**Name: Gaurav Kumar Singh**

**Registration Number: 19BCE2119**

**Course: CSE3501-ELA-L21+L22 (ISAA)**

**Lab FAT**

**Question set: 2a**

Develop a Machine Learning based Malware detection system using any of the following Logistic Regression [30]

b. Using Cisco Packet Tracer configure LAN that acts as a connector between two computers to enable message transfer.  [20]

**1)**

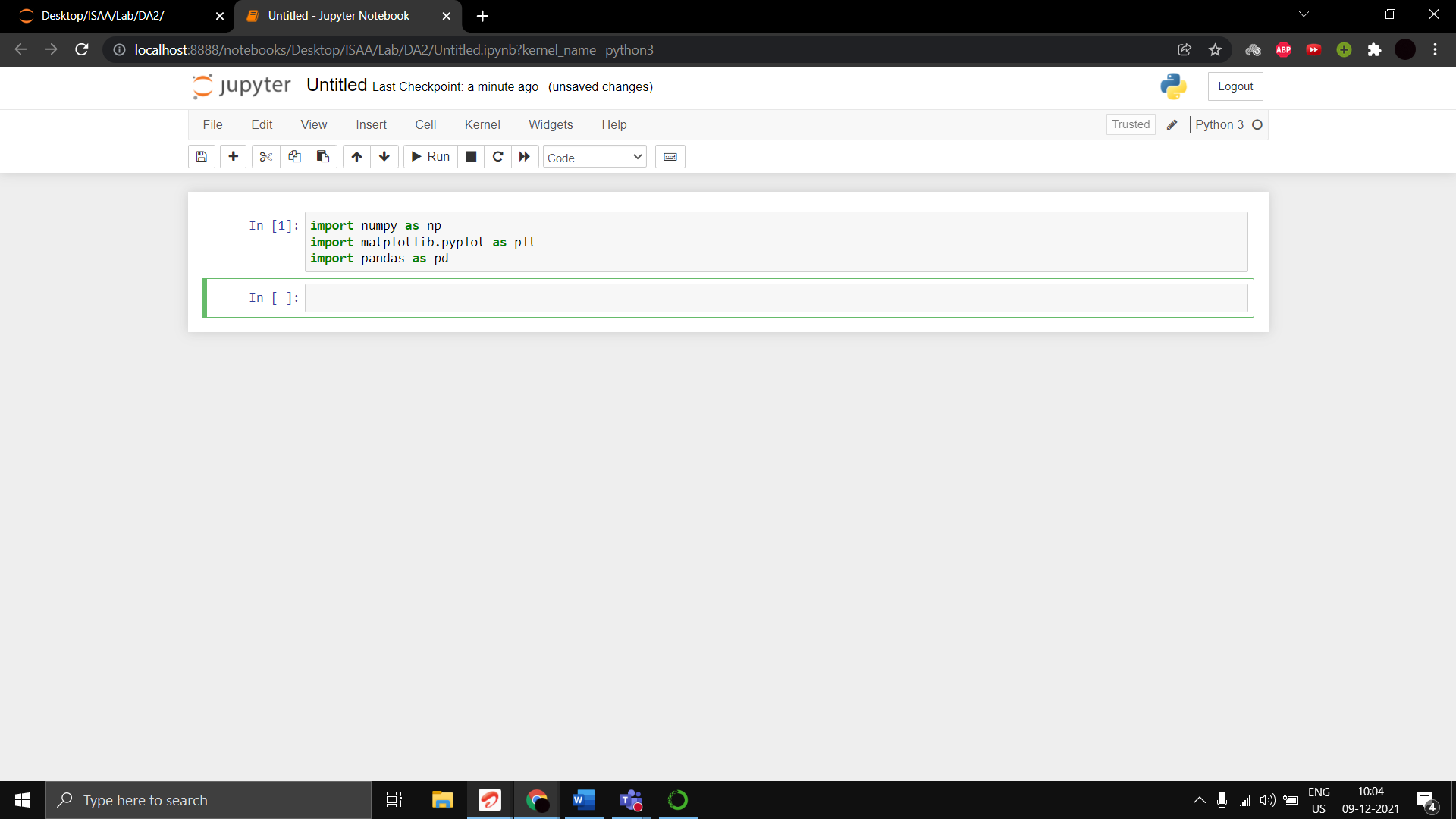
**Logistic Regression** [**(Full Code)**](#_QUESTION_2a_FULL)

Importing python libraries such as NumPy, matplotlib and pandas for data reading and manipulation.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd



Reading dataset and truncating irrelevant columns

dataset = pd.read\_csv('Malware\_dataset.csv')

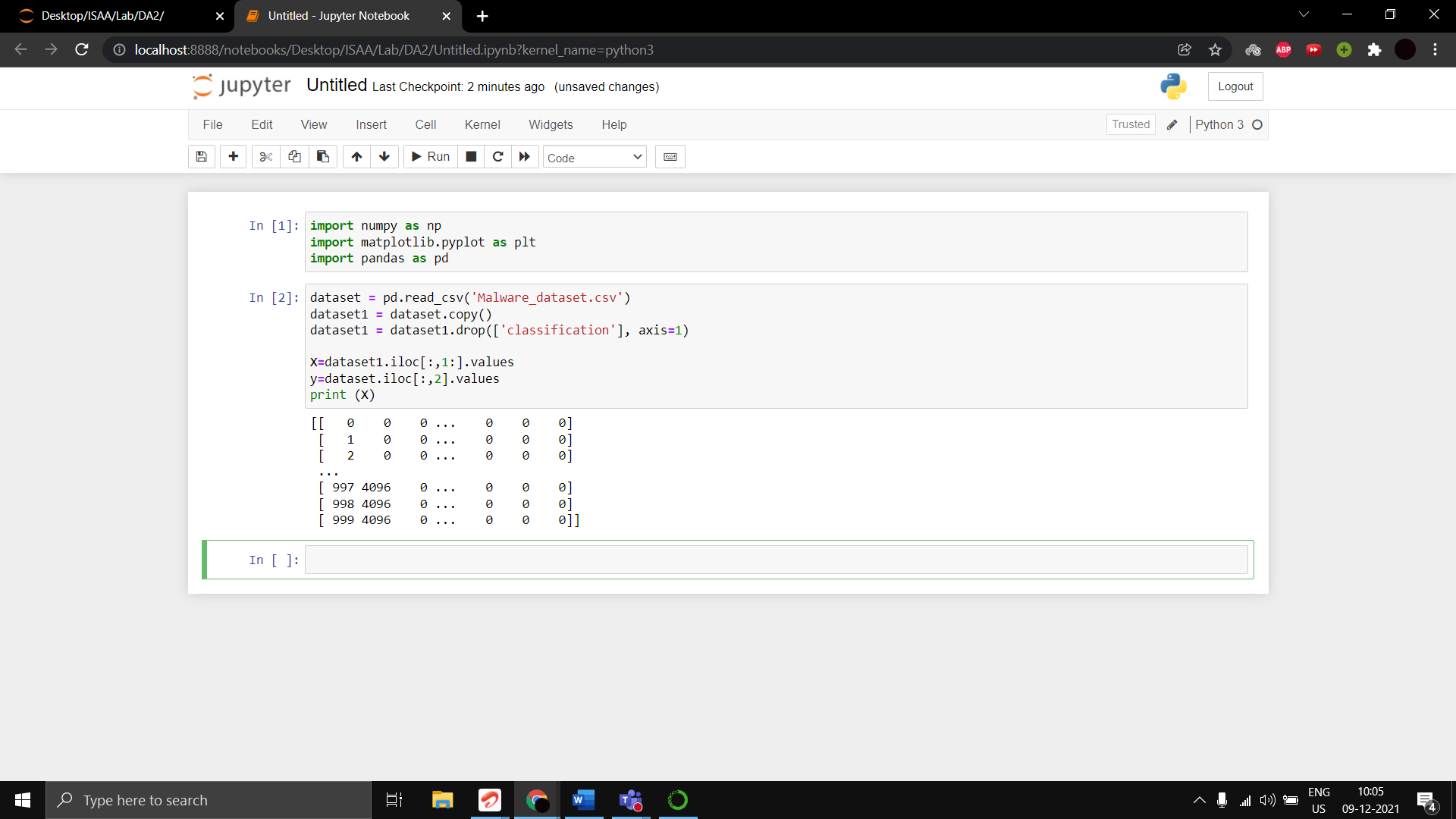
dataset1 = dataset.copy()

dataset1 = dataset1.drop(['classification'], axis=1)

X=dataset1.iloc[:,1:].values

y=dataset.iloc[:,2].values

print (X)



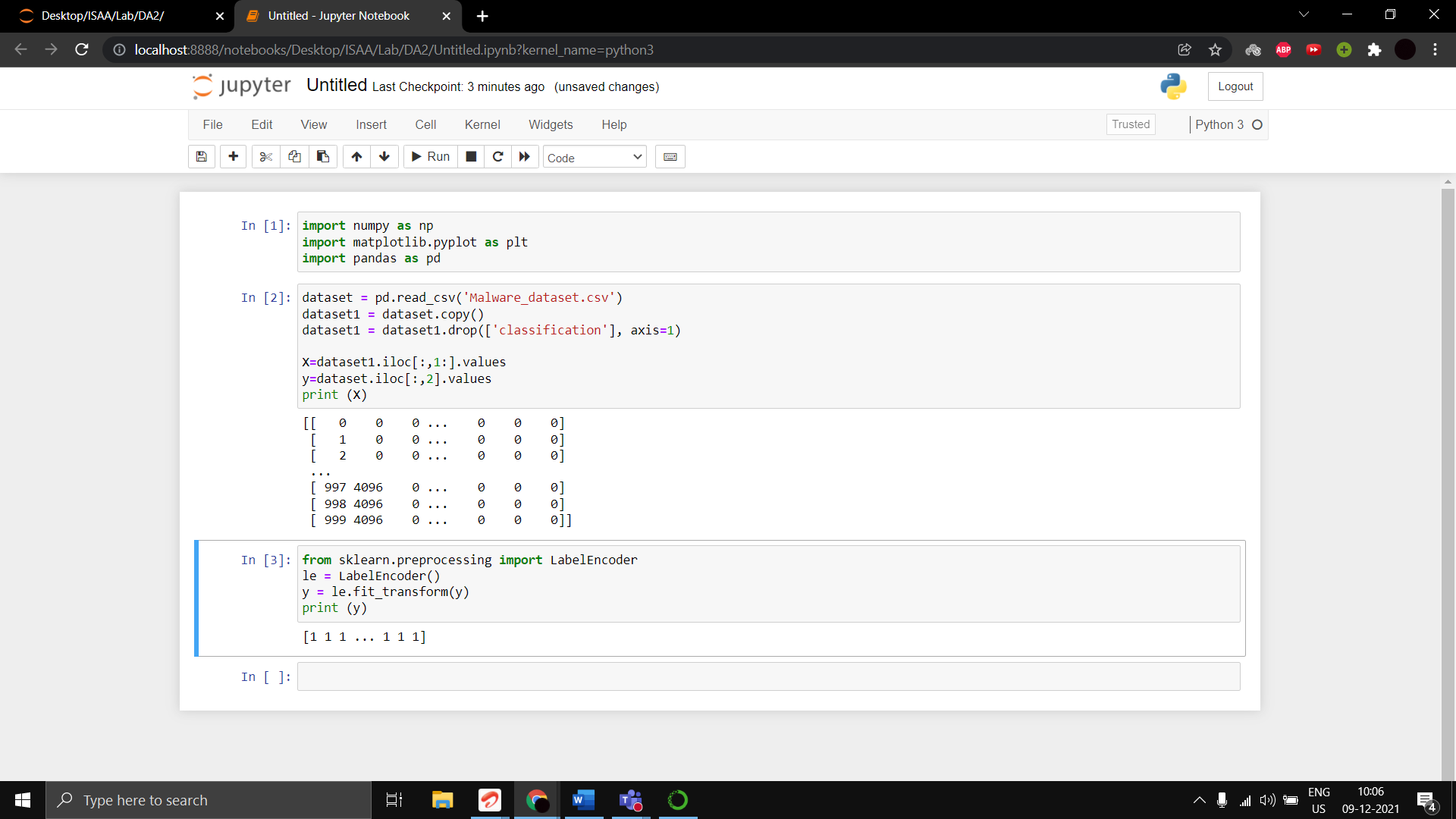
Importing sklearn.preprocessing module to transform classification dataset using label encoder for classification.

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

y = le.fit\_transform(y)

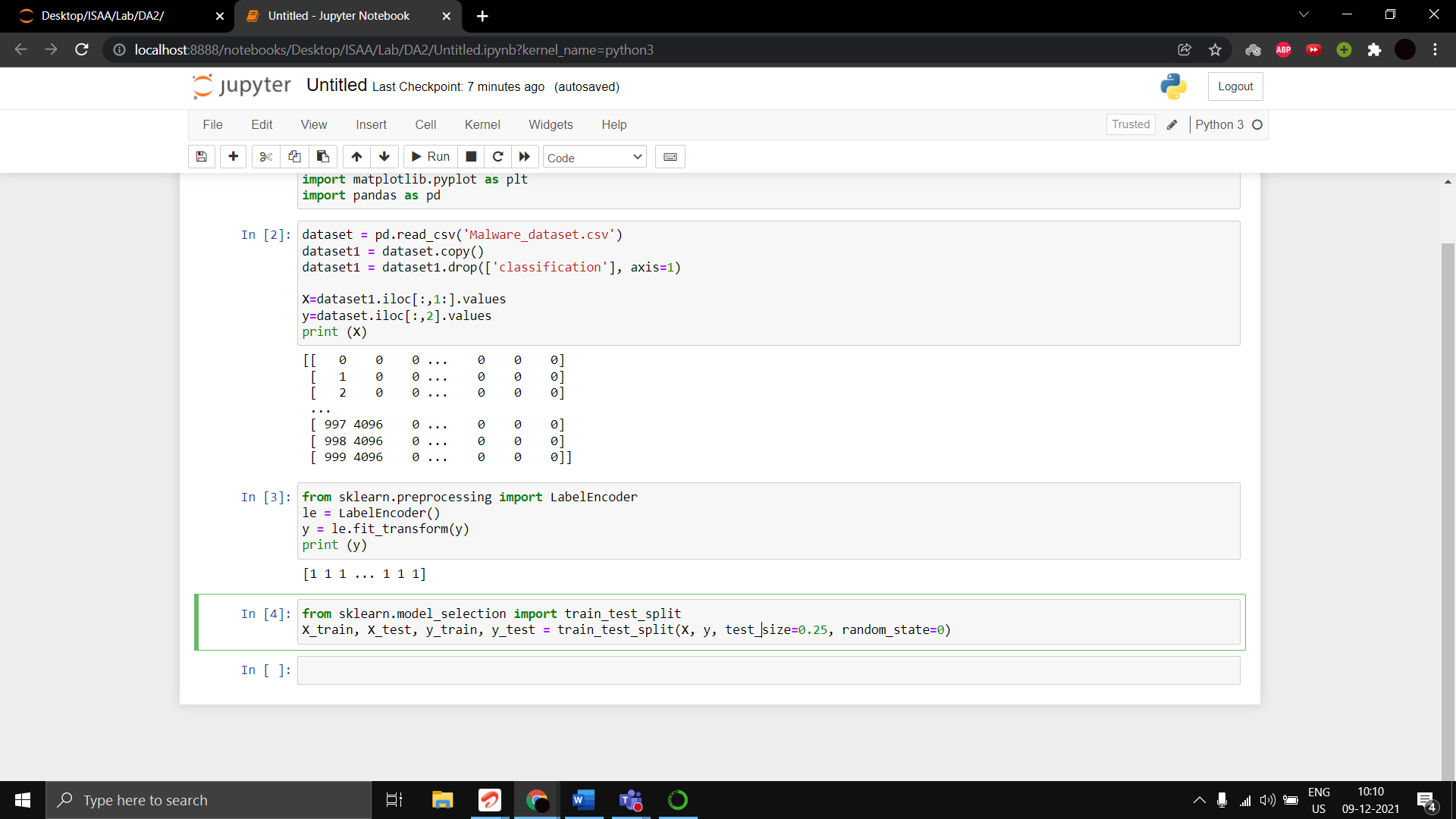
print (y)



Splitting dataset into training and testing dataset to calculate accuracy precision and recall on.  
Test size: Train size:: 1: 3

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=0)



Printing the training and testing datasets

print("Xtrain")

print(X\_train)

print("Xtest")

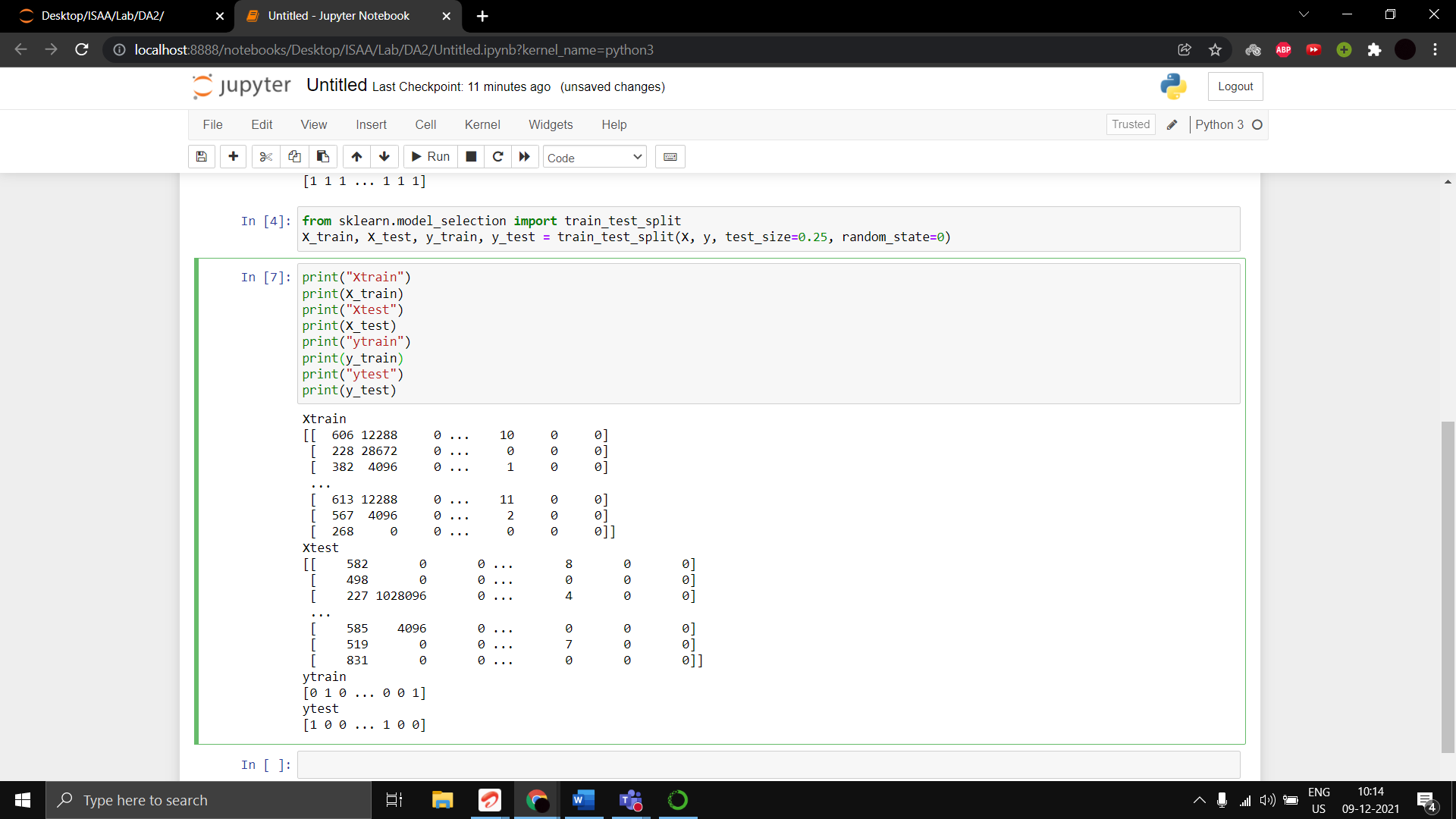
print(X\_test)

print("ytrain")

print(y\_train)

print("ytest")

print(y\_test)



Scaler transform of the training dataset

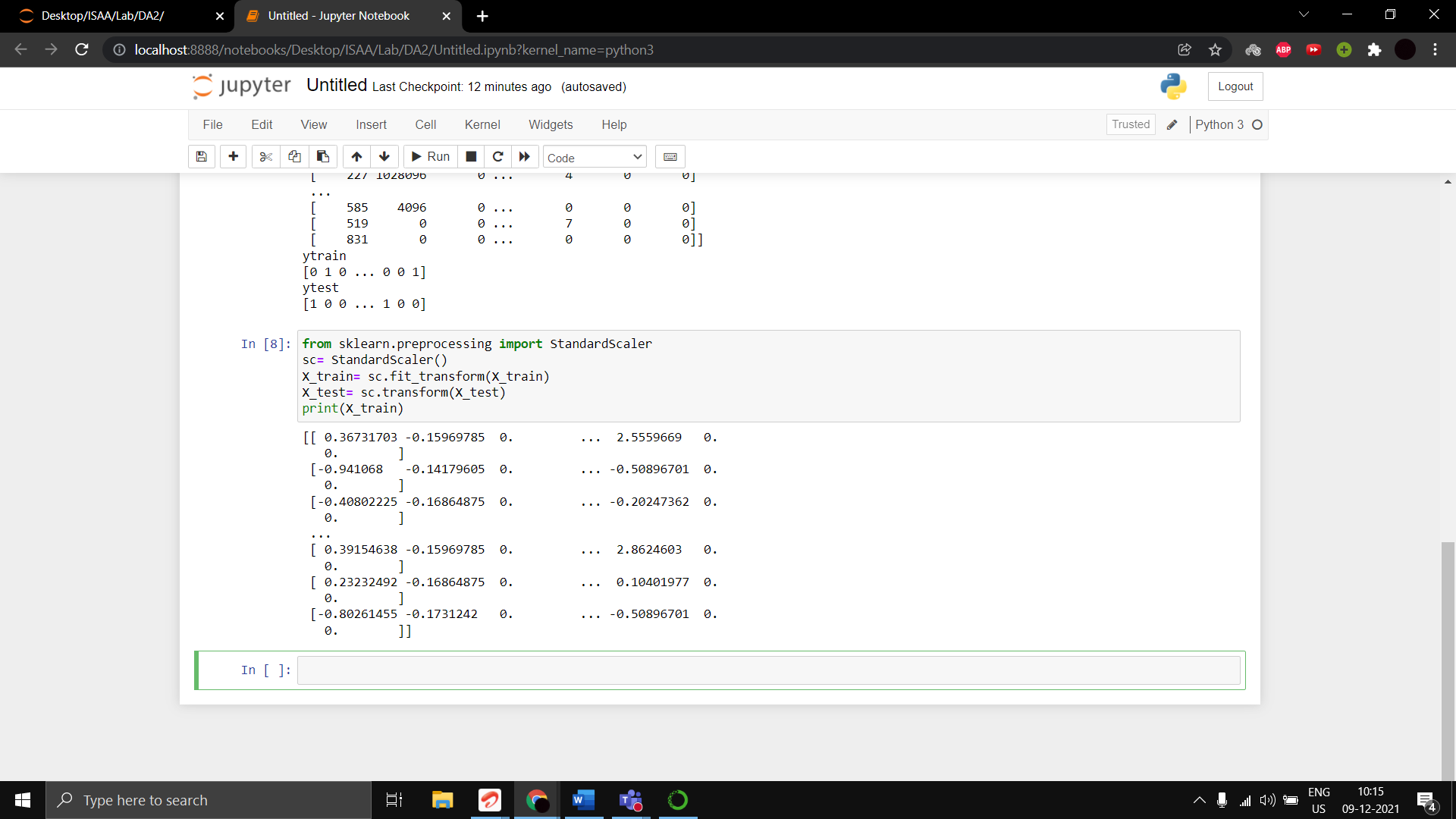
from sklearn.preprocessing import StandardScaler

sc= StandardScaler()

X\_train= sc.fit\_transform(X\_train)

X\_test= sc.transform(X\_test)

print(X\_train)

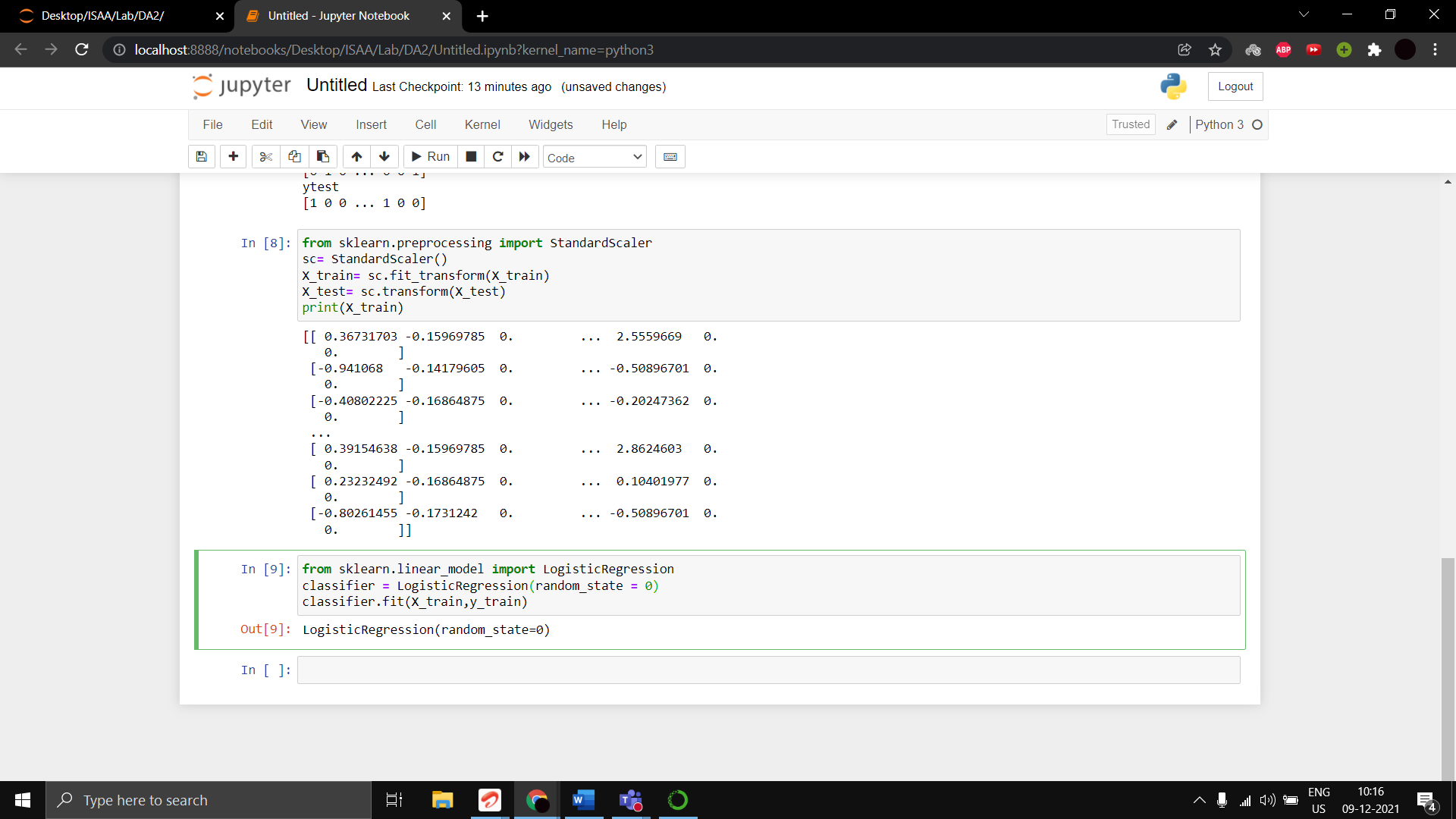


Training the algorithm using Logistic regression

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

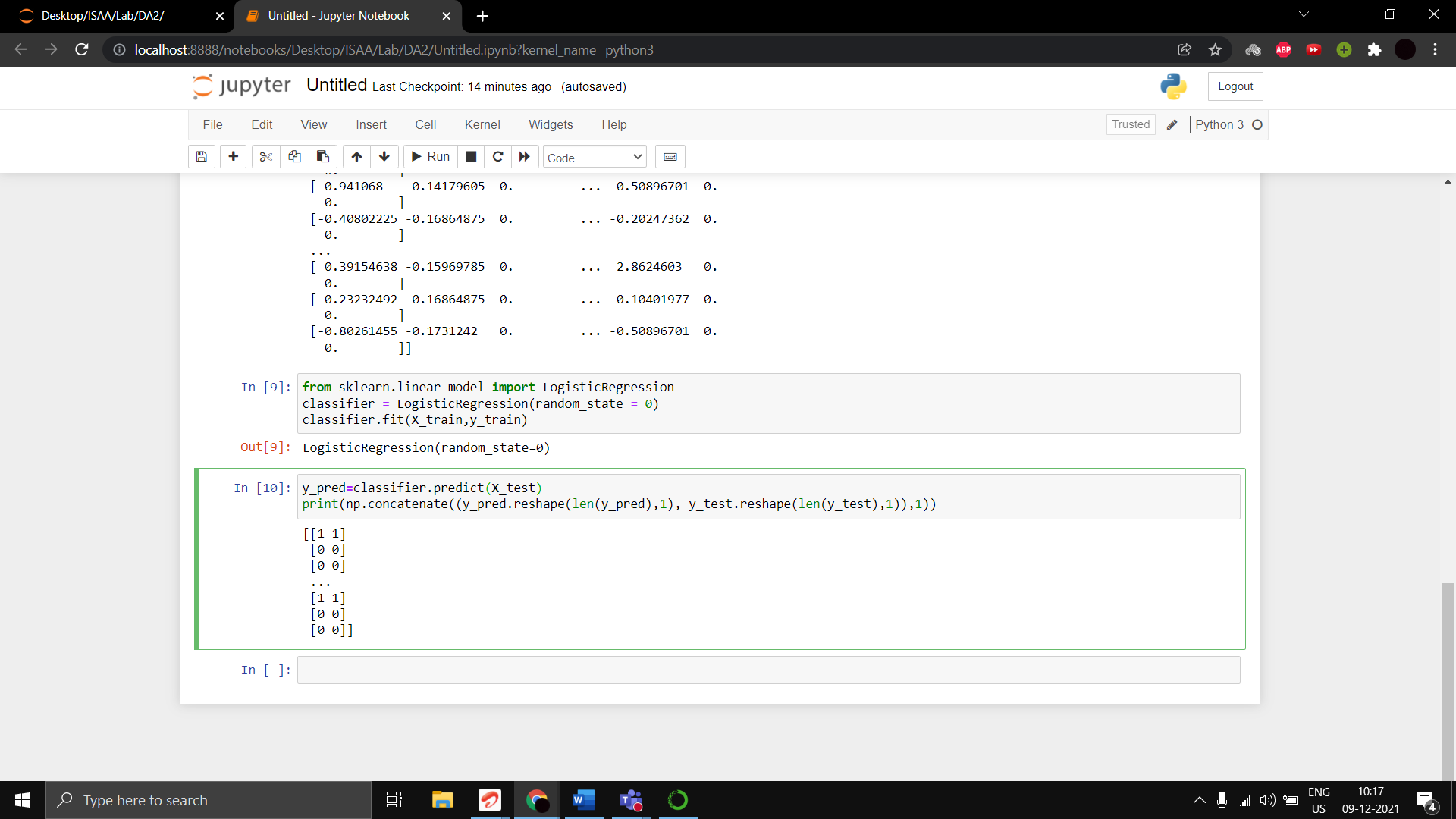
classifier.fit(X\_train,y\_train)



Testing the algorithm on the test split dataset

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

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))



Accuracy Score

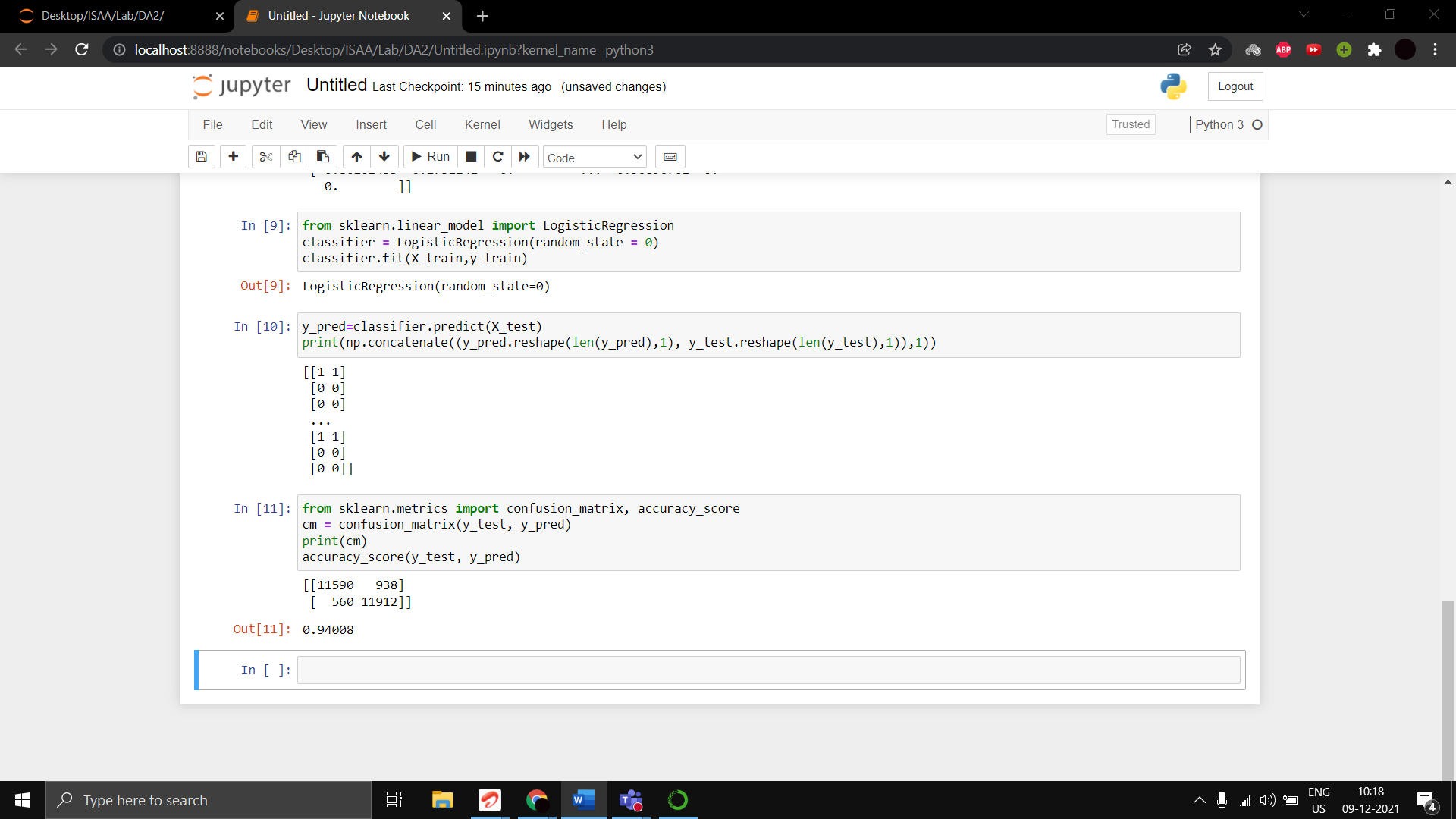
from sklearn.metrics import confusion\_matrix, accuracy\_score

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

print(cm)

accuracy\_score(y\_test, y\_pred)

Accuracy Score: 0.94008



Precision Score

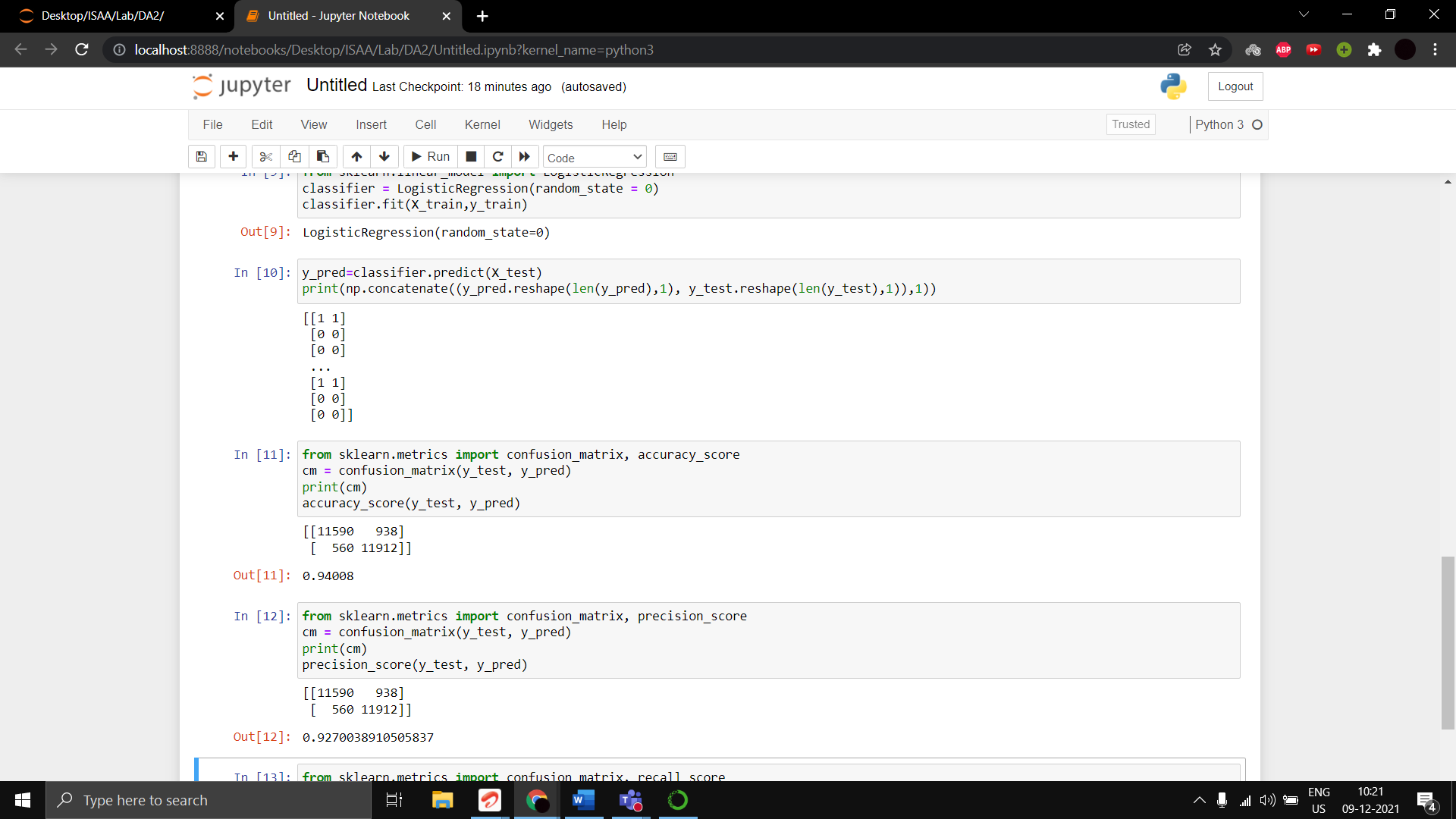
from sklearn.metrics import confusion\_matrix, precision\_score

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

print(cm)

precision\_score(y\_test, y\_pred)

Precision Score: 0.9270038910505837



Recall Score

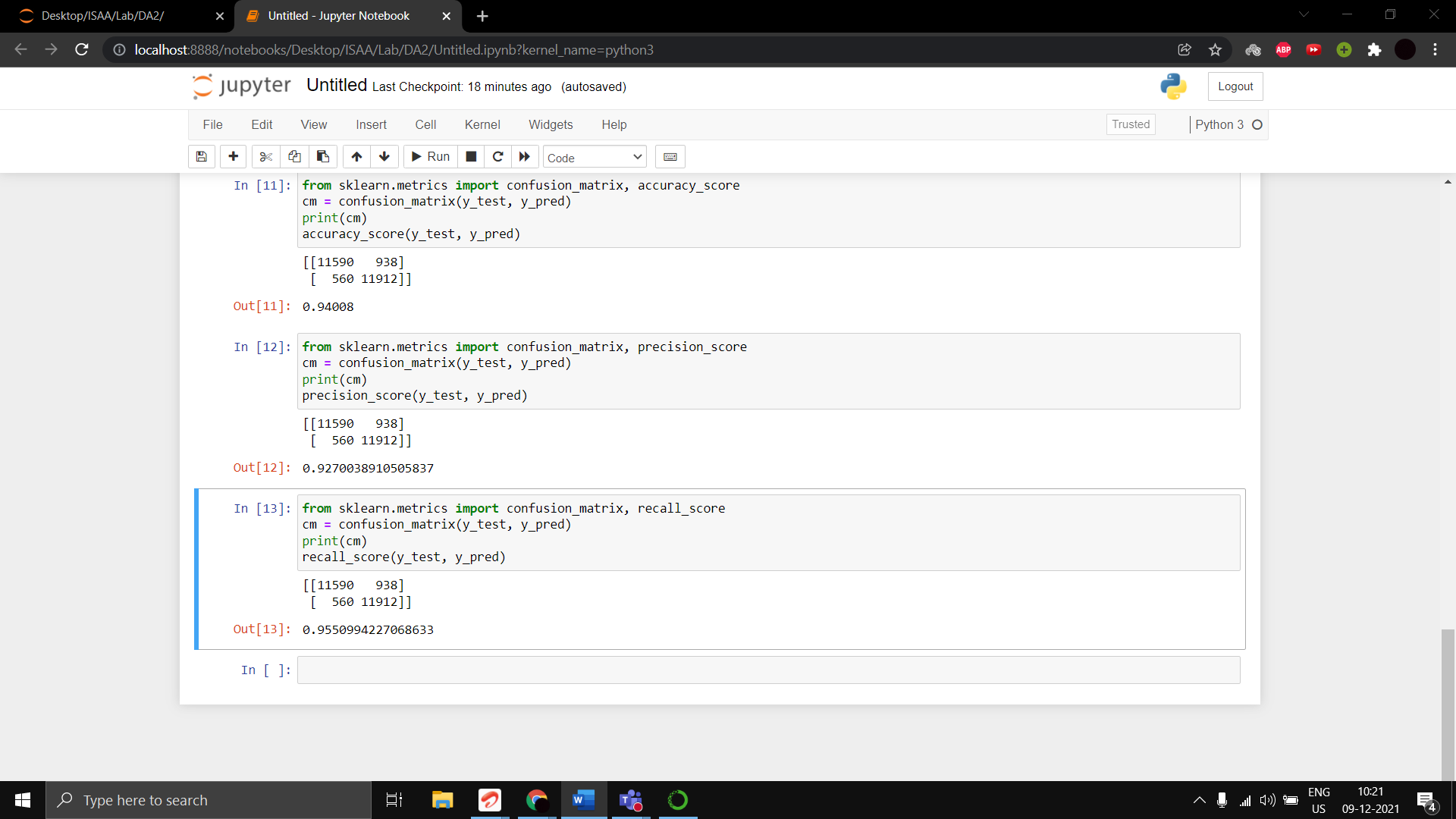
from sklearn.metrics import confusion\_matrix, recall\_score

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

print(cm)

recall\_score(y\_test, y\_pred)

Recall Score: 0.9550994227068633



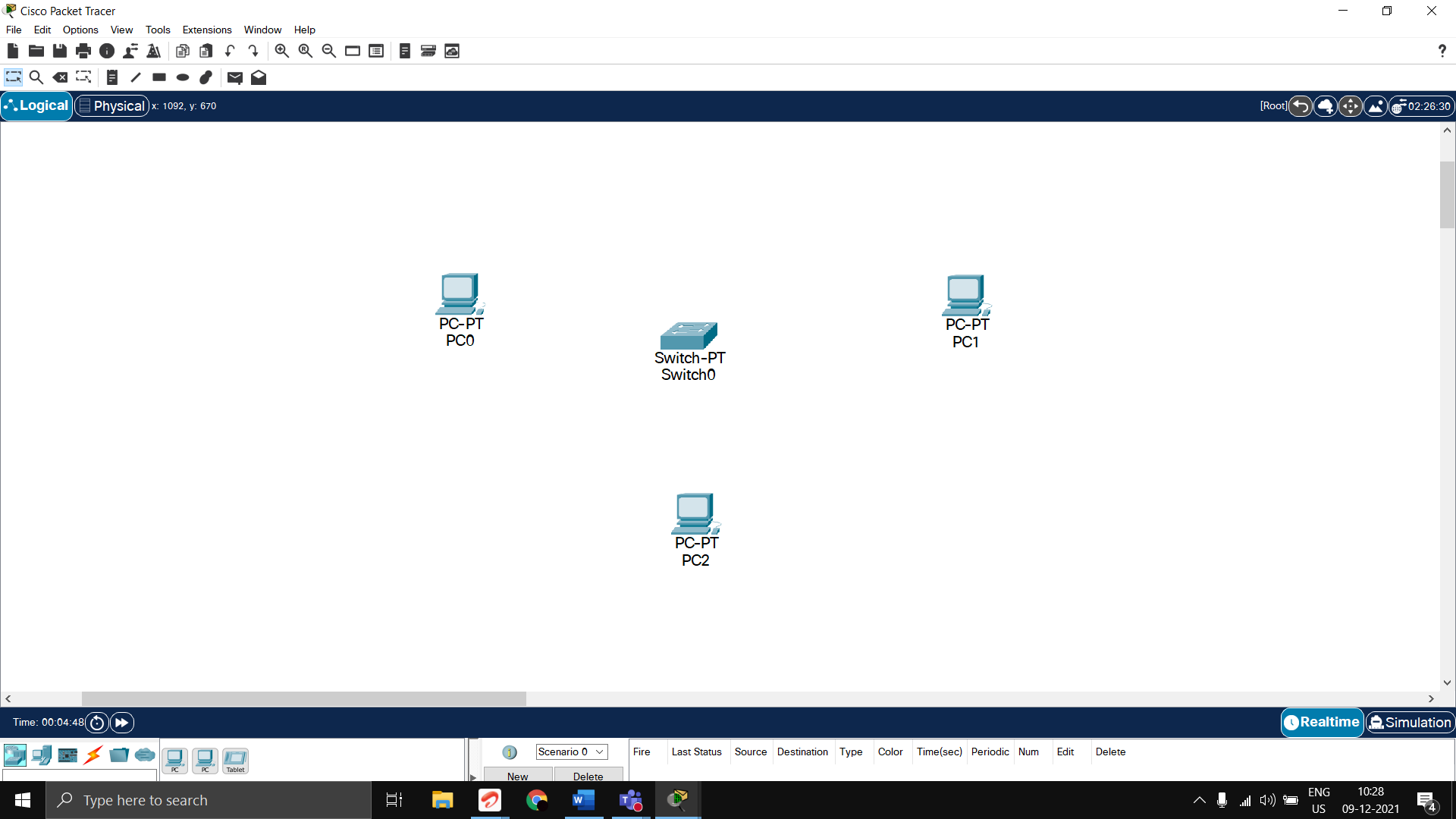
**2)**

Components used: PC

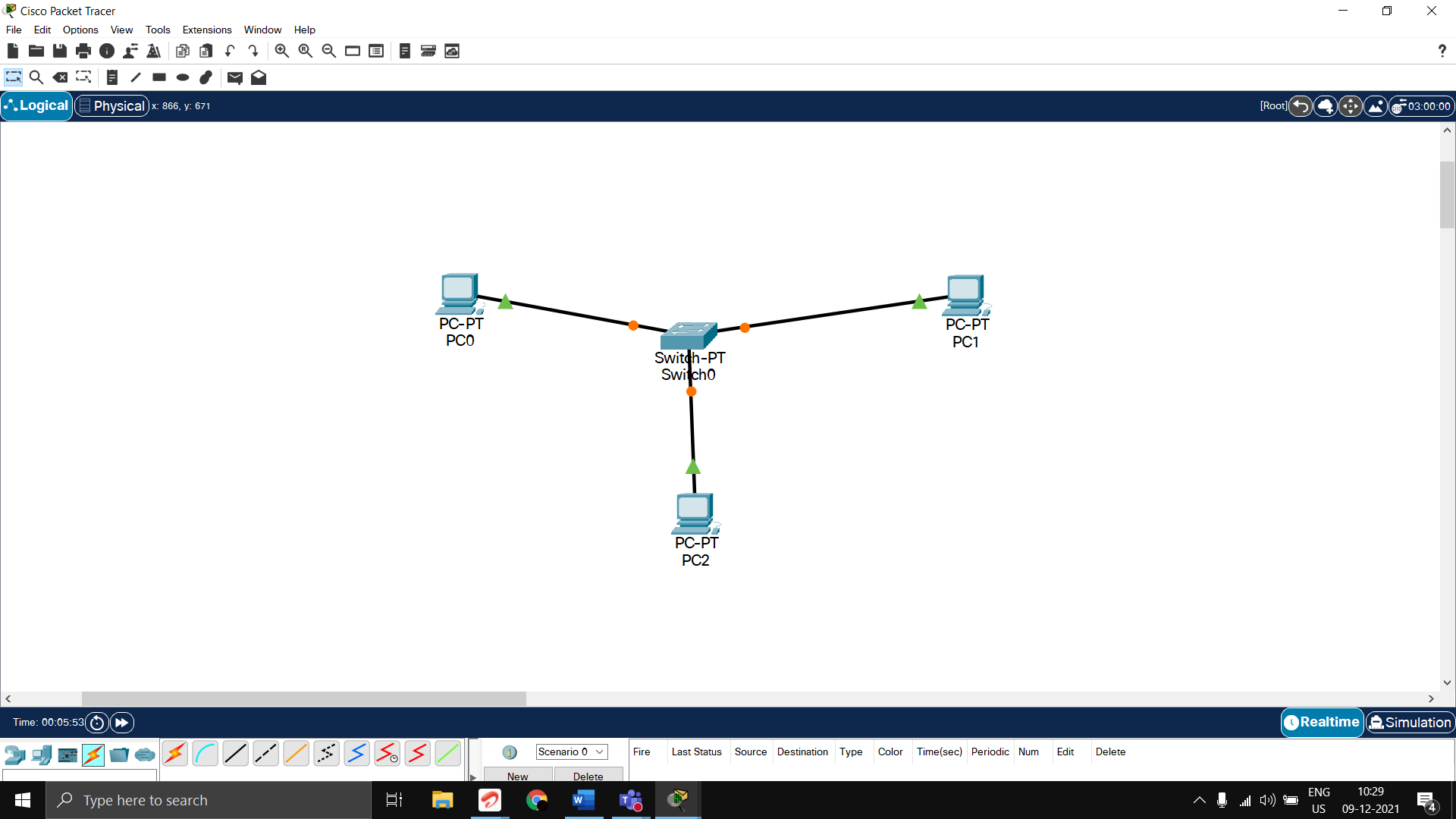
Switch

STEPS:

1. Placing the components (PC and Switch) {PC and PT-Switch}

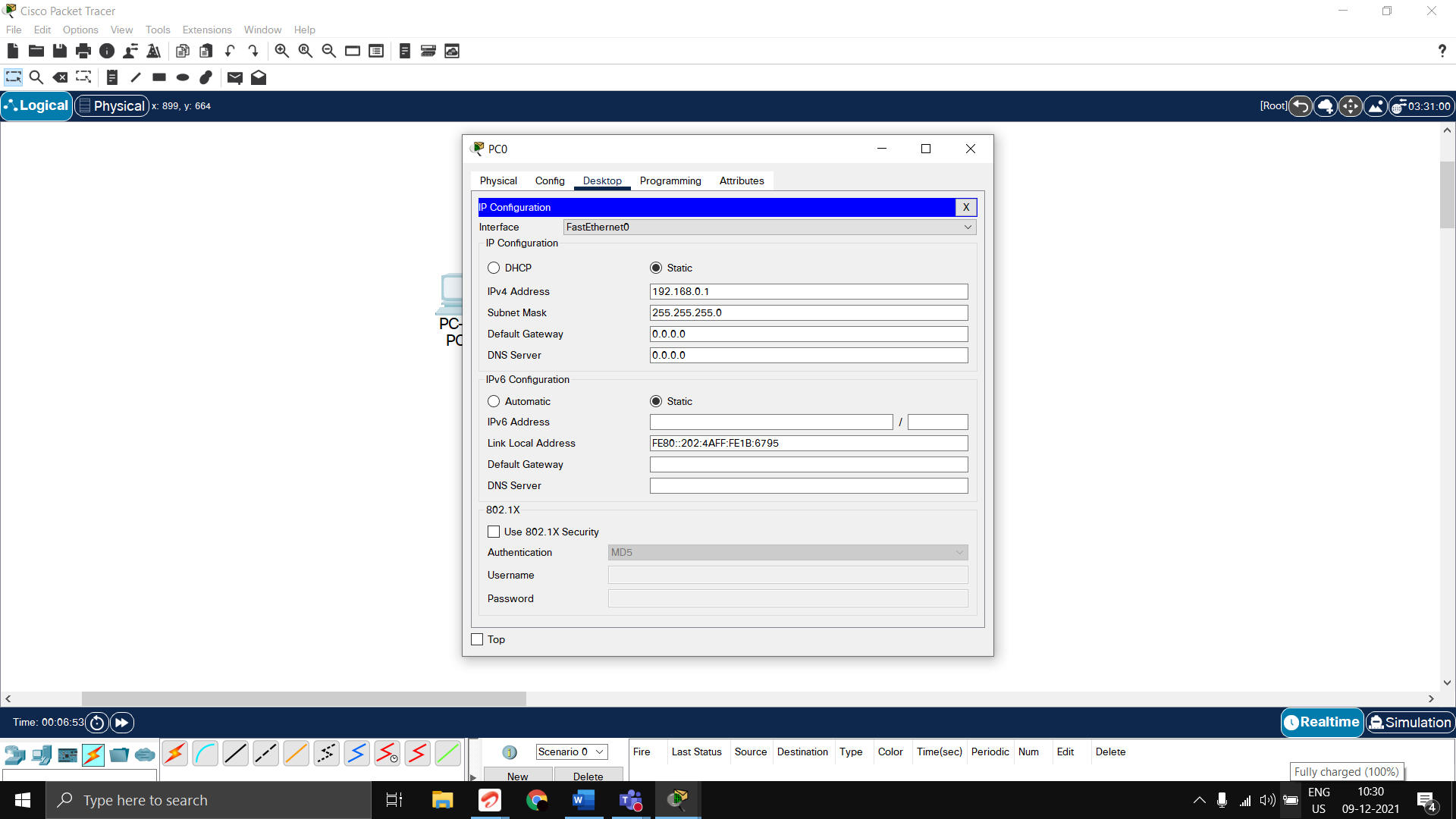


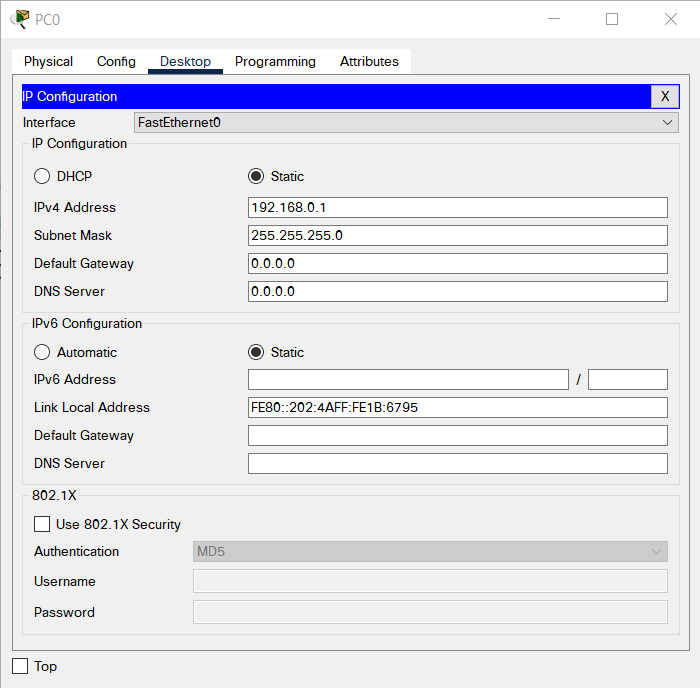
1. Connecting PCs to the Switch



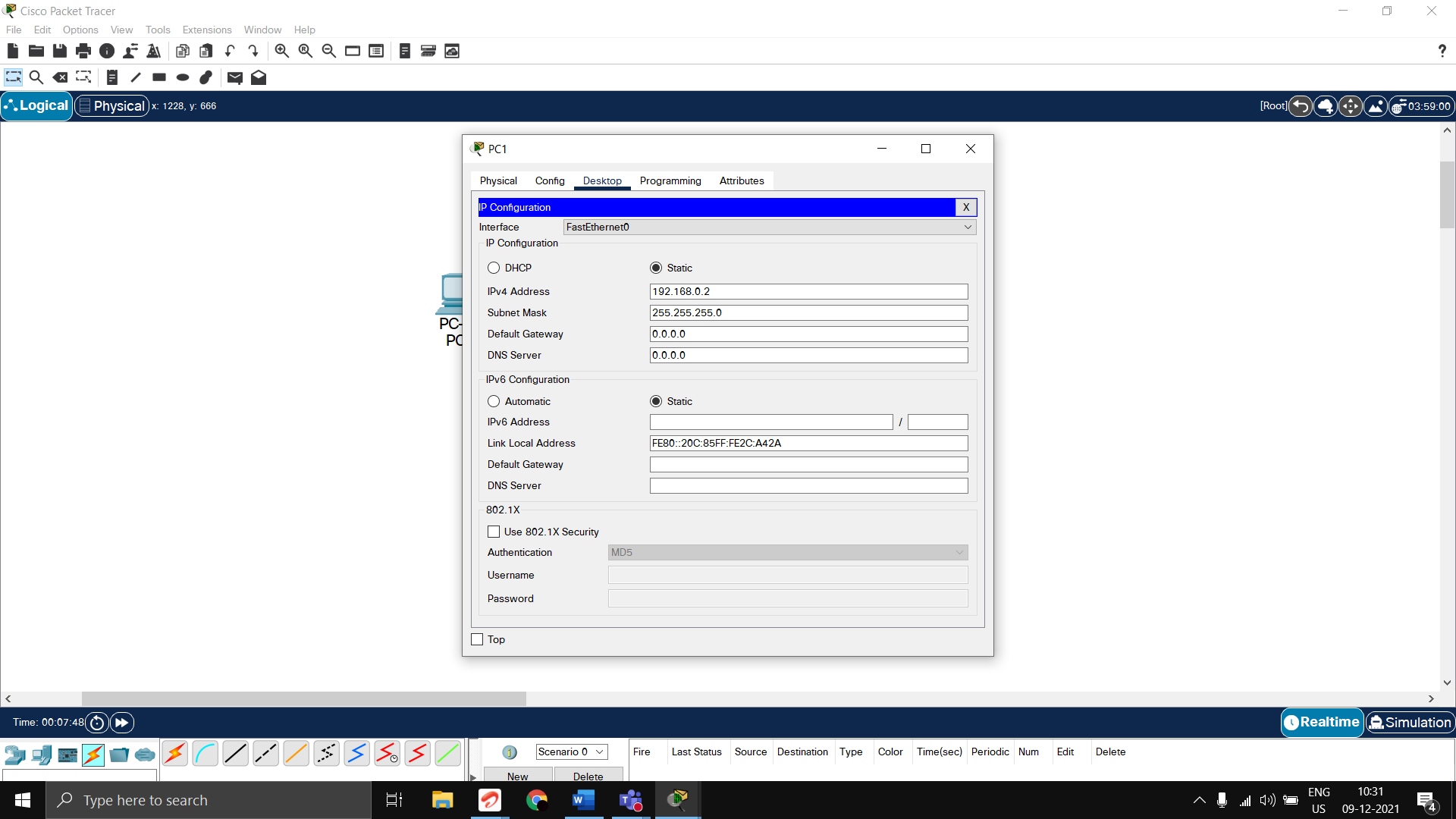
1. Configuring IP addresses and Masks of the PCs

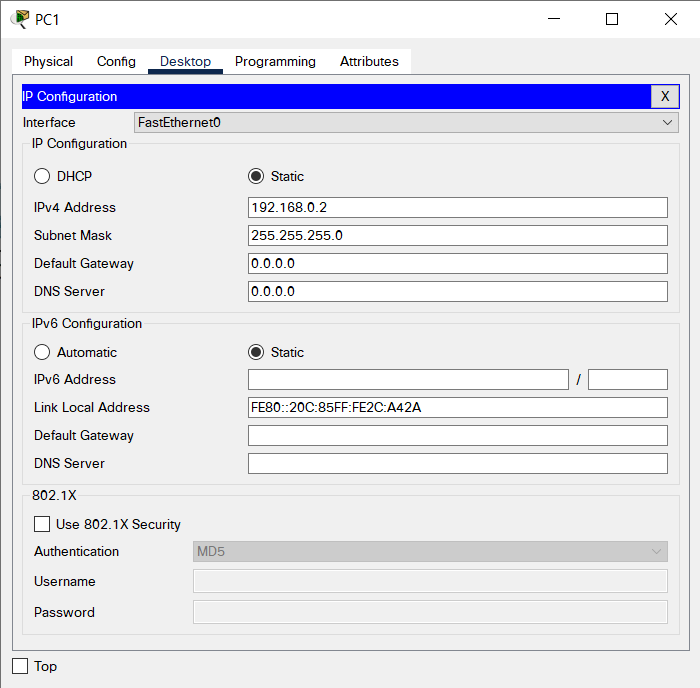
PC0



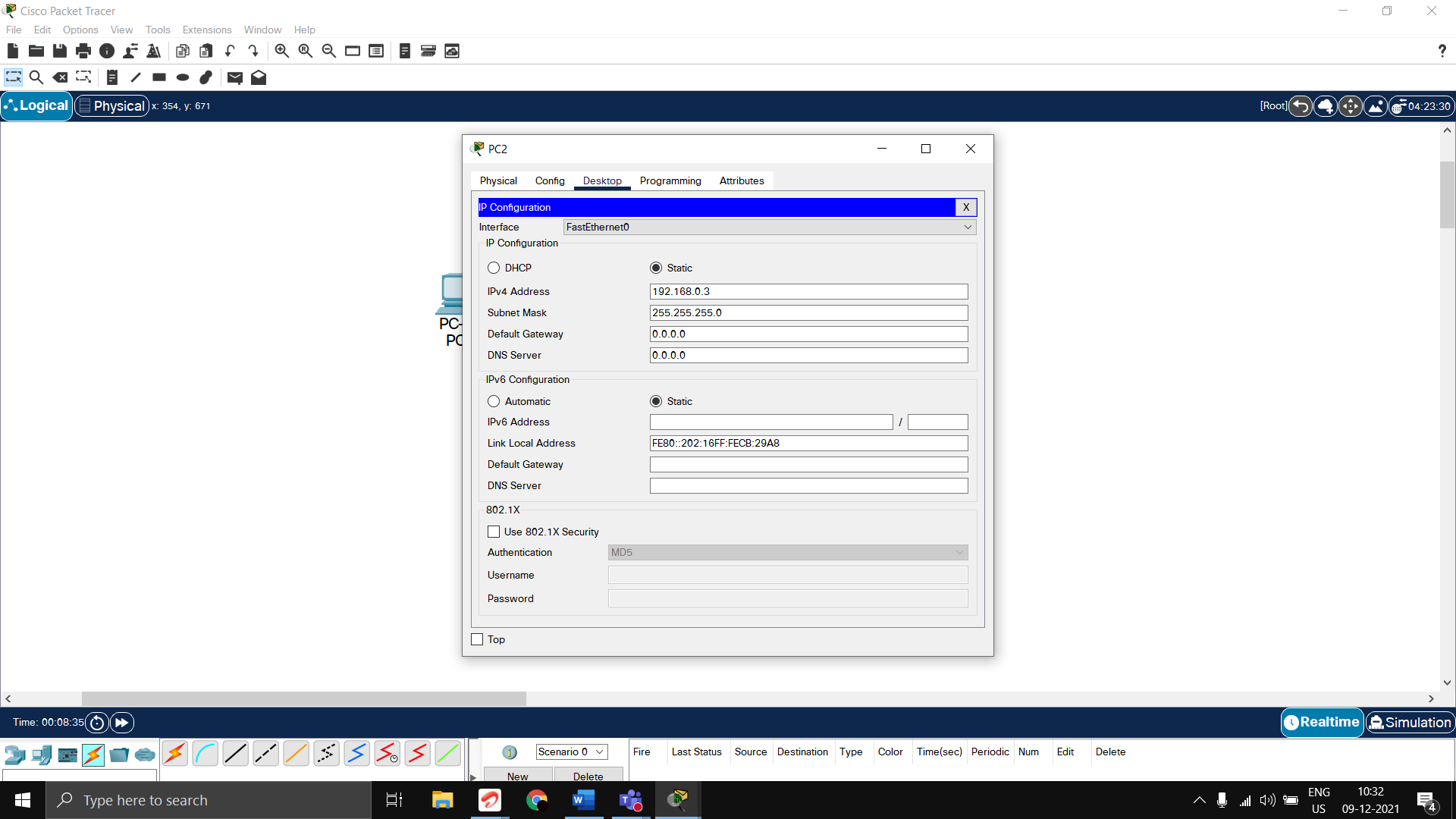


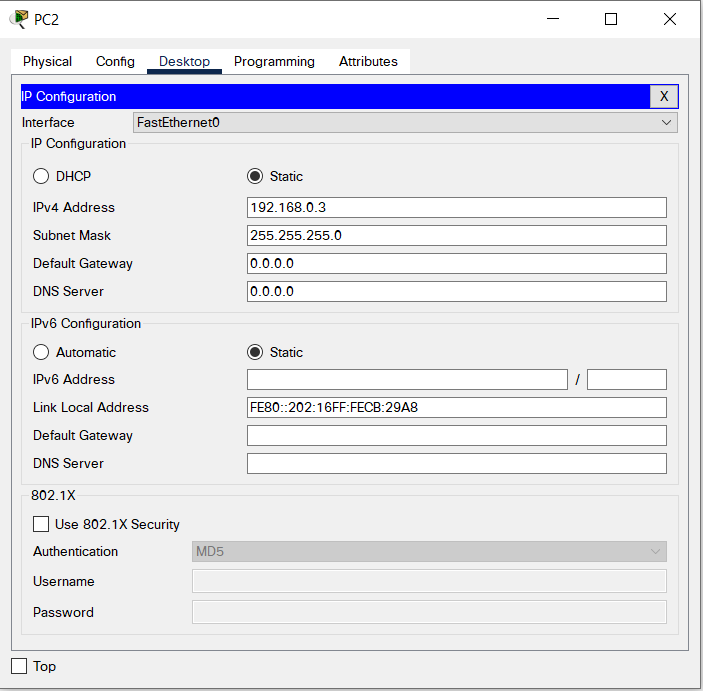
PC1





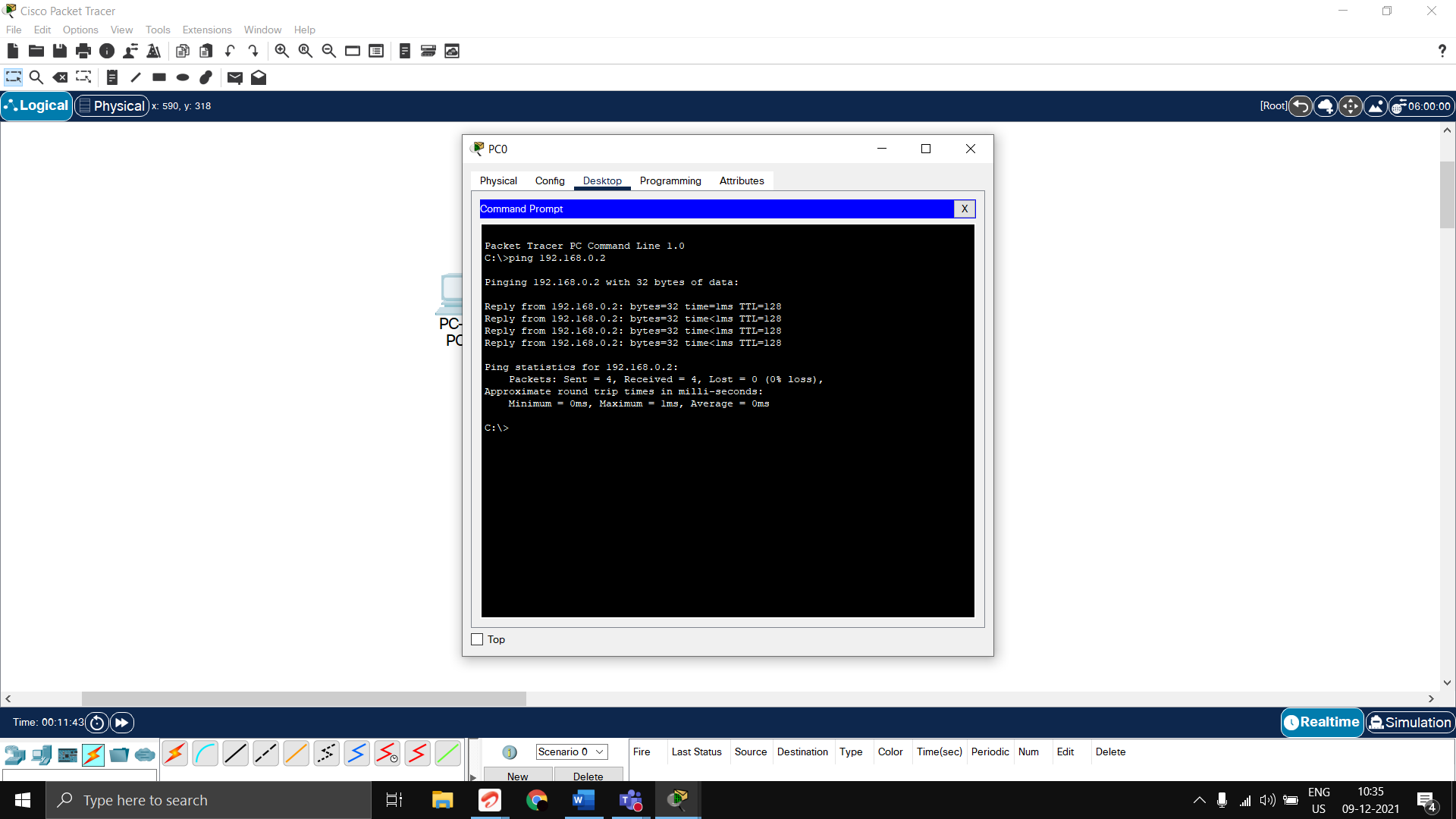
PC2

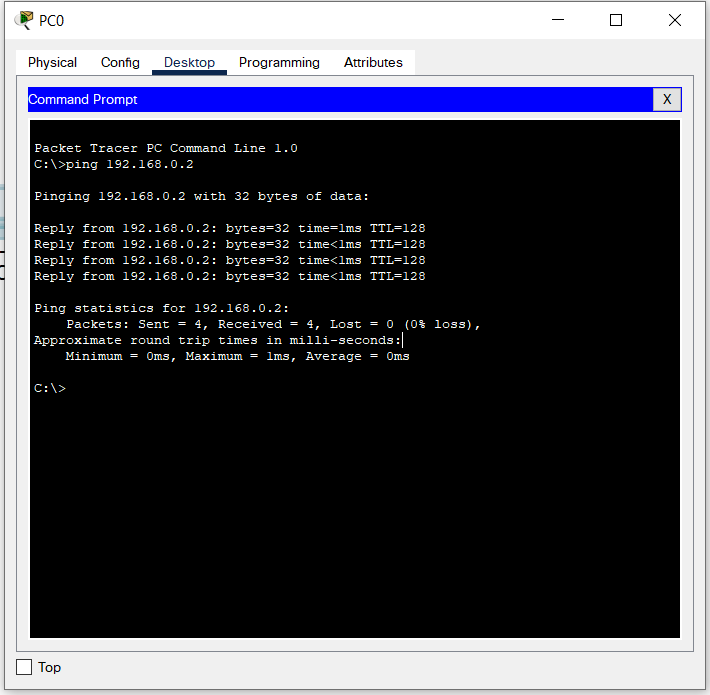




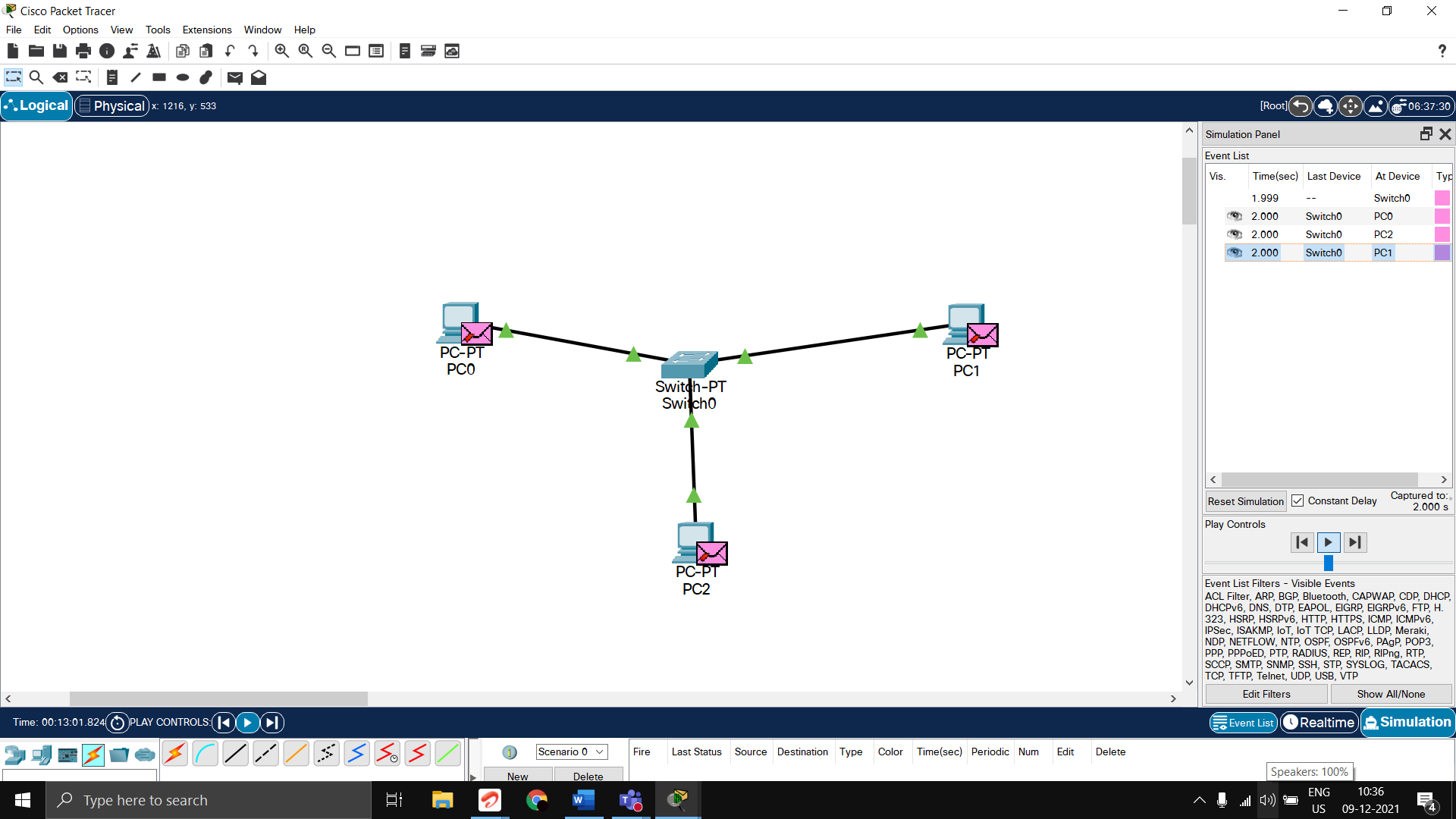
1. Checking the configuration using ping command and simulation  
   a) Ping Command (Using PC0 to ping PC1)

Command: ping 192.168.0.2

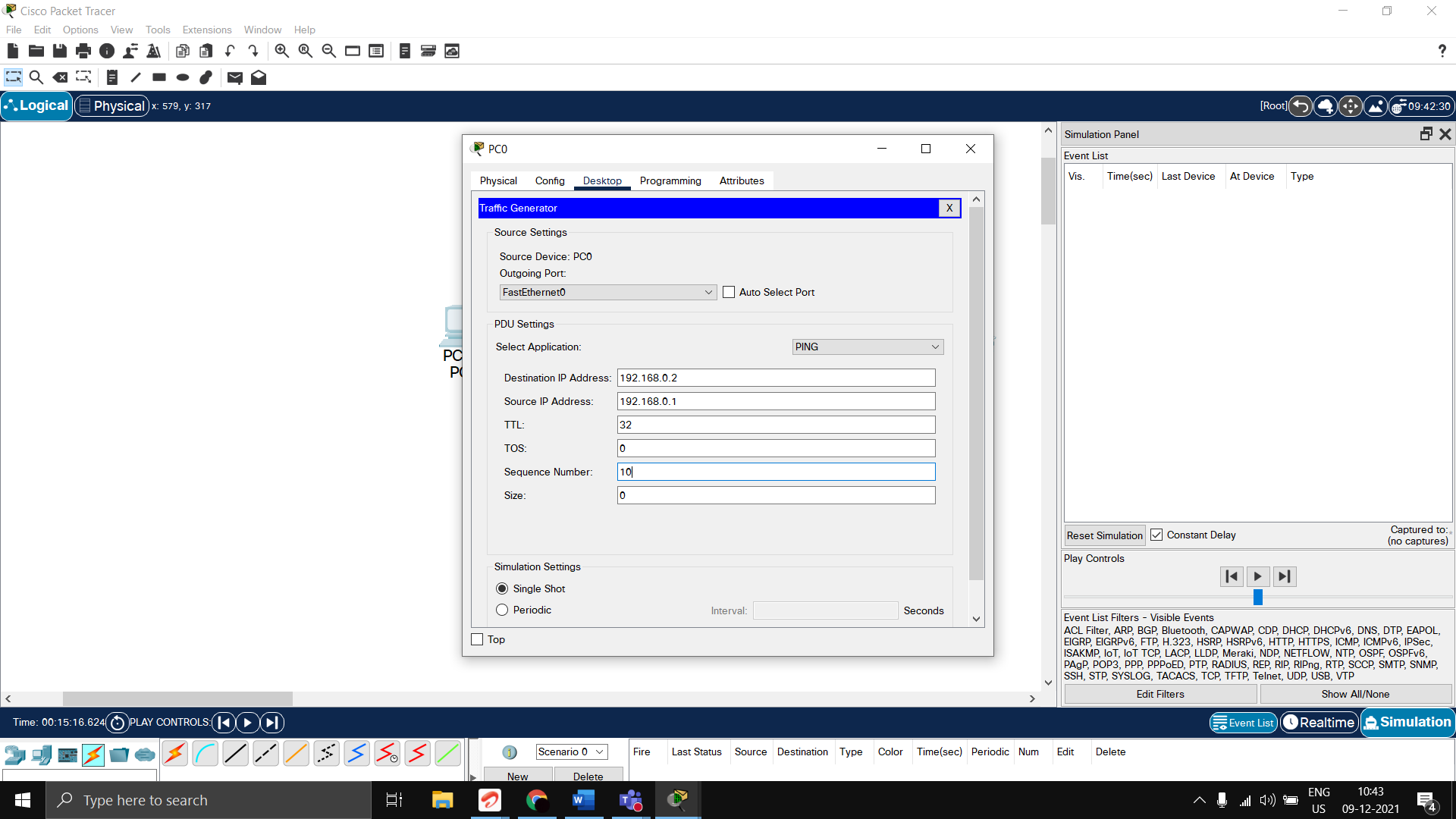


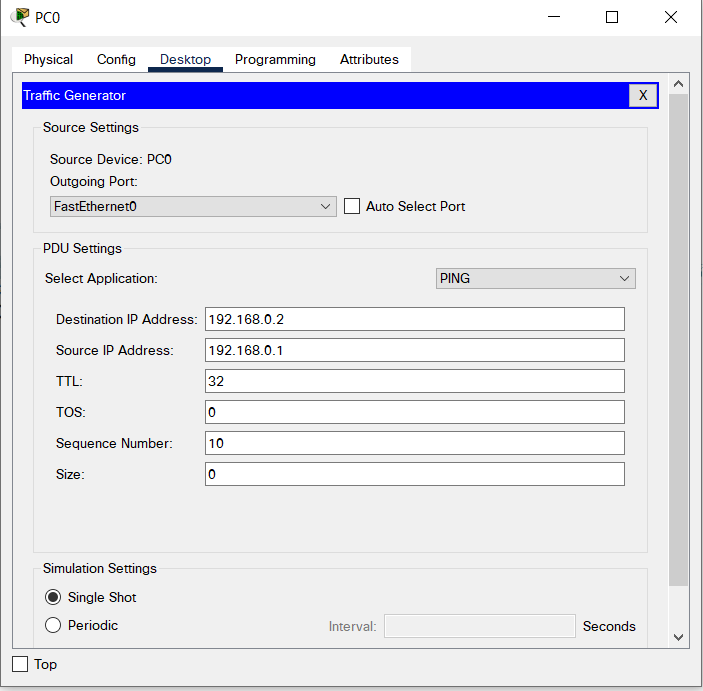


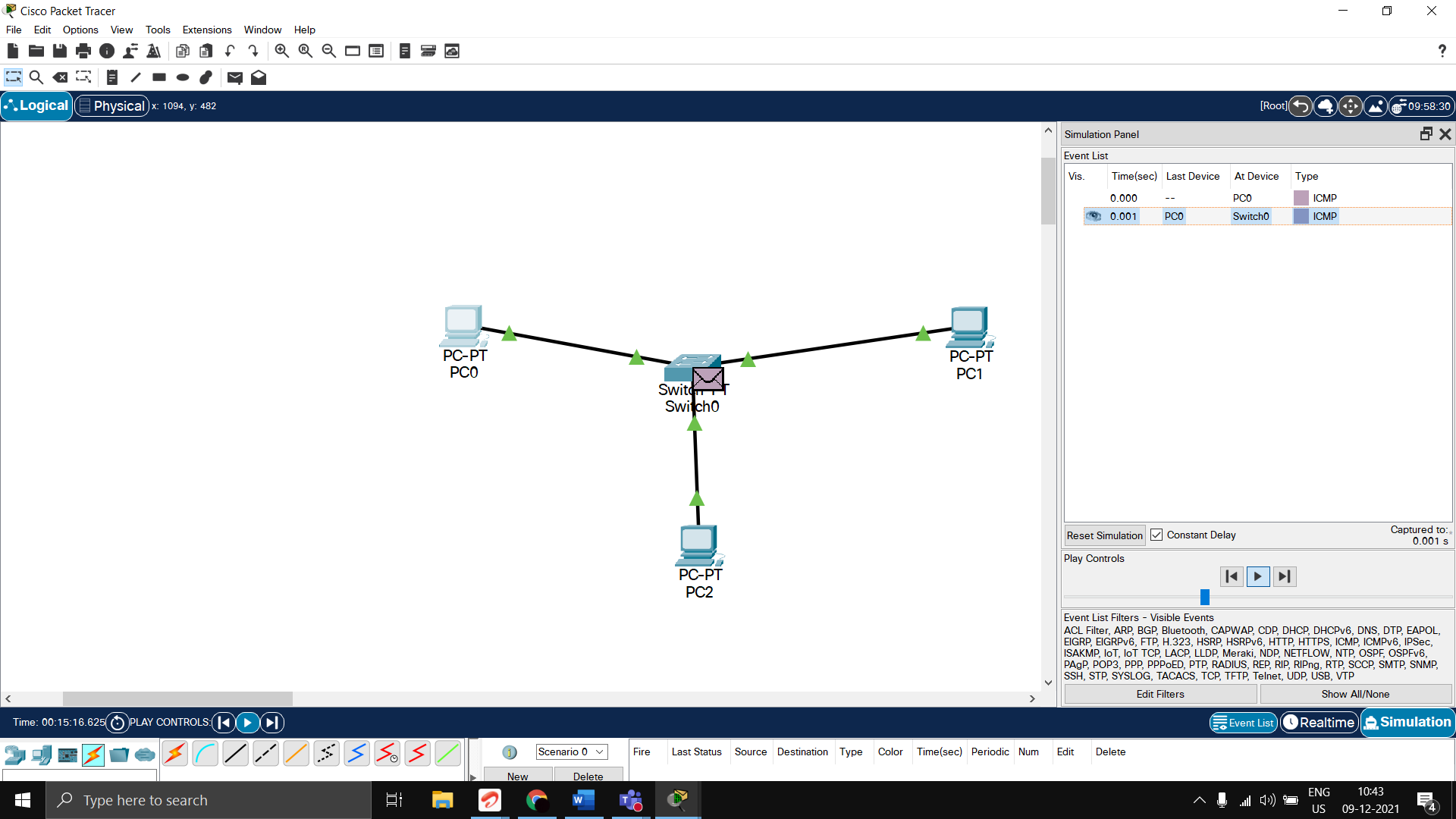
b) Using Simulation

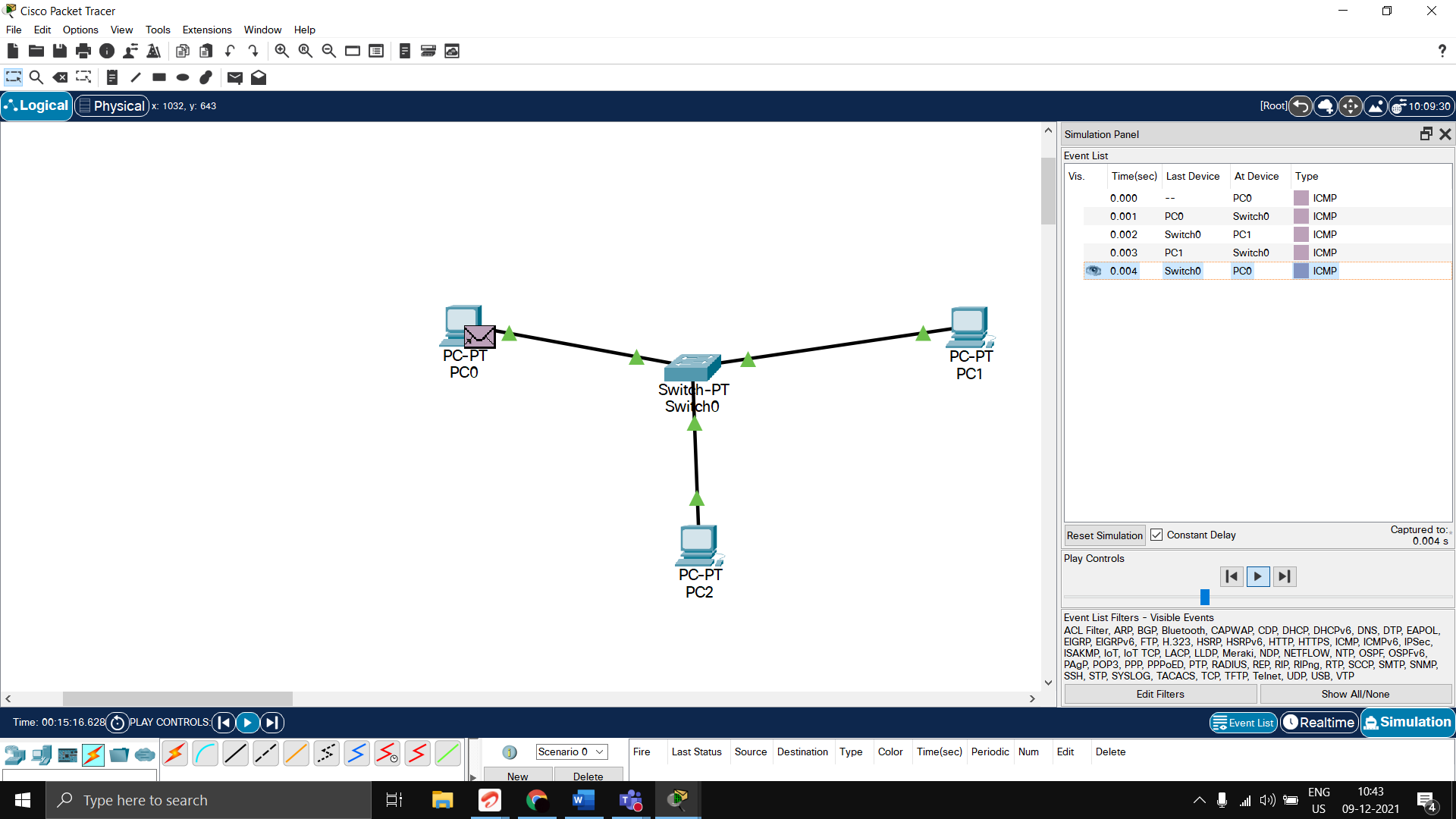


Configuring and adding a packet to be sent during simulation









# QUESTION 2a FULL CODE

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv('Malware\_dataset.csv')

dataset1 = dataset.copy()

dataset1 = dataset1.drop(['classification'], axis=1)

X=dataset1.iloc[:,1:].values

y=dataset.iloc[:,2].values

print (X)

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

y = le.fit\_transform(y)

print (y)

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=0)

print("Xtrain")

print(X\_train)

print("Xtest")

print(X\_test)

print("ytrain")

print(y\_train)

print("ytest")

print(y\_test)

from sklearn.preprocessing import StandardScaler

sc= StandardScaler()

X\_train= sc.fit\_transform(X\_train)

X\_test= sc.transform(X\_test)

print(X\_train)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

classifier.fit(X\_train,y\_train)

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

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))

from sklearn.metrics import confusion\_matrix, accuracy\_score

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

print(cm)

accuracy\_score(y\_test, y\_pred)

from sklearn.metrics import confusion\_matrix, precision\_score

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

print(cm)

precision\_score(y\_test, y\_pred)

from sklearn.metrics import confusion\_matrix, recall\_score

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

print(cm)

recall\_score(y\_test, y\_pred)