ZOMBIE DETECTION USING MACHINE LEARNING

1.Introduction

1.1Project Overview

The Zombie Detector project aims to use machine learning to distinguish between humans and zombies in images. The system leverages deep learning algorithms to analyze visual data and make accurate classifications.

1.2 Project Objective

The primary objective of this project is to build a robust machine learning model that can detect zombies based on visual features. This model will assist in scenarios like enhancing security systems in themed parks, adding realism to video games, and supporting creative media productions.

2. Project Initialization and Planning Phase

Activity 1: Define Problem Statement

The problem statement for this project is to develop a machine learning model that can accurately differentiate between humans and zombies in visual data. This involves analyzing images and identifying specific features that distinguish zombies from humans.

Ref. template:
Zombie detection Problem Statement Report:

Activity 2: Project Proposal (Proposed Solution)

Our proposed solution involves using deep learning techniques, particularly convolutional neural networks (CNNs), to train a model on labeled datasets containing images of humans and zombies. The model will be designed to analyze visual features and make accurate classifications.

Ref. template:
Zombie detection Project Proposal Report:

Activity 3: Initial Project Planning

Initial project planning includes defining the project scope, setting timelines, and allocating resources. Key objectives include data collection, model development, and performance evaluation. This phase also involves risk assessment and mitigation planning.

Ref. template:
Zombie detection Initial Project Planning Report:

3. Data Collection and Preprocessing Phase

Activity 1: Data Collection Plan, Raw Data Sources Identified

Initial project planning includes defining the project scope, setting timelines, and allocating resources. Key objectives include data collection, model development, and performance evaluation. This phase also involves risk assessment and mitigation planning.

Ref. template:
Zombie detection Raw Data Sources Report:

Activity 2: Data Quality Report

Data quality will be ensured by verifying the accuracy of labels, addressing missing values, and handling outliers. This step is crucial to build a reliable dataset for training and evaluating the machine learning model.

Ref. template:
Zombie detectioin Data Quality Report:

Activity 3: Data Exploration and Preprocessing

Data exploration involves analyzing the dataset to understand patterns, distributions, and outliers. Preprocessing includes handling missing values, scaling, and encoding categorical variables. These steps enhance data quality and ensure effective model training.

Ref. template:
Zombie detection Data Exploration and Preprocessing Report:

4. Model Development Phase

Activity 1: Feature Selection Report.

The feature selection report outlines the rationale behind choosing specific features for the zombie detection model. It evaluates the relevance, importance, and impact on predictive accuracy.

Activity 2: Model Selection Report

The model selection report details the rationale behind choosing CNNs for zombie detection. It considers the strengths of CNNs in handling image data, their adaptability, and overall predictive performance.

Activity 3: Initial Model Training Code, Model Validation and Evaluation Report

Initial model training involves implementing the chosen CNN architecture on the dataset. The model will be trained, validated, and evaluated using metrics such as accuracy, precision, recall, and F1-score.

5. Model Optimization and Tuning Phase

Activity 1: Hyperparameter Tuning Documentation

Hyperparameter tuning involves optimizing model parameters to enhance performance. This step includes experimenting with different learning rates, batch sizes, and network architectures.

Activity 2: Performance Metrics Comparison Report

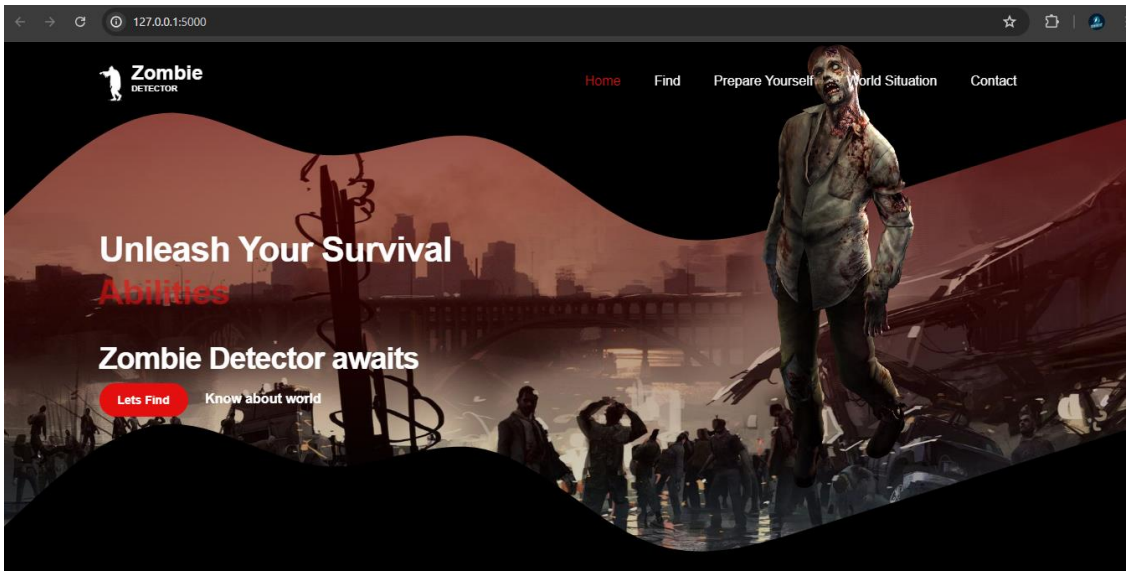
The performance metrics comparison report contrasts the baseline and optimized metrics for the model. This assessment provides a clear understanding of the refined predictive capabilities achieved through hyperparameter tuning.

Activity 3: Final Model Selection Justification

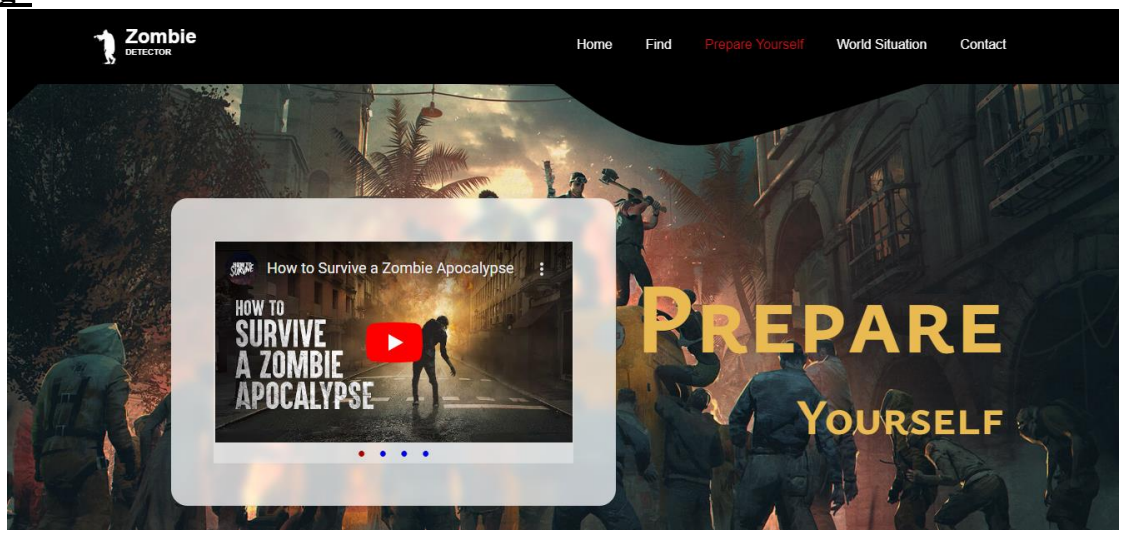
The final model selection justification articulates the rationale for choosing the optimized model. It highlights the model's accuracy, ability to handle complexity, and alignment with project objectives.

6.Results

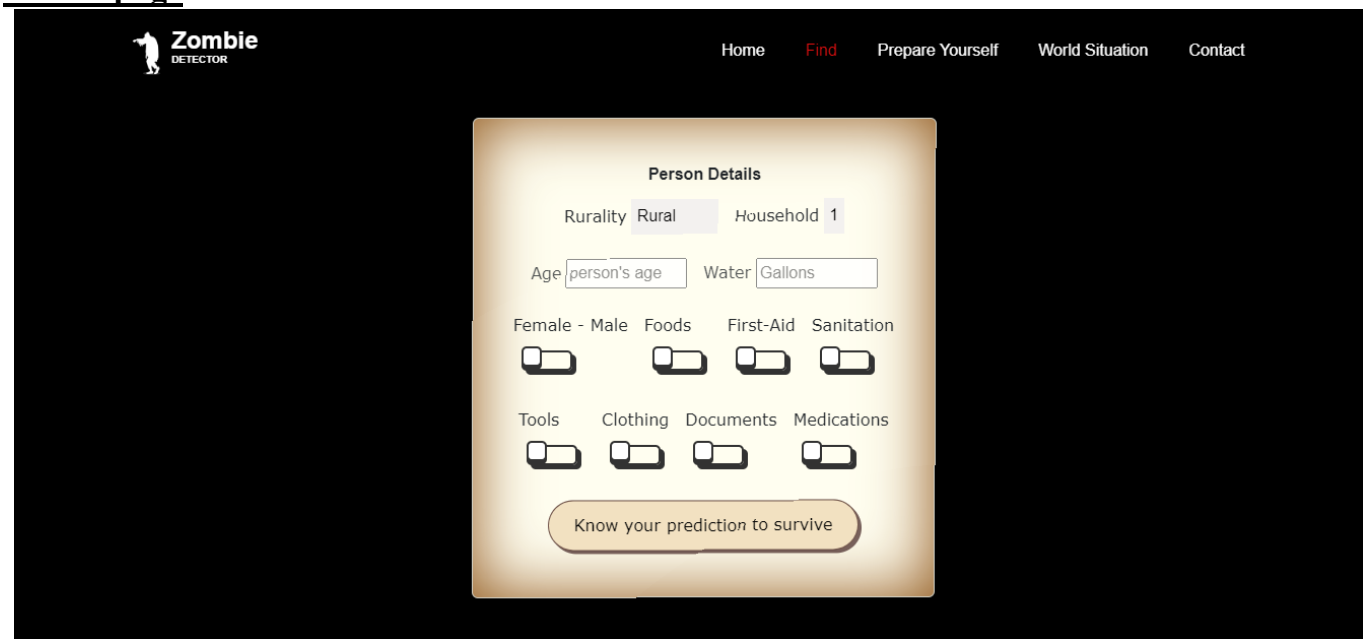
Index Page

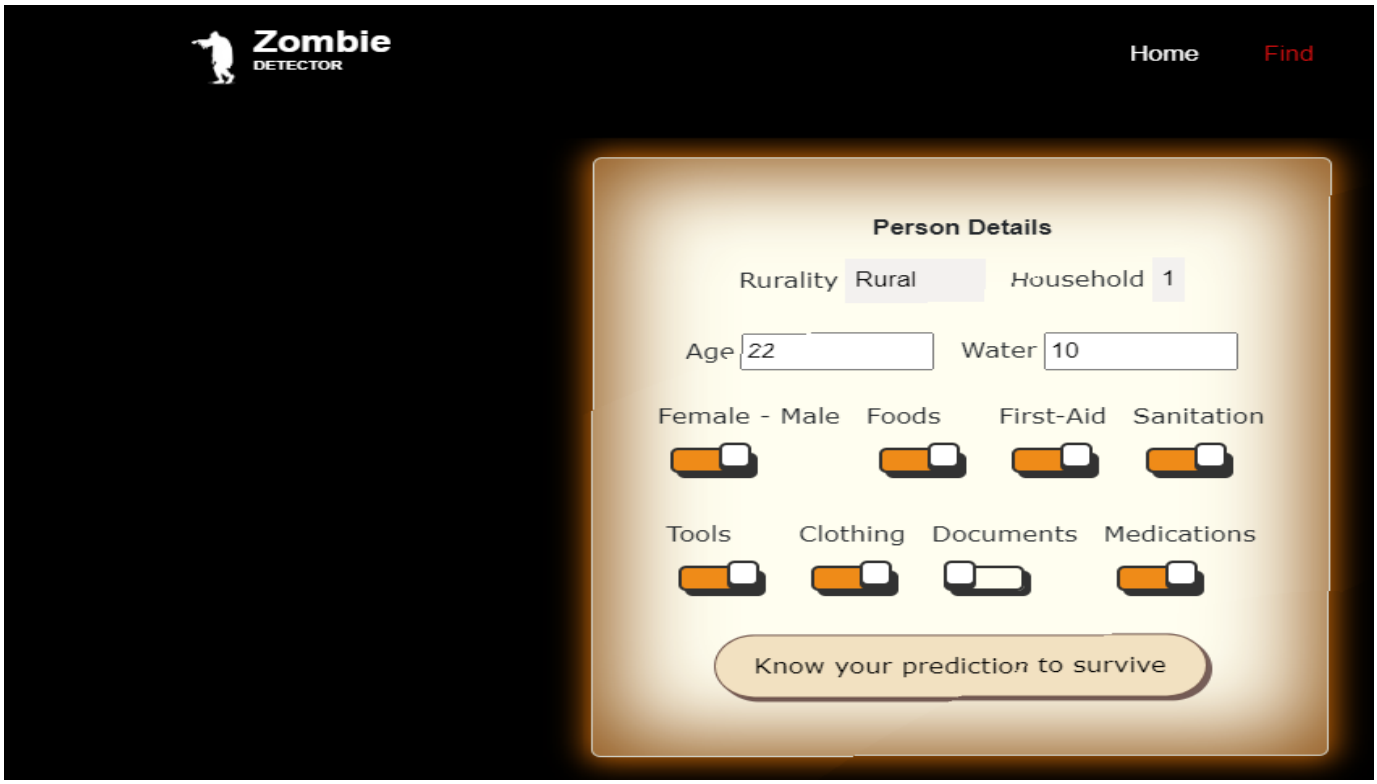


About page

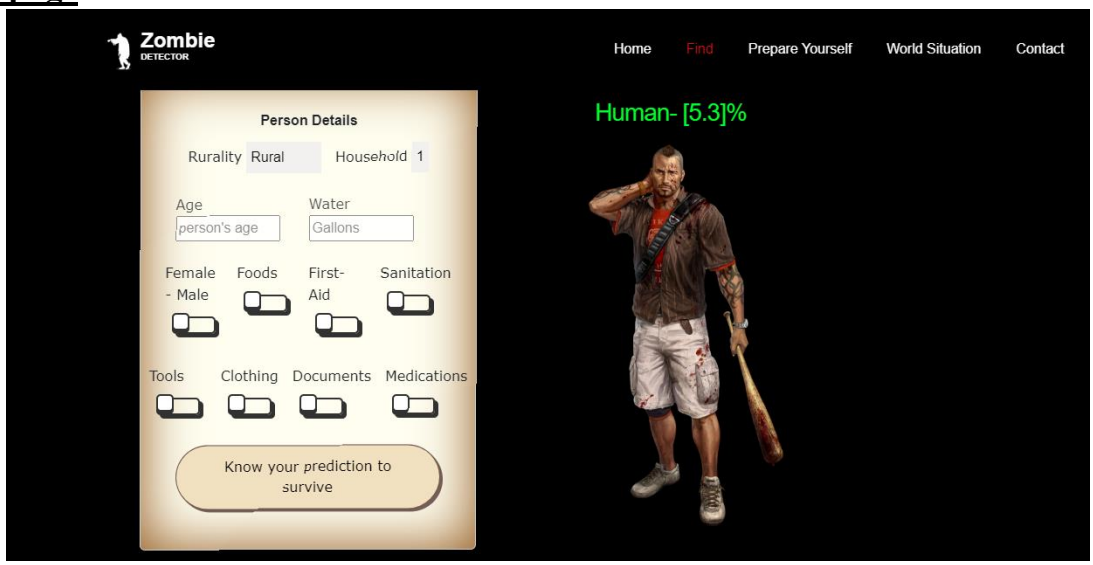


Details page





Predict page



7.Advantages & Disadvantages

ADVANTAGES:

1. **Enhanced Security:** The model can be used in security systems to detect zombies in real-time, enhancing safety in themed parks and events.
2. **Realism in Media:** The model can add realism to video games and movies by accurately distinguishing between humans and zombies.
3. **Support for Creative Productions:** The model can support creative media productions by providing reliable zombie detection.

DISADVANTAGES:

1. **Data Dependency:** The accuracy of the model depends heavily on the quality and diversity of the training data.
2. **Computational Resources:** Training deep learning models requires significant computational resources, which may not be accessible to all users.
3. **Potential for Bias:** The model may inherit biases present in the training data, leading to inaccurate classifications.

8.Conclusion

In this project, we aimed to develop a machine learning model for detecting zombies in images. Using CNNs, we trained and optimized a model that achieved high accuracy in distinguishing between humans and zombies. The results demonstrate the potential applications of this model in enhancing security systems, adding realism to media, and supporting creative productions.

Model Implementation

Three different machine learning models were implemented to predict honey prices:

1. **Random Forest**: This ensemble learning method was used for its robustness and ability to handle complex data relationships.
2. **Decision Tree**: This model was chosen for its simplicity and interpretability, making it easy to understand how purity levels impact honey prices.
3. **Linear Regression**: As a fundamental regression technique, this model provided a baseline for understanding the linear relationship between purity and price.

- Our analysis revealed that purity is a significant factor influencing honey prices. Among the three models, the Random Forest model demonstrated the highest accuracy, effectively capturing the non-linear relationships in the data. The

Decision Tree model also performed well, providing clear insights into how different purity levels affect pricing. The Linear Regression model, while less accurate, confirmed a positive correlation between purity and price.

9.Future Scope

Expansion of Data Sources

Expand data collection to include more diverse and representative datasets, including images from various sources and environments.

Integration of Advanced Techniques

Further integrate advanced machine learning techniques, such as transfer learning and ensemble methods, to improve model accuracy and robustness.

Real-time Implementation

Develop real-time implementation of the model for live zombie detection in security systems and interactive media.

Collaboration with Industry

Collaborate with industry partners in entertainment, security, and creative media to refine and deploy the model in real-world applications.

10.Appendix

10.1 Source Code

Index Page:

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <meta name="description" content="">
        <meta name="author" content="">

        <title>Zombie Detector</title>

        <!-- CSS FILES -->
```

```
        <link href="/static/aos/aos.css" rel="stylesheet">

        <link rel="preconnect" href="https://fonts.googleapis.com">

        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

        <link
href="https://fonts.googleapis.com/css2?family=DM+Sans:ital,wght@0,400;0,500;0,700;1,400&disp
lay=swap" rel="stylesheet">

        <link href="../static/css/bootstrap.min.css" rel="stylesheet">

        <link href="../static/css/bootstrap-icons.css" rel="stylesheet">

        <link href="../static/css/style.css" rel="stylesheet" >
        <link rel="preconnect" href="https://fonts.googleapis.com">
        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
        <link
href="https://fonts.googleapis.com/css2?family=Merriweather:ital,wght@1,900&family=Sora&famil
y=Unbounded:wght@300;400&family=Ysabeau+SC:wght@700&display=swap" rel="stylesheet">
        <!-- <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"> -->
    </head>

    <body>

        <main>
            <nav class="navbar navbar-expand-lg">
                <div class="container">
                    <a class="navbar-brand d-flex align-items-center" href="">
                        <img src="../static/images/logo.png" class="navbar-brand-image img-
fluid" alt="Zombie">
                        <span class="navbar-brand-text">
                            Zombie
                            <small>Detector</small>
                        </span>
                    </a>
                    <div class="d-lg-none ms-auto me-3">
                        <a class="btn custom-btn custom-border-btn" data-bs-
toggle="offcanvas" href="#offcanvasExample" role="button" aria-
controls="offcanvasExample">UNESCO Login</a>
                    </div>

                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
```

```
                    <div class="collapse navbar-collapse flex" id="navbarNav">
                        <ul class="navbar-nav ms-lg-auto">
                            <li class="nav-item">
                                <a class="nav-link click-scroll" href="#section_1">Home</a>
                            </li>

                            <li class="nav-item">
                                <a class="nav-link click-scroll" href="#section_2">Find</a>
                            </li>

                            <li class="nav-item">
                                <a class="nav-link click-scroll" href="#section_3">Prepare
Yourself</a>
                            </li>

                            <li class="nav-item">
                                <a class="nav-link click-scroll" href="#section_4">World
Situation</a>
                            </li>

                            <li class="nav-item">
                                <a class="nav-link click-scroll" href="#section_5">Contact
</a>
                            </li>
                        </ul>
                    </div>
                </div>
            </nav>




            <section class="hero-section d-flex justify-content-center align-items-center"
id="section_1">

                <div class="section-overlay"></div>

                <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path
fill="#000000 " fill-opacity="1"
d="M0,224L34.3,192C68.6,160,137,96,206,90.7C274.3,85,343,139,411,144C480,149,549,107,617,122.
7C685.7,139,754,213,823,240C891.4,267,960,245,1029,224C1097.1,203,1166,181,1234,160C1302.9,13
9,1371,117,1406,106.7L1440,96L1440,0L1405.7,0C1371.4,0,1303,0,1234,0C1165.7,0,1097,0,1029,0C9
60,0,891,0,823,0C754.3,0,686,0,617,0C548.6,0,480,0,411,0C342.9,0,274,0,206,0C137.1,0,69,0,34,
0L0,0Z"></path></svg>

                <div class="container">
                    <div class="row">

                        <div class="col-lg-6 col-12 mb-5 mb-lg-0" >
```

```
                            <h1 class="cd-headline rotate-1 text-white mb-4 pb-2">
                                <span>Unleash Your Survival </span>
                                <span class="cd-words-wrapper">
                                    <b class="is-visible">Instincts</b>
                                    <b>Abilities</b>
                                    <b>Resilience</b>
                                </span>
                            </h1>
                            <h2 class="text-white">Zombie Detector awaits</h2>

                            <div class="custom-btn-group">
                                <a href="#section_2" class="btn custom-btn smoothscroll me-
3">Lets Find</a>

                                <a href="#section_3" class="link smoothscroll"><b>Know about
world</b></a>
                            </div>
                        </div>

                        <div class="images" >

                            <img src="/static/images/a1.png" class="zombie img-fluid ">

                        </div>

                    </div>
                </div>

                <svg class="bottom"xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440
320"><path fill="#000000" fill-opacity="1"
d="M0,224L34.3,192C68.6,160,137,96,206,90.7C274.3,85,343,139,411,144C480,149,549,107,617,122.
7C685.7,139,754,213,823,240C891.4,267,960,245,1029,224C1097.1,203,1166,181,1234,160C1302.9,13
9,1371,117,1406,106.7L1440,96L1440,320L1405.7,320C1371.4,320,1303,320,1234,320C1165.7,320,109
7,320,1029,320C960,320,891,320,823,320C754.3,320,686,320,617,320C548.6,320,480,320,411,320C34
2.9,320,274,320,206,320C137.1,320,69,320,34,320L0,320Z"></path></svg>
            </section>


            <section class="about-section section-padding" id="section_2">

                <div class="container">

                    <div class="row1">
                        <!-- style="display: none;" -->

                        <div class="col-xs-6 output1" style="margin: auto;">
                            <h3 id ="outz"  class="col-xs-6 text-center m-3" >
                                {{zombie}}
                            </h3>
```

```
                            <div class="col-xs-6 z1"  >
                                <img src="/static/images/z2.png" class=" img-fluid "
style="max-width: 65%;">
                            </div>
                        </div>
                        <div class="col-xs-6 card">
                            <!-- <p>Write here</p> -->
                            <form id="myForm" method="post" action="#section_2">
                                <header class=" text-center" style="padding-top: 5%;">
                                  <b>Person Details</b>
                                </header>
                                <span class="message"></span>
                                <fieldset style="display: flex;justify-content: center;">
                                <label class="em">
                                <span>Rurality</span>

                                <select class="input" type="select" required="" id="rurality"
name="rurality">
                                    <option value=0>Rural</option>
                                    <option value=1>Sub-urban</option>
                                    <option value=2>Urban</option>
                                </select>

                            </label class="em">
                            <label class="em">
                                    <span>Household</span>
                                    <select class="input" type="number" required=""
id="household" name="household">
                                      <option>1</option>
                                      <option>2</option>
                                      <option>3</option>
                                      <option>4</option>
                                      <option>5</option>
                                      <option>6</option>
                                      <option>7</option>
                                      <option>8</option>
                                      </select>
                                    </label>

                                </fieldset>
                                <fieldset style="display: flex;justify-content: center;">
                                <label class="em">
                                  <span class="text-center">Age</span>
                                  <input placeholder="person's age" class="input"
type="number" id="age" name="age" required="">
                                </label>
                                <label class="em">
                                    <span>Water</span>
                                    <input placeholder="Gallons" class="input" type="number"
id="water" name = "water" required="">
```

```
                                </label>

                            </fieldset>
                                <fieldset style="display: flex;justify-content: center;">
                                <label class="em">
                                    <span>Female - Male</span>
                                    <label class="switch">
                                        <input class="toggle" type="checkbox"  id="sex"
name="sex">

                                        <span class="slider"></span>
                                        <span class="card-side"></span>
                                    </label>
                                </label>
                                <label class="em">
                                    <span>Foods</span>
                                    <label class="switch">
                                        <input class="toggle" type="checkbox" id ="food" name
="food">

                                        <span class="slider"></span>
                                        <span class="card-side"></span>
                                    </label>
                                </label>
                                <label class="em">
                                    <span >First-Aid</span>
                                    <label class="switch">
                                        <input class="toggle" type="checkbox" id="aid"
name="aid">

                                        <span class="slider"></span>
                                        <span class="card-side"></span>
                                    </label>
                                </label>
                                <label class="em">
                                    <span>Sanitation</span>
                                    <label class="switch">
                                        <input class="toggle" type="checkbox" id="sanitation"
name="sanitation">

                                        <span class="slider"></span>
                                        <span class="card-side"></span>
                                    </label>
                                </label>

                                </fieldset>
                                <fieldset style="display: flex;justify-content: center;">
                                    <label class="em">
                                        <span>Tools</span>
                                        <label class="switch">
                                            <input class="toggle" type="checkbox" id="tools"
name="tools">

                                            <span class="slider"></span>
                                            <span class="card-side"></span>
```

```
                                        </label>
                                    </label>

                                    <label class="em">
                                        <span>Clothing      </span>
                                        <label class="switch">
                                            <input class="toggle" type="checkbox"
id="clothing" name="clothing">
                                            <span class="slider"></span>
                                            <span class="card-side"></span>
                                        </label>
                                    </label>
                                    <label class="em">
                                        <span>Documents</span>
                                        <label class="switch">
                                            <input class="toggle"
type="checkbox"id="documents" name="documents">
                                            <span class="slider"></span>
                                            <span class="card-side"></span>
                                        </label>
                                    </label>
                                    <label class="em">
                                        <span>Medications</span>
                                        <label class="switch">
                                            <input class="toggle" type="checkbox"
id="medications" name="medications">
                                            <span class="slider"></span>
                                            <span class="card-side"></span>
                                        </label>
                                    </label>
                                    </fieldset>
                                    <fieldset>
                                    <label class="em">
                                        <button class="button">
                                            Know your prediction to survive
                                          </button>
                                          </label>

                                 </fieldset>
                                        <!-- <button id ="showButton" class="flipButton"
type="submit" onclick="toggleCard()">Submit</button> -->
                                </form>
                            </div>
                            <svg class="filter">
                              <filter id="wavy2">
                                <feTurbulence x="0" y="0" baseFrequency="0.0001" numOctaves="5"
seed="1"></feTurbulence>
                                <feDisplacementMap in="SourceGraphic"
scale="13"></feDisplacementMap>
                              </filter>
```

```
                        </svg>
                        <div class="output" style="margin: auto;">

                          <div  style="max-width: 65%; ">
                              <h3 id ="outz" class="text-center m-3" style="color: rgb(0,
255, 42);">
                                  {{human}}
                              </h3>
                              <img src="/static/images/human4.png" class="zombie img-fluid
z2" style="  margin: auto;">
                          </div>
                      </div>

                  </div>
              </div>
            </section>


            <section class="section-bg-image" id="section_3">
                <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path
fill="#000000" fill-opacity="1"
d="M0,224L34.3,192C68.6,160,137,96,206,90.7C274.3,85,343,139,411,144C480,149,549,107,617,122.
7C685.7,139,754,213,823,240C891.4,267,960,245,1029,224C1097.1,203,1166,181,1234,160C1302.9,13
9,1371,117,1406,106.7L1440,96L1440,0L1405.7,0C1371.4,0,1303,0,1234,0C1165.7,0,1097,0,1029,0C9
60,0,891,0,823,0C754.3,0,686,0,617,0C548.6,0,480,0,411,0C342.9,0,274,0,206,0C137.1,0,69,0,34,
0L0,0Z"></path></svg>

                <div class="container">
                    <div class="row" style="flex-direction: row-reverse; padding-bottom:
5%;padding-right: 2%;">
                        <div class="col-lg-5 save-text">
                            <span class="save navbar-brand-text">
                                Prepare
                              <medium class="text-right">Yourself</medium>
                            </span>
                        </div>
                        <div class="col-lg-6">
                            <div class="section-bg-image-block">
                                <div class="sliderv">
                                    <div><iframe
src="//www.youtube.com/embed/T_7oXlqIyzA?rel=0" allowfullscreen frameborder="0"
name="slider"></iframe></div>
                                    <span>
                                      <a
href="//www.youtube.com/embed/T_7oXlqIyzA?rel=0&autoplay=1" target="slider">●
                                  </a><a
href="//www.youtube.com/embed/whV0U2RuUrs?rel=0&autoplay=1" target="slider">●
                                  </a><a
href="//www.youtube.com/embed/4r08wJs9kHY?rel=0&autoplay=1" target="slider">●
```

```
                                    </a><a
href="//www.youtube.com/embed/wNt8JS99PrY?rel=0&autoplay=1" target="slider">●</a>
                                      </span>
                                  </div>
                              </div>
                          </div>

                      </div>

                  </div>
                  <svg viewBox="0 0 1265 144" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"><path fill="#000000" d="M 0 40 C 164 40 164 20 328
20 L 328 20 L 328 0 L 0 0 Z" stroke-width="0"></path> <path fill="#000000" d="M 327 20 C
445.5 20 445.5 89 564 89 L 564 89 L 564 0 L 327 0 Z" stroke-width="0"></path> <path
fill="#000000" d="M 563 89 C 724.5 89 724.5 48 886 48 L 886 48 L 886 0 L 563 0 Z" stroke-
width="0"></path><path fill="#000000" d="M 885 48 C 1006.5 48 1006.5 67 1128 67 L 1128 67 L
1128 0 L 885 0 Z" stroke-width="0"></path><path fill="#000000" d="M 1127 67 C 1196 67 1196 0
1265 0 L 1265 0 L 1265 0 L 1127 0 Z" stroke-width="0"></path></svg>

              </section>




              <section class="contact-section section-padding" id="section_4">
                  <div class="container">
                      <div class="col-lg-5 col-12 " style="margin: auto;">
                          <h2 class="mb-4 pb-2 text-center">World Situation</h2>
                          <div class="world-parent">
                              <img src="../static/images/hashmap.jpg"
class="world">
                          </div>

                      </div>
                  </div>
              </section>
          </main>
          <section class="foot justify-content-center align-items-center" id="section_5">
              <footer class="site-footer">
              <div class="container">
                  <div class="row">

                      <div class="col-lg-6 col-12 me-auto mb-5 mb-lg-0">
                          <a class="navbar-brand d-flex align-items-center" href="index.html">
                              <img src="../static/images/logo.png" class="navbar-brand-image
img-fluid" alt="">
                              <span class="navbar-brand-text" style="color:red">
                                  Zombie
                                  <small>Detector</small>
                              </span>
```

```
                        </a>
                    </div>



                    <div class="col-lg-2 col-12 ms-auto">
                        <ul class="social-icon mt-lg-5 mt-3 mb-4">
                            <li class="social-icon-item">
                                <a href="https://github.com/" class="social-icon-link bi-
github"></a>
                            </li>
                            <li class="social-icon-item">
                                <a href="https://www.linkedin.com/feed/" class="social-icon-
link bi-linkedin"></a>
                            </li>
                            <li class="social-icon-item">
                                <a href="https://www.instagram.com/akhilbollaboina/"
class="social-icon-link bi-skype"></a>
                            </li>
                        </ul>
                        <p class="copyright-text"><a rel="nofollow"
href="https://github.com/" target="_blank">akhilbollaboina</a></p>
                    </div>

                </div>
            </div>


            <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320" style="background-
image: url(/static/images/fotter.jpg)"><path fill="#000000 " fill-opacity="1"
d="M0,224L34.3,192C68.6,160,137,96,206,90.7C274.3,85,343,139,411,144C480,149,549,107,617,122.
7C685.7,139,754,213,823,240C891.4,267,960,245,1029,224C1097.1,203,1166,181,1234,160C1302.9,13
9,1371,117,1406,106.7L1440,96L1440,0L1405.7,0C1371.4,0,1303,0,1234,0C1165.7,0,1097,0,1029,0C9
60,0,891,0,823,0C754.3,0,686,0,617,0C548.6,0,480,0,411,0C342.9,0,274,0,206,0C137.1,0,69,0,34,
0L0,0Z"></path></svg>
            <!-- <div class="fotter-overlay"> -->

            </div>
            <!-- <div class="container" ></div> -->


            </footer>
        </section>

        <!-- JAVASCRIPT FILES -->
        <script src="../static/aos/aos.js"></script>
        <script src="../static/js/bootstrap.bundle.min.js"></script>
        <script src="../static/js/jquery.min.js"></script>
        <script src="../static/js/jquery.sticky.js"></script>
        <script src="../static/js/click-scroll.js"></script>
```

```
        <script src="../static/js/animated-headline.js"></script>
        <script src="../static/js/modernizr.js"></script>
        <script src="../static/js/custom.js"></script>


    </body>
</html>
```

APP.PY

```
from flask import Flask, render_template, request
import pickle

app = Flask(__name__)

# Load the ML model from pickle file
with open('model.pkl', 'rb') as file:
    model = pickle.load(file)

@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 - "+str((predictions*100.0)*1.0)+"%"
        else:
            h = "Human- "+str((predictions*100.0)*1.0)+"%"
```

```
        # 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)
```

10.2 Code snippetes

Data collection

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.compose import make_column_transformer
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
import pickle
import warnings
warnings.filterwarnings('ignore')
```

```
zombies = pd.read_csv("/content/zombies.csv", index_col="zombieid")
```

Run cell (Ctrl+Enter)
cell has not been executed in this session

executed by VR Gaming yt
Wednesday 10 July 2024
executed in 0.037 s

			sex	rurality	household	water	food	medication	tools	firstaid	sanitation	clothing	documents
1	Human	18	Female	Rural	1	0	Food	Medication	No tools	First aid supplies	Sanitation	Clothing	NaN
2	Human	18	Male	Rural	3	24	Food	Medication	tools	First aid supplies	Sanitation	Clothing	NaN
3	Human	18	Male	Rural	4	16	Food	Medication	No tools	First aid supplies	Sanitation	Clothing	NaN
4	Human	19	Male	Rural	1	0	Food	Medication	tools	No first aid supplies	Sanitation	Clothing	NaN
5	Human	19	Male	Urban	1	0	Food	Medication	No tools	First aid supplies	Sanitation	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
196	Zombie	68	Male	Suburban	1	0	Food	No medication	No tools	No first aid supplies	Sanitation	Clothing	Documents
197	Zombie	71	Male	Suburban	1	8	No food	No medication	tools	First aid supplies	No sanitation	Clothing	NaN
198	Zombie	76	Female	Urban	1	0	No food	No medication	tools	First aid supplies	Sanitation	Clothing	Documents
199	Zombie	82	Male	Urban	1	0	No food	No medication	No tools	No first aid supplies	No sanitation	NaN	NaN
200	Zombie	85	Male	Urban	1	0	No food	Medication	No tools	No first aid supplies	Sanitation	Clothing	NaN

```
(200, 13)
```

```
zombies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 200 entries, 1 to 200
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   zombie      200 non-null    object
 1   age         200 non-null    int64
 2   sex         200 non-null    object
 3   rurality    200 non-null    object
 4   household   200 non-null    int64
 5   water       200 non-null    int64
 6   food        200 non-null    object
 7   medication  200 non-null    object
 8   tools       200 non-null    object
 9   firstaid    200 non-null    object
 10  sanitation  200 non-null    object
 11  clothing    126 non-null    object
 12  documents   66 non-null     object
dtypes: int64(3), object(10)
memory usage: 21.9+ KB
```

Handling categorical values

```
dtypes: int64(3), object(10)
memory usage: 21.9+ KB
```

```
np.sum(zombies.isnull())
```

```
zombie          0
age             0
sex             0
rurality        0
household       0
water           0
food            0
medication      0
tools           0
firstaid        0
sanitation      0
clothing       74
documents     134
dtype: int64
```

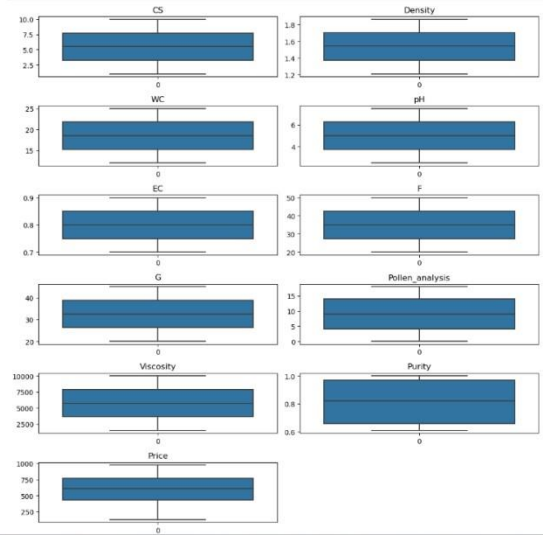
```
dtype: object
```

```
[ ] #Add new level and recode NA to "No clothing"
    new_categories = pd.unique(zombies["clothing"]).tolist()
    # Remove the NaN value from the list
    new_categories = [x for x in new_categories if pd.notna(x)]
    new_categories.append('No clothing')

    zombies["clothing"] = pd.Categorical(zombies["clothing"], categories=new_categories)
    zombies['clothing'].fillna('No clothing', inplace=True)
```

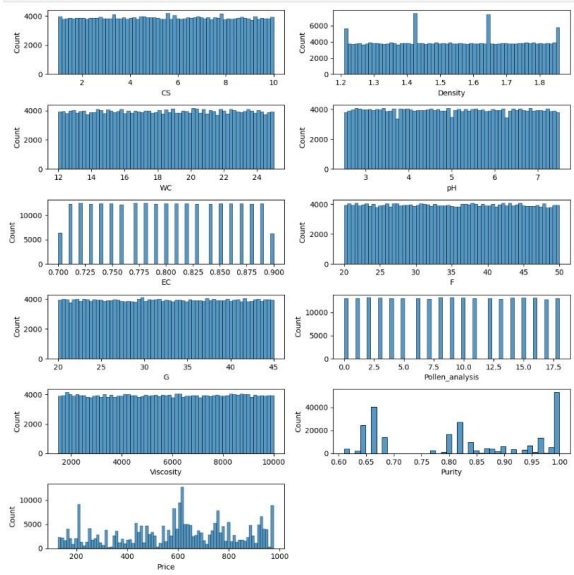
```
# Initializing LabelEncoder object
le = LabelEncoder()

# Converting categorical columns to numerical using Label Encoding
zombies['sex'] = le.fit_transform(zombies['sex'])
zombies['rurality'] = le.fit_transform(zombies['rurality'])
zombies['food'] = le.fit_transform(zombies['food'])
zombies['medication'] = le.fit_transform(zombies['medication'])
zombies['tools'] = le.fit_transform(zombies['tools'])
zombies['clothing'] = le.fit_transform(zombies['clothing'])
zombies['documents'] = le.fit_transform(zombies['documents'])
zombies['water'] = le.fit_transform(zombies['firstaid'])
zombies['sanitation'] = le.fit_transform(zombies['sanitation'])
zombies['firstaid'] = le.fit_transform(zombies['firstaid'])
```

visual analysis / finding outliers

```
[10]: plt.figure(figsize=(11,11))
for i,col in enumerate(data.columns):
    plt.subplot(6,2,i+1)
    sns.boxplot(data[col])
    plt.title(col)
plt.tight_layout()
```



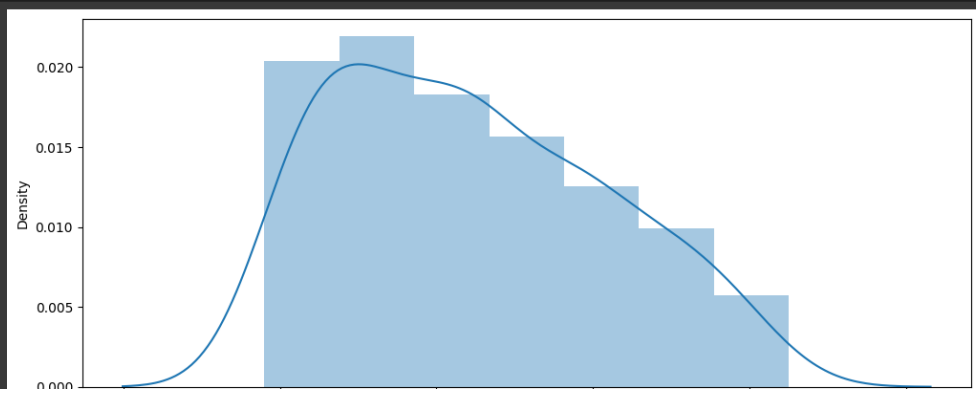

```
[11]: plt.figure(figsize=(11,11))
for i,col in enumerate(data.columns):
    plt.subplot(6,2,i+1)
    sns.histplot(data[col])
plt.tight_layout()
```



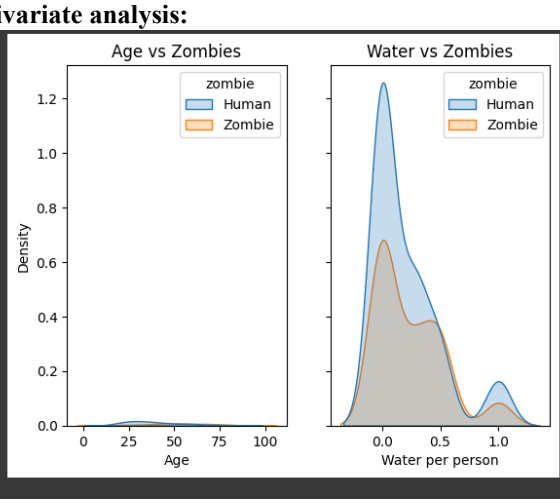
Univariayte analysis:

Bivariate analysis:

```
#Plotting Age
fig = plt.figure(figsize=(12,5))
sns.distplot(zombies['age'])
plt.show()
```



Multivariate analysis:



Model Building:

```
[ ] # Create water-per-person
    zombies['water.person'] = zombies['water'] / zombies['household']
    # Examine the new variable
    print(zombies['water.person'].describe())
```

```
count    200.000000
mean       0.214917
std        0.286678
min        0.000000
25%        0.000000
50%        0.000000
75%        0.333333
max        1.000000
Name: water.person, dtype: float64
```

```
[ ] # Create the ageZombies graph
    f, (ageZombies, waterPersonZom) = plt.subplots(1, 2, sharey=True)
    ageZombies = sns.kdeplot(data=zombies, x="age", fill=True, hue="zombie", ax=ageZombies)
    ageZombies.set(title="Age vs Zombies", xlabel="Age", ylabel="Density")

    # Create the waterPersonZom graph
    waterPersonZom = sns.kdeplot(data=zombies, x="water.person", fill=True, hue="zombie", ax=waterPersonZom)
    waterPersonZom.set(title="Water vs Zombies", xlabel="Water per person", ylabel="Density")

    plt.show()
```

Linear Regression model

```
[23]: lr=LinearRegression()
      lr.fit(xtrain,ytrain)
      ypred=lr.predict(xtest)
      print(ypred)
      print("training accuracy",lr.score(xtrain,ytrain))
      print("testing accuracu",lr.score(xtest,ytest))
      mse=mean_squared_error(ypred,ytest)
      print("mean squared error:",mse)
      r2_lr=r2_score(ypred,ytest)
      print("r2 score",r2_lr)
```

```
[704.66490019 461.9722937  731.88227169 ... 460.51045169 513.20263708
 609.95242294]
training accuracy 0.18924656267633866
testing accuracu 0.1951221617711716
mean squared error: 43985.58793110899
r2 score -3.2814933916061033
```

Decision tree model

```
[24]: dt=DecisionTreeRegressor()
      dt.fit(xtrain,ytrain)
      ypred=dt.predict(xtest)
      print(ypred)
      print("training accuracy",dt.score(xtrain,ytrain))
      print("testing accuracu",dt.score(xtest,ytest))
      mse=mean_squared_error(ypred,ytest)
      print("mean squared error:",mse)
      r2_dt=r2_score(ypred,ytest)
      print("r2 score",r2_dt)
```

```
[946.46 621.56 926.3  ... 626.3  219.42 825.17]
training accuracy 1.0
testing accuracu 0.9999996378442584
mean squared error: 0.01979136766100429
r2 score 0.9999996378446946
```

Random Forest Regressor

[25]:
```
rf=RandomForestRegressor()
rf.fit(xtrain,ytrain)
ypred=rf.predict(xtest)
print(ypred)
print("training accuracy",rf.score(xtrain,ytrain))
print("testing accuracu",rf.score(xtest,ytest))
mse=mean_squared_error(ypred,ytest)
print("mean squared error:",mse)
r2_rf=r2_score(ypred,ytest)
print("r2 score",r2_rf)
```

```
[946.46   621.56   926.3    ... 626.3    219.42   825.0917]
training accuracy 0.999999955348806
testing accuracu 0.9999997275270402
mean squared error: 0.014890313488231902
r2 score 0.9999997275267476
```

Comparing All The Models.

[26]:
```
Accuracys =pd.DataFrame({"models":["LinearRegression","DecisionTree","RandomForestRegressor"],
                         "R2score":[r2_lr,r2_dt,r2_rf]})
Accuracys
```

[26]:

	models	R2score
0	LinearRegression	-3.281493
1	DecisionTree	1.000000
2	RandomForestRegressor	1.000000

Testing The Model

[27]:
```
print(rf.predict(sts.transform([[6.78,1.22,14.84,3.5,0.83,41.63,26.52,0,7691.92,1.0]])))
```

```
[543.41]
```

[28]:
```
print(rf.predict(sts.transform([[5.55,4.55,66.5,4,555.2,55.3,666.7,5,888.6,66.5,]])))
```

```
[684.45]
```

[29]:
```
print(rf.predict(sts.transform([[6.56,4.54,67.5,4,545.2,54.3,454.7,4,77.4,0.12,]])))
```

```
[454.17]
```

[30]:
```
print(rf.predict(sts.transform([[2.81,1.75,23.04,6.29,0.76,39.02,33.63,3,4844.5,0.68]])))
```

```
[645.24]
```

Save And Load The Best Model

[ ]:
```
import pickle
pickle.dump(rf,open("honmodel.pkl","wb"))
```

[ ]:

10.3

Project Demo Link : http://127.0.0.1:5000/#section_2