UE17CS303
MACHINE LEARNING

Aniketh S Hegde	PES120170104
Ruthu G S	PES120170856
Shamitha K	PES120170973

Separating Stars from Quasars: Machine Learning Investigation Using Photometric Data

ANIKETH S HEGDE
Computer Science
PES University
anikethhegde15@gmail.com

RUTHU GS
Computer Science
PES University
gsbindu183@gmail.com

SHAMITHA K
Computer Science
PES University
shamitha2511@gmail.com

***Abstract*-**- The K Nearest Neighbor (KNN) method has widely been used in the applications of data mining and machine learning due to its simple implementation and distinguished performance. The value of k is automatically determined, is varied for different data, and is optimal in terms of classification accuracy. The construction of the model reduces the dependency on k and makes classification faster.

I. Introduction

K - Nearest Neighbor (KNN) is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. KNN is a Supervised learning technique 'K' in KNN is the number of nearest neighbors used to classify or (predict in case of continuous variable/regression) a test sample [1]. K-NN doesn't have a training phase as such. But the prediction of a test observation is done based on the K-Nearest Neighbors. The performance of KNN is seen in Section [III].[2] However, to apply KNN we need to choose an appropriate value for k, and the success of classification is very much dependent on this value. In a sense, the KNN method is biased by k. There are many ways of choosing the k value, but a simple one is to run the algorithm many times with different k values and choose the one with the best performance.

For high-dimensional data (e.g., with number of dimensions more than 10) dimension reduction is usually performed prior to applying the *k*-NN algorithm in order to avoid the effects of the curse of dimensionality. The curse of dimensionality in the *k*-NN context basically means that Euclidean distance is unhelpful in high dimensions because all vectors are almost equidistant to the search query vector (imagine multiple points lying more or less on a circle with the query point at the centre the distance from the query to all data points in the search space is almost the same) [3].

Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables [4],Principle Component Analysis (PCA) applied on the dataset to reduce dimensionality to 2D to project rough visualization of decision Boundary in 2D which in detail seen in Section [V].

II. Performance Measures

A. Accuracy

Accuracy is the ratio of number of correct predictions to the total number of input samples.
Accuracy = (TP+TN)/(TP+TN+FN+FP)

Recall

Recall is the fraction of the total amount of relevant instances that were actually retrieved.
Recall = (TP)/(TP+FN)

Precision

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances [5]
Precision = (TP)/ (TP + FP)

F1 Score

F1 score combines precision and recall relative to a specific positive class -The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0[6]
F1score = (2*((precision*recall)/ (precision + recall)))

III. K-Nearest Neighbor

A. Binary Classification Using K-Nearest Neighbor

Binary Classification in K – Nearest Neighbor (KNN) is actually framed using Feature Space Euclidean Geometry, making it a parameter-free learning algorithm [7]

B. K-Nearest Neighbor Algorithm

1) Calculate "d(X,Xi)" i=1,2,3,……………..,n; where d denotes the Euclidean distance between the points.

2) Arrange the calculated n Euclidean distances in non-decreasing order.

3) Let K be a +ve integer, take the first K distances from this sorted list.

4) Find those K- points corresponding to these K-distances.

5) Let Ki denotes the number of points belonging to the ith class among K points i.e.; K>=0.
6) if Ki>Kj for all i not equal to j then put x in class i[8].

IV. Problem Statement

This dataset consists of Galex and sdss photometric data which should be used in classifying stars and quasars

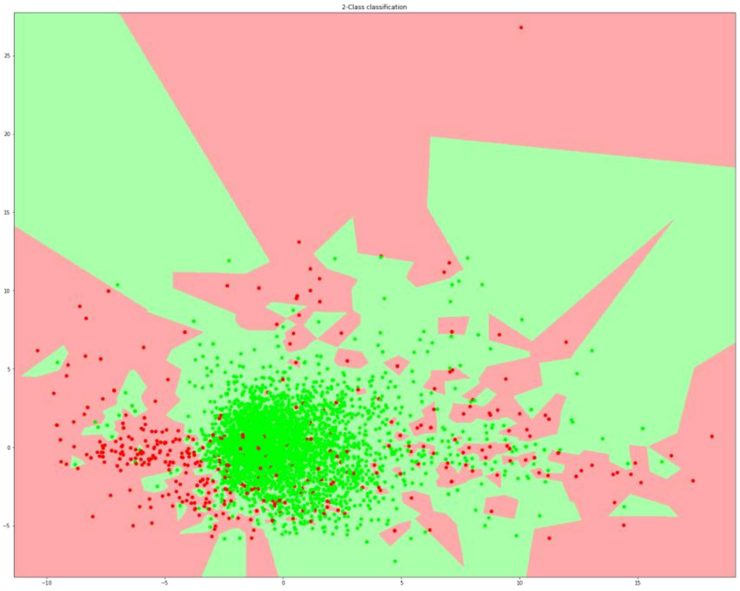
V. Experiment And Results

This Classification problem deals with separating stars and quasars. The analysis is performed using K-Nearest Neighbors (KNN).

The dataset is read and 60% of the dataset is considered as train data, 20% as validation and as 20% test data. For every data point in train set, the Euclidean distance from the test data point is calculated. Using the list of neighbors and corresponding distance, the K “Nearest” Neighbors are found. Prediction of the class level of that point is then done by taking the class level occurring the most times in the K nearest neighbors.
The correlation between columns are found. If a pair of columns have more than 95% correlation, one of them is dropped out. This reduces the dimension of the dataset to 25.

Prediction of labels is done for every value in the validation set using k value from 3 to square root of total number of instances by 2.
In the experiment class 0 has been considered as negative outcome and class 1 has been considered as positive outcome. Using these, the accuracy is calculated for every value of k. The best value of k is one for which the accuracy is the highest and thus obtained k is used for classification on the test data.

Principle Component Analysis (PCA) has been used to reduce Dimensionality from 37 to 2. So, that it is possible to plot approximate decision boundary on 2-Dimensional graph. Decision Boundary obtained after dimensionality reduction using PCA and KNN algorithm with K= 1 is plotted with np.meshgrid.
The Scatter Plot of predicted labels is plotted on the meshgrid. This graph provides a approximate visualization of the misclassification and correct prediction of the K-Nearest Neighbor (KNN) algorithm on the given dataset on a 2-Dimensional plane.



2-Dimensional plane
X -axis ->PCA 1
Y -axis ->PCA 2

Result
The classification performed using K-Nearest Neighbors (KNN) on this dataset results in 90.2% accuracy and 94.7% F1 Score.

VI. References

[1] Wikipedia https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[2] Springer https://link.springer.com/chapter/10.1007/978-3-540-39964-3_62

[3]KNN Wikipedia https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[4]Geeks For Geeks https://www.geeksforgeeks.org/ml-principal-component-analysispca/

[5] Wikipedia https://en.wikipedia.org/wiki/Precision_and_recall

[6] Ritchieng https://www.ritchieng.com/machinelearning-f1-score/

[8] KNN algorithm https://dataaspirant.com/2016/12/23/k-nearest-neighbor-classifier-intro/

.