



Classification Algorithms

Submitted By:

Kautuk Desai	50247648
Sujay Vijay Purandare	50205931
Palwinder Singh	50247454

Table of Contents:

1. Nearest Neighbor
2. Naïve Bayes
3. Decision Tree
4. Random Forests Implementation
5. Boosting Implementation

1.K-Nearest Neighbors

1.1Algorithm Flow:

1)The idea is to measure distance of every record from every other record in training dataset. We use Euclidean distance to measure these distances. The attributes are normalized before measuring the distances.

2)We then sort the distances and identify k nearest neighbors by taking top k of these sorted distances for each record.

3)We use class labels to decide the maximum votes to predict class for a particular test data. The maximum votes are counted by taking maximum count of classes to which k closest neighbors belong.

1.2 Choice Description:

1) Value of K: The value of K greatly influences the result of this algorithm. If K is set to be large, it includes the points from other neighborhood. If value of k is small, it incurs noise. We avoided an even value of K because there might be a case such that exactly $k/2$ records are labeled wrongly. We tried the values of k from 3 to 9 and found 100% accuracy. The reason may be that the training set is small enough to incur any inaccuracies.

2)Distance type: We chose Euclidean distance because we are working in continuous space. The dimensions are relevant and scaled properly which is ideal for Euclidean distances.

1.3 Results:

Data set 1

K= 2

Mean accuracy is : 0.954912280702
Mean precision is : 0.972613871636
Mean recall is : 0.933585643952
Mean fmeasure is : 0.951724849177

K=5

Mean accuracy is : 0.9551788820815369
Mean precision is : 0.9718745644599303
Mean recall is : 0.9142711801701813
Mean fmeasure is : 0.9324623198464908

K=10

Mean accuracy is : 0.964545406593408
Mean precision is : 0.9990283590805330
Mean recall is : 0.9364424464424465
Mean fmeasure is : 0.9421166773605798

Data set 2

K=2

Mean accuracy is : 0.6696536775536797
Mean precision is : 0.5166205553703703
Mean recall is : 0.371715554661717
Mean fmeasure is : 0.428537036926

K=5

Mean accuracy is : 0.6664477335800185
Mean precision is : 0.5280285152840729
Mean recall is : 0.33858550583924
Mean fmeasure is : 0.4115266565266565

K=10

Mean accuracy is : 0.6805797101449274
Mean precision is : 0.5360347501319369
Mean recall is : 0.3755602178523231
Mean fmeasure is : 0.344546025705301

1.4 Pros of KNN:

- 1) It is simplest among all available algorithms and provides the flexibility to modify as the data type changes.
- 2) It allows us to choose any distance function we want (Euclidean, Hamming etc) for each feature independent of other features. This is useful in the case where dataset contain mixed datatypes.
- 3) We don't have to make any assumptions about characteristics of concept to be learned.
- 4) Complex concepts can be learned by local approximation using simple procedures.

1.5 Cons of KNN:

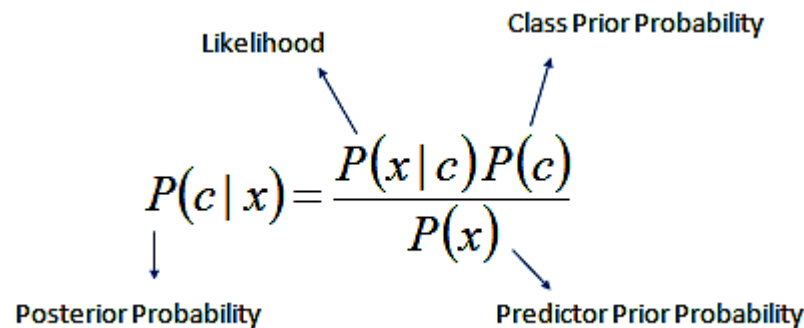
- 1) It is computationally expensive to find the k nearest neighbors when dataset is very large.
- 2) Performance depends on number of dimensions we have.
- 3) The model cannot be interpreted since there is no description of learned concepts.

2. Naive Bayes

2.1 Algorithm Flow:

1) Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

2) We use following formula:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$


The diagram illustrates the components of the Naive Bayes formula. It shows the equation $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Four labels with arrows point to specific parts of the formula: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

3) In algorithm, we also segregate continuous and categorical features and handle them separately. For continuous features, we calculate columnwise mean and standard deviation and calculate Gaussian Distribution, while for categorical features, we calculate prior and posterior probability.

4) During testing, we calculate conditional probability for each class and multiply continuous/categorical probability to get class probability. Finally, we assign class with highest probability as the class for that sample.

2.2 Choice Description:

1) Zero Probability: We need to use Laplacian correction if posterior probability is zero. (Please note that we haven't implemented this in our code).

2) Continuous feature: To check whether the value is continuous, we convert them to float. If it throws error, then the feature is categorical. For continuous features, we need to calculate Gaussian probability. For categorical, we need to calculate discrete probability.

2.3 Pros:

1) It is mostly faster than other classification algorithms.

2) It is comparatively easier to implement.

3) The labels are independent of each other.

2.4 Cons:

- 1) The algorithm doesn't work satisfactorily if data is co-dependent.
- 2) We may run into the possibility of incurring zero posterior probability for categorical data.

2.5 Results:

Data set 1

K= 2

Mean accuracy is : 0.926161354089449
Mean precision is : 0.9055059523809524
Mean recall is : 0.8898335582546111
Mean fmeasure is : 0.8974370698918586

K=5

Mean accuracy is : 0.9351788820815369
Mean precision is : 0.9218745644599303
Mean recall is : 0.9042711801701813
Mean fmeasure is : 0.9124623198464908

K=10

Mean accuracy is : 0.9356593406593408
Mean precision is : 0.9202835908053301
Mean recall is : 0.9064424464424465
Mean fmeasure is : 0.9121166773605798

Data set 2

K=2

Mean accuracy is : 0.6796536796536797
Mean precision is : 0.5266203703703703
Mean recall is : 0.6717171717171717
Mean fmeasure is : 0.5901360544217686

K=5

Mean accuracy is : 0.6864477335800185
Mean precision is : 0.5380285152840729
Mean recall is : 0.6903549850583924
Mean fmeasure is : 0.6015266565266565

K=10

Mean accuracy is : 0.6905797101449274
Mean precision is : 0.5460347501319369
Mean recall is : 0.6972602178523231
Mean fmeasure is : 0.6041546025705301

3. Decision tree

3.1 Algorithm Flow:

- 1) First we created a decision tree on the basis of training data using node splitting
- 2) Then I checked for the best nodes split by using left and right according to the given depth (of tree)
- 3) For the best split we used GINI index which is used to assign the classes.
- 4) For test cases, pass the test sample from the tree and simply prune the tree from its root.
- 5) Assign the class and evaluate accuracy and other evaluation measures.

3.2 Choice Description:

Continuous features: In order to check the continuous, we first convert each value to float, if it gives the value error then the feature is categorical.

Categorical features: Since it is a decision tree, the categorical data can be used as a value like any other value. For test cases we need to detect the continuous and categorical features in order to prune the trees.

Best Feature: Next for evaluating the best feature, we used GINI index which is the evaluation metric used to give the best split in the decision tree. The lower GINI index, the better is the split and therefore those features have lowest gini values.

Post-processing: Results are evaluated through cross-validation (10 fold where this value is user input).

3.3 Pros:

- 1) We can combine this technique with other ensemble techniques.
- 2) We don't have to handle categorical data separately.

3.4 Cons:

- 1) It is slower than other techniques.
- 2) It becomes complex for large parameters values for large datasets.

3.5 Cross Validation :

It is a technique of assessing how the results of statistical analysis will generalize to independent data set. It is mainly used when our goal is prediction and when we want to estimate how accurately a predictive model will perform in practice. We have implemented 10-fold cross validation. To do that we first shuffle the data set and then identify its size required for 10-fold validation. We then segregate the data into test and train set by partitioning it as per the size identified previously. We then perform regular algorithm on this set to compute the performance metric. After this step, next set of test dataset elements are extracted and the same process as before is performed (called as next fold).

This process is repeated 10 times and performance metrics are evaluated for each of the folds. And finally we take the average of performance metric across the 10 folds.

3.6 Results:

Dataset1:

- 1) Mean precision: 65%

- 2)Mean recall: 65%
- 3)Mean Accuracy: 75%
- 4)Mean F1: 63.33%

Dataset2:

- 1)Mean precision:58.44%
- 2)Mean recall: 43.97%
- 3)Mean Accuracy: 69.25%
- 4)Mean F1: 48.98%

4.Random Forest

4.1 Algorithm Flow:

- First ,we get the input parameters as number of trees needed and selecting random features from the tree.
- After that we create a decision tree on the basis of training data by selecting random rows with replacement (referred to as bagging).
- Check for the best nodes split by pruning left and right according to the given depth of the tree
- For the best split we used GINI index which is used to assigning the classes.
- For test cases, pass the test sample from the tree and simply prune the tree from it's root.
- Assign the class and evaluate accuracy and other evaluation measures.
- Repeat all the steps from 2 if the trees are more than one and aggregate the accuracy.

4.2 Pros

- 1)It is more accurate than decision trees.
- 2)It is also faster than decision trees.

4.3 Cons

- 1)Result can vary for each iteration.
- 2)We may incur some inaccuracies.

4.4 Cross Validation :

It is a technique of assessing how the results of statistical analysis will generalize to independent data set. It is mainly used when our goal is prediction and when we want to estimate how accurately a predictive model will perform in practice. We have implemented 10-fold cross validation. To do that we first shuffle the data set and then identify its size required for 10-fold validation. We then segregate the data into test and train set by partitioning it as per the size identified previously. We then perform regular algorithm on this set to compute the performance metric. After this step, next set of test dataset elements are extracted and the same process as before is performed (called as next fold).

This process is repeated 10 times and performance metrics are evaluated for each of the folds. And finally we take the average of performance metric across the 10 folds.

4.5 Results:

Dataset1:

- 1)Mean precision: 55%
- 2)Mean recall: 60%
- 3)Mean Accuracy: 55%
- 4)Mean F1: 55.67%

Dataset2:

- 1)Mean precision:30.53%
- 2)Mean recall: 29.29%
- 3)Mean Accuracy: 63.89%
- 4)Mean F1: 21.212%

5.Boosting

The boosting approach is used to give more weights on weak samples i.e. which are misclassified. The boosting technique used is adaBoost(adaptive) which is described below. Final prediction is weighted average of all the classifiers with weight representing the training accuracy.

5.1 Algorithm flow:

- Take the training set and weight them equally i.e. $1/\text{number of training data}$
- Randomly choose weights (bagging) and check the misclassified samples and weight them more than the correctly classified samples.
- Calculate and save the staging values alpha and trees for further prediction of test labels.
- Update the weights by using the given alpha and repeat from step 2 for given number of iterations.
- In testing phase, for every test samples, get the prediction label using saved trees and corresponding to the trees assign alpha for that label by adding it.
- Check the maximum value of alpha aggregated in the label which will be the class of the test sample.

5.2 Pros

- 1)There is no model bias, so we can get more accuracy.
- 2)It is faster than decision tree and also more accurate.

5.3 Cons

- 1)It may be more time consuming because of weight selection.
- 2)The over boosting may lead to overfitting.

5.4 Cross Validation :

It is a technique of assessing how the results of statistical analysis will generalize to independent data set. It is mainly used when our goal is prediction and when we want to estimate how accurately a predictive model will perform in practice. We have implemented 10-fold cross validation. To do that we first shuffle the data set and then identify its size required for 10-fold validation. We then segregate the data into test and train set by partitioning it as per the size identified previously. We then perform regular algorithm on this set to compute the performance metric. After this step, next set of test dataset elements are extracted and the same process as before is performed (called as next fold).

This process is repeated 10 times and performance metrics are evaluated for each of the folds. And finally we take the average of performance metric across the 10 folds.

5.5 RESULTS :

Dataset1:

- 1)Mean precision: 85.6%
- 2)Mean recall: 87.23%
- 3)Mean Accuracy: 88.73%
- 4)Mean F1: 85.68%

Dataset2:

- 1)Mean precision: 85.6%
- 2)Mean recall: 87.23%
- 3)Mean Accuracy: 88.73%
- 4)Mean F1: 85.68%