Classification Of Stars And Quasars

ML FINAL PROJECT REPORT

(AUG-DEC 2019)

Pushpavathi K N

PES1201701347
Computer Science
PES University
Bangalore

Swathi M

PES1201700256
Computer Science
PES University
Bangalore

Sneha Nemadi

PES1201701333
Computer Science
PES University
Bangalore

GITHUB LINK:https://github.com/swatzshetty/Classification-stars-and-quasars-knn

Abstract— **Major challenges of large-scale photometric surveys is the separation of the different classes of sources, especially stars and quasars. Both types of sources have a compact optical morphology and are hence difficult to separate without spectroscopic data. In such cases, we are considering the other parameters of the sources such as their optical variability or their optical colors are necessary to distinguish between stars and quasars. To classify these two, we have used KNN model (Supervised Learning model). To evaluate the correctness of this classifier, we report the accuracy.**

Keywords—K- Nearest Neighbors (KNN)

I. Introduction

Machine learning involves the use of statistics to make useful predictions and to learn essential features; classification is the process of marking separations between categories in data. Supervised machine learning techniques make use of labelled data to make predictions of future unseen data. In the present context, the labels in the data comprise of numeric indications of a source being a star or a quasar. We are using KNN, a machine learning classifier to classify these stars and quasars without spectroscopic data, by using other parameters.

II. Methodology

A. Preprocessing of data.

We have dropped the unnamed , galex_objid ,sdss_objid, spectrometric_redshift , pred columns.
From the dataset given as they don't contribute to knowing the distances between two instances.we are dropping redshift column as its difficult to observe those values for galaxical objects.

B. Choosing our model.

KNN

k*-nearest neighbors algorithm** (k*-NN**) is a non-parametric method used for Classification and Regression.
A k-nearest-neighbor algorithm, often abbreviated k-nn, is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are in.

The k-nearest-neighbor is an example of a "lazy learner" algorithm, meaning that it does not build a model using the training set until a query of the data set is performed.

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).
We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

C. We chose KNN algorithm

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.

To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret Output
2. Calculation Time
3. Predictive Power

KNN algorithm fairs across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time.

D. Working of KNN algorithm.

K nearest neighbor algorithm is very simple. It works based on minimum distance from the query instance to the training samples to determine the K-nearest neighbors to be the prediction of the query instance.

Distances are calculated by using

1.Euclidean distance

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$

2.Manhatten Distance.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

E. Choosing K value in KNN

A small value of k means that noise will have a higher influence on the result and a large value make it computationally expensive. Its advisable to choose as an odd number if the number of classes is 2 and another simple approach to select k is set k=sqrt(n).

we are choosing k = 7 for all tests after examining the accuracies for k=5,7,9.

F. Pseudocode of KNN algorithm

Implementing a KNN model by following the steps below:

1. Load the data

2. Initialise the value of K.
3. For getting the predicted class, iterate from 1 to total number of training data points:
    - Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
    - Sort the calculated distances in ascending order based on distance values.
    - Get top k rows from the sorted array.
    - Get the most frequent class of these rows.
    - Return the predicted class.

G. Results of our model

We tested our model on catalog1,catalog2 for k values 5,7,9 and came to the conclusion of using k=7.
The results of this test are as follows:
For cat1:
k=5 Accuracy: 98.65546218487%
k=7 Accuracy: 98.15126050420%
k=9 Accuracy: 97.058823529411%

We trained and tested our model on Catalog3 for different splits.(k=7)
The following are the accuracies we got:

Split	Accuracy %	Precision %	Recall %
80-20	98.1045752	98.66667	97.496706
70-30	97.516339	98.17708	96.915167
60-40	96.503268	97.117963	95.769993

Split	Class 0 accuracy	Class 1 accuracy
80-20	98.702983139	97.496706193
70-30	98.138297872	96.915167095
60-40	97.220426632	95.7699933906

we trained our model using k-fold cross validation with 5 folds.The scores are as follows:
Scores: (For 5 folds)
[93.65598430346633,
95.16023544800524,
93.5251798561151,
94.6370176586004,
94.44081098757357]
Mean Accuracy: 94.284%

Testing results of cat1,cat2,cat4 for k=7 :

catalog	Accuracy %	class 0 Accuracy %	class 1 Accuracy %
1	98.15126	100.00	96.30252
2	97.29018	99.93806	94.6423
4	87.302265	82.313354	92.458161

References:

[1] Separating Stars from Quasars: Machine Learning Investigation Using Photometric Data,Simran Makhija,*, Snehanshu Saha, Suryoday Basak, Mousumi Das.
[2] Machine Learning Mastery website.
[3] https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/