Feature Extraction using KNN and Classification using FastKNN

[1] P. Praneetha, [2] A. Ramana Lakshmi, [3]G. Lalitha Kumari, [4] M. Ram Gopal, [1]M. Tech student, [2,3] Asst. Professor, M. Tech (Ph. D), [4] Asst. Professor, M. Tech, [1,2,3,4] Dept of CSE, PVP Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India, [1]praneethaparimi35@gmail.com, [2]aramanalakshmi@gmail.com, [3]ramgopal.musunuri@gmail.com, [4]lalithajoy.nuthakki@gmail.com

Abstract-Feature analysis is one of the major components of Machine Learning. It consists of two processes, Feature Extraction and Feature Selection. The Feature Selection has an importance in data mining, soft computing, and big data analytic areas. In this task, we propose an intuitive way of extraction and selecting the features from the chosen data through KNN (K- Nearest Neighbour) method that improves the efficiency and effectiveness by reducing computation time, to improve the performance metrics such as accuracy, prediction rate and minimize the error and also for better understanding of data. In this process, FastKNN (Fast K- Nearest Neighbour) model, GLM, and SVM model are used for classification and prediction process for the datasets. From the observations of results, Feature Extraction with KNN model and classification using FastKNN method would give better results when compared with Feature Extraction with KNN model with GLM and also it is observed that Feature Extraction with KNN is more or less similar to SVMRFE Feature Extraction method.

Keywords: Feature Extraction, KNN (K Nearest Neighbour), FastKNN, GLM (Generalized linear model), SVM (Support Vector Machine), SVMRFE (Support Vector Machine with Recursive Feature Elimination).

I. INTRODUCTION

In the Resent years technology is widely using in every area. The use of sensors is increasing day by day; the data that the sensor is storing may contain high dimensions and also the high amount of redundant data. Generally in machine learning problems, there will be the large data to process, the higher dimensionality data may contain the features that are correlated and hence, resulting in getting redundant data. As there is an increase in features; it is difficult to compute the data. In this case, dimensionality reduction is used.

Dimensionality Reduction is a process of converting the higher dimensional data to lower dimensional data leaving the meaning of features as same. [1] Feature selection and feature extraction are known as dimensionality reduction techniques. Dimensionality of features can be reduced by the feature extraction by projecting the new features with lower dimensionality to the new feature space with the original features. In the feature space, it is difficult to match the original features and new features. As there is no proper meaning between the newly extracted features and the original features, analysis of new features is difficult.

Another technique in dimensionality reduction is Feature selection, known as attribute selection. [1] It selects the subset of original features without making any changes and remains the same meaning, hence redundancy is reduced and relevance to the class labels is increased.

II. RELATED WORK

Many researchers propose Feature Extraction methods, such as support vector machine (SVM), Naive Bayes, support vector machine with recursive feature elimination (SVM-RFE), K-means, F-score, and ReliefF. SVM-RFE achieves more accuracy than the remaining methods in disease prediction [2]. This method is having better performance in gene selection and multiclass classification [3]. The SVM-RFE method for feature selection can remove irrelevant features and linear redundant features using correlation coefficient technique by computing relation between the features; but it cannot remove the non-linear redundant features [4]. The gene selection method *t*-statistics embedded in support vector machine (SVM-*t*) with recursive support vector machine (RSVM), identifies significant genes than SVMRFE in little recurrence. It is limited to linear features as support vectors are far for nonlinear features [5].

The Fast KNN classification algorithm identifies k close vectors by computing the partial distance search. For the labels c in d dimensions FastKNN computes in O(k.c.d) complexity. This method is very efficient for the application which requires the minimal error rate and minimal complexity [6]. This algorithm is fast in the case of multi-dimensional classification data [7] and takes less

DOI : 10.18231/2454-9150.2018.0434

computational time than KNN algorithm.

The KNN is a non linear mapping classification algorithm therefore; the KNN method can classify both the linear and non linear features. Hence it can remove non linear redundant features. Many of the neural network approaches face convergence problem, they need long training time. But the KNN method does not require the training of data, hence no convergence problem arises, therefore it is easy to implement and maintain.

In this paper, we proposed a KNN method for Feature Extraction. The KNN method implements KNN (k-Nearest Neighbor) classifier based on the ANN (Approximate Nearest Neighbor). By using KNN method, feature engineering is performed and extracts high informative features from the data. This method provides better predictive performance and reduces the log loss. It finds the k nearest neighbors for every point in O(N log N) time. It increases the classification accuracy and reduces computational time for high dimensional data.

A. K-Nearest Neighbor classifier:

The KNN classification algorithm is a non-parametric and instance-based learning algorithm [8] i.e., KNN algorithm does not learn a model automatically it takes the training examples as a knowledge to predict the target instance. KNN is more suitable for the diagnosis [9].The algorithm takes a training dataset which contains examples that are classified with labels into several categories. For each unlabeled instance in the test data, KNN selects k nearest instances from the training data, where k is the no. of nearest neighbors to consider. The class which contains the majority of nearest neighbors is the class of the target instance. The following algorithm is the KNN algorithm which takes training data as input and finds class label of target data.

B. K-Nearest Neighbor Algorithm:

```
Input:  1. Training data with class labels as P={1,2,..., n}.
        2. The test instance T.
Algorithm:
for (i=1to n)
    The distance dt between T and $P_i$ is computed.
    if (i<=K)
        Consider $P_i$ as nearest neighbor and add it in the neighbors set.
    if ($P_i$ is the nearest point to any point in nearest neighbor set)
        Delete the neighbor which is having farther distance of all the neighbor points in the set and add $P_i$ in
        the nearest neighbor set.
end for
Output:  Class label of T.
```

K-Nearest Neighbor Algorithm

It takes the input as training data and test data and finds the class label for target instance**.** KNN Calculates the distance of instances as dt(T, P_i) where i=1,2,3,….n; and dt represents the distance of two points T and P_i. The distance is calculated by using Euclidean distance. The distances that are computed is arranged from minimum distance to maximum distance. Let k is a positive real number, represents the number of nearest neighbors. KNN finds the k nearest distances and corresponding points. Let c be the class label and k_c be the no. of points belongs to cth class of k points i.e. $k \geq 0$. If $k_j < k_i$ for all $i \neq j$ then assign T in class c.

C. Advantages of Proposed model:

- FastKNN builds the classifier with large dataset in few seconds.
- FastKNN takes O(N log N) time for computing the k neighbors from every point.
- KNN takes training data without knowing about the structure of data.
- If any new instances are added, then KNN doesn't need of training the data again.
- KNN avoids over-fitting by generating training features using n-fold cross-validation.
- KNN is very easy for computing distances of multidimensional space while comparing with many of the machine learning methods.
- FastKNN makes a nonlinear mapping of original space and projects into linear, where classes are linearly separable.
- FastKNN provides better predictive performance by computing probabilities from the inverse of nearest neighbors distance.

D. KNN method for Feature Extraction:

Feature extraction process will alters the features and acts as shrinkage of class probabilities. The newly extracted features are the sequences of the original features. We propose KNN method for feature extraction and generate new features. It generates K*C new features, by computing the distance for every point of K nearest neighbours of all classes and the test instance, where K and C are the nearest neighbours require to compute and their class labels. Then, perform k-fold cross validation to avoid over-fitting.

In this feature extraction process, distance is calculated using ANN (Approximate Nearest Neighbor). ANN includes the Euclidean distance method and Manhattan distance method.

- if U = (u_1, u_2,...,u_n) and V = (v_1, v_2,...,v_n) are the two distinct points, then the Euclidean distance dt from U to V is as follows:

$$dt(U,V) = dt(V,U) = \sqrt{(v_1 - u_1)^2 + (v_2 - u_2)^2 + \cdots + (v_n - u_n)^2}$$

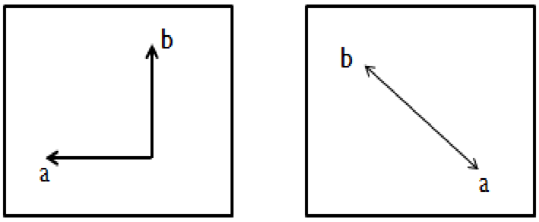
DOI : 10.18231/2454-9150.2018.0434

$$= \sqrt{\sum_{p=1}^{n}(v_i - u_i)^2} \qquad \text{-----------} \quad (\text{eq1})$$

- The Manhattan distance determines the distance when the data points follows grid path. The distance metrics is the aggregation of the differences of factors of two items. The formula for calculating the distance between the two points U = (u_1, u_2,...,u_n) and V = (v_1, v_2,...,v_n) is:

$$\sum_{p=1}^{n}|u_p - v_p| \qquad - - - - - (\text{eq2})$$

Where n represents no. of variables and u_p is pth point in U and v_p is pth point in V. The figure 1 shows the representation of Manhattan distance and Euclidean distance.



a. Manhattan distance **b. Euclidean distance**

Figure 1: representation of Manhattan distance and Euclidean distance

E. *Algorithm for Feature extraction with KNN:*

Input: 1.Training dataset T= (1, 2,…, n) and test data as s.
2. Number of class labels as c, distance as dt.

Algorithm:

```
for ( i from 1 to c)
    for ( j from 1 to k)
    Calculate the distance dt from s to $k_j$ in T using the eq1 or eq2.
    end for
end for
Now perform k-fold cross validation on the k*c features.
```

Output: generates new features.

Feature Extraction with KNN algorithm.

The algorithm for Feature Extraction with KNN is taking the input as original data and performs feature extraction technique to generate new reduced features.

F. *Fastknn classifier:*

The fastknn implements k-Nearest Neighbor (fast KNN) [8] method to develop the large datasets quickly. It computes the k nearest neighbors for every point in training and testing set in O(N log N) times using KD-tree. It provides shrinkage estimator to the related classes, based on the inverse distances of the nearest neighbors and computes class label for the test instance.

$$pt(p_a \in q_b) = \frac{\sum_{n=1}^{n}\left(\frac{1}{dt_{an}}.(m_{an} \in q_b)\right)}{\sum_{n=1}^{n}\left(\frac{1}{dt_{an}}\right)} \quad - - - - - (\text{eq3})$$

The probability pt of finding the class label of p_a is shown in eq3, where p_a is the a^{th} test instance, q_b is the b^{th} class label, m_{an} is the k^{th} nearest neighbor of p_a, and dt_{an} is the distance between p_a and m_{an}. This estimator follows the weighted voting rule, i.e., the neighbors that are close to p_a will have more influence on predicting p_a's label and reduces the log loss.

G. *Example of FastKNN classifier:*

The algorithm is taking the input as training data which contains the points of three class labels is shown in the figure 3. Now compute the distance to every point from the test instance p as shown on the figure 4. Arrange all the distances in the increasing order as in figure 5. Then compute the class label of the target instance p by the weight voting rule. From the figure 6 there are two nearest neighbors from class star and one nearest neighbor from circle and rectangle classes. According to the FastKNN method the class label having majority of neighbors is the class of the target instance. Therefore, the target instance p belongs to thestar class.

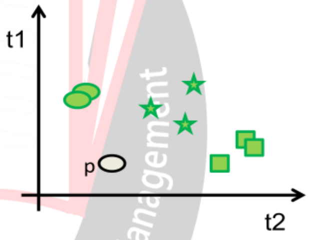


Figure 3: Training data with three class labels

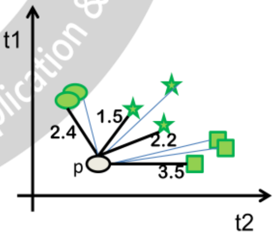
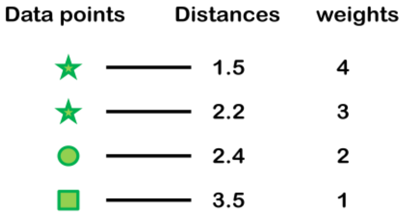


Figure 4: Computing the distance of points



Data points	Distances	weights
★	1.5	4
★	2.2	3
●	2.4	2
■	3.5	1

Figure 5: Finding nearest neighbours

DOI : 10.18231/2454-9150.2018.0434

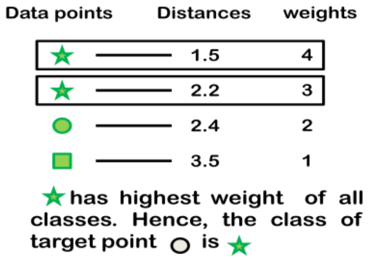

Data points
Distances
weights
★ 1.5 4
★ 2.2 3
● 2.4 2
■ 3.5 1
★ has highest weight of all classes. Hence, the class of target point ○ is ★

Figure 6: predicting the class of target variable

H. Algorithm for FastKNN:

1. FastKNN takes the input, training data and test data.
2. Compute the distance from the test instance and the k nearest points in training data using eq1 or eq2.
3. Assign the weights to all the k nearest neighbor points in the inverse distance.
4. Compute the probability of finding the class label for the target instance using eq3.
5. Assign the class label to target instance using weight voting rule, this selects the class label which is closest to the target instance.

Algorithm for FastKNN

The algorithm for FastKNN, Takes input data as training data and finds the probability of finding the class label of testing data using weight voting method**.**

III. DIAGNOSING WISCONSIN BREAST CANCER WITH THE FASTKNN ALGORITHM

In medical, diagnosis is a difficult task because, the information may contain ambiguity, as the reason may be insufficiency of information and may be having misleading characters. Hence to achieve better results in diagnosis, considering diagnosis as a classification problem and apply machine learning techniques for the classification.

Wisconsin Breast Cancer dataset is considered for diagnose using the machine learning classification techniques. Many researchers [10] have performed different classification techniques on Wisconsin Breast Cancer problem; the data contains the information of breast masses indicating the malignant cancer. The University of Wisconsin has 569 records and 32 features, in which 10 numerical attributes are determined for each cell: patient's id, diagnosis, radius, area, texture, smoothness, perimeter, compactness, symmetry, concavity, concave points, and fractal dimension. Except the id, all attributes are required.

Each sample is either malignant "M" or benign "B". The chosen classification technique has to classify the data into bening and malignant classes. The other features comprises of mean, standard error and worst values for 10 different characteristics of cell nuclei. We see the machine learning approach for detecting cancer by applying the KNN algorithm for Feature Extraction and FastKNN for classification in the figure 7.

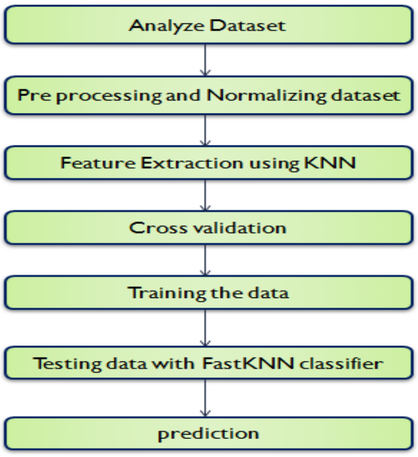

Analyze Dataset
Pre processing and Normalizing dataset
Feature Extraction using KNN
Cross validation
Training the data
Testing data with FastKNN classifier
prediction

Figure 7: Flow chart of a diagnostic procedure

The figure 8 is the representation of the evaluation of performance of the KNN classifier, where it classifies x_1 malignant and x_2 benign classes. The figure 9 is the representation of the improvement of performance in classifying the classes x_1 and x_2 and representing in linear projections in linear space using FastKNN classifier.

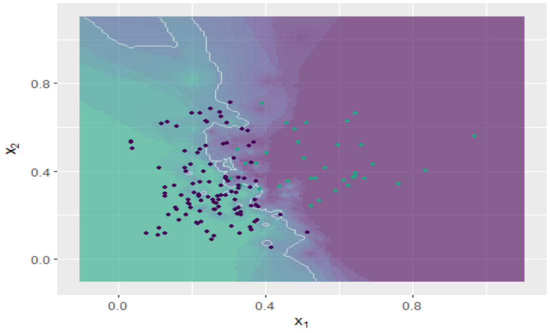


Figure 8: Evaluation of performance

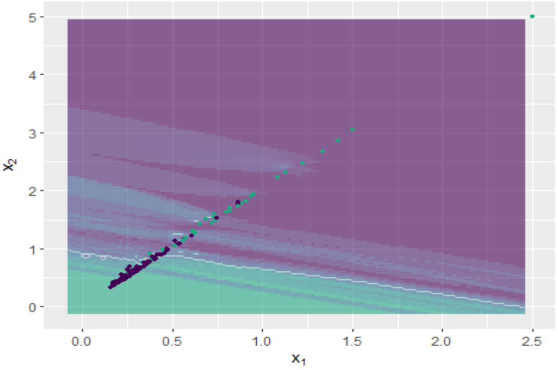


Figure 9: Improvements of performance

IV. COMPARISON OF CLASSIFICATION METHODS

A. GLM:

Generalized linear model uses the linear regression model when the parameter is linear and is normally distributed. If more probability of distribution is needed, then it uses link

DOI : 10.18231/2454-9150.2018.0434

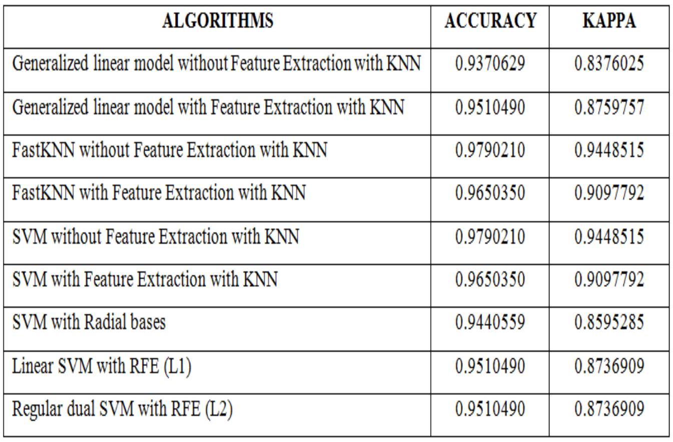
function. There are many link functions depending on the distribution of response. The commonly used link functions are Poisson, Bernoulli, binomial, categorical and multinomial distributions. The model that uses other than linear and binomial is called as logistic regression model. GLM uses OSL (Ordinary Least Squares) fitting method to estimate unknown parameter values. The GLM with logistic regression avoids overfitting. It has restrictions of data for distribution of probability as linear or logistic regression.

B. SVM:

For a data having high dimensions, SVM (Support Vector Machine) is effective method for classification. It takes each attribute as a point in dimensional space S. Then it finds the hyper-plane that divides the different classes with the maximum distance from hyper-plane and the nearest point known as margin. It is efficient classification method when there is a clear separation of classes with margin. It takes high computation time for large dataset and it is difficult to perform when there is more noise in data.

C. SVM-RFE:

SVM-RFE (Support Vector Machine with Recursive Feature Elimination) is a feature selection technique, which recursively removes irrelevant features using ranking rule. It takes the training data, performs SVM classification and computes ranking to all features using weight vectors. Then it eliminates the features that having less rank. It will not identify the redundant features and cannot classify non linear data accurately. It is also difficult to find the weights for non linear kernel.



ALGORITHMS	ACCURACY	KAPPA
Generalized linear model without Feature Extraction with KNN	0.9370629	0.8376025
Generalized linear model with Feature Extraction with KNN	0.9510490	0.8759757
FastKNN without Feature Extraction with KNN	0.9790210	0.9448515
FastKNN with Feature Extraction with KNN	0.9650350	0.9097792
SVM without Feature Extraction with KNN	0.9790210	0.9448515
SVM with Feature Extraction with KNN	0.9650350	0.9097792
SVM with Radial bases	0.9440559	0.8595285
Linear SVM with RFE (L1)	0.9510490	0.8736909
Regular dual SVM with RFE (L2)	0.9510490	0.8736909

Table 1 Results of the different classification techniques obtained on the dataset

From the table 1, we compare the performance of various classification methods for diagnosing the dataset, such as KNN method for Feature Extraction and GLM (general linear model), SVM (support vector machine), FastKNN classifiers for classification. We examined the results of individual methods of classification with and without Feature extraction using KNN and also the result of SVM-RFE (support vector machine with recursive feature elimination) feature extraction method.

While comparing the results, GLM with and without feature extraction using KNN method has accuracy of 95% which is less than the FastKNN and the SVM classifiers. The computation time for SVM-RFE is high than the KNN, but the accuracy of FastKNN is similar to the SVM classifier methods with decreasing of computational time.

V. CONCLUSION

For the analysis of data, KNN method and SVMRFE method is used for Feature Extraction and classification methods such as GLM, FastKNN classifier, and SVM classifier are performed individually and the results are computed for each classifier. So in this aspect Breast Cancer dataset is considered for feature analysis. From the observations of results FastKNN classifier with KNN Feature Extraction model would give better result when compared with GLM with KNN Feature Extraction and also it is observed that KNN Feature Extraction is having more or less similar accuracy to SVMRFE Feature Extraction method. Hence there is an improvement in accuracy for prediction with minimize the error rate and decreasing of computational time using KNN method.

VI. REFERENCS

[1] Mykola Pechenizkiy Dept. of Computer Science and Information Systems, The Impact of Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5

[2] Sandeep Kaur, Dr. Sheetal Kalra(2016), Feature Extraction using support vector machine in disease prediction

[3] Xiaobo Li, Xue Gong, Xiaoning Peng and Sihua Penga, SSiCP: a new SVM based Recursive Feature Elimination Algorithm for Multiclass Cancer Classification

[4] Zong-Xia Xie, Qing-Hua, and Da-Ren Yu(2006),Improved Feature Selection Algorithm Based on SVM and Correlation

[5] Chen-An Tsai, Chien-Hyun Huang,Ching-Wei Chang, Chun-Houh Chen(2012), Recursive Feature Selection with Significant Variables of Support Vectors

[6] Wen-Jyi Hwang and Kuo-Wei Wen,(1998) Fast-KNN classification algorithm based on partial distance search

[7] Olivier Cuisenaire and Benoit Macq, FAST K-NN classification with an optimal k-distance transformation algorithm

[8] Manish Sarkar and Tze-Yun Leong Application of K-Nearest Neighbors Algorithm on Breast Cancer Diagnosis Problem

[9] Y. Song, J. Huang, D. Zhou et al., Informative K-nearest neighbor pattern classification, in: Proceedings of the PKDD, Springer-Verlag,2007, pp. 248–264. IOS press, Amsterdam.

[10] T. O. M. Mitchell, Machine Learning, McGraw-Hill, New York, 1997

DOI : 10.18231/2454-9150.2018.0434