**Report**

**Stellar Classification**

**Name: Tharun Prabhakar**

**PS. No: 40027784**

**Table of Contents**

|  |  |
| --- | --- |
| **S.NO** | **Title** |
| 1 | **Problem Statement** |
| 2 | **About Dataset** |
| 3 | **Algorithm Used** |
| 4 | **Experimentation/Implementation** |
| 5 | **Results** |
| 6 | **Conclusion** |

**Problem Statement**

In astronomy, stellar classification is the classification of stars based on their spectral characteristics. The classification scheme of galaxies, quasars, and stars is one of the most fundamental in astronomy. The early cataloguing of stars and their distribution in the sky has led to the understanding that they make up our own galaxy and, following the distinction that Andromeda was a separate galaxy to our own, numerous galaxies began to be surveyed as more powerful telescopes were built. This dataset aims to classify stars, galaxies, and quasars based on their spectral characteristics.

**About Dataset**

The data consists of 100,000 observations of space taken by the SDSS (Sloan Digital Sky Survey). Every observation is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy or quasar.

1. obj\_ID = Object Identifier, the unique value that identifies the object in the image catalogue used by the CAS
2. alpha = Right Ascension angle (at J2000 epoch)
3. delta = Declination angle (at J2000 epoch)
4. u = Ultraviolet filter in the photometric system
5. g = Green filter in the photometric system
6. r = Red filter in the photometric system
7. i = Near Infrared filter in the photometric system
8. z = Infrared filter in the photometric system
9. run\_ID = Run Number used to identify the specific scan
10. rereun\_ID = Rerun Number to specify how the image was processed
11. cam\_col = Camera column to identify the scanline within the run
12. field\_ID = Field number to identify each field
13. 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)
14. class = object class (galaxy, star or quasar object)
15. redshift = redshift value based on the increase in wavelength
16. plate = plate ID, identifies each plate in SDSS
17. MJD = Modified Julian Date, used to indicate when a given piece of SDSS data was taken
18. fiber\_ID = fiber ID that identifies the fiber that pointed the light at the focal plane in each observation

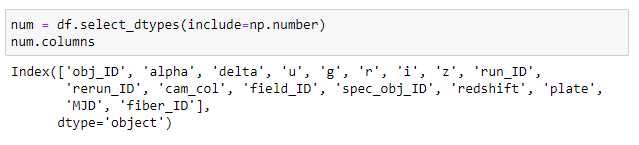
**Description of Dataset:**

The dataset has 1 lakh rows and 18 columns with 17 numerical columns and 1 categorical column that is the predictor variable or the class variable. The name as well as data type of the column is shown in the diagram below.

**Graphical user interface, text, application

Description automatically generated**

Numerical data types include obj\_ID, alpha, delta, u, g, i, r, z, rerun\_ID, cam\_col, field\_ID, spec\_obj\_ID, redshift, plate, MJD, fiber\_ID



Categorical data types include class column

Graphical user interface, text, application

Description automatically generated

The Class column has 3 unique values with Galaxy with highest frequency having 59445 counts followed by Star with 21594 and Quasar with 18961. This shows that the data is imbalanced.

Chart, bar chart

Description automatically generated

Count Plot show the count of each class

Chart, pie chart

Description automatically generated

Pie Chart representation of the Class variable

**Distribution of Data:**

Distribution of Numerical variables has been checked and variables follow normal distribution. Ideal dataset should have all features to be normally distributed, if not appropriate treatment has to performed to get most accurate result from the model

Chart, line chart

Description automatically generated

Chart

Description automatically generated

Chart, histogram

Description automatically generated

Text

Description automatically generated

Skewness values representation

Skewness values of the numerical variables indicate mild skewness with u, g and z having highly negative skewness and plate, redshift and field\_ID having positive skewness. This might be the cause of outliers or extreme values. For treatment of skewness normal transformation or Standard transformation using Standard Scalar or z-score scalar.

**Heatmap:**

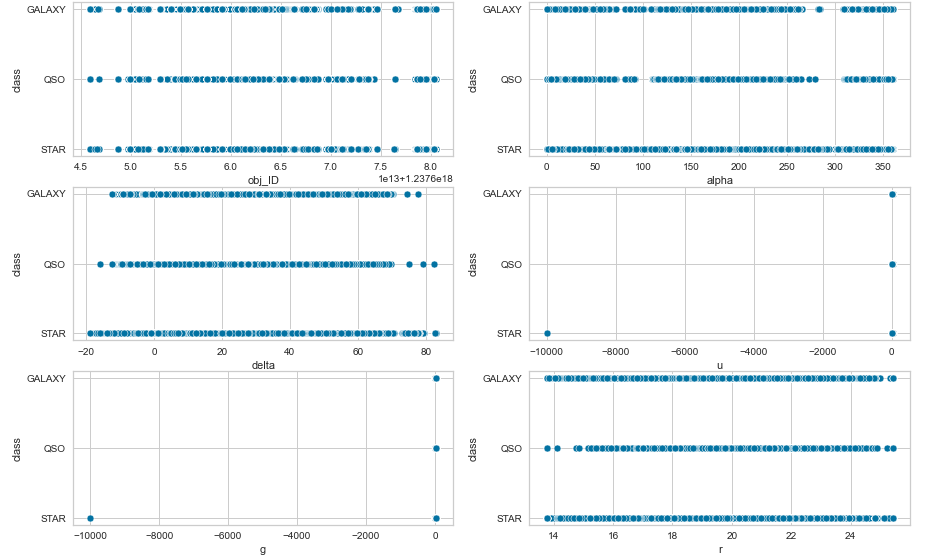
The Heat map was plotted, and correlation matrix has been derived. The Heatmap is generally used to check for multicollinearity between the independent/predicting variables. More positive value the matrix has more positively related are the variables and more negative the value then more negatively correlated are the variables and when it’s zero it means that the variables are not correlated. From the matrix we can see that rerun\_ID has high correlation with all the variables because of its singular value for all the data making it insignificant variable. Multicollinearity can be seen in the data meaning the predicting variables are affected by themselves, this might lead to improper predictions or inaccurate results to the overall model. The heatmap is shown below for reference.

A picture containing text, scoreboard

Description automatically generated

Heatmap Representation of Numerical Variables

**Distribution of datapoints between classes:**



Chart

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Distribution Graph between class labels and variables

From above graphs we can see how data points are distributed. Since rerun\_ID has only one value it occurs at the same point for all classes. Rest of the variables have equally distributed values for all classes

**Algorithms Used**

* **Support Vector Machine:**

SVM is the most popular and widely used supervised learning algorithm that is used for both Classification and Regression problems, but most predominantly used in Classification Problems. The basic idea of this algorithm is to best fit boundary the best classifies the data and has optimal space for future data. The best fit line or boundary is known as Hyperplane. The datapoints from different class with the least distance from each other is used to create the hyperplane. These data points are known as support vectors. There two types of SVM Linear and Non-Linear. Linear kind is used when the data can be split linearly by a line, that is if the dataset has two classes. Non-Linear kind is used when data cannot be differentiated by a line, mostly when the dataset has more than two classes. This dataset uses Non-Linear SVM kernel ‘RBF’ radial based function which tries to fit non-linear hyperplane.

* **Decision Trees:**

Decision trees is a popular supervised algorithm used in regression and classification problems. This algorithm builds trees in top to bottom approach from root at the top and branches or leaves at the bottom. Decision Trees uses features based on their importance and the feature with the most importance forms the root node and tree gets built with features in descending order the final node or feature is known as terminal or leaf node. Each node denotes a condition on a feature value. Each branch represents the outcomes of the condition. The outcome nodes are called the child node.

* **Random Forest:**

Random Forest is an Ensemble technique that uses bagging principles where homogeneous models be built independently, and their outputs are aggregated at the end. Random in Random Forest means the subsets of variables that are selected at random to build a decision tree and Forest means forest of decision trees. Random Forest consists of several independent decision trees that operate as an ensemble. The output of each tree is aggregated to obtain the result. A bootstrap sample is used to train the model. Bootstrap sample means random sampling with replacement and observations that are not selected is known as out-of-bag samples (OOB). Random Forest builds trees of stumps that is trees with one root node and two child nodes.

* **SMOTE:**

SMOTE analysis is used when the dataset is imbalanced that one class has more majority than the other. Imbalanced data is handled by Up sampling minority class, Down sampling majority class, changing the performance metrics, trying synthetic sampling approach or use different algorithm. SMOTE (Synthetic Minority Oversampling Technique) is one the most used techniques to deal with imbalanced data. It generates synthetic samples for the minority class. One drawback of SMOTE is it doesn’t work well with high dimensional datasets. It can either be used to remove majority class observations to balance the dataset but may lead to bias or be used to oversample to increase minority class observations but tends to produce duplicate records that tends to overfit the data.

**Experimentation**

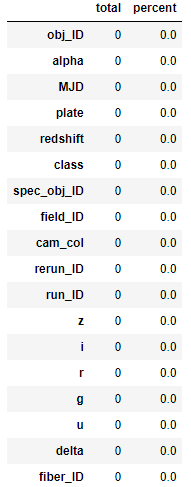
**Missing Value/Null Value Treatment**

**Text

Description automatically generated**

Missing and null value treatment is done so that model can get meaningful data and get trained as model can’t interpret null values. Hence proper treatment should be done before further processing of data. If a column or feature has more than 60% of null values, it is considered as insignificant feature and hence can be removed from training the model. For numerical data type the missing or null values can be replaced

either by mean or median based on the distribution of feature. If a feature has outliers that can be seen using boxplot the missing values are replaced by median and if the feature does not have outliers, it can be replaced by mean and mean gets heavily influenced by the presence of outliers. For categorical feature mode of that feature is used that is the category with highest frequency will be used to replace missing and null values.



From the above table we can see that there are no null or missing values in the dataset, so treatment is not needed.

**Outliers’ treatment:**

When there are extreme values for a feature that are outside the interquartile range (IQR) the feature is said to outliers. A boxplot is usually used to represent and check for outliers. An ideal dataset should have features with no outliers because outliers impact the models training and accuracy of the model as some models are ineffective against outliers. Outlier treatment is usually done either by removing the data or rows containing outliers. The drawback of this method is that important information or data that has high influence might get removed causing the model to get poor training and underperform. This method is done when there are mild or low outliers and not suitable for extreme outliers. Another way to treat is by clipping the value so that no information is lost. This is done by modifying the extreme values to 75th quartile is its more than it or to 25th quartile if its less than that. So, all values are between the interquartile range (IQR). This method is used when there are more outliers.

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

From the above boxplot representation, we can see that there are outliers present in r, i, redshift, field\_ID hence these features need to be treated before doing train test split. Clipping of values is done because there are many outliers.

Text

Description automatically generated

A function is created that gets the dataframe and column that needs to be treated as input. The 25th and 75th quantile value is obtained for that feature and interquartile range is calculated from which the upper limit and lower limit are calculated. If the value is an extreme value, then it is replaced by either upper if the value is greater than upper limit or lower value if it is less than lower limit.

**Chart, box and whisker chart

Description automatically generated**

**Chart, box and whisker chart

Description automatically generated**

**Chart, box and whisker chart

Description automatically generated**

**Chart

Description automatically generated**

Features after outlier treatment

**Statistical Analysis of Variables:**

Statistical analysis of variables is usually done to check for significant variables which usually eliminates high levels of dimensionality and to improve the performance of the model. For categorical variables Chi2 test is done that checks if the class variable depends on independent variables and for numerical variables One-way ANOVA is done which checks the mean distribution of feature in turn used to tell if the variable is significant or not.

Text

Description automatically generated

Since the dataset contains numerical variables, One-way ANOVA is used to check the significance of the variables. The p-value is compared with the level of significance of 0.05% and if its less than 0.05 the variable that it’s a significant and if its more the variable is insignificant.

Table

Description automatically generated with medium confidence

Table containing List of features and their significance

From the above table we can see that delta and rerun\_ID are insignificant variables and needs to be removed



So, the final dataset has 15 columns to predict the class labels.

**Encoding of Class variable:**

Categorical variables need to be encoded so that it is recognised by the algorithm for training and testing. For encoding encoders can be used such as Label Encoder and Ordinal Encoder from sklearn.preprocesing library. In this dataset Label encoder has been used to change the categorical variables to numerical

Graphical user interface, text, application, email

Description automatically generated

**Train-Test split:**

Train-test split is done to split the data into training set and test set. Train-test split is imported from sklearn.model\_selection. The testing set is used for testing the performance of the model and training set is used for training the model. For large dataset the test size is usually 5% of the total data and for small dataset usually it is 25% or 30%.

Text

Description automatically generated

Since the dataset has 100000 data 95000 is used as training set and rest 5000 is used as test set for testing purpose. Random state is used while performing train-test split so that the samples are selected at random and not on order.

**Model Building:**

After the completion of pre-processing of data model building takes place. Three base models were built SVM, Decision Tree, Random Forest.

For SVM kernel used was ‘RBF’ because linear cannot be used for dataset with 3 class labels and ‘RBF’ Radial Basis Function fits a nonlinear hyperplane to the dataset and predictions were made and the results were stored.

Text

Description automatically generated

For Decision Tree the criterion used was ‘Gini’ for getting the information gain and arranging of features in order of importance. All features of the final dataset have been used to build the Decision Tree.

Text

Description automatically generated

For Random Forest the criterion used was ‘gini’ like Decision Tree and with 100 estimators that is number of trees to be built to train the model and rest options with default values.

Text

Description automatically generated

All the models use random state value as 10 so that samples while training is like all the models so that they get trained similarly.

Smote was then used to up-sampling so that least class labels get their count increased similar to the majority class. The drawback of this method is that data duplication can occur which might provide inaccurate results.

Graphical user interface, text

Description automatically generated

Chart, bar chart

Description automatically generated

Before Using SMOTE

Chart, bar chart

Description automatically generated

After Using SMOTE

From the above graph we can see that count of minority class has increased.

After SMOTE the same models are built with the same condition and random state to check if the performance has increased or not.

**Results**

**Confusion Matrix**

The confusion matrix is used check how the model has performed in predicting by comparing the actual labels and gives information of misclassified predictions.

**Receiver Operating Characteristics and Area Under Curve:**

ROC is a performance metric that is used to evaluate how good a model is performing. It uses Truth Positive Rate and False positive Rate and uses it to plot the graph. A threshold line is given for reference an ideal curve should be away as much as possible from the curve and the AUC score should be high as much as possible. As seen from the diagram below the graph in the first column indicates how well the model predicts when AUC score is 1 that is it able to separate the classes accurately. As the AUC score reduces, we can see that there will some misclassifications in the predictions and as the score reaches 0.5, we can see that the model cannot differentiate between the classes.

Diagram

Description automatically generated with medium confidence

Types of ROC curve

**Training Score:**

The training score gives the accuracy score of how the model has predicted. The accuracy score is usually between 0 to 1. An ideal model should have an accuracy score of 1 but generally a score of 80 and above is preferred for a good model. Training score is the accuracy of the model in predicting train data with labels

**Test Score:**

The test score gives the accuracy score of how the model works with test data.

**Results of Base SVM:**

**Table

Description automatically generated with low confidence**

Confusion Matrix

**Chart, treemap chart

Description automatically generated**

Pictorial Representation of Confusion Matrix

Chart, line chart

Description automatically generated

Base SVM ROC curve

Training Score:

Graphical user interface, text, application

Description automatically generated

Test Score:

Graphical user interface, application

Description automatically generated

**Results of Base Decision Tree:**

Text, table

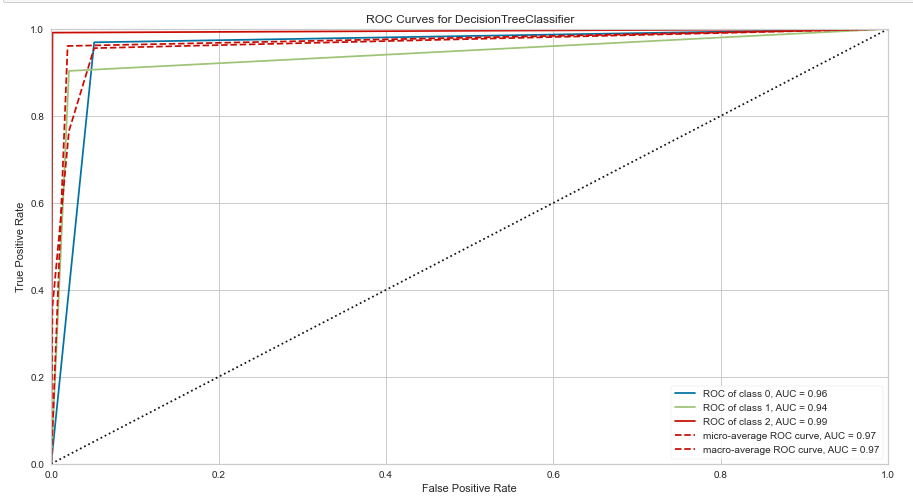
Description automatically generated

Confusion Matrix

Chart, treemap chart

Description automatically generated

Graphical representation of Confusion Matrix



Base Decision Tree ROC curve

Training Score:

Graphical user interface, application, Word

Description automatically generated

Test Score:

Graphical user interface, application, Word

Description automatically generated

**Base Random Forest Results:**

Graphical user interface, text, table

Description automatically generated

Confusion Matrix of Base Random Forest

A picture containing chart

Description automatically generated

Pictorial Representation of Confusion Matrix of Base Random Forest

Chart, scatter chart

Description automatically generated

ROC curve of Base Random Forest

Training Score:

Graphical user interface, text

Description automatically generated

Test Score:

Graphical user interface, application

Description automatically generated

**SMOTE SVM Results:**

Graphical user interface, text

Description automatically generated

Confusion Matrix

A picture containing background pattern

Description automatically generated

Pictorial Representation of confusion matrix

Chart, line chart

Description automatically generated

ROC curve of SMOTE SVM

Training Score:

Graphical user interface

Description automatically generated

Test Score:

Graphical user interface, text

Description automatically generated

**SMOTE Decision Tree Result:**

Graphical user interface, text, application

Description automatically generated

Confusion Matrix

A picture containing square

Description automatically generated

Pictorial Representation of Confusion Matrix

Chart, scatter chart

Description automatically generated

SMOTE Decision Tree ROC curve

Training Score:

Graphical user interface, application

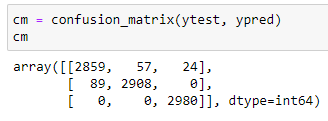
Description automatically generated

Test Score:

Graphical user interface, text, application

Description automatically generated

**SMOTE Random Forest Results:**



Confusion Matrix

A picture containing application

Description automatically generated

Pictorial Representation of Confusion Matrix

Chart, line chart, scatter chart

Description automatically generated

SMOTE Random Forest ROC curve

Training score:

Graphical user interface, application, Word

Description automatically generated

Test Score:

Graphical user interface

Description automatically generated with medium confidence

**Conclusion**

From the graphs and accuracy score of SVM base model we can see that the model isn’t efficient enough to give accurate result meaning there will be misclassifications. The confusion matrix shows us that the model predicts all the data mainly as Galaxy and 4 observations as Stars. The ROC curve of SVM base model also tells us that the there are misclassifications while predicting. The accuracy of SVM base model was around 0.59 for both test and train meaning that even though the predictions are wrong they are consistent in the result. After the performance of SMOTE the models score and results reduced because increase in data points made the algorithm difficult to predict or categorise with the help hyperplane. Hence the score has reduced indicating the poor performance of the model.

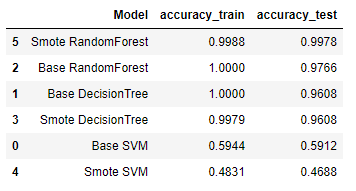
From the graphs and accuracy score of Decision Tree base model we can see that the model is performing remarkably with training data and well as with test data with accuracy score of 1.0 for training set and 0.9608 for test data. The ROC curve produced also shows that model performs good. From observing the confusion matrix, we can see that there very low misclassifications and has predicted the class labels well with 2863,898 and 1043 predictions of Galaxy, Stars and Quasars respectively. After the SMOTE operation we can see that the accuracy score has been slightly reduced may be due to some duplication of data cause by SMOTE but with an overall improvement ROC curve. The confusion matrix too has good classifications of 2794,2869,2974 actual predictions of Galaxy, Star and Quasars respectively.

The Graphs and accuracy of score of Random Forest base model informs that it works very well with training and test data having scores of 1.0 for train set and 0.9766 for test set. The confusion matrix has high actual predictions compared to the rest of the models. ROC curve also has maximum AUC in the base model. After SMOTE the models actually has improved performance making Random Forest models the best out of the 6 models.

The reason for the failure of the SVM model might be due to high dimensionality, multicollinearity making it difficult for the hyperplane to be produced. Due to the overall performance of Random Forest models being high, the two models can be selected as he best fit models.

**Selecting the best fit model:**

The best fit model is the one that not only has high accuracy for training set but also for the test set. The difference between the training set and test set should also be as low as possible so that we don’t select an overfit model as our best fit model.



From the above result we can see that SMOTE based Random Forest model is the best fit model as it as least difference between the train and the test accuracy and also has the highest score for the test set.

**References:**

* [Stellar Classification Dataset - SDSS17 | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17)
* [Support Vector Machine (SVM) Algorithm - Javatpoint](https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm)
* [Python | Decision tree implementation - GeeksforGeeks](https://www.geeksforgeeks.org/decision-tree-implementation-python/?ref=lbp)
* [Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu | Towards Data Science](https://towardsdatascience.com/understanding-random-forest-58381e0602d2)
* [SMOTE for Imbalanced Classification with Python (machinelearningmastery.com)](https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/)