Security_Bots\TheBasicBot\trustbot.py

```
###############
# In this file the following should happen with the key requirements.
# Human agency and oversight: as it should include something so it's not looking it up but predicting it.
# Technical Robustness and safety: Safely transfer and receive information.
# Privacy and data governance: about the same as the last requirement but also not save or obfuscate who asked for this.
# Accountability: As how can the user contact somebody who can say what this bot does with the data and how it's processed.
#
# Some of these requirements cannot be fully implemented into how it can work because there is no platform where it can connect to.
# Or where it can send information to.
#
# Although I haven't read the GDPR fully yet I will try of what I know to implement it
#
###############

# Imports
import pickle
import pandas as pd
import hashlib

# Ask for the information
user_input_Age = 35
user_input_Sex = "M"
user_input_Country = "FR"

# Change it from charter to number
if user_input_Sex == "M":
    user_input_Sex = 1
else:
    user_input_Sex = 0

# Load the dictionary from the file for country code to number
with open('info/country_code_to_id.pkl', 'rb') as f:
    country_code_to_id = pickle.load(f)

# Transform the given country code to a hash
def md5hash(s: str):
    return hashlib.md5(s.encode('utf-8')).hexdigest() # or SHA, ...

hash_user_input_Country = md5hash(user_input_Country)

user_input_Country=country_code_to_id.get(hash_user_input_Country) # France as an example
print(user_input_Country)

# Create the user input
user_input = pd.DataFrame({
    'age': [user_input_Age],  # 35 years old
    'sexCode': [user_input_Sex],  # is a man
    'geo\\TIME_PERIOD_ID': [user_input_Country]  # lives in France
})

# Load the model
with open('info/model.pkl', 'rb') as f:
    model = pickle.load(f)

```

```
# load the target variable for the DataFrame at the end
with open('info/targets.pkl', 'rb') as f:
    targets = pickle.load(f)

# Predict the deaths from 1990 till 2022
user_pred = model.predict(user_input)

user_prediction_df = pd.DataFrame(user_pred, columns=targets)
print(user_prediction_df)

send_json = {
    "Foreword": "This is a prediction on what amount of deaths could be between the years of 1990 and 2022. This is based on the provided age, sex, and country/region code",
    "DataFrame": user_prediction_df,
    "Contact": "When you have problems with the result or want to know more on how this works you can contact <xxx@xxx.xx>."
}

print(send_json)

# Further in this needs to be converted to activity stream json. Below you see the beginning of it.
# I'm not sure if I'm doing this right

base_activitypub_json = {
  "@context": "https://www.w3.org/ns/activitystreams",
  "summary": "Basic trust bot collection",
  "type": "Collection",
  "totalItems": 3
}

Foreword_item = {
    "type": "Note",
    "name": send_json["Foreword"]
}

DataFrame_item = {
    "type": "object",
    "name": send_json["DataFrame"]
}

Contact_item  = {
    "type": "Note",
    "name": send_json["Contact"]
}

items_json = {"items":  []}
items_json["items"].append(Foreword_item)
items_json["items"].append(DataFrame_item)
items_json["items"].append(Contact_item)

base_activitypub_json.update(items_json)
print("What will be send: \n",base_activitypub_json)
```