Classification of Stellar Dataset, SDSS17, using KNN and SVC

**Abstract** – Stellar classification is one of the most important aspects of astronomy. This paper discusses the classification of a Stellar Dataset provided by the Sloan Digital Sky Survey, using supervised learning methods: k-nearest neighbors and support vector classifier. The raw data consists of 18 columns and 100,000 records. The features include spectral characteristics, such as ultraviolet filter, green filter, red filter, etc. These can be combined and used to determine the origin of the light, which could be either a Star, a Galaxy, or a Quasar. The large dataset is beneficial in training the model to get a higher accuracy but also has an impact on the runtime. Hence, this report will be discussing the method used to improve the performance along with the details of data pre-processing, feature selection, libraries used from Scikit-learn to aid our classification and model optimization for increasing accuracy.

1. **INTRODUCTION**

With datasets as huge as this one, and increasing in size as you read this, it has become very important to use automated techniques for stellar classification. In supervised learning, both KNN and SVC are well equipped to tackle this objective.

*Understanding Stars, Quasars and Galaxies1*

We all know that we are revolving around a massive star, flying through space at the speed of 2.1 million km/hr! While the sun is a star, i.e., a burning ball of gas, the rest of the planets, their satellites, asteroids etc, together make up our galaxy, the Mily Way. While the distinction between the two is clear, we also have another class, which is called Quasar. Scientists believe that quasars are younger galaxies. These have been found very far from our galaxy, but their numbers do increase as we approach the edge of the observable universe. Quasars are considered separate from galaxies as they are much brighter and contain much higher amount of radiation. Termed as “*active galactic nucleus*”, the main cause of radiation has been found to be an active

supermassive black hole in the centre of the quasars. An average Quasar would be approximately 27 trillion times brighter than our sun and about 1000 times brighter than our Milky Way. To put this in perspective, a quasar at a distance of Pluto from earth, would evaporate all of Earth’s water in ~1/5th of a second.

Scientists use various data points for classification, which includes redshift, which is the displacement/shift of the light’s wavelength towards the red part of the spectrum, which is highly affected by the travel distance. Along with redshift, we will also be using other spectral characteristics, i.e., ultraviolet, green light, red light, near infrared and infrared.

**OBJECTIVES**

* Use of supervised learning methods for stellar classification
* Use data exploration to find corelation between variables
* Improve the performance of the used models w.r.t. turnaround time and accuracy
* Optimize the accuracy of KNN by finding the right ‘k’ value
* Use confusion matrix and the ROC curve for tracking accuracy

1. **LITERATURE REVIEW**

Let us now go through some papers that have delved into machine learning methods for stellar classification and similar objectives.

The paper titled “*Identifying galaxies, quasars, and stars with machine learning: A new catalogue of classifications for 111 million SDSS sources without spectra*” by A. O. Clarke, A. M. M. Scaife, R. Greenhalgh and V. Griguta [2], speaks in detail about the usage of a random forest classifier to a dataset of 111 million unlabelled records. The model was trained on a dataset of 1.55 million labelled sources, which clearly shows the efficiency that such models can work with. Along with classification of sources, the team aimed to investigate of class imbalance and implications of transfer learning, in cases where the recorded magnitudes might be lesser as compared to the training set. The paper was successfully able to establish the correlation of photometry (redshift and other filters mentioned earlier) to the classification of the sources while achieving an F1 scores of 0.990, 0.953, and 0.977 for galaxies, quasars, and stars, respectively, and classification probabilities greater than 0.9. The paper was successfully able to establish that the model worked as expected and maintained the F1 score, on applying to fainter sources, when at least 50% of the training set was brighter.

Another paper named “*Photometric identification of blue horizontal branch stars*” by K. W. Smith, C. A. L. Bailer-Jones, R. J. Klement and X. X. Xue [3], looks into using machine learning methods, i.e., KNN and SVM and KDE for identifying blue horizontal branch stars (extremely hot stars with low luminosity), using photometric data. The aim of the project was to compare the performance of the three aforementioned techniques. They also delve into the role of prior probabilities in performance of the classifier and the effect of contamination. They established that ultraviolet, green and red-light measurements were most important in classifying blue horizontal branch stars. SVM was the best performing model as per their results and had the least amount of contamination.

In another paper “*Unsupervised star, galaxy, QSO classification*” by C. H. A. Logan and S. Fotopoulou [4], the researchers aimed to used HDBSCAN, which is an unsupervised hierarchical clustering algorithm. This project also uses similar photometric data. The researchers suggest that machine learning can viably support the unprecedented volume of data that we can expect in the near future, to reliably classify sources, among other problems. The paper also evaluates several dimension reductions algorithms and finds PCA to give the most optimum results. They also used additional data to improve their accuracy, and got the final results as 98.9, 98.9, and 93.13.

Finally, in the paper, “*Determining spectroscopic redshifts by using k nearest neighbor regression*” by S. D. Kügler, K. Polsterer and M. Hoecker [5], the aim was to determine redshifts in spectra for emission and absorption features by using KNN regression technique. The paper also advocates of using machine learning techniques handling the huge datasets form the likes of “Large Sky Area Multi-Object Fiber Spectroscopic Telescope” (LAMOST) that could collect billions of datapoints in a fraction of second. The team was able to correctly estimate the redshift, along with future estimation of spectra with high precision using KNN. An important implication of being able to estimate future redshifts would be the reliable identification of rare objects that show a change in spectral features over time.

1. **DATA PROCESSING**

**Data source and description**

The dataset was found on Kaggle and was released by Sloan Digital Sky Survey, that has been surveying since 2000.

The data consists of 100,000 observations of space and contains 17 columns along with the “class” column that categorizes the data into stars, galaxies and quasars. The photometric data consists of:

* u = Ultraviolet filter in the photometric system
* g = Green filter in the photometric system
* r = Red filter in the photometric system
* i = Near Infrared filter in the photometric system
* z = Infrared filter in the photometric system
* redshift = redshift value based on the increase in wavelength
* alpha = Right Ascension angle
* delta = Declination angle

As seen in the literature review, these photometric datapoints are the most important part of classification. Apart from these, we also have:

* spec\_obj\_ID = Unique ID used for optical spectroscopic objects (this means that 2 different observations with the same spec\_obj\_ID must share the output class)
* plate = plate ID, identifies each plate in SDSS

Remaining fields (identification and date) are: obj\_ID, run\_ID, rerun\_ID, field\_ID, MJD (modified julian date), fiber\_ID and cam\_col (Camera column to identify the scanline within the run).

The data did not have any null values, so we move on to feature selection.

**Feature Selection**

From the dataset, the id and date fields were first taken out as they are used for identification of SDSS runs only. As mentioned earlier, the photometric data (alpha, delta, u, g, r, i, z and redshift) was maintained as it plays the most important role in classification. The pairplot sufficiently shows the distinction between the classes using the above-mentioned features. For instance:

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated ‘*redshift*’ X ‘*alpha*’ ‘*redshift*’ X ‘*delta*’

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated‘*redshift*’ X ‘*u*’ ‘*redshift*’ X ‘*i*’

QSO Galaxy Star

For spec\_obj\_ID, plate and cam\_col, boxplots were used to further confirm if they are able to help us differentiate between the 3 classes. Below are the plots for each:

* Chart, box and whisker chart

  Description automatically generatedcam\_col:
* Chart, box and whisker chart

  Description automatically generatedspec\_obj\_ID
* Chart, box and whisker chart

  Description automatically generatedplate:

As it can bee seen from the plots, both plate and spec\_obj\_ID show differentiation in classes while cam\_col does not. Hence, plate and spec\_obj\_ID is maintained in the list of features. After checking and confirming the correlation between the selected features, the total count of is 10.

**External Libraries Used**

* Pandas
* Numpy
* Matplotlib/Seaborn
* Scikit-learn

**Data normalization**

In the raw data, we can see a huge difference in the values of individual features. For instance, for a particular record, the value of “i” is 19.16573 while that of spec\_obj\_ID is 6.543777e+18. This huge difference in values has a major impact on classification techniques like KNN and severely decrease their accuracy. Also, given the large size of the dataset, numbers at this scale will slow down the model. Hence, the data was standardized using *StandardScaler* from the *sklearn.preprocessing* library. This resulted in a significant increase in accuracy, which will be discussed later in the report.

1. **LEARNING METHODS**

For the purpose of classification of the stellar dataset, we will be using k-nearest neighbors and support vector classifier.

**K-nearest neighbors**

KNN is a supervised learning technique that uses all the available data points to classify each data point based on their nearest neighbors. The number of nearest neighbors to be considered, i.e. ‘k’ can be changed to better fit the data. The prediction algorithm first calculates the distance from x to all points in the data and sorts the distance in increasing order. Then, based on the majority of the labels of the ‘k’ closest points, the label of x is assigned. We can see in the below illustration [7], how the value of k would impact the prediction of the red datapoint.

Diagram, schematic

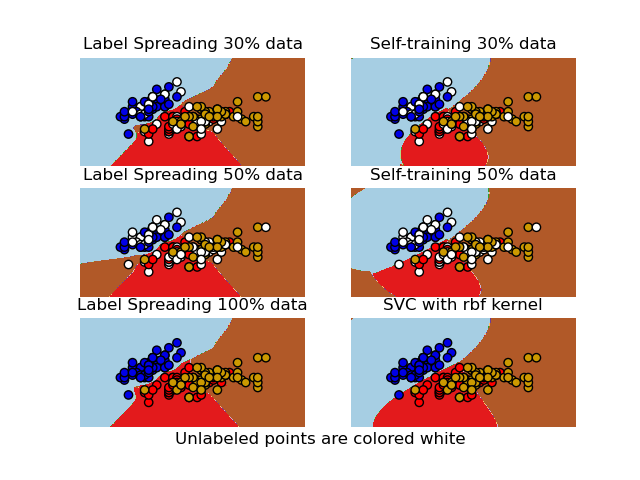
Description automatically generated

**Justification for use of KNN**

The variation in the dataset for each class is somewhat low and KNN is known to have low bias. This enables KNN to maintain low bias-low variance in both training and test data, and hence produce higher accuracy. KNN is also regarded as “well-suited” for classification of low dimensional data. Since we have only 10 features, it isn’t very complicated for KNN algorithm to calculate the distances and store all of them for classification.

**Support Vector Classifier**

SVC is another supervised learning technique that works very well a large dataset. The SVC algorithm separates the classes by maximizing the distance between the datapoints and the hyperplane. The hyperplane can be tuned by using *gamma* and *c*. *Gamma*is usedfor non-linear hyperplanes. A higher value of gamma will overfit the model on the training set while a lower value will underfit. *C*isthe value that controls the trade-off between classification of training points and a smoother decision boundary. The below illustration explains the hyperplane and class distinction [8].



**Justification for use of SVC**

SVC works very well with large datasets such as ours. In this method, the model is more dependent on a subset of points rather than trying to include any outliers. This allows the model to generalize well and reduces the risk of overfitting, in turn producing more accurate results for large datasets while also working well new data classification.

1. **ANALYSIS, TESTING AND RESULTS**

As discussed earlier, and seen in feature selection, the photometric data plays a major role in classification of data. Redshift has the highest correlation with our classes as compared to the other features.

**RESULTS**

For stellar classification, both the techniques have performed well and classified the sources into the correct classes i.e., stars, quasars and galaxies. As previously mentioned, the data was first cleaned by removing unwanted columns (ids, dates). For feature selection, pairplot, boxplot and correlation were used, leading us to a set of 10 features. In the next steps, the data was split into 70% training and 30% test sets and run through the KNN model once without standardizing and then after standardizing.

**KNN Results**

The KNN model was implemented by using *KNeighborsClassifier* from the *sklearn.neighbors* library. However, it is important to find the optimum *‘k’* value for the KNN model. To achieve this, a loop was setup to check values of k between 1-40 (a higher upper limit could be used but that would mean a much larger time tradeoff). Upon running the loop and plotting the results, the below results were seen:

Chart, scatter chart

Description automatically generated

Using this plot, we can clearly identify that the optimum value of k = 3. Moving ahead with this value, the model was run and the result upon the first run had an f1-score of 0.69. The reason behind this was that the data wasn’t yet standardized. Hence, standardization was done by using *StandardScaler* from the *sklearn.preprocessing* library. After this, the KNN model with k = 3 was rerun and the f1-score improved to 0.95. This was expected as classification techniques benefit significantly from data normalization. Results of the confusion matrix and classification report: Table

Description automatically generated with medium confidence

A screenshot of a cell phone

Description automatically generated with medium confidence

**SVC Results**

Since the data had already been standardized, the SVC model was implemented by using *SVC* from the *sklearn.svm* library. Upon running the model, SVC achieved a slightly higher f1-score of 0.96. Results as below:

Calendar

Description automatically generated with low confidence

A screenshot of a cell phone

Description automatically generated with medium confidence

The score is high, but it could be optimised further using *GridSeacrh* to optimize the values of *gamma* and *c.* However, given the size of the dataset the tradeoff with time for running the gridsearch would be exponentially high. Hence, I decided not to use it in this model.

1. **CONCLUSION**

We have successfully trained and implemented two supervised learning methods, i.e., KNN and SVC on a stellar dataset of 100,000 records with significantly high accuracy score with optimising our classification model. This achieves the set goal for the project. It was also established that data standardization plays a very important role in improving the efficiency of classification models.

Upon comparing the two models used in this project, I found that a standalone KNN model runs much faster than SVC. However, finding the right value of k is crucial, which takes a long time. Upon optimising the k value, the KNN model reached an accuracy of 0.94. On the other hand, SVC was able to achieve a slightly higher score of 0.96, without further optimisation of gamma and c. Hence, there might be room for some improvement but as mentioned earlier, the trade off with time would be enormous. In conclusion, the time taken in our case ended up being similar for both models. However, SVC achieved a higher score and was much simpler to implement owing to no optimisation. One thing to note is that both models’ lowest accuracy was in quasars. Exploring additional/other features might help improving these scores.

In the longer run, it seems that SVC will be a better fit for classification of stellar datasets. The optimization of gamma and c only needs to be run once on the training data and can be used for a long time before needing to re-optimize. The model will also be more accurate than KNN as we saw. KNN also suffers from the “curse of dimensionality” and cannot handle high dimensional data. Hence, it isn’t the most future-proof model. On the other hand, KNN can be used for anomaly detection i.e., finding rare objects using photometric data. The algorithm for KNN makes it faster in searching for outliers.

Understanding the universe is a colossal task but it starts by understanding its most abundant constituents, i.e., stars, quasars and galaxies. As suggested by all cited papers as well, machine learning can make notable difference in research times, and someday, hopefully soon, will become the basis of space exploration.

1. **REFERENCES**
2. Quasar properties:

<https://earthsky.org/astronomy-essentials/definition-what-is-a-quasar/>

1. *Identifying galaxies, quasars, and stars with machine learning: A new catalogue of classifications for 111 million SDSS sources without spectra* by A. O. Clarke, A. M. M. Scaife, R. Greenhalgh and V. Griguta

<https://doi.org/10.1051/0004-6361/201936770>

1. *Photometric identification of blue horizontal branch stars* by K. W. Smith, C. A. L. Bailer-Jones, R. J. Klement and X. X. Xue

<https://doi.org/10.1051/0004-6361/201014381>

1. *Unsupervised star, galaxy, QSO classification* by C. H. A. Logan and S. Fotopoulou

<https://doi.org/10.1051/0004-6361/201936648>

1. *Determining spectroscopic redshifts by using k nearest neighbor regression* by S. D. Kügler, K. Polsterer and M. Hoecker

<https://doi.org/10.1051/0004-6361/201424801>

1. Stellar Classification dataset from Kaggle:

<https://www.kaggle.com/fedesoriano/stellar-classification-dataset-sdss17>

1. KNN illustration:

[https://miro.medium.com/max/1400/0\*jqxx3-dJqFjXD6FA](https://miro.medium.com/max/1400/0*jqxx3-dJqFjXD6FA)

1. SVC illustration:

<https://scikit-learn.org/stable/_images/sphx_glr_plot_semi_supervised_versus_svm_iris_001.png>