HEALTH ANALYSIS USING ML

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the
requirement for the award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

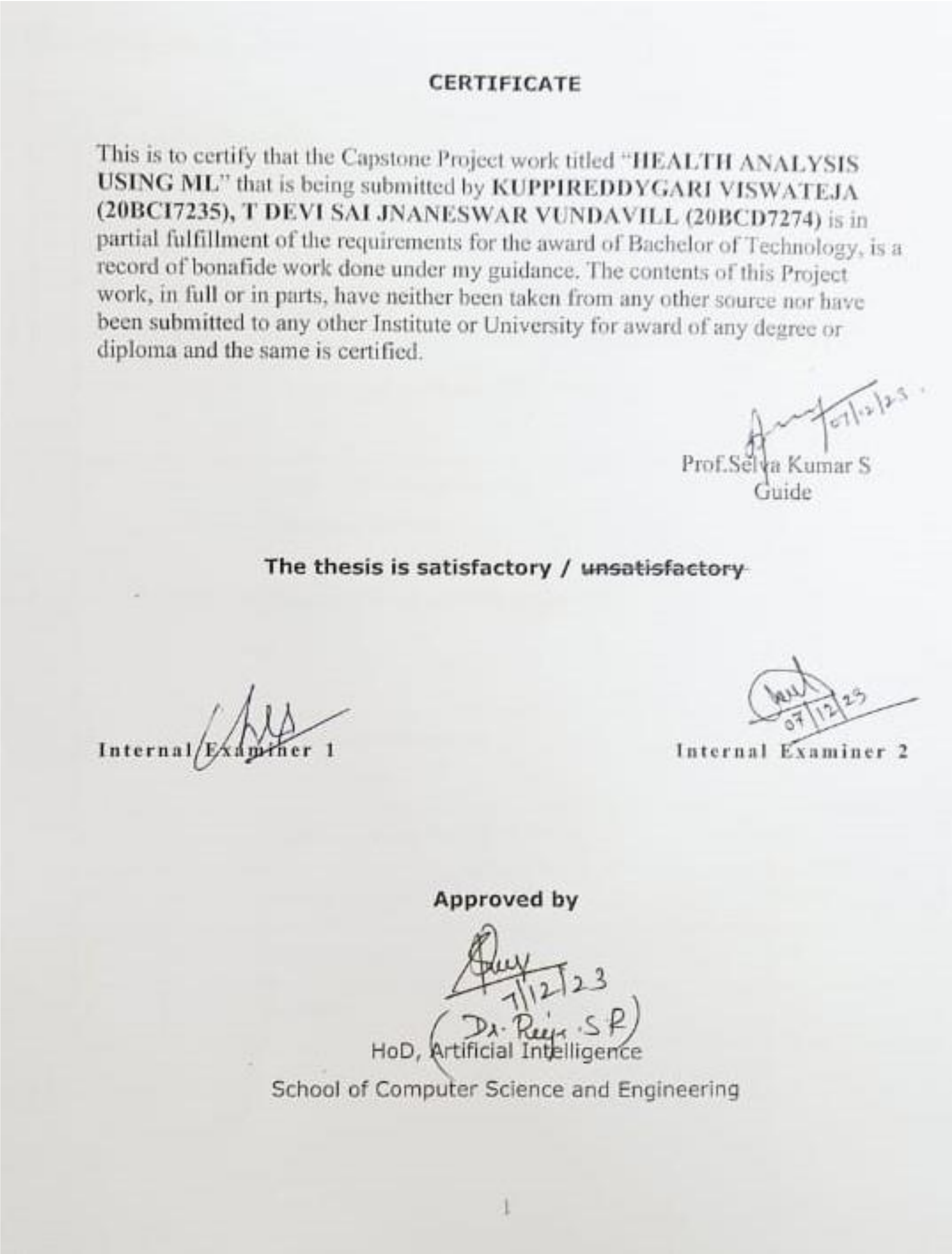
KUPPIREDDYGARI VISWATEJA (20BCI7235)
T DEVI SAI JNANESWAR VUNDAVILL (20BCD7274)

Under the Guidance of

Prof.Selva Kumar S

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

DECEMEBR 2023



CERTIFICATE

This is to certify that the Capstone Project work titled "**HEALTH ANALYSIS USING ML**" that is being submitted by **KUPPIREDDYGARI VISWATEJA (20BCI7235), T DEVI SAI JNANESWAR VUNDAVILL (20BCD7274)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

07/12/23

Prof.Selva Kumar S
Guide

The thesis is satisfactory / ~~unsatisfactory~~

Internal Examiner 1

07/12/23

Internal Examiner 2

Approved by

7/12/23

(Dr. Reeja .S R)

HoD, Artificial Intelligence

School of Computer Science and Engineering

Table of Content:

List of Figures

Abstract:

This project introduces an innovative method to estimate an individual's immunity power by merging lifestyle factors with machine learning techniques. It utilizes a dataset comprising approximately 470 responses from a custom Google Form, encompassing a wide range of data including age, gender, sleep patterns, diet, stress, water intake, medical conditions, and frequency of doctor visits. The project leverages Python for data preprocessing, employing Label Encoding for categorical variables, and utilizes a Random Forest Regressor for model creation. The model's reliability is assured through training and evaluation based on the r2_score metric. This unique blend of lifestyle analytics and sophisticated computational strategies provides deeper insights into the factors affecting immunity. The project's practical aspect is manifested in a user-friendly Streamlit-based interface, where users can input their lifestyle data. The application, in real-time, utilizes the trained model to predict immunity levels and offers personalized health advice based on these predictions. This feature is particularly relevant in today's health-aware society, offering individuals valuable and actionable insights into their health and well-being, thereby demonstrating the vast potential of integrating machine learning into personal health management and promoting more informed health decisions.

CHAPTER 1
INTRODUCTION

In the current era, where personal health and wellness are paramount, there is a growing need for tools that can provide insights into an individual’s health status. Recognizing this need, this project focuses on developing an innovative solution to predict an individual’s immunity power. The core idea is to amalgamate lifestyle data with advanced machine learning techniques to offer a personalized health assessment. The project utilizes a dataset, self-compiled through a Google Form, which includes responses regarding various lifestyle factors like diet, sleep patterns, stress levels, and medical history. By applying machine learning algorithms to this data, the project aims to derive meaningful predictions about an individual’s immunity power. The choice of a Random Forest Regressor model for this task stems from its efficacy in handling complex, non-linear relationships inherent in such diverse data.

The implementation of this model in a user-friendly interface, using Streamlit, marks a significant stride in making advanced health analytics accessible to the general public. This interface allows users to input their lifestyle data and receive an immediate prediction of their immunity power, along with personalized health suggestions. This approach not only democratizes access to health analytics but also encourages individuals to engage actively with their health data.

the integration of Streamlit for the user interface is a pivotal aspect. Streamlit's intuitive and interactive platform enhances user experience, making it simpler for individuals with varying degrees of tech-savviness to interact with the model. The interface will be designed to guide users through a seamless process of inputting their lifestyle data, including dietary habits, sleep duration and quality, stress levels, exercise routines, and relevant medical history.

The project also places a strong emphasis on data privacy and security, understanding the sensitive nature of personal health data. Robust data encryption and anonymization protocols will be implemented to ensure user data is securely stored and processed. This aspect is crucial in building trust and encouraging wider participation.

In addition, there will be a continuous improvement mechanism in place. The model will evolve through machine learning algorithms that learn from new data, enhancing accuracy and reliability over time. User feedback will also be integral, providing insights into how the application can be further refined to meet user needs more effectively.

Another important aspect of the project is the educational component. The application will not only provide predictions but also educate users about the factors affecting their immunity. This will be achieved through informational tooltips and links to credible health resources within the app. By increasing awareness about the impact of lifestyle choices on health, the project aims to empower users to make informed decisions about their wellbeing.

To ensure the project's success and relevance, collaborations with healthcare professionals and wellness experts are planned. Their insights will guide the development of personalized health suggestions, ensuring they are practical, scientifically sound, and tailored to individual needs.

Finally, the project envisions expanding its scope to include predictive analytics for other aspects of health, leveraging the power of machine learning to offer a comprehensive health assessment tool. This expansion could potentially include mental health indicators, risk assessments for chronic diseases, and recommendations for preventive healthcare measures.

Overall, this project represents a significant step forward in personalized health analytics, combining advanced technology, user-friendly design, and a commitment to health education and empowerment.

1.1 Objections:

To achieve our goal of enhancing personal healthcare through technology, we have embarked on a comprehensive project encompassing several key objectives. Initially, we focus on developing a robust machine learning model, specifically a Random Forest Regressor, to accurately predict an individual's immunity power based on various lifestyle factors. This task requires meticulous data collection and preprocessing, including the use of a custom Google Form for data gathering and Label Encoding for data formatting. Concurrently, we are committed to designing a user-friendly interface using Streamlit, ensuring that our application is accessible and easy to use for individuals seeking to understand their immunity levels. In addition to providing predictive insights, our application offers personalized health recommendations, empowering users with actionable advice to potentially improve their immunity and overall well-being.

We understand the importance of continuous improvement and validation; hence, we are dedicated to regularly testing and refining our model to enhance its predictive accuracy. Ultimately, our project aims to contribute significantly to the field of personal healthcare by increasing accessibility and awareness about the impact of lifestyle choices on immunity, thereby enabling individuals to make informed health decisions.

1. **Develop a Predictive Model:** The primary objective is to build a robust machine learning model capable of accurately predicting an individual’s immunity power based on various lifestyle factors. This involves selecting the appropriate algorithm, in this case, Random Forest Regressor, and fine-tuning it for optimal performance.
2. **Data Collection and Preprocessing:** Gathering a comprehensive and diverse dataset through a custom Google Form, followed by meticulous preprocessing, including Label Encoding, to ensure the data is in a format suitable for the machine learning model.
3. **User-Friendly Interface Creation:** Designing and deploying a Streamlit-based application that provides a simple and intuitive interface for users to input their lifestyle data and receive predictions. This objective focuses on enhancing user engagement and ease of access to complex health predictions.
4. **Health Recommendations:** Offering personalized health suggestions based on the predicted immunity scores. This aims to provide users with actionable insights to potentially improve their immunity and overall health.

5. **Healthcare Accessibility and Awareness:** The broader objective is to contribute to the field of personal healthcare by providing an accessible tool that raises awareness about the impact of lifestyle choices on immunity. It aspires to empower users with knowledge and guidance to make informed decisions regarding their health and well-being.
6. **Model Validation and Improvement:** Continuously validate and improve the predictive accuracy of the model. This involves regularly testing the model against new data, fine-tuning parameters, and possibly integrating more complex algorithms or additional relevant data sources to enhance prediction accuracy.

1.2 Background and Literature Survey

The concept of predicting individual health outcomes, particularly immunity power, through data analytics has garnered significant attention in recent years. With the advancement of machine learning and data processing technologies, the potential to analyze complex lifestyle data and predict health outcomes has expanded tremendously. The increasing prevalence of lifestyle-related health issues and the growing awareness of preventive healthcare have further fueled interest in this field. The use of machine learning models, like the Random Forest Regressor, has shown promising results in various health prediction domains due to their ability to handle large, diverse datasets and capture complex, non-linear relationships.

In the context of personal health management, there's a rising trend towards self-monitoring and wellness tracking, facilitated by various digital health tools and applications.

1) Title : Inflammation and Activated Innate Immunity in the Pathogenesis of Type 2 Diabetes

The article emphasizes the critical role of the body's immune system, particularly inflammation and innate immunity, in the development of type 2 diabetes. Factors like nutrition and physical inactivity are pivotal in this context. Poor nutrition, characterized by high-calorie, sugar-rich, and unhealthy fat diets, can induce obesity, a major risk factor for type 2 diabetes. Obesity leads to an inflammatory response as part of the innate immune system's defence mechanism. This response can disrupt normal insulin function, leading to insulin resistance, a hallmark of type 2 diabetes. Similarly, lack of physical activity exacerbates this process. Exercise is known for its anti-inflammatory effects and its role in maintaining healthy blood sugar levels. Without regular physical activity, the body is more susceptible to obesity and insulin resistance, fueling the cycle of inflammation and increasing diabetes risk.

Additionally, the article discusses how age, genetic predisposition, and stress contribute to the onset of type 2 diabetes through immune system activation. Aging is associated with 'inflammaging', a state of chronic low-grade inflammation that can affect insulin sensitivity. Genetics also play a crucial role; certain genes make individuals more prone to inflammation and dysfunctional immune responses, elevating diabetes risk. Stress, too, is a significant factor. It triggers the release of hormones like cortisol, which can raise blood sugar levels and affect insulin sensitivity. Moreover, stress induces an inflammatory response, part of the innate immune reaction, further linking it to the development of type 2 diabetes. This complex interplay of factors underscores the importance of understanding the immune system's role in metabolic disorders, offering new perspectives on diabetes management and prevention.

2) Title : Potential Immune Indicators for Predicting the Prognosis of COVID-19 and Trauma: Similarities and Disparities

This is an article about potential immune indicators for predicting the prognosis of COVID-19 and trauma. It discusses the challenges of predicting the prognosis of COVID-19 in the context of other inflammatory diseases. The article also investigates the similarities and differences of common inflammatory mediators between patients with COVID-19 and trauma. The authors propose that these mediators may help to accurately predict the severity of COVID-19 complications.

The scientific article investigates the role of immune system mediators in predicting COVID-19 outcomes, particularly in the context of trauma. Additionally, it acknowledges the challenges in accurately predicting COVID-19 outcomes due to the dynamic nature of immune responses

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the data collection, proposed system, working methodology and software details.
- Chapter 3 discusses the results obtained after the project was implemented.
- Chapter 4 concludes the report.
- Chapter 5 consists of codes.

CHAPTER 2

Data Collection Information

In our project, we have meticulously designed a Google Form to collect comprehensive data that delves into various lifestyle and health-related aspects of individuals. This data is invaluable for understanding the intricate correlations between lifestyle choices and immunity power. We gather demographic details such as age and gender, which are fundamental in personalizing the predictive model. Additionally, we inquire about daily habits, including sleep duration and meal frequency, as these are critical factors that can significantly impact an individual's health and immunity.

We also focus on dietary preferences, asking detailed questions about the consumption of fruits, vegetables, fast foods, and soft drinks. These dietary habits are essential indicators of overall health and are likely to influence immunity. Further, we recognize the importance of mental health and its impact on physical well-being. Therefore, we include queries about stress levels, as stress can profoundly affect immune function.

Moreover, understanding the importance of hydration, we collect data on water intake, as adequate hydration is crucial for maintaining optimal immune function. We also consider the presence of any pre-existing medical conditions, as these can directly or indirectly impact an individual’s immune system. Lastly, we assess the frequency of doctor visits, which can provide insights into the individual's overall health awareness and medical history.

By collecting and analyzing this wide array of data, we aim to draw meaningful insights and correlations. This holistic approach allows us to build a more accurate and effective predictive model, ultimately contributing to our goal of empowering individuals with knowledge about their health and assisting them in making informed decisions to improve their immunity and overall well-being.

Question and options asked in google form,

1. Your Age

- 20 or below
- 21-30
- 31-40

- 40+

2. Gender
    - Male
    - Female
3. Average Sleeping Hours
    - Less than 6 hours
    - 6 hours
    - More than 6 hours
4. Average Number of Meals per Day
    - Always 3 (breakfast, lunch, dinner)
    - Sometimes skips breakfast
    - Sometimes skips lunch
    - Sometimes skips dinner
5. How Often Do You Consume Fruits and Vegetables in a Week?
    - Everyday
    - Rarely
    - 3-4 days a week
6. How Often Do You Consume Fast Food or Processed Food in a Week?
    - Everyday
    - Rarely
    - 3-4 days a week
7. How Often Do You Consume Soft Drinks in a Week?
    - Everyday
    - Rarely
    - 3-4 days a week
8. Stress Levels
    - Low
    - Moderate
    - High

9. Water Intake in a Day

- Very low
- Low
- Moderate
- High

10. Do You Have Any Existing Medical Conditions?

- No
- Yes

11. On Average, How Frequently Do You Visit a Doctor in a Span of 3 Months?

- 0
- 1-2 times
- 3-4 times
- More than 4 times

Form:

https://docs.google.com/forms/d/1w5gsUMttET9sEH5s4nLwVlQzczwoG5Rziiew7UXmfUY/edit

2.1 Proposed System

Our project, "Health Analysis using ML," follows a systematic and detailed methodology, as illustrated in our flowchart. The process begins with:

1. **Data Collection and Preprocessing**: We start by gathering data through a Google Form, ensuring a comprehensive dataset covering various health and lifestyle parameters. This data is then meticulously preprocessed using Python. The preprocessing involves cleaning the data, handling missing values, and transforming categorical data into a numerical format using Label Encoding. This step is crucial as it prepares the raw data for effective analysis and modeling.
2. **Model Development and Deployment**: The core of our project lies in the development of a predictive model. We use the Random Forest Regressor algorithm due to its efficiency and accuracy in handling complex datasets. Once the model is trained and fine-tuned, we serialize both the trained model and the label encoders into a .pkl file. This serialization is vital for deploying the model in a real-world application, ensuring that it can be easily integrated and used for predictions.
3. **Application and User Interaction**: To make our model accessible to users, we develop a user-friendly interface using the Streamlit framework. This interface allows users to input their personal health and lifestyle data in a simple and intuitive manner. Once the data is inputted, our application processes it using the deployed model. The model predicts the user's immunity power based on the input data, and the predicted score is displayed to the user. This interactive feature is not just about providing a prediction; it also educates users about their health and encourages them to make informed lifestyle choices.
4. **Health Recommendations and Insights**: A unique aspect of our application is its ability to provide personalized health recommendations based on the predicted immunity score. If a user's immunity power is estimated to be low, the app suggests actionable steps like dietary changes, exercise routines, and lifestyle adjustments to help improve their immunity. For users with moderate or high immunity scores, the application offers advice to maintain or enhance their current health status.

5. **Continuous Improvement and Feedback Loop**: We also incorporate a feedback mechanism in our application. Users can provide feedback on the accuracy of predictions and the usefulness of health recommendations. This feedback is invaluable for the continuous improvement of our model and the overall user experience. By analyzing user feedback, we can make necessary adjustments to our model and interface, ensuring that our application remains relevant, accurate, and user-centric.

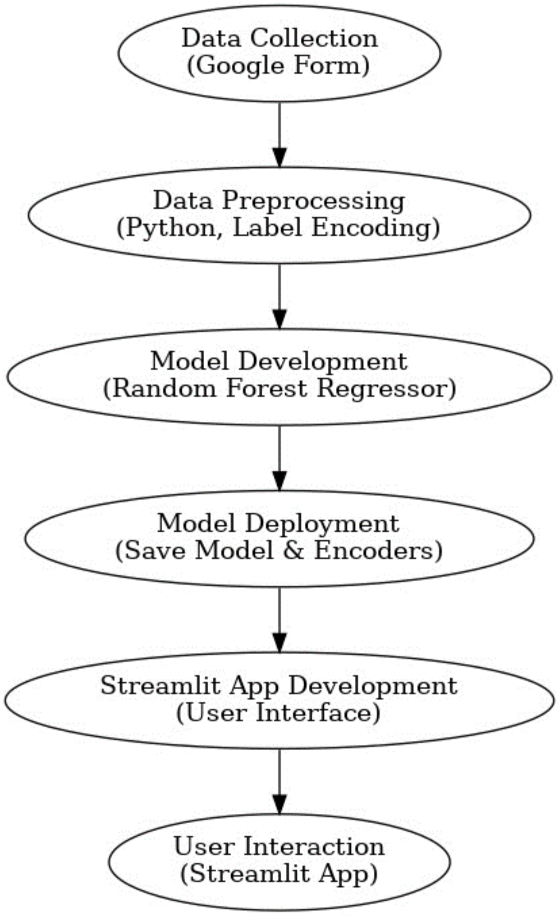


Fig1: flow chart of proposed model

2.2 ML Working methodology:

In our project, we initiated the process by gathering data through Google Forms, which we then meticulously converted into an Excel format. This crucial first step paved the way for the subsequent stages of our project. During the data preprocessing phase, our primary focus was on preparing the data for analysis. This entailed cleaning the data, addressing any missing values, and ensuring that all variables were formatted correctly for effective modeling. This careful preparation of the data was essential to ensure the accuracy and reliability of the models we intended to evaluate. Moving into the analysis phase, we explored various regression models to identify which one would provide the best R-squared (R2) score.

The R2 score is a key indicator in regression analysis, reflecting the model's ability to explain the variability of the data. After a comprehensive evaluation and comparison of different models, we discovered that the Random Forest Regressor outshined the others in terms of R2 score. This led us to select the Random Forest Regressor for our project. Subsequently, we deployed this model into a .pkl (pickle) file. This was a significant step as it allowed us to serialize the model, making it easy to integrate into various applications, such as web services or data analysis tools. This crucial step of transitioning from model development to deployment marked an important milestone in our project, facilitating the practical application of our analytical findings.

Here is the detailded working methodology explanation of ML models,

1. **Library Import and Data Loading**: The code begins by importing necessary libraries for data manipulation, model training, and evaluation. The dataset is then loaded from a CSV file into a pandas DataFrame.
2. **Preprocessing - Label Encoding**: Categorical variables within the dataset are transformed into numeric codes using **LabelEncoder**, allowing for the machine learning models to interpret the data correctly.
3. **Feature-Target Split**: The DataFrame is split into features (**X**) and the target variable (**y**), which in this case is 'Your immunity power'.
4. **Training-Testing Split**: The features and target variable are divided into training and testing sets, using 80% of the data for training and 20% for testing, with a random state set for reproducibility.

5. **Model Initialization**: Multiple regression models including Linear Regression, Random Forest, Gradient Boosting, SVR, and KNN are initialized and stored in a dictionary for subsequent training and evaluation.
6. **Model Evaluation**: A function **evaluate_model** is defined to fit each model on the training data, predict on the testing data, and calculate various performance metrics like R2 score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).
7. **Performance Comparison**: Each model's performance is assessed, and the results are printed out. The Random Forest model is identified to have the highest R2 score, indicating it as the best performing model.
8. **Model Serialization**: The best model (Random Forest) and the label encoders are saved into a dictionary named **immunity_dict**. This dictionary is then serialized into a **.pkl** file using **joblib**, facilitating easy deployment in a production environment.
9. **Interactive Prediction**: A function **get_user_input** is defined to capture real-time user input for all features required by the model. The input is then encoded using the previously fitted label encoders.
10. **User Input Prediction and Display**: The user's encoded input is reshaped and fed into the trained Random Forest model to predict the user's immunity power, which is then displayed as output. This interactive process allows for individualized predictions based on user-provided data.

This working model essentially encapsulates the end-to-end process of training, selecting, and deploying a machine learning model, and it includes a user interface for real-time predictions.

In our project, we initiated the process with data collection via Google Forms, which we meticulously converted into an Excel format for preprocessing. This involved cleaning the data, handling missing values, and ensuring correct formatting for modeling. We evaluated various regression models, selecting the Random Forest Regressor for its outstanding R2 score. The model and label encoders were then serialized into a .pkl file. Our workflow included importing necessary libraries, loading and preprocessing data, splitting it into training and testing sets, and evaluating multiple models. The best-performing model was serialized for deployment, and we integrated an interactive prediction feature in our application, enabling users to receive real-time immunity power predictions based on their data input. This end-to-end methodology encapsulates the essence of our project, blending data science with user-friendly application development.

2.3 Streamlit Working Methodology

In our project, we've developed a sophisticated Streamlit app interface, `app.py`, that acts as the user interface for our machine learning model, which predicts an individual's immunity power. This Python script is meticulously structured to ensure a seamless user experience. Initially, the script initializes by loading our trained model and its corresponding label encoders from a pickle file named 'Immunity_dict.pkl'. These label encoders are crucial as they transform categorical text data into a numerical format that our model can interpret. Additionally, we load our dataset from a CSV file 'HA.csv', which provides the necessary data for the dropdown options in the Streamlit interface.

The Streamlit interface is set up with a welcoming title, 'Predict Your Immunity Power', and an engaging image 'immu.png' displayed prominently. Users interact with a series of dropdown menus, each populated with specific data from the dataset, allowing them to input their details like age group, gender, sleeping hours, and more.

Upon receiving user inputs, a helper function `encode_input` encodes these inputs using the relevant label encoder for each attribute. When the user hits the 'Predict' button, these inputs are encoded and prepared for the model prediction.

The core of the app lies in its ability to predict and communicate the user's immunity power. The script feeds the encoded inputs into the model, which then calculates an immunity score. This score is not only presented to the user in a clear, understandable format but is also accompanied by personalized health suggestions. These suggestions vary based on the immunity score, ranging from dietary advice and lifestyle changes for lower scores to encouragement for maintaining good health practices for higher scores. This tailored approach in our app makes it a powerful tool for individuals to understand and potentially enhance their immunity power.

The workflow of the script is as follows:

1. **Initialization**:
   - The trained model and its corresponding label encoders are loaded from a pickle file named 'Immunity_dict.pkl'. Label encoders are used to convert categorical text data into a model-understandable numeric form.
   - The dataset used for the dropdown options in the Streamlit interface is loaded from a CSV file named 'HA.csv'(dataset csv file).
2. **Streamlit Interface Setup**:
   - The title of the app is set to 'Predict Your Immunity Power', and an image with the name 'immu.png' is displayed at the top of the app using the full column width.
   - A series of dropdown menus are created using **st.selectbox**, populated with unique values from each relevant column in the dataset. These dropdowns capture user input for various attributes such as age group, gender, sleeping hours, etc.
3. **User Input Encoding**:
   - A helper function **encode_input** is defined to encode the user's input using the appropriate label encoder for each attribute.
   - When the 'Predict' button is pressed, the script encodes the inputs from all dropdowns, preparing them for model prediction.
4. **Prediction and Output**:
   - The encoded inputs are fed into the model to predict the user's immunity power.
   - The predicted immunity score is rounded and displayed to the user.
   - Depending on the immunity score, the app provides tailored health suggestions to help the user potentially improve their immunity power. For example, if the prediction is below 5, it suggests a diet rich in fruits, regular exercise, and adequate sleep, among others. For moderate scores, it suggests staying hydrated and considering probiotics, while for good scores, it encourages maintaining a healthy lifestyle.

Overall, the **app.py** streamlit interface serves as a practical tool allowing users to get an estimate of their immunity power based on their lifestyle choices and provides personalized recommendations for health improvement. Streamlit is an open-source app framework written in Python, designed to turn data scripts into shareable web apps with minimal effort. It allows for rapid prototyping of machine learning models and data dashboards by providing a simple and intuitive API that requires no prior experience with web development. Streamlit's interactive widgets enable users to manipulate their data and view the results in real-time, making it a popular choice for data scientists and engineers looking to quickly and efficiently showcase their work.

CHAPTER 3

Results and discussion

In Machine learning model building, as my dependent column has continues values, I used regression models for prediction. I have used Linear Regression, Random Forest, Gradient Boosting, SVR, KNN. I got good R2_score in Random Forest, so I have used it in further process. Here is the details about results of models,

Linear Regression –

R2 Score: 0.27502320468106534, MAE: 1.1634025857344832, MSE: 2.3994959686625443, RMSE: 1.5490306545264185

Random Forest –

R2 Score: 0.7430632381603692, MAE: 0.5227659574468085, MSE: 0.8503978723404255, RMSE: 0.9221701970571514

Gradient Boosting –

R2 Score: 0.4976236635504593, MAE: 0.8910721476401893, MSE: 1.6627428654896805, RMSE: 1.28947387158084

SVR –

R2 Score: 0.3250448608007067, MAE: 0.9493627731776714, MSE: 2.2339365149256825, RMSE: 1.4946359138350993

KNN –

R2 Score: 0.3103559582834673, MAE: 1.1148936170212767, MSE: 2.2825531914893613, RMSE: 1.5108120966848795

The best performing model is: Random Forest with R2 score: 0.7430632381603692

In the process of building our machine learning model, we meticulously evaluated several regression algorithms to predict continuous values effectively. Our dependent variable required a model capable of handling continuous data, leading us to test various regression models including Linear Regression, Random Forest, Gradient Boosting, Support Vector Regression (SVR), and K-Nearest Neighbors (KNN). Each model was assessed based on its R2 score, Mean Absolute Error

(MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to determine its predictive accuracy and reliability.

Our results were quite revealing. Linear Regression, often considered a baseline in regression analysis, showed an R2 score of 0.275, indicating a relatively low predictive capability. Similarly, SVR and KNN produced R2 scores of 0.325 and 0.310 respectively, suggesting that these models were not optimal for our dataset. Gradient Boosting performed better with an R2 score of 0.498, but it still fell short of our expectations.

However, the Random Forest Regressor stood out with its significantly higher R2 score of 0.743, indicating a robust ability to predict our target variable accurately. Its MAE of 0.523, MSE of 0.850, and RMSE of 0.922 further cemented its superiority over the other models. These metrics collectively demonstrated Random Forest's effectiveness in handling the complexity of our dataset, providing a balance between bias and variance, and capturing the underlying patterns more accurately than its counterparts.

Consequently, we chose the Random Forest Regressor for further development and deployment in our project. Its high R2 score not only represented its excellent predictive power but also ensured that the model could be reliably used in practical applications, such as in our Streamlit-based application for predicting individual immunity power. This decision was pivotal in ensuring that our project delivered accurate, reliable, and valuable insights to its users, thereby significantly contributing to the field of personal health analytics.

Why randomforest

The Random Forest Regressor was selected as our model of choice due to its exceptional performance metrics, particularly the R2 score of 0.743, which far surpassed those of other tested models. This high R2 score indicates a strong correlation between the predicted values and actual values, reflecting the model’s ability to capture the variance of the dependent variable accurately. Additionally, the Random Forest algorithm is less prone to overfitting compared to other models due to its ensemble approach, which builds multiple decision trees and aggregates their results. Its inherent ability to handle non-linear relationships and interactions between variables makes it robust for a diverse set of data. Moreover, the lower MAE and RMSE values suggest better prediction accuracy with less deviation from the actual values, reinforcing our confidence in its predictive reliability. These factors collectively make the Random Forest Regressor a solid and trustworthy choice for deployment in predicting continuous outcomes like immunity power.

ML code output Explanation: The code provided is a series of Python instructions using several key libraries for machine learning model training and evaluation. Initially, `pandas` is used to load a dataset from a CSV file, which is the starting point for any data-driven model. The dataset is then prepared by encoding categorical variables using `LabelEncoder` from the `sklearn.preprocessing` module, turning text values into a numeric format that machine learning algorithms can process.

After encoding, the dataset is split into features (`X`) and the target (`y`), followed by a further split into training and testing sets using `train_test_split`. This is critical for training the model on one subset of data and validating it on another to prevent overfitting and ensure the model's generalizability.

The code then initializes several regression models from the `sklearn` library, including Linear Regression, Random Forest, Gradient Boosting, SVR, and KNN. Each model is trained and evaluated using metrics like R2 score, MAE, MSE, and RMSE to determine its performance. Among these, Random Forest Regressor is identified as the best performing model based on the R2 score.

Finally, the best model and the label encoders are saved into a dictionary and serialized into a `.pkl` file with `joblib`. This allows the model to be easily loaded later for making predictions or deploying in an application, as demonstrated in the Streamlit code for the user interface. The process encapsulates the end-to-end workflow of a machine learning task, from preprocessing to model selection, evaluation, and deployment.

3.1 Visual results:

This are some of our visual finding from the data we have collected from the users through google forms.

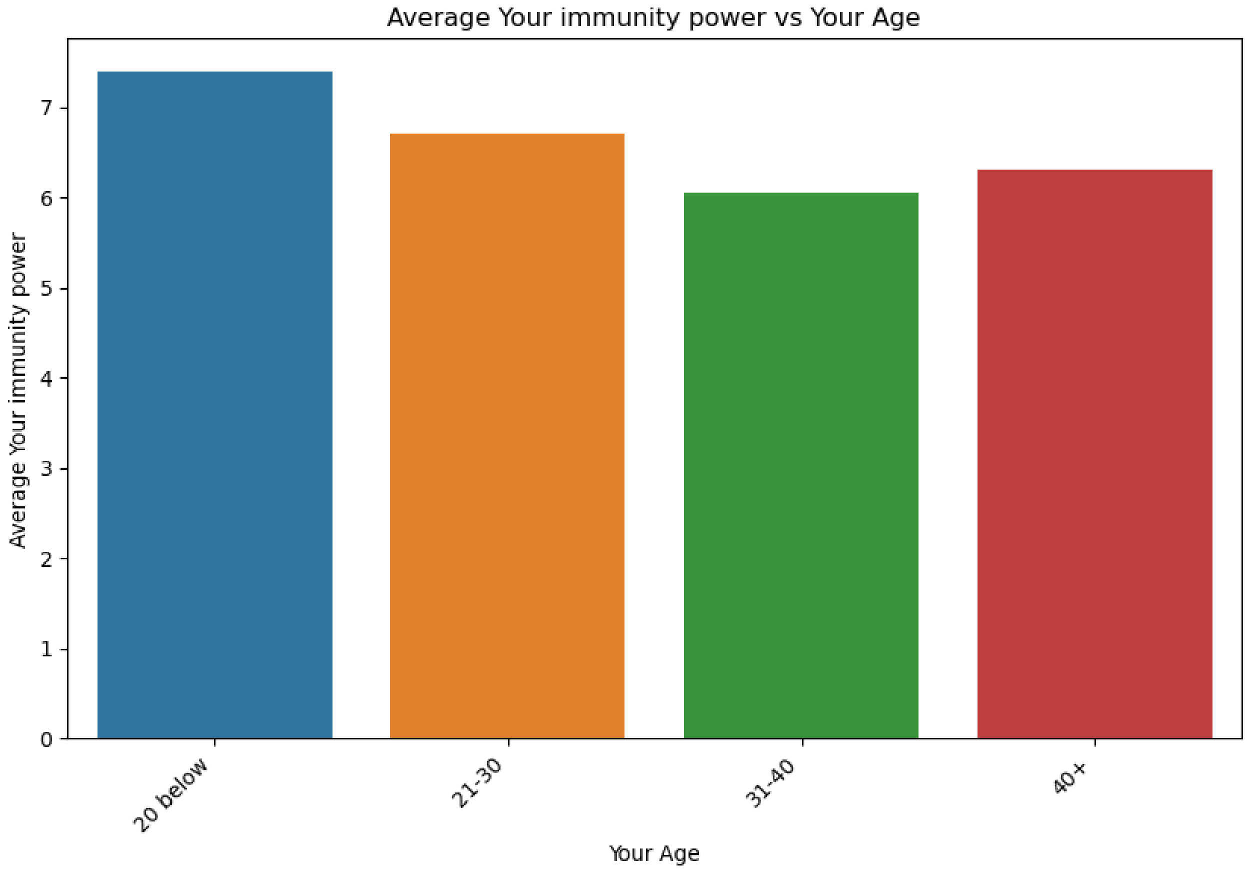


Figure 2 (Your Age vs. Immunity Power): This bar graph indicates that younger individuals (20 and below) report a higher average immunity power, which slightly decreases as the age group increases.

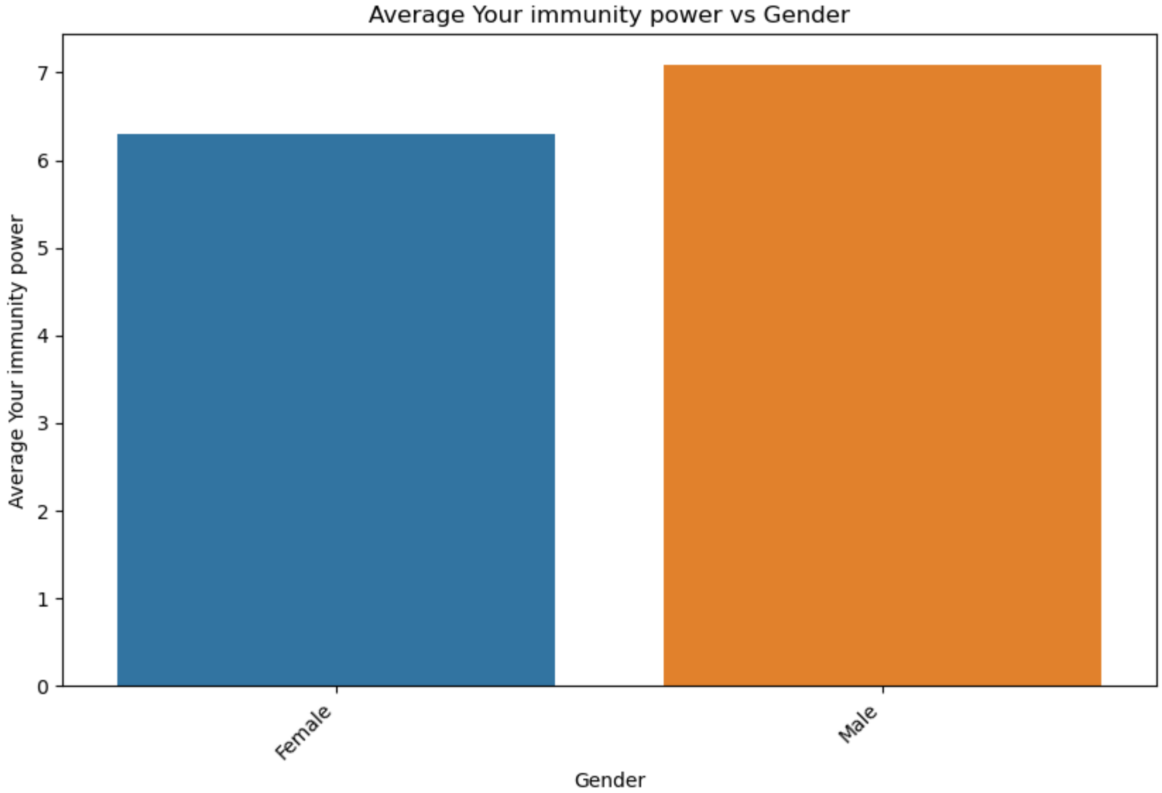


Figure 3 (Gender vs. Immunity Power): The bar chart shows a distinction between genders, with males reporting a slightly higher average immunity power compared to females, suggesting a potential gender-related pattern in immunity self-assessment.

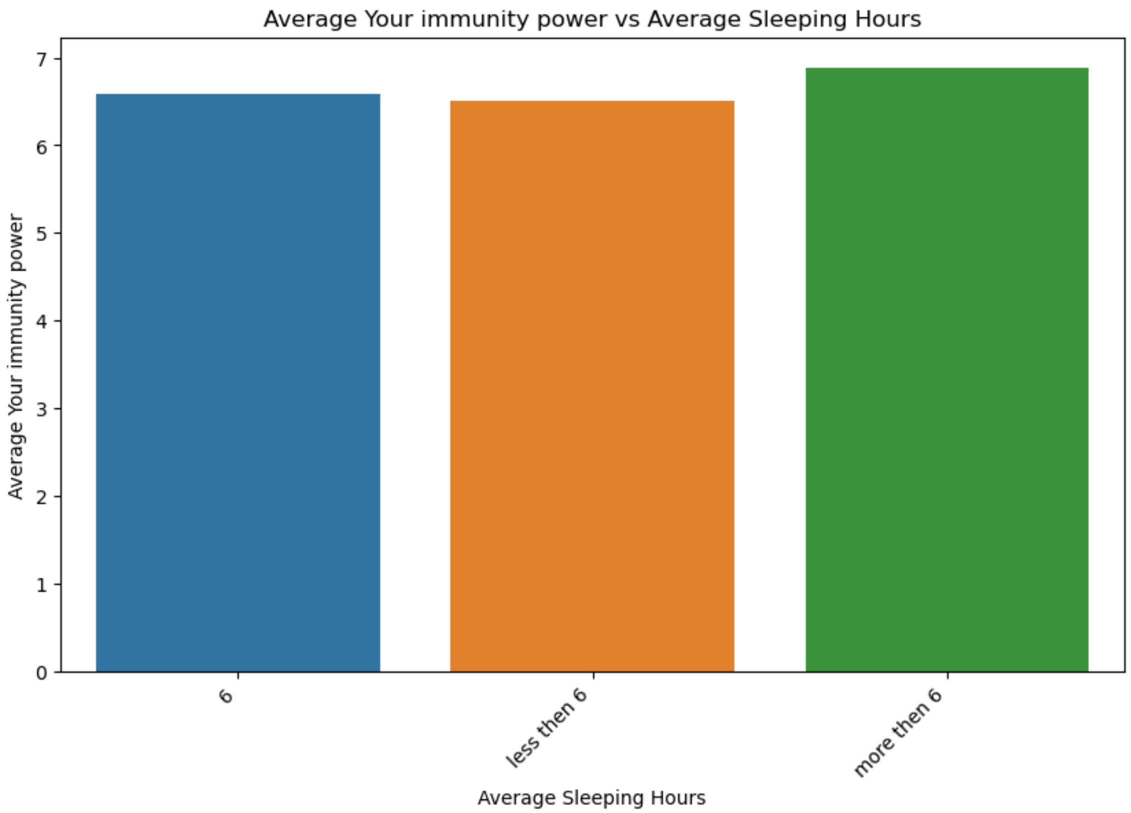


Figure 4 (Average Sleeping Hours vs. Immunity Power): This visualization suggests that individuals reporting '6' hours of sleep have a slightly lower average immunity power than those who sleep 'less than 6' or 'more than 6' hours, highlighting the non-linear relationship between sleep and perceived immunity strength.

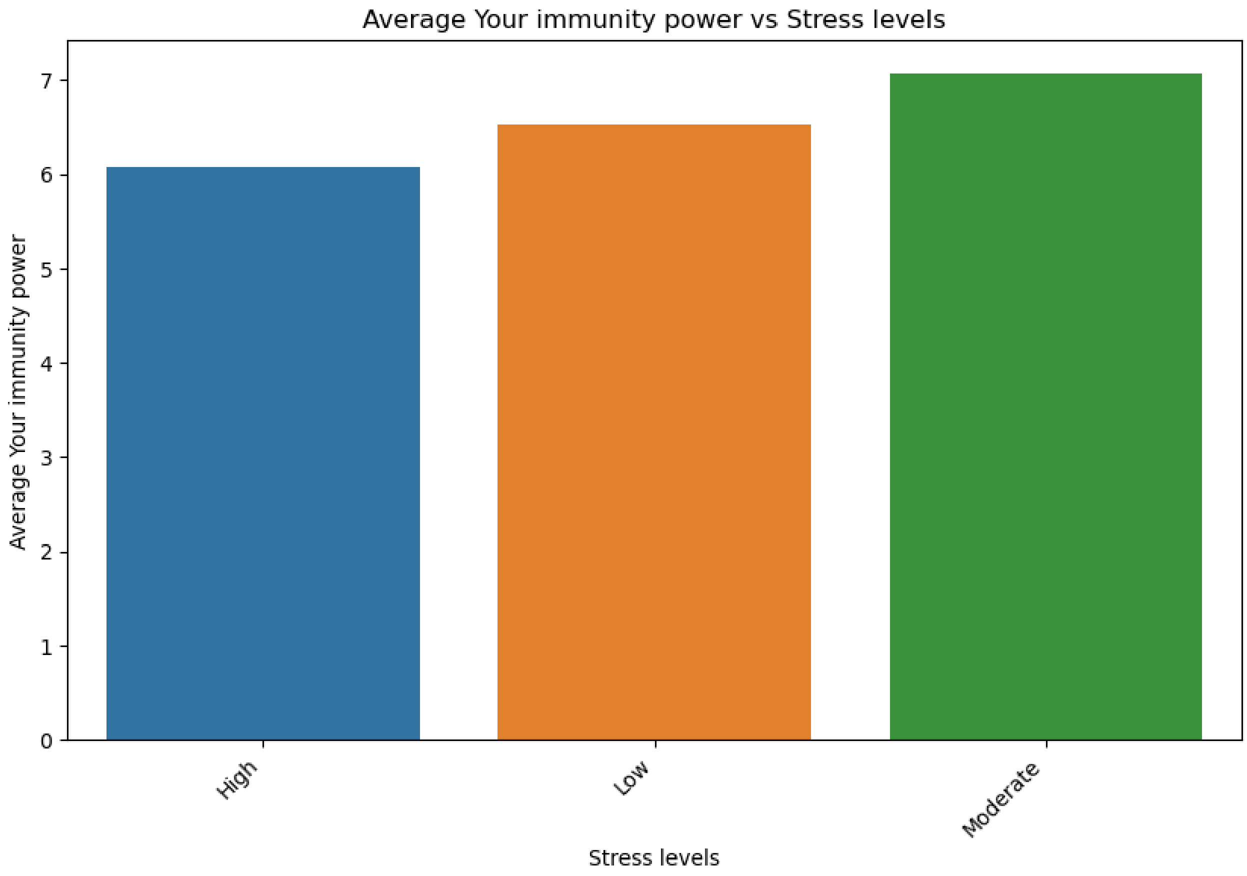


Figure 5 (Stress Levels vs. Immunity Power): The chart presents an interesting trend where individuals with 'Low' stress levels report higher average immunity power, followed by 'Moderate' and 'High' stress levels, implying that lower stress may be associated with better self-perceived immunity.

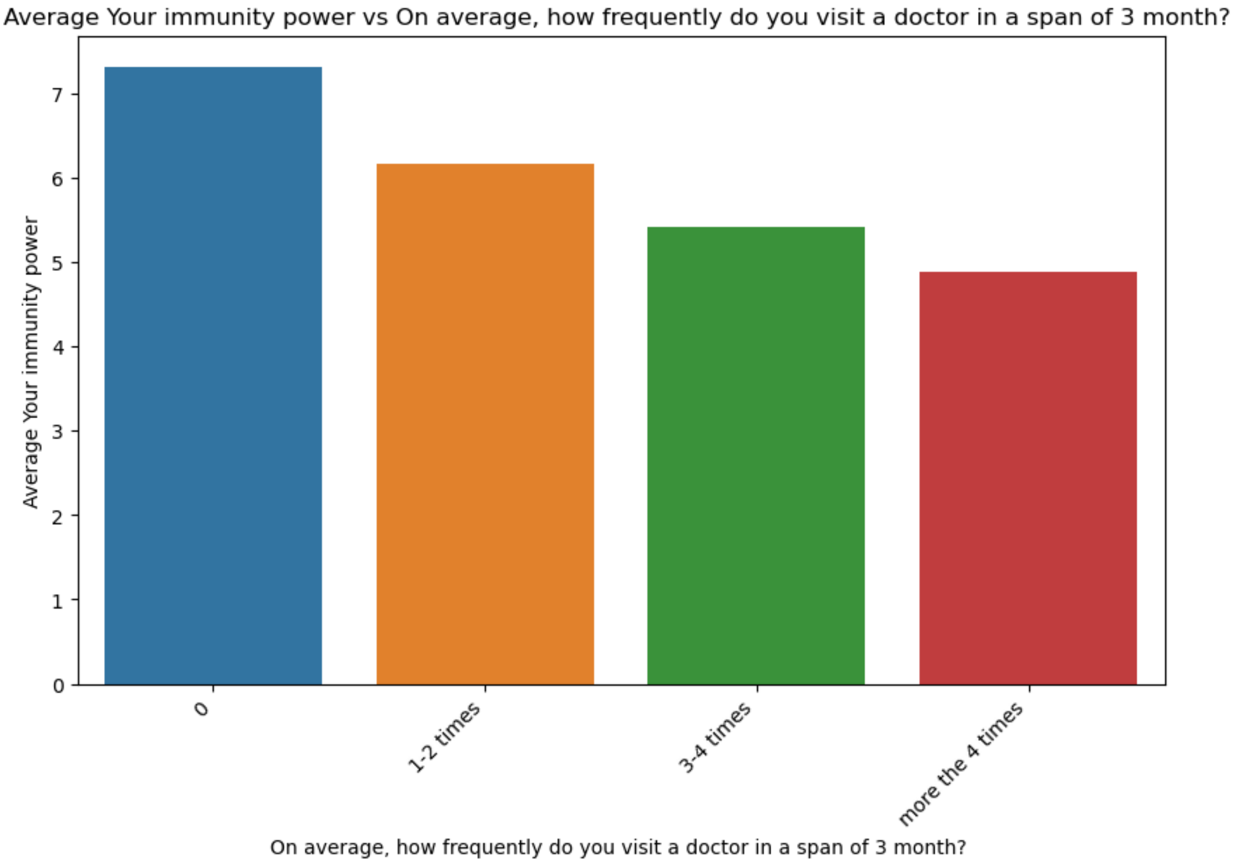


Figure 6 (Frequency of Doctor Visits vs. Immunity Power): The final graph suggests that people who visit the doctor '0' times in 3 months perceive their immunity power to be the highest, with a gradual decrease as the frequency of visits increases, which may reflect a general trend in self-perceived health status and its impact on healthcare utilization.

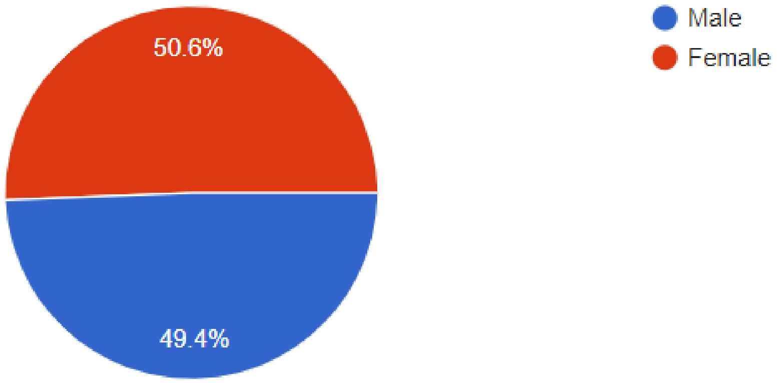


Figure 7 : percentage of gender responses

The pie chart indicates a well-balanced distribution of responses from male and female participants in your survey, with 50.6% male and 49.4% female respondents. This near-equal split demonstrates a conscientious effort to ensure gender representation in your data collection, which can provide a more nuanced understanding of how different factors may affect immunity across genders. This balanced approach enhances the reliability of your findings and supports the development of a robust machine learning model.

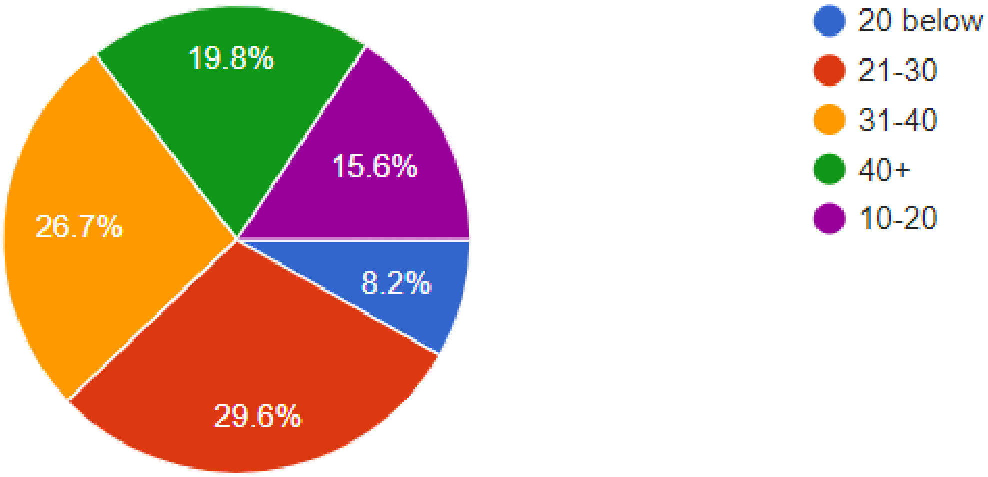


Figure 8: Age groups responded to the google form

The pie chart represents the age distribution of respondents who participated in your Google Form survey. It shows a diverse range of age groups, which is beneficial for the analysis. The largest segment of responses comes from individuals aged 21-30, comprising 29.6% of the total. This is followed by the 31-40 age group at 26.7%, and the 20 and below category at 19.8%. Those aged 40 and above account for 15.6% of responses, while the 10-20 age group represents 8.2%. This variety in age demographics allows for a broader understanding of immunity across different life stages, enhancing the overall data quality for your machine learning model.

CHAPTER 4
Conclusion and Future work:

In conclusion, my project on predicting individual immunity power has successfully leveraged a range of lifestyle and health-related data to create a model that provides personalized insights. The evaluation of various machine learning models led me to select the Random Forest algorithm due to its superior performance across multiple metrics. The Streamlit application I developed offers an intuitive interface for users to interact with the model, contributing to an accessible and user-friendly experience.

Building on the success of the immunity power prediction project, there's substantial potential for advancement. By integrating a wider array of predictive variables, such as genetic markers and a detailed history of immunizations, the model could offer even more personalized and precise predictions. Additionally, the incorporation of machine learning interpretability tools can provide users with clear explanations of how their data influences their immunity score, increasing trust and transparency in the model's predictions.

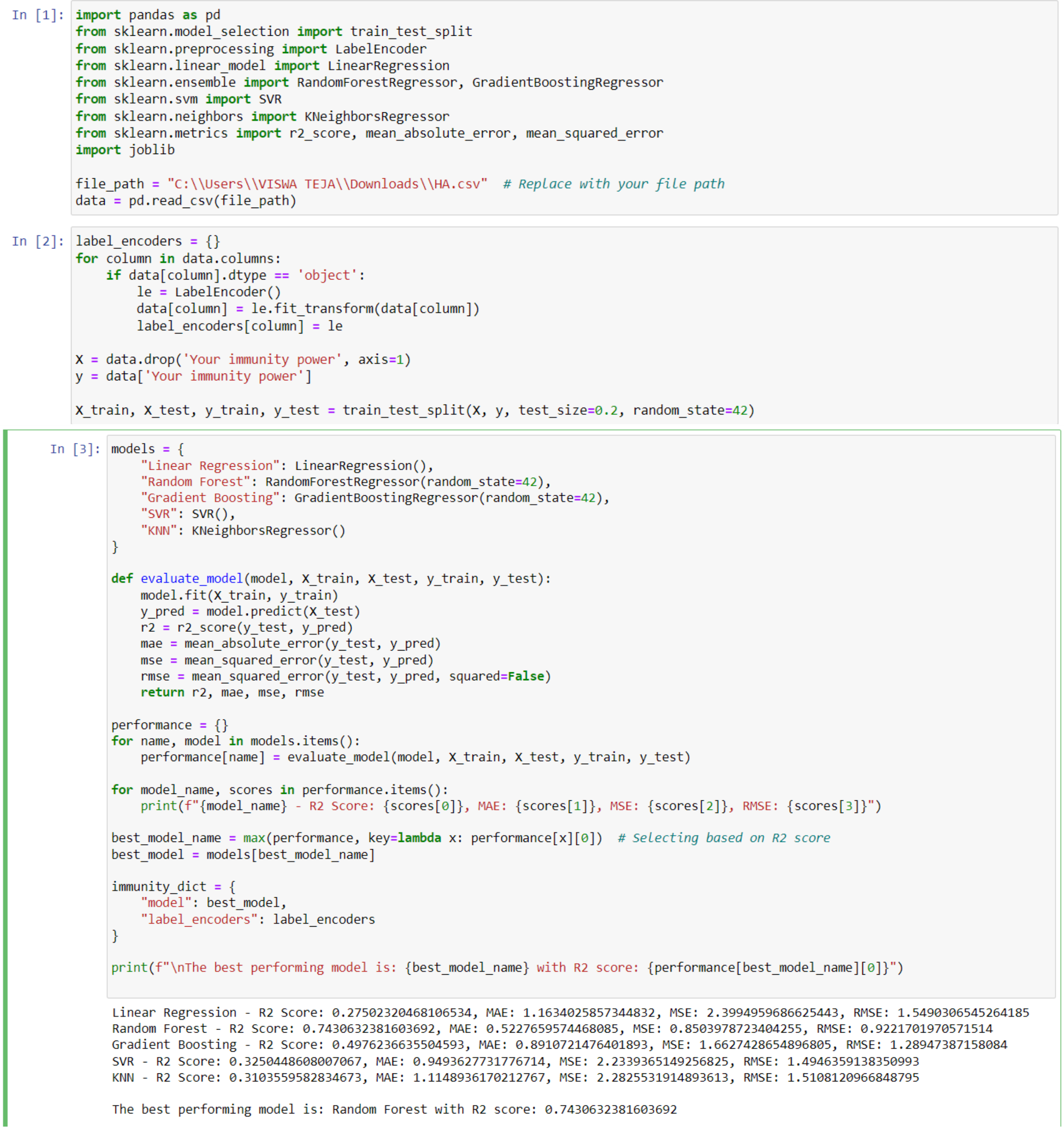
Collaborations with healthcare providers and epidemiologists could also enhance the model's capabilities. Such partnerships may allow access to anonymized clinical data, enabling the model to learn from a broader set of health indicators and outcomes. Furthermore, user feedback loops can be instrumental in refining the model. By allowing users to report back on the accuracy of their predictions and the effectiveness of recommended actions, the system can be fine-tuned for greater accuracy.

A mobile app could also be developed, leveraging the convenience of smartphones to reach a wider audience. With such an app, users could receive regular updates on their immunity status and timely health advice. This could also be paired with a community feature, where users can share their experiences and tips for boosting immunity, fostering a supportive environment focused on collective health improvement.

As personal and public health increasingly converge with technology, this project stands as a testament to the potential of AI in empowering individuals with the knowledge to lead healthier lives. By continuously iterating and enhancing the model with the latest technological and medical insights, the project can remain at the forefront of preventative health care innovation.

CHAPTER 5
CODE & OUTPUT

ML code to choose best model:



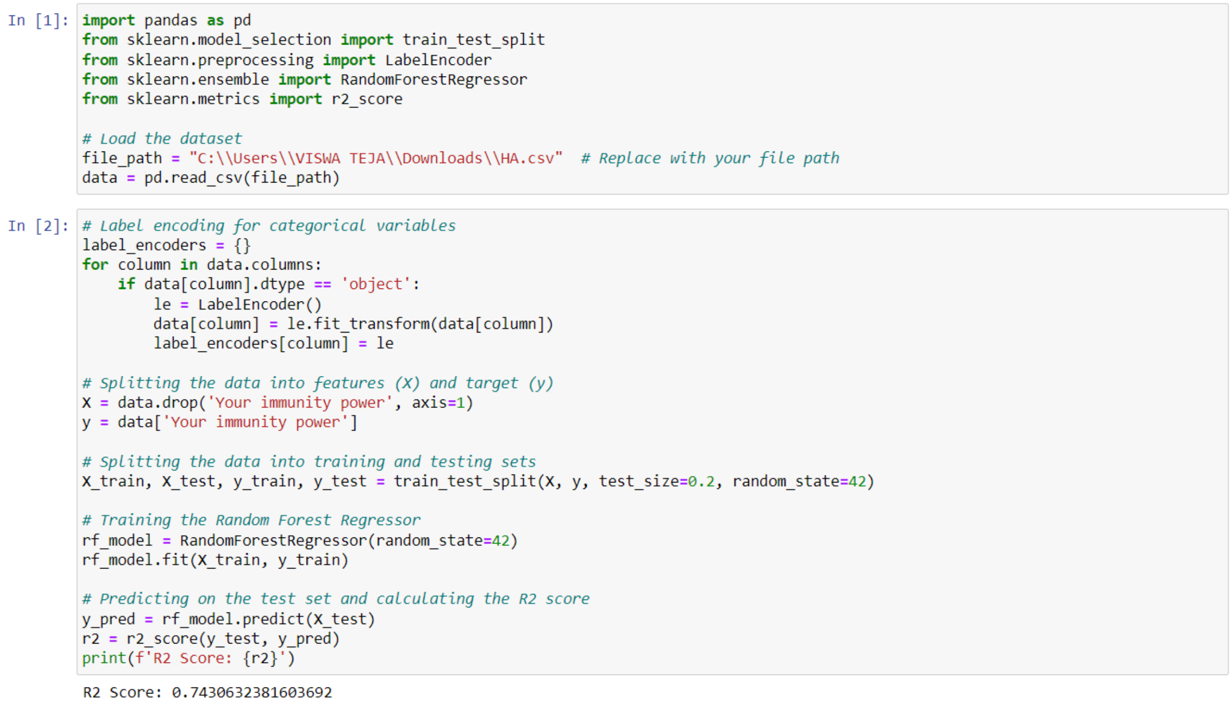
```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        from sklearn.linear_model import LinearRegression
        from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
        from sklearn.svm import SVR
        from sklearn.neighbors import KNeighborsRegressor
        from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
        import joblib

        file_path = "C:\\Users\\VISWA TEJA\\Downloads\\HA.csv"  # Replace with your file path
        data = pd.read_csv(file_path)
```

```
In [2]: label_encoders = {}
        for column in data.columns:
            if data[column].dtype == 'object':
                le = LabelEncoder()
                data[column] = le.fit_transform(data[column])
                label_encoders[column] = le

        X = data.drop('Your immunity power', axis=1)
        y = data['Your immunity power']

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [3]: models = {
            "Linear Regression": LinearRegression(),
            "Random Forest": RandomForestRegressor(random_state=42),
            "Gradient Boosting": GradientBoostingRegressor(random_state=42),
            "SVR": SVR(),
            "KNN": KNeighborsRegressor()
        }

        def evaluate_model(model, X_train, X_test, y_train, y_test):
            model.fit(X_train, y_train)
            y_pred = model.predict(X_test)
            r2 = r2_score(y_test, y_pred)
            mae = mean_absolute_error(y_test, y_pred)
            mse = mean_squared_error(y_test, y_pred)
            rmse = mean_squared_error(y_test, y_pred, squared=False)
            return r2, mae, mse, rmse

        performance = {}
        for name, model in models.items():
            performance[name] = evaluate_model(model, X_train, X_test, y_train, y_test)

        for model_name, scores in performance.items():
            print(f"{model_name} - R2 Score: {scores[0]}, MAE: {scores[1]}, MSE: {scores[2]}, RMSE: {scores[3]}")

        best_model_name = max(performance, key=lambda x: performance[x][0])  # Selecting based on R2 score
        best_model = models[best_model_name]

        immunity_dict = {
            "model": best_model,
            "label_encoders": label_encoders
        }

        print(f"\nThe best performing model is: {best_model_name} with R2 score: {performance[best_model_name][0]}")
```

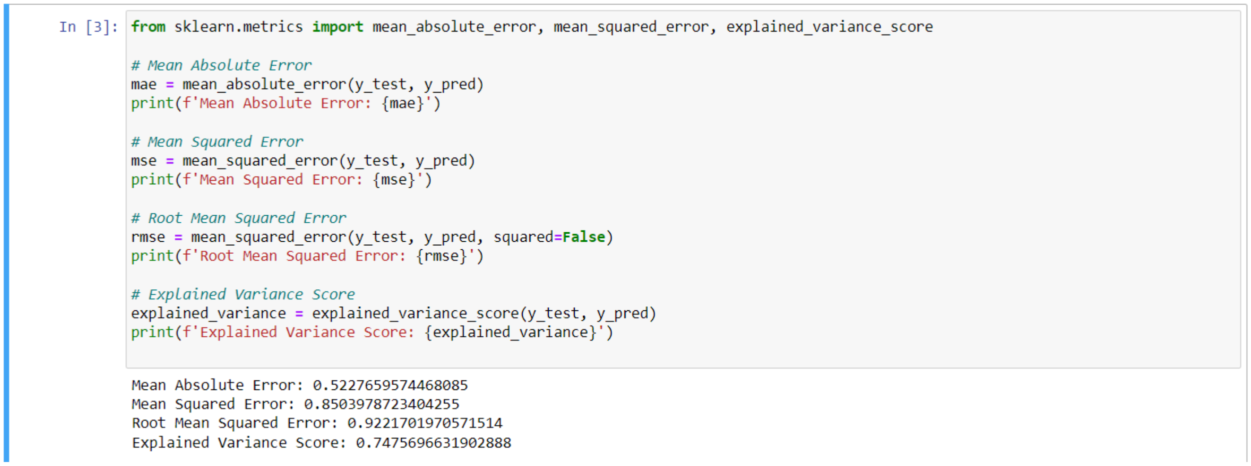
```
Linear Regression - R2 Score: 0.27502320468106534, MAE: 1.1634025857344832, MSE: 2.3994959686625443, RMSE: 1.5490306545264185
Random Forest - R2 Score: 0.7430632381603692, MAE: 0.5227659574468085, MSE: 0.8503978723404255, RMSE: 0.9221701970571514
Gradient Boosting - R2 Score: 0.4976236635504593, MAE: 0.8910721476401893, MSE: 1.6627428654896805, RMSE: 1.28947387158084
SVR - R2 Score: 0.3250448608007067, MAE: 0.9493627731776714, MSE: 2.2339365149256825, RMSE: 1.4946359138350993
KNN - R2 Score: 0.3103559582834673, MAE: 1.1148936170212767, MSE: 2.2825531914893613, RMSE: 1.5108120966848795

The best performing model is: Random Forest with R2 score: 0.7430632381603692
```

ML code to save best model to pkl file:

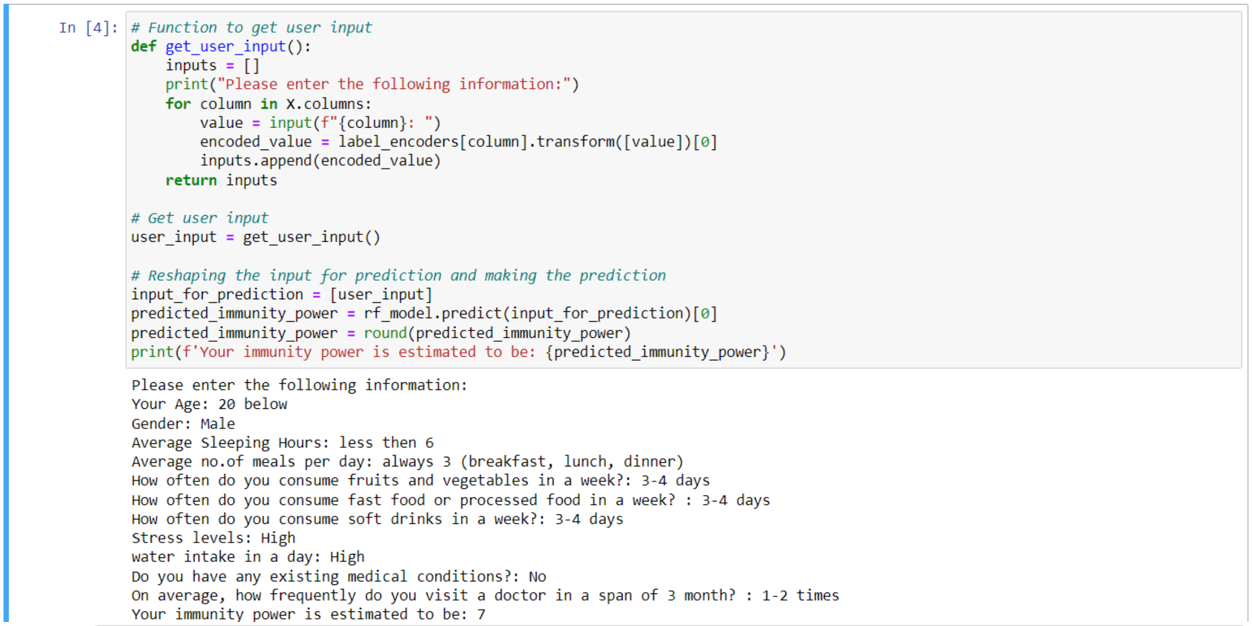


```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        from sklearn.ensemble import RandomForestRegressor
        from sklearn.metrics import r2_score

        # Load the dataset
        file_path = "C:\\Users\\VISWA TEJA\\Downloads\\HA.csv"  # Replace with your file path
        data = pd.read_csv(file_path)
```

```
In [2]: # Label encoding for categorical variables
        label_encoders = {}
        for column in data.columns:
            if data[column].dtype == 'object':
                le = LabelEncoder()
                data[column] = le.fit_transform(data[column])
                label_encoders[column] = le

        # Splitting the data into features (X) and target (y)
        X = data.drop('Your immunity power', axis=1)
        y = data['Your immunity power']

        # Splitting the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

        # Training the Random Forest Regressor
        rf_model = RandomForestRegressor(random_state=42)
        rf_model.fit(X_train, y_train)

        # Predicting on the test set and calculating the R2 score
        y_pred = rf_model.predict(X_test)
        r2 = r2_score(y_test, y_pred)
        print(f'R2 Score: {r2}')

        R2 Score: 0.7430632381603692
```

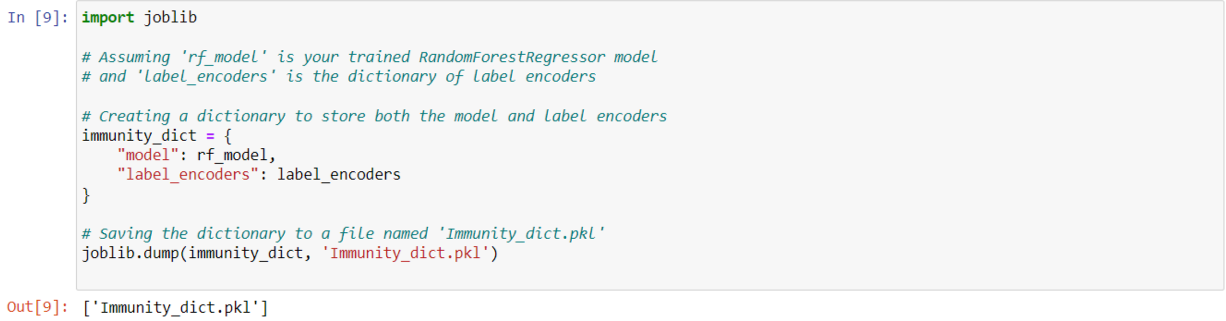


```
In [3]: from sklearn.metrics import mean_absolute_error, mean_squared_error, explained_variance_score

        # Mean Absolute Error
        mae = mean_absolute_error(y_test, y_pred)
        print(f'Mean Absolute Error: {mae}')

        # Mean Squared Error
        mse = mean_squared_error(y_test, y_pred)
        print(f'Mean Squared Error: {mse}')

        # Root Mean Squared Error
        rmse = mean_squared_error(y_test, y_pred, squared=False)
        print(f'Root Mean Squared Error: {rmse}')

        # Explained Variance Score
        explained_variance = explained_variance_score(y_test, y_pred)
        print(f'Explained Variance Score: {explained_variance}')

        Mean Absolute Error: 0.5227659574468085
        Mean Squared Error: 0.8503978723404255
        Root Mean Squared Error: 0.9221701970571514
        Explained Variance Score: 0.7475696631902888
```



```
In [4]: # Function to get user input
        def get_user_input():
            inputs = []
            print("Please enter the following information:")
            for column in X.columns:
                value = input(f"{column}: ")
                encoded_value = label_encoders[column].transform([value])[0]
                inputs.append(encoded_value)
            return inputs

        # Get user input
        user_input = get_user_input()

        # Reshaping the input for prediction and making the prediction
        input_for_prediction = [user_input]
        predicted_immunity_power = rf_model.predict(input_for_prediction)[0]
        predicted_immunity_power = round(predicted_immunity_power)
        print(f'Your immunity power is estimated to be: {predicted_immunity_power}')
```

```
Please enter the following information:
Your Age: 20 below
Gender: Male
Average Sleeping Hours: less then 6
Average no.of meals per day: always 3 (breakfast, lunch, dinner)
How often do you consume fruits and vegetables in a week?: 3-4 days
How often do you consume fast food or processed food in a week? : 3-4 days
How often do you consume soft drinks in a week?: 3-4 days
Stress levels: High
water intake in a day: High
Do you have any existing medical conditions?: No
On average, how frequently do you visit a doctor in a span of 3 month? : 1-2 times
Your immunity power is estimated to be: 7
```



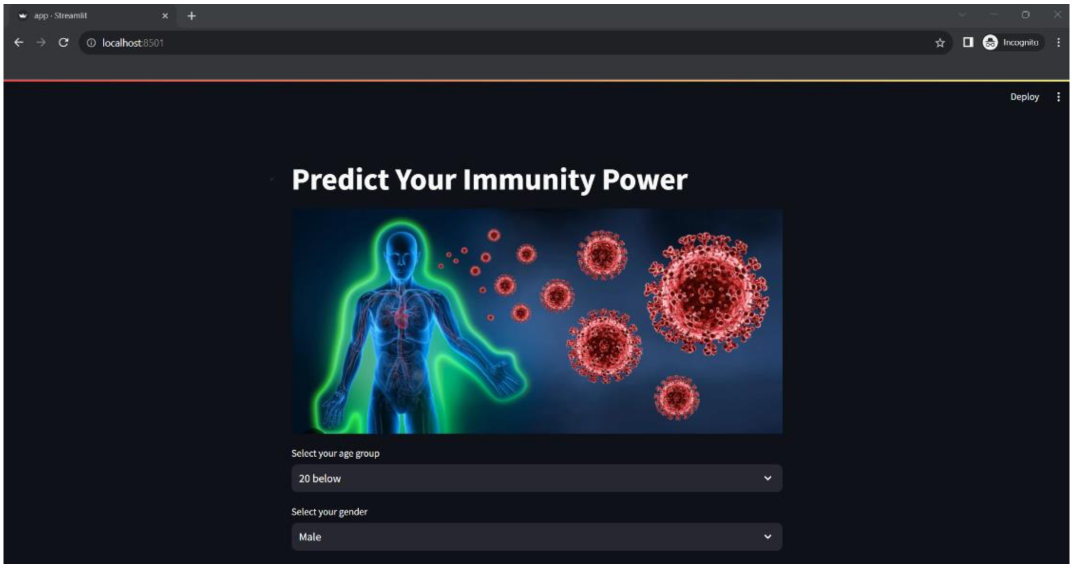
```
In [9]: import joblib

        # Assuming 'rf_model' is your trained RandomForestRegressor model
        # and 'label_encoders' is the dictionary of label encoders

        # Creating a dictionary to store both the model and label encoders
        immunity_dict = {
            "model": rf_model,
            "label_encoders": label_encoders
        }

        # Saving the dictionary to a file named 'Immunity_dict.pkl'
        joblib.dump(immunity_dict, 'Immunity_dict.pkl')

Out[9]: ['Immunity_dict.pkl']
```

Streamlit code and output:

App.py code:

```
import streamlit as st
import joblib
import pandas as pd

model_data = joblib.load(open('Immunity_dict.pkl', 'rb'))
model = model_data['model']
label_encoders = model_data['label_encoders']

ha = pd.read_csv('HA.csv')

st.title('Predict Your Immunity Power')
st.image('immu.png', use_column_width=True)


def encode_input(input, column_name):
    encoder = label_encoders[column_name]
    return encoder.transform([input])[0]


age = st.selectbox("Select your age group", ha['Your Age'].unique())
gender = st.selectbox("Select your gender", ha['Gender'].unique())
sleep = st.selectbox("Average sleeping hours per day", ha['Average Sleeping
Hours'].unique())
meals = st.selectbox("Average no. of meals per day", ha['Average no.of meals per
day'].unique())
fv = st.selectbox("Your average fruits and vegetables consumption in a week?",
             ha['How often do you consume fruits and vegetables in a
week?'].unique())
ff = st.selectbox("Fast foods consumption in a week?",
             ha['How often do you consume fast food or processed food in a week?
'].unique())
sd = st.selectbox("Soft drinks consumption in a week?", ha['How often do you
consume soft drinks in a week?'].unique())
stress = st.selectbox("Your Stress levels", ha['Stress levels'].unique())
water = st.selectbox("Your water intake in a day", ha['water intake in a
day'].unique())
medical = st.selectbox("Do you have any existing medical conditions",
                       ha['Do you have any existing medical conditions?'].unique())
doctor = st.selectbox("On average, how frequently do you visit a doctor in a span of
3 month?",
                      ha['On average, how frequently do you visit a doctor in a span of 3
month? '].unique())

if st.button('Predict'):
    # Encode user inputs
    encoded_inputs = [
        encode_input(age, 'Your Age'),
        encode_input(gender, 'Gender'),
        encode_input(sleep, 'Average Sleeping Hours'),
        encode_input(meals, 'Average no.of meals per day'),
        encode_input(fv, 'How often do you consume fruits and vegetables in a week?'),
        encode_input(ff, 'How often do you consume fast food or processed food in a
week? '),
        encode_input(sd, 'How often do you consume soft drinks in a week?'),
        encode_input(stress, 'Stress levels'),
        encode_input(water, 'water intake in a day'),
        encode_input(medical, 'Do you have any existing medical conditions?'),
        encode_input(doctor, 'On average, how frequently do you visit a doctor in a span
of 3 month? ')
    ]

    # Make prediction
    prediction = model.predict([encoded_inputs])[0]
    prediction = round(prediction)

    # Display the prediction
    st.write(f'Your estimated immunity power is: {prediction}')

    if prediction < 5:
        st.write("*** suggestions for you *** ")
        st.write("Your immunity power is low. Here are some suggestions to improve
it:")
        st.write("- Eat a diet high in fruits, vegetables, and whole grains.")
        st.write("- Exercise regularly.")
        st.write("- Maintain a healthy weight.")
        st.write("- Get adequate sleep.")
        st.write("- Try to minimize stress.")
    elif prediction < 8:
        st.write("*** suggestions for you *** ")
        st.write("Your immunity power is moderate. Here are some tips to enhance it:")
        st.write("- Include more vitamin C rich foods in your diet like citrus fruits.")
        st.write("- Stay hydrated and drink plenty of water.")
        st.write("- Consider probiotics.")
        st.write("- Get regular medical screening tests for people in your age group and risk category.")

    else:
        st.write("Your immunity power is good. Keep maintaining your healthy lifestyle!")
```

Output webpage screenshots :



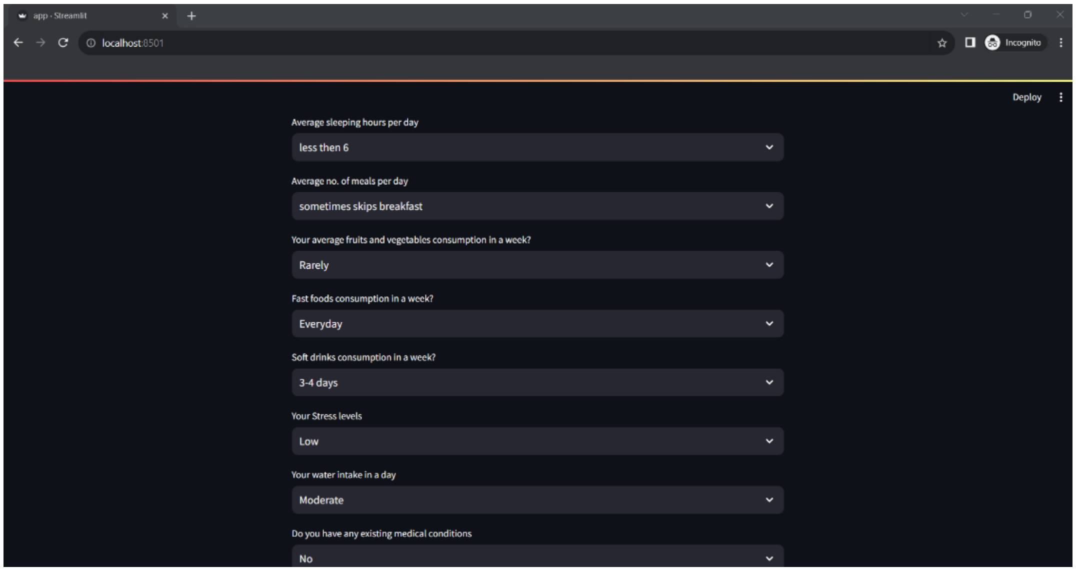
Predict Your Immunity Power

Select your age group

20 below

Select your gender

Male



Average sleeping hours per day

less then 6

Average no. of meals per day

sometimes skips breakfast

Your average fruits and vegetables consumption in a week?

Rarely

Fast foods consumption in a week?

Everyday

Soft drinks consumption in a week?

3-4 days

Your Stress levels

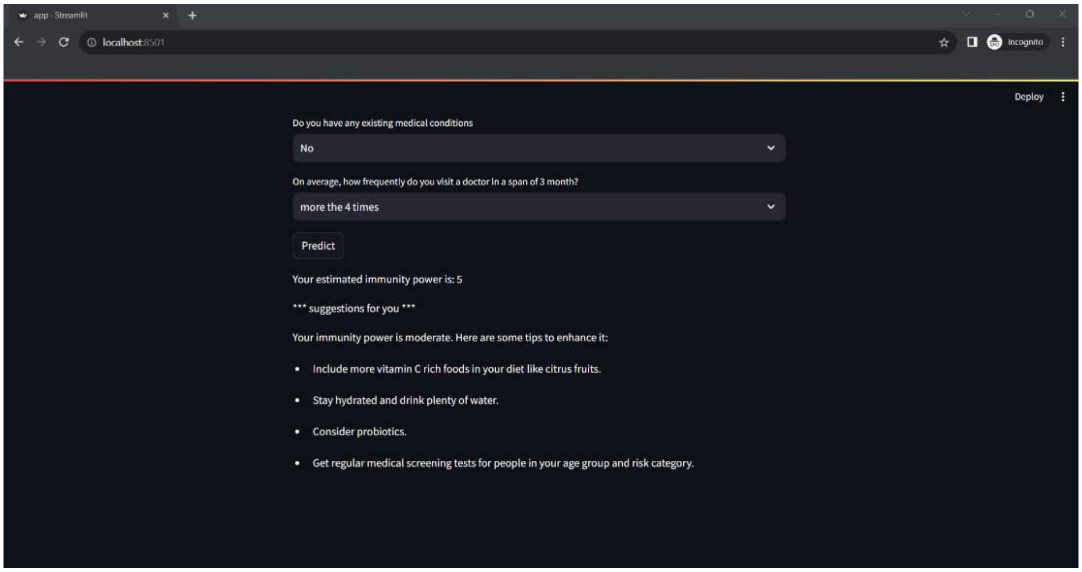
Low

Your water intake in a day

Moderate

Do you have any existing medical conditions

No



Do you have any existing medical conditions

No

On average, how frequently do you visit a doctor in a span of 3 month?

more the 4 times

Your estimated immunity power is: 5

*** suggestions for you ***

Your immunity power is moderate. Here are some tips to enhance it:

- Include more vitamin C rich foods in your diet like citrus fruits.
- Stay hydrated and drink plenty of water.
- Consider probiotics.
- Get regular medical screening tests for people in your age group and risk category.

Output Explanation:

The Streamlit application presented in the screenshots showcases an intuitive and engaging user interface for the 'Predict Your Immunity Power' tool. At first glance, the app features a clean layout with a compelling title and relevant imagery that aligns with the health theme. The interface consists of dropdown menus, allowing users to easily input their personal health information corresponding to the various lifestyle factors that the machine learning model analyzes.

The simplicity of the interface is notable, with clear categorizations and selections that ensure the user can navigate without confusion. Each dropdown menu corresponds to a question from the original data collection form, encompassing areas such as age, gender, diet, sleep, and stress levels. This coherent structure facilitates an effortless user experience, making the process of data entry as straightforward as possible.

Upon submission of their data, users receive an estimated immunity power score, followed by customized health suggestions. These suggestions are particularly noteworthy as they are not generic but tailored to the immunity score predicted by the Random Forest model. For instance, users with a lower immunity power score receive comprehensive lifestyle and dietary recommendations aimed at bolstering their immune system.

The code snippet provided gives valuable insight into the backend operations of the app. It includes the data loading phase, the encoding of user inputs through a function, and the predictive model's utilization to estimate immunity power. The code is well-structured and modular, indicating a thoughtful design that adheres to good coding practices.

Overall, the application represents a seamless fusion of data science and user interface design, offering a tool that is not only functional but also user-centered and educational. This tool exemplifies how machine learning can be harnessed to provide individualized health insights, encouraging users to engage proactively with their wellbeing.

References:

1. https://streamlit.io/
2. https://www.health.harvard.edu/staying-healthy/how-to-boost-your-immune-system
3. https://diabetesjournals.org/care/article/27/3/813/22995/Inflammation-and Activated-Innate-Immunity-in-the
4. https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/immune-system
5. https://www.healthline.com/health/food-nutrition/foods-that-boost-the-immune-system
6. https://www.houstonmethodist.org/blog/articles/2020/mar/5-ways-to-boost-your-immune-system/
7. https://www.google.com/search?q=health+immunity+booster&oq=health+immunity+&gs_lcrp=EgZjaHJvbWUqBwgBEAAYgAQyCggAEEUYFhgeGDkyBwgBEAAYgAQyBwgCEAAYgAQyBwgDEAAYgAQyBwgEEAAYgAQyBggFEEUYPDIGCAYQRRg8MgYIBxBFGDzSAQg2NTQxajBqN6gCALACAA&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:7b28023a,vid:5RjEYf9887w,st:0

Biodata:

Name : Kuppireddygari Viswateja
Roll.no : 20BCI7235
Email : viswateja.20bci7235@vitap.ac.in
Permanent address : Hill view Apartments, KT road,Tirupati,517501

Name: T DEVI SAI JNANESWAR VUNDAVILLI
Roll.no: 20BCD7274
Email: saijnaneswar.20bcd7274@vitap.ac.in
Permanent address: 6-48 Rangampeta, opposite to union Bank, 533291,