AI Course

Capstone Project   
Final Report

For students (instructor review required)

ⓒ2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright to this document.

This document is a literary property protected by copyright law, so reprinting and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from the copyright holder.

|  |
| --- |
| Drug Recommendation System - SmartMEDs |

30/1/2024

**The Outliers**

Madhumitha Ichapuram

Ranim Alfaraj

Ahmed Alghali

Abdelwahid Eltayeb

Taha Nasir

Mustafa Eltayeb

Content

**1. Introduction**

1.1. Background Information

1.2. Motivation and Objectives

1.3. Members and Role Assignments

1.4. Schedule and Milestones

**2. Project Execution**

2.1. Data Acquisition

2.2. Training Methodology

2.3. Workflow

2.4. System Diagram

**3. Results**

3.1. Data Preprocessing

3.2. Exploratory Data Analysis (EDA)

3.3. Modeling

3.4. User Interface

3.5. Testing and Improvements

**4. Projected Impact**

4.1. Accomplishments and Benefits

4.2. Future Improvements

**5. Team Member Review and Comment**

**6. Instructor Review and Comment**

1. Introduction

**1.1. Background Information**

Every day we see new advancements in technology contributing to the improvement of society in many fields, one of those fields being medicine. There are instances when patients are unable to schedule medical appointments due to several reasons and therefore require alternatives to the traditional methods of medical assistance. Machine learning has made life much easier and is now being implemented in many aspects of our daily lives. Our project aims to create a recommender system that prescribes drugs based on the symptoms a patient might be feeling. The dataset used in this project provides patient reviews on specific drugs, related conditions the patient has been suffering from, and a 10-star patient rating reflecting overall patient satisfaction.

**1.2. Motivation and Objectives**

This project equips patients to manage minor ailments independently, mitigating their reliance on costly medical consultations and reclaiming valuable time. By enabling the self-resolution of non-critical health concerns, independent of waiting rooms and appointments, patients achieve not only financial benefits but also the restoration of precious personal time. Furthermore, this project provides medical professionals with the opportunity to refine treatment recommendations through the utilization of patient feedback, thereby attaining a comprehensive understanding of drug efficacy across diverse demographic groups. Notably, this initiative extends beyond immediate personal advantages, encouraging patients with self-management skills and fostering a preventive healthcare approach, potentially mitigating the future necessity for intricate medical interventions. This project lays the foundation for a more equitable and data-informed healthcare system, paving the way for a future where personalized medicine empowers individuals and optimizes overall well-being.

**1.3. Members and Role Assignments**

|  |  |
| --- | --- |
| **Roles assigned** | **Members** |
| Data reading and EDA | Ahmed |
| Data Cleaning | Ranim MadhumithaAhmed |
| Data Preprocessing | Ahmed |
| Feature Engineering | MadhumithaAhmed |
| Natural Language techniques | Ahmed |
| API Integration | Madhumitha |
| Modeling and Tuning | RanimMadhumithaAbdalwahidMustafaAhmed |
| Prompt Engineering | Taha RanimAhmed |
| Customizing ChatGPT | Taha |

**Note**:

While every one of us contributed more to one area (each one of us has a main task). Still, each one of us worked in different areas (e.g. in modeling, Abdulwahid applied a KNN model, Ranim Applied a Random Forest, and Ahmed Applied the Decision Tree)

**1.4. Schedule and Milestones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Start date** | **End date** | **Responsibility** | **Deliverable** |
| **Data acquisition** | 10/1/2024 | 13/1/2024 | All members | Code  Data Report |
| **Preprocessing of the data** | 14/1/2024 | 27/1/2024 | Ahmed | Code  Data  Report |
| **Build and train the model** | 19/1/2024 | 31/1/2024 | All members | Code Report |
| **Evaluate the Performance** | 24/1/2024 | 31/1/2024 | Ahmed Madhumitha | Code Report |
| **Finalizing the Final report** | 27/1/2024 | 31/1/2024 | All members | Report |
| **Working on the PPT** | 29/1/2024 | 31/1/2024 | All members | PPT |
| **Preparing for the demo** | 31/1/2024 | 31/1/2024 | All members |  |
| **Submission of Capstone Final Report, Presentation and Coding Reference** | 31/1/2024 | 31/1/2024 | All members | Report  Presentation Coding |
| **Demo** | 1/2/2024 | 1/2/2024 | All members | PPT |

**Milestones**

* Exploratory Data Analysis (EDA): in the data used, the main challenges faced were null, duplicate, and out-of-range values in many columns, as well as a highly imbalanced target condition skewed towards the "Birth Control" class. Other tasks included handling similar data with different spellings and cleaning issues like regex values, text noise, and HTML entities from scraped web data. Analysis showed that after Date Parsing, all years contributed equally to the dataset, rendering the features nearly evenly distributed across different years and, consequently, making some columns seemingly useless.
* Data Preprocessing: over 100 unique conditions had incorrect spelling and therefore needed to be manually corrected. Other steps included the removal of regex, stop words, and sentence tokenization, followed by extracting lemmatized text. To address potential issues with models reacting randomly to zero values, data encoding covered the range [1,2,3]. Further enhancements involved utilizing NLTK Sentiment intensity scores to categorize reviews into positive, negative, and neutral, as well as mapping both ratings and useful count columns in a consistent manner. Lastly, LabelEncoder was applied to the drug\_name and condition (target) columns.
* Modeling: The models used were Random Forest, Decision Tree, and KNN, with the Decision Tree having the highest accuracy value with 0.71.
* Deployment: a custom GPT was created, allowing us to integrate our model into the ChatGPT 4.0 interface for a well-structured use case. The result was a user interface that allows users to input their medical conditions which are then put throughout the model to present a structured formatted response.

**2. Project Execution**

2.1. Data Acquisition

The dataset provides patient reviews on specific drugs, and related conditions and a 10-star patient rating reflecting overall patient satisfaction. The Dataset is from the UCI machine-learning repository; it has 215063 instances and 7 columns in a TSV format. It was scrapped from drugs.com but was acquired from UCI Machine Learning Repository.

It contains null and out-of-range values.

**2.2. Training Methodology**

We divided the dataset into x\_train, x\_test, y\_train, and y\_test, with a testing size of 20%. We used hyperparameter tuning in our training process and methods to find the best parameter, like GridSearch CV and cross-validation.

**Decision Tree:**

Decision tree is a simple model that is used for classification and regression problems.

the criterion used was *entropy* in the baseline model, but then we used grid search cv to tune the parameters using param\_grid of (param\_grid = {

'criterion': ['entropy', 'gini'],

'max\_depth': [None, 10, 20, 30],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [1, 2, 4],

'max\_features': [None, 'sqrt', 'log2']

})

and five cross-validations, it gives the highest accuracy among all other models due to its fast runtime and we used all our dataset to train it. The other models were using sampled data.

**KNN (K-Nearest Neighbors):**

* Distance Metric: Select an appropriate distance metric (e.g., Euclidean, cosine similarity) to measure similarity between reviews.
* Parameter Tuning: Experiment with different values of k (number of neighbors) to find the optimal setting.
* Prediction: For a new review, identify its k nearest neighbors in the training set and assign the most common medical condition among those neighbors.
* Considerations: KNN is simple to implement and interpret but might be less efficient for large datasets and sensitive to feature scaling.

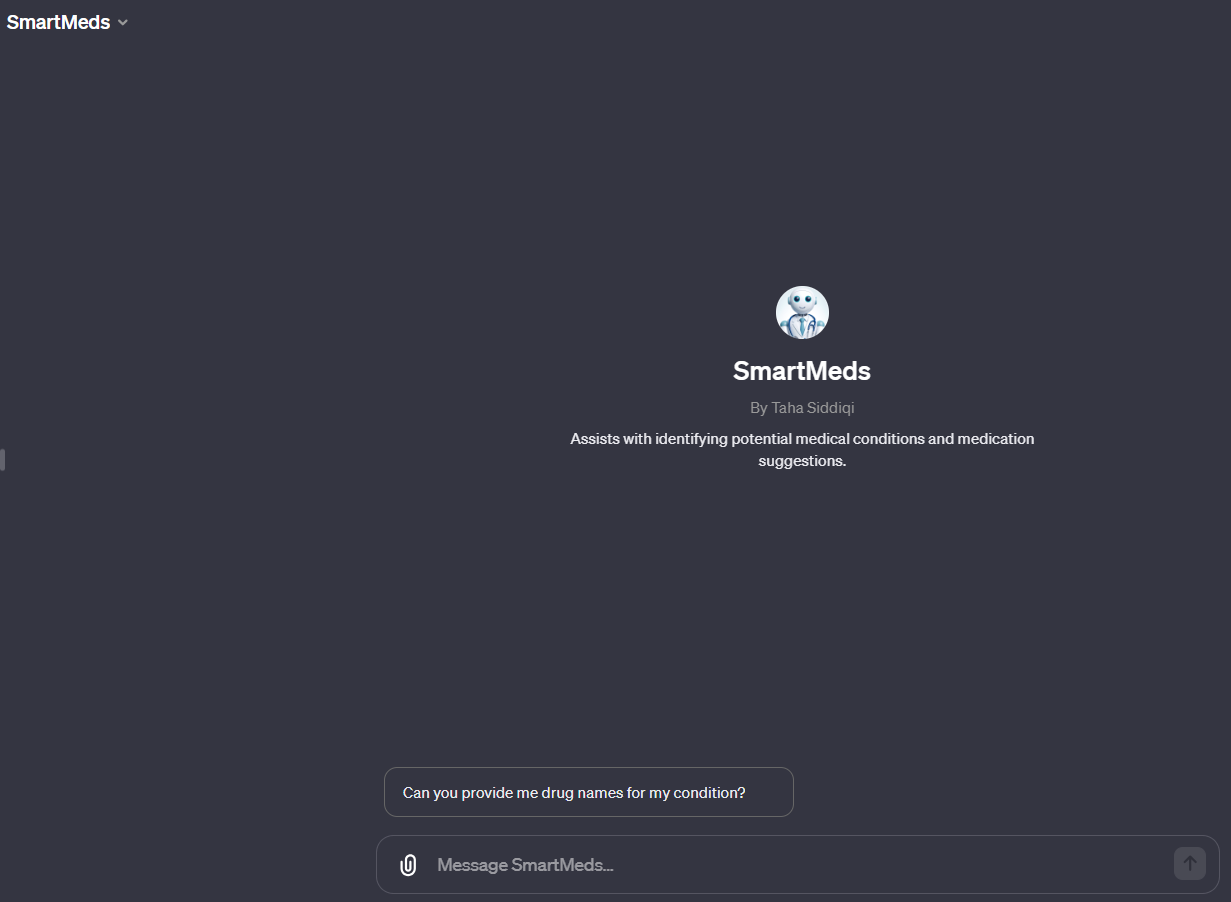
**Random Forest:**

* Model Selection: Select the number of trees in the forest and other hyperparameters, such as the depth of each tree and the minimum number of samples required to split a node. While it can work well with default settings, hyperparameter tuning can enhance the performance of this model.
* Training: The model is trained on the training set using a group of decision trees, with each tree trained on a bootstrap sample of the data with random subsets of features considered at each split.
* Prediction: Once the model is trained on historical data it can be used to predict the medical condition for new patient reviews or conditions.
* Considerations: Random Forest is robust to overfitting and performs well on a variety of datasets. It can handle high-dimensional data, and nonlinear relationships, and is less sensitive to hyperparameter tuning compared to some other models. This model provides built-in feature importance, aiding in understanding which features.
* Contribute most to predictions. However, this model can be computationally intensive, especially with many trees and the individual trees' decision paths may be challenging to interpret compared to simpler models like decision trees.

**Performance matrix equations:**

**ChatGPT 4.0:**

ChatGPT 4.0 allows users to create custom GPT use cases allowing us to integrate our model to their interface for a well-structured use case. The model was imported as a dataset and presented through prompts and instructions allowing the GPT to present our data as a functional final draft. The user interface allows users to input their medical conditions which are then put throughout the model to present a structured formatted response.



**2.3. Workflow**

We divided the work into different spaces and notebooks. We had a first draft of data cleaning, and the outcome was a CSV file, which helped us add any cleaning functionalities to the first draft. We use it to produce a more reliable and cleaned dataset every time the data passes through draft number 2 for a cleaner version that contains manual detection and forward-detecting instances. After getting our CSV file, due to the high complexity and size of the data, we export the final versions as a CSV file. and decided that everyone would try modeling with different combinations of parameters on their own (to prevent RAM and session Crashes).

**2.4. System Design**

We combined different approaches and techniques to build this system We used natural language techniques to highlight the importance of recommending a drug based on the review We brought our processed data and fit it into the CustomChatGPT model to make a conversational bot, and that required some prompt engineering methods the bot is trained to text in calm and friendly tone.tone of texting It provides you with the top 5 possible medications based on your condition and you can take it even further by asking the bot about the community review of a certain medication

3. Results

**3.1. Data Preprocessing**

The data preprocessing consists of cleaning null and duplicate values by dropping them.

There are features like NLTK Vader score that give a sentiment about the ***review*** and put it into 3 categories (positive, negative, and neutral)

We mapped both the ***rating*** and ***useful\_count*** columns the same way. When handling the ***drug\_name*** and ***condition*** (our target) columns, we used LaberEncoder

**3.2. Exploratory Data Analysis (EDA)**

**Our data**

Is unique: An identifier for each drug name.

drug name: The name of the drug for which a review is made.

condition: The name of the medical condition for which the medicine is used. review: The review made by patients for a particular medicine.

rating: Ratings given by the patients to each medicine are given on a scale of 10, where 10 represents the maximum efficacy.

date: date of review entry.

usefulCount: The number of users who found the review useful.

The EDA was challenged due to the amount of data and its biases. We discovered that our data needs to be cleaned as soon as we detect null values that are duplicated, even in a range, e.g., finding non-making sense observation ‘wrong information’ values and outliers. We go beyond that when we set our DataFrame to view the maximum rows and follow line by line, and we spot a pattern of similar conditions with different spellings. Besides many of the data collection problems we discovered, the biggest one was the bias of our data reviews, which was highly skewed towards the class of *Birth control***,** indicating that the dataset is highly imbalanced.

**3.3. Modeling**

To get the most promising machine learning model before tweaking the hyper-parameters, we start by trying different methods and models

we used:

1. Random Forest
2. Random Forest Regressor
3. K-nearest Neighbor
4. Decision Tree

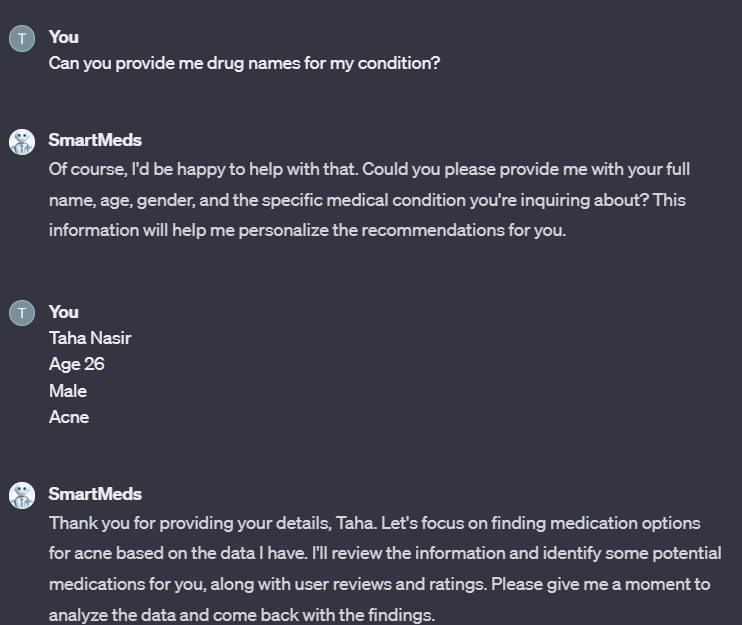
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model name** | **Sklearn model name** | **Accuracy** | **Mean Absolute Error (Degrees)** | **F-1 Score** | **Recall** | **Precision** |
| Random Forest | RandomForestClassifier | 0.68 | 85.01 | 0.68 | 0.68 | 0.68 |
| Random Forest Regressor | RandomForestRegressor | 0.69 | 82.54 | 0.69 | 0.69 | 0.69 |
| Decision Tree | DecisionTreeclassifier | 0.71 | 75.94 | 0.72 | 0.72 | 0.72 |
| KNN | KNeighborclassifier | 0.69 | 82.54 | 0.69 | 0.69 | 0.69 |

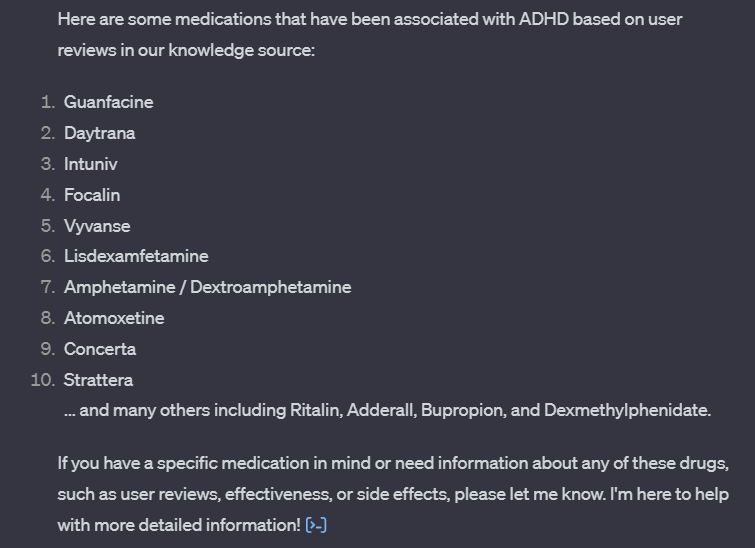
Table 1: The performance evaluation matrices of ML models

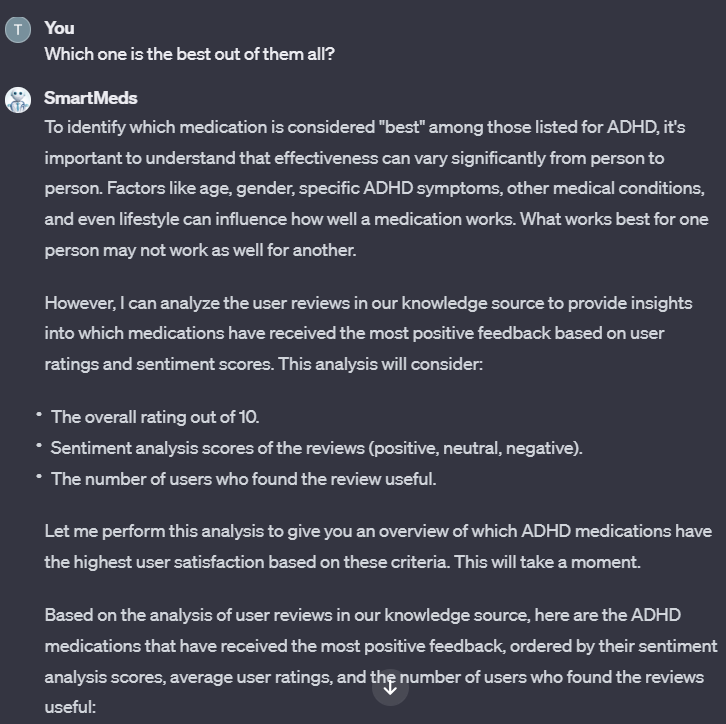
Table 2: The Hyperparameter Tuning of ML models

|  |  |
| --- | --- |
| **Model name** | **Hyperparameter Tuning** |
| Random Forest | Used with RandomizedSsearch to find the best  Max depth= 18  No. of estimators=2255 |
| Random Forest Regressor | Max depth= 18  No. of estimators=255 |
| Decision Tree | Used the best parameters using GridsearchV  Criterion = entropy  max depth = [10,20, 25,50] |
| KNN | Useds grid search with cross-validation, the best n\_neighbors was found to be n-neighbors=9 |

**3.4. User Interface**







**3.5. Testing and Improvements**

We fixed the biases of our ***condition*** columns by replacing the similar condition with one label and removing out of range values. We scaled the sentiment into 3 groups, which made it easier for the model to use. We used a more optimized version and tuned one of the models.

The training process needed more time, and it crashed when using more powerful models like ***catboost***, ***SDG***, and ***XGBboost,*** so we trained a simple model first ***decision tree*** and sampled the data for a decent model like ***RandomForest***.

There were attempts to reduce the training time by converting the Data types into (int) rather than (float64)

**4.1. Accomplishments and Benefits**

With this project, we have accomplished exactly what we wanted. We made it much easier for people who don’t have the time, financial capabilities, and such to access a knowledgeable medicine AI with a dependable database that can assign them roughly which medicine will help aid whatever ailment the patient may suffer from. This will hopefully be the blueprint to integrate such ideas into the real world and benefit the society we live in as a whole.

**4.2. Future Improvements**

In the future, we are aiming to extract more information from our lemmatized review columns, like TF-IDF word cloud and vectorizer. Train the data in a more powerful model to improve the accuracy and relevance of medication suggestions. This includes considering additional factors such as patient demographics or medical history and fine-tuning the weighting of different factors in the recommendation process. Also, Apply more API For a larger database and more information retrieval. Moreover, we can implement functionality to provide real-time updates and alerts to users, such as notifications about new medications or changes in recommendations based on updated information or user feedback.

5. Team Member Review and Comment

|  |
| --- |
| <ATTACH A TEAM PICTURE HERE> |

|  |  |
| --- | --- |
| NAME | REVIEW and COMMENT |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

6. Instructor Review and Comment

|  |  |  |
| --- | --- | --- |
| CATEGORY | SCORE | REVIEW and COMMENT |
| IDEA | \_\_/10 |  |
| APPLICATION | \_\_/30 |  |
| RESULT | \_\_/30 |  |
| PROJECT MANAGEMENT | \_\_/10 |  |
| PRESENTATION & REPORT | \_\_/20 |  |
| TOTAL | \_\_/100 |  |