Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 12, December 2022

Comparative analysis of K-Means, SVM, Decision Tree and Naive Bayes in Predicting Diabetes Presence

Rohan Almeida, Rajeev Dessai, Ayush Noorani, Samuel Godinho, [1] Amogh Pai Raiturkar [2]
[1] Student, Msc-IT (Computer Science Department), Parvatibai Chowgule College Arts And Science
[2] Assistant Professor (PG, Computer Science Department), Parvatibai Chowgule College Arts And Science

***Abstract*:- In the context of rapidly growing amounts of data present today it is imperative to quickly dig out information from this data as information determines a large portion of the decision making process. Data mining allows us to do exactly that.**
Data mining is the process of turning raw data into useful information as it is an analytical process that allows us to find patterns, anomalies or correlations within large data sets that helps us acquire knowledge to predict outcomes or to validate findings. In order to find these patterns there are several algorithms.
This paper discusses the following algorithms.

1) **Decision Tree**
2) **Support Vector Machines**
3) **KNN**
4) **Naive Bayes**

The aforementioned algorithms are described as classification algorithms, it provides an interesting starting point for the analysis of them. The purpose of this paper is to analyze each of these algorithms to compare their prediction accuracy and features. We will be using diabetes prediction as the basis of this analysis and the Pima Indians Diabetes data set from kaggle.com as our input to determine whether a patient is diabetic or not. This paper will also highlight various other shortcomings or benefits of each algorithm and in which situations each of these algorithms perform the best.

Keywords: Data Mining, Comparative analysis, Classification algorithms, Decision Tree, Support Vector Machines, KNN, Naive Bayes

INTRODUCTION

Data mining is a process of going through a large dataset, processing them in some manner so as to identify some relationships or patterns that can be used to solve a problem or attend to a business need.
Classification algorithms are supervised learning algorithms that categorize structured as well as unstructured data into classes or categories.
There are two types of classification algorithms:

1) Binary classification: classifies data into one of two possible outcomes/classes.
2) Multi - class classification: classifies data into one of many possible outcomes/classes.
3) Multi - Label classification: classifies data into multiple non - mutually exclusive outcomes/classes.

Classification algorithms learn from the training data that is provided and then based on this, predict the outcome/class on the unclassified test dataset/data point.
Types of learners in classification algorithms:

1) Lazy learners: takes less time for training but comparatively more time for prediction as it stores the training data until it receives the test dataset/ data point.
   Example: Knn.
2) Eager learners: takes more time in the learning stage and less time during prediction as it makes a model based on the training set.
   Example: Naive Bayes, Decision Tree.

SCOPE OF THE RESEARCH

The purpose of this research is to do a comparative analysis on the algorithms like naive bayes, decision tree, SVM and KNN for predicting diabetes and to tell which model is best for such a case..

BACKGROUND RESEARCH

Decision Tree

A decision tree is a flowchart-structure where each internal node represents an attribute, the branches represent the decision rule and the leaves represent the outcome. Decision trees closely simulate human thinking. The tree is constructed by choosing the best attribute to split the records and making that attribute the decision node. This done until:

1. All the tuples belong to the same attribute value.
2. There are no more remaining attributes.
3. There are no more instances.

The attribute is selected or rather the splitting rule is decided based on selection measure, in this paper we have accounted for Information Gain (Entropy) and Gini Index.

Information Gain

$$\text{Info(D)}= - \sum_{i=1}^{m} pi \log_2 pi$$

$$\text{Info}_A\text{(D)}=\sum_{j=1}^{V} \frac{|Dj|}{|D|} \text{ X Info}(D_j)$$

$$\text{Gain(A)=Info(D)- Info}_A\text{(D)}$$

Gini Index

$$\text{Gini}(D)=1-\sum_{i=1}^{m}\text{Pi}^2$$

$$\text{Gini}_A(D)=\frac{|D1|}{|D|}\text{Gini}(D_1)+\frac{|D2|}{|D|}\text{Gini}(D_2)$$

$$\Delta Gini(A)=Gini(D)-Gini_A(D).$$

Naive Bayes

Naive bayes is a probabilistic classifier it is based on applying Bayes theorem.The Bayes Theorem asserts that the likelihood of the second event given the first event multiplied by the probability of the first event equals the conditional probability of an event depending on the occurrence of another event.The possibility of an event happening given that another event has already occurred (via assumption, presumption, statement, or evidence) is known as conditional probability.

$$\underset{1}{P(A\mid B)}=\frac{\overset{2}{P(B\mid A)}\cdot\overset{3}{P(A)}}{\underset{4}{P(B)}}$$

1.It is what we are trying to check, the probability of A given that B has already occurred.
2.Probability of B given that A has already occurred.
3.Probability of A occurring.
4.Probability of B occurring.

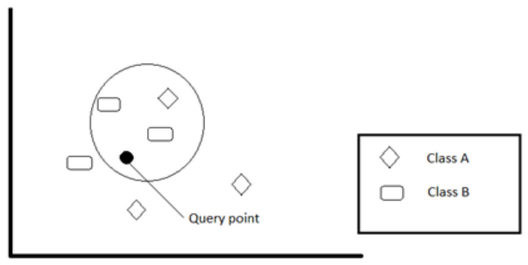
Naive Bayes makes an assumption that each feature makes an independent and equal contribution to future events.
Use cases:
Face Recognition: used to identify the faces or its other features,like nose,mouth,eyes.
Weather prediction: it is used to predict if weather will be good or bad.

Medical Purpose: Patients are diagnosed with the information the classifier provides.It can be used to indicate if the patient is at high risk for certain diseases and conditions

News classification: Naive bayes classifier ,google new recognizes whether the news is political,world news etc.

K Nearest Neighbors

The KNN method is one of the simplest machine learning algorithms and is used extensively in classification problems due to its very adaptable and basic design. KNN classifies the new data points based on the similarity measure of the earlier stored data points. The approach is well known for being used to address regression and classification problems for data with a variety of label counts, noise levels, ranges, and contexts.



The distance between the test data point and the training data point is found out with the help of the Euclidean distance formula:

$$d=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

d = Euclidean distance between the two points

(x1, x2) = first point

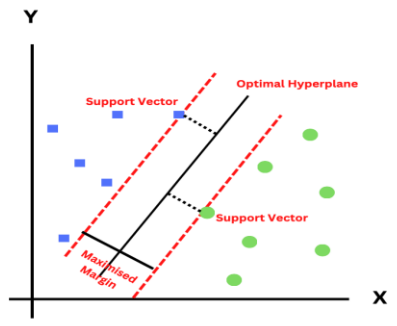
(y1, y2) = second point

Support Vector Machine

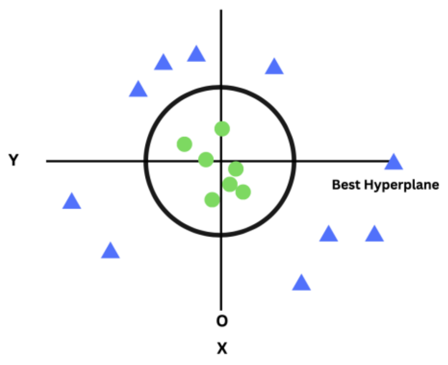
One of the most popular supervised learning algorithms used for both classification and regression problems is called Support Vector Machine or SVM . However, it is mainly used for machine learning classification problems. Inorder to separate the classes in an n-dimensional space from the multiple decision lines/boundaries , we need to find the best decision boundary to help classify the data point, so that you can easily classify the new data point in the correct category in the future. This best boundary is known as the SVM hyperplane. The hyperplane created must always have a maximum margin, which showcases the maximum distance between data points. SVM selects the extreme points/vectors that will help create the hyperplane. The data points or vectors are closest to the hyperplane thus affecting the position of the hyperplane. Since these vectors support the hyperplane, it is therefore called a support vector.

There are 2 types of SVM:-

Linear SVM: When data is linearly separable data, linear SVM is used. This means that a dataset can be divided into two classes using a single straight line. Such data is called linearly separable data, and the classifier is known as a linear SVM classifier.



- Non-linear SVM: For nonlinearly separated data non-linear SVM is used. In other words if a dataset cannot be classified using a straight line, then such data is called non-linear data and the classifier used is called the non-linear SVM classifier. Kernel function transforms the set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces.



Implementation and methodology

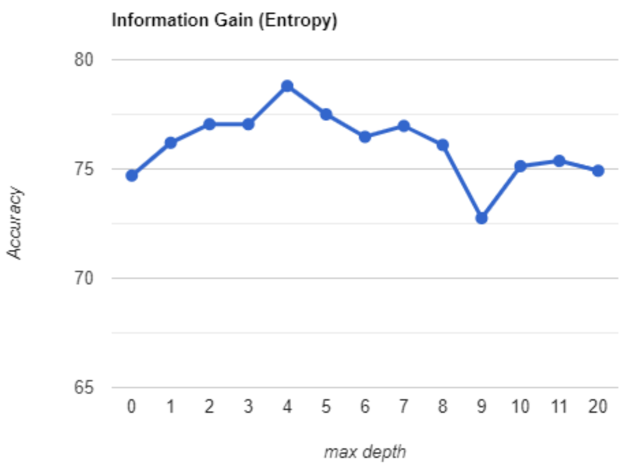
Decision Tree's

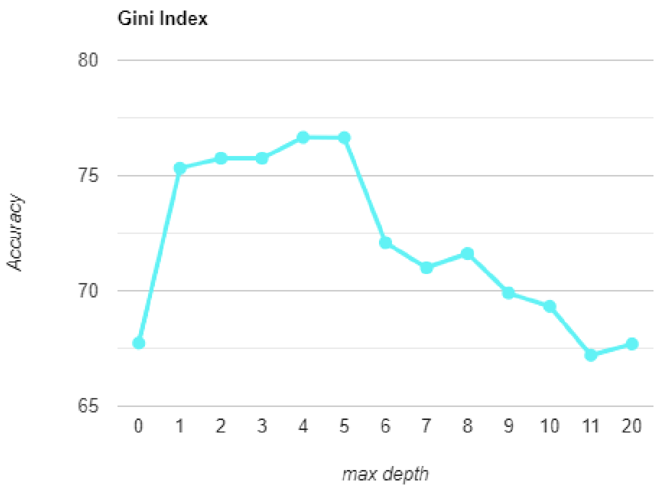
Below are some graphs and tables denoting the overall accuracy of Decision Trees using both Information Gain (Entropy) and Gini Index.

Recall (at max depth of 4)

	Non-Diabetic	Diabetic
Information Gain (Entropy)	90%	61%
Gini Index	85%	62%

Max depth is the constraint applied on the depth of the decision tree.
The tree was optimized by selecting the best selection measure and constraining the depth of the decision tree. From the observed graphs we can see that overall our highest





accuracy and recall values were achieved using entropy with a max depth of 4.

It is interesting to note that while entropy has the higher accuracy initially, gini index starts to increase its accuracy as the depth increases but both start to fall to values akin to that of not constraining the tree depth to a depth of 4.

This led us to ask: What if we change the size of our training size? In all above testing 70% of the total data set was used. The following accuracies are observed when we start to change our training size:

For Information Gain (Entropy):

% of data used for training	Accuracy at no max depth specified:	Best Accuracy:
90%	73%	79% at max depth 2
70%	74.7%	78.8% at max depth 4
50%	67.05%	74% at max depth 1
30%	66.83%	74% at max depth 1 and 2

The same is observed with the Gini Index. As our training size changes so does our accuracy (in some cases it produced a better accuracy than entropy).

Naive Bayes

Here Pima Indian diabetes dataset used for demonstration.We have used Gaussian Naive bayes model which a model of Naive bayes be used to predict if a particular subject has diabetes or not.

A Gaussian distribution the mean value is x=np where n is the number of events and p is probability of any given integer value of x.

Classification report

	precision	recall	f1-score	support
0	0.80	0.88	0.84	155
1	0.69	0.55	0.61	76
accuracy			0.77	231
macro avg	0.74	0.72	0.73	231
weighted avg	0.76	0.77	0.76	231

The classification report shows the main classification metrics as we can see that the precision for predicting the people who don't have diabetes is 80% and the precision for predicting the people that have diabetes is 69% which is a bit lower. Precision tells us how accurately the model does the classification.Recall tells us the ability of the model to find all relevant cases.Here the recall for people who don't have diabetes is 88% and for the people who have diabetes is 55%, the overall recall is 77% . F-1 score tells the performance of the model here the performance is 84% and 61% .Support defined as the number of samples that accurately represent each class of target values.By using Gaussian NB we get an accuracy of 77%.

K Nearest Neighbours

The supervised machine learning algorithm k-nearest neighbour (KNN) is primarily employed for classification tasks. For predicting diseases, it has been employed extensively. Using the features and labels of the training data, the KNN, a supervised algorithm, forecasts the categorization of unlabelled data.

The k nearest training data points (neighbors), which are calculated using the Euclidean distance formula, are ranked based on the distance from the test data point and given a rank. The ones closest to the testing query, are typically used by the KNN method to classify datasets using a training model identical to the testing query.

Total dataset of 768 rows. The attributes which were considered for the model were Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI and Age. 70% of the total data was used for training the model and the remaining 30% was used as test data.

The accuracy (0 – 1 scale) observed for different values of "k" are as follows:

k - value	Accuracy observed
3	0.696969696967
5	0.688311688311
50	0.722943722943
100	0.688311688311
110	0.688311688311
120	0.705627705627
150	0.640692640692
200	0.640692640692
250	0.640692640692

As observed from the above outcomes, we see that after a certain point of "k" value (150 in this case), we see no change in the accuracy. The highest accuracy for this model can be observed between "k" values between 50 – 120.

Support Vector Machine

The dataset for diabetes used here for demonstration is taken from Pima. The model used here is support vector machine which predicts if a particular subject has diabetes or not.

	Precision	Recall	F1-Score	Support
0	0.76	0.92	0.83	155
1	0.71	0.39	0.51	76
Accur-acy			0.75	231
Macro avg	0.74	0.66	0.67	231
Weigh-ted avg	0.74	0.75	0.73	231

The above table interprets the classification table which shows the main classification metrics. The table shows us the precision for predicting the people who dont have diabetes is 76%. In comparison to this we also see that the precision for predicting that people have diabetes is 71% which is not very low.

Where Precision helps us to understand how the model does the classification accurately, recall on the other hand tells us the ability of the model to find all relevant cases.

In the above diagram we can see that the recall for people who don't have diabetes is 92% where as the recall for people with diabetes is 39%. Thus, the overall recall is 75%. Thus, we see that the recall for people who dont have diabetes is very high compared to those who have diabetes. Inorder to know the performance of the model, we can use the F-1 score.

From the above table we see that the performance is 83% and 51% respectively. Support on the other hand signifies the number of samples that accurately represent each class of target values. By using SVM we get an accuracy of 74%.

Conclusion

Decision Tree

The observations presented highlights a major drawback of the decision tree algorithm:

It is sensitive to the data it trains on. When we start to provide smaller data sets for training we run into the issue of underfitting (it cannot accurately capture the underlying data trend. Performs well only on training data and not on testing data.) and once we increase the size of our data set we run on the issue of overfitting (starts learning from noise and inaccurate entries). However in the case of the diabetes data set entries are mostly accurate and hence overfitting isn't largely an issue. These aforementioned issues can be mitigated or at the very least reduced by altering the max depth of the tree as seen in the table above. When the training size changes the best accuracy is achieved at different max tree depths.

Other drawbacks of decision trees:
1) Higher training time (depending on selection measure too)
2) A small change in the data set can cause a significant change in the structure of the decision tree.

Decision trees do have certain benefits as well:

1) Requires less effort in data preparation or pre-processing. In the analysis we only specified our feature columns and target columns and performed no pre-processing
2) Decision trees are easier to understand as they mimic human thinking making it easy to explain and follow
3) Decision trees can handle high dimensional data producing good accuracy.

Naive Bayes

Naive Bayes is simple to implement; you don't have to specify many additional hyperparameters (Hyperparameters are variables whose values influence the learning process and define the model parameter values that a learning algorithm ultimately learns.) like other models.Naive bayes does not require very large dataset because of the independent class assumption,naive bayes classifier learn quickly the high dimensional features with small training data compared to other models which require large datasets to train.Naive is very useful when you have small dataset. Naive bayes can be used for both continuous and discrete data.

There are a few drawbacks of Naive Bayes.
Naives bayes deals with zero frequency problem.If in your test data has a value which was not present in the training data set,the model will assign it a probability of zero and won't be able to predict in this regard.Naive bayes predictions are not taken seriously because its called a lousy estimator.It makes the erroneous assumption that each feature is independent.
Although it may seem wonderful in principle, you hardly ever discover a set of independent traits in practice.
Naive Bayes does a good job in predicting the case where a person who doesn't have diabetes but it falls behind to predict if a person has diabetes.

K Nearest Neighbours

K Nearest Neighbours is a very simple supervised learning algorithm which is mostly used for classification problems. Being able to make highly accurate predictions, kNN algorithm can be used for applications that require high accuracy and do not necessarily require a model which is human readable.

Some downfalls of kNN algorithm are:

1) Prediction time is slow as it adopts a lazy learner, hence the prediction time will increase with the increase in size of the dataset.

2) Consumes more memory as it has to store the training data as well.

3) Since it is a distance-based algorithm, it requires a lot of computation when working with a large dataset.

4) Knn is sensitive to outliers and missing values in the dataset.

5) The more examples, predictors, or independent variables there are, the slower the algorithm becomes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 12, December 2022

Support Vector Machine

SVM is a supervised machine learning algorithm which can be used for classification or regression challenges, but is mainly used in classification problems. The SVM algorithm helps us to plot each data item as a point in n-dimensional space (where n is a number of features) and the value of each feature is the value of a specific coordinate. Then we do the classification by finding the hyperplane, which distinguishes the two classes very well. SVm can be used in handwriting recognition, face detection, email classification, in web pages and many more areas.

Some of the pros that we have come across are:-

1. SVM has a L2 Regularization feature. So we can say that it has good generalization capabilities that prevent it from over-fitting.

2. Kernel trick can be used to efficiently handle non-linear data using SVM.

3. Classification problems can be solved using SVM while for regression problems SVR (Support Vector Regression) is used.

4. The hyperplane is not affected greatly due to some small changes in the data. So we can say that the SVM model is stable.

Some of the cons of SVM are listed below:

1. It is tricky and complex to choose an appropriate Kernel function while handling the non-linear data.

2. A Lot of memory is required since you have to store all the support vectors in the memory and this number grows abruptly with the training dataset size.

3. Before applying SVM, we must feature scale the variables.

4. When it comes to large datasets, SVM takes a long training time.

REFERENCES

[1] https://www.datacamp.com/tutorial/decision-tree-classification-python
[2] https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
[3] https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1
[4] https://www.w3schools.com/python/pandas/default.asp
[5] https://scikit-learn.org/stable/
[6] https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a
[7] https://www.educba.com/decision-tree-advantages-and-disadvantages/
[8] https://www.upgrad.com/blog/naive-bayes-classifier/
[9] https://www.tutorialspoint.com/scikit_learn/scikit_learn_gaussian_naive_bayes.htm
[10] https://www.analyticsvidhya.com/blog/2021/11/implementation-of-gaussian-naive-bayes-in-python-sklearn/
[11] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
[12] https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html
[13] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
[14] http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-svm.html?m=1