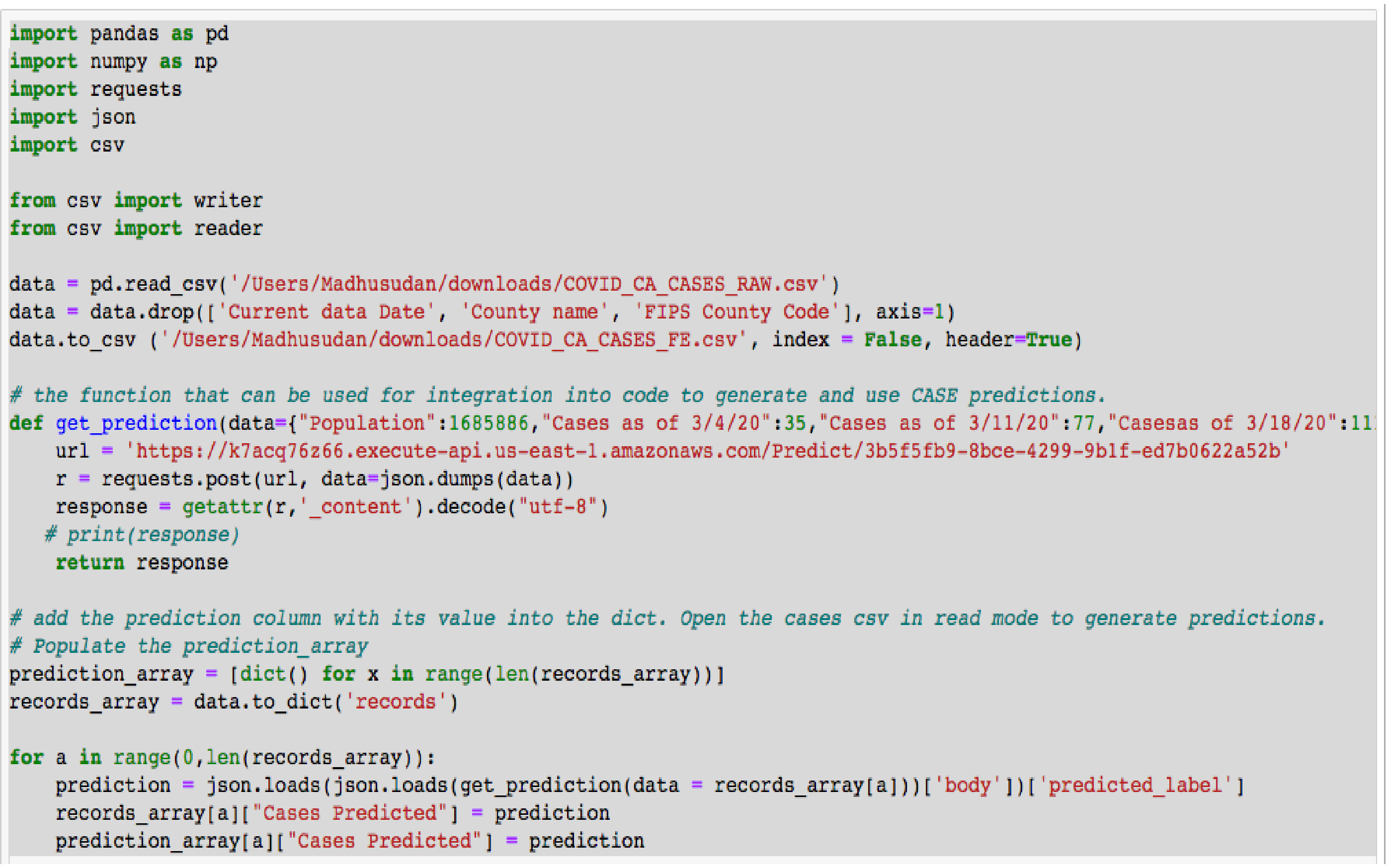
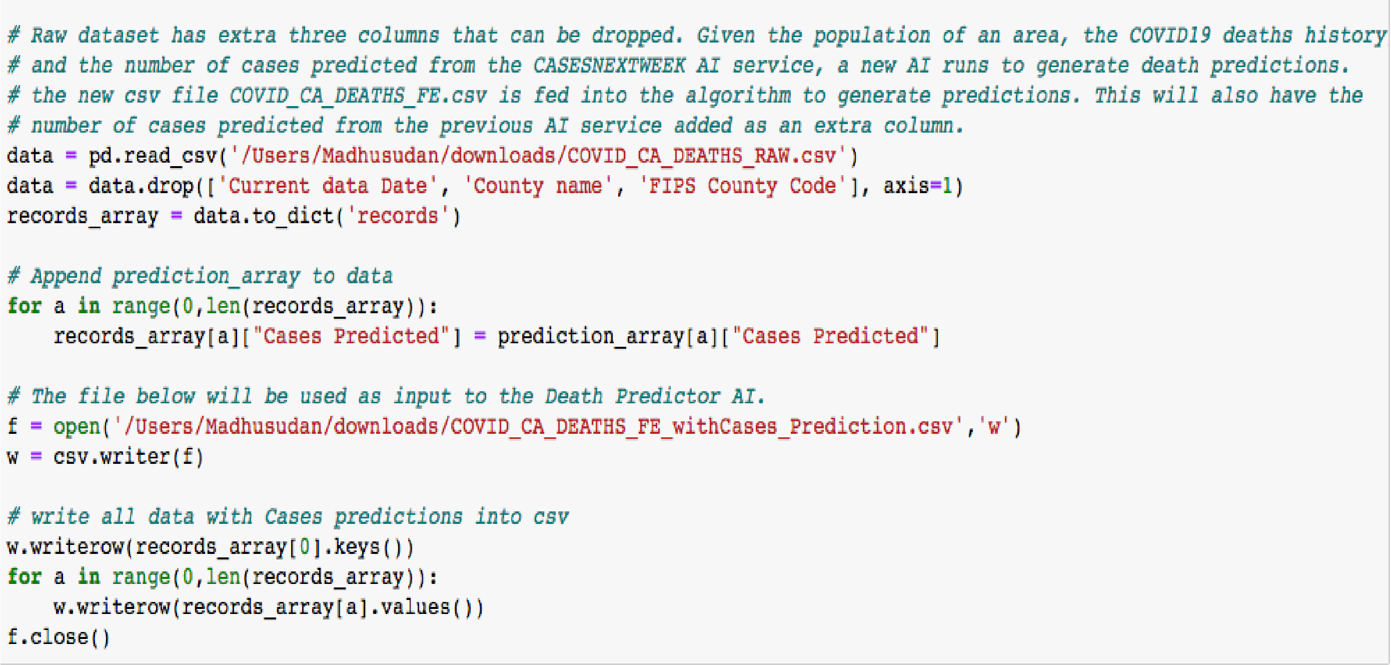
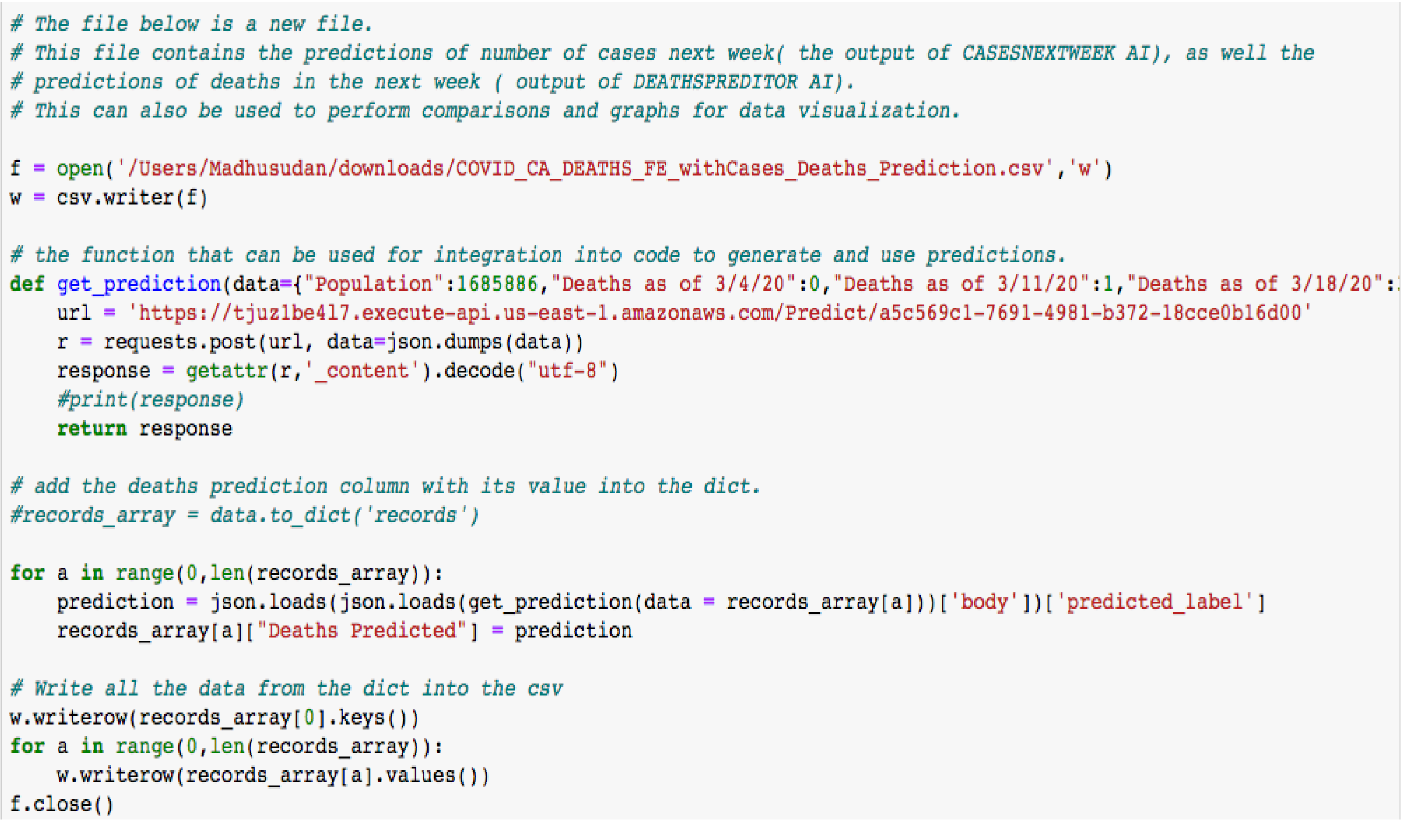


```python
import pandas as pd
import numpy as np
import requests
import json
import csv

from csv import writer
from csv import reader

data = pd.read_csv('/Users/Madhusudan/downloads/COVID_CA_CASES_RAW.csv')
data = data.drop(['Current data Date', 'County name', 'FIPS County Code'], axis=1)
data.to_csv ('/Users/Madhusudan/downloads/COVID_CA_CASES_FE.csv', index = False, header=True)

# the function that can be used for integration into code to generate and use CASE predictions.
def get_prediction(data={"Population":1685886,"Cases as of 3/4/20":35,"Cases as of 3/11/20":77,"Casesas of 3/18/20":11
    url = 'https://k7acq76z66.execute-api.us-east-1.amazonaws.com/Predict/3b5f5fb9-8bce-4299-9b1f-ed7b0622a52b'
    r = requests.post(url, data=json.dumps(data))
    response = getattr(r,'_content').decode("utf-8")
   # print(response)
    return response

# add the prediction column with its value into the dict. Open the cases csv in read mode to generate predictions.
# Populate the prediction_array
prediction_array = [dict() for x in range(len(records_array))]
records_array = data.to_dict('records')

for a in range(0,len(records_array)):
    prediction = json.loads(json.loads(get_prediction(data = records_array[a]))['body'])['predicted_label']
    records_array[a]["Cases Predicted"] = prediction
    prediction_array[a]["Cases Predicted"] = prediction
```



```python
# Raw dataset has extra three columns that can be dropped. Given the population of an area, the COVID19 deaths history
# and the number of cases predicted from the CASESNEXTWEEK AI service, a new AI runs to generate death predictions.
# the new csv file COVID_CA_DEATHS_FE.csv is fed into the algorithm to generate predictions. This will also have the
# number of cases predicted from the previous AI service added as an extra column.
data = pd.read_csv('/Users/Madhusudan/downloads/COVID_CA_DEATHS_RAW.csv')
data = data.drop(['Current data Date', 'County name', 'FIPS County Code'], axis=1)
records_array = data.to_dict('records')

# Append prediction_array to data
for a in range(0,len(records_array)):
    records_array[a]["Cases Predicted"] = prediction_array[a]["Cases Predicted"]

# The file below will be used as input to the Death Predictor AI.
f = open('/Users/Madhusudan/downloads/COVID_CA_DEATHS_FE_withCases_Prediction.csv','w')
w = csv.writer(f)

# write all data with Cases predictions into csv
w.writerow(records_array[0].keys())
for a in range(0,len(records_array)):
    w.writerow(records_array[a].values())
f.close()
```



```python
# The file below is a new file.
# This file contains the predictions of number of cases next week( the output of CASESNEXTWEEK AI), as well the
# predictions of deaths in the next week ( output of DEATHSPREDITOR AI).
# This can also be used to perform comparisons and graphs for data visualization.

f = open('/Users/Madhusudan/downloads/COVID_CA_DEATHS_FE_withCases_Deaths_Prediction.csv','w')
w = csv.writer(f)

# the function that can be used for integration into code to generate and use predictions.
def get_prediction(data={"Population":1685886,"Deaths as of 3/4/20":0,"Deaths as of 3/11/20":1,"Deaths as of 3/18/20":
    url = 'https://tjuz1be417.execute-api.us-east-1.amazonaws.com/Predict/a5c569c1-7691-4981-b372-18cce0b16d00'
    r = requests.post(url, data=json.dumps(data))
    response = getattr(r,'_content').decode("utf-8")
    #print(response)
    return response

# add the deaths prediction column with its value into the dict.
#records_array = data.to_dict('records')

for a in range(0,len(records_array)):
    prediction = json.loads(json.loads(get_prediction(data = records_array[a]))['body'])['predicted_label']
    records_array[a]["Deaths Predicted"] = prediction

# Write all the data from the dict into the csv
w.writerow(records_array[0].keys())
for a in range(0,len(records_array)):
    w.writerow(records_array[a].values())
f.close()
```