**ZOMBIE DETECTOR: Using ML To Save Lives During A Zombie Apocalypse**

Mini Project documentation submitted to

**JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD**

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

Submitted By

**AKHIL BOLLABOINA (21UK1A6750)**

**MANIKARNA JAMPALA (21UK1A6714)**

**PAVANI JAKKULA (21UK1A6713)**

**SRUTHI KATKURI (21UK1A6761)**

Under the guidance of

**DR REKHA GANGULA**

Asst.Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

**VAAGDEVI ENGINEERING COLLEGE**

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

**VAAGDEVI ENGINEERING COLLEGE WARANGAL**



**CERTIFICATE**

This is to certify that the Mini project report entitled “**ZOMBIE DETECTOR: Using ML To Save Lives During A Zombie Apocalypse**” is being submitted by **AKHIL BOLLABOINA(21UK1A6750), MANIKARNA JAMPALA(21UK1A67), PAVANI JAKKULA(21UK1A67), SRUTHI KANDURI (21UK1A6762)**, partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

**Project Guide**    **Head of the Department**

**G.REKHA Dr. K. SHARMILAREDDY**

**(Asst.professor) (Professor)**

**ACKNOWLEDGEMENT**

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this mini project in the institute.

We extend our heartfelt thanks to **Dr. K. SHARMILAREDDY,** Head of the Department of CSD, Vaagdevi Engineering College for providing us necessary infrastructure and there by giving us freedom to carry out the mini project.

We express heartfelt thanks to the guide **G.REKHA**, Asst. Professor, Department of CSD for her constant support and giving necessary guidance for completion of this mini project.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

**AKHIL BOLLABOINA (21UK1A6750)**

**MANIKARNA JAMPALA (21UK1A6714)**

**PAVANI JAKKULA (21UK1A6713)**

**SRUTHI KATKURI (21UK1A6761)**

**ZOMBIE DETECTOR: Using ML To Save Lives During A Zombie Apocalypse**

|  |  |
| --- | --- |
| TABLE OF CONTENTS | S.NO |
| 1. ZOMBIE DETECTOR: Using ML To Save Lives During A Zombie Apocalypse | **5** |
| 2. Define problem/problem understanding | **5** |
| 3. Data Collection &preparation | **6-10** |
| 4. Exploratory Data Analysis | **11-18** |
| 5. Model Building | **19** |
| 6. Performance Testing & Hyperparameter Tuning | **20-21** |
| 7. Model Building | **22-25** |
| 8. Project Demonstration & Documentation | **26** |
|  |  |
|  |  |
|  |  |

**ZOMBIE DETECTOR: Using ML To Save Lives During A Zombie Apocalypse**

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.

In summary, the purpose of this project is to use machine learning to identify supplies that are associated with safety during a zombie outbreak, raise awareness about emergency preparedness, and assist emergency responders in making informed decisions.

**Technical Architecture**

**Define Problem / Problem Understanding**

**Data Collection & Preparation**

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

**Collect The Dataset**

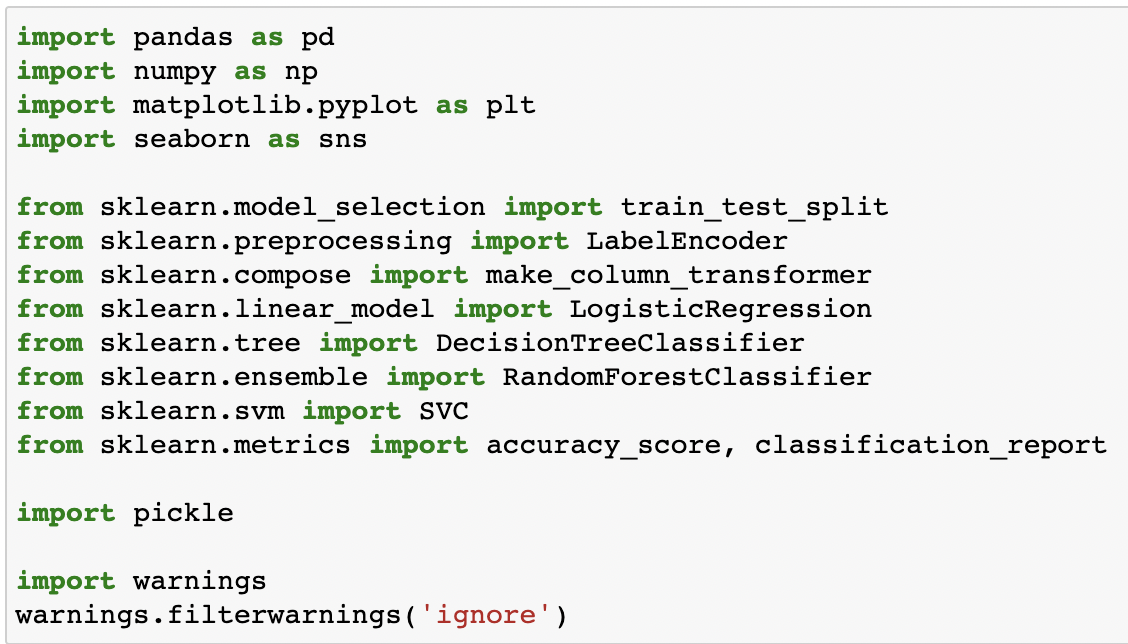
There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.  
In this project, we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link: <https://www.kaggle.com/datasets/kingabzpro/zombies-apocalypse>

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.  
Note: There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

**Importing the libraries**

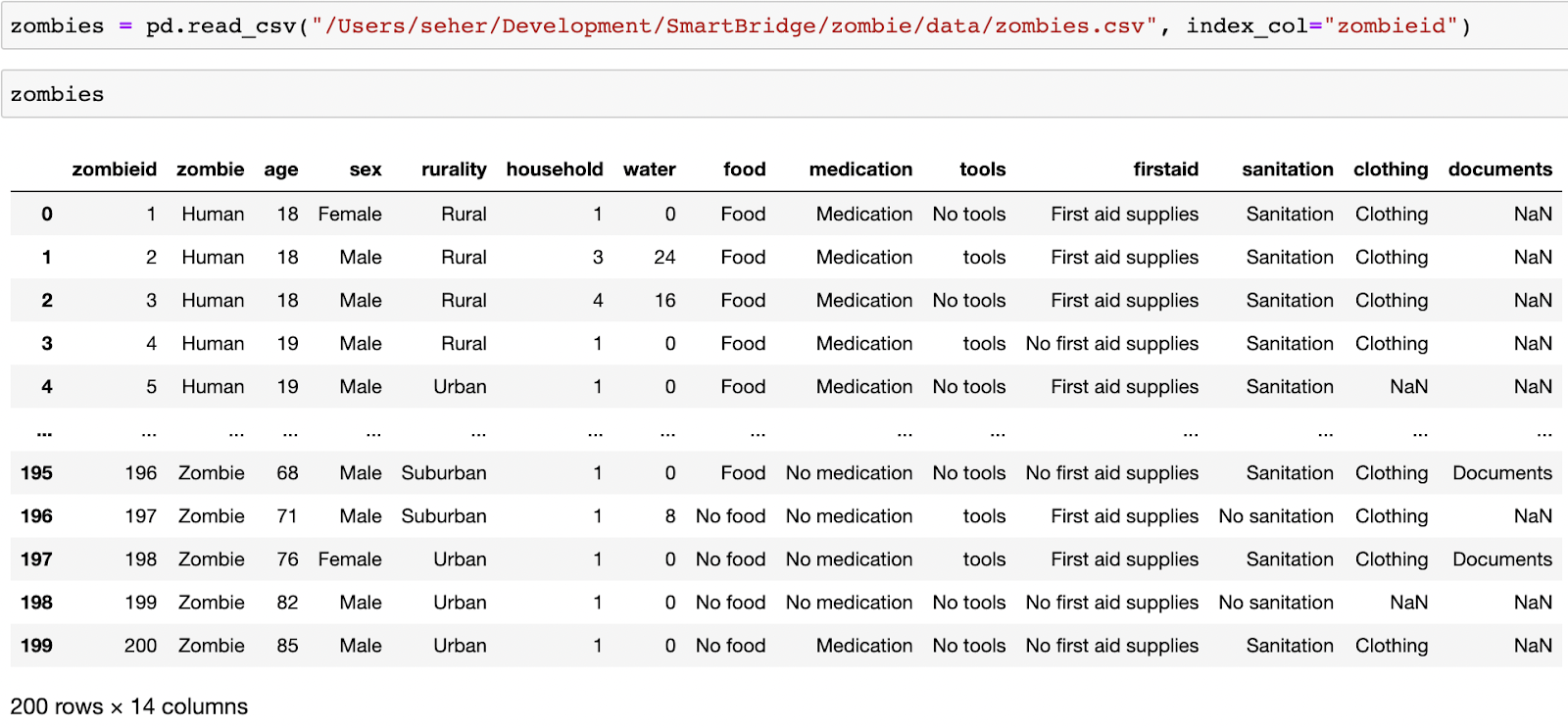
Import the necessary libraries as shown in the image.



**Read the Dataset**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas, we have a function called read\_csv() to read the dataset. As a parameter, we have to give the directory of the csv file.

****



Zombies Apocalypse | Kaggle..

To figure out which supplies are associated with safety in Zombie Apocalypse..

<https://www.kaggle.com/datasets/kingabzpro/zombies-apocalypse>

**Data Preparation**

As we have understood how the data is, let's pre-process the collected data.

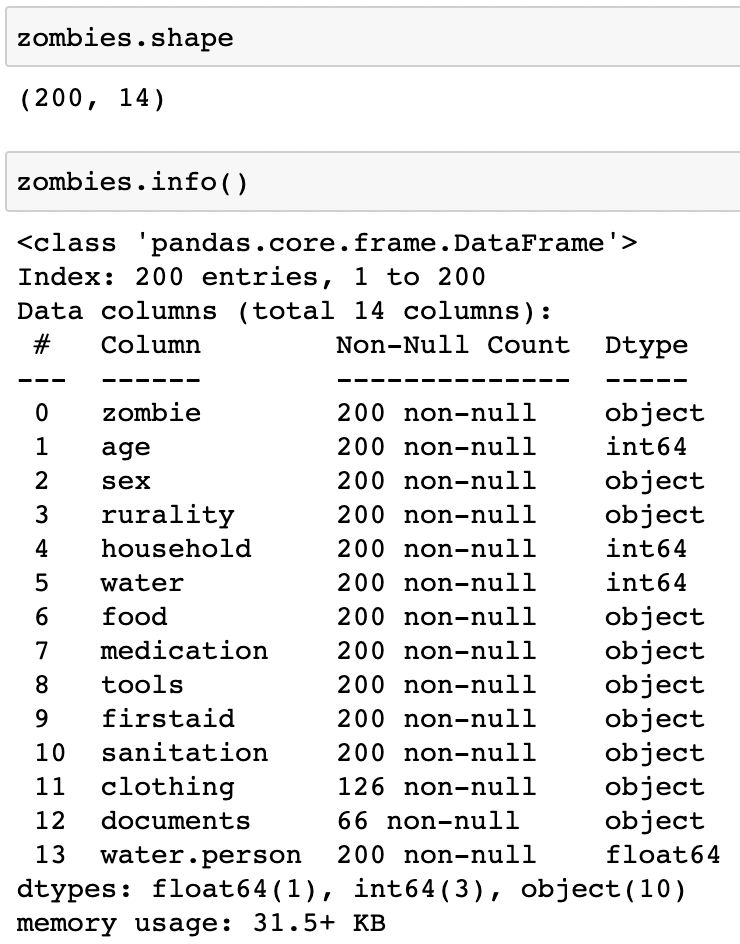
The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling categorical data

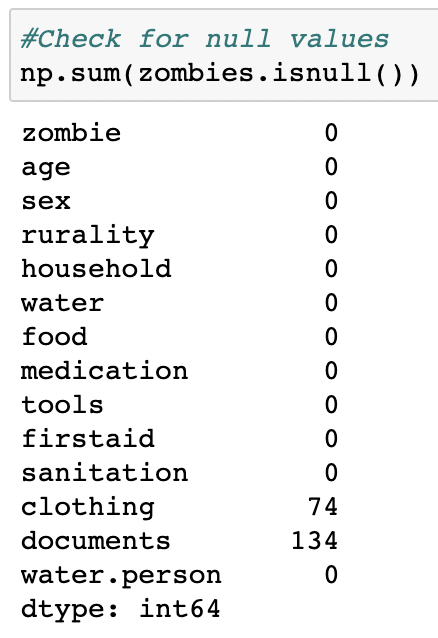
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

**Handling missing values**

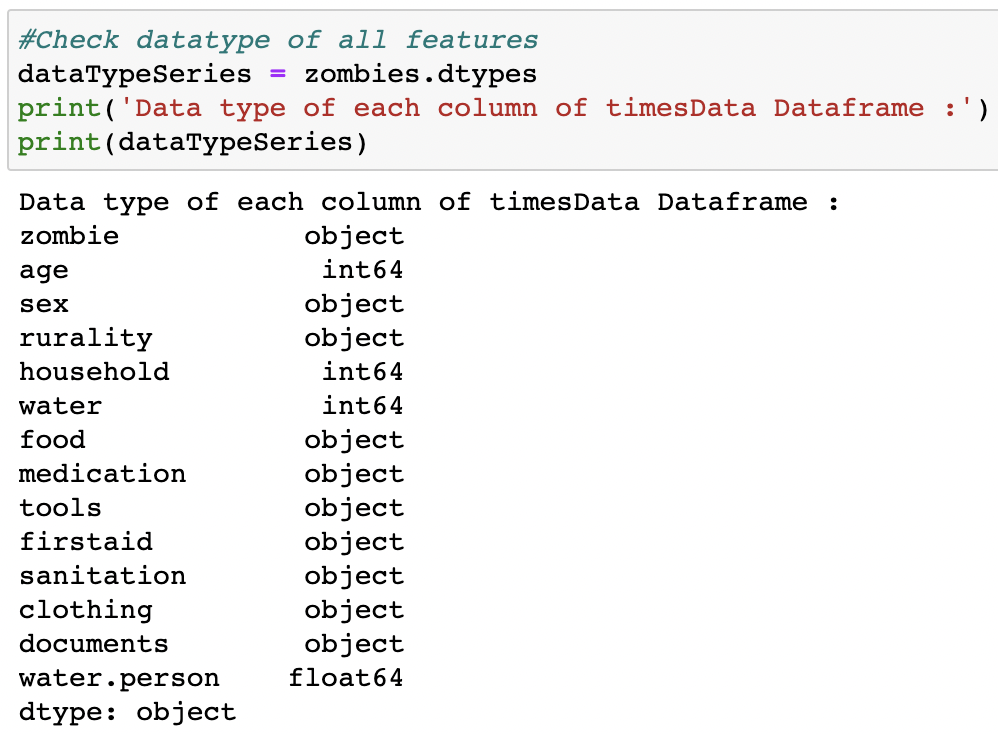
Let’s find the shape of our dataset first. To find the shape of our data, the .shape method is used. To find the data type, .info() function is used.

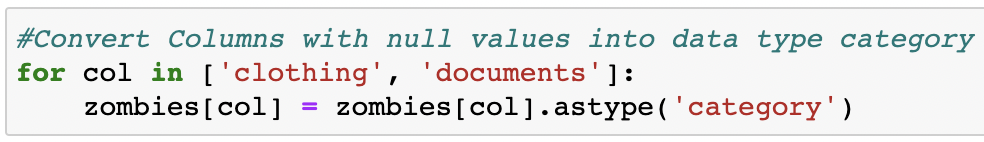


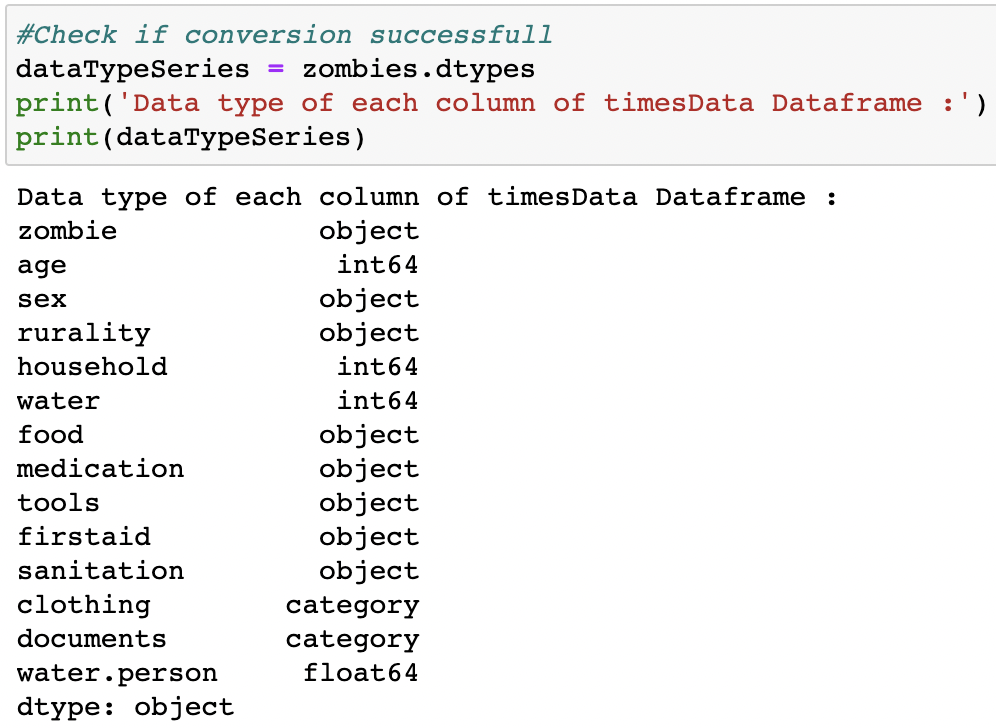
For checking the null values, .isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are 200 null values present in our dataset in broad\_impact. So we can drop the column.



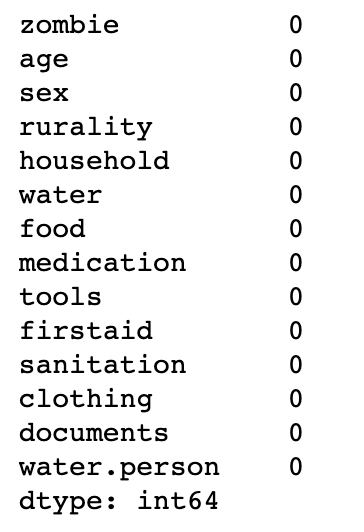
We can check for data types to find out categorical data





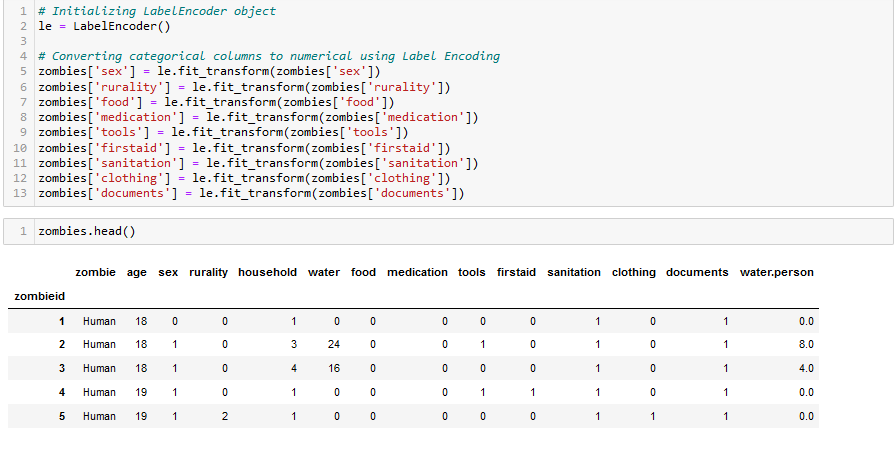






**Handling Categorical Values**

We will deal with categorical values after splitting the dataset into dependent and Independent variables



**Exploratory Data Analysis**

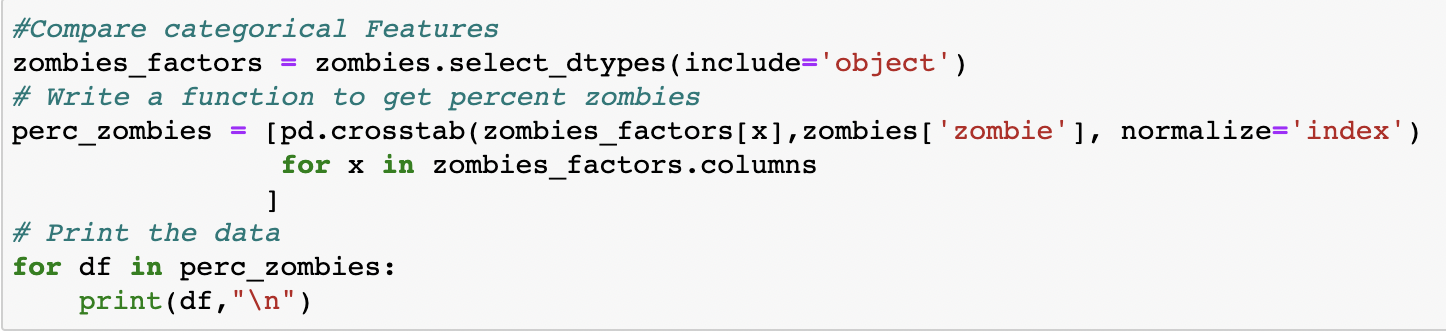
**Descriptive Statistical**

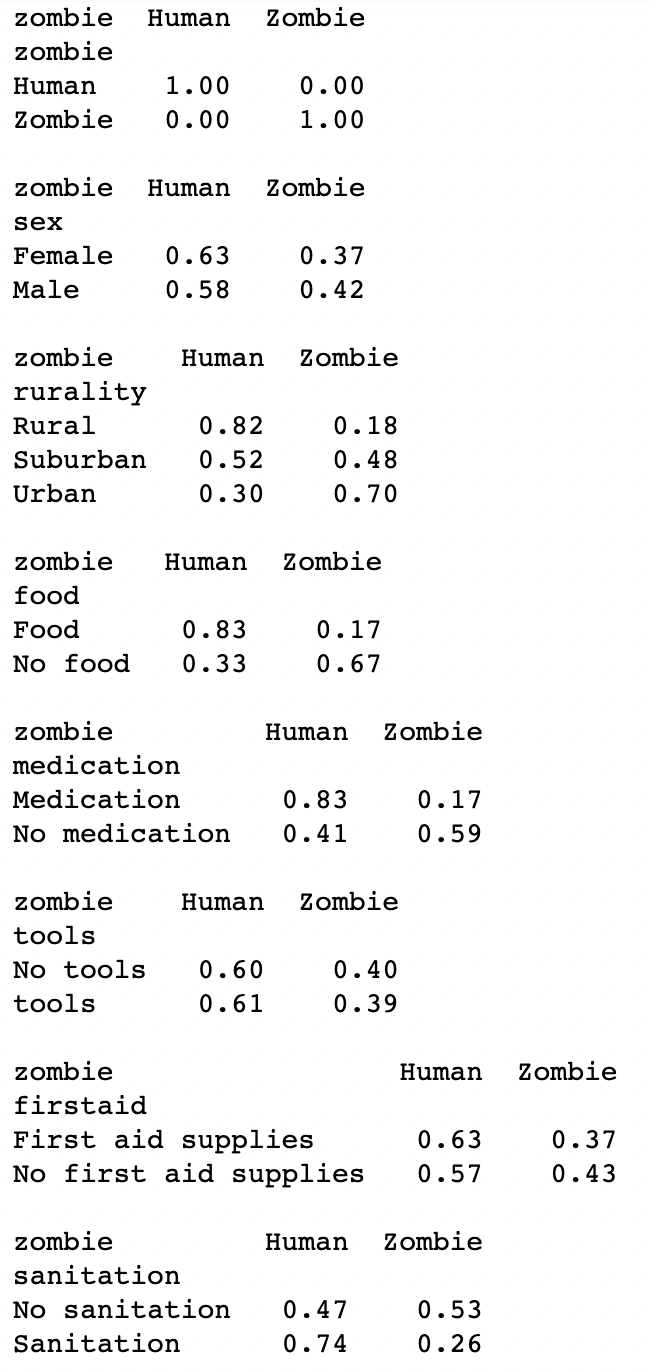
Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

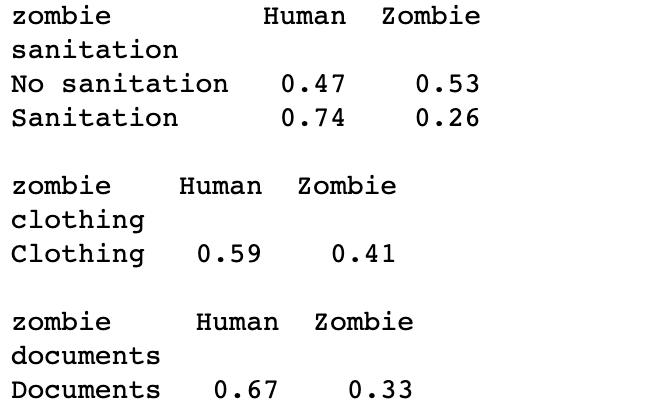


**Compare Humans and Zombies - Categorical features**

It is important to compare humans and zombies to identify differences in supplies. We need to identify the characteristics and supplies that differ between humans and zombies? Do zombies live in urban areas? Or are they more common in rural areas? Is water critical to staying human? Is food critical to staying human?







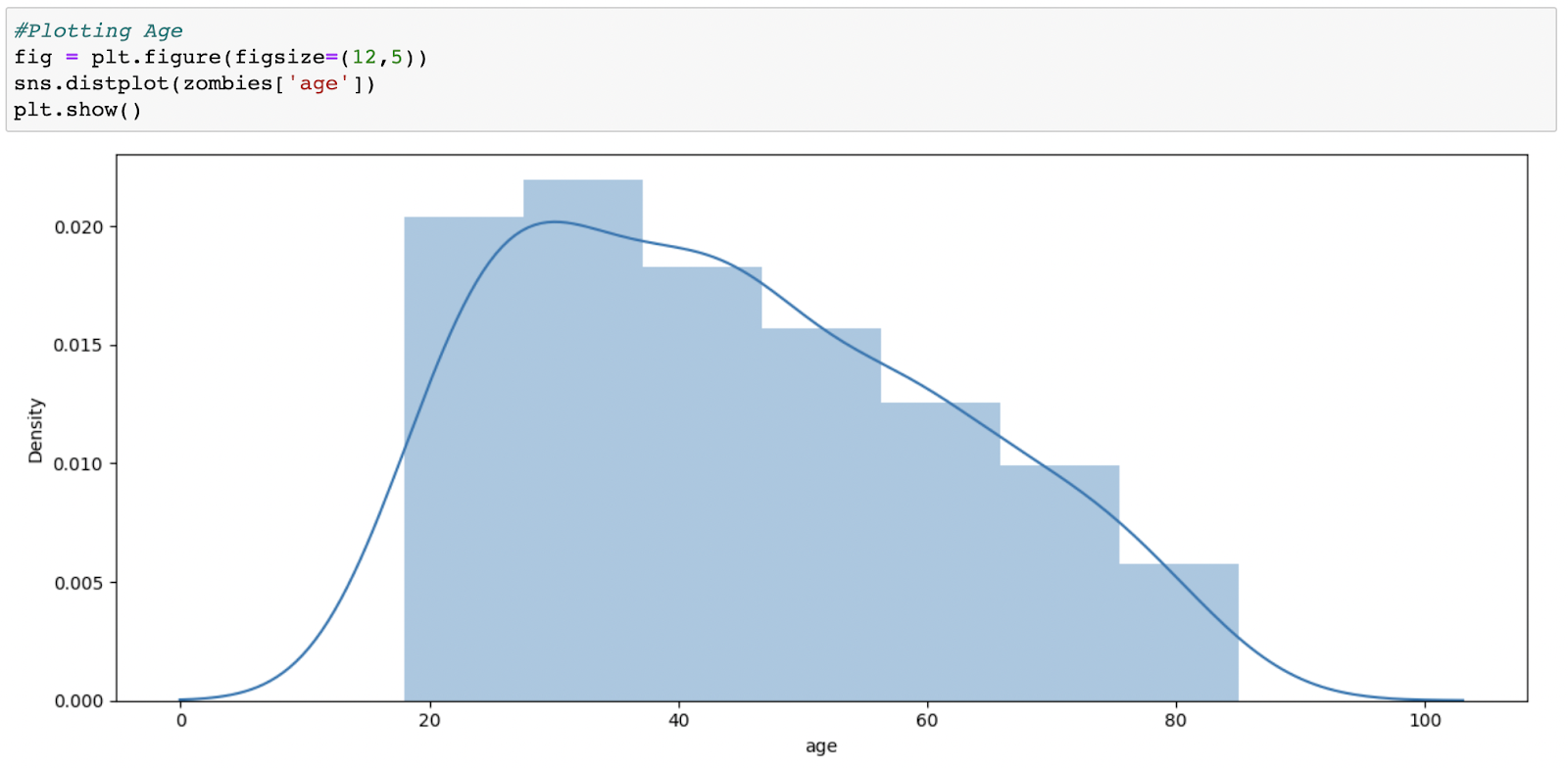
**Visual Analysis**

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions. We will be using seaborn and plotly packages from python to do so.

\

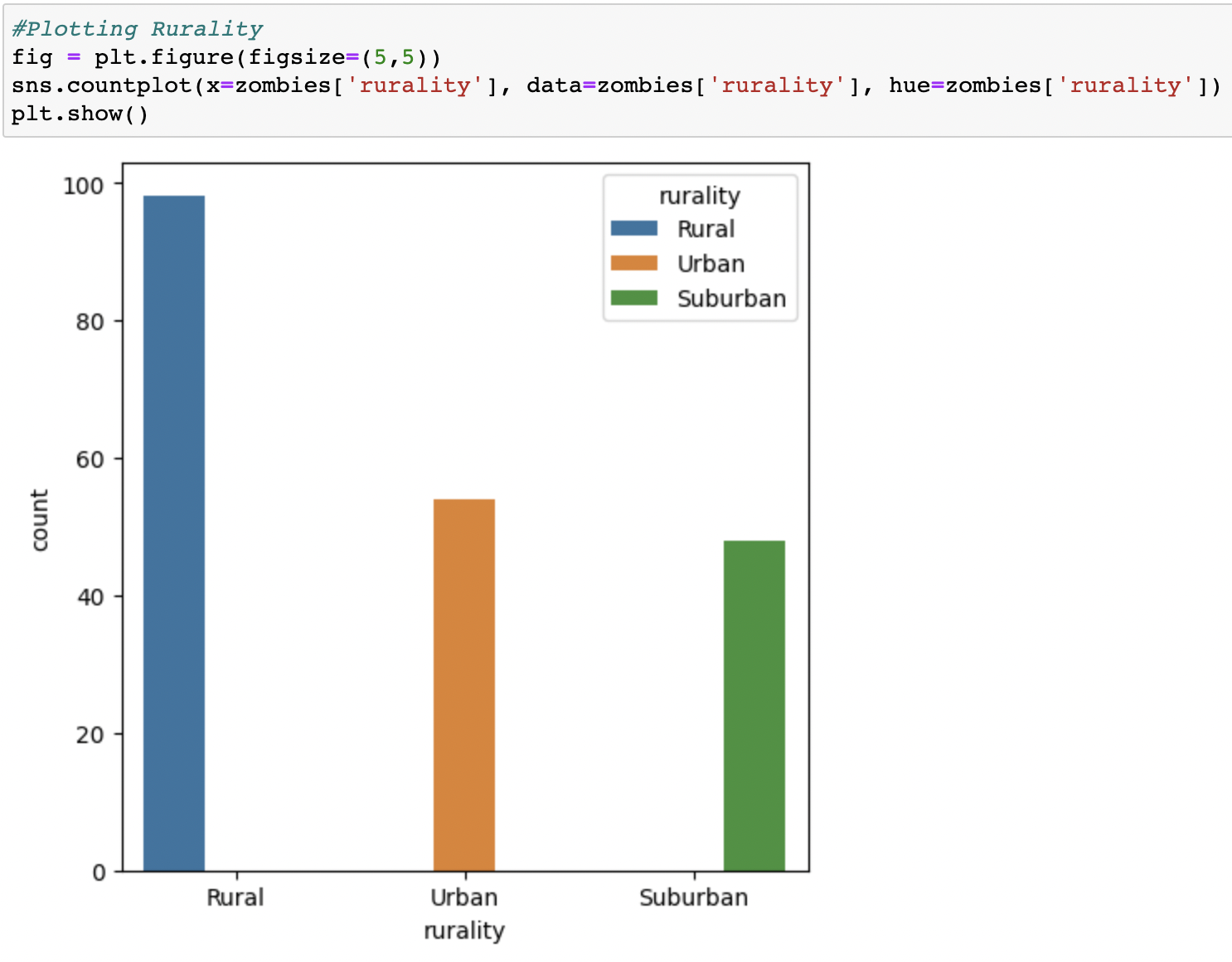
**Univariate analysis**

Seaborn package provides the distplot function. With the help of distplot, we can find the age distribution among the people.



From the plot we can see that most candidates are between the age 20 to 40.

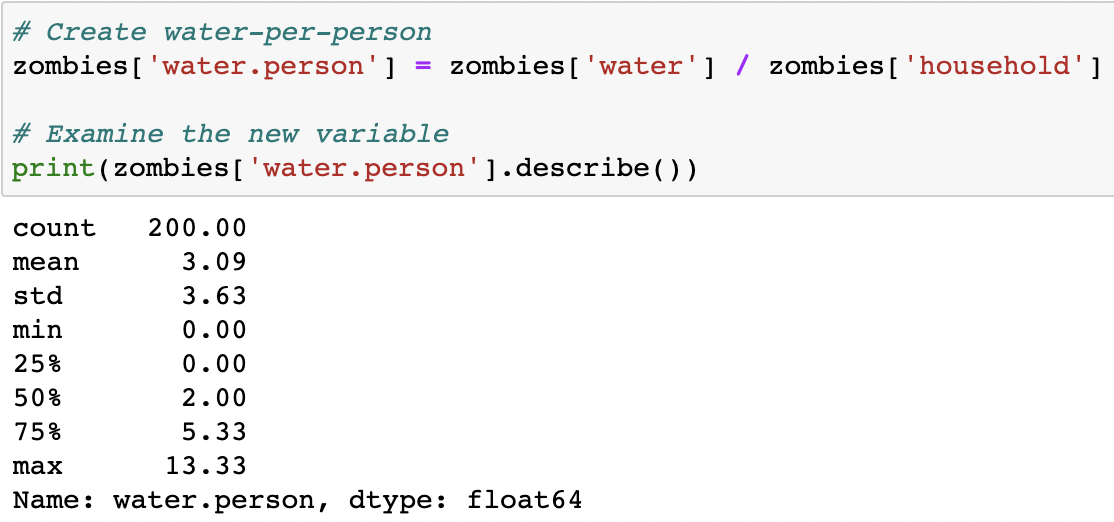
Plotting categorical data Rurality



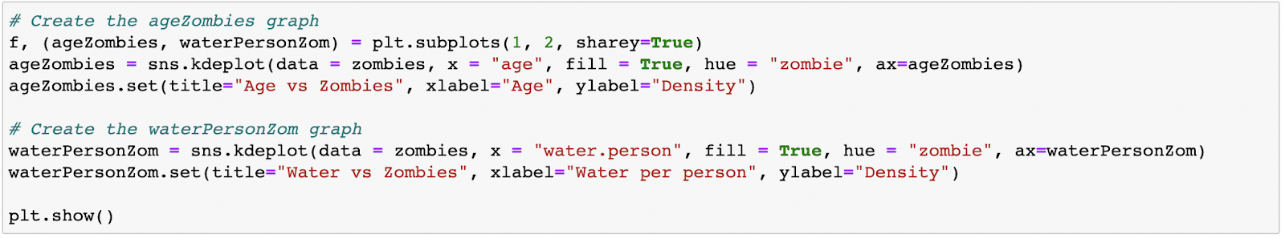
From the above graph, we can see that most of the candidates are rural residentials.

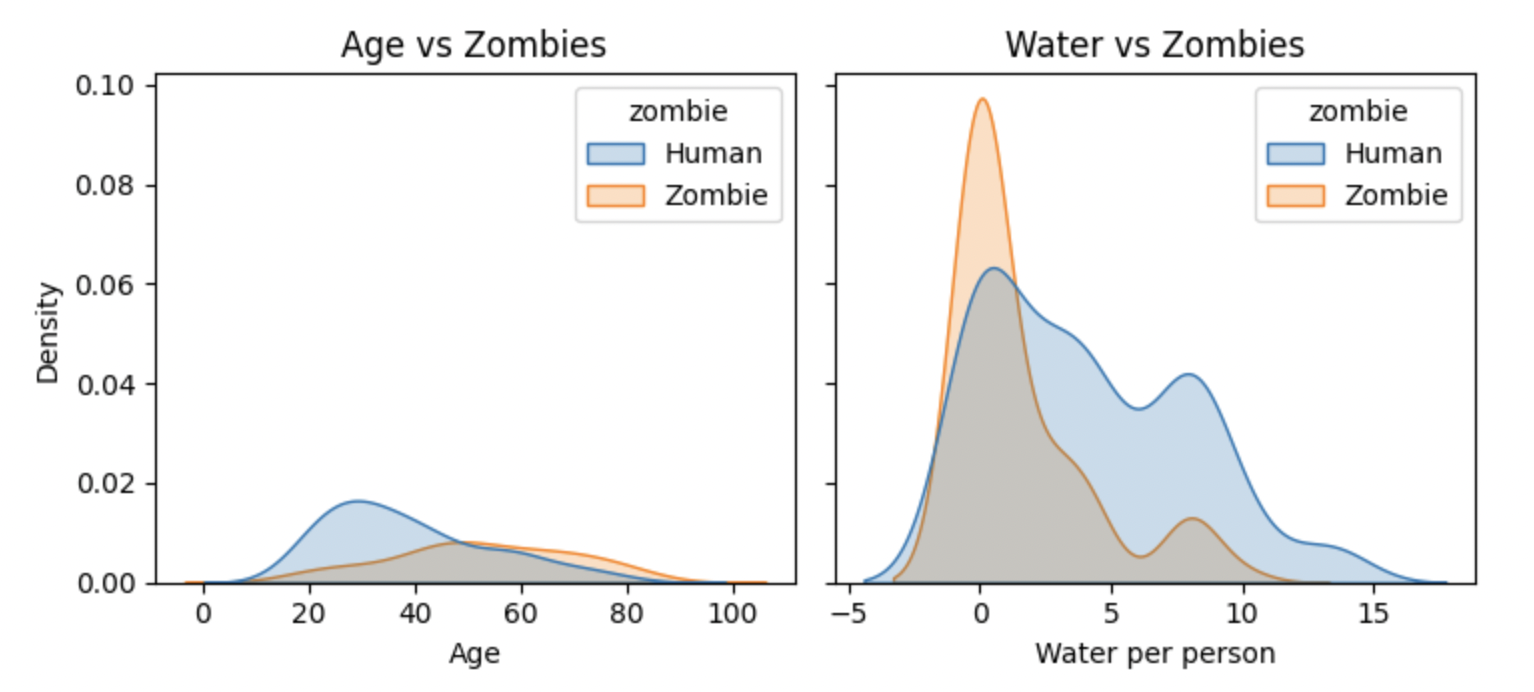
**Bivariate Analysis**

To find the relation between two features we use bivariate analysis. Comparing humans and zombies is important to identify differences in supplies. First we create a new variable water.person that indicates water per person and we examine it.



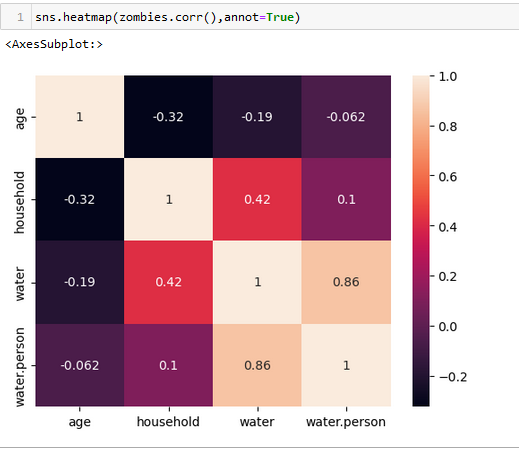
Then we create and compare age vs zombies and water vs zombies graphs using seaborn kdeplot function.





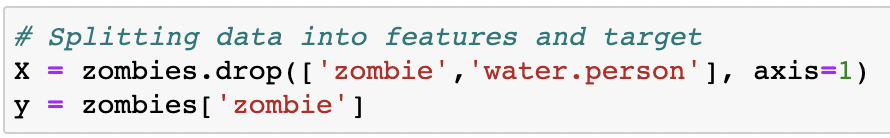
It looks like those who turned into zombies were older and had less available clean water. This suggests that getting water to the remaining humans might help protect them from the zombie hoards! Protecting older citizens is important, so we need to think about the best ways to reach this group.

**Multivariate Analysis**

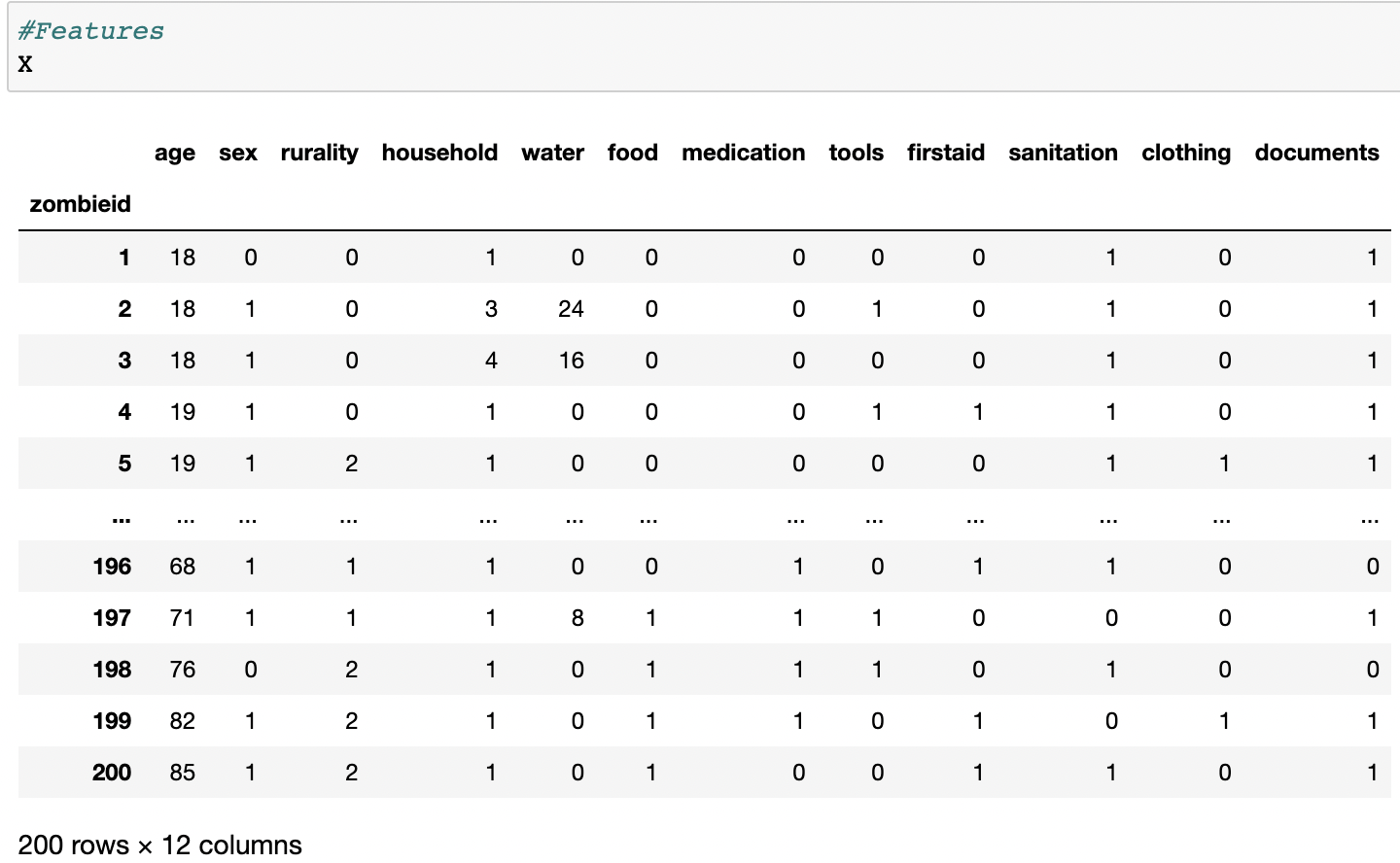


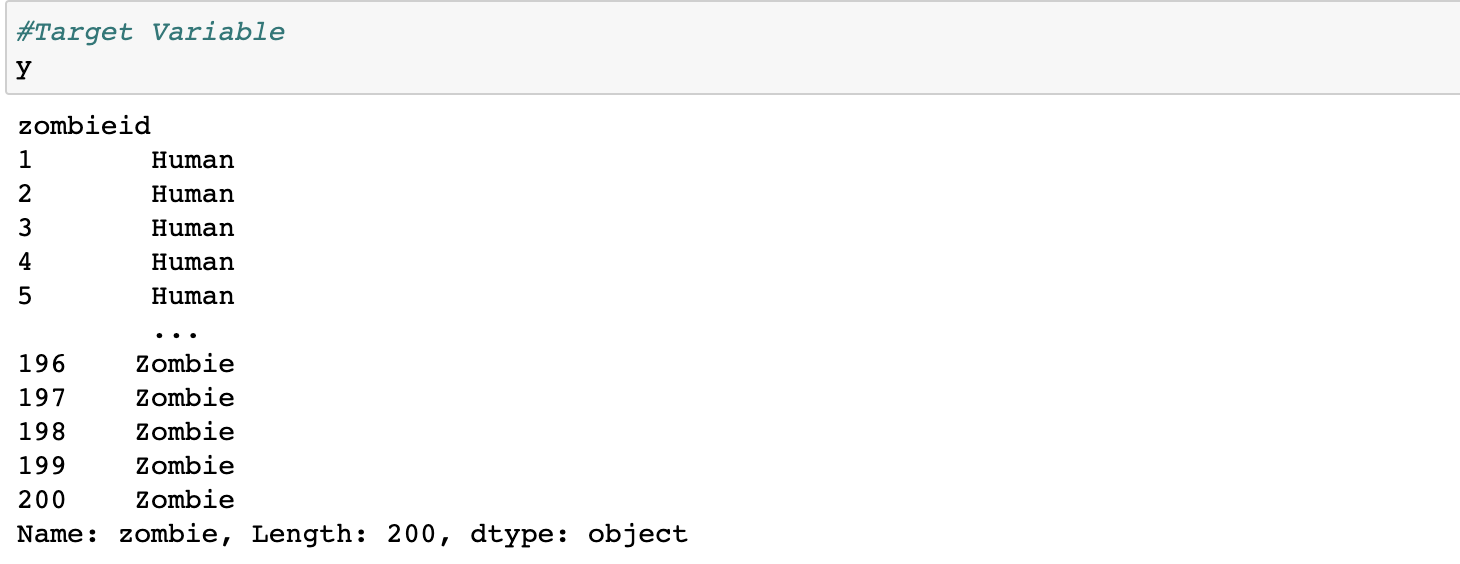
**Splitting data into train and test**

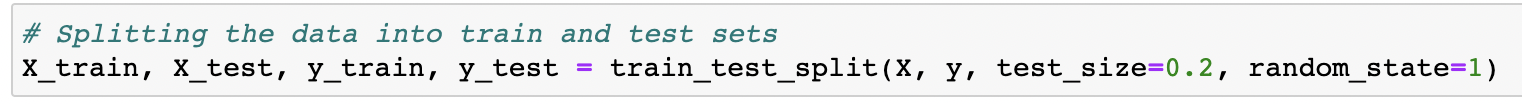
Now let’s split the Dataset into train and test sets. First split the dataset into X and y and then split the data set



Here x and y variables are created. On x variable, features are passed. And on y target variable is passed. For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.







**Model Building**

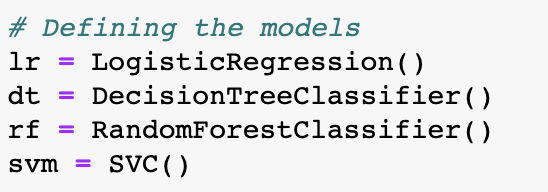
**Training The Model In Multiple Algorithms**

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance.

**Defining the models**  
First we define four different machine learning models. All four are classification algorithms. We use their respective classes from sklearn library to define them.   
This includes 

* LogisticRegression()
* DecisionTreeClassifier()
* RandomForestClassifier() and
* SVC().

We store them in variables lr, dt, rf and svm respectively.

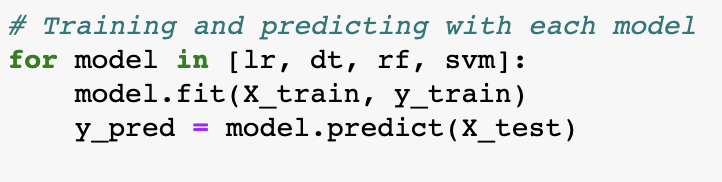


**Training and predicting with each model**

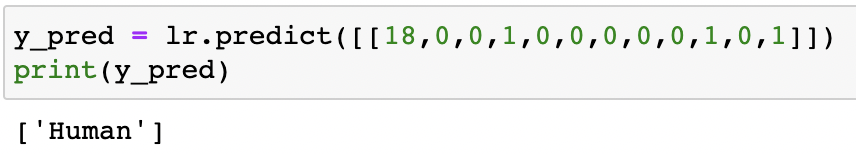
A for loop iterates through each of the four models.

Inside the loop, the current model is trained using the fit() method with the training dataset X\_train and y\_train.

The trained model then makes predictions on the test dataset X\_test using the predict() method and the predictions are stored in y\_pred.



**Testing The Model**

Here we have tested with the Logistic Regression algorithm. You can test with all algorithms. With the help of predict() function.  


**Performance Testing & Hyperparameter Tuning**

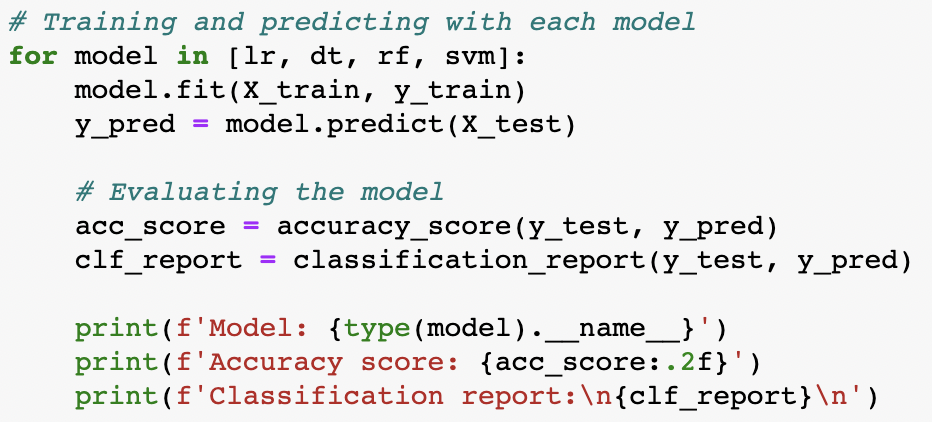
Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for classification tasks including accuracy, precision, recall, support and F1-score.

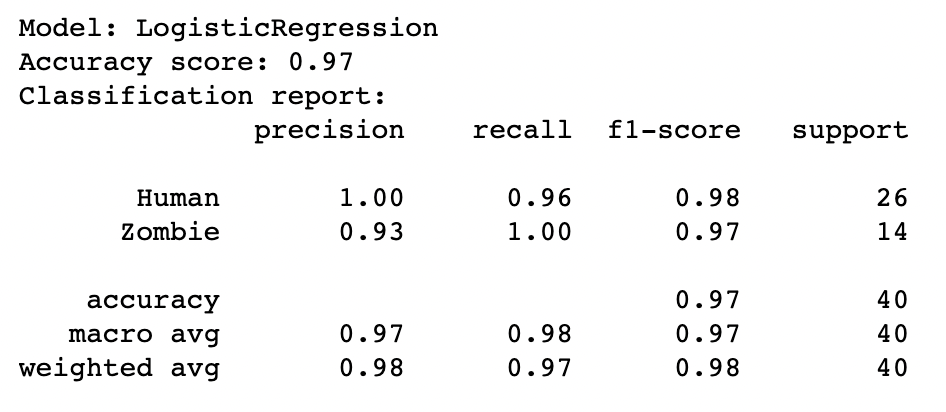
**Compare the models**

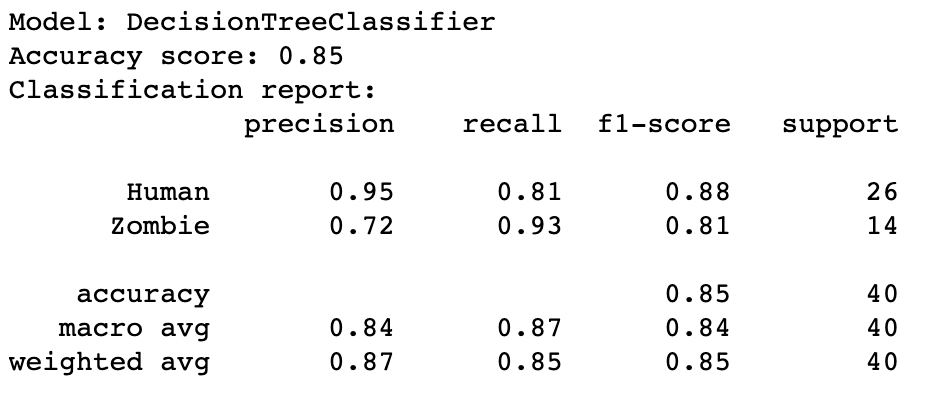
A for loop iterates through each of the four models. Inside the loop, The accuracy of the model is evaluated by comparing the predictions with the actual labels using the accuracy\_score() function, and the result is stored in acc\_score.

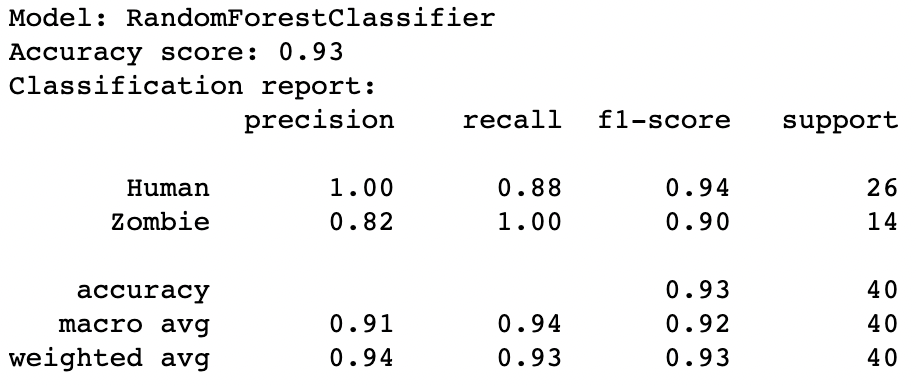
The classification\_report() function is used to generate a text report of the main classification metrics (precision, recall, F1-score, and support) for each class, and the result is stored in clf\_report.

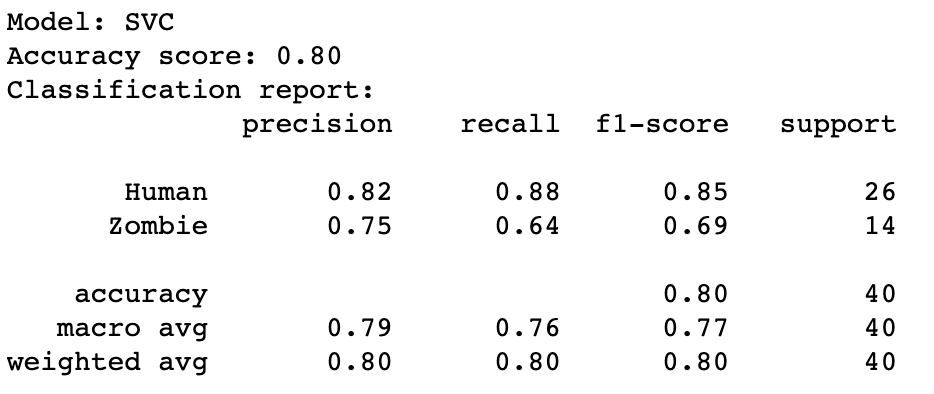
The results are printed to the console using the print() function. The model name, accuracy score, and classification report are displayed for each model, and the loop iterates until all four models have been trained and evaluated.











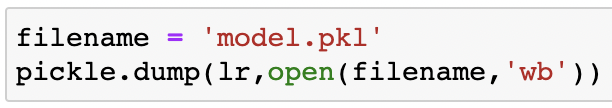
After calling the function, the results of models are displayed as output. From the three models logistic regression is performing well.

**Testing Model With Multiple Evaluation Metrics**

**Model Deployment**

**Save The Best Model**

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.



**Integrate With Web Framework**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

**Building Html Pages:**

* index.html

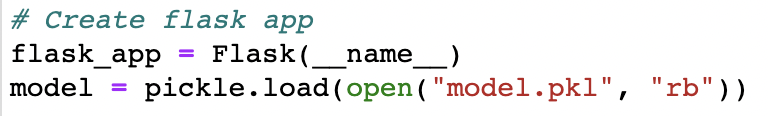
For this project create 1 HTML file namely and save them in the templates folder.

**Build Python code:**

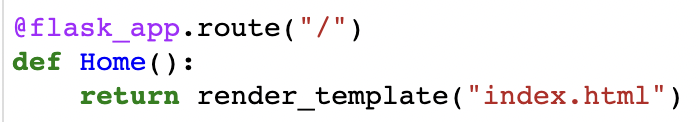
Import the libraries



Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module ( name ) as argument.



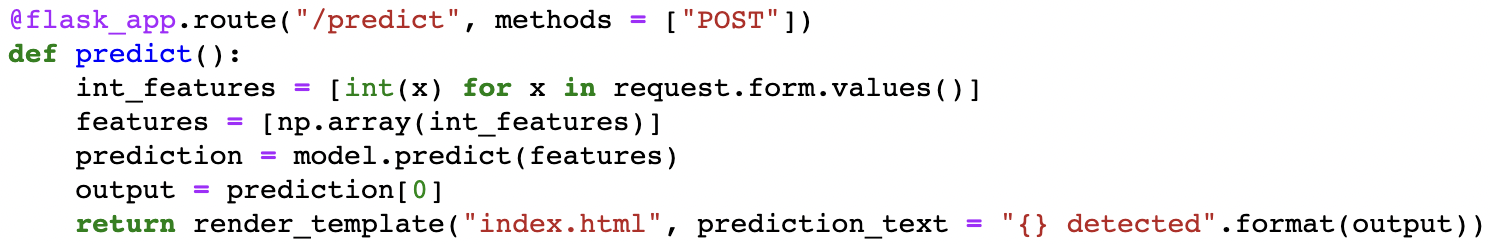
Render HTML Page:



Here we will be using a declared constructor to route to the HTML page which we have created earlier.

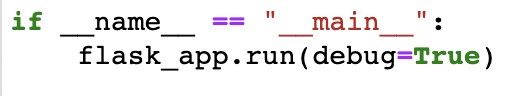
In the above example, ‘/’ URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:



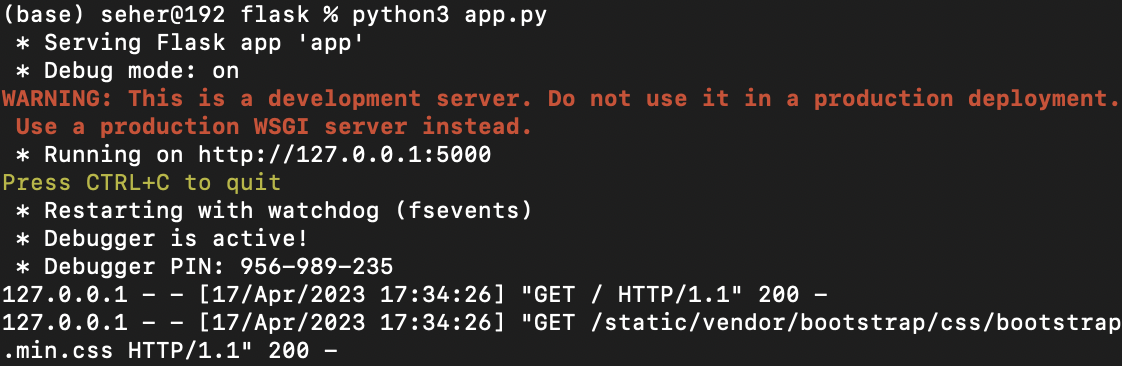
Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier.

**Main Function:**



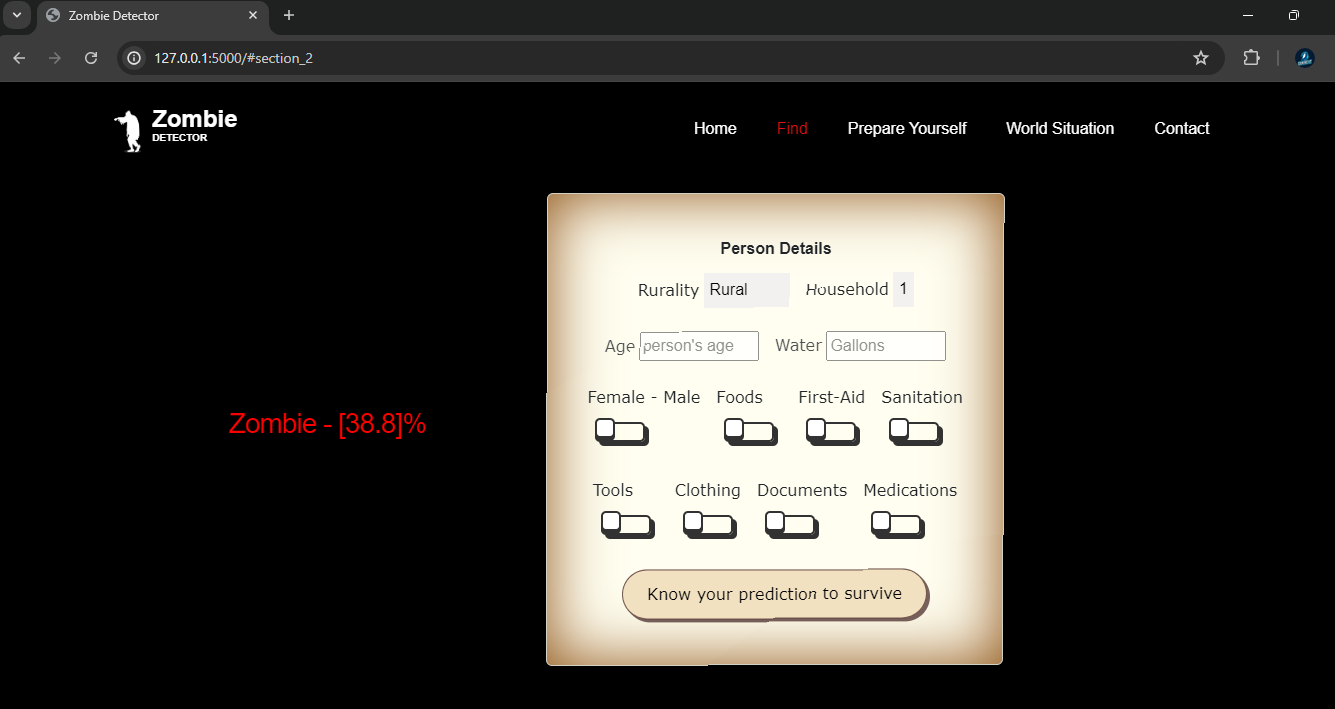
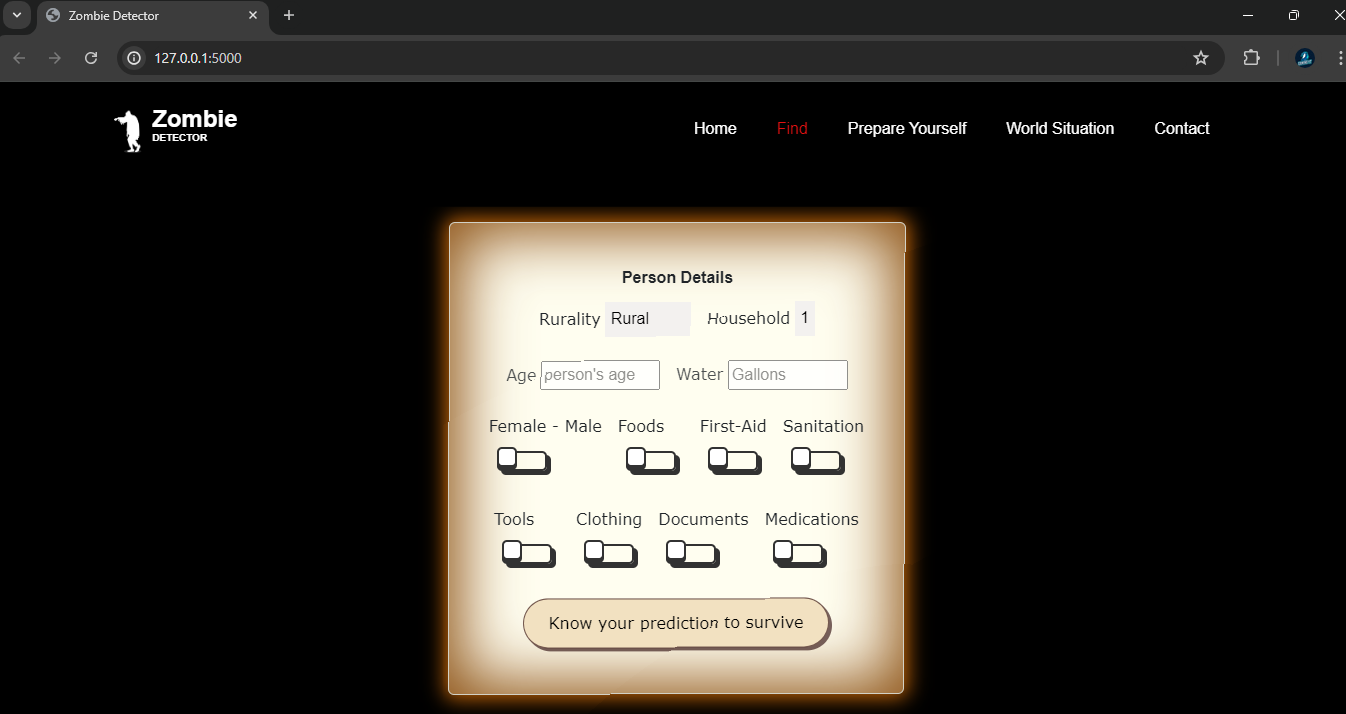
**Run the Web Application:**

* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.



Now,Go the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result





**Project Demonstration & Documentation**

Below mentioned deliverables to be submitted along with other deliverables

Activity 1:- Record explanation Video for project end to end solution

Activity 2:- Project Documentation-Step by step project development procedure

Create document as per the template provided