**Project Title: Healthy Bites**

**Developed By Ibrahim Irfan**

**Project Overview**

A dietary food recommendation system is a tool that is designed to help businesses in the food industry improve customer satisfaction and loyalty by providing personalized recommendations for food and drinks based on an individual's preferences and past orders. The primary objective of a food recommendation system is to increase sales and revenue by encouraging customers to try new menu items and by encouraging repeat business. This can be achieved by analyzing customer data and using machine learning algorithms to understand patterns in customer behavior and preferences, and by presenting recommendations that are tailored to the individual's tastes and needs. By using a food recommendation system, businesses can improve their customer retention rates, increase customer satisfaction, and ultimately drive growth and profitability. Dietary food recommendation systems also help individuals who are on a diet and trying to maintain a healthy lifestyle plan out their daily meal intake. In this project, we will develop, train and test a Machine Learning model that analyzes users' dietary preferences, health goals, and nutritional requirements, ultimately providing personalized meal plans and recommendations. The solution will be used in a website application or even a mobile app. The problem is an unsupervised, offline and a model-based learning problem, since we are building a recommendation model from the whole unlabled data.

**Scope**

The scope of this project involved creating a Machine Learning model which recommends handmade products to consumers based on attributes such as nutritional information and recipes. The model could then be integrated into a mobile or web application using a backend infrastructure.

**Objectives**

* Gaining knowledge of AI and Machine Learning
* Understanding the concept of Supervised and Unsupervised Learning
* Collected relevant datasets
* Performing data processing and cleaning
* Developing a model to recommend dietary meals
* Fine Tuning the model
* Testing and evaluating the model
* Finding a conclusion

**Tools Used**

The model was developed using the Python programming language with the Jupyter Notebook environment. The main purpose of using Jupyter Notebook instead of an IDE such as PyCharm or Sublime Text is because Jupyter Notebook allows you to integrate markdown text along with Python code within one file. The Python libraries involved in the development process include Pandas, NumPy, Scikit-Learn, Seaborn, and Matplotlib.

**Development Process**

The development process was broken down into 4 steps:

1. **Data Collection and Pre-Processing:** There were two datasets used to pre-process the data required to develop the model. The data was first cleaned and analyzed before feature extraction techniques were used to separate the required data to train and test the model.
2. **Model Exploration:** Once the dataset was cleaned and finalized, I split the data into training set and testing set. I then experimented the model using different techniques such as Random Forest, Decision Trees and GuassianNB.
3. **Model Refinement:** Once I had completed experimenting on the model, I refined the model by creating a scaling and pipeline function.
4. **Testing and Evaluation:** After refining the model, I tested and evaluated the model by comparing the results of each classifier before coming to a conclusion.

**Key Decisions Made**

The key decisions I had to make during the process of developing the model included:

* Separating the categorical and numerical variables
* Performing univariate analysis
* Setting max values for calories, fat, saturated fat, cholesterol, sodium, carbohydrates, fiber, sugar and protein attributes.
* Exporting the final dataset into a csv and json file.

**Conclusion**

At the end of this project, we created a machine learning model that predicts what dietary meals a user trying to stay fit is most likely to choose from. The model can be integrated into a backend infrastructure for further use.

We can conclude that the model performs best using the K-Nearest Neighbour algorithm with an accuracy of 94.5% while the worst-case scenario is when the model's performance using ANN with an accuracy of 81.38%