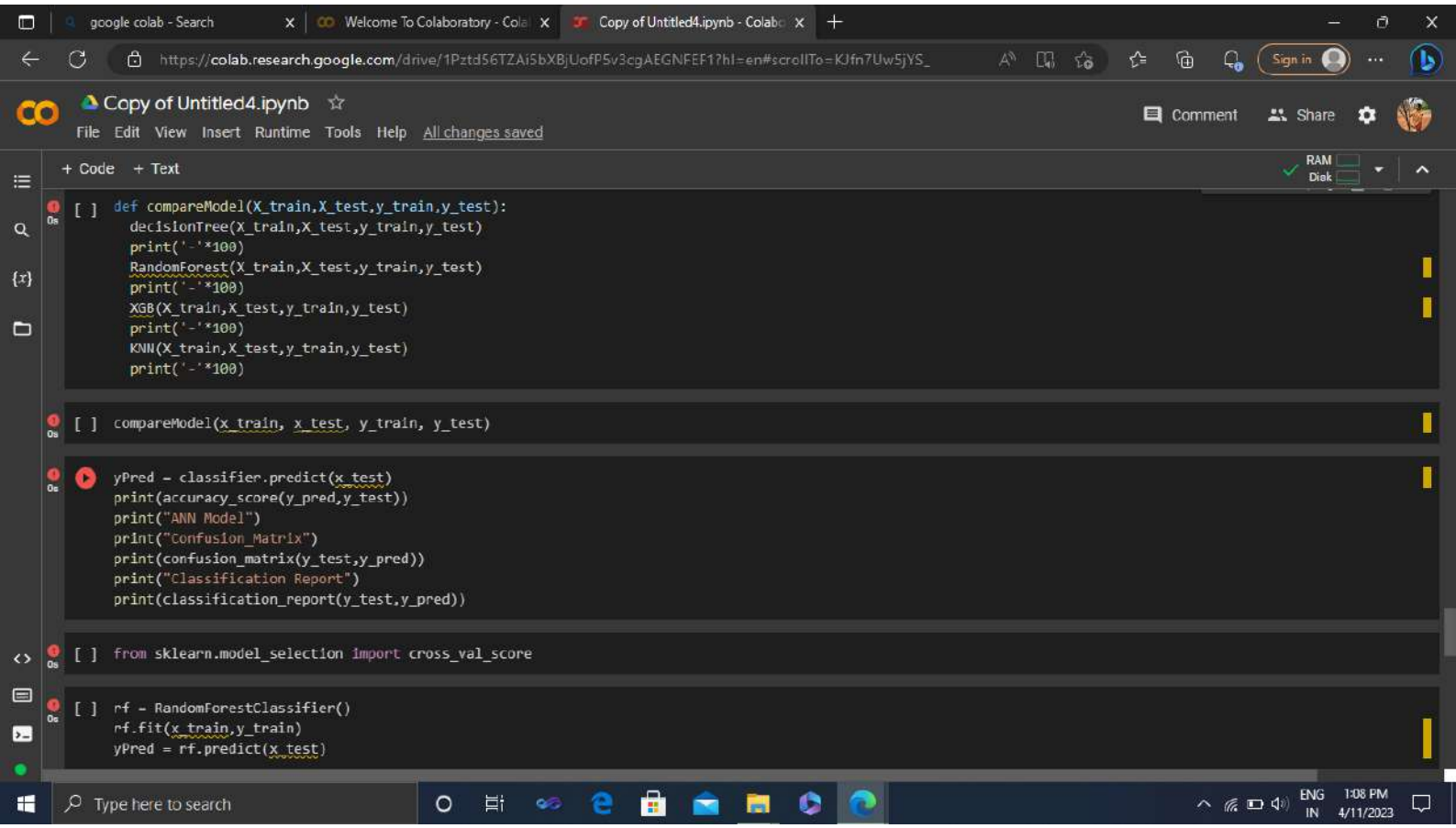
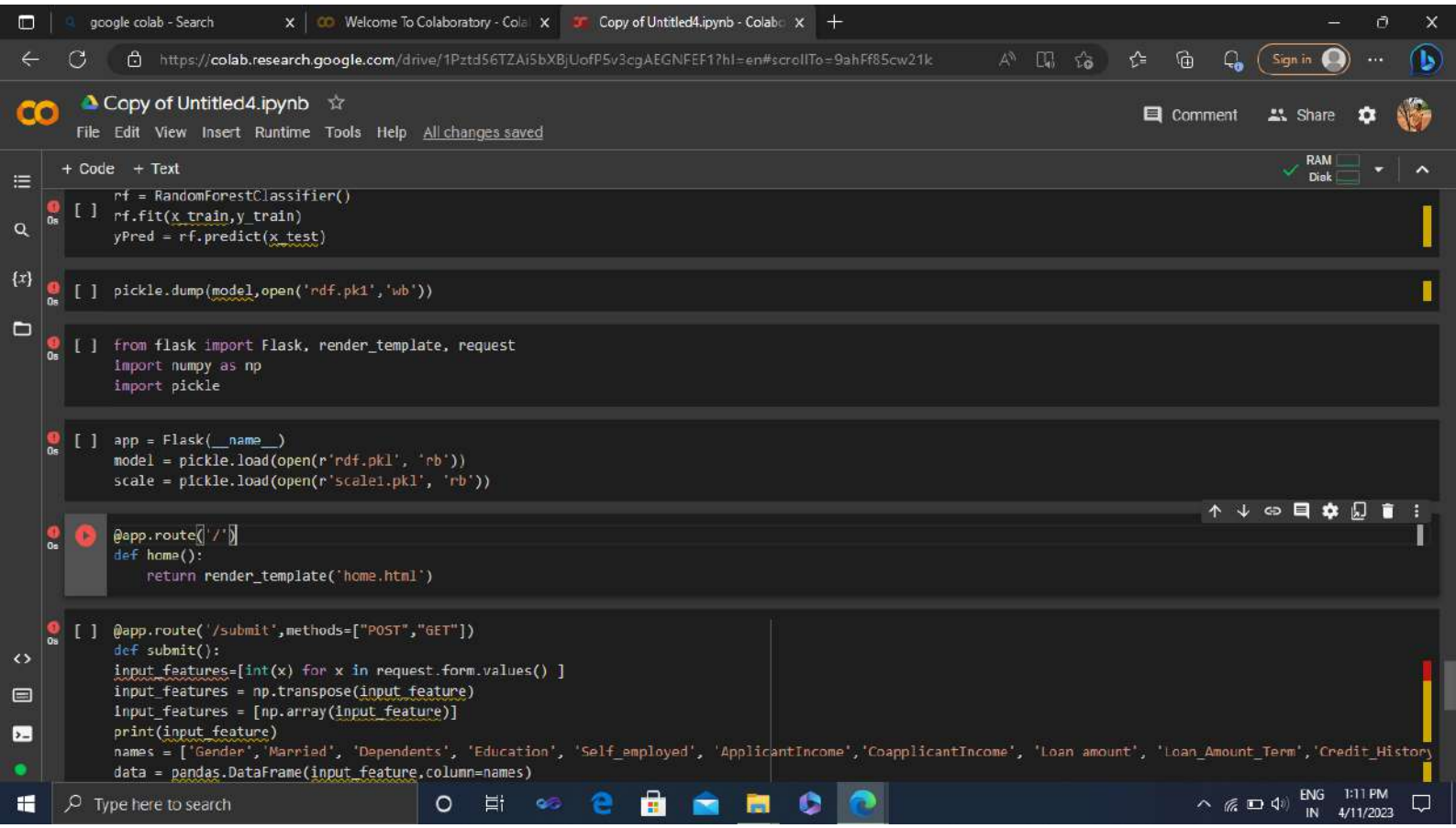

google colab - Search
Welcome To Colaboratory - Cola
Copy of Untitled4.ipynb - Colabo
https://colab.research.google.com/drive/1Pztd56TZAi5bXBjUofP5v3cgAEGNFEF1?hl=en#scrollTo=KJfn7Uw5jYS_
Sign in
Copy of Untitled4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[ ] def compareModel(X_train,X_test,y_train,y_test):
decisionTree(X_train,X_test,y_train,y_test)
print('-'*100)
RandomForest(X_train,X_test,y_train,y_test)
print('-'*100)
XGB(X_train,X_test,y_train,y_test)
print('-'*100)
KNN(X_train,X_test,y_train,y_test)
print('-'*100)
[ ] compareModel(x_train, x_test, y_train, y_test)
yPred = classifier.predict(x_test)
print(accuracy_score(y_pred,y_test))
print("ANN Model")
print("Confusion Matrix")
print(confusion_matrix(y_test,y_pred))
print("Classification Report")
print(classification_report(y_test,y_pred))
[ ] from sklearn.model_selection import cross_val_score
[ ] rf = RandomForestClassifier()
rf.fit(x_train,y_train)
yPred = rf.predict(x_test)
Type here to search
ENG
IN
1:08 PM
4/11/2023


google colab - Search
Welcome To Colaboratory - Cola
Copy of Untitled4.ipynb - Colabo
https://colab.research.google.com/drive/1Pztd56TZAi5bXBjUofP5v3cgAEGNFEF1?hl=en#scrollTo=9ahFf85cw21k
Sign in
Copy of Untitled4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
yPred = rf.predict(x_test)
pickle.dump(model,open('rdf.pk1','wb'))
from flask import Flask, render_template, request
import numpy as np
import pickle
app = Flask(__name__)
model = pickle.load(open(r'rdf.pkl', 'rb'))
scale = pickle.load(open(r'scale1.pkl', 'rb'))
@app.route('/')
def home():
return render_template('home.html')
@app.route('/submit',methods=["POST","GET"])
def submit():
input_features=[int(x) for x in request.form.values()]
input_features = np.transpose(input_feature)
Input_features = [np.array(input_feature)]
print(input_feature)
names = ['Gender','Married', 'Dependents', 'Education', 'Self_employed', 'ApplicantIncome','CoapplicantIncome', 'Loan amount', 'Loan_Amount_Term','Credit_History
data = pandas.DataFrame(input_feature,column=names)
Type here to search
ENG
IN
1:11 PM
4/11/2023