Proudly supported by

SAP edunet foundation code unnati

PROJECT TITLE: MEDPREDICT

CU_CP_TEAM-5637

TEAM LEADER: MITHESH RAYABARAPU

TEAM MEMBERS:

ARAVINDA SWAMY TELU

HARISH PASUNOORI

ABSTRACT

In modern healthcare, the early detection and prevention of health conditions are crucial for improving patient outcomes. MEDPREDICT is an innovative project designed to predict and visualize patient health conditions using vital signs such as heart rate, blood pressure, body temperature, and oxygen levels. By leveraging the power of Artificial Intelligence (AI), Machine Learning (ML), and Data Analytics, this project aims to provide healthcare professionals with real-time, data-driven insights into the health status of their patients. The system uses advanced machine learning algorithms to analyze historical and real-time patient data, enabling accurate predictions of potential health risks such as heart attacks, respiratory distress, and other critical conditions. Along with these predictions, MEDPREDICT offers visualizations that help healthcare providers quickly understand trends and make informed decisions. Early warnings generated by the system allow for timely interventions, potentially preventing serious health complications. Ultimately, MEDPREDICT seeks to enhance healthcare decision-making by turning data into actionable insights, improving patient care, and promoting a more proactive approach to healthcare management.

INTROUCTION

The healthcare industry is increasingly embracing technology to improve patient care, streamline operations, and enhance decision-making. One of the most promising advancements is the use of data analytics and artificial intelligence (AI) to predict health conditions before they become critical. While medical professionals use their expertise to diagnose and treat patients, the process of identifying potential health risks often depends on timely access to accurate data. This is where MEDPREDICT comes into play.

MEDPREDICT is a project aimed at predicting and visualizing patient health conditions based on real-time vital data. By utilizing advanced techniques in AI, machine learning (ML), and data analytics, the system analyzes patients' vital signs, such as heart rate, blood pressure, oxygen saturation, and temperature, to forecast potential health issues. These predictions not only provide early warnings of health risks but also generate visual insights that help healthcare providers make informed decisions and take timely action.

The core objective of MEDPREDICT is to move from reactive to proactive healthcare. By identifying health risks early, the system empowers healthcare professionals to intervene before conditions worsen, thus improving patient outcomes. The integration of AI and ML allows for the continuous monitoring and analysis of patient data, providing personalized insights tailored to each individual’s health status.

Ultimately, MEDPREDICT seeks to enhance the quality of healthcare by transforming raw patient data into actionable intelligence. This technology-driven approach promises to make healthcare more efficient, timely, and effective, benefiting both patients and healthcare providers alike.

PROBLEM STATEMENT

Predicting and Visualizing Patient Health Conditions Based on Vital Data

Doctors work hard to keep us healthy, but it's tough to catch every potential problem early, especially when relying only on individual vital signs. Often, health issues become serious before they're easily noticeable. We need a way to look deeper into those numbers, to find hidden patterns and predict potential problems before they escalate. This project aims to create a system that helps doctors see health risks sooner, ultimately leading to better care and healthier lives for everyone.

Healthcare professionals have always been faced with the challenge of predicting patient health issues based on a variety of factors. While they do their best to diagnose and treat conditions based on clinical data, the process can sometimes be slow or incomplete due to a variety of reasons, including a lack of real-time data or the complexity of predicting certain conditions.

The key challenge that MEDPREDICT addresses is the ability to predict patient health conditions in a more efficient and timely manner by analyzing their vital signs. This system will help detect early signs of conditions like heart problems, respiratory distress, and other critical health issues by using data from patients' heart rate, blood pressure, temperature, and oxygen levels.

METHODOLOGY

Data Collection

The first step in the process is gathering the necessary data. For this project, we'll be collecting vital health data, including heart rate, blood pressure, temperature, respiratory rate, and oxygen saturation. This data can be gathered from wearable devices or directly inputted by medical professionals into patient records.

Data Preprocessing

Once the data is collected, it needs to be cleaned and preprocessed. This step ensures that any inconsistencies, errors, or missing values are addressed. We also perform feature engineering, where we create new variables from the data to enhance the performance of our prediction models.

AI & ML Model Development

We will then use machine learning to build models that can predict potential health conditions based on the processed data. Different algorithms, such as logistic regression, decision trees, and neural networks, will be tested and trained on historical patient data. These models will learn the relationships between vital signs and health conditions, enabling them to predict future health issues.

Model Evaluation

Once the models are built, we will evaluate their performance using metrics such as accuracy, precision, recall, and F1-score. This will help us understand how well the models are performing and whether they can generalize well to new, unseen data.

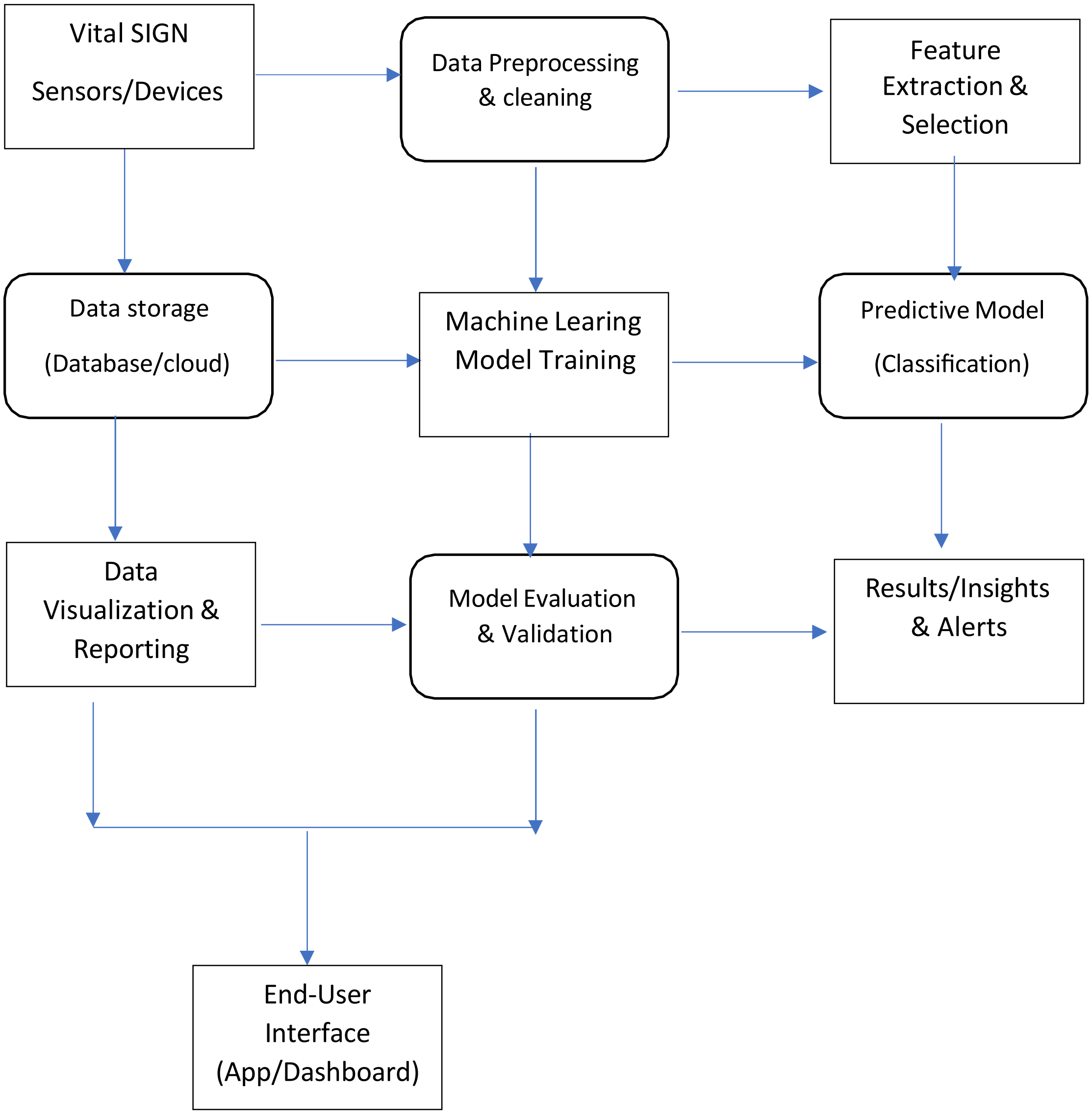
Visualization and Insights

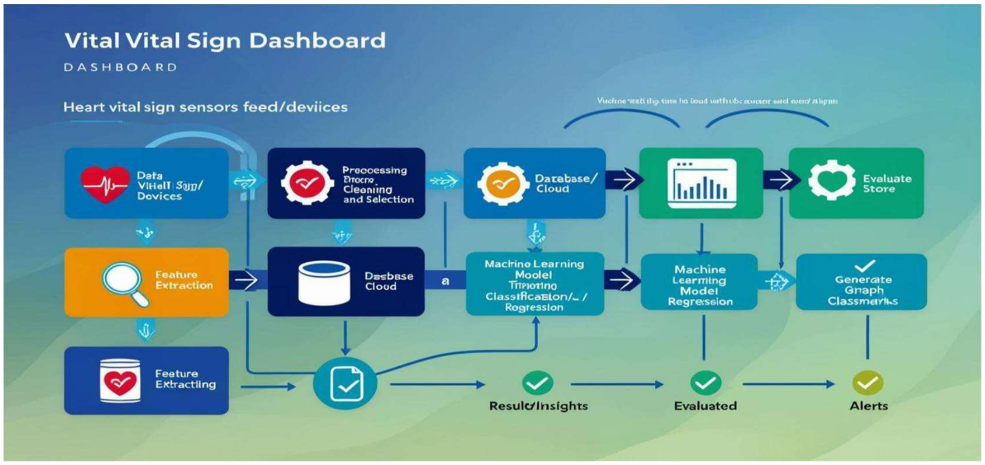
Visualization is a crucial part of this project. Using tools like Python's Matplotlib and Seaborn, we will create graphs and charts that help visualize the predicted health conditions. For healthcare professionals, we will develop interactive dashboards using Tableau or Power BI that will allow them to explore these predictions in real-time.

Deployment and Integration

The final system will be deployed into healthcare environments, where it will integrate with existing platforms and allow healthcare providers to monitor patient health. Special attention will be paid to security and privacy, ensuring that all patient data is kept confidential and compliant with healthcare regulations like HIPAA.

SYSTEM ARCHITECUTRE


Vital SIGN Sensors/Devices
Data Preprocessing & cleaning
Feature Extraction & Selection
Data storage (Database/cloud)
Machine Learing Model Training
Predictive Model (Classification)
Data Visualization & Reporting
Model Evaluation & Validation
Results/Insights & Alerts
End-User Interface (App/Dashboard)


Vital Vital Sign Dashboard
DASHBOARD
Heart vital sign sensors feed/deviices
Data ViHeIl Sjgn/ Devices
Preoocessing Dnocro Cleaning and Selection
Database/ Cloud
Evaluate Store
Feature Extraction
Dasbase Cloud
a
Machino Learning Moolel Tiriorino Classificaeion/ /Rogression
Machine Learming Model Regression
Generate Graph Classmarks
Feature Extractiing
Resulc/Insights
Evaluated
Alerts

SYSTEM REQUIREMENTS

Hardware Requirements:

- Processor: Intel i5/i7 or AMD Ryzen 5/7 (or higher)
- RAM: Minimum 8GB (Recommended: 16GB for better performance)
- Storage: Minimum 100GB free space (SSD recommended for faster processing)
- Internet: Stable high-speed internet connection for cloud-based operations.

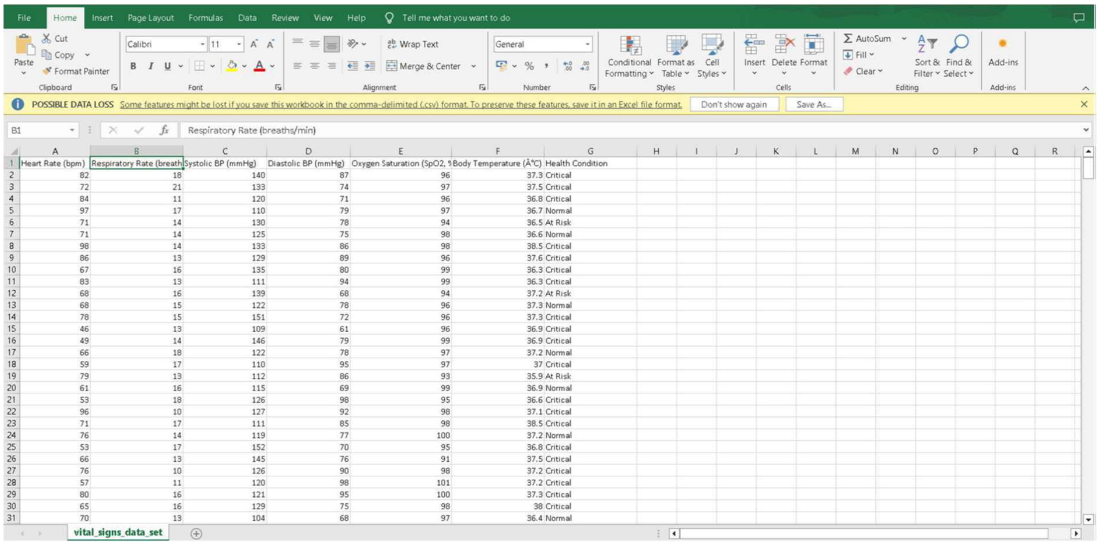
Software Requirements:

- Operating System: Windows 10/11, macOS, or Linux (Ubuntu 20.04+ recommended)
- Programming Environment: Python 3.8+, Jupyter Notebook/Google Colab
- Libraries: Scikit-learn, TensorFlow/PyTorch, Pandas, NumPy, Matplotlib, Seaborn, XGBoost
- Database: MySQL, PostgreSQL, or MongoDB (depending on data storage needs)

Technologies Used

- Programming Languages: Python (for data processing and modeling)
- Libraries and Frameworks:
- Data Processing: Pandas, NumPy, streamlit, joblib
- Machine Learning: Scikit-learn, TensorFlow, PyTorch, XGBoost
- Visualization: Matplotlib, Seaborn, Plotly
- Tools: Jupyter Notebook, Google Colab, Kaggle

Dataset:



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Respiratory Rate (breaths/min)

	A	B	C	D	E	F	G
1	Heart Rate (bpm)	Respiratory Rate (breath	Systolic BP (mmHg)	Diastolic BP (mmHg)	Oxygen Saturation (SpO2, 9	Body Temperature (Â°C)	Health Condition
2	82	18	140	87	96	37.3	Critical
3	72	21	133	74	97	37.5	Critical
4	84	11	120	71	96	36.8	Critical
5	97	17	110	79	97	36.7	Normal
6	71	14	130	78	94	36.5	At Risk
7	71	14	125	75	98	36.6	Normal
8	98	14	133	86	98	38.5	Critical
9	86	13	129	89	96	37.6	Critical
10	67	16	135	80	99	36.3	Critical
11	83	13	111	94	99	36.3	Critical
12	68	16	139	68	94	37.2	At Risk
13	68	15	122	78	96	37.3	Normal
14	78	15	151	72	96	37.3	Critical
15	46	13	109	61	96	36.9	Critical
16	49	14	146	79	99	36.9	Critical
17	66	18	122	78	97	37.2	Normal
18	59	17	110	95	97	37	Critical
19	79	13	112	86	93	35.9	At Risk
20	61	16	115	69	99	36.9	Normal
21	53	18	126	98	95	36.6	Critical
22	96	10	127	92	98	37.1	Critical
23	71	17	111	85	98	38.5	Critical
24	76	14	119	77	100	37.2	Normal
25	53	17	152	70	95	36.8	Critical
26	66	13	145	76	91	37.5	Critical
27	76	10	126	90	98	37.2	Critical
28	57	11	120	98	101	37.2	Critical
29	80	16	121	95	100	37.3	Critical
30	65	16	129	75	98	38	Critical
31	70	13	104	68	97	36.4	Normal

Front End

```
 appimport streamlit as st
import numpy as np
import joblib

# Load Model and Label Encoder with Error Handling
try:
    model = joblib.load("health_predictor_model (7).pkl")
    label_encoder = joblib.load("label_encoder (4).pkl")
except Exception as e:
    st.error(f"Error loading model or encoder: {e}")
    model, label_encoder = None, None

# Function to Predict Health Condition
def predict_health_condition(spo2, temp, resp_rate, heart_rate, sys_bp, dia_bp):
    if model is None or label_encoder is None:
        st.error("Model or encoder not loaded properly.")
        return "Error"

    try:
        input_data = np.array([[spo2, temp, resp_rate, heart_rate, sys_bp, dia_bp]])
        prediction = model.predict(input_data)[0]
        condition = label_encoder.inverse_transform([prediction])[0]
        return condition
    except Exception as e:
        st.error(f"Prediction Error: {e}")
        return "Error"

# Streamlit UI
st.title("Health Condition Prediction")
st.write("Enter your health parameters below:")


# Input fields with default values, min and max constraints

heart_rate = st.number_input("Heart Rate (bpm)", min_value=40, max_value=200,
value=70)

resp_rate = st.number_input("Respiratory Rate (breaths/min)", min_value=12,
max_value=40, value=16)

sys_bp = st.number_input("Systolic BP (mmHg)", min_value=90, max_value=200,
value=120)

dia_bp = st.number_input("Diastolic BP (mmHg)", min_value=60, max_value=120,
value=80)

spo2 = st.number_input("Oxygen Saturation (SpO2, %)", min_value=50,
max_value=100, value=98)

temp = st.number_input("Body Temperature (°C)", min_value=35, max_value=42,
value=37)


# Prediction when button is pressed

if st.button("Predict"):

    condition = predict_health_condition(spo2, temp, resp_rate, heart_rate, sys_bp,
dia_bp)


    st.subheader("Prediction Result:")

    if condition == "Normal":

        st.success("Your health condition is **Normal**. ✅")

    elif condition == "At Risk":

        st.warning("Your health condition is **At Risk**. ⚠️")

    elif condition == "Critical":

        st.error("Your health condition is **Critical**. 🚨

    else:

        st.error("Error in prediction. Please check input values.")


    # Suggestions based on condition

    st.subheader("Health Suggestions:")
```

```
suggestions = {
    "Normal": [
        ✅ Maintain a balanced diet with fruits, vegetables, and whole grains.",
        ✅ Exercise regularly (at least 30 minutes a day, 5 days a week).",
        "✅ Get 7-9 hours of sleep.",
        '✅ Monitor your vital signs regularly.",
        ✅ Schedule annual check-ups with your healthcare provider."
    ],
    "At Risk": [
        ✅ Maintain a balanced diet with fruits, vegetables, and whole grains.",
        ✅ Exercise regularly (at least 30 minutes a day, 5 days a week).",
        "✅ Get 7-9 hours of sleep.",
        '✅ Monitor your vital signs regularly.",
        ✅ Schedule annual check-ups with your healthcare provider."
    ],
    "Critical": [
        ✅ Seek immediate medical attention if experiencing severe symptoms.",
        ✅ Follow prescribed medications and treatment plans strictly.",
        ✅ Limit physical activity until cleared by a doctor.",
        ✅ Consume light, easily digestible foods and maintain hydration.",
        ✅ Ensure someone is available to assist and monitor your health."
    ]
}

for suggestion in suggestions.get(condition, []):
    st.write(suggestion)
```

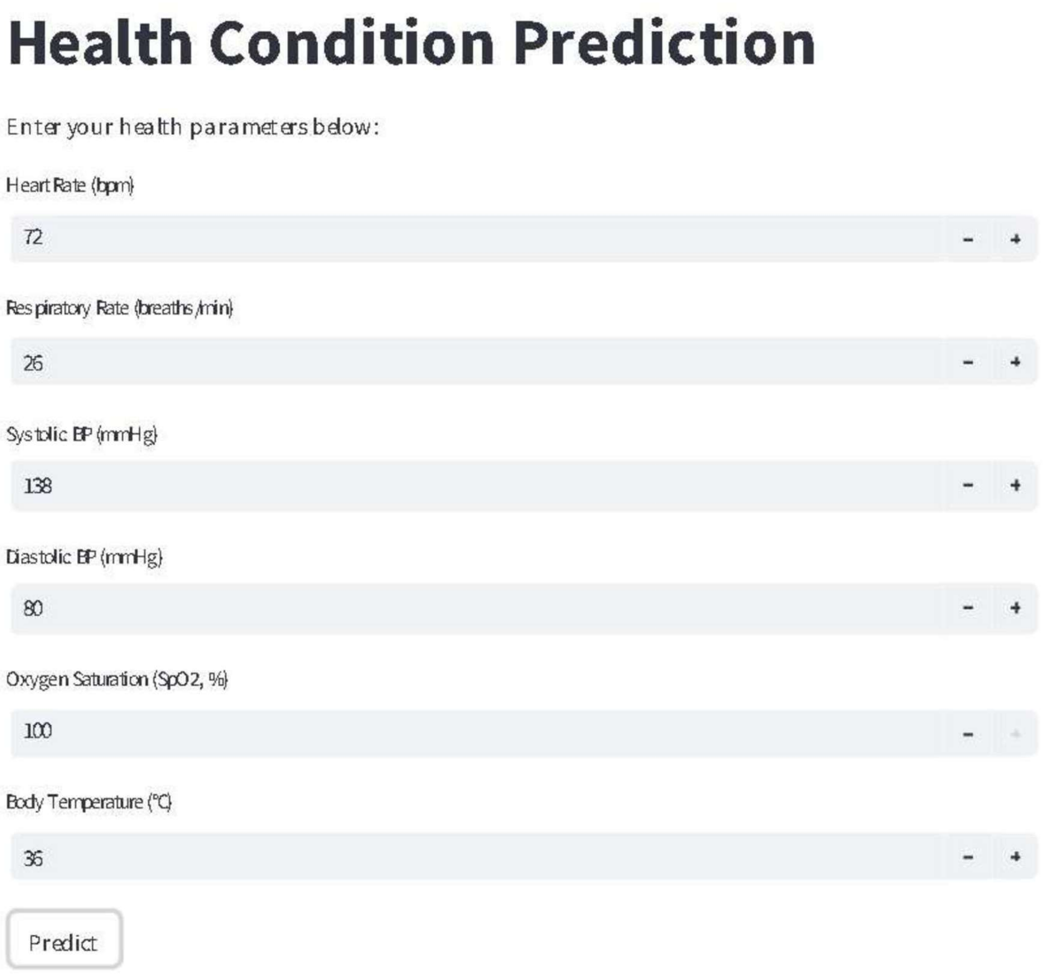
Backend:

```
#import libaries
import pandas as pd
import numpy as np
#insert dataset
df = pd.read_csv("vital_signs_data_set.csv")

print(df.head())
# Check for missing values
print(df.isnull().sum())
 # Check data types
print(df.dtypes)
#convert datatype to int
df["Body Temperature (°C)"] = df["Body Temperature (°C)"].astype(int)
df["Oxygen Saturation (SpO2, %)"] = df["Oxygen Saturation (SpO2, %)"].astype(int)
# Mapping the 'Health Condition' to integers
health_condition_map = {'Normal': 0, 'At Risk': 1, 'Critical': 2}
df["Health Condition"] = df["Health Condition"].map(health_condition_map)
# Check data types
print(df.dtypes)
print(df.describe())
import joblib
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
# Function to classify health condition
def classify_condition(row):
    if row["Oxygen Saturation (SpO2, %)"] < 92 or row["Heart Rate (bpm)"] > 120 or
row["Systolic BP (mmHg)"] > 180:
        return "Critical"
    elif row["Systolic BP (mmHg)"] > 140 or row["Diastolic BP (mmHg)"] > 90 or row["Heart Rate (bpm)"] > 100:
    else:
        return "Normal"
# Apply classification
df["Health Condition"] = df.apply(classify_condition, axis=1)
# Features & Labels
X = df.drop("Health Condition", axis=1)
y = df["Health Condition"]


# Encode labels
le = LabelEncoder()
y = le.fit_transform(y)
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
# Evaluate Model
accuracy = accuracy_score(y_test, model.predict(X_test))
print(f"Model Accuracy: {accuracy * 100:.2f}%")
# Save Model
joblib.dump(model, 'health_predictor_model.pkl')
joblib.dump(le, 'label_encoder.pkl')
print("Model and label encoder saved successfully.")
```

OUTPUT:



Health Condition Prediction

Enter your health parameters below:

Heart Rate (bpm)

72

Respiratory Rate (breaths/min)

26

Systolic BP (mmHg)

138

Diastolic BP (mmHg)

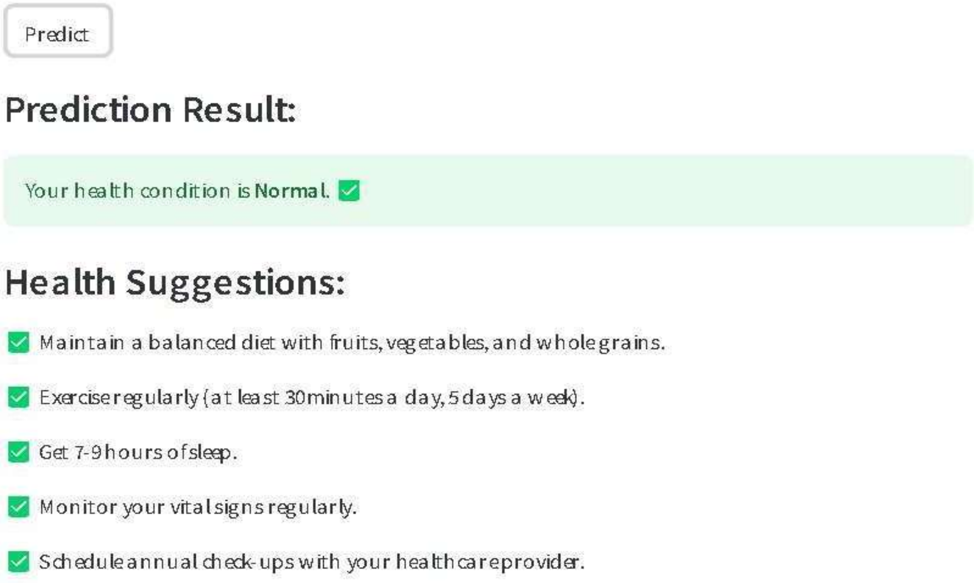
80

Oxygen Saturation (SpO2, %)

100

Body Temperature (°C)

36



Prediction Result:

Your health condition is **Normal**. ✅

Health Suggestions:

✅ Maintain a balanced diet with fruits, vegetables, and whole grains.

✅ Exercise regularly (at least 30 minutes a day, 5 days a week).

✅ Get 7-9 hours of sleep.

✅ Monitor your vital signs regularly.

✅ Schedule annual check-ups with your healthcare provider.

RESULTS

The primary goal of **MEDPREDICT** is to provide accurate predictions and actionable insights about a patient's health, and the expected results from this project are as follows:

- **Accurate Predictions**: The system will be able to predict various health conditions such as heart attacks, stroke, and respiratory distress, using the real-time data from patient vital signs.
- **Visual Insights**: By turning data into visual insights, healthcare providers will be able to see trends in patient health over time. This makes it easier to understand how a patient's condition is progressing or improving.
- **Early Warning Alerts**: One of the most important features is the ability to send early warnings when a patient's vital signs show signs of potential health risks. This gives medical teams a head start in taking preventative measures.
- **Informed Healthcare Decisions**: By providing accurate predictions and real-time insights, **MEDPREDICT** will help healthcare professionals make informed decisions and act swiftly to prevent serious health complications.

FUTURE SCOPE:

- **Integration with IoT Devices:** Connecting with real-time IoT-enabled medical devices for continuous health monitoring.
- **Federated Learning Implementation:** Enabling privacy-preserving data analysis by training models across decentralized data sources.
- **Expansion to More Health Conditions:** Extending the model to predict and analyze a wider range of diseases and health anomalies.
- **Personalized Health Insights:** Enhancing AI-driven recommendations for lifestyle modifications, diet plans, and medication adherence.
- **Cloud-Based and Mobile Integration:** Developing cloud-supported mobile applications to allow remote patient monitoring and telemedicine applications.
- **Collaboration with Healthcare Institutions:** Partnering with hospitals and research institutions to refine predictive models and validate real-world applications.
- **AI-Driven Treatment Optimization:** Implementing reinforcement learning to assist doctors in prescribing personalized treatment plans.
- **Global Health Monitoring Systems:** Creating a framework for large-scale health surveillance to detect and prevent disease outbreaks.

CONCLUSIONS

In conclusion, **MEDPREDICT** has the potential to transform how healthcare providers monitor and manage patient health. By using AI and ML, we can predict potential health conditions based on patients' vital data, helping doctors and nurses to intervene before issues escalate. Real-time predictions, coupled with visual insights, can improve decision-making, optimize healthcare resources, and ultimately save lives.

The project represents a significant step towards the integration of advanced technologies in healthcare, not only making the process of diagnosis more efficient but also more proactive. With early intervention and personalized health insights, **MEDPREDICT** aims to offer tangible benefits to patients and healthcare providers alike.