Disease Prediction from symptoms

Machine learning project

progress report 1

Team members :

1. Sama shalabi
2. Laith asbeh

methods we are using

1. Support vector machine
2. Random Forest
3. Gaussian NB

Datasets

We got a cleaned dataset from www.codespeedy.com and used it in the project dataset link.

Abstract:

This project aims to develop a machine learning-based system for predicting infectious diseases based on user-reported symptoms.
By leveraging three algorithms—Decision Tree, Random Forest, and NaiveBayes—the system provides automated and accurate predictions to support healthcare professionals in disease diagnosis.
Preliminary results indicate high accuracy.

Introduction:

Accurate and timely disease diagnosis is critical in managing infectious diseases effectively. This project aims to harness the power of machine learning to build a predictive system using Decision Tree, Random Forest, and NaiveBayes algorithms.

The system is designed to assist healthcare professionals by automating the diagnostic process based on symptoms, ultimately improving healthcare outcomes.
The significance lies in the potential to provide a reliable, scalable, and interpretable tool for early disease detection.

Detailed Methodology:

1. Data Collection and Preprocessing

Source of Data: A dataset containing symptoms as input features and corresponding diseases as target labels was acquired.
Data Cleaning: we checked for missing values and there is no missing values
Feature Engineering: Symptom data was encoded numerically using techniques like one-hot encoding for compatibility with machine learning models.
Data Splitting: The dataset was divided into training (80%) and testing (20%) subsets to evaluate model performance.

2. Model Training

Decision Tree: A simple yet interpretable algorithm was trained to create a tree-based predictive model.
Random Forest: builds multiple decision trees during training and combines their outputs (by majority vote for classification or averaging for regression) to make predictions.
NaiveBayes: This probabilistic classifier was trained to handle high-dimensional data efficiently.
Support vector machine: finding the best boundary (or hyperplane) that separates data points of different classes in a dataset.

3. Model Evaluation

Performance metrics such as accuracy, precision, recall, and F1-score were used to compare model effectiveness.
Cross-validation techniques ensured the reliability and stability of results across various data splits.

4. Implementation

A prototype system was developed to accept user symptoms and output the most likely disease using the trained models.

Preliminary Results:

- About the dataset :

We used the dataset from www.codespeedy.com, and the dataset was already cleaned and ready, the training dataset includes 4921 samples with 129 features (symptoms) and 41 labels (disease), and testing data with 43 samples with 129 features (symptoms) and 41 labels (disease)

- Data analysis :

We Checked whether the dataset was balanced or not.
Then we used 1000 and split the training data for training and testing data for training and evaluating the models : Train: (800, 129), (800, 1), Test: (200, 129), (200, 1)

We executed cross-validation on the data with the models: Decision tree, Random forest, and Naive Bayes then we trained the models with all 129 features and 1000 samples.
Then we calculated the evaluation measures and got the following results :

- DecisionTree: Accuracy: 1.0, Precision: 1.0, Recall: 1.0, F1 score: 1.0 with mean: 1.0
- random forest : Accuracy: 1.0, Precision: 1.0, Recall: 1.0, F1 score: 1.0 with mean: 1.0
- NaiveBayes : Accuracy: 1.0, Precision: 1.0, Recall: 1.0, F1 score: 1.0 with mean: 1.0

Second time we tried to execute a cross-validation on 2500 samples => Train: (2000, 129), (2000, 1), Test: (500, 129), (500, 1)

- Then we calculated the evaluation measures and got the following results :
  - DecisionTree: Accuracy: 1.0, Precision: 1.0, Recall : 1.0,F1 score: 1.0 with mean: 1.0
  - random forest : Accuracy: 1.0, Precision: 1.0,Recall: 1.0,F1 score: 1.0 with mean: 1.0
  - NaiveBayes : Accuracy: 1.0, Precision: 1.0,Recall: 1.0,F1 score: 1.0 with mean : 1.0

But got a different class prediction from each model

The third time we did the same steps but with all 129 features with all samples => Train: (3936, 129), (3936, 1) , Test: (984, 129), (984, 1)

And got the same results

- the evaluation measures and got the following results :

- DecisionTree: Accuracy : 1.0 , Precision : 1.0 ,Recall : 1.0 ,F1 score : 1.0 with mean : 1.0
- randomforest : Accuracy: 1.0, Precision: 1.0 ,Recall: 1.0 ,F1 score: 1.0 with mean : 1.0
- NaiveBayes : Accuracy: 1.0, Precision: 1.0 ,Recall: 1.0 ,F1 score: 1.0 with mean : 1.0

But got a different class prediction from each model.

Challenges:

finally from the previous results we concluded that all the models are overfitting the data and are not predicting the labels correctly and their measures are resulting in the max value = 1, after looking for the reasons we discovered that the problem is the dataset its selfs, that the majority of the samples are duplicated multiple time, for the whole 3936 samples there are 43 unique samples for each label

Next Steps:

We will fix the data by deleting the duplicates, do the cross-validation again, and evaluate the model. In addition, we will look for a better dataset so that we can evaluate the models and get better predictions.