AI-Driven Health Diagnosis and Recommendations System

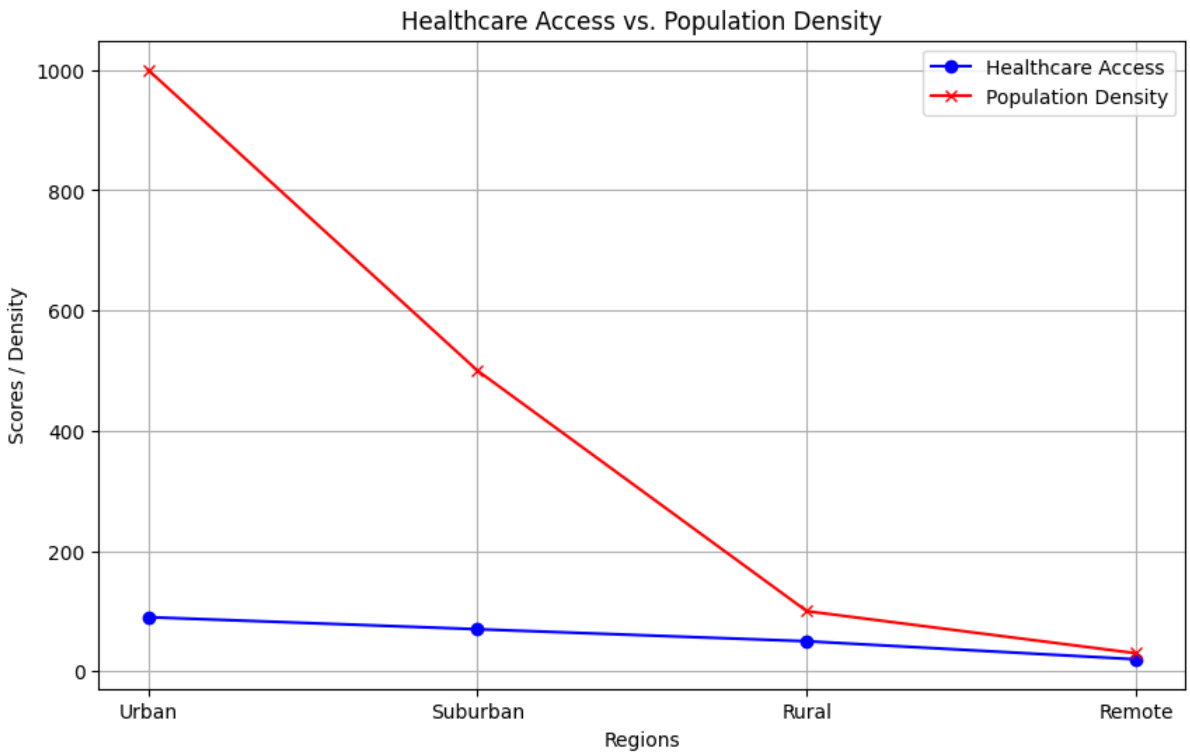
(HealthCare Zone)

Name: Prakhar Nagar.

PROBLEM STATEMENT

With limited access to immediate healthcare in various regions, individuals often face delays in receiving medical care and initial diagnosis, which can impact treatment outcomes. This project addresses the need for a preliminary, AI-driven diagnostic tool that allows users to input their symptoms and receive an initial disease prediction along with health recommendations. This solution empowers users to make informed health decisions by offering actionable advice without waiting for formal consultations. The AI-based system focuses on:

1. Delivering disease predictions based on user symptoms.
2. Providing tailored health recommendations, including medications, diet plans, and workout suggestions.

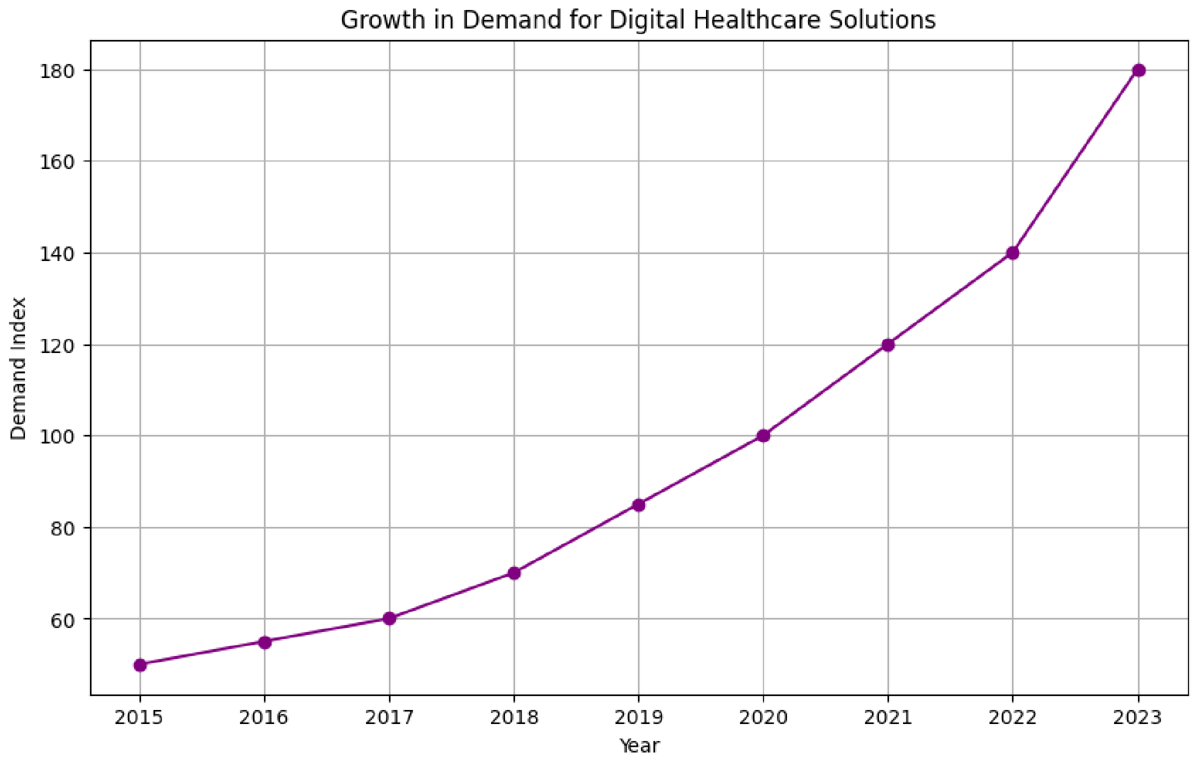


Healthcare Access vs. Population Density

BUSINESS NEED ASSESSMENT

There is a growing demand for **self-diagnostic and telemedicine solutions**, driven by advancements in AI and increasing healthcare digitalization. AI-powered diagnostics not only make healthcare more accessible but also help reduce the load on medical professionals by addressing non-critical cases independently. Key needs addressed include:

- **Immediate diagnostic insights**: By providing quick assessments, users can better understand their health concerns.
- **Cost efficiency for healthcare systems**: Reducing minor consultation loads, especially for primary care, can lead to cost savings.
- **Expanding access to care**: This system reaches populations with limited access to in-person medical consultations, fulfilling a significant healthcare gap.



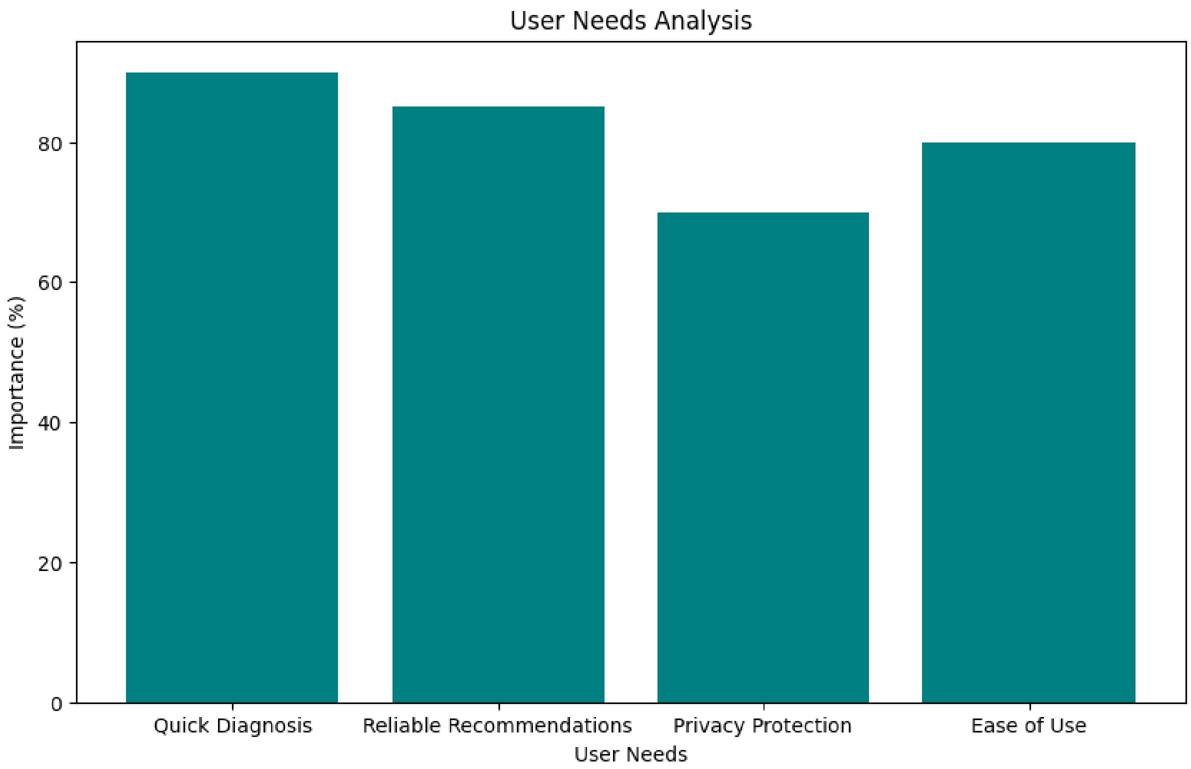
Growth in Demand for Digital Healthcare Solutions

TARGET SPECIFICATIONS AND CHARACTERIZATIONS

This AI health system aims to meet the following specifications to serve a diverse user base:

- **Model Accuracy**: The SVC model must predict diseases with high accuracy based on user-provided symptoms, leveraging a symptom-to-disease dictionary for mapping.

- **Recommendation System**: The app should pull relevant, personalized recommendations, including diet and exercise, specifically for each predicted disease.
- **User Accessibility**: Designed for intuitive use, making it suitable even for non-technical users. The interface should be clear and direct, with easy input and output methods for symptoms and results.
- **Data Privacy**: As the system handles sensitive health data, it must comply with data protection laws and ensure user data is securely stored and processed.

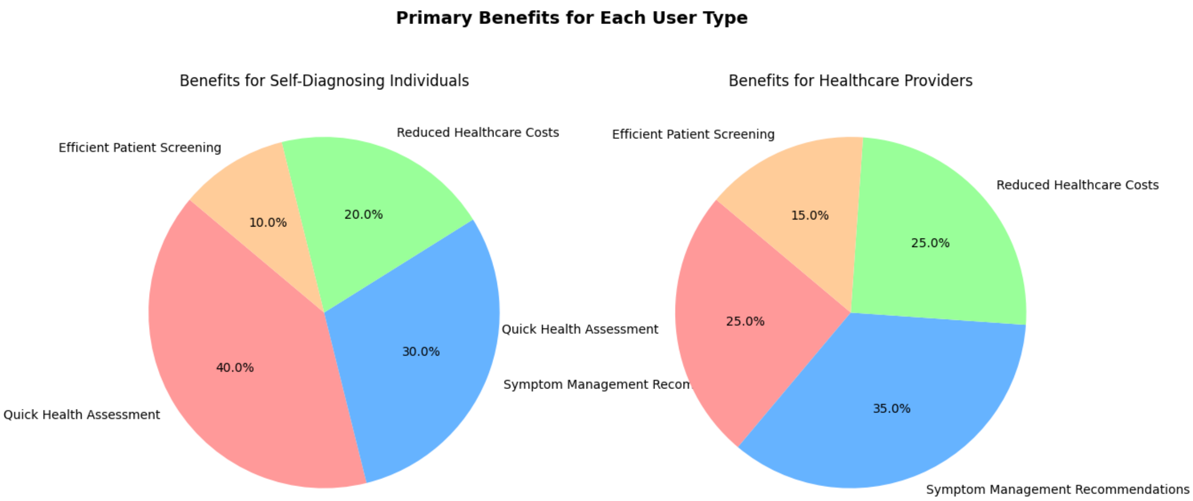


User Needs Analysis

CONCEPT GENERATION

The primary aim is to **empower users with an accessible AI-driven tool** for initial health diagnostics and advice. The potential applications extend beyond individual use, as healthcare providers can integrate this system for initial screenings, which can save time for both patients and doctors. Core aspects include:

- **User Self-Diagnosis**: The application allows users to receive a preliminary diagnosis by simply entering symptoms.
- **Adaptability for Medical Providers**: Healthcare providers may also use this system for pre-visit screening, helping them prioritize patient care and resource allocation.

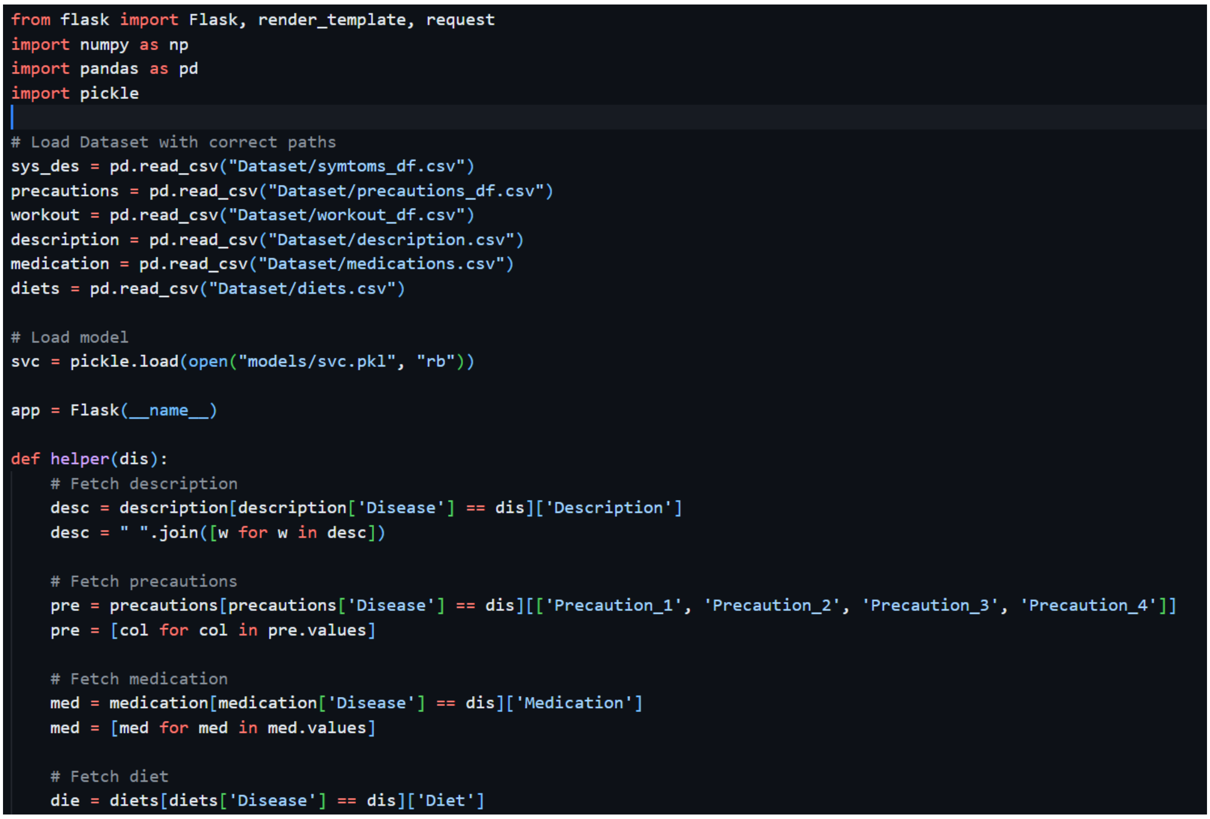


User and Provider Benefits

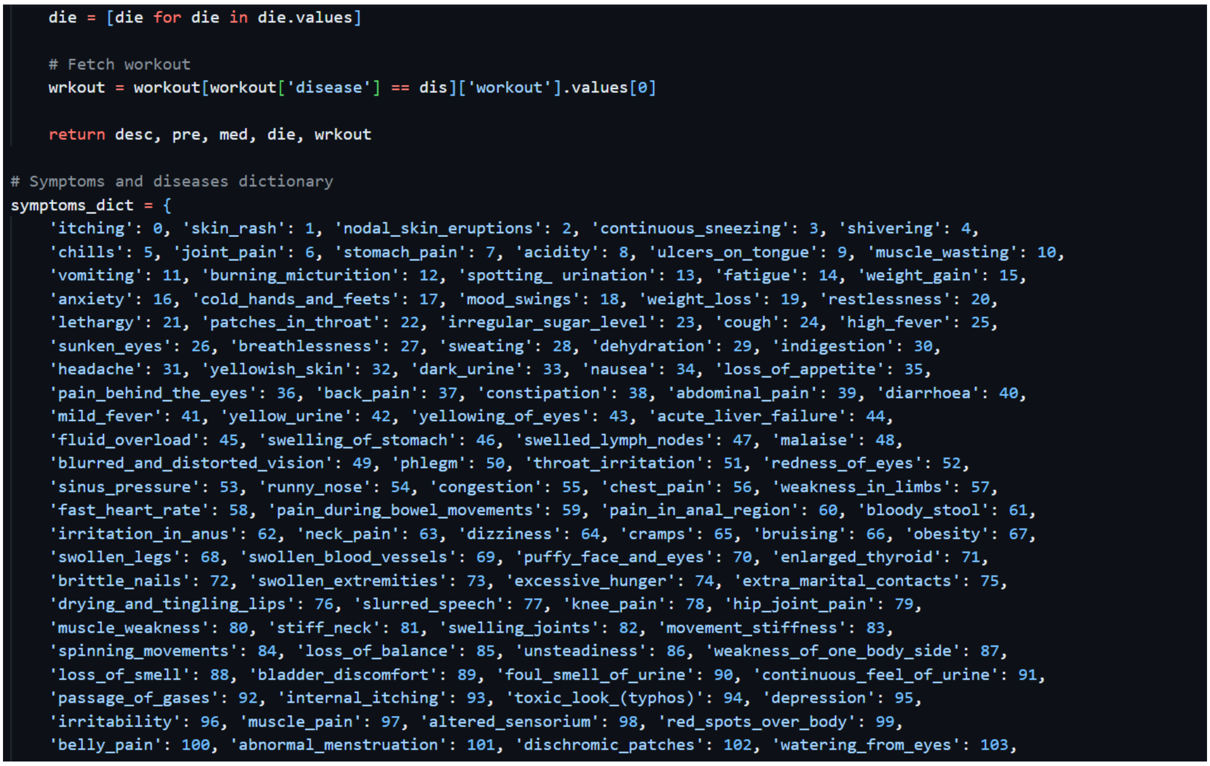
CONCEPT DEVELOPMENT

The health diagnosis system is composed of two main components:

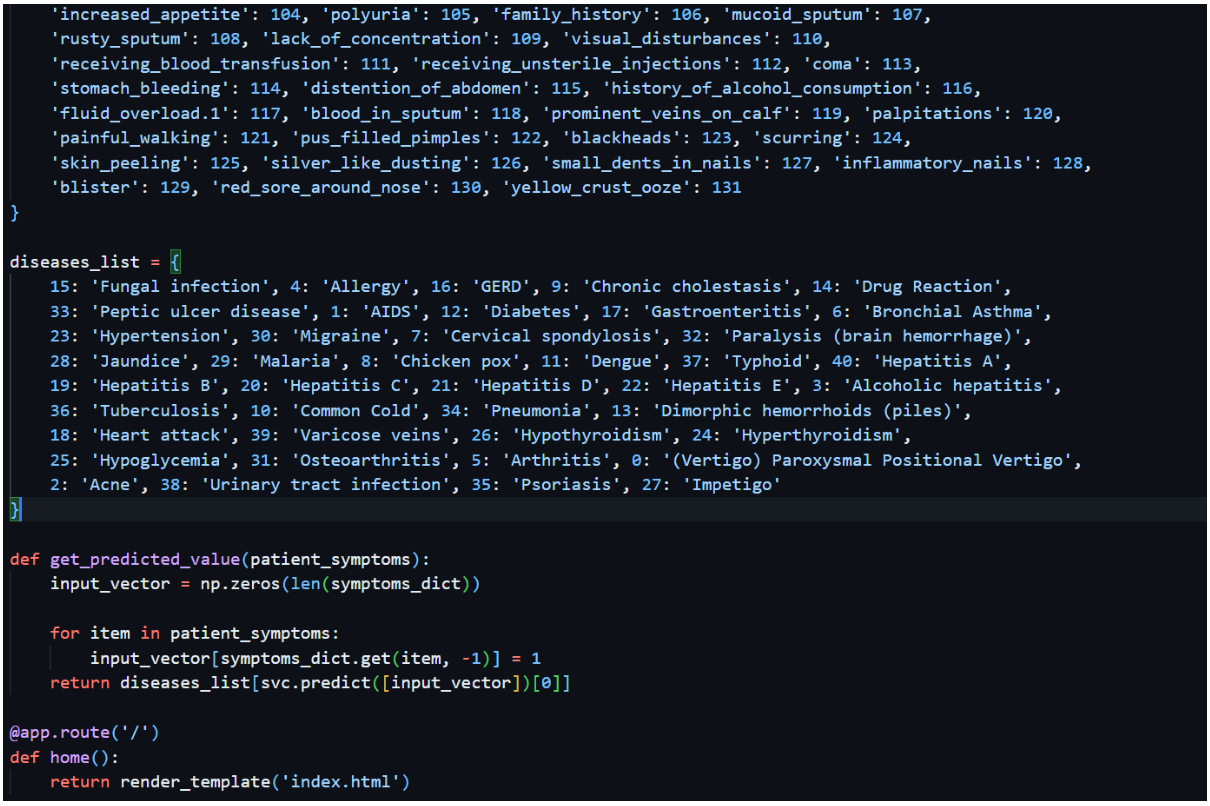
1. **Machine Learning Model (SVC)**: The core model is trained to predict diseases based on a binary symptom vector, where each symptom is encoded in the form of a feature vector.



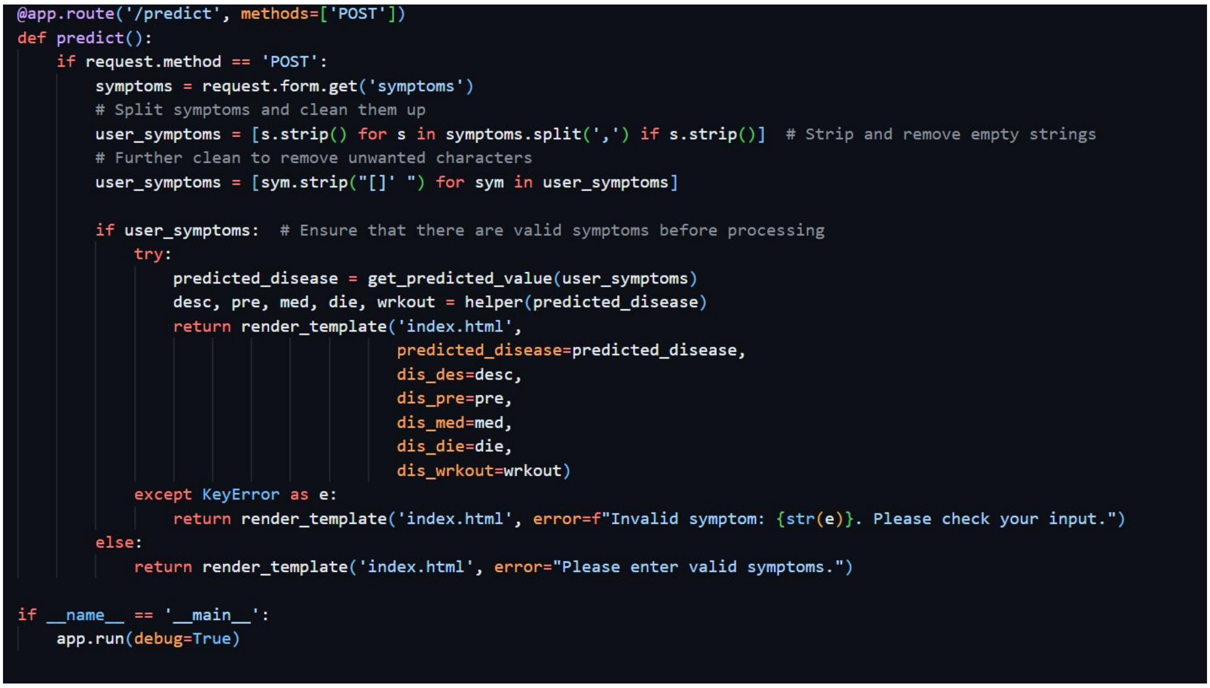
```python
from flask import Flask, render_template, request
import numpy as np
import pandas as pd
import pickle

# Load Dataset with correct paths
sys_des = pd.read_csv("Dataset/symtoms_df.csv")
precautions = pd.read_csv("Dataset/precautions_df.csv")
workout = pd.read_csv("Dataset/workout_df.csv")
description = pd.read_csv("Dataset/description.csv")
medication = pd.read_csv("Dataset/medications.csv")
diets = pd.read_csv("Dataset/diets.csv")

# Load model
svc = pickle.load(open("models/svc.pkl", "rb"))

app = Flask(__name__)

def helper(dis):
    # Fetch description
    desc = description[description['Disease'] == dis]['Description']
    desc = " ".join([w for w in desc])

    # Fetch precautions
    pre = precautions[precautions['Disease'] == dis][['Precaution_1', 'Precaution_2', 'Precaution_3', 'Precaution_4']]
    pre = [col for col in pre.values]

    # Fetch medication
    med = medication[medication['Disease'] == dis]['Medication']
    med = [med for med in med.values]

    # Fetch diet
    die = diets[diets['Disease'] == dis]['Diet']
```



```python
    die = [die for die in die.values]

    # Fetch workout
    wrkout = workout[workout['disease'] == dis]['workout'].values[0]

    return desc, pre, med, die, wrkout

# Symptoms and diseases dictionary
symptoms_dict = {
    'itching': 0, 'skin_rash': 1, 'nodal_skin_eruptions': 2, 'continuous_sneezing': 3, 'shivering': 4,
    'chills': 5, 'joint_pain': 6, 'stomach_pain': 7, 'acidity': 8, 'ulcers_on_tongue': 9, 'muscle_wasting': 10,
    'vomiting': 11, 'burning_micturition': 12, 'spotting_ urination': 13, 'fatigue': 14, 'weight_gain': 15,
    'anxiety': 16, 'cold_hands_and_feets': 17, 'mood_swings': 18, 'weight_loss': 19, 'restlessness': 20,
    'lethargy': 21, 'patches_in_throat': 22, 'irregular_sugar_level': 23, 'cough': 24, 'high_fever': 25,
    'sunken_eyes': 26, 'breathlessness': 27, 'sweating': 28, 'dehydration': 29, 'indigestion': 30,
    'headache': 31, 'yellowish_skin': 32, 'dark_urine': 33, 'nausea': 34, 'loss_of_appetite': 35,
    'pain_behind_the_eyes': 36, 'back_pain': 37, 'constipation': 38, 'abdominal_pain': 39, 'diarrhoea': 40,
    'mild_fever': 41, 'yellow_urine': 42, 'yellowing_of_eyes': 43, 'acute_liver_failure': 44,
    'fluid_overload': 45, 'swelling_of_stomach': 46, 'swelled_lymph_nodes': 47, 'malaise': 48,
    'blurred_and_distorted_vision': 49, 'phlegm': 50, 'throat_irritation': 51, 'redness_of_eyes': 52,
    'sinus_pressure': 53, 'runny_nose': 54, 'congestion': 55, 'chest_pain': 56, 'weakness_in_limbs': 57,
    'fast_heart_rate': 58, 'pain_during_bowel_movements': 59, 'pain_in_anal_region': 60, 'bloody_stool': 61,
    'irritation_in_anus': 62, 'neck_pain': 63, 'dizziness': 64, 'cramps': 65, 'bruising': 66, 'obesity': 67,
    'swollen_legs': 68, 'swollen_blood_vessels': 69, 'puffy_face_and_eyes': 70, 'enlarged_thyroid': 71,
    'brittle_nails': 72, 'swollen_extremeties': 73, 'excessive_hunger': 74, 'extra_marital_contacts': 75,
    'drying_and_tingling_lips': 76, 'slurred_speech': 77, 'knee_pain': 78, 'hip_joint_pain': 79,
    'muscle_weakness': 80, 'stiff_neck': 81, 'swelling_joints': 82, 'movement_stiffness': 83,
    'spinning_movements': 84, 'loss_of_balance': 85, 'unsteadiness': 86, 'weakness_of_one_body_side': 87,
    'loss_of_smell': 88, 'bladder_discomfort': 89, 'foul_smell_of_urine': 90, 'continuous_feel_of_urine': 91,
    'passage_of_gases': 92, 'internal_itching': 93, 'toxic_look_(typhos)': 94, 'depression': 95,
    'irritability': 96, 'muscle_pain': 97, 'altered_sensorium': 98, 'red_spots_over_body': 99,
    'belly_pain': 100, 'abnormal_menstruation': 101, 'dischromic_patches': 102, 'watering_from_eyes': 103,
```



```python
    'increased_appetite': 104, 'polyuria': 105, 'family_history': 106, 'mucoid_sputum': 107,
    'rusty_sputum': 108, 'lack_of_concentration': 109, 'visual_disturbances': 110,
    'receiving_blood_transfusion': 111, 'receiving_unsterile_injections': 112, 'coma': 113,
    'stomach_bleeding': 114, 'distention_of_abdomen': 115, 'history_of_alcohol_consumption': 116,
    'fluid_overload.1': 117, 'blood_in_sputum': 118, 'prominent_veins_on_calf': 119, 'palpitations': 120,
    'painful_walking': 121, 'pus_filled_pimples': 122, 'blackheads': 123, 'scurring': 124,
    'skin_peeling': 125, 'silver_like_dusting': 126, 'small_dents_in_nails': 127, 'inflammatory_nails': 128,
    'blister': 129, 'red_sore_around_nose': 130, 'yellow_crust_ooze': 131
}

diseases_list = {
    15: 'Fungal infection', 4: 'Allergy', 16: 'GERD', 9: 'Chronic cholestasis', 14: 'Drug Reaction',
    33: 'Peptic ulcer disease', 1: 'AIDS', 12: 'Diabetes', 17: 'Gastroenteritis', 6: 'Bronchial Asthma',
    23: 'Hypertension', 30: 'Migraine', 7: 'Cervical spondylosis', 32: 'Paralysis (brain hemorrhage)',
    28: 'Jaundice', 29: 'Malaria', 8: 'Chicken pox', 11: 'Dengue', 37: 'Typhoid', 40: 'Hepatitis A',
    19: 'Hepatitis B', 20: 'Hepatitis C', 21: 'Hepatitis D', 22: 'Hepatitis E', 3: 'Alcoholic hepatitis',
    36: 'Tuberculosis', 10: 'Common Cold', 34: 'Pneumonia', 13: 'Dimorphic hemorrhoids (piles)',
    18: 'Heart attack', 39: 'Varicose veins', 26: 'Hypothyroidism', 24: 'Hyperthyroidism',
    25: 'Hypoglycemia', 31: 'Osteoarthritis', 5: 'Arthritis', 0: '(Vertigo) Paroxysmal Positional Vertigo',
    2: 'Acne', 38: 'Urinary tract infection', 35: 'Psoriasis', 27: 'Impetigo'
}

def get_predicted_value(patient_symptoms):
    input_vector = np.zeros(len(symptoms_dict))

    for item in patient_symptoms:
        input_vector[symptoms_dict.get(item, -1)] = 1
    return diseases_list[svc.predict([input_vector])[0]]

@app.route('/')
def home():
    return render_template('index.html')
```



```
@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        symptoms = request.form.get('symptoms')
        # Split symptoms and clean them up
        user_symptoms = [s.strip() for s in symptoms.split(',') if s.strip()]  # Strip and remove empty strings
        # Further clean to remove unwanted characters
        user_symptoms = [sym.strip("[]' ") for sym in user_symptoms]

        if user_symptoms:  # Ensure that there are valid symptoms before processing
            try:
                predicted_disease = get_predicted_value(user_symptoms)
                desc, pre, med, die, wrkout = helper(predicted_disease)
                return render_template('index.html',
                                       predicted_disease=predicted_disease,
                                       dis_des=desc,
                                       dis_pre=pre,
                                       dis_med=med,
                                       dis_die=die,
                                       dis_wrkout=wrkout)
            except KeyError as e:
                return render_template('index.html', error=f"Invalid symptom: {str(e)}. Please check your input.")
        else:
            return render_template('index.html', error="Please enter valid symptoms.")

if __name__ == '__main__':
    app.run(debug=True)
```

2. **Flask Web Application**: The web application is designed to display the predicted disease and health recommendations in an intuitive format. It includes error handling for invalid inputs and ensures clean, organized user interactions.

CODE IMPLEMENTATION (SMALL SCALE)

Code Highlights

1. **Helper Function**: The helper() function retrieves key information such as description, precautions, medications, diets, and workouts associated with each disease from CSV datasets.

2. **Prediction Function**: The get_predicted_value() function converts user symptoms into a binary vector for prediction, which the SVC model uses to predict the most likely disease.

3. **Flask Routes**:

   - /: Displays the home page where users enter symptoms.

   - /predict: Processes the form input and returns the predicted disease and associated health recommendations.

FINAL PRODUCT PROTOTYPE

The prototype includes the following functionalities:

1. **User Symptom Input and Prediction**: Allows users to input symptoms, which are processed by the SVC model to predict the disease.

2. **Recommendation Display**: Displays personalized health recommendations, including medications, diets, and workout suggestions, specific to the predicted disease.

3. **Error Handling**: Prevents empty or incorrect inputs from processing, ensuring the user experience remains smooth.

BUSINESS MODEL

Consulting Business Model:

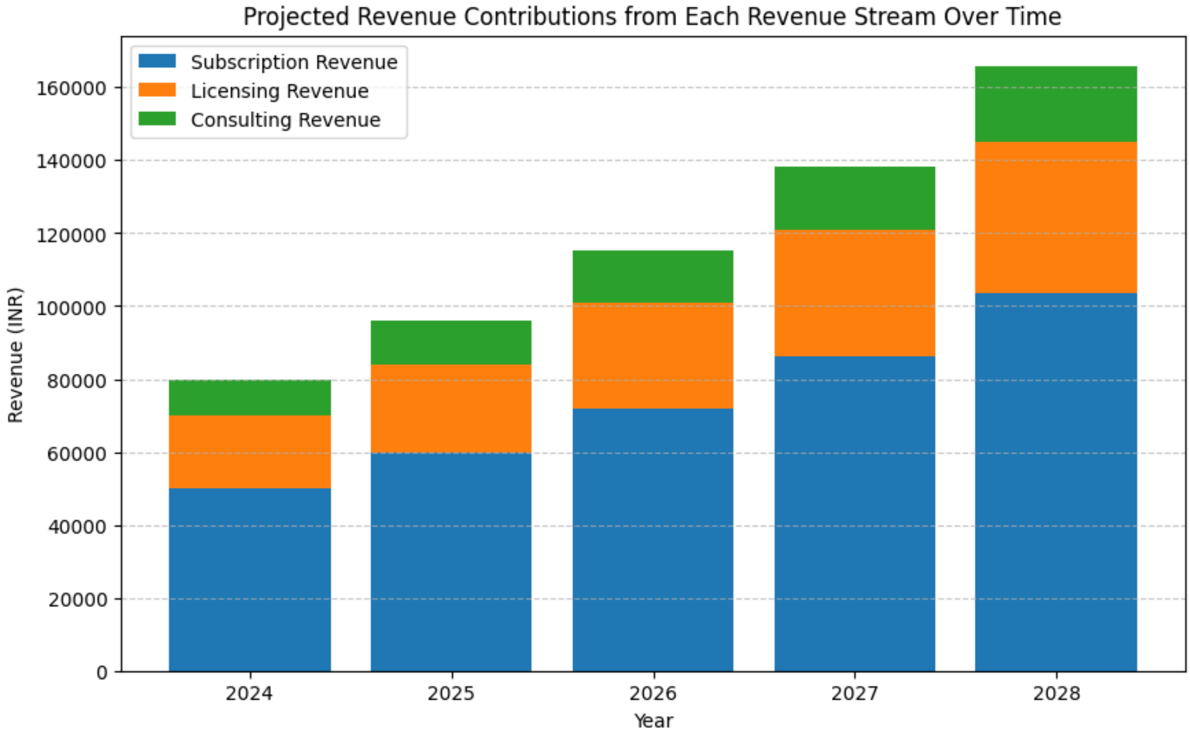
1. **Value Proposition**: This diagnostic tool provides accessible and affordable preliminary health assessments, with a focus on fast delivery and reliable recommendations.

2. **Revenue Channels**:

   - **Subscription Model**: Monthly access fees for individual users, enhancing user retention.

   - **Licensing Agreements**: For healthcare providers, offering integrated diagnostic tools to improve efficiency.

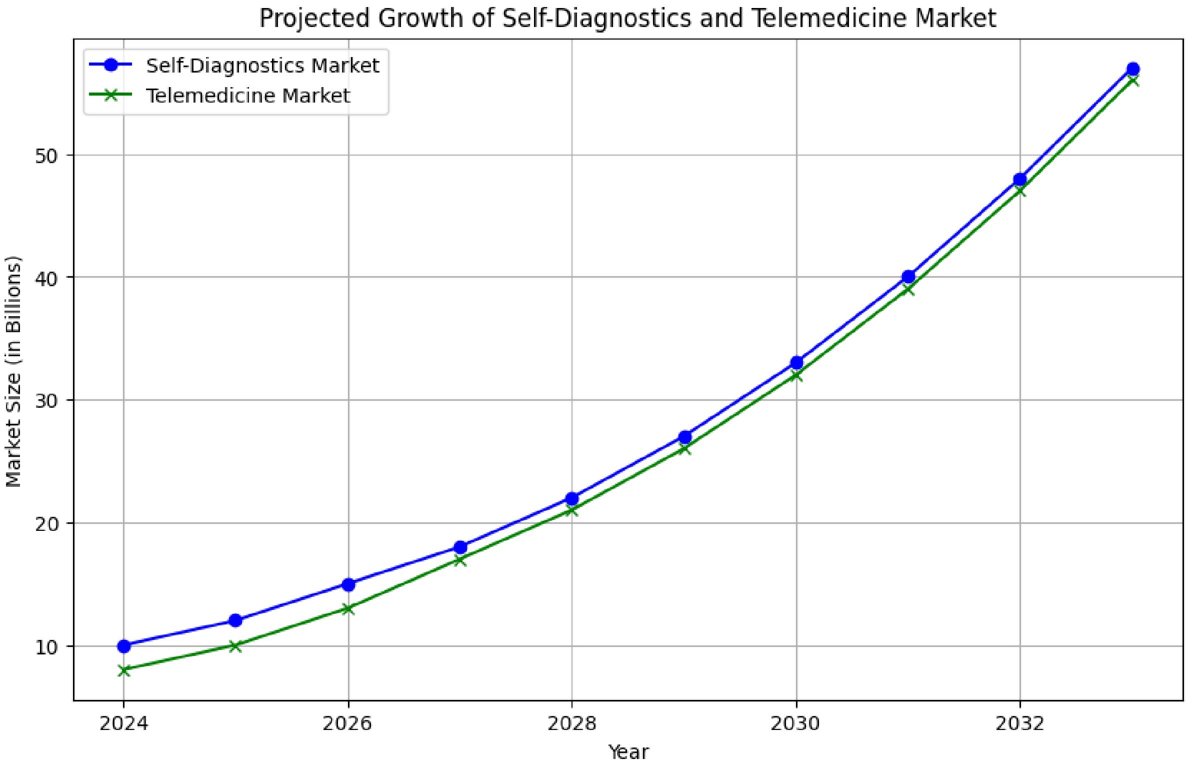
3. **Expansion Opportunities**: Potential to grow by partnering with health insurers or pharmaceutical companies to offer tailored health solutions.



MARKET ANALYSIS

The AI diagnostic tool market is expected to grow as telemedicine and self-diagnostic tools become more widely accepted. This system appeals to:

1. **Individual Users**: Users who want quick health assessments.
2. **Healthcare Providers**: Clinics and telemedicine services seeking tools to improve patient intake processes and initial assessments.



OPERATING PLAN

A core team of three to four professionals—two ML practitioners and one full-stack developer—will be sufficient to develop and maintain this product. The system can be hosted on cloud platforms, allowing for flexibility in scaling up based on user demand. Model retraining can occur periodically to adapt to new health data and improve accuracy.

MARKETING PLAN

Initial marketing efforts will target individual users and healthcare providers through digital channels. Emphasis will be placed on showcasing the tool's ability to provide accurate, timely health assessments. Expansion will involve outreach to clinics and partnerships with digital healthcare providers.

FINANCIAL EQUATION

Assumptions:

- **Monthly Subscription Fee: ₹500**
- **Monthly Operating Costs: ₹2,000**

Revenue Calculation

For 300 users in a given month:

Revenue (June) = 500 × 300 – 2000 = ₹ 1,49,800

As the user base grows, monthly revenue is expected to increase proportionally, making this a scalable business model.

References:

1. **SPD Group.** "Machine Learning and AI in the Healthcare Industry." SPD Group, 2021. Available at: https://spd.group/machine-learning/machine-learning-and-ai-in-food-industry/.
2. **Columbus, L.** "10 Ways Machine Learning Is Revolutionizing Supply Chain Management." *Forbes*, June 11, 2018. Available at: https://www.forbes.com/sites/louiscolumbus/2018/06/11/10-ways-machine-learning-is-revolutionizing-supply-chain-management.
3. **Analytics Vidhya.** "A Comprehensive Guide to Time Series Analysis." Analytics Vidhya, October 2021. Available at: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/.
4. **IBEF.** "India to Become the EdTech Capital of the World." *Indian Brand Equity Foundation (IBEF)*. Available at: https://www.ibef.org/blogs/india-to-become-the-edtech-capital-of-the-world.
5. **IBEF.** "Industry Insights: Real Estate and Biotechnology Sectors in India." *Indian Brand Equity Foundation (IBEF)*, 2021. Available at: https://www.ibef.org/industry/biotechnology-india/infographic.
6. **Investopedia.** "Business Model Examples Explained." *Investopedia*, 2022. Available at: https://www.investopedia.com/terms/b/businessmodel.asp.
7. **Tracxn Technologies Ltd.** "Market Data for Telemedicine and Diagnostics." *Tracxn EdTech Report*, 2023. Available at: https://www.tracxn.com.

GitHub Link (for the project):

https://github.com/PRAKHARNAGAR2003/HealthCare-Zone