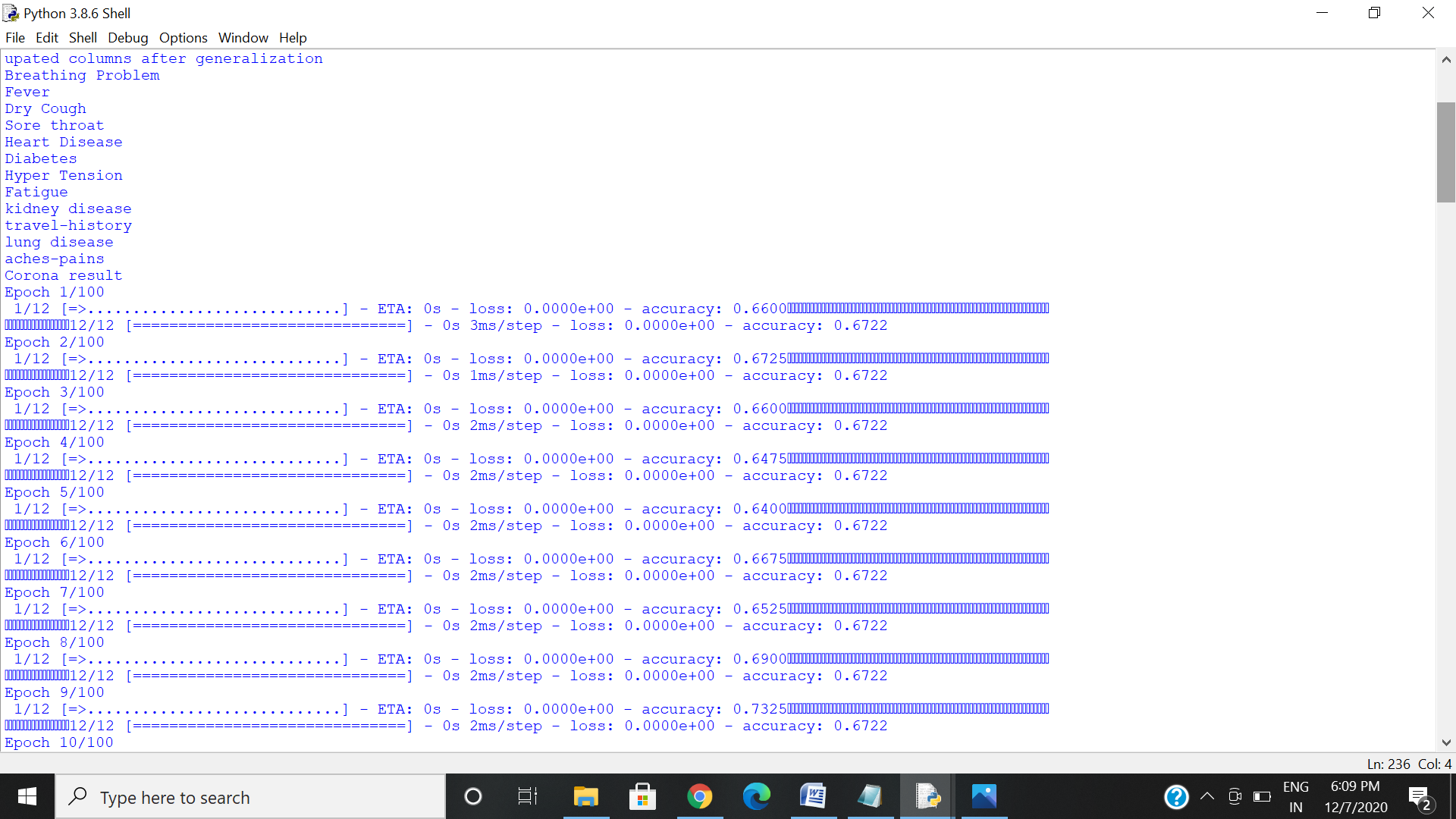
plt.xlabel('Epochs')

plt.ylabel('Accuracy')

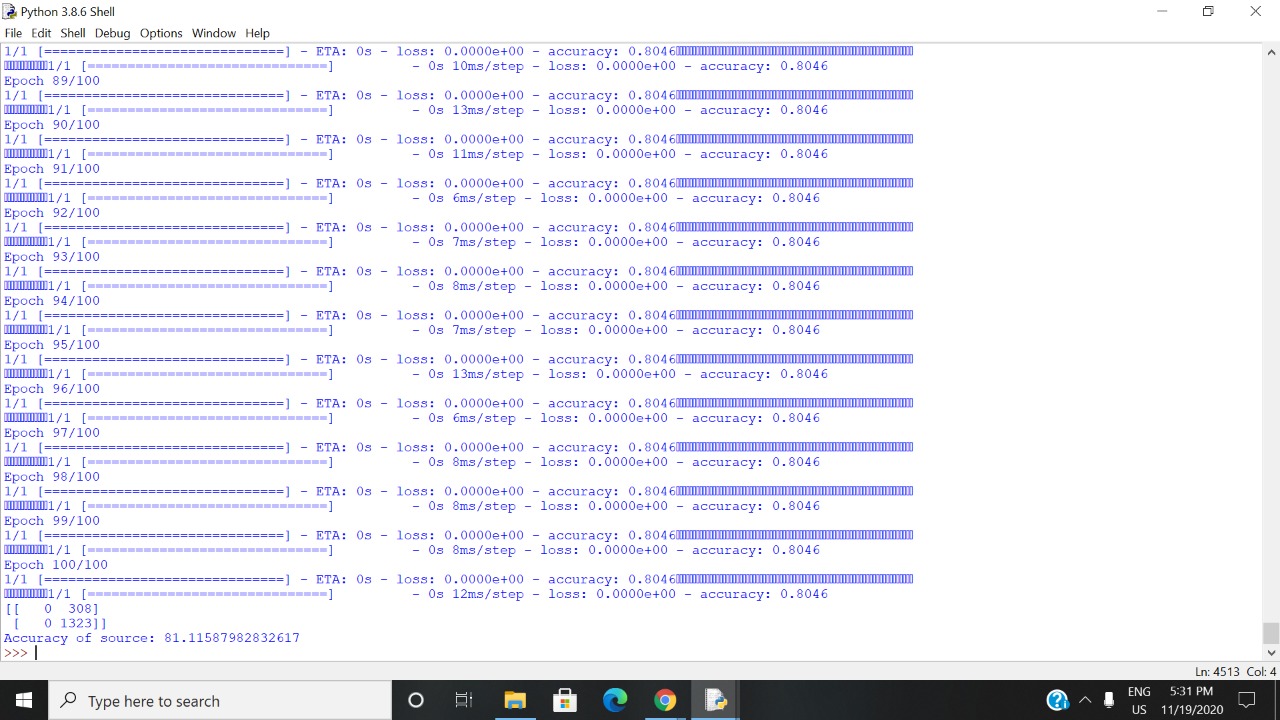
plt.legend()

plt.show()

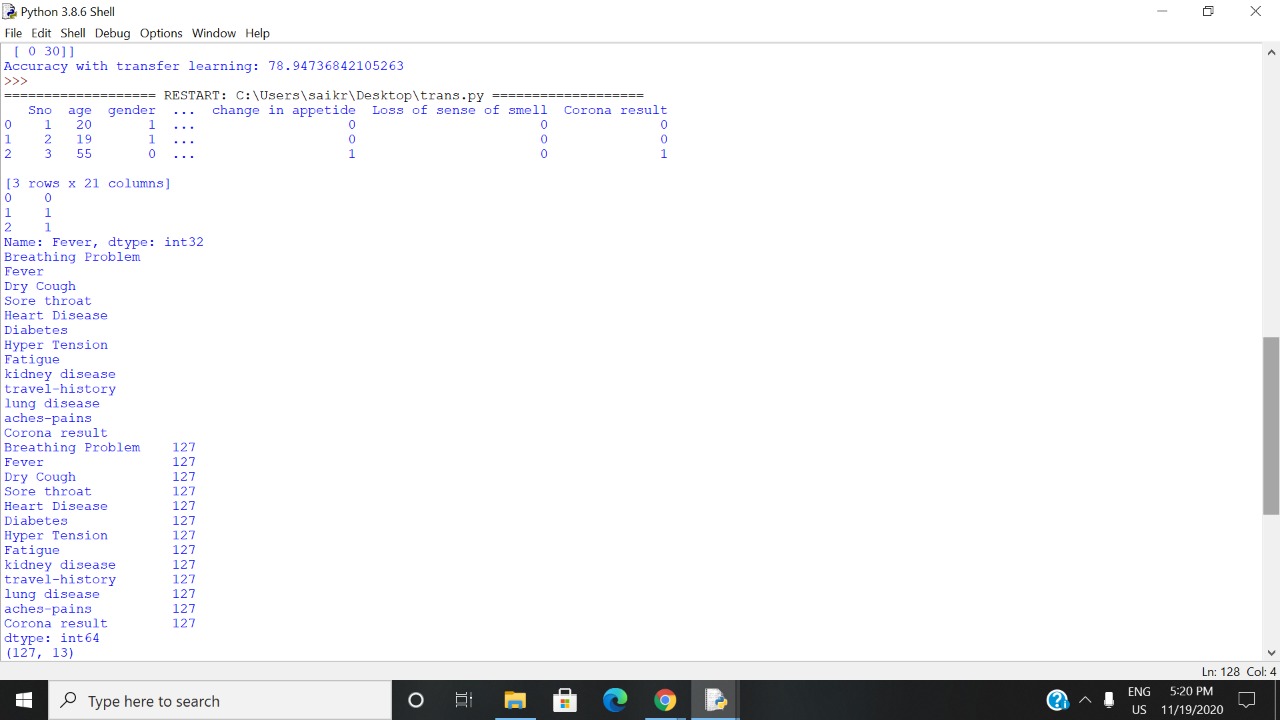
**Output screens:**

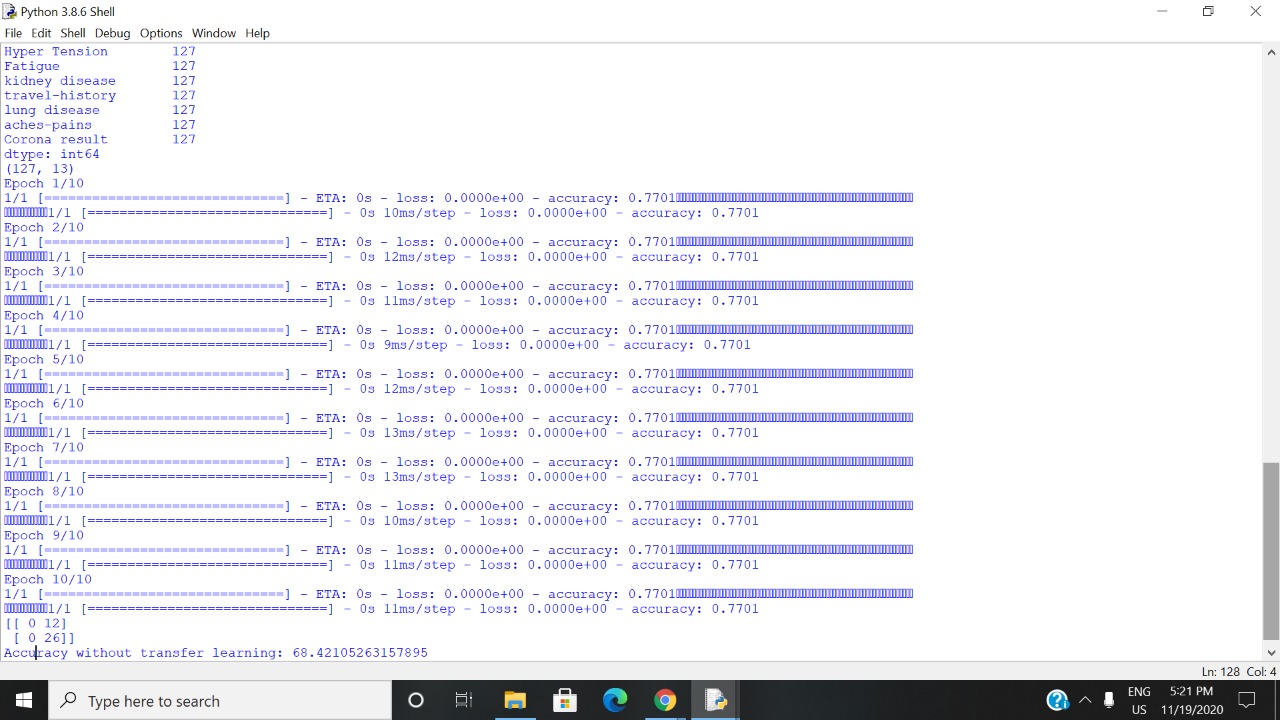
****

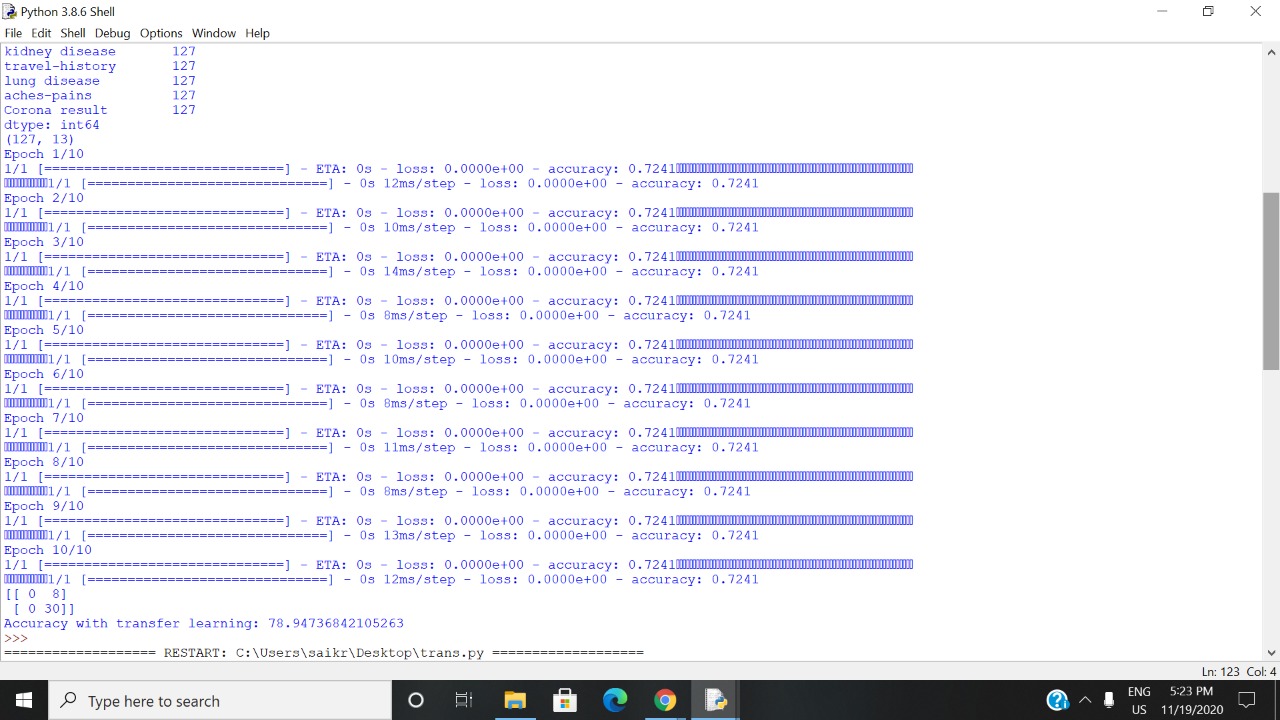
Factors after generalization



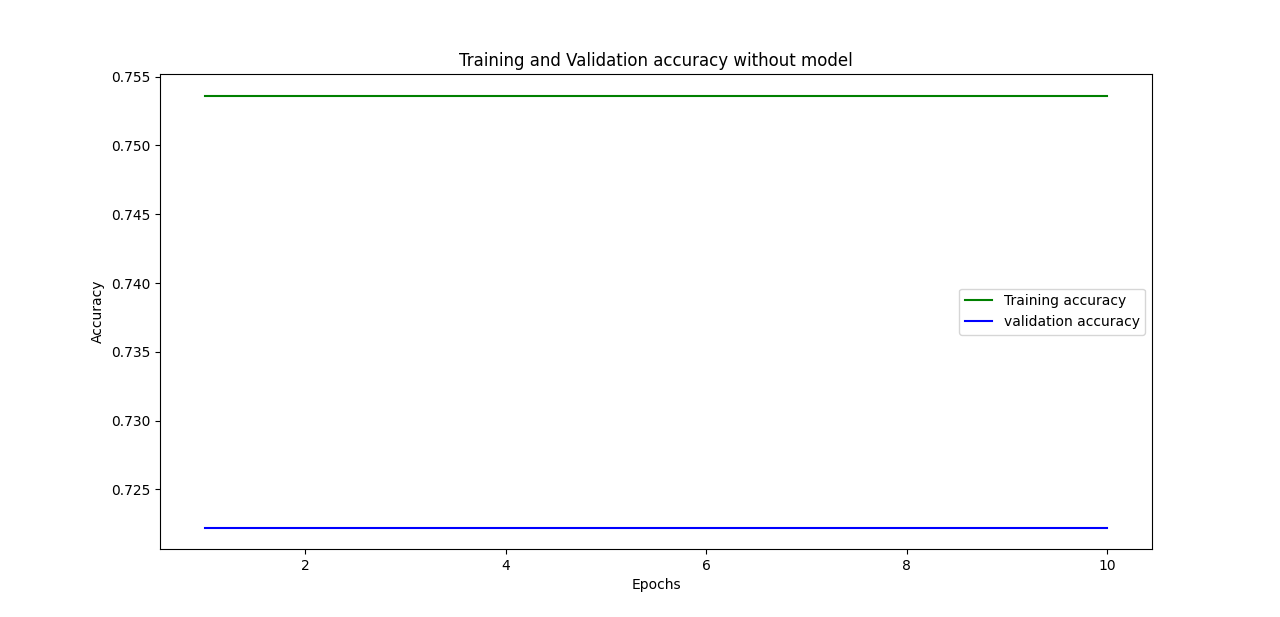
Accuracy of source model

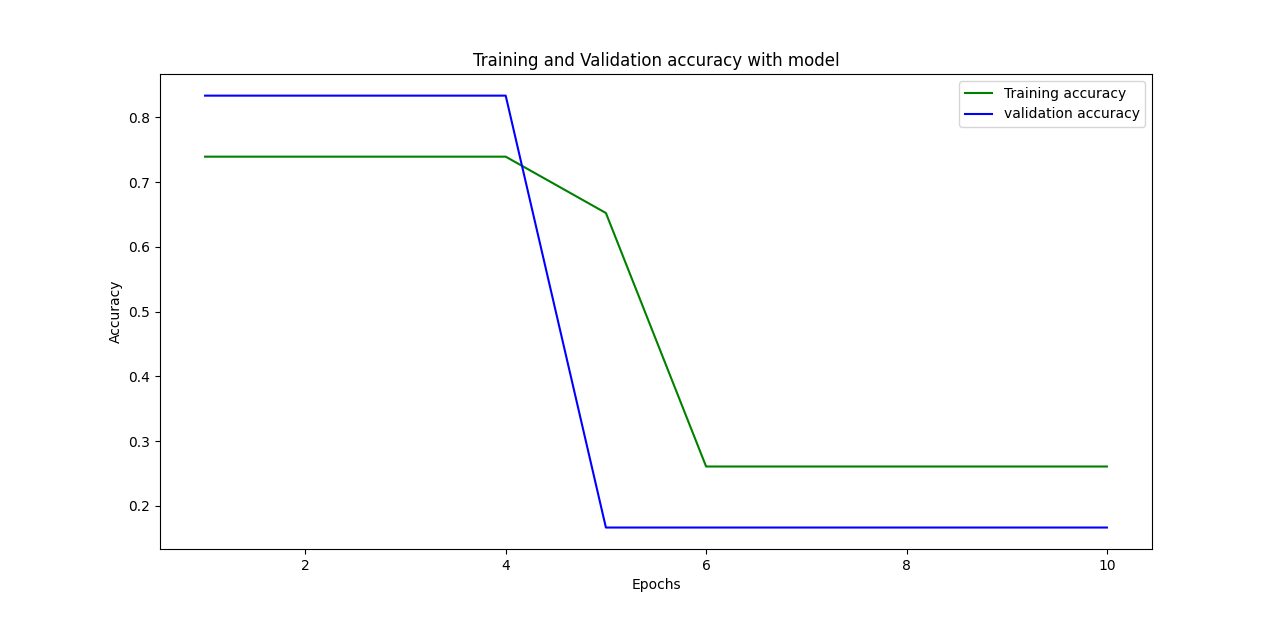


Accuracy without transfer learning

Accuracy after using transfer learning model

**Graphs**

****

****