Information Retrieval and Web Search Engines

Name-Surname: Abdulkader Saoud

Student 1 No: 21011941

Name-Surname: Esam Halimeh

Student 2 No: 20011908

Instructor: Prof. Dr. Mehmet Sıddık Aktaş

Motivation:

Nowadays, machine learning has begun to impact all areas related to healthcare. Doctors are required to diagnose each incoming patient. Having this process performed by a machine can be more effective. Training a machine on historical datasets using machine learning algorithms and then using it for the diagnosis of specific diseases can be faster and more time efficient. This allows for the possibility of starting treatment earlier, thereby increasing the potential for rapid patient healing. In today's rapidly evolving healthcare aspect, using machine learning in this manner ultimately can significantly contribute to enhance overall outcomes in the healthcare sector. For this reason, we have created a disease prediction model that assists doctors. Within the provided dataset, various diseases and their symptoms are present. We have used **Decision Tree**, **Naïve Bayes**, and **k-Nearest Neighbors** (kNN) supervised machine learning algorithms for the most suitable and accurate results. Also, we have used **python** programming language to implement this project.

Disease Dataset Representation:

The dataset utilized for this project is taken from https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset/data.

Disease	Symptom 1	Symptom 2	Symptom 3	Symptom 4
Fungal infection	itching	Skin rash	Nodal skin eruptions	Dyschromic patches
Allergy	Continuous sneezing	shivering	chills	Watering from eyes

It provides disease entries and associated symptoms. In the presented example, the first column denotes the disease name, while the subsequent columns represent various symptoms. To address redundancy, redundant lines were removed during preprocessing, reducing the initial count from 4921 to 305, decreasing the risk of overfitting. To encode the disease and symptom names, a one-hot encoding approach was adopted. While an alternative method involving numerical representation of each symptom was considered, concerns about

potential model confusion due to mathematical relationships prompted the selection of one-hot encoding. This approach assigns a distinct column for each symptom, representing its presence with binary values (1 for existence, 0 for absence). Despite the resulting increase in data size, this methodology enhances the model's clarity.

Disease	Abdominal pain	Abnormal menstruation	Acidity
15	0	0	0
16	0	0	1

The ending result looks like this. The processed data now comprises 131 symptoms and 41 diseases, providing a structured and encoded foundation.

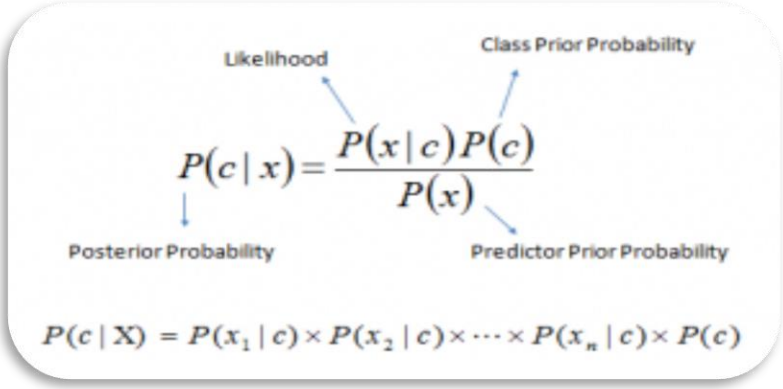
I. Decision Tree Algorithm:

Decision tree is one of the powerful tools of supervised learning algorithms used for both classification and regression tasks. The fundamental concept involves recursively partitioning the dataset based on the most crucial features, leading to a tree-like structure where each internal node represents a decision based on a feature, and each leaf node corresponds to a predicted outcome.
We used the DecisionTreeClassifier from scikit-learn which has these parameters:

- **criterion:** This parameter determines the function to measure the quality of a split. "Gini" and "entropy" are the supported criteria. Gini impurity is used for Gini, and information gain is used for entropy. We chose to use Gini.
- **max_depth:** The maximum depth of the tree. Increasing may cause overfitting. We chose 50.
- **min_samples_split:** The minimum number of samples required to split an internal node. We chose two.
- **min_samples_leaf:** The minimum number of samples required to be at a leaf node. 1.
- **max_features:** The number of features to consider when looking for the best split. 16.

II. Naïve Bayes Classification Algorithm:

The Naïve Bayes classifier algorithm is a supervised machine learning technique used for classification tasks. It based on Bayes theorem and assume that there are no dependencies among attributes. It is made to simplify the computations involved and, hence is called "naïve". This algorithm computes the probability of a data point belonging to each class and assigns it to the class with the highest probability. Naïve Bayes is known for its simplicity, efficiency, and effectiveness in handling high-dimensional data. This algorithm has several variants to accommodate different types of data and assumptions about the distribution of attributes. The dataset used in this project is contains boolean data (0's and 1's). So that **Bernoulli Naïve Bayes** algorithm have been used. The binomial model is useful if dataset features are boolean.



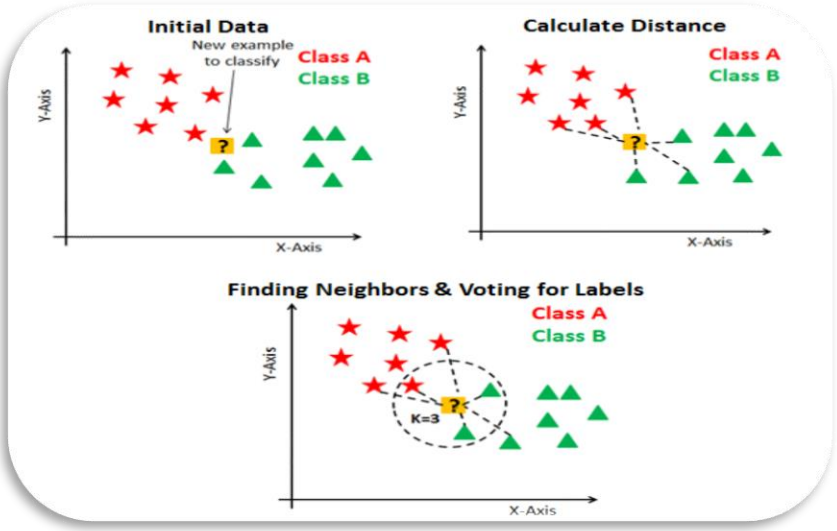
- P(c|x) is the **posterior probability** of class (c) given predictor (x,attribute).
- P(c) is the **prior probability** of class.
- P(x|c) is the **likelihood** which is the probability of attribute given class.
- P(x) is the **prior probability** of the attribute. It can be eliminated when comparing because it is same for all classes.

We have 41 classes in the dataset and X represents the symptoms. $P(x|c_i)$ will be computed for each class. Due to the initial assumption which says attributes are independent, $P(X|c_i) = p(x_1|c_i) * p(x_2|c_i) \ldots * p(x_m|c_i)$. Given new data point x, classification is to derive the maximum posteriori probability $P(c_i|x)$. Data point belongs to the class that has maximum posteriori probability.

In algoritm implementation, we have used BernoulliNB from scikit-learn library.

III. k-Nearest Neighbors Algorithm:

k-Nearest Neighbors (kNN) is a type of supervised machine learning algorithm used for both regression and classification. kNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the k number of points which is closest to the test data. The kNN algorithm calculates the probability of the test data belonging to the classes of k training data and class holds the highest frequent will be selected. kNN does not build model from the training data instead it stores the dataset. When faced with new data at the classification time it performs actions to determine the class to which the new data belongs. So, classification time is linear in training set size for each test case.



There are many methods to calculate distance between two points. In our implementation we have used **Euclidean Distance.** Euclidean distance is calculated as the square root of the sum of the squared differences between a new training data point vector (x_i) and a test data point vector (x_j).

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

In algoritm implementation, we have used KNeighborsClassifier from scikit-learn library. We used two parameters to implement this algorithm:

- **n_neighbors:** defines the number of closest neighbors selected in classification time. We set k = 5 with cross validation method by trying range of k values and as the dataset is binary it is recommended to set k as odd number to avoid ties and determining majority class.
- **Metric:** defines the distance method which will be used.

Conclusion:

The results of each algorithm we have achieved shown below;

Algorithm	Time (seconds)	Accuracy	Precision	Recall
Decision Tree	2.122	0.886	0.903	0.885
kNN	0.003	0.95	0.943	0.951
Naïve Bayes	0.005	0.967	0.964	0.967

Where;

Time: the time is taken for training model.

Accuracy: this metric measures the performance of classification. The ratio of correctly classified data to the total number data in dataset.

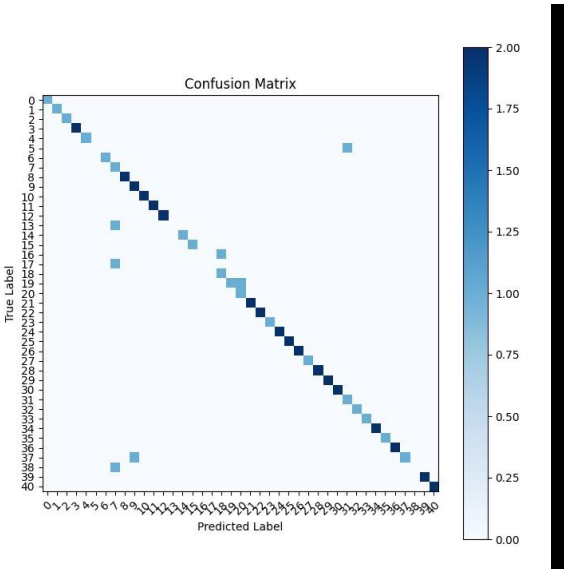
Precision: this metric is the fraction of relevant (truly predicted target) data among the retrieved data (true and false predicted target).

Recall: this metric is the fraction of relevant data that were retrieved. truly predicted of total actual true target.

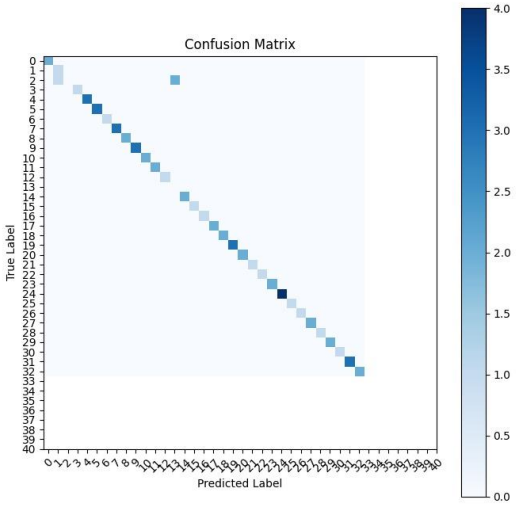
These results show that, decision tree gives high ratio in the performance metrics. the time taken by decision tree is longer than others because in code implementation it runs until all possible rules' accuracies found. Selects the highest accuracy to achieve best prediction. Normally it finds some rules that it doesn't give optimum result. kNN with k=5 is fastest one and has higher accuracy, recall and precision proportions than decision tree. On the other hand, Naïve Bayes gives the best results when compared to two other algorithms. So, using Naïve Bayes classification algorithm with this dataset gives the most accurate predictions.

The confusion matrix is provided to show the performance of classification models we have created and trained. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class.

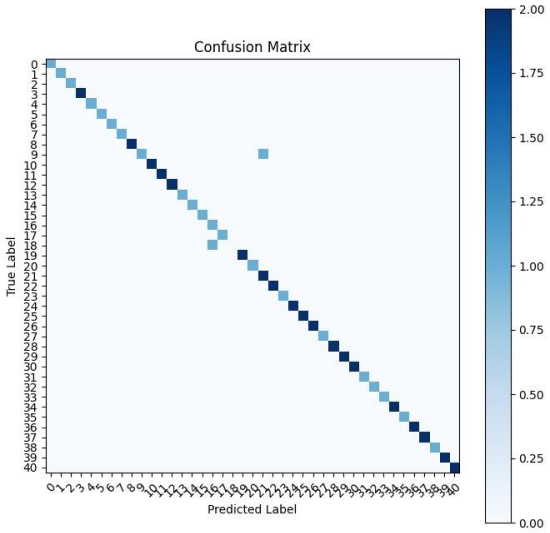
Decision Tree Algorithm



kNN Algorithm



Naïve Bayes Algorithm



Team Member Evaluation

Team Member Name	Rating for Individual Contribution (100% Max)
Abdulkader Saoud	100%
Esam Halimeh	100%