# **Healthcare Sentimental Analysis and Prediction**

This program is designed to perform sentiment analysis on healthcare reviews. It takes a dataset of reviews, processes the text data, and classifies the sentiments as positive, neutral, or negative based on the ratings provided. The program follows a structured approach to text preprocessing, feature extraction, and model training using a Random Forest classifier.

## **Data Cleaning or Wrangling**

Data cleaning or wrangling is the process of transforming raw data into a structured and usable format. It is a crucial step in data analysis as it ensures the data is consistent, accurate, and ready for analysis. This process includes handling missing values, normalizing text data, removing unnecessary elements (such as stop words and punctuation), and preparing the data for further analysis or model building.

**Data:**

* **Raw Dataset:** The initial dataset downloaded from the online source. **A screenshot of a computer

  Description automatically generated**
* **Cleaned Dataset:** The dataset after processing, including text normalization and sentiment labeling.A screen shot of a computer

  Description automatically generated

**Dependencies:**

* **Pandas (pandas):** Used for data manipulation and analysis. It provides data structures like DataFrames to efficiently manage and clean data.
* **Numpy (numpy):** Essential for numerical computing and handling arrays in conjunction with Pandas.

## **Data Analysis**

Data analysis involves examining, cleaning, transforming, and modeling data to discover useful information, draw conclusions, and support decision-making. In this context, the analysis includes exploring the distribution of sentiments (positive, neutral, negative) within the dataset and generating descriptive statistics. Visualizing the data is also an essential part of analysis, as it helps in understanding trends, patterns, and relationships within the data.

* **Descriptive Statistics:** Provides summaries of numerical data, such as mean, count, and standard deviation, which are essential for understanding the basic characteristics of the dataset.
* **Data Visualization:** Visualizing data through graphs and plots allows for a more intuitive understanding of the data's distribution and relationships between variables. Tools like Matplotlib and Seaborn are commonly used for creating various types of visualizations, such as histograms, bar plots, and heatmaps.

**Result:**

* **Distribution of Sentiments:** The sentiment distribution plot reveals a higher frequency of positive sentiments compared to neutral and negative ones. A bar graph with different colored squares

  Description automatically generated
* **Word Cloud:** The word cloud highlights the most frequent terms in the reviews, with common words reflecting prevalent themes and sentiments. 
* **Confusion Matrices:** Both models (Random Forest and SVM) show good performance, with the confusion matrices indicating high accuracy and revealing common misclassifications. A diagram of a diagram

  Description automatically generated with medium confidence

**Dependencies:**

* **Pandas (pandas):** Used for performing data analysis operations, such as computing descriptive statistics and analyzing the distribution of sentiment labels.
* **Matplotlib (matplotlib):** A plotting library used for creating static, animated, and interactive visualizations in Python. It is essential for generating graphs and plots to visualize data trends and distributions.
* **Seaborn (seaborn):** A statistical data visualization library based on Matplotlib, providing a high-level interface for drawing attractive and informative statistical graphics.
* **WordCloud (wordcloud): A** library for generating word clouds, which visualize the frequency of words in text data, helping to identify prominent terms and patterns in textual data.

## **Prediction Model**

A prediction model is a mathematical tool used to predict future outcomes based on historical data. In the context of text data, machine learning models are often used to classify text into categories (e.g., positive, neutral, negative sentiments). The model typically involves several steps: text vectorization, model training, prediction, and evaluation.

* **Text Vectorization:** Converts text data into numerical features that can be processed by machine learning algorithms. TF-IDF (Term Frequency-Inverse Document Frequency) is commonly used to reflect the importance of words in a document relative to a collection of documents.
* **Model Training:** Involves fitting a machine learning model, such as a Random Forest classifier and SVM to the training data.
* **Prediction and Evaluation:** The trained model predicts sentiment labels for unseen data, and its performance is evaluated using metrics such as accuracy and a classification report.
* **Result:** The performance of the trained models (Random Forest and SVM) is evaluated on the test data.

1. **Random Forest Classifier:** **A screenshot of a computer screen

   Description automatically generated**
2. **Support Vector Machine (SVM):**

**A screenshot of a computer

Description automatically generated**

**Dependencies:**

* **Scikit-learn (sklearn):** A machine learning library in Python that provides tools for model building, including classification, regression, and clustering algorithms.
* **NLTK (nltk):** The Natural Language Toolkit, used for text processing tasks like tokenization, stop words removal, and lemmatization.