

# **CHAPTER-1**

## **INTRODUCTION**

## **1.1 Title of Project:**

### **Click Connect**

## **1.2 Introduction**

The Clinic Connect project is a full-stack website designed to enhance healthcare management by utilizing machine learning and appointment scheduling functionalities. The platform provides healthcare providers and patients with a comprehensive solution for managing appointments, scheduling consultations, and receiving personalized recommendations based on predictive analytics.

The website is built using various technologies, including ReactJS, Node.js, and MongoDB, to ensure a seamless and intuitive user experience. The system incorporates machine learning algorithms to provide predictive analytics that aid healthcare providers in making informed decisions. The system also allows patients to schedule appointments with their preferred doctors and receive personalized recommendations based on their medical history and symptoms.

The Clinic Connect project addresses the challenges of traditional healthcare management systems by providing a unified platform that streamlines the process of healthcare management. The platform offers various features, including appointment scheduling, patient management, billing, inventory management, and predictive analytics. The system's user-friendly interface makes it easy for healthcare providers and patients to interact with the platform and receive the desired healthcare services.

### **2.1 Objective: -**

- The objective of the Clinic Connect project is to develop a full-stack website that utilizes machine learning and appointment scheduling functionalities to enhance healthcare management. The platform aims to provide healthcare providers and patients with a comprehensive solution that enables them to manage appointments, schedule consultations, and receive personalized recommendations based on predictive analytics. The system aims to streamline the process of

healthcare management by providing a unified platform for various clinic-related activities, such as patient management, appointment scheduling, billing, and inventory management.

## **2.2 Scope**

The scope of the Clinic Connect project includes the development of a full-stack website that offers a range of features and functionalities, including appointment scheduling, patient management, billing, and inventory management. The platform incorporates machine learning algorithms to provide predictive analytics that aid healthcare providers in making informed decisions and enable patients to receive personalized recommendations. The system will be designed to be scalable and robust, ensuring that it can handle a large volume of users and data.

## **2.3 Problem Statements**

appointment scheduling and management, the efficient handling of patient records and billing, and the need to offer personalized care to patients. The current system lacks a centralized solution that can integrate machine learning and appointment scheduling functionalities, resulting in inefficiencies, delays, and poor patient outcomes. The Clinic Connect project aims to address these challenges by developing a comprehensive solution that leverages modern technologies to enhance healthcare management. The healthcare industry faces several challenges, including the need for improved

## **2.4 Motivation**

The healthcare industry faces several challenges, including the need for improved appointment scheduling and management, the efficient handling of patient records and billing, and the need to offer personalized care to patients. The current system lacks a centralized solution that can integrate machine learning and appointment scheduling functionalities, resulting in inefficiencies, delays, and poor patient outcomes. The motivation behind the Clinic Connect project is to address these challenges by developing a comprehensive solution that leverages modern technologies to enhance healthcare management. The platform aims to improve the patient experience, reduce wait times, and improve the accuracy of diagnosis and treatment by providing healthcare providers with predictive analytics based on patient data. The platform also aims to improve the efficiency of healthcare management by providing a centralized solution for managing appointments, patient records, billing, and inventory management, leading to better outcomes for patients and healthcare providers.

# **CHAPTER-2**

## **TECHNOLOGY /PLATFORM OVERVIEW**

## **2.1 Software requirements:**

- Operating System: Windows 10
- IDE: - Visual Code, Pycharm, Google Colab
- Browser: - Goggle Chrome
- Languages: - Python
- Library: - Pandas, numpy, matplotlib, scikit learn, seaborn, Flask
- Model: Linear Regression, logistic regression.

## **2.2 Hardware requirements:**

- Processor: Intel I5
- RAM :8GB
- ROM:50GB
- System Type: 64 -bit
- Monitor :15” Color Monitor
- Mouse
- Keyboard

## **2.3 Software Features:**

### **2.3.1 PYTHON: -**

Python is a high-level, interpretive, interactive and scripting language. Python is designed to be very readable. While other languages use punctuation, they generally use English words and contain fewer patterns than other languages.

- Python is interpreted - Python is executed by the interpreter at runtime. You do not need to calculate before running your program.
- This is similar to PERL and PHP.

- Python is interactive - you can quickly switch to Python and interact directly with the interpreter to write your programs.
- Python is object-oriented - Python supports the object-oriented style or programming technique of encapsulating code in objects.
- Python is a Beginner's Language - Python is an excellent language for beginner programmers and supports the development of a wide variety of applications, from simple scripts to WWW browsers and games.

## **CHAPTER-3**

### **METHODOLOGY**



We use a variety of classical and cutting-edge methods, including 90%-10% training and data separation and learning methods. We use examples from the literature to reduce the time required for training. Linear regression is our basic technique. The open-source package Scikit-Learn was used for most of the examples used.

### **3.1 Linear Regression**

Linear regression was chosen as the first model because of its simplicity and short training time. These features do not have any special maps and are used directly as vectors. Regularization was not used as the results were clearly less variable.

### **3.2 Logistic Regression**

Logistic regression is a statistical model used to predict the probability of a binary outcome based on one or more independent variables. It is a type of regression analysis commonly used in machine learning and statistics for classification problems where the dependent variable is categorical or binary.

In logistic regression, the dependent variable, also known as the outcome or response variable, is typically binary and can take only two possible values, such as "yes" or "no," "success" or "failure," or 1 or 0. The independent variables, also called predictors or features, can be either continuous or categorical.

### **3.3 Dataset Discussion**

For this project, we are using the dataset on used medical cost from all over the India, available on Kaggle.

File Edit Selection View Go Run ... Search

nbserver-18284-open.html app.py home.html doctorRecommend.html query.html Medical Cost Insurance-checkpoint.ipynb

C:\Users\mimsh> OneDrive > Desktop > Medical-Cost-Prediction-master > .ipynb\_checkpoints > Medical Cost Insurance-checkpoint.ipynb > data.info()

+ Code + Markdown Run All Clear All Outputs Outline ... Select Kernel

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

```

[1] Python

```

data = pd.read_csv('./insurance.csv')
data.head()

```

[2] Python

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER SQL CONSOLE COMMENTS

```

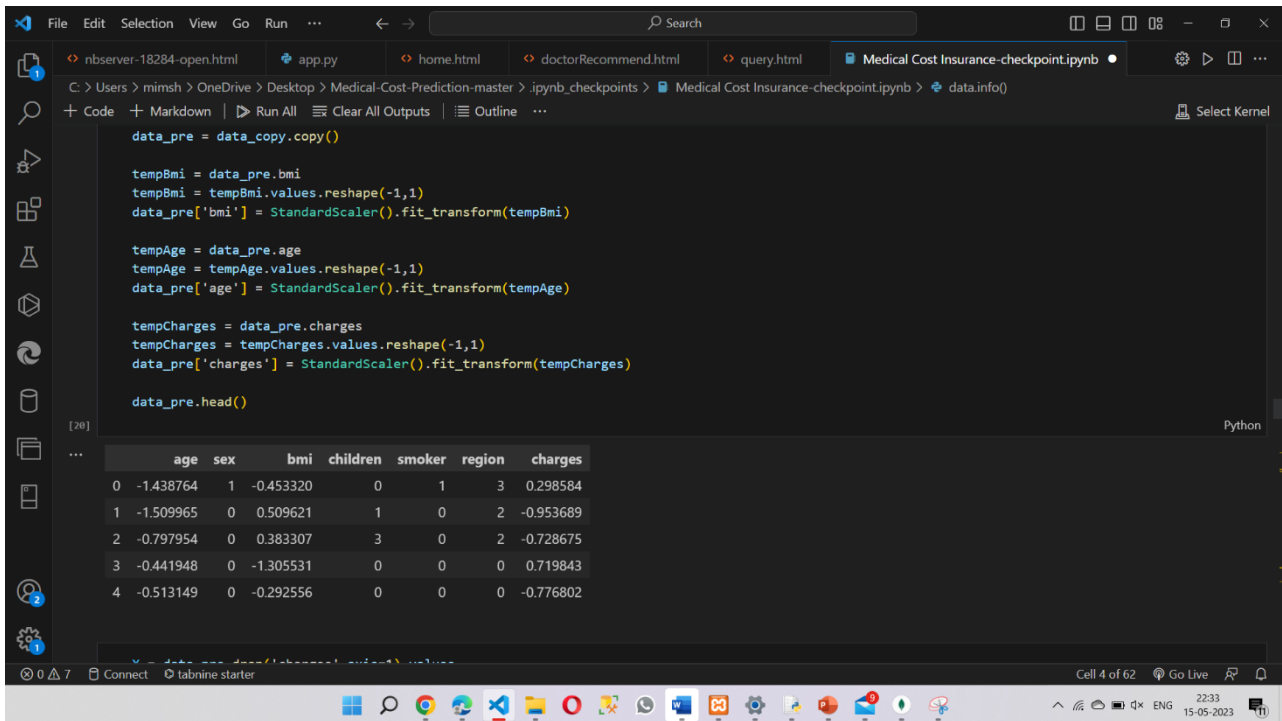
<pymongo.cursor.Cursor object at 0x0000023073573790>
1
127.0.0.1 - - [15/May/2023 18:17:28] "POST /submit HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2023 18:17:28] "GET /static/Dr.AbhaySimba.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2023 18:17:28] "GET /static/Dr.RakeshKumarRaushan.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2023 18:17:28] "GET /static/Dr.Rakesh.jpg HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2023 18:28:18] "GET /doctorRecommend HTTP/1.1" 200 -

```

Cell 4 of 62 Go Live

22:32 15-05-2023

**Fig: Dataset with 4 rows**



```
data_pre = data_copy.copy()

tempBmi = data_pre.bmi
tempBmi = tempBmi.values.reshape(-1,1)
data_pre['bmi'] = StandardScaler().fit_transform(tempBmi)

tempAge = data_pre.age
tempAge = tempAge.values.reshape(-1,1)
data_pre['age'] = StandardScaler().fit_transform(tempAge)

tempCharges = data_pre.charges
tempCharges = tempCharges.values.reshape(-1,1)
data_pre['charges'] = StandardScaler().fit_transform(tempCharges)

data_pre.head()
```

[20]

	age	sex	bmi	children	smoker	region	charges
0	-1.438764	1	-0.453320	0	1	3	0.298584
1	-1.509965	0	0.509621	1	0	2	-0.953689
2	-0.797954	0	0.383307	3	0	2	-0.728675
3	-0.441948	0	-1.305531	0	0	0	0.719843
4	-0.513149	0	-0.292556	0	0	0	-0.776802

### 3.4 Model Deployment

- We use an open-source python framework called Flask.
- Helps us build web applications for data science and machine learning in no time.
- Flask works by reading code directly from the public GitHub repository.  
So, we need a repository with `o` as a .pkl file for our model for the app to work.

# **CHAPTER 4**

## **LANGUAGES USED**

## 4.1 HTML

HTML, stands for Hypertext Markup Language and is the standard markup language used to create web pages and web applications. It is the foundation of the World Wide Web and is responsible for the structure and presentation of web content. Here are some key points about HTML: Structure: HTML uses a markup style with elements and symbols to describe different parts of a web page. Points are represented by symbols enclosed in right angles (<>). Tags: HTML tags are used to mark up and describe the structure of web pages. They can be divided into two types: boxes and spaces. A text box has an opening title, content, and a closing tag, while a tag tag has no content and closes by itself. Elements: HTML elements consist of tags and elements that represent different types of content on a web page. For example, represents the top element, represents a paragraph, and represents an image. Attributes: HTML tags can contain attributes that provide additional information or change the behavior of the element. Attributes are contained in an element's opening tag and have a name and a value. For example, the src attribute in the tag specifies the URL of the image. Document Structure: HTML documents usually have several main sections: and.



HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1999.

### 4.1.1 Features of HTML

Markup language: HTML is a markup language used to structure and organize the content on a web page.

Structure and semantics: HTML provide a range of elements and tags to define the structure and semantics of content, allowing developers to convey meaning and hierarchy to browsers and other tools.

1. Cross-platform compatibility: HTML is supported by all major web browsers and can be rendered on different operating systems and devices, ensuring broad accessibility and compatibility.
2. Hyperlinking: HTML enables the creation of hyperlinks that allow users

to navigate between web pages, enhancing the interconnected nature of the World Wide Web.

3. Media integration: HTML supports the embedding and presentation of various types of media, including images, audio, video, and interactive content, enhancing the user experience.
4. Forms and input handling: HTML provides form elements that allow users to input data and interact with web applications. It supports different input types, validation, and submission to server-side processing.
5. Accessibility features: HTML incorporates accessibility features, such as alt attributes for images, semantic elements, and ARIA roles, enabling developers to create web content that is inclusive and usable by people with disabilities.
6. Extensibility: HTML can be extended and enhanced using other technologies like CSS and JavaScript, allowing developers to create dynamic and interactive web pages.
7. Mobile responsiveness: HTML offers features like responsive design, media queries, and viewport meta tags to ensure that web pages adapt and display appropriately on different screen sizes and devices.
8. SEO-friendly: HTML provides elements and attributes that can help optimize web pages for search engine visibility, such as meta tags, heading tags, and descriptive content.

### **4.1.2 Why to learn HTML**

1. Fundamental Web Language: HTML is the fundamental language of the web. It provides the structure and organization for web content, allowing you to create and format text, images, links, and other elements. By learning HTML, you gain the essential skills to build and customize web pages.

2. Independence and Control: Learning HTML gives you independence and control over your online presence. Instead of relying solely on pre-built website builders or content management systems, you can create and
3. modify web pages according to your specific needs. This control allows for greater flexibility and creativity in designing and maintaining your online projects.
4. Collaboration and Communication: HTML is a universal language understood by web developers and designers worldwide. By learning HTML, you can effectively communicate and collaborate with professionals in the field. Whether you're working on a team project or seeking advice from peers, HTML knowledge enables seamless collaboration and sharing of ideas.
5. Career Advancement: HTML is a fundamental skill sought after in the tech industry. Understanding HTML opens up a range of career opportunities in web development, web design, content creation, digital marketing, and
6. more. It serves as a solid foundation for expanding your skills and branching into other web technologies such as CSS and JavaScript.



### **4.1.3 Advantages**

- Easy to learn and understand.
- Platform-independent and supported by all major web browsers.
- Provides a standardized structure for organizing web content.
- Allows for the creation of hyperlinks, enabling seamless navigation between pages.
- Supports multimedia integration, such as images, audio, and video.
- Incorporates accessibility features, making web content more inclusive and usable.
- Provides a foundation for building responsive and mobile-friendly websites.

### **4.1.4**

### **4.1.5**

### **4.1.6 Disadvantages**

- Complex interactivity requires additional scripting languages like JavaScript. Design capabilities are limited compared to CSS.
- HTML lacks built-in security features.
- Compatibility issues can arise across different web browsers
- Offline functionality requires additional technologies. Ensuring full accessibility compliance can be complex.

### **4.1.7 Why to use HTML**

HTML is used to create the structure and content of web pages. It provides a standardized way to organize and present information on the internet. By using HTML, you can create web pages that are accessible, compatible across different

browsers, and easily navigable through hyperlinks. HTML serves as the foundation for web development, allowing you to create and deliver content on the World Wide Web.

## 4.2 CSS

CSS, which stands for Cascading Style Sheets, is a styling language used to describe the visual presentation and formatting of HTML (Hypertext Markup Language) and XML (extensible Markup Language) documents. It allows web developers to control the layout, colors, fonts, and other visual aspects of web pages.

CSS works by selecting HTML elements using selectors and applying style properties to them. The style properties define how the selected elements should appear, such as the font size, color, margin, padding, background, and more. CSS provides a wide range of style properties and values that allow developers to customize the appearance of individual elements or groups of elements.



#### 4.2.1 WHY CSS

- **Separation of Concerns:** CSS enables the separation of presentation (styles) from the structure and content (HTML) of a web page. This separation improves code organization, readability, and maintainability, making it easier to update the visual appearance without altering the underlying structure.
- **Visual Styling:** CSS allows developers to control the visual aspects of web pages, including colors, fonts, sizes, margins, paddings, and more. It provides precise control over the presentation, allowing for consistent branding and a cohesive design across the website.
- **Layout and Positioning:** CSS provides layout mechanisms that allow developers to control the placement and arrangement of elements on a web page. Flexbox and CSS Grid are examples of layout techniques that offer responsive and flexible positioning options, enabling developers to create dynamic and adaptive layouts.
- **Responsive Web Design:** CSS facilitates the creation of responsive web designs that adapt to different screen sizes and devices. With media queries, developers can apply specific styles based on factors like screen width, allowing for a seamless and optimized user experience across desktop, tablet, and mobile devices.

### CSS Advantages

- **Separation of Concerns:** CSS allows the separation of presentation (styles) from the structure and content (HTML) of a web page. This separation improves code organization, readability, and maintainability. Changes can be made to the style without impacting the underlying structure, making it easier to update and modify the design of a website.
- **Consistent Styling:** CSS enables consistent styling across multiple web pages. By defining styles in a centralized CSS file, developers can apply the same styles to different HTML elements throughout the website. This ensures a cohesive and uniform visual appearance, providing a better user experience.
- **Efficient Updates:** With CSS, making changes to the style of a web page becomes more efficient. By modifying a single CSS rule, developers can update the appearance of multiple elements simultaneously. This saves time and effort, especially in larger projects, as it eliminates the need to manually update each individual element.

## 4.3 Flask

Flask is a lightweight and flexible web framework for Python. It is designed to make web development in Python simple and easy, providing the necessary tools and features to build web applications. Flask follows the principle of being a "micro" framework, meaning it has a minimalistic core and focuses on simplicity and extensibility.

Some key features of Flask include:

- **Routing:** Flask provides a routing system that allows developers to map URL patterns to specific functions or views, making it easy to handle different HTTP requests and navigate between pages.
- **Templating:** Flask supports a templating engine (Jinja2) that allows developers to create dynamic and reusable HTML templates. Templating enables the separation of logic and presentation, making it easier to generate dynamic content for web pages.
- **HTTP Request and Response Handling:** Flask provides convenient methods for handling incoming HTTP requests and generating HTTP responses. This includes handling form submissions, handling file uploads, setting cookies, and more.
- **Web Development Tools:** Flask offers a range of tools and utilities to aid web development, including a built-in development server for testing, a debug mode for troubleshooting, and a command-line interface (CLI) for managing the application.
- **Integration with Databases:** Flask seamlessly integrates with various database systems, such as SQLite, MySQL, PostgreSQL, and more. It provides support for connecting to databases, executing queries, and managing data within the web application.
- **Flask Extensions:** Flask has a rich ecosystem of extensions that enhance its functionality. These extensions provide additional features such as authentication, session management, caching, form handling, and integration with third-party services.
- **Lightweight and Scalable:** Due to its minimalist design, Flask is lightweight and has minimal overhead. This makes it efficient and suitable for small to medium-sized projects. Flask also allows developers to scale their applications by integrating with other libraries and frameworks as needed.
- **Flask is highly regarded for its simplicity, flexibility, and ease of use.** It is popular among Python developers who want a lightweight framework for building web applications, APIs, and prototypes quickly.



#### 4.3.1 The benefit of the flask

There are impressive features to use the flask in your web framework. Like-

- Integrated support for unit testing
- Built-in development server and fast debugger
- Restful request dispatching
- Unicode base
- Support for cookies
- Templating jinja2
- WSGI 1.0 compliant
- Plus, flask gives you some premier control to develop your project.
- HTTP request handling function
- Flask has a modular design and lightweight so that it can easy to transit into web framework with some extension
- You can plug your favorite ORM
- Basic fundamental API is nicely shaped and coherent
- Highly flexible
- It is easy to deploy the flask in production

## **Conclusion:**

Flask is the most policed and feature-rich micro framework. Flask comes with all its benefit of the fast template, strong WSGI features, and extensive documentation. Flask gives lots of good features, vast no of extension facility for a new project.

### **4.3.2 Objective**

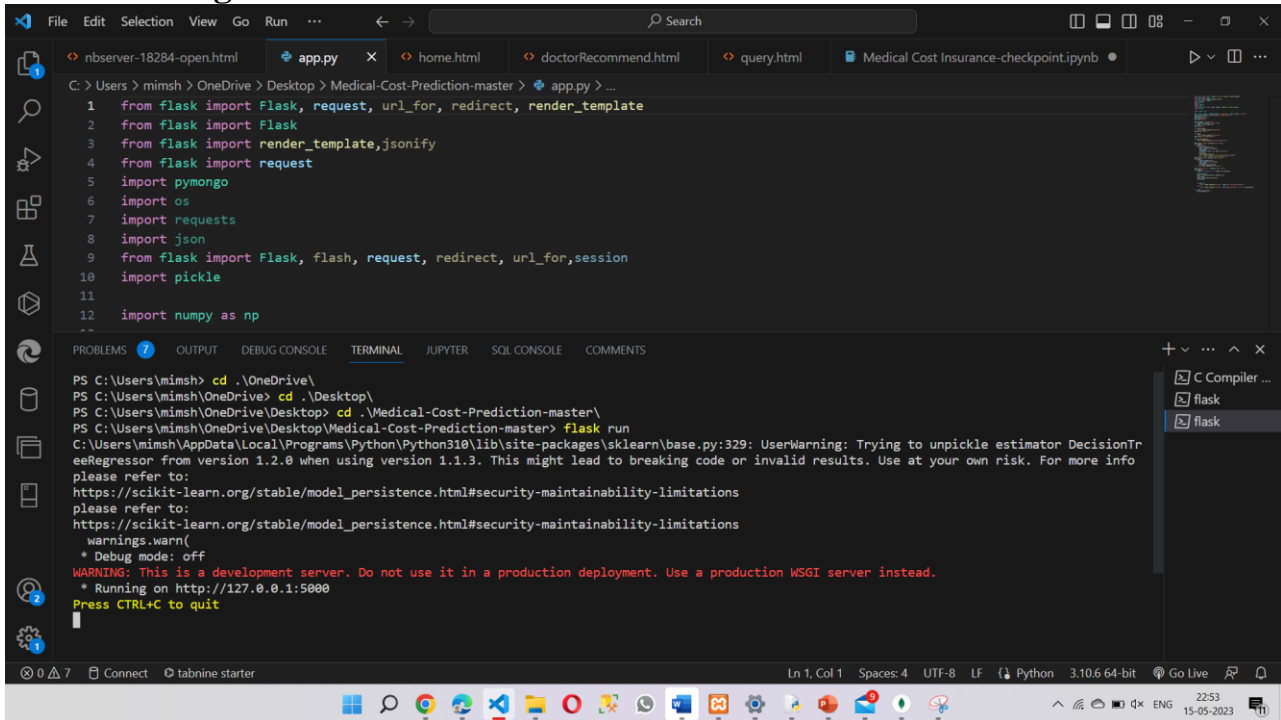
- The objective of our project is to recommend doctors.
- Data used in this article is publicly available at Kaggle.
- They can detect disease using various factors.
- We don't need to do a lot of manual works.

# **CHAPTER-5**

## **OUTPUT**



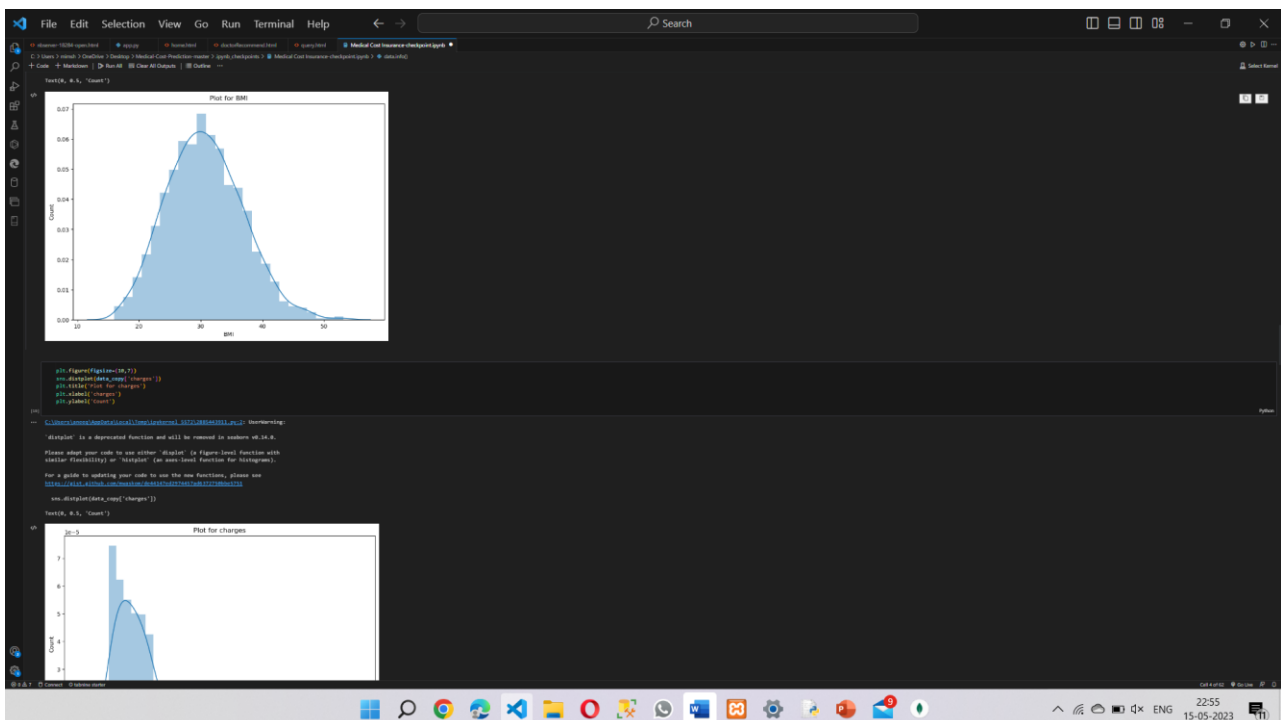
## 5.1 Starting Server:



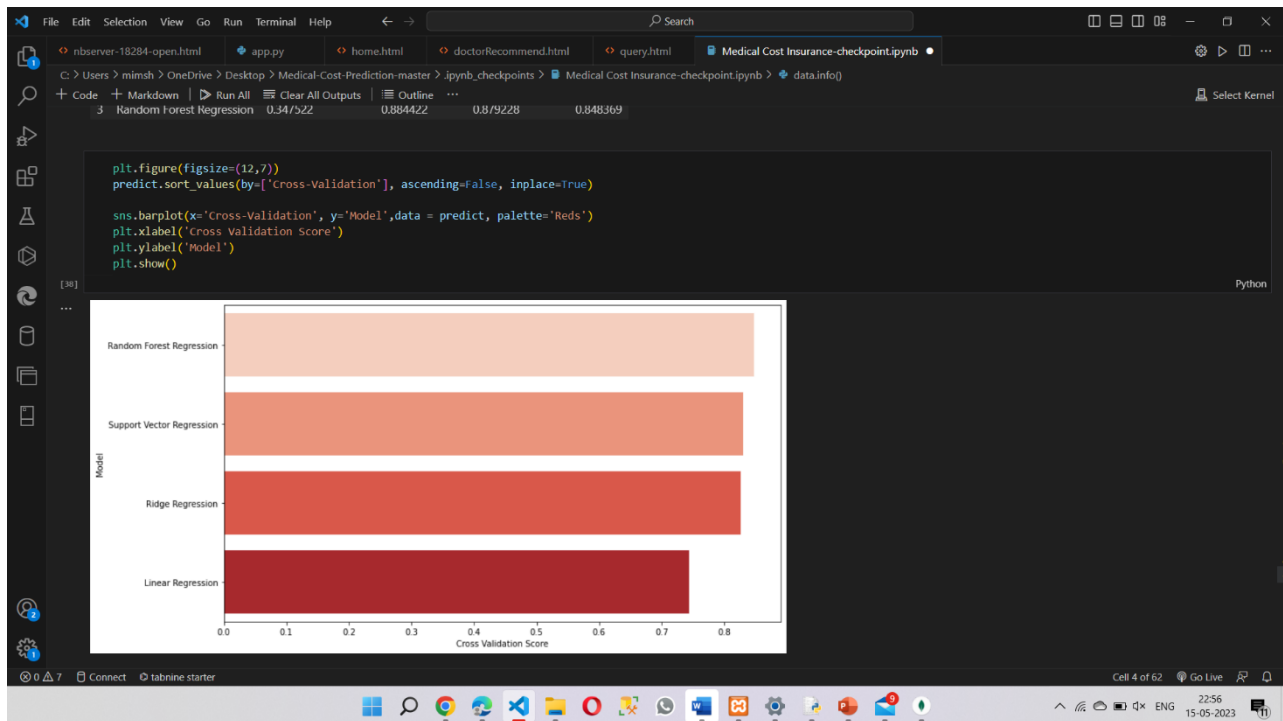
```
1 from flask import Flask, request, url_for, redirect, render_template
2 from flask import Flask
3 from flask import render_template, jsonify
4 from flask import request
5 import pymongo
6 import os
7 import requests
8 import json
9 from flask import Flask, flash, request, redirect, url_for, session
10 import pickle
11
12 import numpy as np
```

```
PS C:\Users\mimsh> cd .\OneDrive\
PS C:\Users\mimsh\OneDrive> cd .\Desktop\
PS C:\Users\mimsh\OneDrive\Desktop> cd .\Medical-Cost-Prediction-master\
PS C:\Users\mimsh\OneDrive\Desktop\Medical-Cost-Prediction-master> flask run
C:\Users\mimsh\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.2.0 when using version 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

## 5.2 Visualization of Trained Data:



## 5.3 Visualization of Test Data:



## 5.3 Homepage

Clinik Connect Home Doctor Recommend Query

# Medical Cost Prediction

Predict the Medical Cost!

Age	Gender	BMI	Childrens	Do you Smoke?	Which Region?
43	1	26	2	1	Bihar

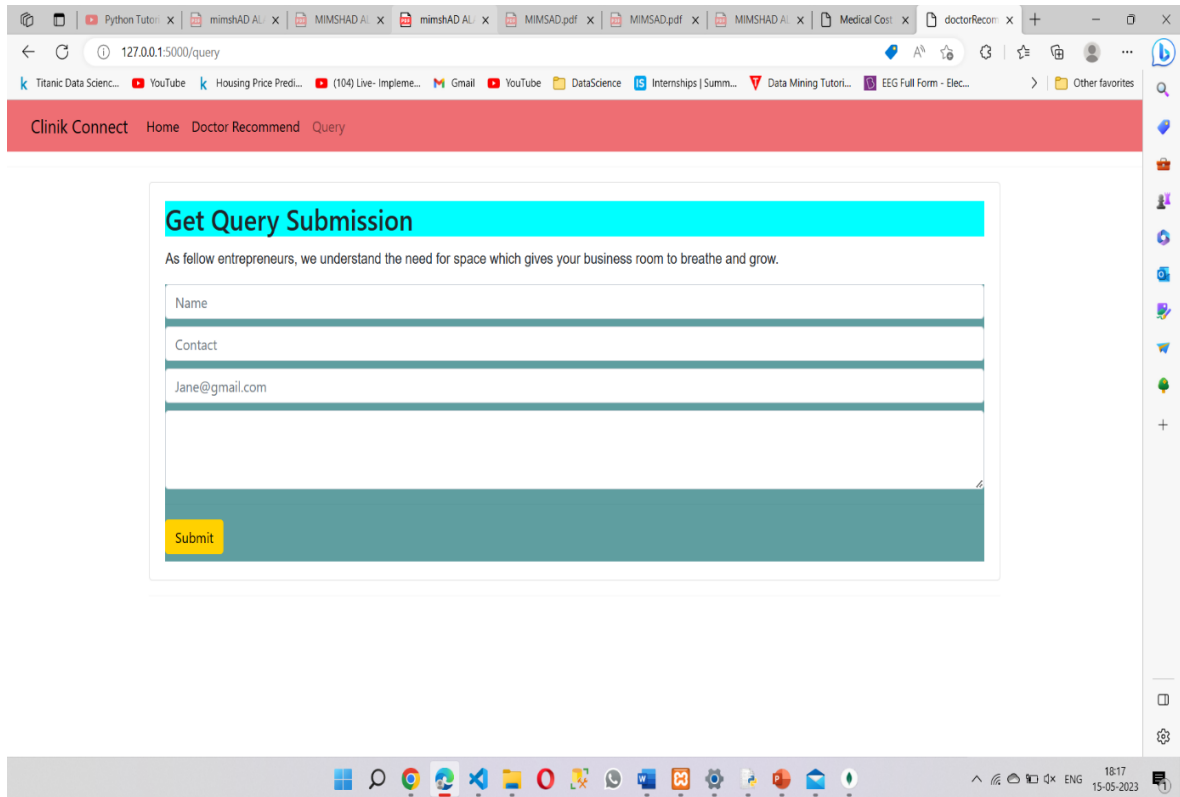
PREDICT PROBABILITY

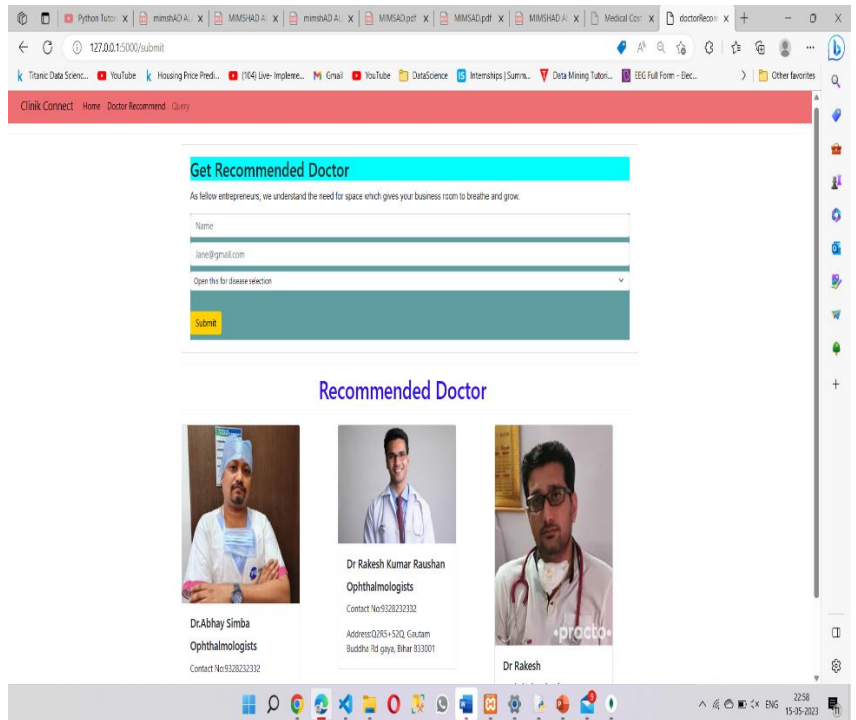
## 5.4 Homepage with Result

Logo

Predicted Amount :

Expected amount is 25041.684





## CHAPTER-6

## **CONCLUSION AND FUTURE WORK**

## **6.1 Conclusion**

1. In conclusion, the development of a Full Stack website project for expert doctor recommendations is essential to address the need for reliable, personalized, and evidence-based health guidance for patients and users. The main features outlined above, including user registration and authentication, personalized recommendations, comprehensive medical information, appointment booking and management, interactive tools and features, content management system, responsive design, data privacy and security, SEO optimization, and feedback and reviews, collectively create a robust and user-friendly platform that can provide valuable health insights and support to users.
2. By leveraging technology and expert medical knowledge, this website project can bridge the gap between patients and expert doctors, empowering users to make informed decisions about their health and well-being. It can also streamline appointment scheduling and content management for the expert doctor and their team, enhancing efficiency and effectiveness.
3. In summary, the development of a Full Stack website project for expert doctor recommendations has the potential to revolutionize the way patients access and receive expert medical guidance online, resulting in improved health outcomes and user satisfaction.

## **6.2 Advantages**

- I.** Good at learning complex and non – linear relationships
- II.** Highly explainable and easy to interpret
- III.** Robust to Outliers
- IV.** No feature Scaling is Required

### **6.3 Limitations:**

- This web app can only work on Linear Regression model.
- The low number of records that have been used.

### **6.4 Future Work:**

The feature scope of this project is to create a full-stack website that provides healthcare management solutions by using machine learning and appointment scheduling functionalities. The website should have the following features:

- A user authentication system that allows different types of users, such as patients, providers, and administrators, to register and log in to the website
- A user profile system that allows users to view and edit their personal information, medical history, preferences, and feedback
- A patient management system that allows providers to view and manage their patients' records, diagnoses, treatments, and follow-ups
- A billing system that allows providers to generate and send invoices to their patients and receive payments
- An inventory management system that allows providers to track and order their medical supplies and equipment
- A predictive analytics system that uses machine learning algorithms to analyze data from various sources and provide personalized recommendations to patients and providers based on their needs and goals
- An appointment scheduling system that uses machine learning algorithms to optimize the allocation of appointment slots by identifying the patients with the highest no-show risk and overbooking them accordingly



- A notification system that sends reminders and alerts to users via email or text message about their appointments, payments, prescriptions, and feedback
- A feedback system that allows users to rate and review their experiences with the website and the providers
- A dashboard system that allows users to view and access various features and information on the website

## **CHAPTER-7**

## **REFERENCE**

## 7.1 References

- [www.analyticsvidhya.com](http://www.analyticsvidhya.com)
- [www.medium.com](http://www.medium.com)
- [www.towardsdatascience.com](http://www.towardsdatascience.com)
- [www.youtube.com](http://www.youtube.com)
- Diaz Ochoa, J. G., Scissor, O., & Schimper, T. (2021). Medical recommender systems based on continuous-valued logic and multi-criteria decision operators, using interpretable neural networks. *BMC Medical Informatics and Decision Making*, 21(1), 186<sup>1</sup>
- Samorani, M., & Blount, L. G. (2020). Machine learning and medical appointment scheduling: Creating and perpetuating inequalities in access to health care. *American Journal of Public Health*, 110(4), 473-474.
- Kureshi, S., Mondal, A., & Mukherjee, N. (2020). Building a trust-based doctor recommendation system on top of multilayer graph database. *Journal of Biomedical Informatics*, 110, 103549<sup>2</sup>
- Liu, Y., Zhang, Y., & Wang, X. (2020). Online doctor recommendation with convolutional neural networks and probabilistic matrix factorization. *Computational Intelligence and Neuroscience*, 2020<sup>3</sup>
- Wang, Y., Liang, C., Zhang, J., & Xu, K. (2019). A personalized doctor recommendation system based on big data analysis. *IEEE Access*, 7, 113894-113903.
- Liu, Z., Wang, Y., Liang, C., & Xu, K. (2018). A personalized doctor recommendation system based on medical record analysis and collaborative filtering algorithm. In *Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1375-1380.