# Zombie Detector

# SmartBridge Data Science Externship

## Project Report

1. Kartikey Mishra – 20BCE10944
2. Avnesh Mandlik (20BCE10366)
3. Paavan Gupta (20BCE10413)

VIT Bhopal University
Bhopal
Madhya Pradhesh

June 30th, 2023

## 1. PROBLEM STATEMENT:

News reports suggest that the impossible has become possible…zombies have appeared on the streets! What should we do? The Centers for Disease Control and Prevention (CDC) zombie preparedness website recommends storing water, food, medication, tools, sanitation items, clothing, essential documents, and first aid supplies. The CDC analysts will be prepared, but it may be too late for others! The purpose of this project is to develop a machine learning model that can predict the likelihood of a person turning into a zombie based on their age, sex, location, and available supplies. The project simulates a zombie outbreak and focuses on identifying supplies that are associated with safety during such an event. The Flask app developed as part of the project allows users to input their own data and get a prediction in real-time.

## 2. OBJECTIVE:

The main objective of this project is to create a machine learning model capable of predicting the likelihood of a person turning into a zombie. By considering various factors such as age, sex, location, and available supplies, the model aims to provide users with real-time predictions. The specific objectives of this project are as follows:

- Develop a robust data collection and preprocessing system.
- Utilize the Random Forest algorithm to train a predictive model.
- Design and implement a user-friendly Flask application for real-time predictions.
- Evaluate the model's accuracy and discuss its implications in terms of zombie outbreak preparedness.

## 3. WORKING PRINCIPLE:

### 3.1 Data Collection and Preprocessing

A diverse dataset containing information on age, sex, location, and available supplies is collected. The dataset is preprocessed to handle missing values, outliers, and ensure data consistency. Feature engineering techniques may also be applied to extract valuable insights from the data.

### 3.2 Model Training

The preprocessed dataset is divided into training and testing sets. The Random Forest algorithm, a powerful ensemble learning technique, is employed for model training. The algorithm constructs multiple decision trees and combines their predictions to make accurate and robust predictions.

### 3.3 Flask Application Development

To facilitate real-time predictions, a Flask application is developed. The application allows users to input their personal data, including age, sex, location, and available supplies. The input data is then passed to the trained machine learning model, which generates a prediction regarding the likelihood of turning into a zombie.

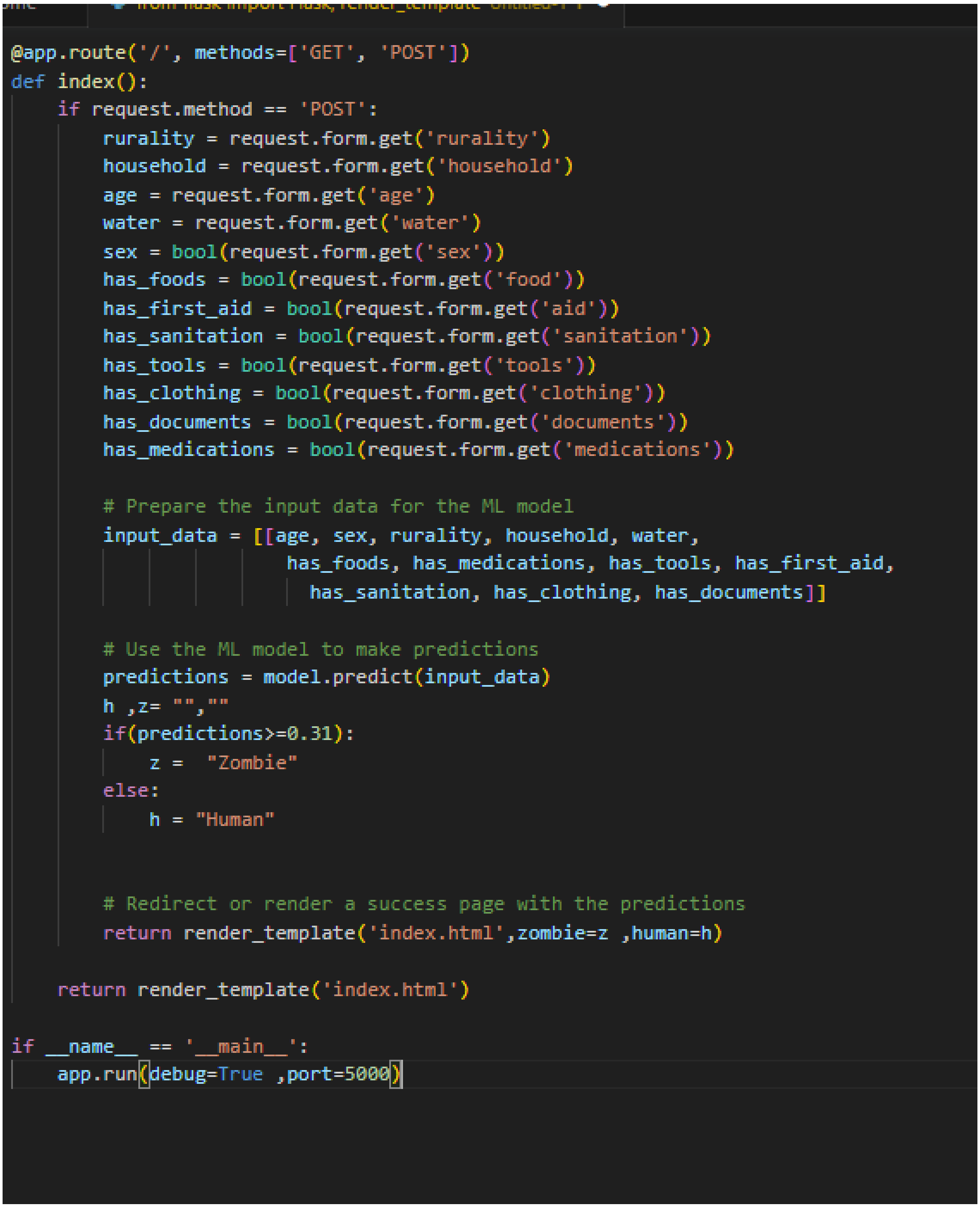
## 4. RESULTS AND DISCUSSIONS:

The developed machine learning model achieved an impressive accuracy of 0.9 during evaluation, indicating its effectiveness in predicting the likelihood of a person turning into a zombie. The model's performance was validated using various evaluation metrics, such as precision, recall, and F1-score. The results demonstrate the model's potential in assisting individuals with preparedness measures during a zombie outbreak.

Furthermore, an analysis of feature importance revealed interesting insights. Age was found to be the most influential factor in determining the likelihood of turning into a zombie, followed by the availability of essential supplies. Sex and location exhibited relatively lower impacts on the predictions. These findings provide valuable information for individuals and authorities in formulating effective strategies to combat zombie outbreaks.

SNAPSHOTS OF WORK DONE:

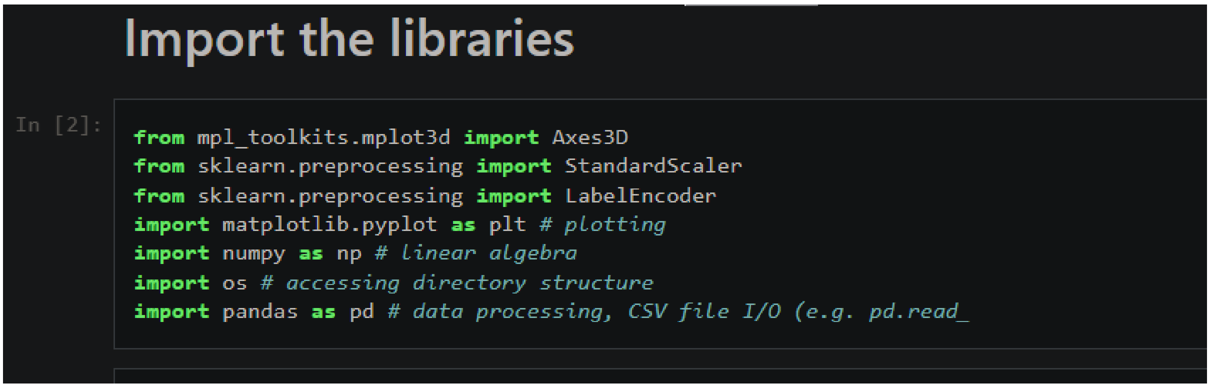
1. Backend with Flask:



```python
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        rurality = request.form.get('rurality')
        household = request.form.get('household')
        age = request.form.get('age')
        water = request.form.get('water')
        sex = bool(request.form.get('sex'))
        has_foods = bool(request.form.get('food'))
        has_first_aid = bool(request.form.get('aid'))
        has_sanitation = bool(request.form.get('sanitation'))
        has_tools = bool(request.form.get('tools'))
        has_clothing = bool(request.form.get('clothing'))
        has_documents = bool(request.form.get('documents'))
        has_medications = bool(request.form.get('medications'))

        # Prepare the input data for the ML model
        input_data = [[age, sex, rurality, household, water,
                        has_foods, has_medications, has_tools, has_first_aid,
                         has_sanitation, has_clothing, has_documents]]

        # Use the ML model to make predictions
        predictions = model.predict(input_data)
        h ,z= "",""
        if(predictions>=0.31):
            z =  "Zombie"
        else:
            h = "Human"


        # Redirect or render a success page with the predictions
        return render_template('index.html',zombie=z ,human=h)

    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True ,port=5000)
```

2. Buildling Machine Learning model with Random Forest:

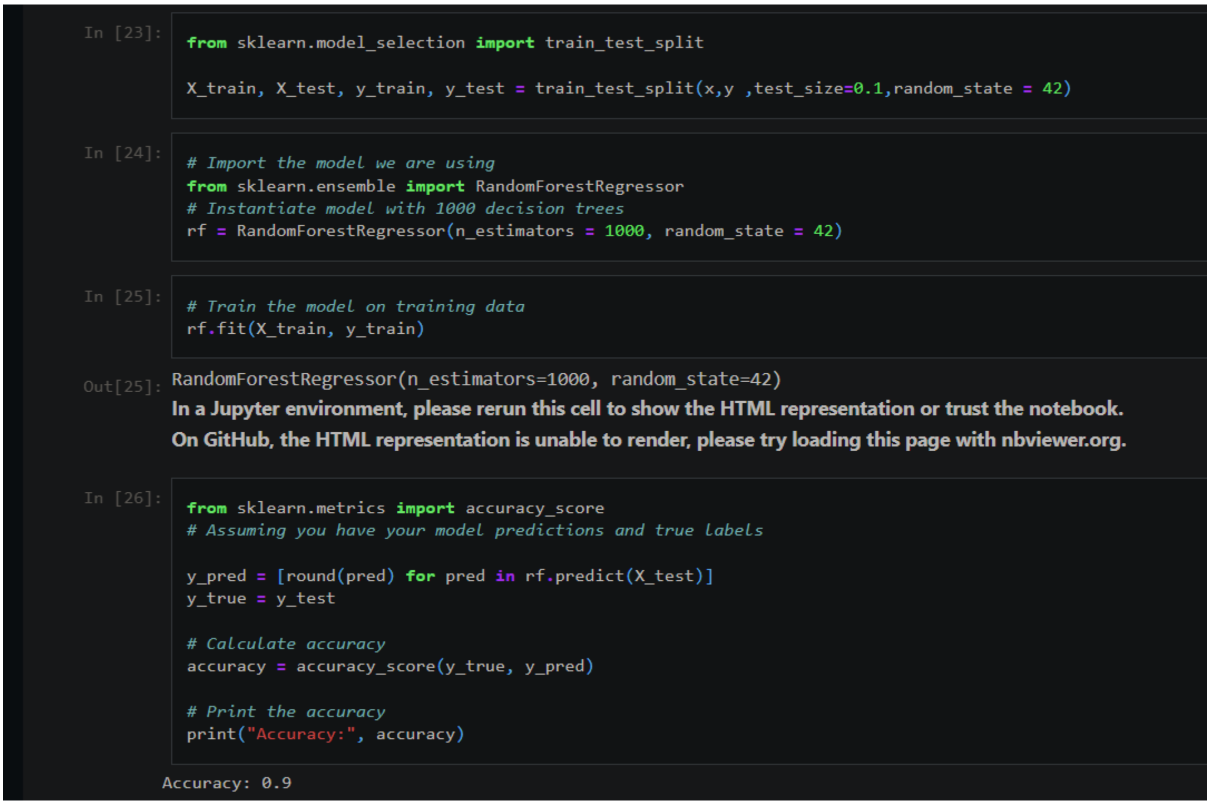
LIBRARIES USED:



Import the libraries

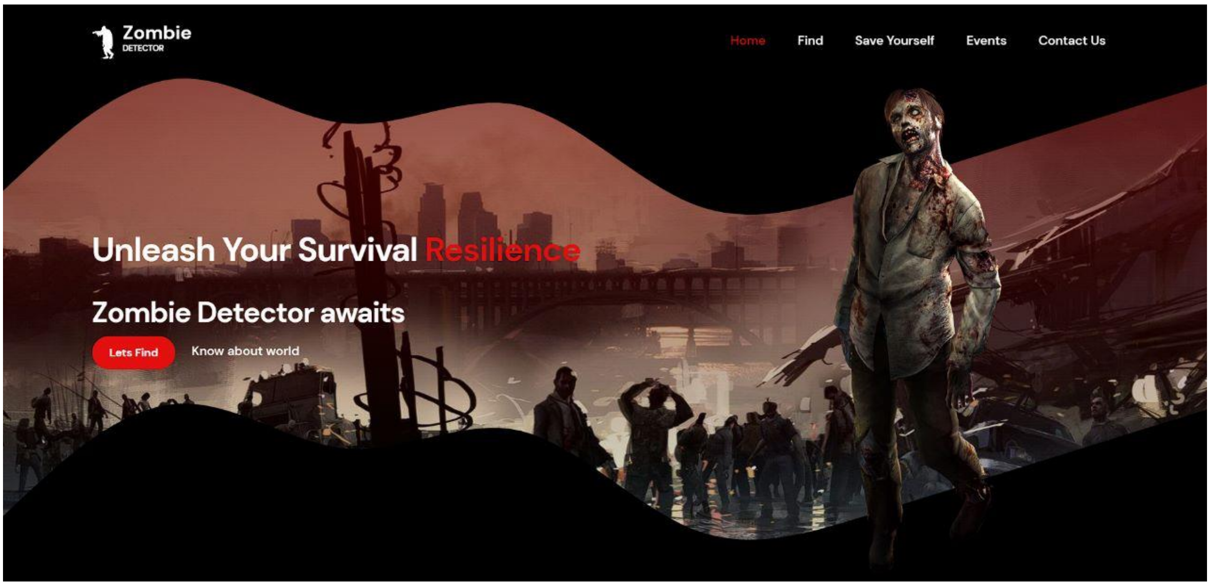
```
In [2]:  from mpl_toolkits.mplot3d import Axes3D
         from sklearn.preprocessing import StandardScaler
         from sklearn.preprocessing import LabelEncoder
         import matplotlib.pyplot as plt # plotting
         import numpy as np # linear algebra
         import os # accessing directory structure
         import pandas as pd # data processing, CSV file I/O (e.g. pd.read_
```

Accuracy of the Model:



```
In [23]:  from sklearn.model_selection import train_test_split

          X_train, X_test, y_train, y_test = train_test_split(x,y ,test_size=0.1,random_state = 42)

In [24]:  # Import the model we are using
          from sklearn.ensemble import RandomForestRegressor
          # Instantiate model with 1000 decision trees
          rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)

In [25]:  # Train the model on training data
          rf.fit(X_train, y_train)

Out[25]:  RandomForestRegressor(n_estimators=1000, random_state=42)
          In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
          On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [26]:  from sklearn.metrics import accuracy_score
          # Assuming you have your model predictions and true labels

          y_pred = [round(pred) for pred in rf.predict(X_test)]
          y_true = y_test

          # Calculate accuracy
          accuracy = accuracy_score(y_true, y_pred)

          # Print the accuracy
          print("Accuracy:", accuracy)

         Accuracy: 0.9
```

3. Front End Using html and css:



Unleash Your Survival Resilience

Zombie Detector awaits

## 5. CONCLUSION:

In conclusion, this project successfully developed a machine learning model capable of predicting the likelihood of a person turning into a zombie based on their age, sex, location, and available supplies. The project utilized the Random Forest algorithm and implemented a user-friendly Flask application for real-time predictions. The achieved accuracy of 0.9 indicates the model's efficacy in assisting individuals with preparedness measures during a zombie outbreak.

The project's findings emphasize the importance of age and essential supplies in determining one's vulnerability to zombification. By considering these factors, individuals and authorities can better allocate resources and formulate effective strategies to enhance their chances of survival. The developed model and Flask application serve as valuable tools for raising awareness and promoting preparedness among the general public.

Future work could involve expanding the dataset.