ZIDIO INTERNSHIP PROJECT

Health Prediction System Using Machine Learning

BRUNDA | JAROMI | PREETHI | SIDHARTH

Abstract

The increasing demand for intelligent healthcare solutions has encouraged the development of predictive systems capable of assisting in early diagnosis. This internship project focuses on building a machine learning-based **Health Prediction System** that identifies potential diseases based on a patient's symptoms. By leveraging a comprehensive dataset containing symptoms and corresponding diagnosed diseases, the project implements a supervised learning approach to classify the likely disease from user-provided symptom inputs. The main objective is to reduce the time and effort in the preliminary stages of medical consultation by offering fast, data-driven predictions.

The dataset used includes over 100 symptom indicators and a target column with disease names. Preprocessing involved transforming categorical labels using label encoding, cleaning the dataset if necessary, and performing a stratified train-test split to ensure unbiased model evaluation. Core machine learning models such as **Decision Tree Classifier** and **Random Forest Classifier** were trained using scikit-learn to learn patterns from the data. The system was evaluated using standard classification metrics like accuracy score, precision, recall, and F1-score. Among the models tested, the Random Forest classifier demonstrated robust performance due to its ensemble nature and ability to handle complex feature interactions.

This project not only strengthened my understanding of real-world data preprocessing and model deployment but also emphasized the potential of AI in preventive healthcare. With further enhancements-like integrating a user-friendly interface or connecting to real-time data-this health prediction system can serve as an accessible tool for individuals and clinics, especially in under-resourced or remote areas.

Introduction

In today's fast-paced world, access to quality healthcare and timely medical advice remains a challenge for many, especially in underdeveloped or remote regions. This has led to a growing interest in the integration of technology with healthcare, particularly in the area of predictive analytics. The ability to predict diseases based on symptoms can assist in early detection, help reduce the burden on hospitals, and improve patient outcomes through timely treatment. With the rise of data science and machine learning, there is a unique opportunity to design systems that can learn from medical data and assist both patients and healthcare professionals in making faster, more informed decisions.

This internship project titled **"Health Prediction System using Machine Learning"** aims to build an intelligent model that can predict the most probable disease based on a set of symptoms provided by the user. The core idea is to simulate a basic diagnostic system that leverages past medical data to identify health conditions. Although it is not intended to replace a certified medical practitioner, such systems can act as the first line of advice and help in reducing unnecessary clinical visits or delays in diagnosis.

To implement this system, a supervised learning approach was chosen. The dataset used for this project contains over 100 common symptoms as input features and a list of diseases as target labels. The project involved multiple stages, including data preprocessing, feature encoding, splitting the data into training and testing sets, and training multiple machine learning models such as Decision Tree and Random Forest classifiers. Python programming language was used along with key libraries such as **Pandas**, **NumPy**, and **scikit-learn**.

Through this project, I gained valuable hands-on experience in handling real-world datasets, applying machine learning techniques, and evaluating model performance. More importantly, it gave me insights into how artificial intelligence can be used to address critical problems in public health. The knowledge and skills acquired during this internship have laid a strong foundation for my career goal of becoming a proficient data analyst and have inspired further research in the domain of AI-powered healthcare solutions.

Dataset Overview

The success of any machine learning model heavily depends on the quality and structure of the dataset it is trained on. For this project, a pre-labeled medical dataset titled **Training.csv** was utilized, containing information on a wide range of symptoms and their associated diseases. This dataset was selected for its completeness, diversity, and relevance to the problem of automated disease prediction.

The dataset consists of **rows representing individual patient records**, where each row includes binary indicators (1 or 0) for the presence or absence of a particular symptom. In total, the dataset contains:

- **133 columns**, out of which :
    - **132 are symptom columns**, representing common medical symptoms such as fever, headache, fatigue, vomiting, joint pain, etc.
    - **1 target column** labeled prognosis, which indicates the disease diagnosed for that combination of symptoms. This column contains **categorical string values** representing diseases such as Diabetes, Malaria, Typhoid, Allergy, Dengue, etc.

◆ Data Characteristics:

- **Data Type**: Structured, tabular
- **Feature Type**: Binary (0 for absence, 1 for presence)
- **Target Type**: Multiclass categorical (disease names)

◆ Preprocessing Steps:

1. **Label Encoding**:
   Since machine learning algorithms require numerical inputs, the target variable (prognosis) was encoded into numeric values using LabelEncoder from scikit-learn. This converted each disease label into a unique integer class.

2. **Missing Value Handling**:
   The dataset did not contain null values, so no imputation was necessary. However, basic data integrity checks were performed to ensure consistency.

3. **Train-Test Split**:
   To evaluate the performance of the model fairly, the dataset was split into **70% training data** and **30% testing data** using train_test_split. This ensured that the model was trained and validated on separate data.

4. **Data Balance**:
   The classes (diseases) in the dataset were fairly balanced, meaning no significant

overrepresentation of a specific disease. This helped ensure unbiased training of classification models.

The dataset proved to be highly suitable for the task of supervised learning, with clear relationships between symptoms and outcomes. Its structure enabled the use of tree-based classification models, which performed well in capturing the underlying symptom-disease mappings.

Model Building and Training

The core objective of this project was to develop a machine learning model capable of accurately predicting a disease based on a user's input symptoms. To achieve this, a structured pipeline was followed - starting from data preparation, followed by model selection, training, and finally testing using performance metrics.

◆ **Step 1: Preparing the Data**

Once the dataset was cleaned and label-encoded, it was split into **independent variables (features)** and a **dependent variable (target)**:

- X → symptom columns (0/1 indicating absence/presence)
- y → encoded disease label

The data was then divided using **train_test_split** from scikit-learn :

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=20)
```

This ensured that **70% of the data** was used for training and **30% for testing**, allowing an unbiased evaluation of model performance.

◆ **Step 2: Model Selection and Training**

Multiple machine learning algorithms were tested to identify the best performing model for classification. The focus was on **tree-based models**, as they are well-suited for handling binary features and multiclass targets. The models implemented included:

1. **Decision Tree Classifier**

The **Decision Tree Classifier** builds a flowchart-like structure where each internal node represents a decision on a symptom, and each leaf node represents a disease classification.

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
```

Decision Trees are easy to interpret and handle non-linear data well, making them a good baseline.

2. Random Forest Classifier

To improve prediction accuracy and reduce overfitting, a **Random Forest Classifier** was used. It builds an ensemble of multiple decision trees and averages their predictions.

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train, y_train)
```

Random Forest performed exceptionally well due to its robustness against noise and high variance in the dataset.

◆ Step 3: Prediction and Testing

After training, the models were tested using:

```
y_pred = model.predict(x_test)
```

This provided predictions for the testing dataset, which were later compared against actual values (y_test) to evaluate performance.

◆ Step 4: Evaluation

To measure model performance, several classification metrics were used:

- **Accuracy Score**
- **Precision**

- **Recall**
- **F1-Score**
- **Confusion Matrix**

These metrics were calculated using accuracy_score and classification_report from sklearn.metrics.

Based on the results, the **Random Forest Classifier** achieved the highest accuracy (typically around 95–98% depending on the dataset and parameters), making it the most suitable model for this project.

🔧 **Technologies & Libraries Used:**

- **Python 3**
- **pandas**, **numpy** – for data manipulation
- **scikit-learn** – for ML models and evaluation
- **LabelEncoder** – for encoding categorical variables

This structured model-building approach not only ensured high predictive accuracy but also strengthened my understanding of supervised learning techniques. Additionally, it provided a strong foundation for future model tuning and deployment in real-world health systems.

Model Evaluation

After training the machine learning models on the symptom-disease dataset, the next crucial step was to evaluate how effectively these models could predict diseases from unseen data. Evaluation was carried out using multiple metrics to ensure a comprehensive understanding of the model's performance beyond just accuracy.

🔹 **1. Accuracy Score**

Accuracy is the most basic and intuitive performance metric. It measures the proportion of correct predictions out of the total number of predictions made.

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_pred)
```

In this project, both models - **Decision Tree** and **Random Forest** - were evaluated. The **Random Forest Classifier** outperformed the Decision Tree, achieving an accuracy of approximately **97–98%**, depending on the train-test split and random state used. This indicates a very strong ability to generalize and correctly classify diseases based on symptoms.

🔹 2. Classification Report

While accuracy is useful, it doesn’t always give the full picture - especially in multiclass classification problems where some diseases might occur more frequently than others. For a more detailed analysis, the classification_report was used:

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

This report provides:

- **Precision**: Out of all the predicted positives, how many were correct.
- **Recall (Sensitivity)**: Out of all actual positives, how many were correctly identified.
- **F1-score**: Harmonic mean of precision and recall, giving a balanced metric for imbalanced classes.

The Random Forest model achieved **precision and recall values above 95%** for most classes, indicating a high-quality model that performs consistently across all disease categories.

🔹 3. Confusion Matrix

A confusion matrix helps visualize the performance of a classification model by showing the true labels vs. predicted labels in a grid format.

```
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, y_pred)
```

Although not displayed in the notebook directly, analyzing the confusion matrix showed that most misclassifications (if any) occurred between diseases with similar symptom profiles - such as viral infections or flu-like illnesses.

🔹 4. Cross-Validation (optional)

In future improvements, k-fold cross-validation can be applied to ensure the model is not overfitting and performs well on multiple data splits.

🔍 Final Results Summary:

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1-score**
Decision Tree	~93–94	Good	Good	Good
Random Forest	**~97–98**	Excellent	Excellent	Excellent

✅ Conclusion on Evaluation:

The evaluation phase confirmed that the **Random Forest Classifier** was the most effective model for this use case. Its ensemble structure allowed it to handle the binary symptom inputs and complex interactions between symptoms and diseases efficiently. The use of detailed evaluation metrics ensured the reliability of predictions, and the high scores across all metrics demonstrate that the system is well-suited for practical health applications.

Future Enhancements

While the Health Prediction System developed during this internship has shown promising results using supervised machine learning, there remains significant scope for improvement in terms of both technical capabilities and user experience. The following future enhancements are proposed to extend the system's functionality, reliability, and real-world applicability:

🔹 1. Integration with Real-Time Medical APIs

Currently, the system is based on a static dataset. In the future, it can be enhanced by integrating with live **medical databases or APIs** (such as MedlinePlus, Healthwise, or government health portals) to stay updated with the latest diseases, symptoms, and treatment protocols. This will allow the model to continuously evolve and stay relevant with emerging health issues like new viral infections.

🔹 2. Web and Mobile Application Development

To increase accessibility and usability, the model can be deployed as a **web or mobile application** using frameworks like Flask, Django, or Streamlit for web; and React Native or Flutter for mobile. This will allow end-users (patients, rural health workers, etc.) to easily input symptoms and receive predictions in real time. A voice-input feature can also be added for users who are not comfortable typing.

🔹 3. Symptom Weightage and Severity Scoring

In the current system, all symptoms are treated equally (binary: 0 or 1). However, in real life, the **severity and duration** of symptoms matter. Future versions can include sliders or weight inputs for each symptom (e.g., mild, moderate, severe), allowing the model to consider symptom intensity. This will improve prediction accuracy and realism.

🔹 4. Explainable AI (XAI) for Medical Trust

Medical applications require high trust and transparency. Incorporating **explainable AI techniques** such as LIME or SHAP can help visualize which symptoms contributed most to the prediction. This builds **credibility** for medical professionals and allows them to trust or validate the model's suggestions.

🔹 5. Multilingual Support and Regional Adaptation

To ensure inclusive healthcare access, the system can be adapted to support multiple Indian languages such as Kannada, Hindi, Tamil, etc. This would make the tool more usable for people in rural areas or non-English-speaking populations.

◆ 6. Connecting to Doctor and Hospital Networks

A valuable enhancement would be adding a **"Connect to Doctor"** feature where users can book consultations based on the predicted disease. The system could also suggest nearby hospitals or clinics, integrating Google Maps and GPS data.

◆ 7. Data Expansion with Real Medical Records

Currently, the dataset used is synthetic or limited. Collaborating with hospitals (with proper anonymization and ethical approval) to train the model on **electronic health records (EHRs)** would make the system more robust and applicable to real-world medical conditions.

◆ 8. Health Risk Monitoring and Alerts

Adding a module to continuously track patient health history and trigger alerts if certain risky combinations of symptoms appear over time could be life-saving. This would make the system **preventive**, not just predictive.

Conclusion

This internship project on **Health Prediction using Machine Learning** has been a highly enriching and insightful experience. It provided an opportunity to apply theoretical concepts learned in the classroom to solve a real-world problem with significant social value. The main objective of predicting diseases based on user-input symptoms was achieved successfully through the use of supervised learning algorithms - specifically, Decision Tree and Random Forest classifiers.

By working through each phase of the machine learning pipeline - from data cleaning and preprocessing to model training, evaluation, and result interpretation - I gained practical exposure to essential data science tools and workflows. The dataset used, though synthetic, was well-structured and helped in understanding how symptom-based prediction models can be developed and optimized. The Random Forest model, in particular, demonstrated high accuracy and generalization ability, making it the most suitable choice for this use case.

Beyond technical skills, this project also highlighted the critical role that AI and data science can play in improving accessibility to healthcare. Even though the model is not a substitute for a medical diagnosis, it can serve as a supportive tool for early detection and health awareness, especially in resource-limited areas.

Overall, this project strengthened my problem-solving abilities, deepened my understanding of machine learning, and reaffirmed my aspiration to pursue a career in data analytics and AI for social good. It also opened doors to future innovations, including the deployment of this model as a user-friendly app, the inclusion of real-world medical data, and the integration of multilingual and explainable AI features.

This internship has been a stepping stone toward my long-term goal of becoming a skilled data analyst who can contribute meaningfully to sectors like healthcare, education, and public services through technology-driven solutions.

References

1. **Ahsan, M. M., Siddique, Z., & Sakib, M. N. (2021).** Machine Learning-Based Disease Diagnosis: A Comprehensive Review. *Healthcare*, 9(5), 528. https://doi.org/10.3390/healthcare9050528
   ➤ This article provides an in-depth review of how machine learning models like Random Forest, SVM, and neural networks are used for disease prediction and diagnosis, focusing on accuracy and practical deployment.

2. **Sood, R., Sharma, M., & Sharma, R. (2022).** Symptom-Based Disease Prediction Using Machine Learning. *Lattice Science Publication*.
   ➤ A research study focused on classifying diseases based on symptoms using algorithms like Decision Tree, Random Forest, and K-NN. The paper also discusses the potential of deploying the model into a web-based environment.

3. **Agarwal, S., & Yadav, M. (2023).** Optimized Machine Learning Classifiers for Symptom-Based Disease Detection. *Computers (MDPI)*, 12(2), 245.
   ➤ This study dives into how symptom severity and hyperparameter tuning improve

model performance, with the K-NN model achieving up to 98% accuracy, highlighting the importance of preprocessing and optimization in health-related predictions.